Documentation 8.0

ZABBIX

29.10.2026

Contents

ユーザーマニュアル **5**

Zabbix ドキュメントへようこそ **5**
- Zabbix を始める . . . 5
- Zabbix Cloud . . . 5
- モバイルアプリ . . . 5
- 開発者センター . . . 5
- コミュニティとその他のリソース . . . 5
- インストールと最初の手順 . . . 6
  - Web インターフェースのインストール . . . 6
  - クイックスタートガイド . . . 11
  - インストール: 追加ガイド . . . 85
  - アップグレード . . . 166
  - 既知の問題 . . . 182
  - 要件 . . . 191
- 新機能とアップグレードに関する注意事項: 8.0 . . . 201
  - テンプレート . . . 206
  - メディアタイプ . . . 208
  - アイテム . . . 209
  - プラグイン . . . 209
  - ディスカバリと自動登録 . . . 210
  - ローレベルディスカバリ . . . 210
  - データベース . . . 210
  - プロセス . . . 211
  - セキュリティ . . . 211
  - 認証 . . . 212
  - ウィジェット . . . 212
  - フロントエンド . . . 213
  - ドキュメント . . . 215
- アップグレードノート . . . 215
  - 破壊的変更 . . . 216
  - その他 . . . 217
  - 新機能とアップグレードに関する注意事項: 8.0.x . . . 219
- Zabbix アプライアンス . . . 219
- プロセス . . . 223
  - サーバー . . . 225
  - プロキシ . . . 251
  - エージェント . . . 297
  - エージェント 2 . . . 327
  - Java ゲートウェイ . . . 389
  - 送信者 . . . 393
  - 取得 . . . 396
  - JS . . . 396
  - Web サービス . . . 398
  - MCP サーバー . . . 400
  - ブリッジアダプター . . . 407
  - 環境変数 . . . 411
- 設定 . . . 413
  - ホストとホストグループ . . . 423
  - アイテム . . . 441
  - トリガーによる障害検出 . . . 661

可視化 . . . 712
テンプレートとテンプレートグループ . . . 736
標準搭載のテンプレート . . . 736
イベント発生時の通知 . . . 750
マクロ . . . 802
ユーザーとユーザーグループ . . . 817
スケジュールされたレポート . . . 827
データエクスポート . . . 830
設定のエクスポート/インポート . . . 839
正規表現 . . . 904
イベントと解決 . . . 910
障害の確認と抑制 . . . 913
障害の手動クローズ . . . 916
メンテナンス期間 . . . 918
タグ付け . . . 922
イベント相関 . . . 925
ディスカバリ . . . 932
ネットワークディスカバリ . . . 932
アクティブエージェントの自動登録 . . . 941
ローレベルディスカバリ . . . 945
仮想マシンの監視 . . . 969
VMware 監視アイテムキー . . . 971
仮想マシンのディスカバリキーのフィールド . . . 989
VMware アイテムの JSON 例 . . . 993
VMware 監視設定の例 . . . 997
Web 監視 . . . 1001
Web 監視アイテム . . . 1011
実際のシナリオ . . . 1012
サービス監視 . . . 1019
サービスツリー . . . 1019
SLA . . . 1023
セットアップ例 . . . 1025
サービス監視のアップグレード . . . 1029
暗号化とセキュリティ . . . 1030
証明書を使用した暗号化 . . . 1038
事前共有鍵を使用した暗号化 . . . 1046
データベースへの安全な接続 . . . 1049
Web インターフェースとサーバー間の安全な接続 . . . 1060
アクセス制御 . . . 1064
暗号化 . . . 1068
Web サーバーの保護 . . . 1069
シークレットの保存 . . . 1072
Google Chrome TLS 証明書の信頼 . . . 1081
セキュリティのベストプラクティス . . . 1082
Web インターフェース . . . 1083
メニュー . . . 1083
Web インターフェースのセクション . . . 1093
ユーザー設定 . . . 1356
グローバル検索 . . . 1360
Web インターフェースのメンテナンスモード . . . 1362
ページパラメータ . . . 1363
定義 . . . 1365
追加の Web インターフェース言語 . . . 1366
独自のテーマを作成する . . . 1367
デバッグモード . . . 1368
Zabbix で使用される Cookie . . . 1369
タイムゾーン . . . 1370
パスワードのリセット . . . 1371
時間期間とホストセレクター . . . 1371
設定パラメータ: Zabbix Web インターフェース . . . 1373
API . . . 1379
メソッドリファレンス . . . 1384
付録 1. 参考文献の解説 . . . 2261

付録 2. 7.4 から 8.0 への変更点 . . . 2267
付録 3. 8.0 の変更点 . . . 2269
拡張機能 . . . 2270
ロード可能なモジュール . . . 2272
Web インターフェースモジュール . . . 2281
付録 . . . 2282
インストールとセットアップ . . . 2282
アイテム . . . 2329
サポートされる関数 . . . 2369
マクロ . . . 2369
バージョン互換性 . . . 2409
時間期間の構文 . . . 2410
単位記号 . . . 2411
コマンド実行 . . . 2412
エスケープの例 . . . 2413
Zabbix とは . . . 2415

Zabbix Cloud **2419**

Zabbix Cloud **2419**
Zabbix Cloud のセットアップと管理 . . . 2419
Zabbix Cloud について学ぶ . . . 2419
クラウドで Zabbix をデプロイする . . . 2419
ノード設定 . . . 2429
概要 . . . 2430
概要タブ . . . 2431
アクセスフィルタータブ . . . 2434
暗号化タブ . . . 2435
バックアップタブ . . . 2435
履歴タブ . . . 2436
メンテナンスタブ . . . 2436
アップグレードタブ . . . 2437
ユーザーの追加 . . . 2439
はじめに . . . 2439
組織へのユーザー追加 . . . 2439
ノードへのユーザー追加 . . . 2440
Zabbix Cloud の主な違い . . . 2441
はじめに . . . 2441
監査ログ . . . 2443
Zabbix Cloud プロキシをセットアップする . . . 2445

モバイルアプリ **2453**

モバイルアプリ **2453**
アプリケーション用に Zabbix をセットアップする . . . 2453
アプリについて学ぶ . . . 2453
アプリ用に Zabbix を設定する . . . 2454
アプリの概要 . . . 2455
セキュアなモバイルアクセス . . . 2463

開発者センター **2464**

開発者センター **2464**
Zabbix で構築する . . . 2464
プラグインの作成 . . . 2464
プラグインインターフェース . . . 2470
Zabbix 用 Python ライブラリ . . . 2472
インストール . . . 2472
クイックスタートガイド . . . 2472
Zabbix API の使用 . . . 2476
Zabbix エージェントからデータを収集する . . . 2479
Zabbix サーバーまたはプロキシへのデータ送信 . . . 2480
デバッグログ . . . 2484
モジュール . . . 2484
モジュールファイルの構造 . . . 2485

ウィジェット . . . 2492
チュートリアル . . . 2501
例 . . . 2528
拡張機能開発における変更点 . . . 2529

Zabbix man ページ **2529**
zabbix_agent2 . . . 2529
zabbix_agentd . . . 2530
zabbix_get . . . 2532
zabbix_js . . . 2534
zabbix_proxy . . . 2535
zabbix_sender . . . 2537
zabbix_server . . . 2541
zabbix_web_service . . . 2544

著作権表示 **2544**

ユーザーマニュアル

Zabbix ドキュメントへようこそ

Zabbix 監視の基本的なセットアップから高度な設定まで、Zabbix を扱うためのリソースです。このマニュアルでは、Zabbix のインストール、設定、運用に必要なすべてを説明しています。

Zabbix 8.0 の新機能

Zabbix を始める

インストール　さまざまなオペレーティングシステムやデータベース構成に対応した、お好みのプラットフォームへの Zabbix のインストール手順を説明します。

要件　Zabbix の導入を成功させるために、サポートされているプラットフォームとソフトウェアの前提条件の一覧です。

クイックスタートガイド　最初の設定手順から最初の障害アラートを受け取るまで、基本を順を追って説明する、簡潔でタスク指向のハウツーです。

Zabbix Cloud

Zabbix Cloud の利用を開始する　初期設定や監視対象デバイスの接続など、Zabbix Cloud インスタンスをすぐに使い始めるためのガイド付き手順です。

Zabbix Cloud とオンプレミスの比較　2 つの導入方法の管理、拡張性、保守の違いを比較し、どちらがワークフローをよりシームレスにサポートできるかを学びましょう。

Zabbix Cloud を探る　Zabbix Cloud の概要、主な機能、特典、サブスクリプションに含まれる内容について説明します。セットアップや使用方法に関するよくある質問については、FAQ をご覧ください。

モバイルアプリ

アプリを入手　Zabbix モバイルアプリ（近日公開）をデバイスにインストールして、外出先でも Zabbix 環境の監視を開始できます。

アプリ用に **Zabbix** をセットアップする　モバイルデバイスから Zabbix に接続できるように、Zabbix ブリッジアダプターをインストールします。次にデバイスを追加し、プッシュ通知を有効にします。

アプリの概要　Zabbix モバイルアプリの基本的な使用方法について説明します。

開発者センター

Web インターフェースモジュール　特定のユースケースに合わせて Zabbix の Web インターフェースを拡張または変更するカスタムモジュールの構築に関するガイドとリファレンス。

ウィジェット　ウィジェットの構造とロジックの詳細、およびニーズに合わせたカスタムダッシュボード要素の作成手順。

プラグイン　Zabbix の機能を拡張したり、外部システムと統合するためのプラグインの開発と管理の概要。

コミュニティとその他のリソース

Zabbix フォーラム　Zabbix のあらゆるレベルの専門知識を持つユーザーが、アイデアを交換し、解決策を見つけ、経験を共有するためのスペースです。

Zabbix ブログ　Zabbix チームとコミュニティの貢献者によるニュース、チュートリアル、事例紹介。

ハウツービデオ　Zabbix を最大限に活用するためのデモンストレーション、ディスカッション、ヒントを紹介するビデオライブラリです。

インストールと最初の手順

Zabbix のセットアップ手順は次のとおりです。

1. 必要なソフトウェアがインストールされ、実行中であることを確認します。
2. Zabbix をインストールします。
3. Zabbix Web インターフェースをインストールします。
4. 監視を開始します。

各手順の詳細は以下に示します。

必要なソフトウェア　Zabbix は、以下のソフトウェア（LAMP スタックと呼ばれることが多い）で適切に動作します。

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

最新の Linux システムの多くでは、PHP がすでにインストールされている場合があります。その場合、データベースサーバーと Web サーバーのセットアップのみが必要になることがあります。

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

Zabbix のインストール　Zabbix をインストールする最も簡単な方法は、ディストリビューションパッケージからインストールすることです（または、コンテナからインストールします）。パッケージによって、必要なソフトウェアの依存関係もインストールされます。

お使いのオペレーティングシステムに合わせた詳しいインストール手順は、**Zabbix Web** サイトで確認できます。

プラットフォームを選択し、表示される手順に従ってください。まずは、Zabbix サーバー、Zabbix エージェント 2、および Web インターフェースをインストールすることをお勧めします。後から Zabbix プロキシや追加のエージェントを追加できます。

インストールプロセスの一環として、データベース（設定データと履歴データを保存するため）を作成し、初期データをインポートします。

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

Web インターフェースがデータベースと通信できるように、データベースのパスワード（前の手順で作成したもの）の入力を求められます。

監視の開始　Zabbix Web インターフェースにログインします。

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

Linux サーバーの監視については、クイックスタートガイドから始めるとよいでしょう。

Web インターフェースのインストール

このセクションでは、Zabbix Web インターフェースのインストール手順について説明します。Zabbix バックエンドのインストール後に、以下の手順を実行してください。

Zabbix Web インターフェースは PHP で記述されており、PHP が設定された Apache や NGINX などのサポート対象 Web サーバーが必要です。必要なソフトウェアとライブラリについては、Web インターフェースの要件を参照してください。

Zabbix Web インターフェースで英語以外の言語を使用するには、PHP gettext 拡張機能とともに、そのロケールを Web サーバーにインストールする必要があります。言語を変更すると、時刻や日付の形式も変更される場合があることに注意してください。

Zabbix Web インターフェースで SSL を設定する方法については、Zabbix Web インターフェースの SSL 設定を参照してください。

インストールウィザード

Web インターフェースをインストールするには、ブラウザで Zabbix フロントエンドの URL を開きます。

- Apache の場合: `http://<server_ip_or_name>/zabbix`
- NGINX の場合: `http://<server_ip_or_name>`

ウェルカム画面

Web インターフェースのインストールウィザードは、ウェルカム画面で開きます。

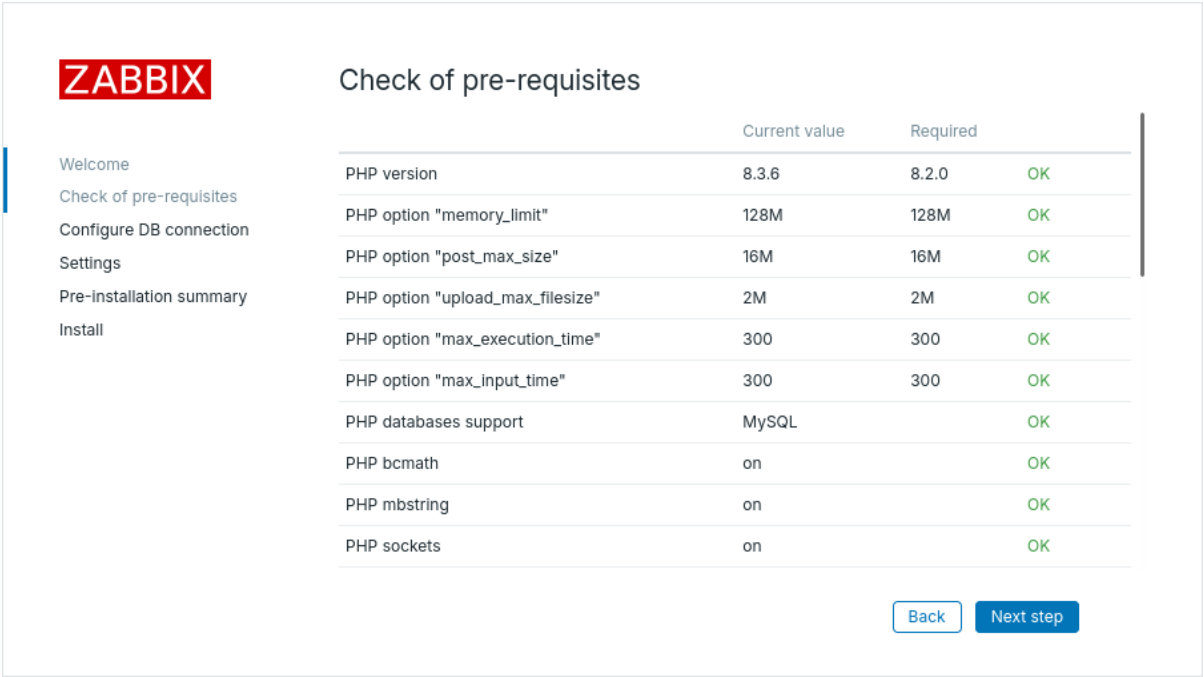


システム全体のデフォルト言語は英語（en_US）で、時刻と日付の形式は米国形式です。システム全体のデフォルト言語を変更するには、デフォルト言語のドロップダウンメニューから目的の言語を選択します。

続行するには、次のステップボタンをクリックします。

前提条件の確認

すべての必須の Zabbix Web インターフェースの前提条件を満たしていることを確認します。



必要な PHP 拡張機能とオプションの PHP 拡張機能、必要な PHP バージョン、およびサポートされているデータベースの一覧については、要件を参照してください。オプションの前提条件を満たしていない場合は *Warning* ステータスで表示されますが、満たしていなくてもセットアップを続行できます。

Web サーバーが使用する `php.ini` ファイルで、PHP 設定ディレクティブに必要な値は次のとおりです。

```
memory_limit = 128M ; 最小値
post_max_size = 16M ; 最小値
upload_max_filesize = 2M ; 最小値
```

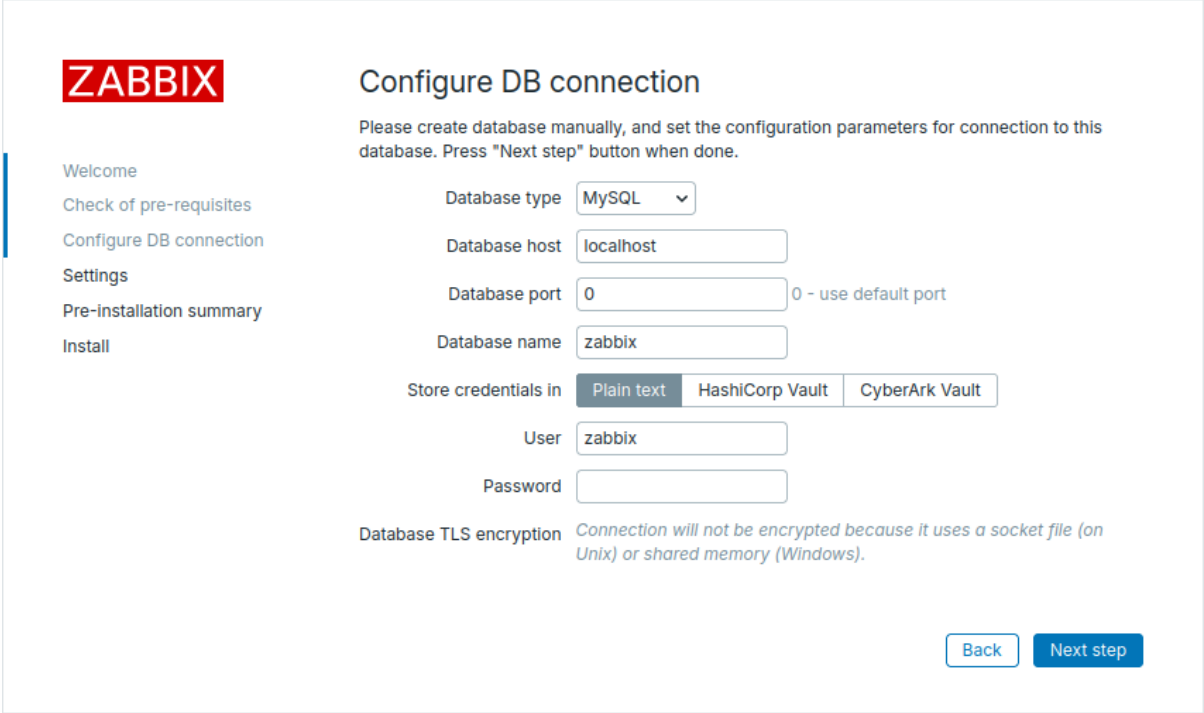
```
max_execution_time = 300 ; 最小値。0 と -1 も使用可能
max_input_time = 300 ; 最小値。0 と -1 も使用可能
session.auto_start = 0 ; 無効にする必要があります
arg_separator.output = "&" ; "&amp;" も使用可能
```

Attention:
Web サーバーのユーザーまたはグループを変更する必要がある場合は、変更後のユーザーまたはグループに PHP セッションディレクトリへの必要な権限があることを確認してください。そうしないと、Zabbix のセットアップを続行できない場合があります。

すべての必須の前提条件を満たしている場合は、次の手順に進みます。

データベース接続の設定

データベースに接続するための詳細を入力します。



Database host フィールドを空のままにすると、デフォルトの Unix ドメインソケットが使用されます。

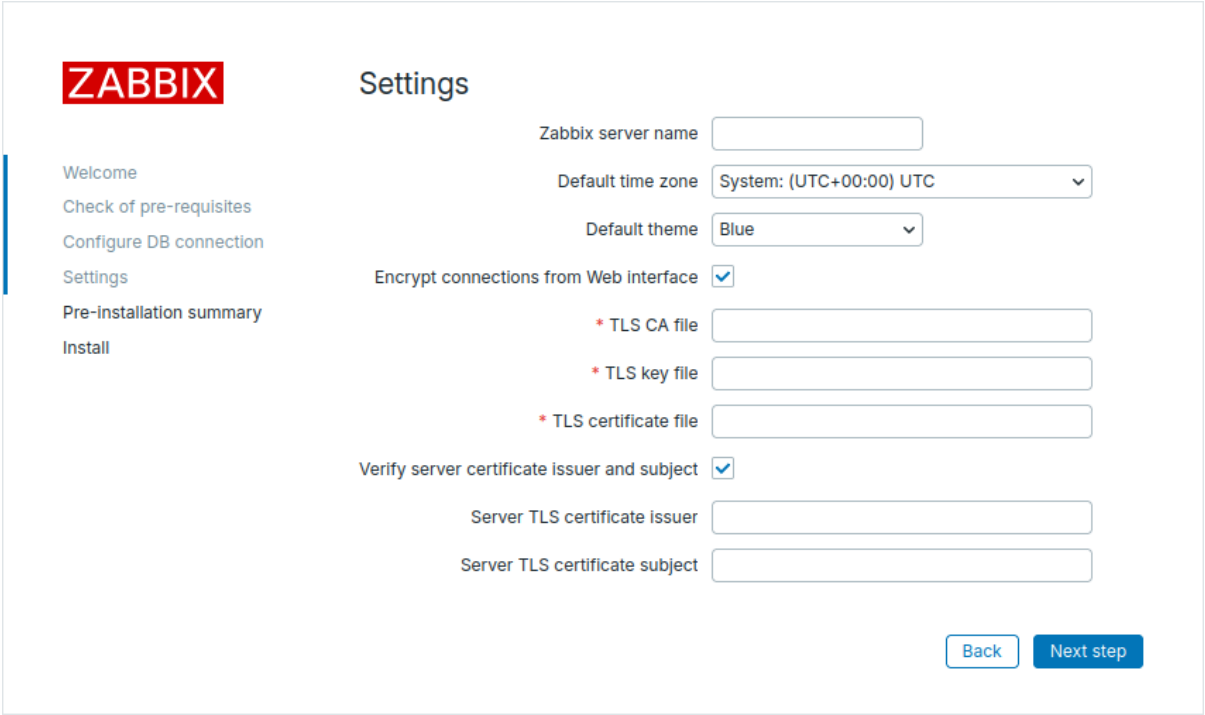
MySQL の場合、*Database host* フィールドに `localhost` を入力しても、デフォルトの Unix ドメインソケットを介して接続されます。ウィザードには、データベースソケットを設定するオプションはありません。カスタムソケットを使用するには、Zabbix サーバーの設定で構成します（たとえば、Zabbix のサーバー設定ファイル（`zabbix_server.conf`）で`DBSocket=` パラメータを設定します）。これにより、Web インターフェースとサーバーからデータベースへの接続設定の整合性が保たれます。

PostgreSQL の場合、*Database host* フィールドには、コンマで区切った複数の `host:port` エントリを指定できます。文字列は分割され、各ポートが *Database port* フィールドと同じ方法で検証されます。検証に成功すると、接続文字列を構成するために、文字列がホストとポートに解析されます。コンマ区切りのリスト内のホストにポートが定義されていない場合は、*Database port* の値がデフォルトのポートとして使用されます。ソケットパス（例: `/var/run/pgbouncer`）が指定されている場合は、代わりにその Unix ドメインソケットが使用されます。

データベースへの安全な接続を設定するには、*Database TLS encryption* チェックボックスを選択し、Web インターフェースからデータベースへに進んでフィールドを設定します。次の場合、*Database TLS encryption* チェックボックスは無効になります。- MySQL の場合、*Database host* に *localhost* が設定されている。- PostgreSQL の場合、*Database host* フィールドの値がスラッシュで始まっている、またはフィールドが空である。

Zabbix が HashiCorp Vault からシークレットを取得するように設定するには、*Store credentials in* で *HashiCorp Vault* を選択し、HashiCorp の設定に進んで展開されたフィールドに入力します。Zabbix が CyberArk Vault からシークレットを取得するように設定するには、*Store credentials in* で *CyberArk Vault* を選択し、CyberArk の設定に進んで展開されたフィールドに入力します。

設定



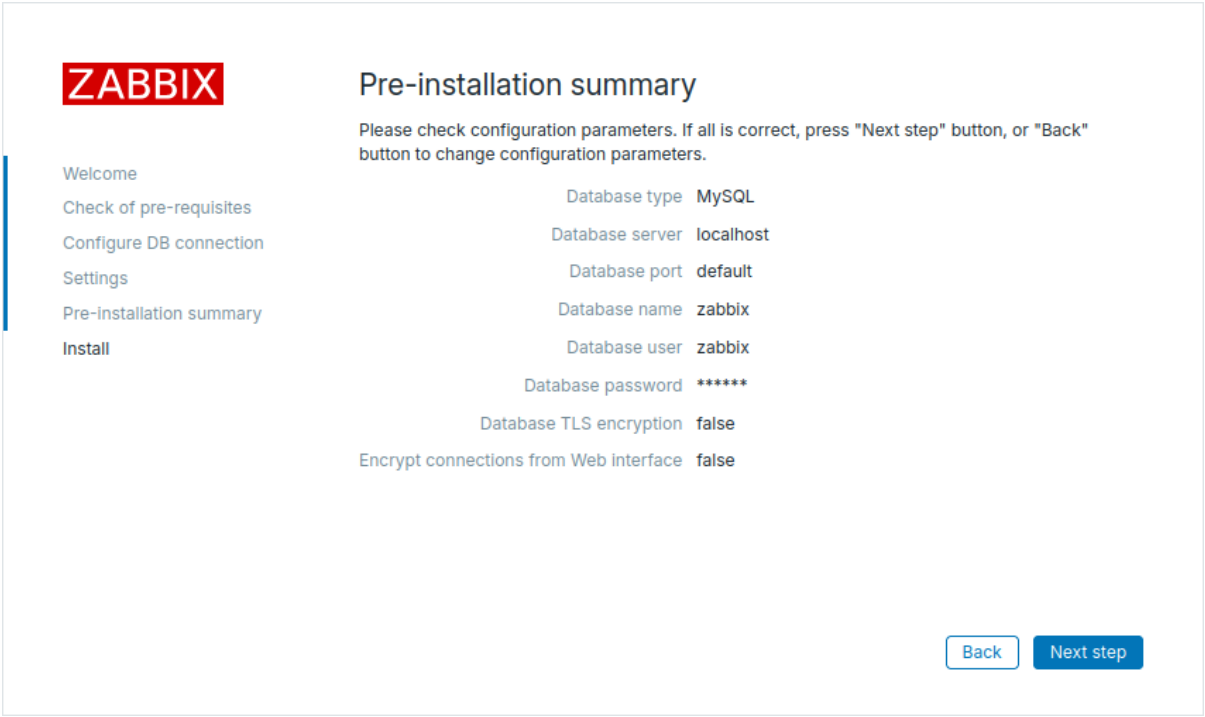
Zabbix サーバーの名前の入力は任意です。ただし、入力した場合はメニューバーとページタイトルに表示されます。

Web インターフェースのデフォルトのタイムゾーンとテーマを設定します。

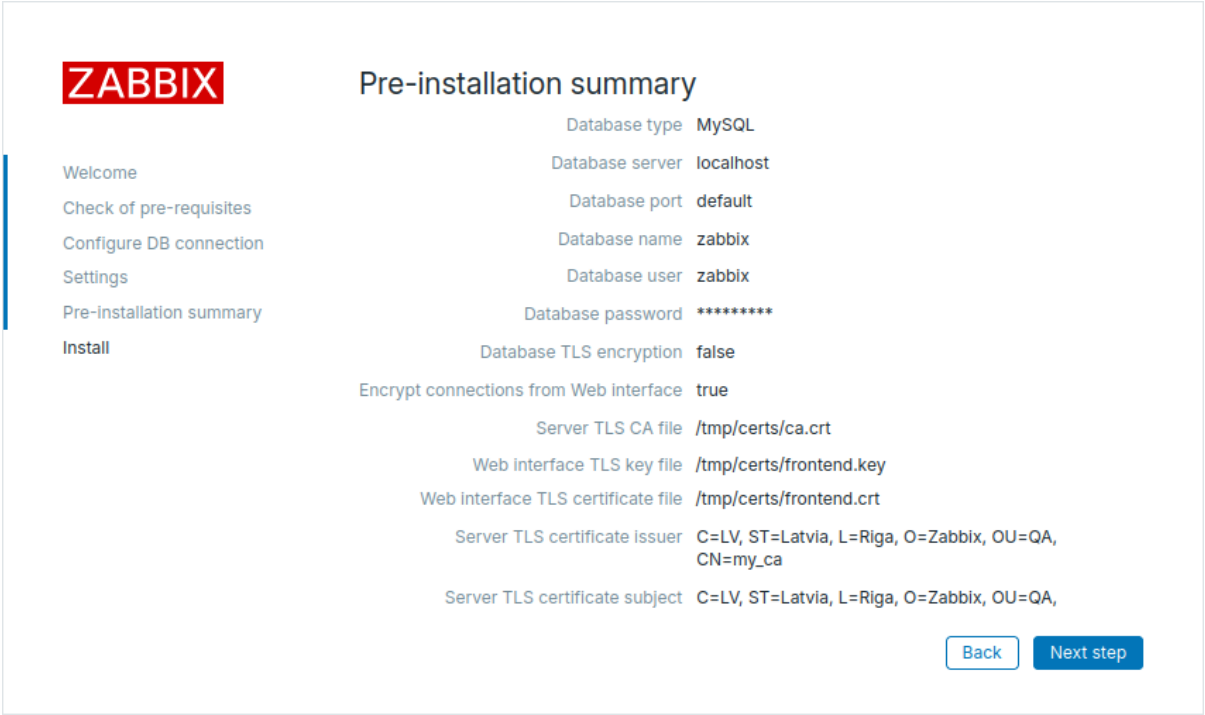
Zabbix Web インターフェースと Zabbix サーバー間の TLS 接続を設定するには、*Web* インターフェースからの接続を暗号化チェックボックスを選択し、TLS 接続の設定に進んで展開されたフィールドに入力します。

インストール前の概要

設定の概要を確認します。



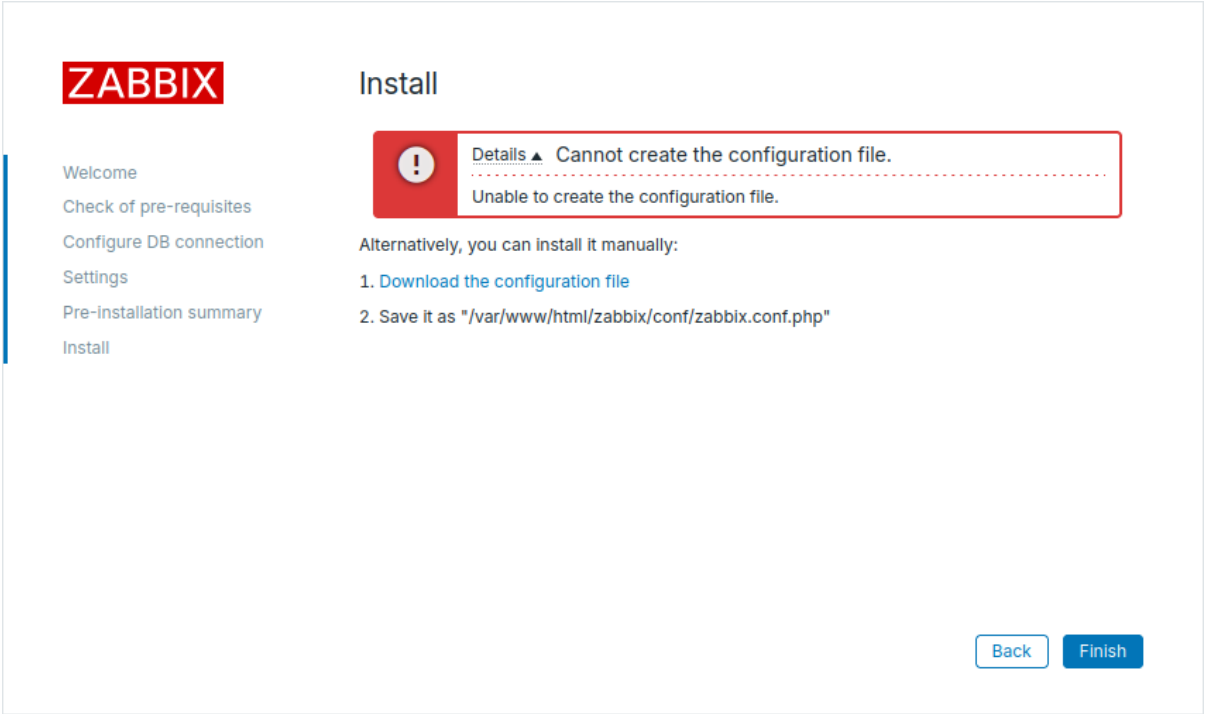
TLS が設定されている場合、ウィザードにその詳細が表示されます。



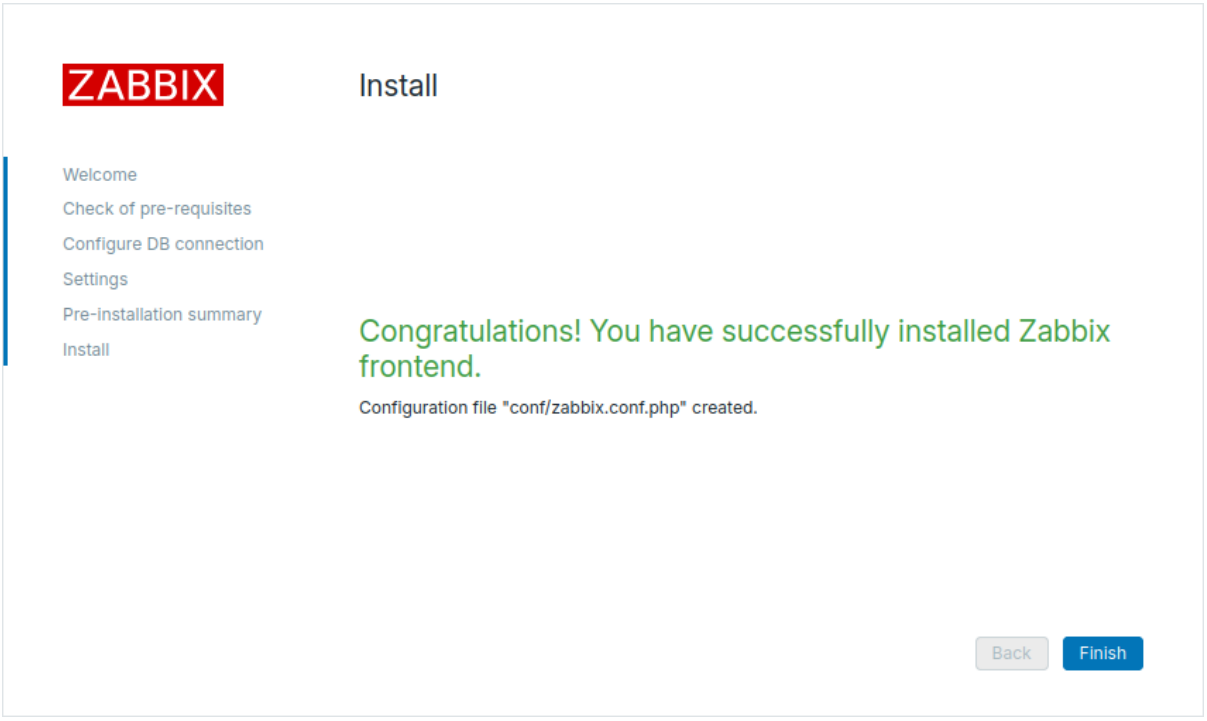
インストール

Web サーバーのユーザーが conf/ ディレクトリへの書き込み権限を持っている場合、設定ファイルは自動的に保存されるため、すぐに次の手順へ進むことができます。

Web サーバーのユーザーが conf/ ディレクトリへの書き込み権限を持っていない場合は、設定ファイルをダウンロードし、Zabbix の PHP ファイルをコピーした Web サーバーの HTML ドキュメントディレクトリ内の conf/ ディレクトリに配置する必要があります。

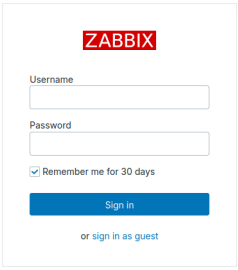


インストールを完了します。



ログイン

デフォルトの認証情報でログインします。- ユーザー名: *Admin* - パスワード: *zabbix*



Web インターフェース設定ファイル

インストール後、Web インターフェース設定ファイル `zabbix.conf.php` のパラメータを更新して、Zabbix Web インターフェースを再設定できます。

設定ファイルのデフォルトパスは、オペレーティングシステムと Zabbix のインストール方法によって異なります。

- パッケージからインストールした場合: `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- ソースからインストールした場合: `/var/www/html/zabbix/conf/zabbix.conf.php`
- アプライアンスの場合: `/etc/zabbix/web/zabbix.conf.php`

Zabbix Web インターフェースの設定方法については、設定ファイルのオプションを参照してください。

クイックスタートガイド

Zabbix をインストールし、そのWeb インターフェースを設定したら、監視を開始できます。

以下のクイックガイドを使用して、一般的なリソースの監視を開始してください。

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

クイックスタートガイドでは、監視に公式の Zabbix テンプレートを使用する方法に重点を置いています。テンプレートは、監視要素をあらかじめ設定したセットです。テンプレートを使用すると監視を拡張できます。1 つの監視対象（デバイス、アプリケーションなど）でテンプレートをテストして調整した後、複数の類似した監視対象に適用できます。

テンプレートをそのまま使用した監視で問題がない場合でも、テンプレートの要素をカスタマイズすることもできます。

テンプレートを構成する基本的な要素をより深く理解するには、基本設定を参照してください。

基本設定

これらのガイドでは、Zabbix 設定の基本要素を紹介します。

- ログインとユーザーの設定
- 新しいホスト
- 新しいアイテム
- 新しいトリガー
- 障害通知の受信
- 新しいテンプレート

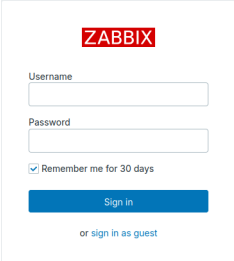
ログインとユーザーの設定

概要

このセクションでは、Zabbix にログインし、システムユーザーを設定する方法を学びます。

ログイン

ブラウザーで Zabbix Web インターフェースの URL を開きます。- Apache の場合: http://<server_ip_or_name>/zabbix - NGINX の場合: http://<server_ip_or_name>



これは Zabbix のウェルカム画面です。ユーザー名 **Admin**、パスワード **zabbix** を入力して、Zabbix スーパーユーザーとしてログインします。すべてのメニューセクションにアクセスできるようになります。

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

永続ログイン

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

30 日間記憶するが有効の場合:

- セッションは 30 日間アクティブなまま維持されます。
- 自動ログアウト は上書きされ、期間が終了するまでサインイン状態が保持されます。
- 30 日以内の今後のアクセスでは、認証情報を再入力せずに自動的にログインされます。

30 日間記憶するが無効の場合:

- 以前に有効化されていた自動ログインは解除されます。
- セッションは、設定された 自動ログアウトの間隔に従って期限切れになります。

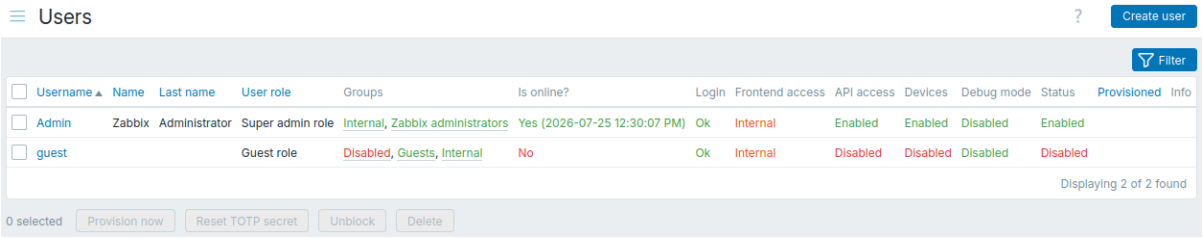
ブルートフォース攻撃に対する保護

5 回連続でログインに失敗した場合、Zabbix インターフェースはブルートフォース攻撃や辞書攻撃を防ぐために 30 秒間一時停止します。

ログインに失敗した際の IP アドレスは、ログイン成功後に表示されます。

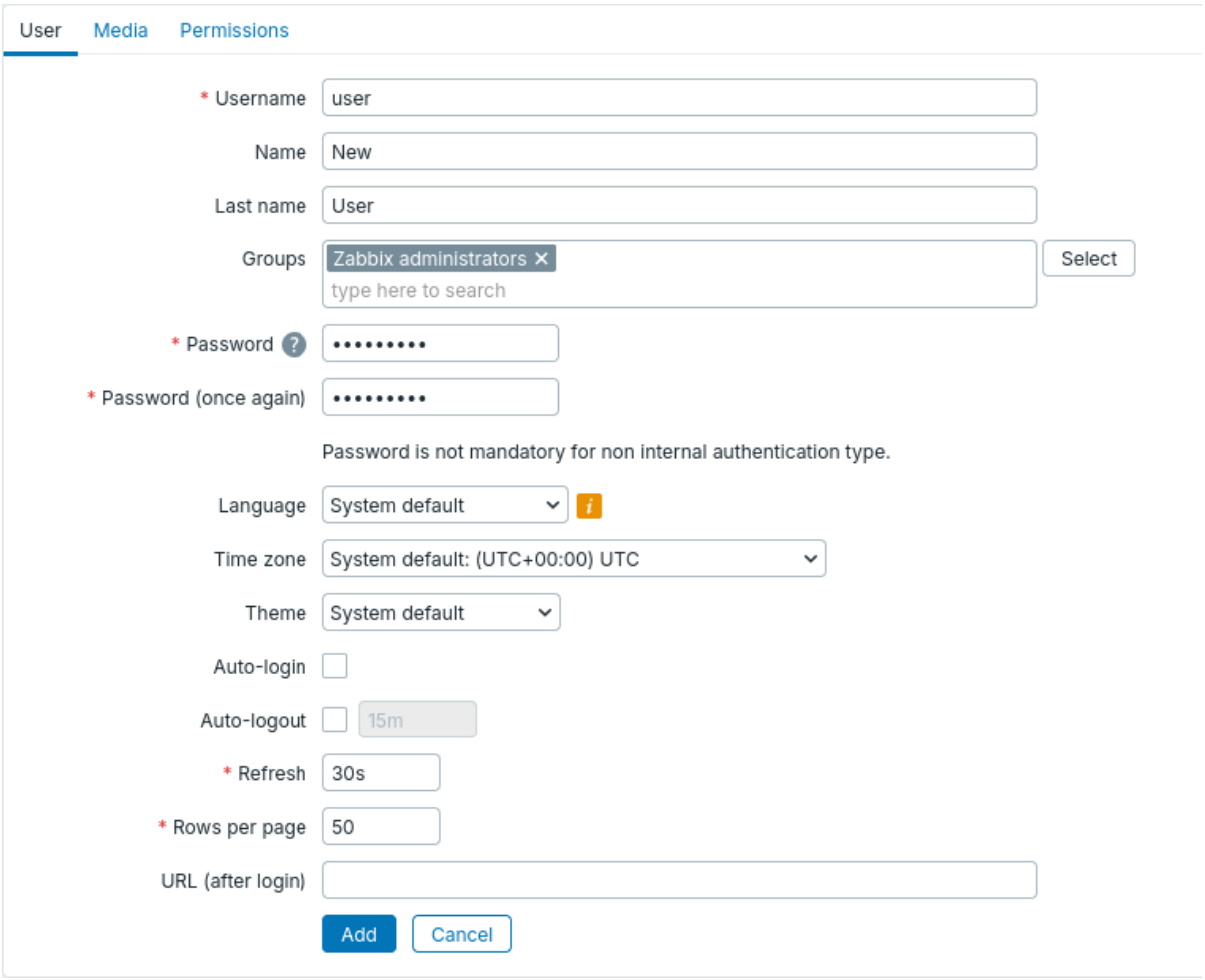
ユーザーの追加

ユーザーに関する情報を表示するには、サイドバーの縦型メニューで *Users* > *Users* に移動します。



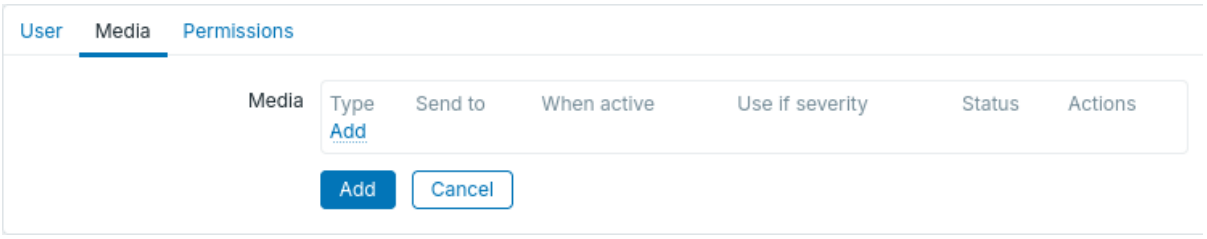
新しいユーザーを追加するには、右上隅の *Create user* を選択します。

新規ユーザーフォームでは、ユーザーを既存のユーザーグループ のいずれかに必ず追加してください。たとえば'Zabbix administrators' です。



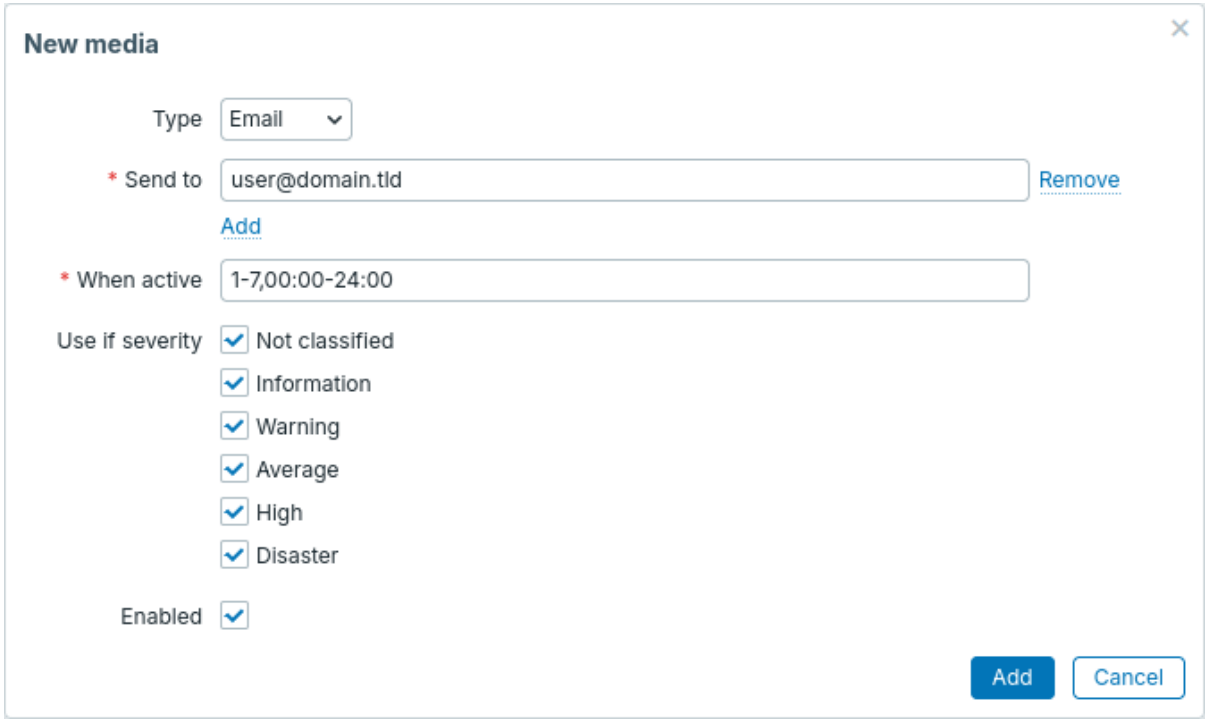
必須入力フィールドには赤いアスタリスクが付いています。この設定フォームの入力フィールドの詳細については、ユーザー設定 ページを参照してください。

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



ポップアップで、ユーザーのメールアドレスを入力します。
 すべてのメディアタイプはデフォルトで無効になっており、*Alerts* > *Media types* で有効化できます。

メディアが有効になる期間を指定できます（形式の説明についてはTime period specification ページを参照してください）。デフォルトでは、メディアは常に有効です。また、メディアを有効にするトリガーの深刻度 レベルをカスタマイズすることもできますが、ここではすべて有効のままにしておきます。



メディアを保存するには *Add* をクリックし、次に *Permissions* タブに移動します。

Permissions タブには必須フィールド *Role* があります。ロールによって、ユーザーが表示できる Web インターフェースの要素と、実行できる操作が決まります。*Select* をクリックし、一覧からいずれかのロールを選択します。たとえば、*Admin role* を選択すると、*Administration* を除くすべての Zabbix Web インターフェースセクションにアクセスできるようになります。後で、権限を変更したり、さらにユーザーロールを作成したりできます。ロールを選択すると、同じタブに権限が表示されます。

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

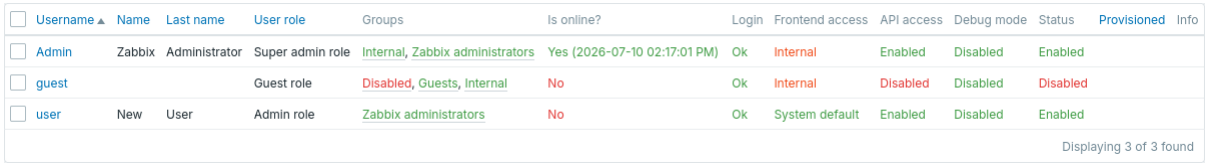
Enabled | Manage own devices

Access to actions

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

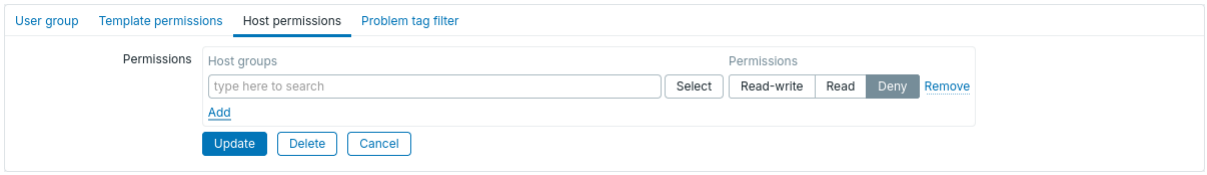
Add | Cancel

ユーザープロパティフォームで *Add* をクリックしてユーザーを保存します。新しいユーザーはユーザー一覧に表示されます。

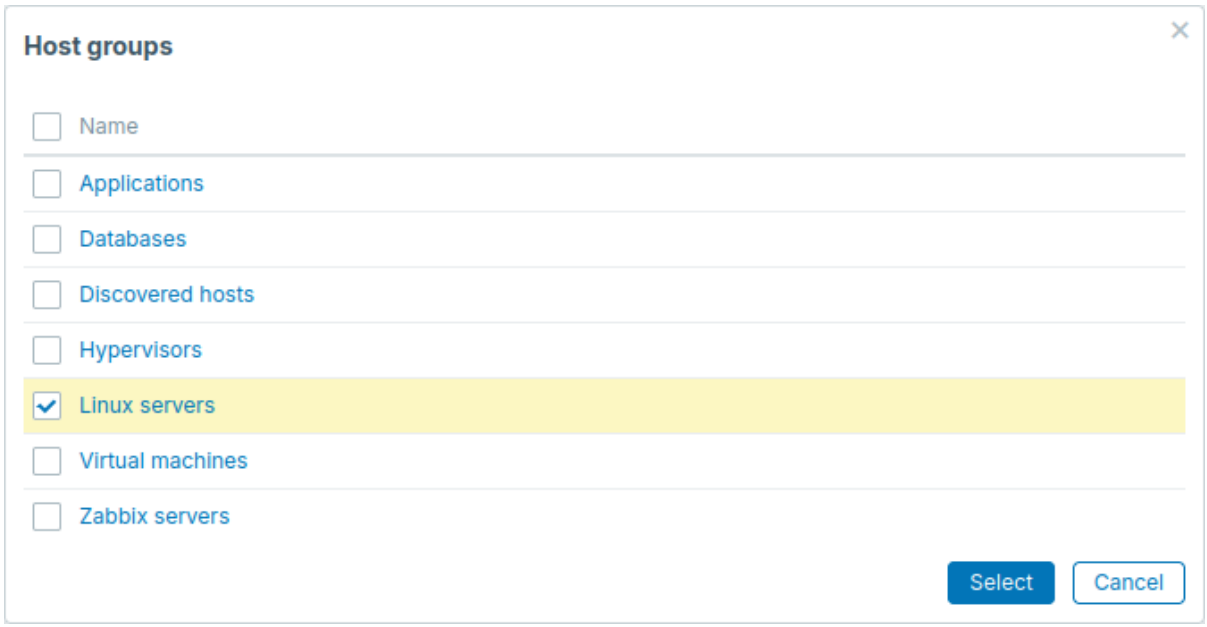


権限の追加

デフォルトでは、新しいユーザーにはホストおよびテンプレートへのアクセス権限はありません。ユーザーに権限を付与するには、*Groups* 列でそのユーザーのグループをクリックします（この場合は *Zabbix administrators*）。*User groups* プロパティフォームで、*Host permissions* タブを開き、ホストグループに対する権限を割り当てます。ホストグループ選択フィールドを表示するには、Add をクリックします。



次に、フィールドの横にある *Select* をクリックして、ホストグループの一覧を表示します。このユーザーには *Linux servers* グループへの読み取り専用アクセスを付与するため、一覧で該当するチェックボックスを選択し、*Select* をクリックして選択を確定します。



Read ボタンをクリックして権限レベルを設定し、続いて *Update* をクリックしてユーザーグループ設定の変更を保存します。

テンプレートに権限を付与するには、*Template permissions* タブに切り替えてテンプレートグループを指定する必要があります。手順はホストグループに権限を割り当てる場合と同じです。テンプレートの概要については、このクイックスタートのNew template セクションを参照してください。

Attention:
Zabbix では、ホストおよびテンプレートへのアクセス権は個々のユーザーではなく、ユーザーグループ に割り当てられます。

完了です。新しいユーザーの認証情報を使ってログインを試してみてください。

新しいホスト

概要

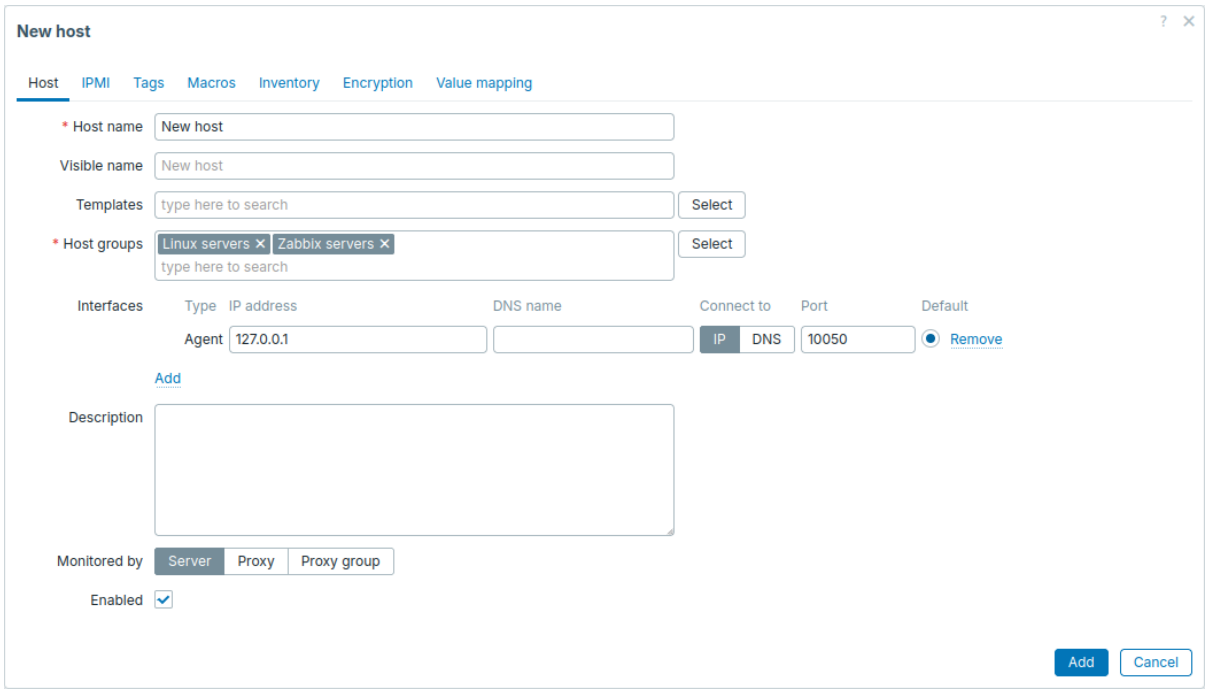
このセクションでは、新しいホストを設定する方法を学びます。

Zabbix におけるホストとは、監視したいネットワーク上のエンティティ（物理、仮想）です。Zabbix で「ホスト」として定義できる対象はかなり柔軟です。物理サーバー、ネットワークスイッチ、仮想マシン、または何らかのアプリケーションなどが該当します。

ホストの追加

Zabbix で設定されているホストの情報は、データ収集 > ホストメニューセクション、および 監視 > ホストメニューセクションで確認できます。「Zabbix server」という名前の事前定義済みホストがすでに 1 つありますが、ここでは別のホストを追加する方法を説明します。

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



必須入力フィールドには赤いアスタリスクが付いています。

ここで最低限入力する必要がある項目は次のとおりです。

ホスト名

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

ホストグループ

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

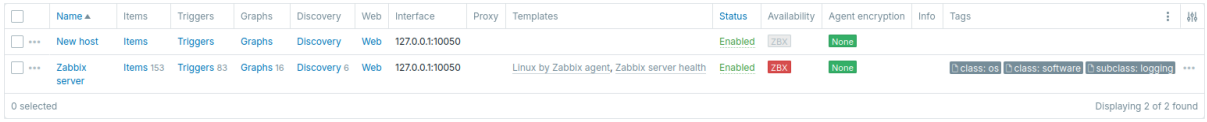
Note:
すべてのアクセス権限は個々のホストではなく、ホストグループに割り当てられます。そのため、ホストは少なくとも 1 つのグループに所属している必要があります。

インターフェース***:*** ***IP*** アドレス

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

その他のオプション は、デフォルト値のまま変更しなくてもかまいません。

入力が完了したら、追加をクリックします。新しいホストがホスト一覧に表示されます。



可用性列には、各インターフェースのホスト可用性を示すインジケーターが表示されます。Zabbix エージェントインターフェースが定義されている場合は、エージェント可用性アイコン（*ZBX* と表示）でホストの可用性を確認できます。

- ZBX - ホストのステータスはまだ確定していません。メトリクスのチェックがまだ実行されていません。
- ZBX - ホストは使用可能です。メトリクスのチェックに成功しています。
- ZBX - ホストは使用できません。メトリクスのチェックに失敗しています（アイコンの上にマウスカーソルを移動するとエラーメッセージが表示されます）。

通信に何らかのエラーが発生している可能性があります。インターフェースの認証情報が正しくないことが原因の場合もあります。Zabbix サーバーが実行中であることを確認し、しばらくしてからページを更新してみてください。

新しいアイテム

概要

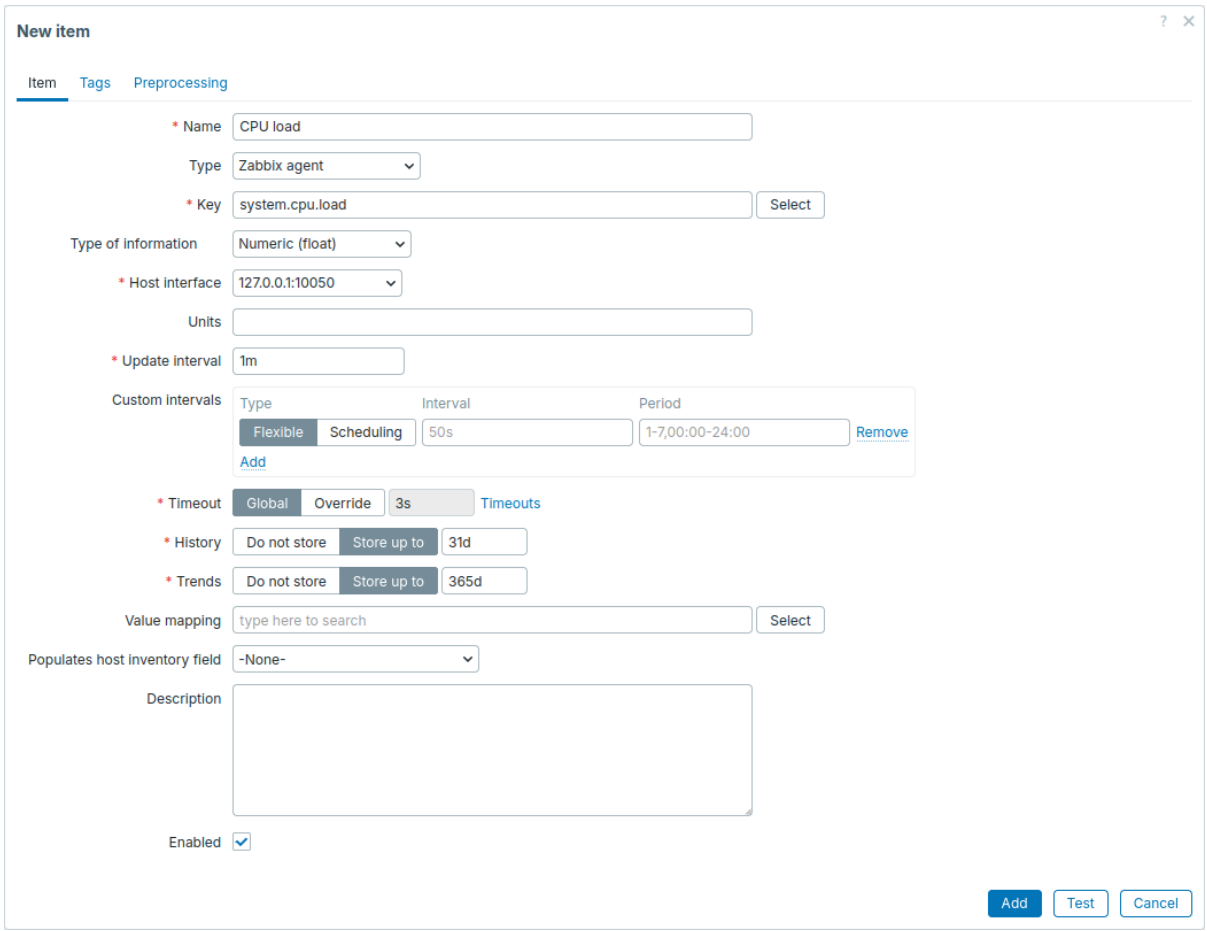
このセクションでは、アイテムの設定方法について学びます。

アイテムは、Zabbix でデータを収集するための基盤です。アイテムがなければデータはありません。なぜなら、ホストからどのようなデータを収集するか、または単一のメトリックを定義するのはアイテムだけだからです。

アイテムの追加

すべてのアイテムはホストを中心にグループ化されています。
そのため、サンプルアイテムを設定するには、データ収集 > ホストに移動し、作成した”New host” を見つけます。

”New host” の行にある アイテムリンクをクリックし、次に アイテムの作成をクリックします。
アイテムの設定フォームが開きます。



必須の入力フィールドには赤いアスタリスクが付いています。

サンプルアイテムに必要な入力情報は次のとおりです。

名前

- 値として *CPU load* を入力します。これは、一覧やその他の場所で表示されるアイテム名です。

キー

- 値として *system.cpu.load* を手動で入力します。これは、収集する情報の種類を識別するアイテムの技術名です。このキーは、Zabbix エージェントに付属する事前定義キー の 1 つにすぎません。

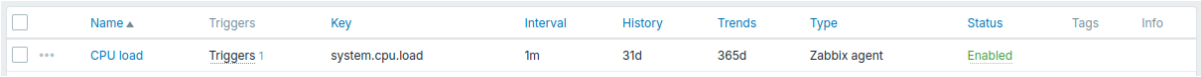
情報の種類

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

Note:
item history の保存日数を 7 日または 14 日に減らすこともできます。
これは、データベースに大量の履歴値を保持させないためのよい方法です。

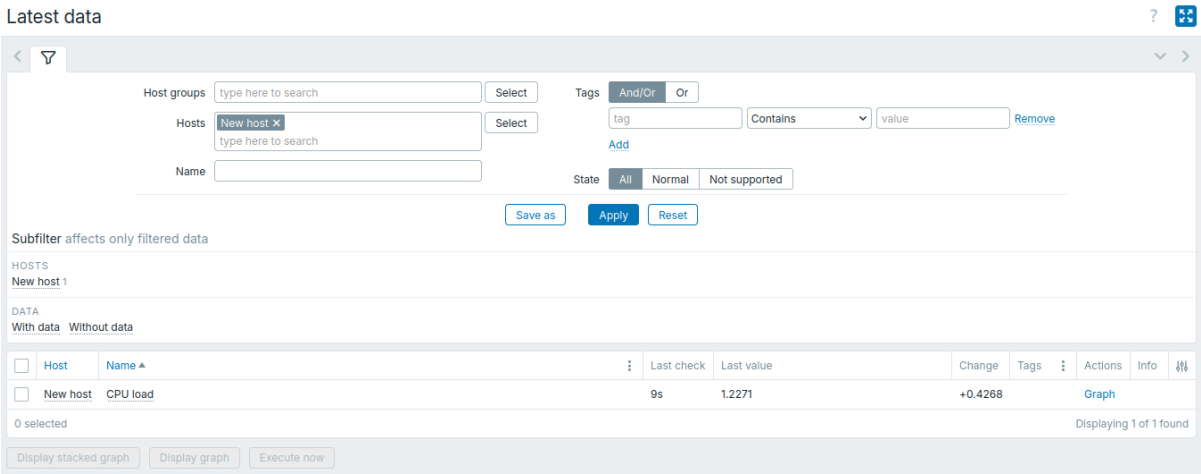
その他のオプション は、既定値のまま変更せずに使用できます。

完了したら、追加をクリックします。新しいアイテムがアイテム一覧に表示されるはずです。



データの表示

アイテムを定義したら、実際にデータを収集しているか気になるかもしれません。その場合は、*Monitoring* > *Latest data* に移動し、フィルターで'New host' を選択して、*Apply* をクリックします。



ただし、最初のデータが届くまでに最大 60 秒かかる場合があります。これは、デフォルトではサーバーが設定変更を読み取り、新しいアイテムの実行を開始する頻度だからです。

'Change' 列に値が表示されない場合は、まだ 1 つの値しか受信されていない可能性があります。30 秒待って、次の値が届くのを確認してください。

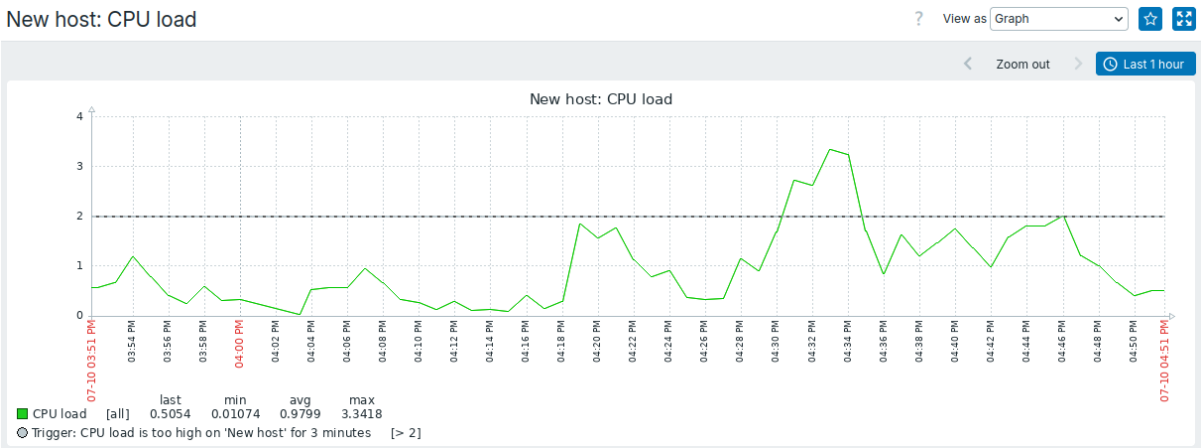
スクリーンショットのようにアイテムに関する情報が表示されない場合は、次の点を確認してください。

- アイテムの'Key' と'Type of information' フィールドを、スクリーンショットとまったく同じように入力していること;
- エージェントとサーバーの両方が稼働していること;
- ホストのステータスが'Monitored' で、可用性アイコンが緑色であること;
- ホストフィルターで選択したホストが正しいこと;
- アイテムが有効になっていること。

グラフ

アイテムがしばらく稼働したら、視覚的に確認してみましょう。シンプルグラフは、追加設定なしで監視対象の数値アイテムに利用できます。これらのグラフは実行時に生成されます。

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



新しいトリガー

概要

このセクションでは、トリガーの設定方法を学びます。

アイテムはデータを収集するだけです。受信したデータを自動的に評価するには、トリガーを定義する必要があります。トリガーには、データに対して許容されるレベルのしきい値を定義する式が含まれます。

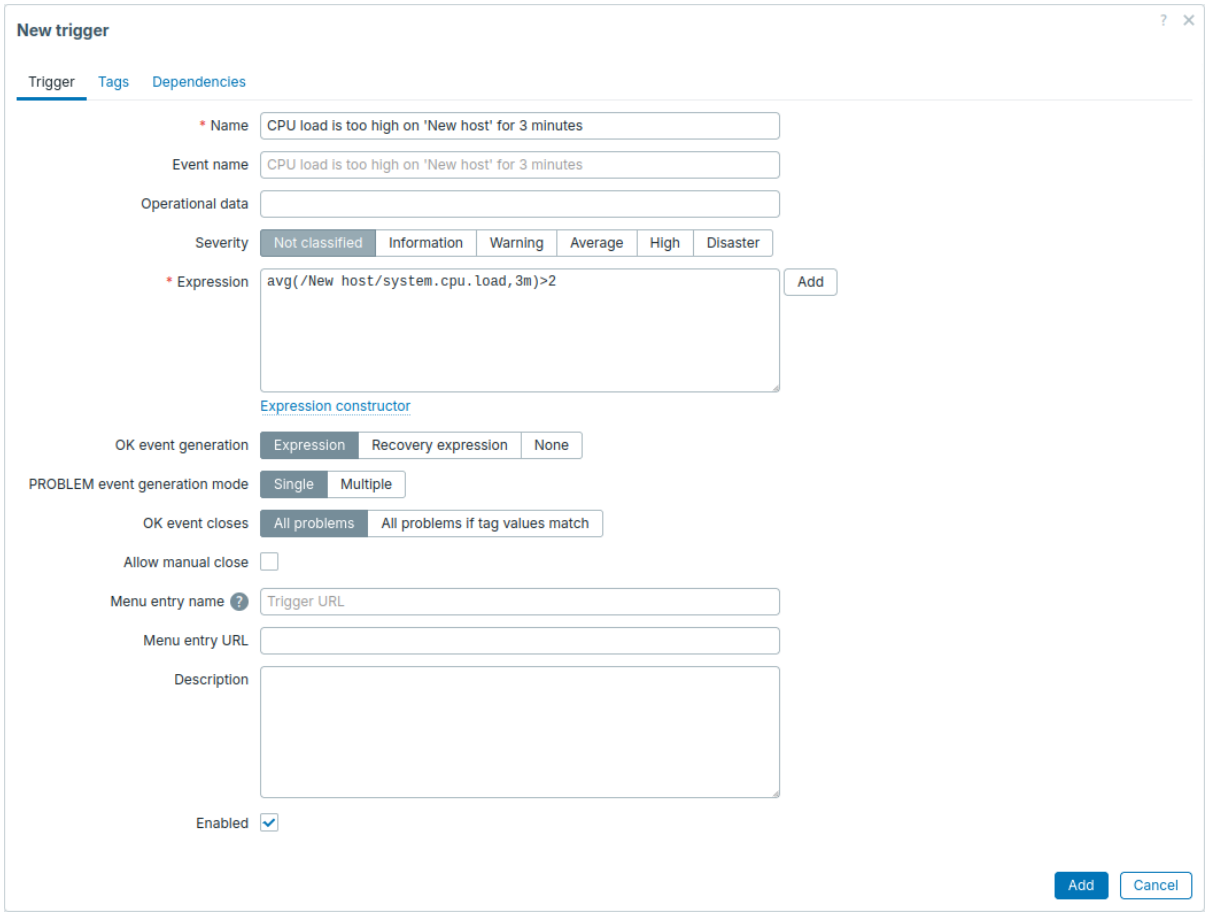
そのレベルを受信データが超えると、トリガーは「発火」し、'Problem' 状態になります。これは、注意が必要な何らかの事象が発生したことを示します。レベルが再び許容範囲に戻ると、トリガーは'Ok' 状態に戻ります。

トリガーの追加

アイテムに対してトリガーを設定するには、次の手順を実行します。

1. データ収集 > ホストに移動します。
2. 'New host' の行にある トリガーリンクをクリックします。
3. トリガーの作成をクリックします。

トリガー設定フォームが開きます。



このトリガーでここに入力する必要がある重要な情報は次のとおりです。

名前

- 値として *CPU load is too high on 'New host' for 3 minutes* を入力します。これは、一覧などに表示されるトリガー名になります。

式

- 次を入力します: `avg(/New host/system.cpu.load,3m)>2`

これがトリガー式です。式は最後の記号まで正しく入力されていることを確認してください。ここでのアイテムキー (`system.cpu.load`) は、アイテムを参照するために使用されます。この式は基本的に、3 分間の CPU 負荷平均値が 2 を超えたときに障害しきい値を超えることを示しています。トリガー式の構文 について、さらに詳しく学ぶことができます。

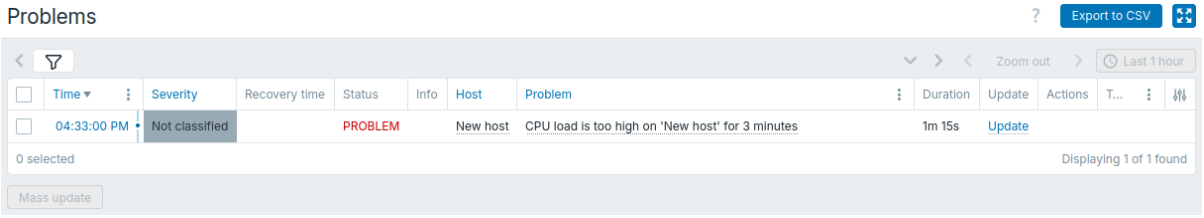
その他のオプション は、既定値のまま変更しなくてもかまいません。

完了したら、追加をクリックします。新しいトリガーがトリガー一覧に表示されるはずです。

トリガーの状態の表示

トリガーを定義したら、その状態を確認したくなる場合があります。

CPU 負荷がトリガーで定義したしきい値を超えた場合、障害は *Monitoring* > *Problems* に表示されます。



ステータス列の点滅は、トリガーの状態が最近変更されたこと、つまり過去 30 分以内に変化があったことを示します。

障害通知の受信

概要

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

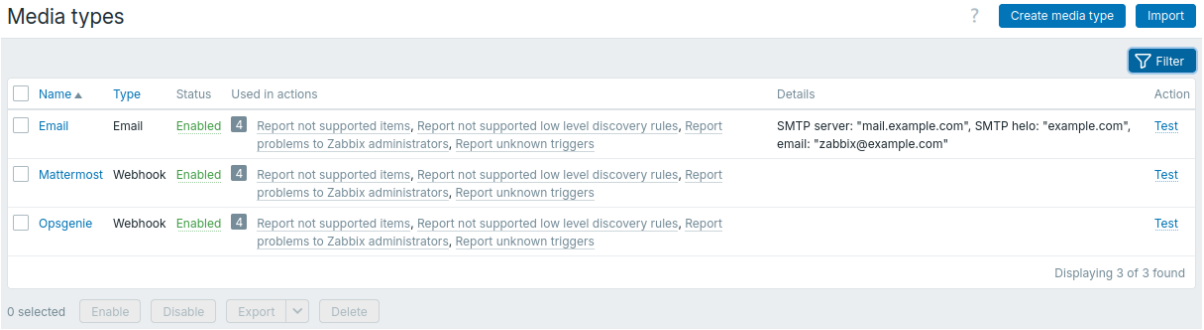
アイテムがデータを収集し、トリガーが障害発生時に「発火」するように設計されている一方で、Zabbix Web インターフェースが継続的に確認されていない場合でも、重要なイベントを通知するためのアラート機構を用意しておくと便利です。

これが通知の役割です。障害通知の配信方法として最も一般的なのはメールであるため、ここではメール通知の設定方法を学びます。

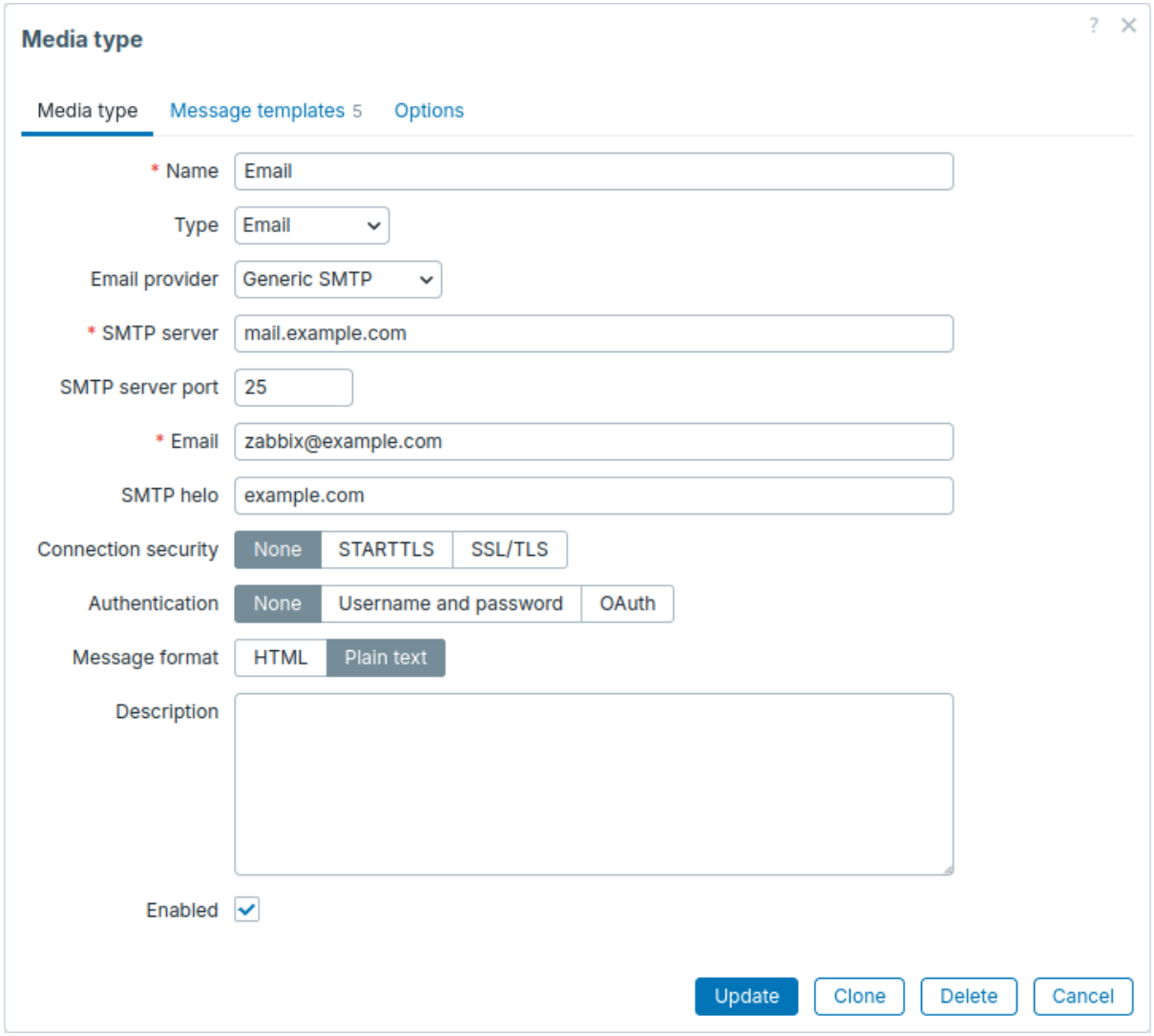
Email 設定

Zabbix には、あらかじめ定義された通知配信方法がいくつかあります。Emailはその 1 つです。

Email 設定を構成するには、*Alerts* > *Media types* に移動し、事前定義されたメディアタイプの一覧で *Email* をクリックします。



Email 設定の構成フォームが開きます。



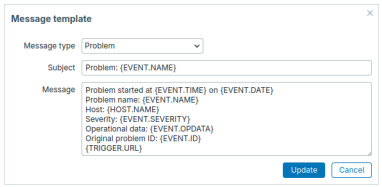
必須入力フィールドには、赤いアスタリスクが付いています。

Media type タブで、*SMTP server*、*SMTP helo*、*Email* の値を環境に適したものに設定します。

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

次に、障害メッセージの内容を定義する必要があります。内容は、*Message templates* タブで設定するメッセージテンプレートによって定義されます。

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

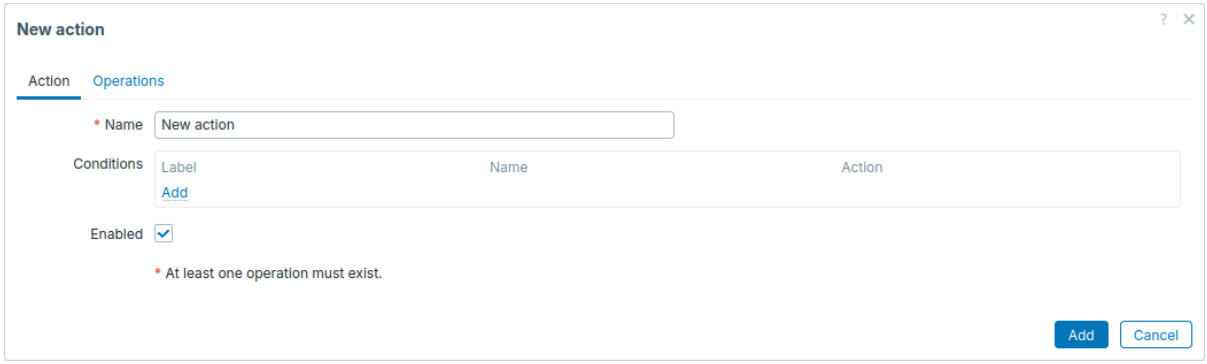


準備ができたら *Add* をクリックし、フォームを保存します。

これで、*Email* を動作するメディアタイプとして設定できました。また、メディアタイプは、特定の配信先アドレスを定義してユーザーにリンクする必要があります（新しいユーザーの設定で行ったのと同じです）。そうしないと使用されません。

新しいアクション

通知の送信は、Zabbix におけるアクション の役割の 1 つです。
そのため、通知を設定するには、*Alerts* > *Actions* > *Trigger actions* に移動し、*Create action* をクリックします。

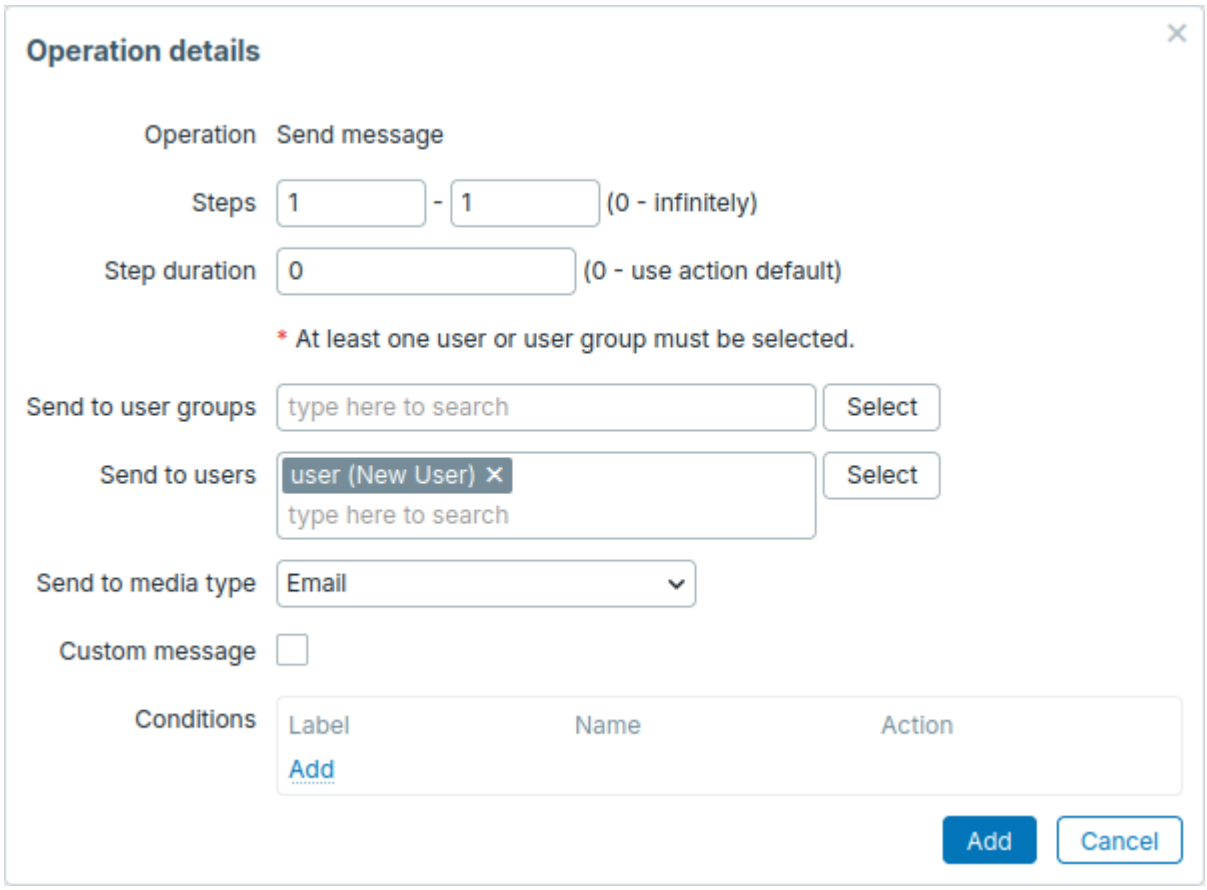


必須の入力フィールドには、赤いアスタリスクが付いています。

このフォームで、アクションの名前を入力します。

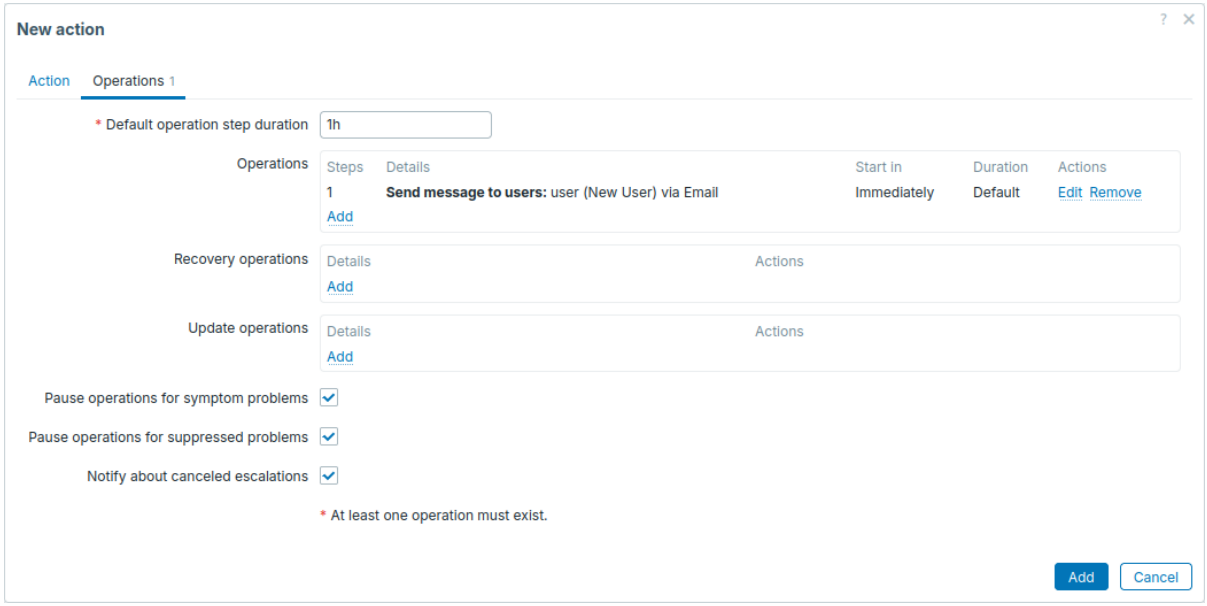
最も単純なケースでは、追加の具体的な条件 を設定しない場合、アクションは'Ok' から'Problem' への任意のトリガー変更時に実行されます。

次に、アクションが何を行うかを定義する必要があります。これは *Operations* タブで設定します。
Operations ブロックの *Add* をクリックすると、新しい操作フォームが開きます。



必須の入力フィールドには、赤いアスタリスクが付いています。

ここで、*Send to Users* ブロックの *Select* をクリックし、定義したユーザー ('user') を選択します。
Send to media type の値として"Email" を選択します。
完了したら、*Add* をクリックすると、操作が追加されます。



これで簡単なアクション設定は完了です。アクションフォームで *Add* をクリックしてください。

通知の受信

通知の送信が設定できたら、実際に受信してみましょう。
そのために、ホストの負荷を意図的に上げて、トリガーが「発火」し、障害通知を受け取れるようにします。

ホストでコンソールを開き、次を実行します:

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

これらのプロセスを 1 つまたは複数実行してもかまいません。

次に *Monitoring* > *Latest data* に移動し、'CPU Load' の値が増加していることを確認します。
トリガーが「発火」するには、'CPU Load' の値が 3 分間連続して'2' を超えている必要があることを覚えておいてください。
それが起きると:

- *Monitoring* > *Problems* で、点滅する **Problem** 状態のトリガーが表示されます;
- メールで障害通知を受け取ります。

Attention:
通知が機能しない場合:

- メール設定とアクションの両方が正しく設定されていることを、もう一度確認してください
- ユーザーの追加 の手順で説明したように、作成したユーザーが、イベントを生成したホストに対して少なくとも読み取り権限を持っていることを確認してください。'Zabbix administrators' ユーザーグループに属するそのユーザーは、ホストが属している'Linux servers' ホストグループに対して少なくとも読み取りアクセス権を持っている必要があります。
- さらに、*Reports* > *Action log* に移動してアクションログを確認できます。

新しいテンプレート

概要

このセクションでは、テンプレートの設定方法について説明します。

これまでに、アイテムとトリガーの設定方法、およびホストの障害通知を受け取る方法について学びました。

これらの手順はそれぞれ非常に柔軟ですが、たとえば 1,000 台のホストに対して同じ設定が必要な場合、多くの手順が必要になるように思われます。何らかの自動化があると便利です。

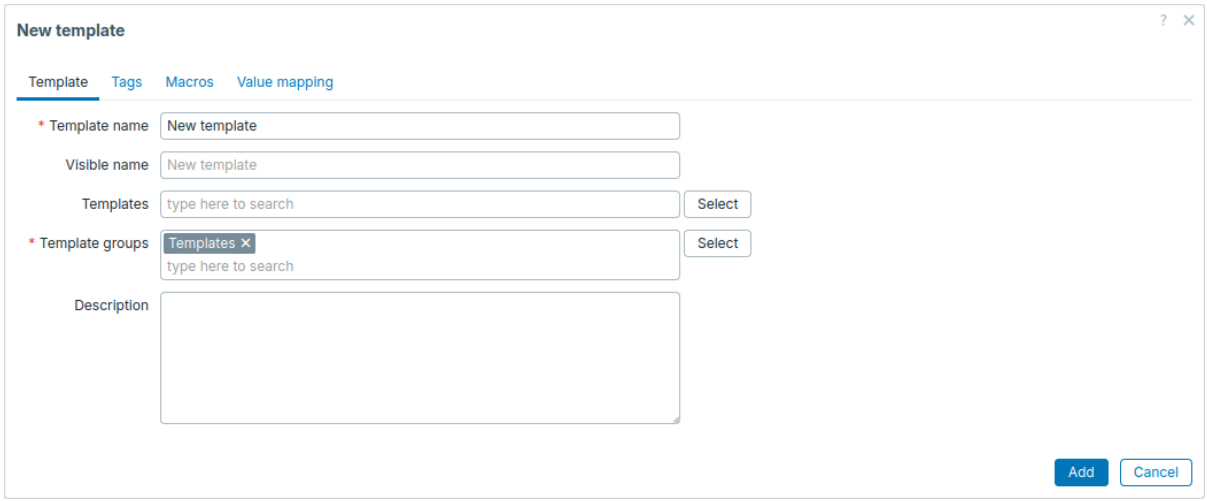
そこで役立つのがテンプレートです。

テンプレートを使用すると、便利なアイテム、トリガー、その他のエンティティをグループ化し、ホストに 1 回の手順で適用して、何度でも再利用できます。

テンプレートをホストにリンクすると、ホストはテンプレートのすべてのエンティティを継承します。つまり、あらかじめ準備した一連のチェックを、非常に迅速に適用できます。

テンプレートの追加

テンプレートを使い始めるには、まず 1 つ作成する必要があります。
そのためには、*Data collection* > *Templates* で *Create template* をクリックします。
これにより、テンプレート設定フォームが表示されます。



必須の入力フィールドには、赤いアスタリスクが付いています。

ここで入力が必要なパラメータは次のとおりです。

Template name

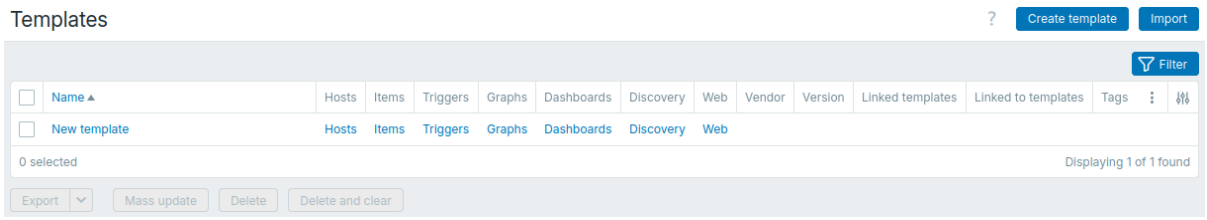
- テンプレート名を入力します。英数字、スペース、アンダースコアを使用できます。

Template groups

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

Note:
テンプレートグループへのアクセス権限は、ホスト権限と同様に、同じ場所にある **Template permissions** タブのuser group 設定で割り当てられます。すべてのアクセス権限は個々のテンプレートではなくグループに対して割り当てられるため、テンプレートを少なくとも 1 つのグループに含めることが必須です。

完了したら、*Add* をクリックします。新しいテンプレートがテンプレート一覧に表示されるはずです。
filter を使ってテンプレートを検索することもできます。

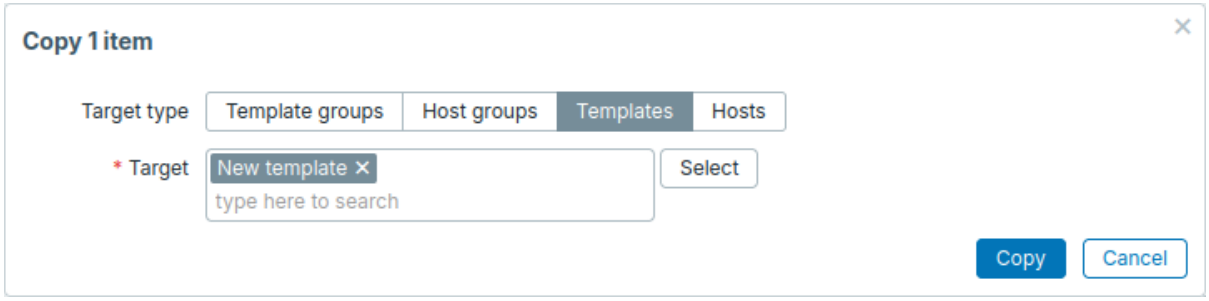


ご覧のとおり、テンプレートは作成されていますが、中身はまだありません。つまり、アイテム、トリガー、その他のエンティティはありません。

テンプレートへのアイテムの追加

テンプレートにアイテムを追加するには、次の手順に従います。

1. データ収集 > ホストに移動します。
2. 一覧から「New host」を見つけ、アイテムのハイパーリンクをクリックします。
3. 一覧で「CPU Load」アイテムのチェックボックスを選択します。
4. 一覧の下にある コピーをクリックします。
5. テンプレートタブを選択します。
6. アイテムのコピー先となるテンプレートを選択します。



必須入力フィールドには、赤いアスタリスクが付いています。

7. コピーをクリックします。

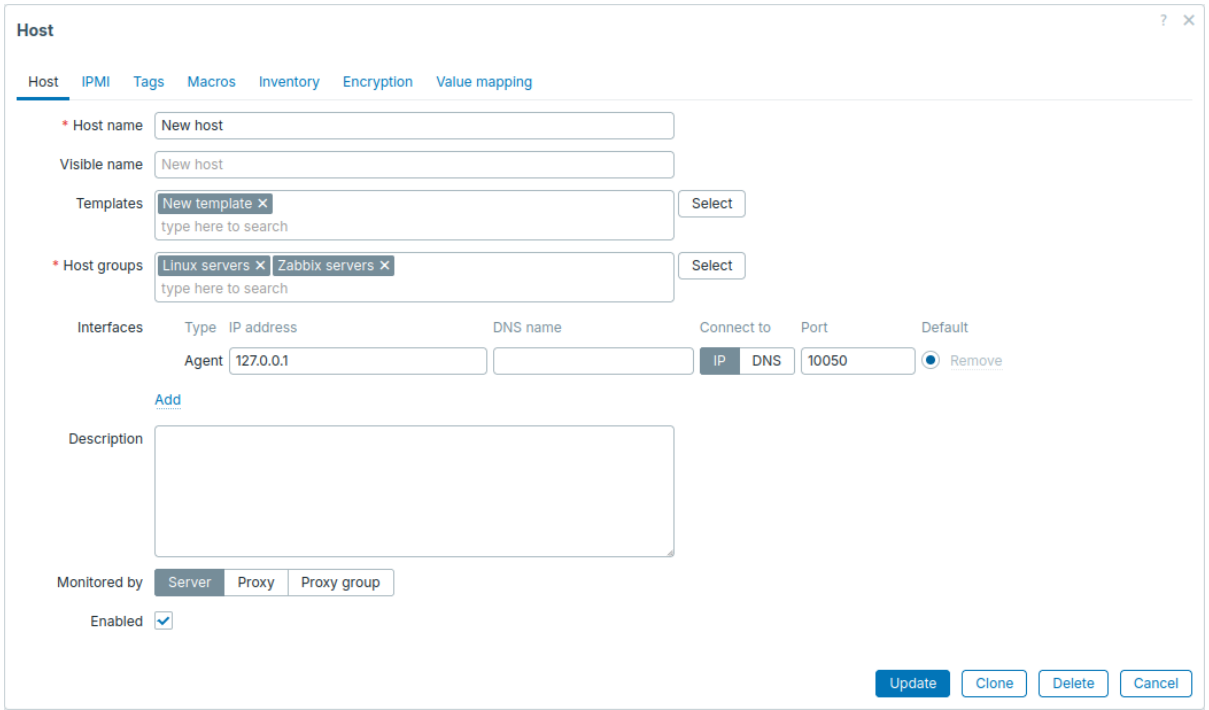
ここで データ収集 > テンプレートに移動すると、「New template」に新しいアイテムが 1 つ追加されているはずです。

ここではひとまず 1 つのアイテムだけにしますが、同様の手順で、テンプレートに他のアイテム、トリガー、またはその他のエンティティを追加できます。これにより、特定の目的（OS の監視、単一アプリケーションの監視など）に必要なエンティティを十分に含むテンプレートを作成できます。

テンプレートをホストにリンクする

テンプレートの準備ができたら、ホストに追加します。手順は次のとおりです。

1. データ収集 > ホストに移動します。
2.「ホストの作成」をクリックして設定フォームを開き、テンプレートフィールドを見つけます。
3. テンプレートフィールドに 新しいテンプレートと入力し始めます。
   新しく作成したテンプレートの名前がドロップダウンリストに表示されます。
4. 下にスクロールして選択します。
   テンプレートフィールドに表示されることを確認します。



5. フォームの 更新をクリックして変更を保存します。

これで、テンプレートに含まれるすべてのエンティティとともに、テンプレートがホストに追加されました。

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

ホストへの事前定義テンプレートのリンク

お気づきかもしれませんが、Zabbix には、さまざまな OS、デバイス、アプリケーション向けの事前定義テンプレートが用意されています。監視をすぐに開始するには、適切なテンプレートをホストにリンクできますが、これらのテンプレートは環境に合わせて調整する必要がある点に注意してください。不要なチェックもあれば、ポーリング間隔が長すぎる、または短すぎる場合もあります。

テンプレート についての詳細情報も参照できます。

Apache Web サーバーの監視

Apache Web サーバーの監視を開始するには、次の手順に従います。

1. Apache の統計情報にアクセスできることを確認します。
2. Apache 監視テンプレートを適用します。
3. 最新データを確認します。

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

Apache 統計へのアクセス

この手順の目的は、Apache Web サーバーの統計情報を Zabbix で読み取れるようにすることです。

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

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

Debian/Ubuntu の場合	RHEL ベースのシステムの場合
`apache2ctl -M | grep status`	`httpd -M | grep status`

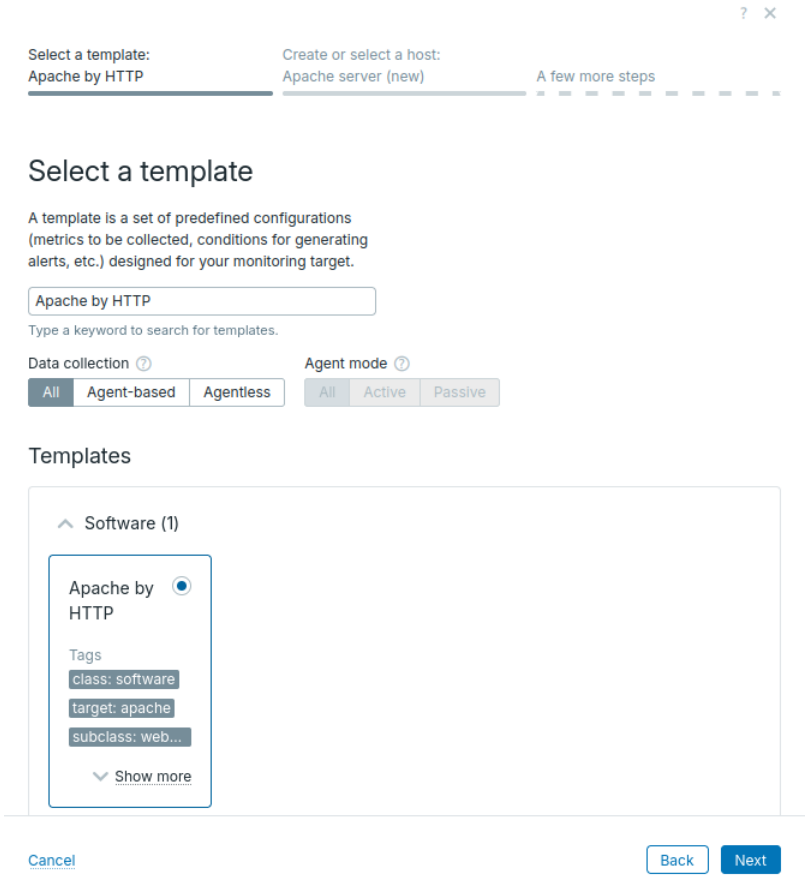
出力に"`status_module (shared)`" が表示される場合、status モジュールはすでに有効です。表示されない場合は、status モジュールを有効にする必要があります。

Apache 監視テンプレートの適用

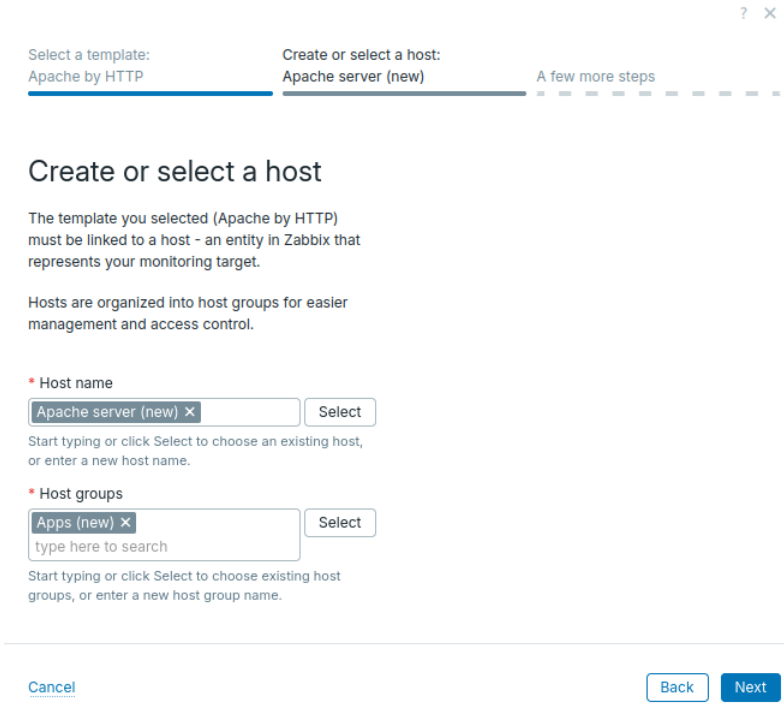
Zabbix では、このテンプレートはすぐに使用できる状態になっています。データ収集、可視化、分析のための事前設定済みの要素が含まれています。

テンプレートを適用するには、次の手順を実行します。

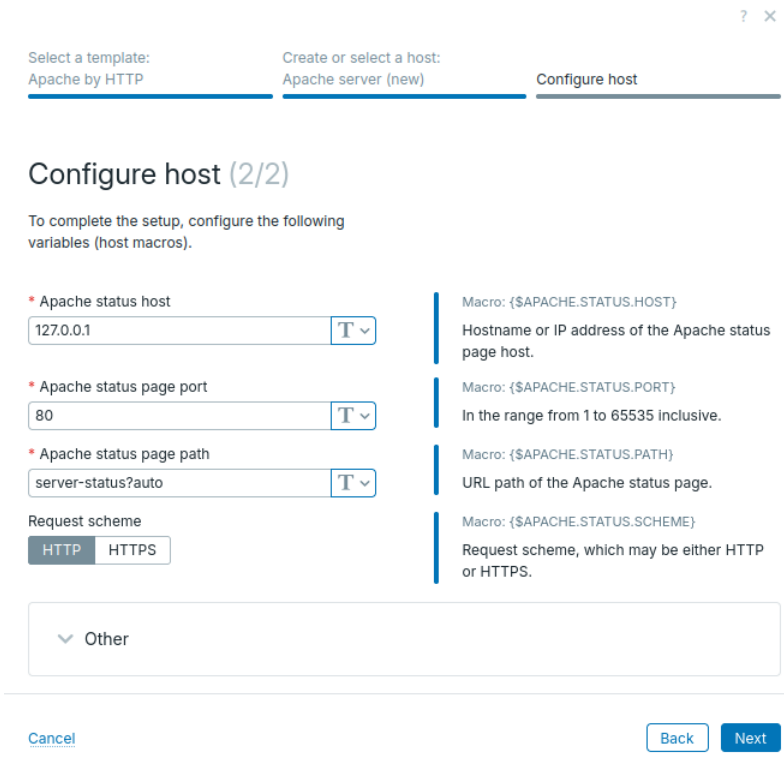
1. データ収集 > ホストでホストウィザード を起動します。
2. *Apache by HTTP* テンプレートを選択します。



3. Apache ホストを作成し、グループを追加します。
   この仮想ホストには、Apache 関連のデータが含まれます。



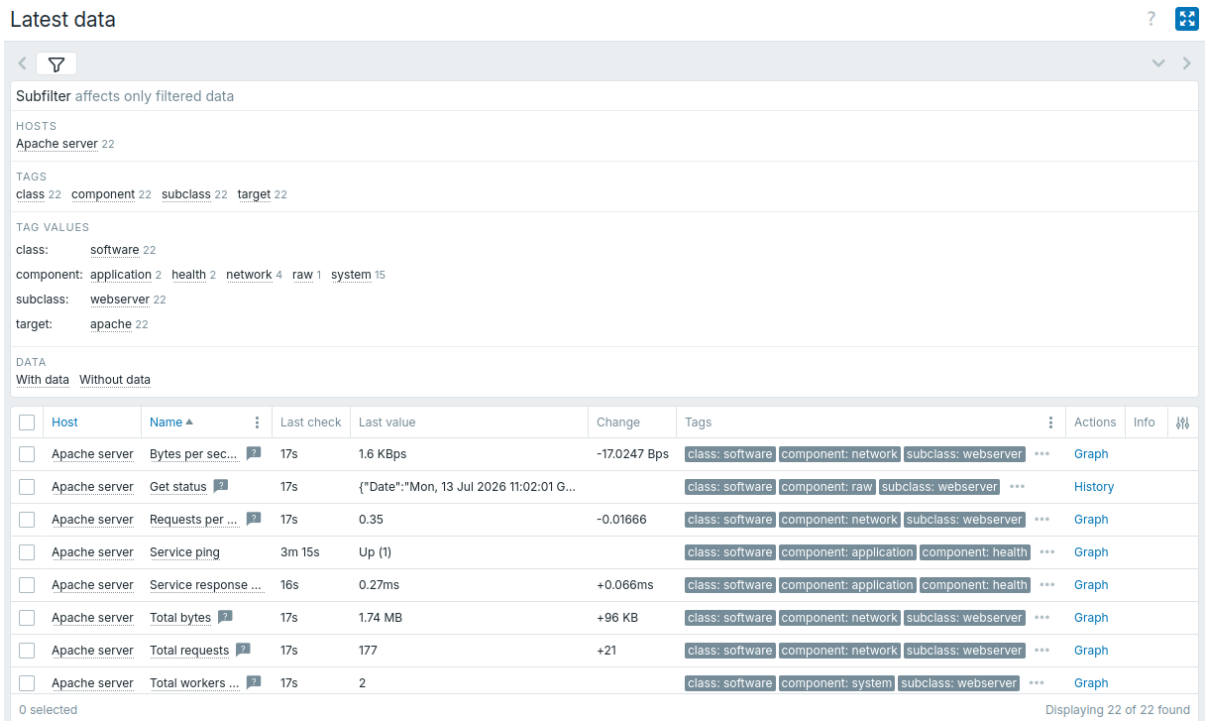
4. ホストを設定します。
   特に、Apache Web サーバーのアドレスを入力します（ローカルマシンでは 127.0.0.1、リモートマシンでは IP アドレス）。



その他の設定はデフォルトのままで構いません。

5. ホストウィザードを完了するには、作成と 完了をクリックします。

最新データを確認する



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

Zabbix は可視化オプションを提供し、何らかの障害が発生した場合にはアラートを送信します。メールでのアラートを設定するには、障害通知の受信を参照してください。

Linux サーバーの監視

Linux サーバーの監視を開始するには、次の手順に従います。

1. Linux 監視テンプレートを適用します。
2. サーバーに Zabbix エージェントをインストールします。
3. 最新データを確認します。

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

Linux サーバーの監視には、Zabbix エージェントを使用します。エージェントのインストール手順は、この手順の中で案内します。

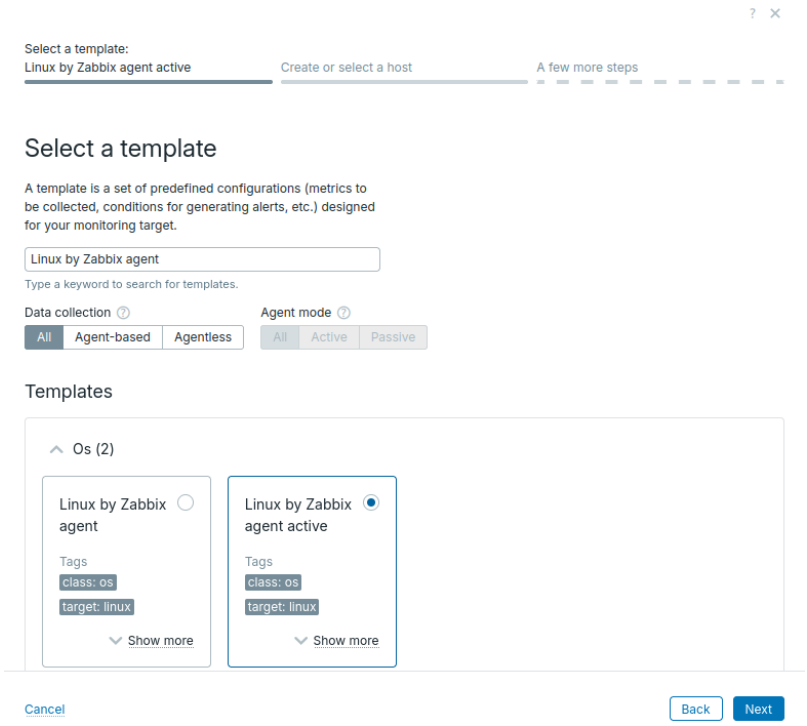
このガイドを使用するには、Zabbix サーバーと Web インターフェースがインストール済みである必要があります。また、Zabbix にデフォルトの *Admin* ユーザーとしてログインしている必要があります。

Linux 監視テンプレートを適用する

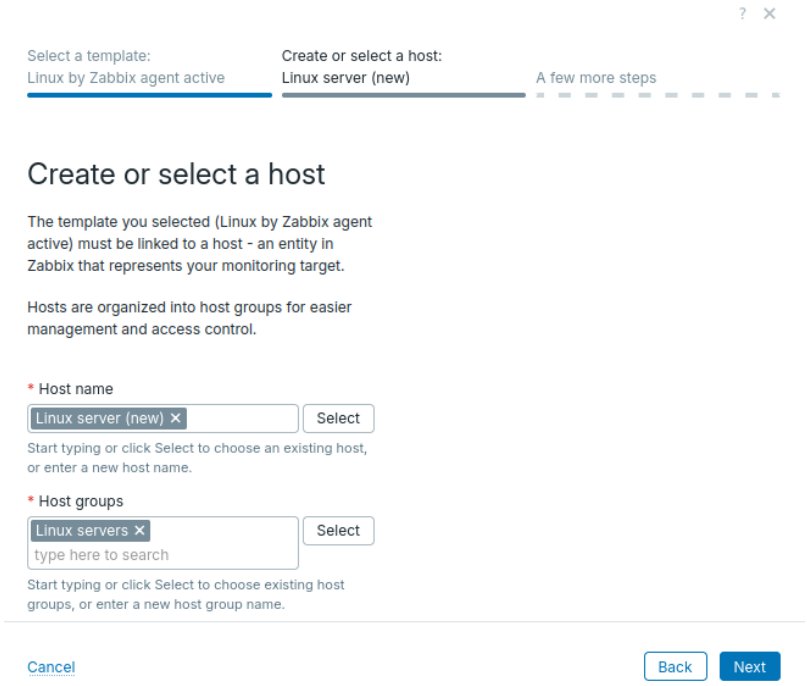
Zabbix では、このテンプレートはすぐに使用できます。データ収集、可視化、分析のために事前設定された要素が含まれています。

テンプレートを適用するには:

1. データ収集 > ホストでホストウィザード を開始します。
2. *Linux by Zabbix agent active* テンプレートを選択します。



3. Linux サーバー用のホストを作成し、ホストグループを追加します。
   この仮想ホストには、Linux サーバー監視に関連するすべてのデータが含まれます。



4. エージェントを設定してインストールします。

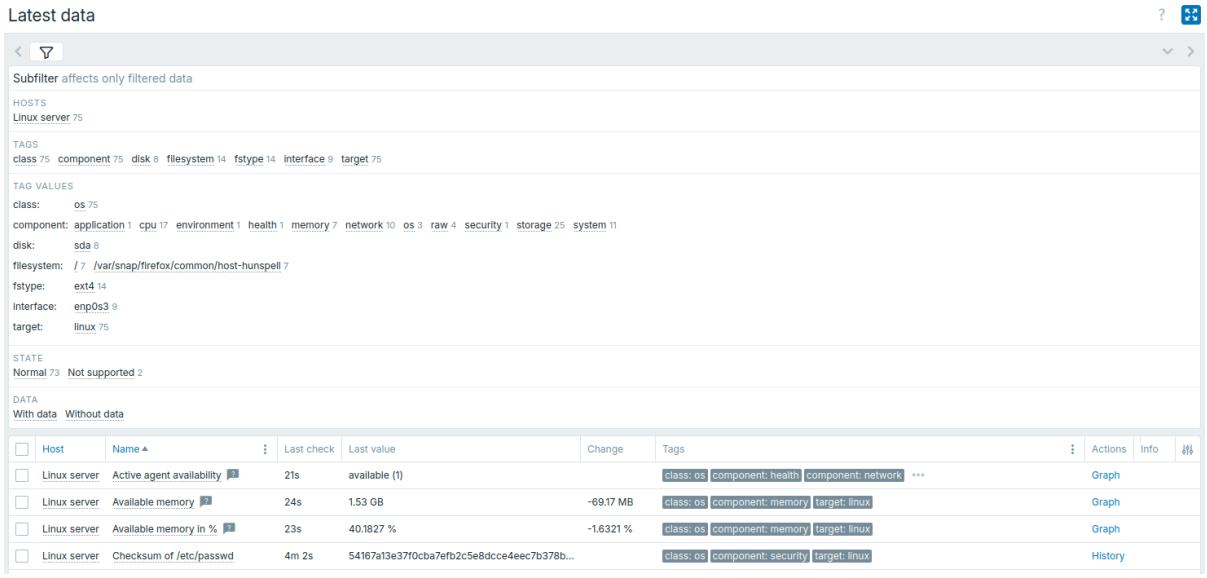


エージェントをインストールするには、Linux サーバー上で表示されたコマンドを実行します。
これにより、エージェントがインストールされ、暗号化を使用して Zabbix サーバーへの接続が設定されます。
 すでにエージェントがインストールされている場合は、この手順をスキップできます（次へをクリックします）。
ただし、エージェントに接続するための事前共有鍵の識別子と事前共有鍵の入力を求められます。

他の設定は既定値のままで構いません。

5. ホストの設定を完了します。作成と 完了をクリックします。

最新データを確認する



おめでとうございます。Linux サーバーの監視が開始されました。

Zabbix には可視化オプションがあり、何らかの障害が発生した場合にはアラートを送信します。メールでのアラートを設定するには、障害通知の受信を参照してください。

MySQL サーバーの監視

MySQL サーバーの監視を開始するには、次の手順に従います。

1. 制限付きの MySQL ユーザーを作成します。
2. MySQL 監視テンプレートを適用します。
3. 最新データを確認します。

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

MySQL サーバーの監視には Zabbix エージェントを使用します。エージェントのインストール手順は、この手順の中で案内します。

このガイドを使用するには、Zabbix サーバーと Web インターフェースがインストール済みである必要があります。また、Zabbix にデフォルトの *Admin* ユーザーとしてログインしている必要があります。

権限を制限した MySQL ユーザーの作成

MySQL の監視に必要な権限だけを付与した MySQL ユーザーを作成することをお勧めします。
<password> を強力で一意のパスワードに置き換えてください。

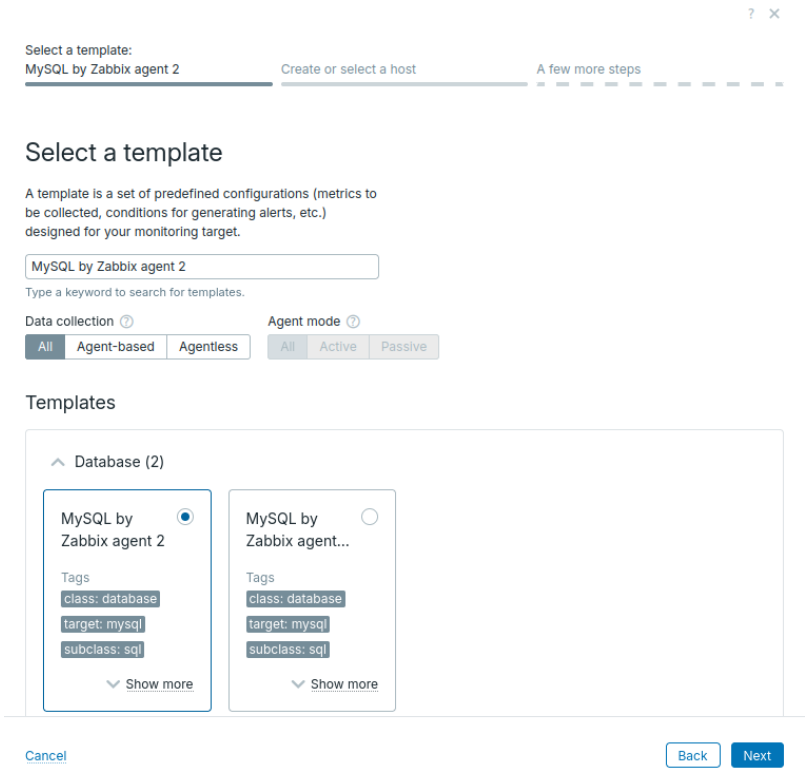
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

MySQL 監視テンプレートの適用

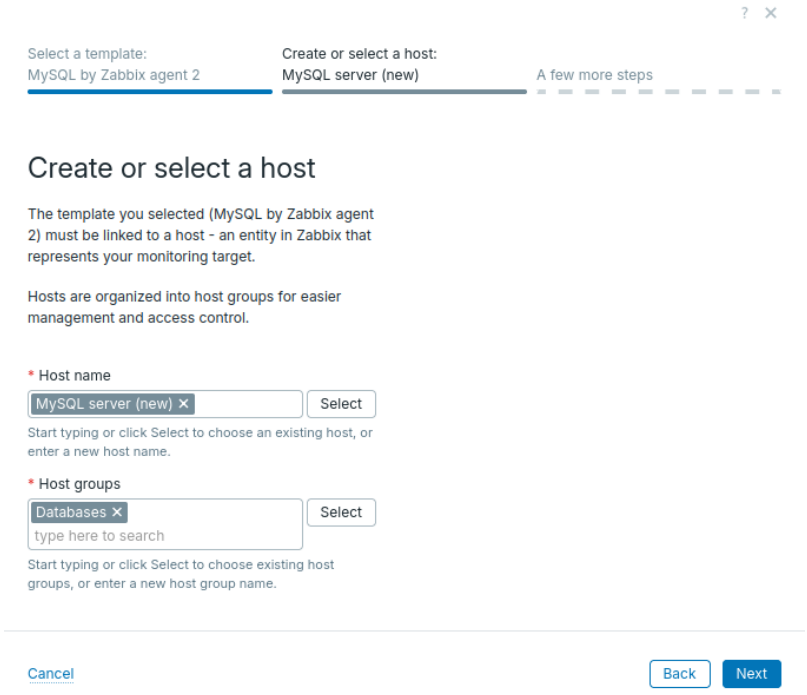
Zabbix では、このテンプレートはすぐに使用できる状態になっています。データ収集、可視化、分析のために事前設定された要素が含まれています。

テンプレートを適用するには、次の手順を実行します。

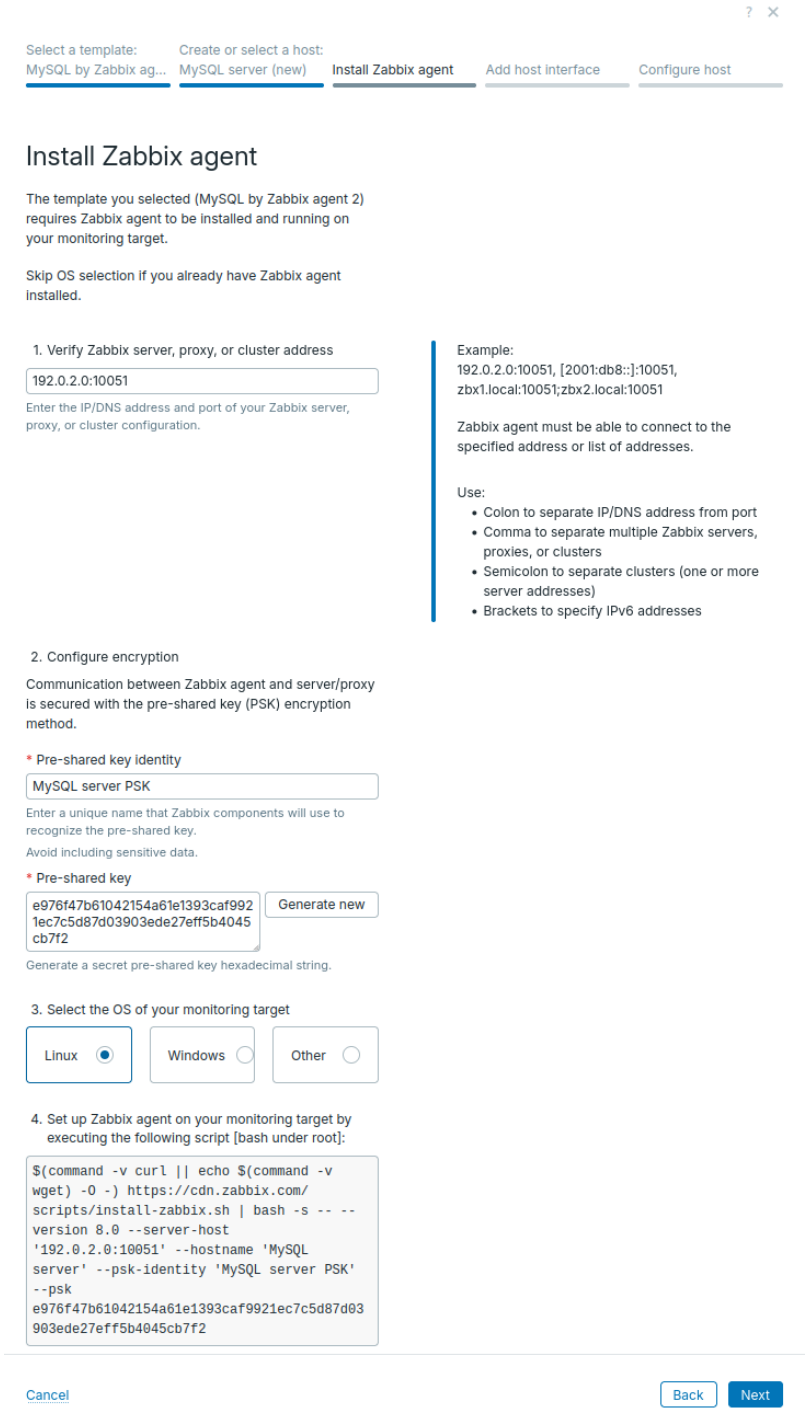
1. データ収集 > ホストでホストウィザード を起動します。
2. *MySQL by Zabbix agent 2* テンプレートを選択します。



3. MySQL サーバーホストを作成し、ホストグループを追加します。
   この仮想ホストには、MySQL サーバー監視に関連するすべてのデータが含まれます。

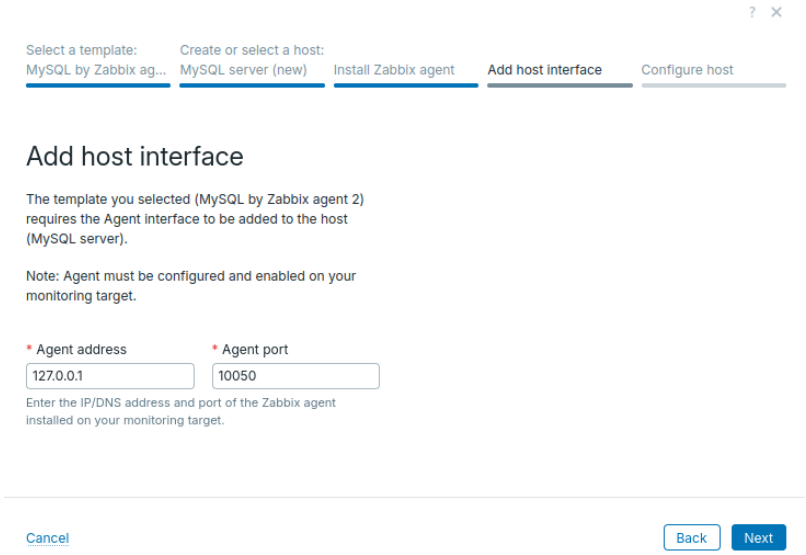


4. エージェント 2 を設定してインストールします（Zabbix agent 2 は、MySQL 監視を標準で提供します）。

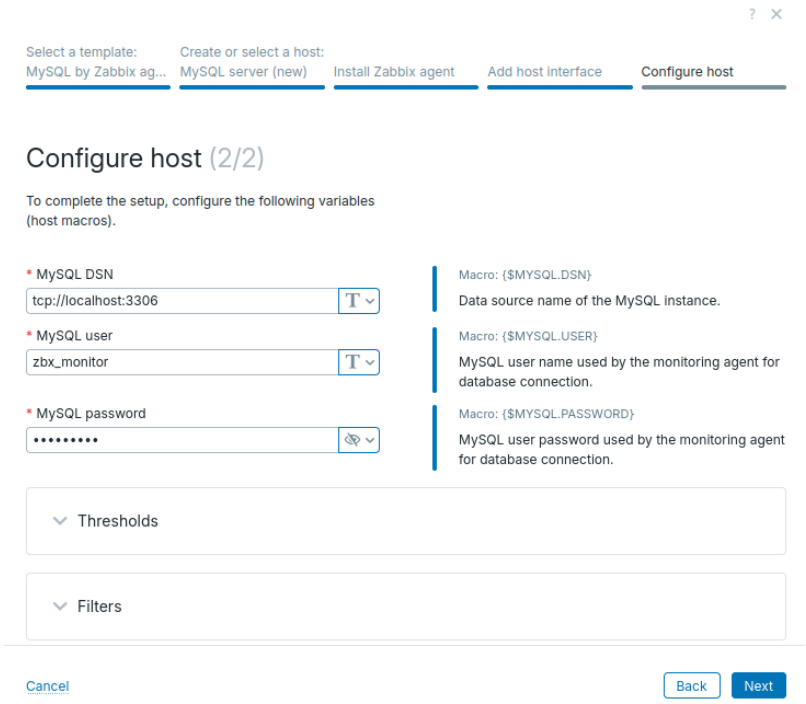


エージェントをインストールするには、MySQL サーバーをホストしているマシンで、提供されたコマンドを実行します。
これにより、エージェントがインストールされ、暗号化を使用して Zabbix サーバーへのエージェントの接続が設定されます。
すでにエージェント 2 がインストールされている場合は、この手順をスキップできます（次へをクリックします）。
ただし、エージェントに接続するための事前共有キーの識別子と事前共有キーの入力を求められます。

5. エージェント（Linux サーバー）の IP アドレスとポートを入力します。



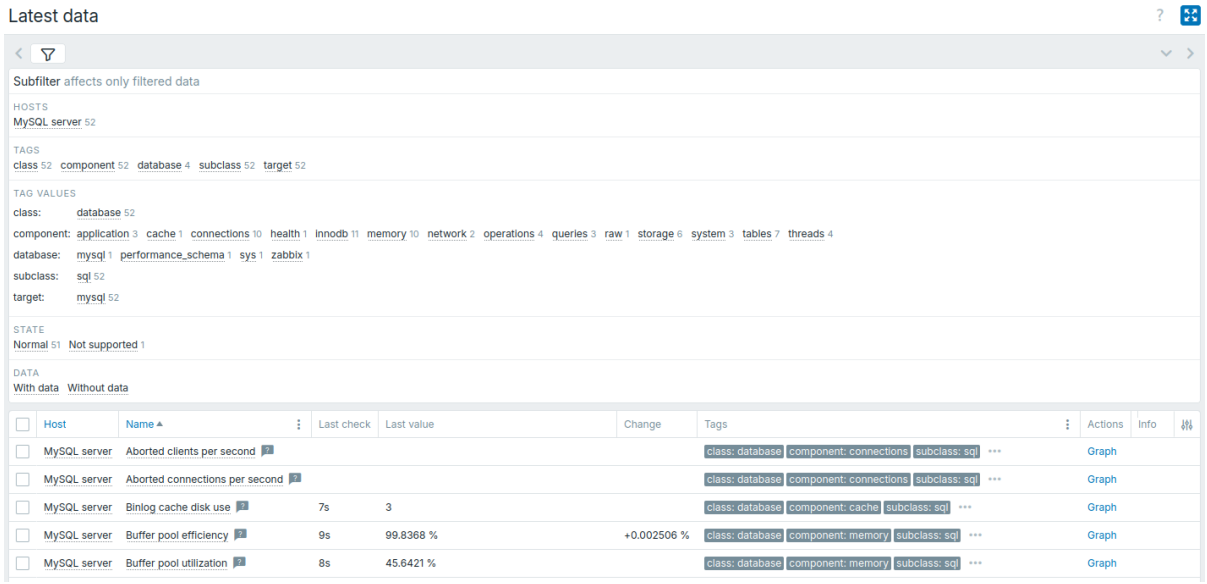
6. ホストの設定を完了します。
   重要な点として、MySQL サーバーの DSN、MySQL ユーザー名、パスワードの入力を求められます。
   作成したパスワードを持つ `zbx_monitor` ユーザーを指定してください。



その他の設定はデフォルトのままで構いません。

6. ホストウィザードを完了するには、作成と 完了をクリックします。

最新データを確認する



おめでとうございます。MySQL サーバーの監視が開始されました。

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

その他のガイド

このドキュメントセクションには、Zabbix を一般的に必要とされるいくつかの監視目標に合わせて設定するための簡単な手順が含まれています。

これは、Zabbix の新規ユーザーを想定して設計されており、タスクの解決に必要な情報を含む他のドキュメントセクションへの案内として使用できます。

以下のクイックリファレンスガイドを利用できます:

- Zabbix エージェントで Linux を監視する
- Zabbix エージェントで Windows を監視する
- HTTP 経由で Apache を監視する
- Zabbix エージェント 2 で MySQL を監視する
- Zabbix で VMware を監視する
- Zabbix でネットワークトラフィックを監視する
- アクティブチェックを使用して Zabbix でネットワークトラフィックを監視する
- Browser アイテムで Web サイトを監視する
- Zabbix エージェント 2 (パッシブ) で Web サイト証明書を監視する
- Zabbix でネットワークスイッチまたはルーターを監視する
- アクティブチェックを使用して Windows イベントログを監視する

Zabbix エージェントで **Linux** を監視する

はじめに

このページでは、Zabbix を使用して Linux マシンの基本的な監視を開始するために必要な手順を順を追って説明します。
このチュートリアルで説明する手順は、Linux ベースのあらゆるオペレーティングシステムに適用できます。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、Linux マシンの基本的な監視を有効にするために必要な最小限の手順をまとめています。より詳細なカスタマイズオプションや、さらに高度な設定が必要な場合は、Zabbix マニュアルのConfigurationセクションを参照してください。

前提条件

このガイドを進める前に、OS 向けの手順に従って ダウンロードしてインストール し、Zabbix サーバーと Zabbix Web インターフェースを用意する必要があります。

Zabbix エージェントのインストール

Zabbix エージェントは、データ収集を担当するプロセスです。

Zabbix サーバーのバージョンを確認します:

```
zabbix_server -V
```

監視対象の Linux マシンに、同じバージョンの Zabbix エージェントをインストールすることを推奨します。
監視要件によっては、Zabbix サーバーがインストールされているのと同じマシンの場合もあれば、まったく別のマシンの場合もあります。

最適なインストール方法を選択してください:

- Docker コンテナーとして実行する - 利用可能なイメージの一覧は Zabbix Docker repository を参照してください。
- Zabbix の packages からインストールする (Alma Linux、CentOS、Debian、Oracle Linux、Raspberry Pi OS、RHEL、Rocky Linux、SUSE Linux Enterprise Server、Ubuntu で利用可能)。
- sources からコンパイルする。

監視のための Zabbix の設定

Zabbix エージェントは、アクティブモードまたはパッシブモードでメトリクスを収集できます（同時に使用可能です）。

> **Note:**
> パッシブチェックは、単純なデータ要求です。Zabbix サーバーまたはプロキシが何らかのデータ（たとえば CPU 使用率）を要求し、Zabbix エージェントが結果をサーバーに返します。アクティブチェックでは、より複雑な処理が必要です。エージェントはまず、独立して処理するためのアイテム一覧をサーバーから取得し、その後、データをまとめて送信します。詳細はPassive and active agent checks を参照してください。

Zabbix が提供する監視テンプレートには、通常 2 つの選択肢があります。Zabbix エージェント用のテンプレートと、Zabbix エージェント（アクティブ）用のテンプレートです。最初の □□□□□□ では、エージェントはパッシブモードでメトリクスを収集します。これらのテンプレートは、通信プロトコルは異なりますが、同一の監視結果を提供します。

以降の Zabbix の設定は、アクティブ またはパッシブ の Zabbix エージェントチェック用テンプレートのどちらを選択するかによって異なります。

パッシブチェック

Zabbix エージェント

1. エージェントがインストールされているマシンで、エージェント設定ファイルを開きます。

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（およびオプションのポート）を、server パラメータに追加します。例:

```
Server=192.0.2.1:10051
```

Zabbix エージェントは、このアドレスを使用して、指定された Zabbix サーバーまたはプロキシからデータを取得するための受信接続のみを受け入れます。

3. Zabbix エージェントを再起動します。

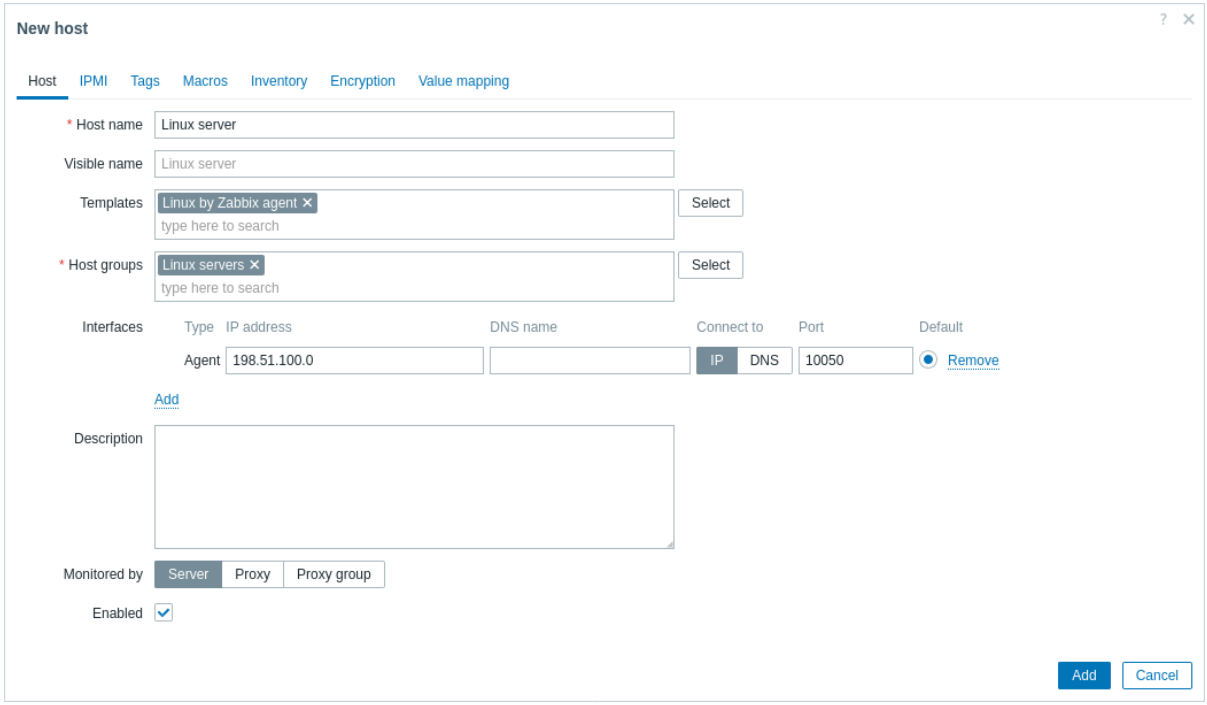
```
systemctl restart zabbix-agent
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

- ホスト名フィールドに、ホスト名を入力します（例: ”Linux server”）。
- テンプレートフィールドに、ホストにリンクされるテンプレート”Linux by Zabbix エージェント” を入力または選択します。
- ホストグループフィールドに、ホストグループを入力または選択します（例: ”Linux servers”）。
- インターフェースパラメーターで、エージェントインターフェースを追加し、エージェントがインストールされている Linux マシンの IP アドレスまたは DNS 名を指定します。



3. 追加をクリックしてホストを追加します。このホストは監視対象の Linux マシンを表します。

アクティブチェック

Zabbix エージェント

1. エージェントがインストールされているマシンで、エージェント設定ファイルを開きます。

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（オプションのポートを含む）を、ServerActive パラメータに追加します。例:

```
ServerActive=192.0.2.1:10051
```

Zabbix エージェントは、このアドレスを使用して Zabbix サーバーの trapper ポート（デフォルト: 10051）に接続し、アクティブチェックの設定データを要求します。

3. Hostname パラメータを定義します。この値は、Zabbix Web インターフェースで定義するホスト名と一致している必要があります。この例では、次のようになります。

```
Hostname=Linux server
```

アクティブチェックでは、Zabbix エージェントがこの値を使用して、サーバーから正しいホスト設定を取得するため、Hostname の値は一致している必要があります。具体的には、エージェントがサーバーへの接続を開始し、Hostname の値を使用して自身を識別します。その後、サーバーはそのホストの監視設定を提供します。これらの値が異なる場合、エージェントは適切な設定を受信できず、メトリクスの欠落や監視上の問題が発生します。

4. Zabbix エージェントを再起動します。

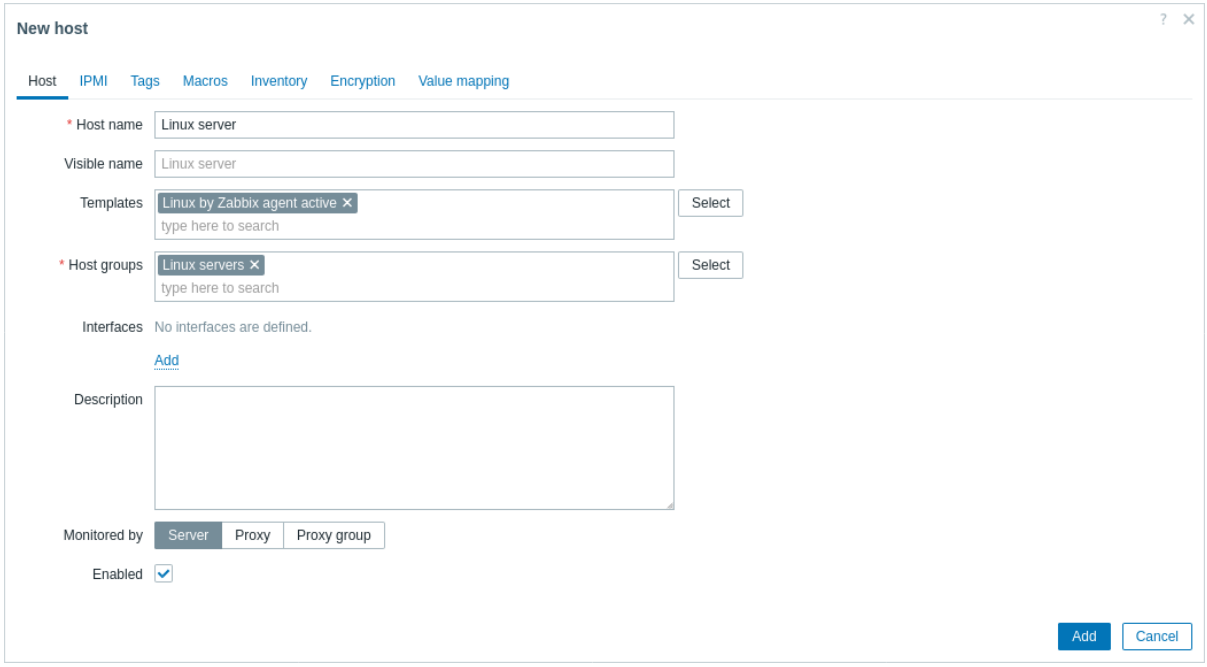
```
systemctl restart zabbix-agent
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

- *Host name* フィールドに、ホスト名（例: ”Linux server”）を入力します。このホスト名は、エージェントの設定ファイルで以前に定義した *Hostname* パラメータの値と一致している必要があります。
- *Templates* フィールドに、ホストにリンクされるテンプレート”Linux by Zabbix agent active” を入力または選択します。
- *Host groups* フィールドに、ホストグループ（例: ”Linux servers”）を入力または選択します。
- *Interfaces* は未定義のままにします。アクティブチェックでは、エージェントがサーバーからの接続を待ち受けるのではなく、サーバーへの接続を開始するため、インターフェースは必要ありません。



3. *Add* をクリックしてホストを追加します。このホストは、監視対象の Linux マシンを表します。

収集されたメトリクスを表示する

おめでとうございます。ここまでで、Zabbix はすでに Linux マシンを監視しています。

収集されたメトリクスを表示するには、*Monitoring* > ホスト
メニューセクション を開き、ホストの横にある *Latest data* をクリックします。

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

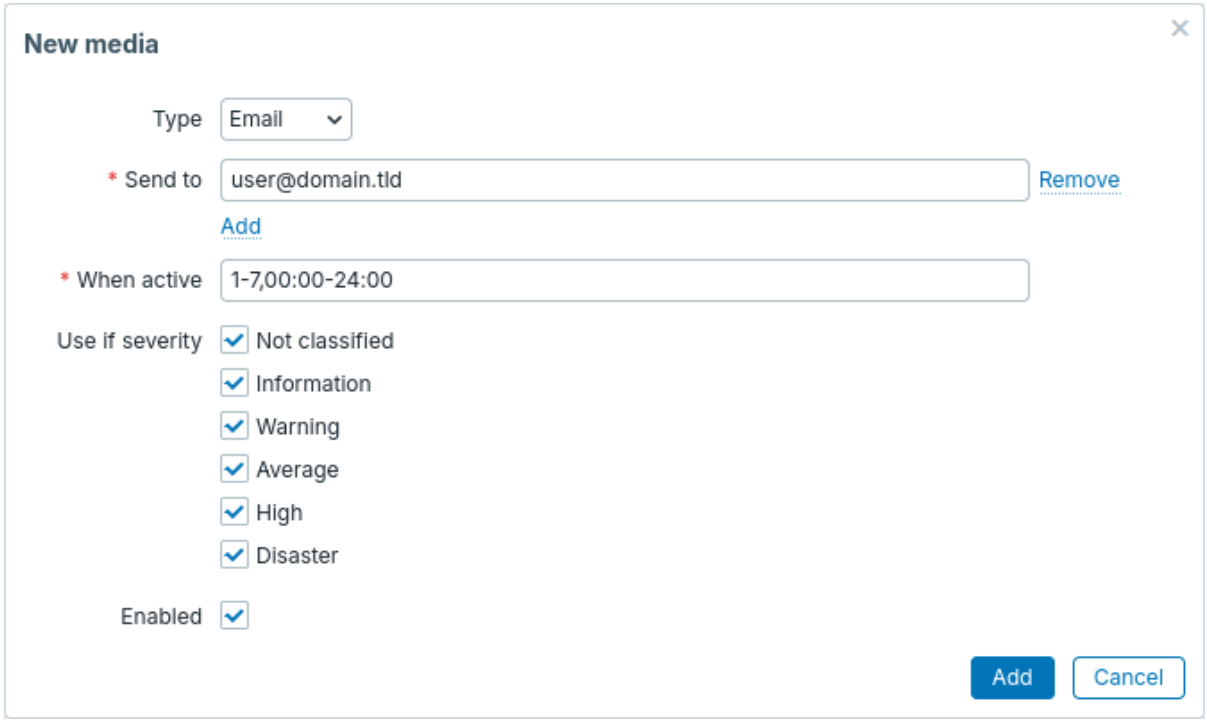
この操作により、Linux サーバーのホストから収集された最新のメトリクス一覧が開きます。

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

障害アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの障害を通知できます。
このガイドでは、メールアラートを送信するための設定手順を説明します。

1. *User settings* > *Profile* に移動し、*Media* タブに切り替えて、
メールアドレスを追加します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出したときに、メールでアラートを受け取れるはずです。

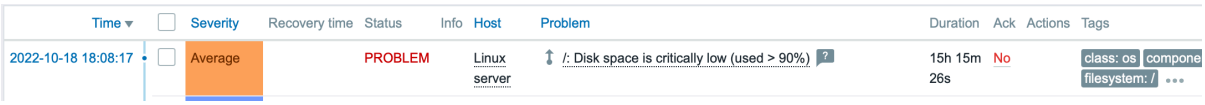
設定をテストする

Linux では、次のコマンドを実行して高い CPU 負荷をシミュレートし、その結果として障害アラートを受け取ることができます。

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

CPU 負荷がしきい値を超えるようにするには、複数の md5sum プロセスを実行する必要がある場合があります。

Zabbix が障害を検出すると、*Monitoring* > *Problems* セクションに表示されます。



アラートが設定されている場合は、障害通知も受信します。

関連項目**:**

- アイテムの作成 - 追加のメトリックの監視を開始する方法（テンプレートを使わないカスタム監視）。
- Zabbix エージェントのアイテム, Windows 用の Zabbix エージェントのアイテム - Windows 上で Zabbix エージェントを使用して監視できるメトリックの完全な一覧。
- 障害のエスカレーション - 複数段階の通知シナリオを作成する方法（例: まずシステム管理者にメッセージを送信し、次に 45 分以内に障害が解決されない場合は、データセンター管理者にメッセージを送信する）。
- パッケージからのインストール - 各種 Linux ディストリビューション向けの公式 RPM および DEB パッケージを使用して Zabbix コンポーネントをインストールし、最新の機能とバグ修正を利用できるようにする方法。

Zabbix エージェントで **Windows** を監視する

はじめに

このページでは、Zabbix を使用して Windows マシンの基本的な監視を開始するために必要な手順を説明します。

このガイドの対象者

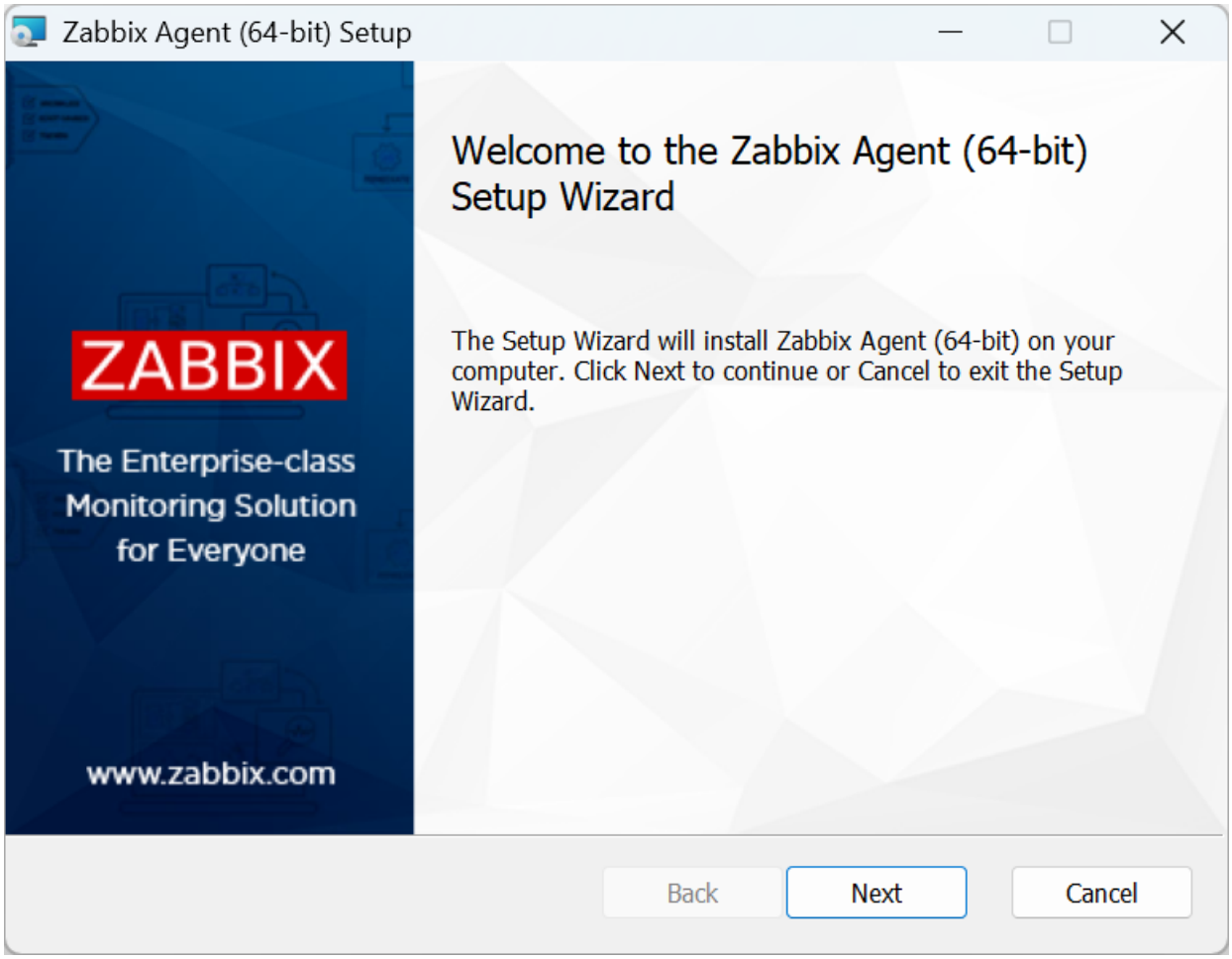
このガイドは、Zabbix を初めて使用するユーザーを対象としており、Windows マシンの基本的な監視を有効にするために必要な最小限の手順を説明します。詳細なカスタマイズオプションを確認したい場合や、より高度な設定が必要な場合は、Zabbix マニュアルの設定 セクションを参照してください。

前提条件

このインストールガイドを進める前に、ご使用の OS 向けの手順に従って、Zabbix サーバーと Zabbix Web インターフェースを ダウンロードしてインストール する必要があります。

Zabbix エージェントのインストール

Zabbix エージェントは、データを収集するためのプロセスです。監視したい Windows マシンにインストールする必要があります。Windows 向けの Zabbix エージェントのインストール手順に従ってください。



監視のための Zabbix の設定

Zabbix エージェントは、アクティブモードまたはパッシブモードでメトリクスを収集できます（同時に使用可能です）。

Note:
パッシブチェックは、単純なデータ要求です。Zabbix サーバーまたはプロキシが何らかのデータ（たとえば CPU 負荷）を要求し、Zabbix エージェントが結果をサーバーに返します。アクティブチェックでは、より複雑な処理が必要です。エージェントはまず、独立して処理するためのアイテム一覧をサーバーから取得し、その後、データをまとめて送信します。詳細はPassive and active agent checksを参照してください。

Zabbix が提供する監視テンプレートには、通常 2 つの選択肢があります。Zabbix エージェント用のテンプレートと、Zabbix エージェント（アクティブ）用のテンプレートです。最初の □□□□□□ では、エージェントはパッシブモードでメトリクスを収集します。これらのテンプレートは、通信プロトコルは異なりますが、同一の監視結果を提供します。

その後の Zabbix の設定は、アクティブまたはパッシブの Zabbix エージェントチェック用テンプレートを選択するかどうかによって異なります。

パッシブチェック

Zabbix Web インターフェース

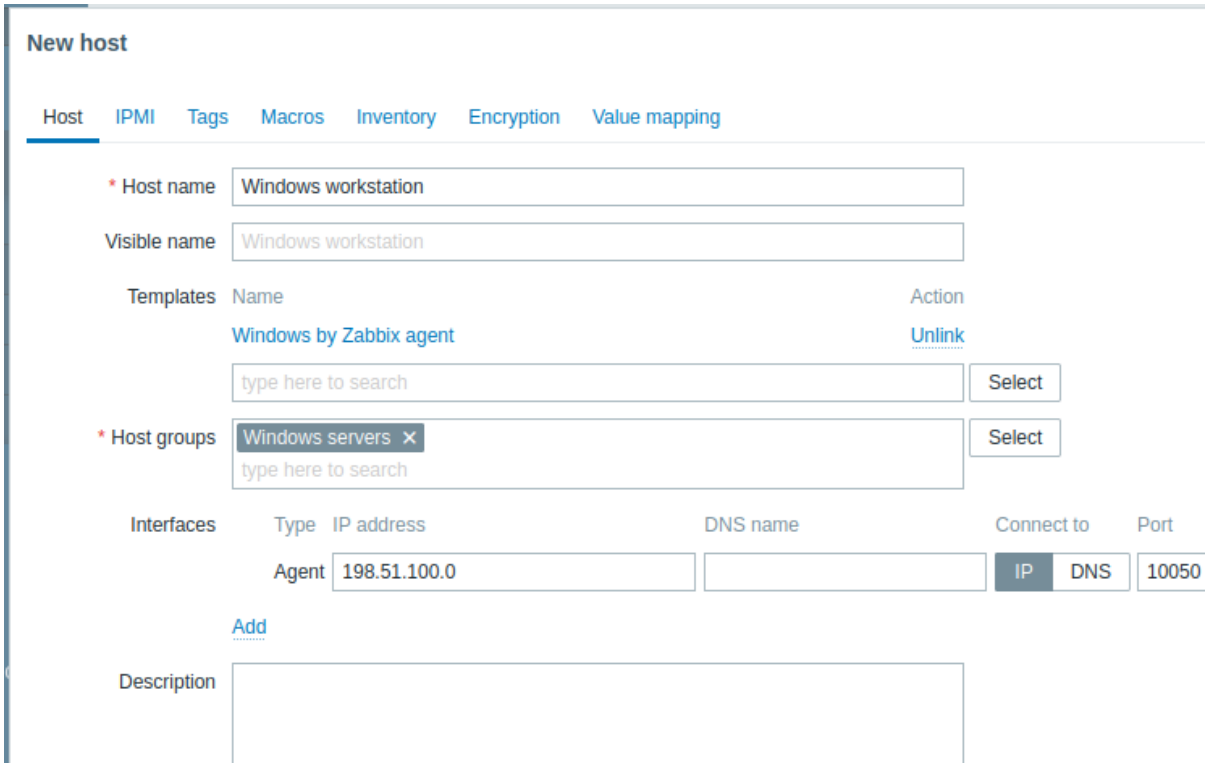
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

このホストは Windows マシンを表します。

3. *Interfaces* パラメータで *Agent* インターフェースを追加し、エージェントがインストールされている Windows マシンの IP アドレスまたは DNS 名を指定します。

4. *Templates* パラメータで、*Windows by Zabbix agent* を入力または選択します。



Zabbix エージェント

パッシブチェックの場合、Zabbix エージェントは Zabbix サーバーの IP アドレスまたは DNS 名を認識している必要があります。エージェントのインストール時に正しい情報を入力している場合は、設定ファイルがすでに更新されています。それ以外の場合は、手動で指定する必要があります。エージェント設定ファイルを開き、Zabbix サーバーの IP アドレスまたは DNS 名を `Server` パラメータに追加します。

例:

```
Server=192.0.2.22
```

アクティブチェック

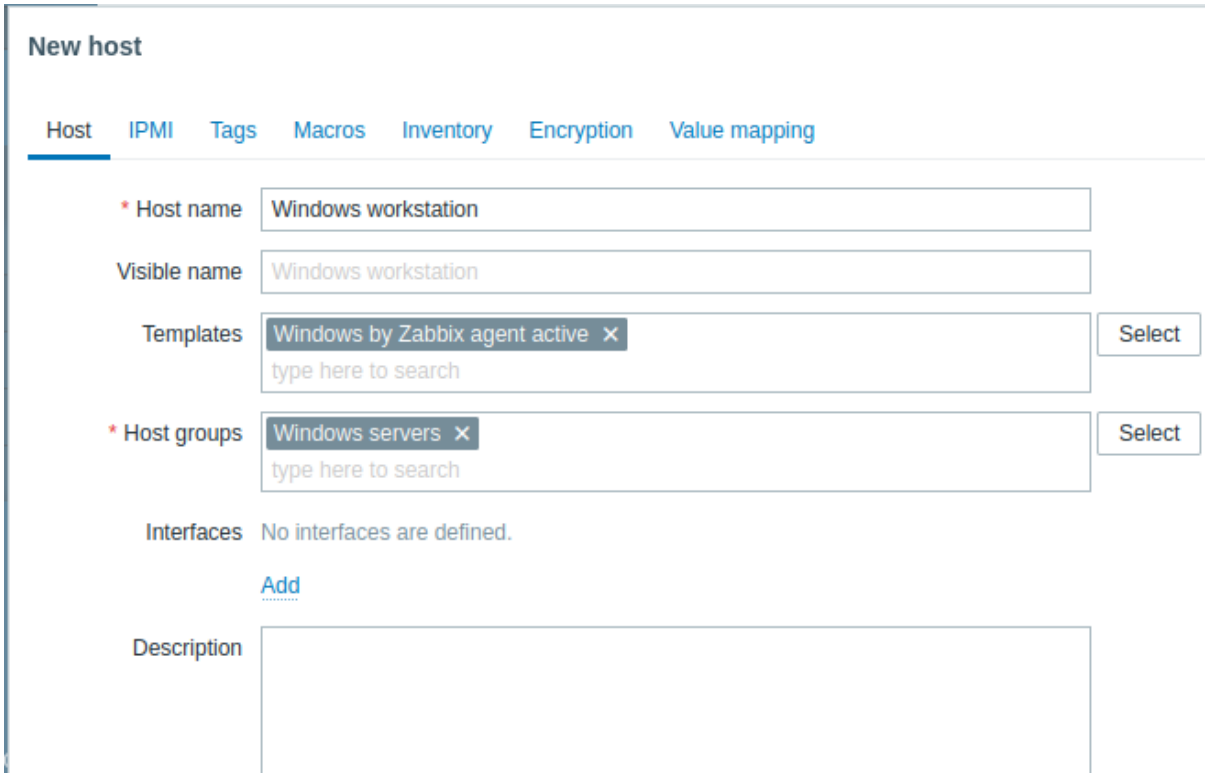
Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

このホストは、Windows マシンを表します。

3. *Templates* パラメーターに、*Windows by Zabbix agent active* を入力または選択します。



Zabbix エージェント

`C:\Program files\Zabbix Agent` フォルダで、`zabbix_agentd.conf` ファイルを開き、次の内容を追加します。

- Zabbix Web インターフェースで作成したホスト名を *Hostname* パラメータに指定します。
- Zabbix サーバーマシンの IP アドレスまたは DNS 名を *ServerActive* パラメータに指定します（Zabbix エージェントのセットアップ時に指定している場合は、あらかじめ入力されていることがあります）。

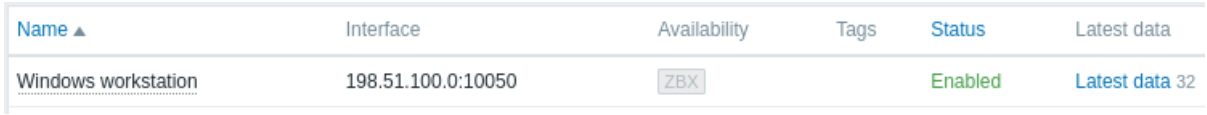
例:

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

収集されたメトリックを表示する

おめでとうございます。ここまでで、Zabbix はすでに Windows マシンを監視しています。

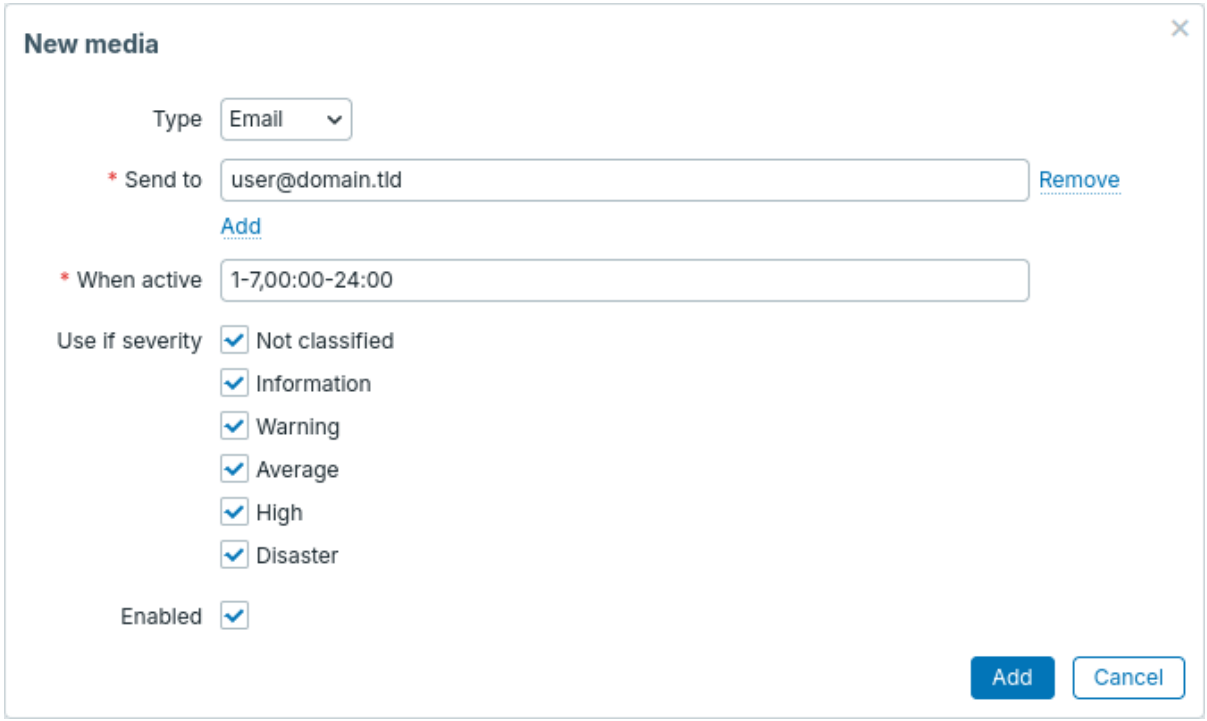
収集されたメトリックを表示するには、*Monitoring* > *Hosts*
メニューセクション を開き、ホストの横にある *Latest data* をクリックします。



障害アラートの設定

Zabbix では、さまざまな方法を使用してインフラストラクチャの障害を通知できます。このガイドでは、メールアラートを送信するための設定手順を説明します。

1. *User settings* > *Profile* に移動し、*Media* タブに切り替えて、メールを追加します。



2. 障害通知の受信 のガイドに従います。

次回、Zabbix が障害を検出すると、メールでアラートを受信できるはずです。

Note:
Windows では、CpuStres ユーティリティを使用して高い CPU 負荷をシミュレートし、その結果として障害アラートを受信できます。

関連項目:

- アイテムの作成 - 追加のメトリクスの監視を開始する方法です（テンプレートを使わないカスタム監視）。
- Zabbix agent アイテム、Windows 向けの Zabbix agent アイテム - Windows 上で Zabbix エージェントを使用して監視できるメトリクスの一覧です。
- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法です（例: まずシステム管理者にメッセージを送信し、障害が 45 分以内に解決されない場合はデータセンター管理者にメッセージを送信する）。

HTTP 経由で **Apache** を監視する

はじめに

このページでは、追加のソフトウェアをインストールせずに Apache Web サーバーの監視をすぐに簡単に開始する方法を紹介します。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、Apache のインストールに対して基本的な監視を有効にするために必要な最小限の手順をまとめています。より詳細なカスタマイズオプションや、さらに高度な設定が必要な場合は、Zabbix マニュアルのConfigurationセクションを参照してください。

前提条件

このインストールガイドを進める前に、OS に応じた手順に従って ダウンロードしてインストール し、Zabbix サーバーと Zabbix Web インターフェースを準備しておく必要があります。

Apache の準備

1. 使用している Apache のバージョンを確認します。

RHEL ベースのシステムでは、次を実行します。

```
httpd -v
```

Debian/Ubuntu では、次を実行します。

```
apache2 -v
```

2. Apache インスタンスでStatus moduleが有効になっていることを確認します。

RHEL ベースのシステムでは、次を実行します。

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

Debian/Ubuntu では、次を実行します。

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

一覧に status_module が表示されない場合は、次を実行してモジュールを有効にします。

RHEL ベースのシステムでは、次を実行します。

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

Debian/Ubuntu では、次を実行します。

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

3. Apache の設定ファイルを編集して、Zabbix サーバーの IP からステータスレポートへアクセスできるようにします。

RHEL ベースのシステムでは、`/etc/httpd/conf.modules.d/status.conf`:

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

Debian/Ubuntu では、`/etc/apache2/mods-enabled/status.conf`:

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

次の行をファイルに追加します（**198.51.100.255** は Zabbix サーバーの IP アドレスに置き換えてください）。

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

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

4. Apache を再起動します

RHEL ベースのシステムでは、次を実行します。

```
sudo systemctl restart httpd
```

Debian/Ubuntu では、次を実行します。

```
sudo systemctl restart apache2
```

5. すべてが正しく設定されているか確認するには、次を実行します（**198.51.100.255** は Zabbix サーバーの IP アドレスに置き換えてください）。

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

応答には Apache Web サーバーの統計情報が含まれているはずです。

Zabbix の監視設定

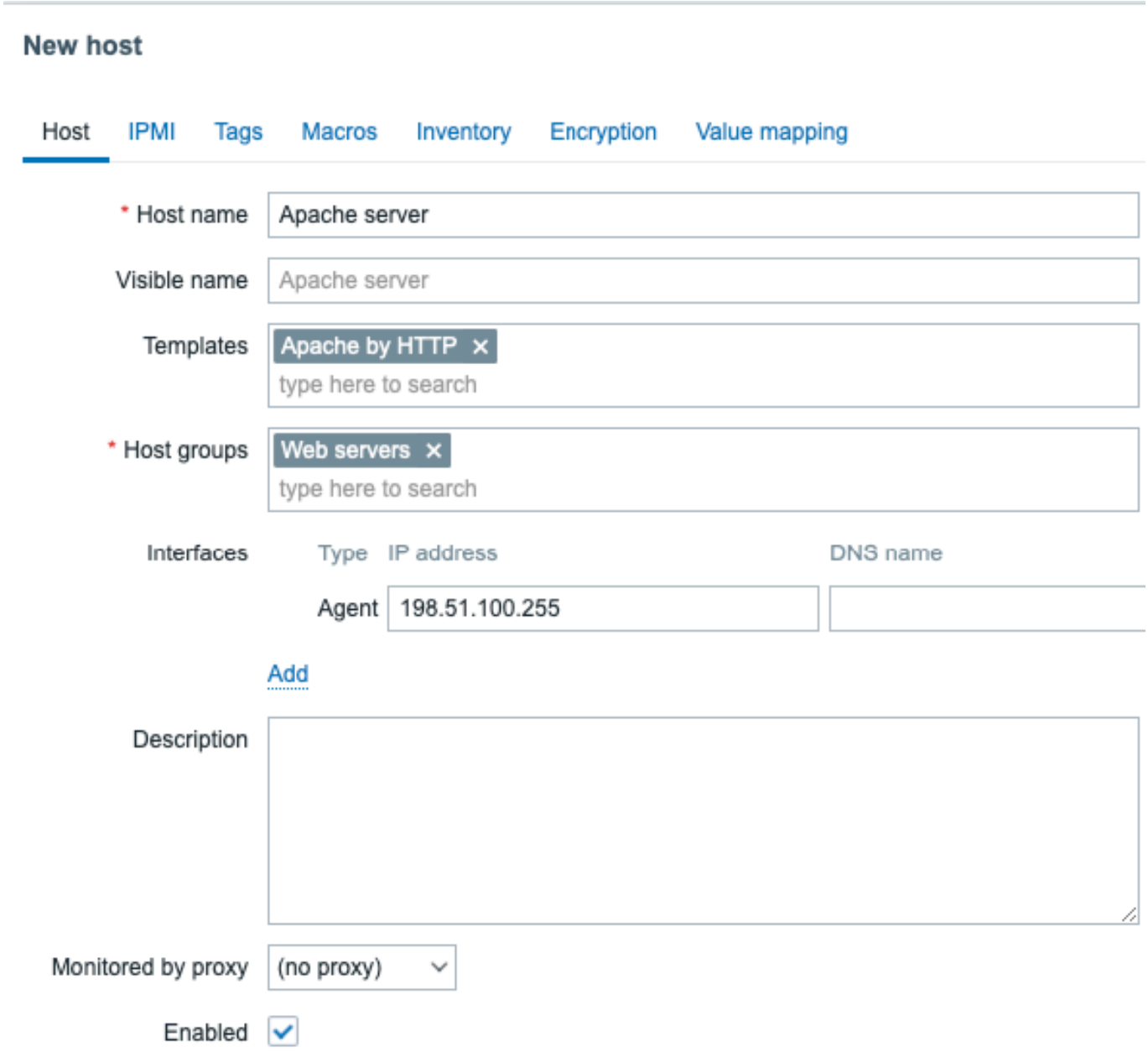
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

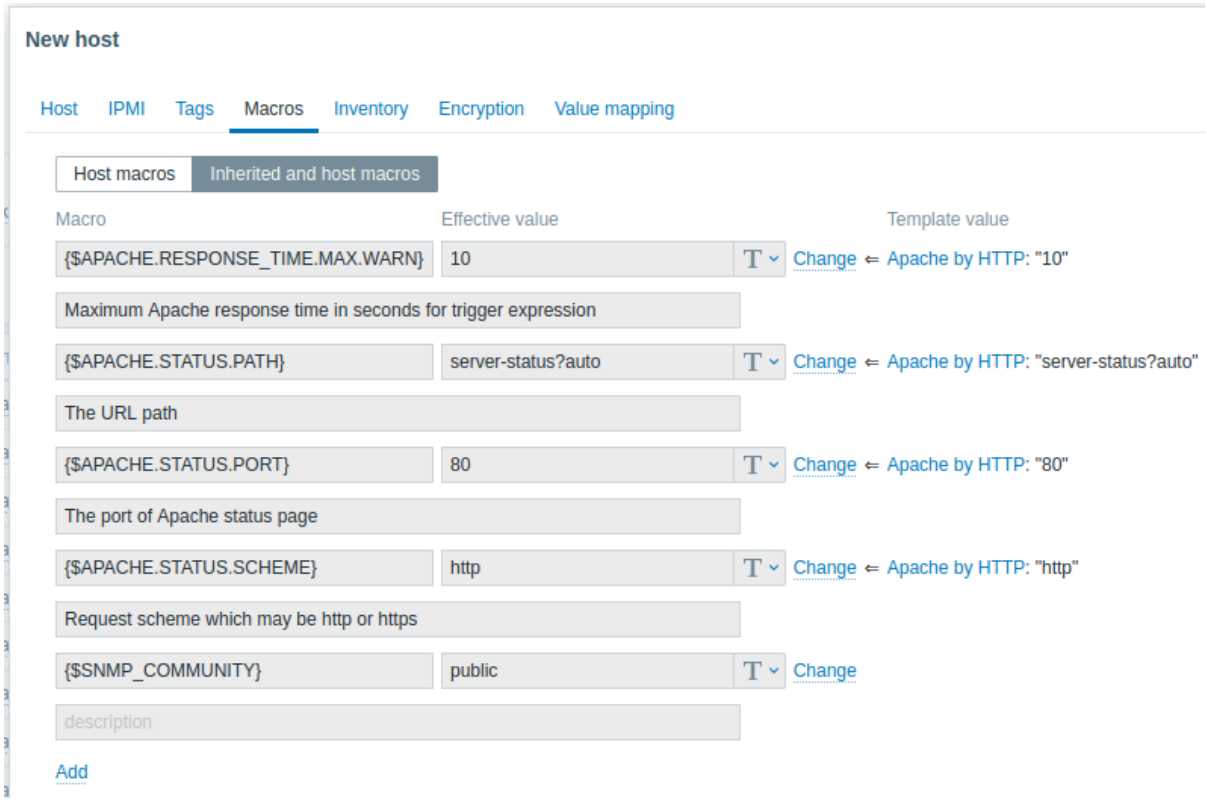
このホストは Apache サーバーを表します。

3. *Interfaces* パラメータで *Agent* インターフェースを追加し、Apache インスタンスの IP アドレスを指定します。このマシンに **Zabbix** エージェントをインストールする必要はありません。インターフェースは {HOST.CONN} マクロの解決にのみ使用されます。このマクロは、Apache インスタンスを特定するためにテンプレートのアイテムで使用されます。

4. *Templates* パラメータで *Apache by HTTP* を入力または選択します。



5. **Macros** タブに切り替え、*Inherited and host macros* モードを選択します。マクロ {$APACHE.STATUS.PORT} と {$APACHE.STATUS.SCHEME} の値が環境設定に合っていることを確認します。デフォルトでは、ポートは 80、スキームは http です。別のポートやスキームを使用する場合は、マクロの値を変更してください。



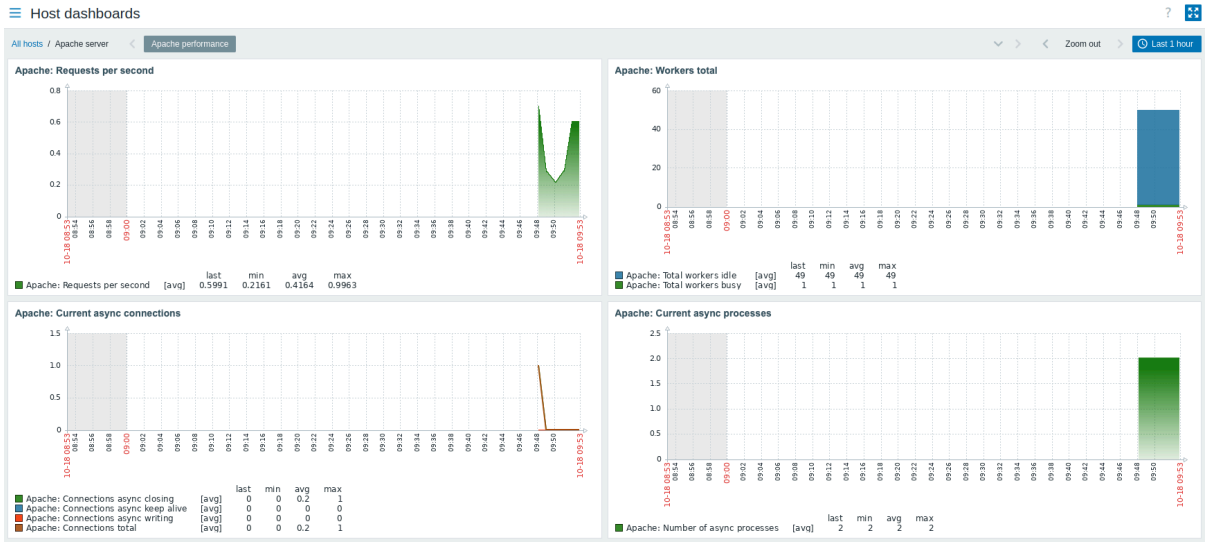
収集されたメトリクスを表示する

おめでとうございます！この時点で、Zabbix はすでに Apache Web サーバーを監視しています。

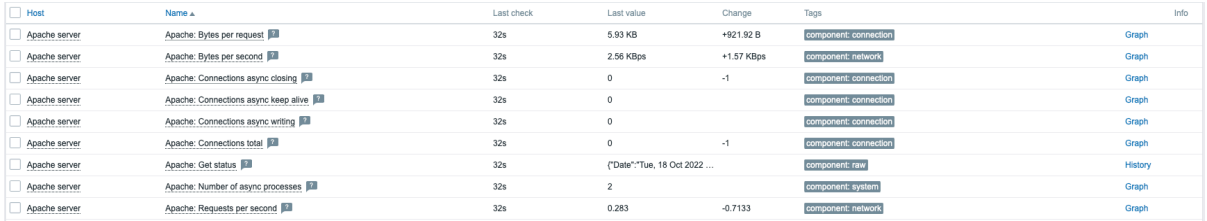
収集されたメトリクスを表示するには、*Monitoring* > *Hosts*
menu section を開き、ホストの横にある *Dashboards* をクリックします。



この操作により、Apache の /server-status ページから収集された最も重要なメトリクスを含むホストダッシュボードが表示されます。



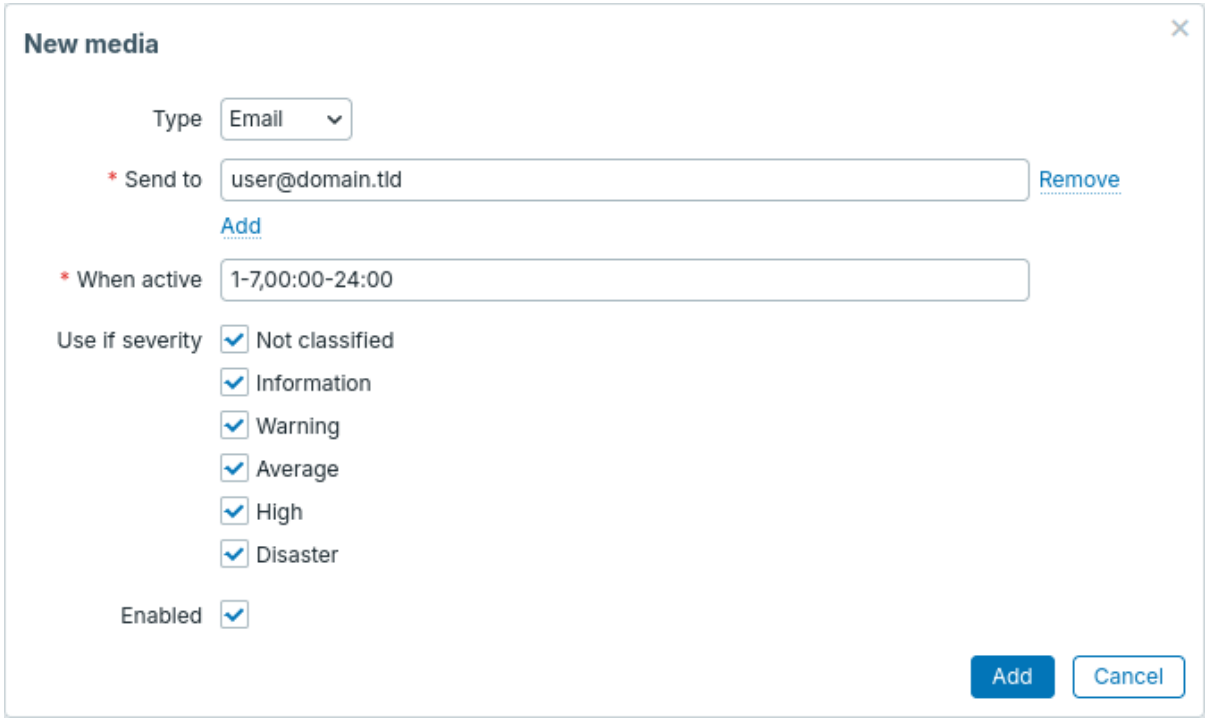
または、*Monitoring* > *Hosts* から *Latest data* をクリックすると、収集された最新のメトリクスをすべて一覧で表示できます。



障害アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの障害を通知できます。
このガイドでは、メールアラートを送信するための設定手順を説明します。

1. ユーザー設定 > プロフィールに移動し、メディアタブに切り替えて、メールを追加します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出したときに、メールでアラートを受信できるはずです。

設定をテストする

実際の障害をシミュレートしてテスト障害アラートを受け取るには、次の手順を実行します。

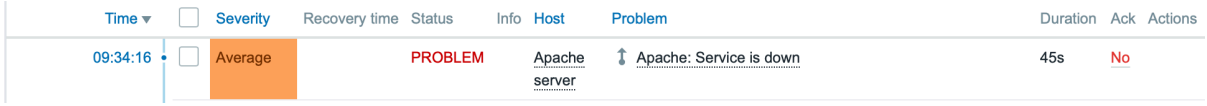
1. Zabbix で *Apache server* ホストの設定を開きます。

2. *Macros* タブに切り替え、*Inherited and host macros* を選択します。

3. {$APACHE.STATUS.PORT} マクロの横にある *Change* をクリックし、別のポートを設定します。

4. *Update* をクリックしてホスト設定を保存します。

5. 数分後、Zabbix は *Apache service is down* という障害を検出します。これは、現在インスタンスに接続できないためです。
この障害は *Monitoring* > *Problems* セクションに表示されます。



アラートが設定されている場合は、障害通知も受信します。

6. 障害を解消して Apache の監視を続けるには、マクロの値を元に戻します。

関連情報**:**

- Web サーバーのセキュリティ強化 - Web サーバーのセキュリティを高めるための推奨設定。
- アイテムの作成 - 追加の監視メトリクスを開始する方法。
- HTTP アイテム - HTTP エージェントを使用してカスタムメトリクスを監視する方法。
- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法
(例: まずシステム管理者にメッセージを送信し、次に、障害が 45 分以内に解決されない場合はデータセンター管理者にメッセージを送信します)。

Zabbix エージェント **2** で **MySQL** を監視する

はじめに

このページでは、MySQL サーバーの基本的な監視を開始するために必要な手順を説明します。

MySQL サーバーを監視するには、いくつかの方法があります。Zabbix エージェント、Zabbix エージェント 2、または Open Database Connectivity (ODBC) 標準です。このガイドでは主に、さまざまな構成でシームレスに設定できるため 推奨される方法である、Zabbix エージェント 2 を使用した MySQL サーバーの監視に焦点を当てています。ただし、このページでは他の方法 についても説明しているため、要件に最も適した方法を自由に選択してください。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、MySQL サーバーの基本的な監視を有効にするために必要な最小限の手順が含まれています。より高度なカスタマイズオプションを探している場合や、さらに詳細な設定が必要な場合は、Zabbix マニュアルのConfigurationセクションを参照してください。

前提条件

このガイドを進める前に、OS 向けの手順に従って ダウンロードしてインストール する必要があります。対象は Zabbix サーバー、Zabbix Web インターフェース、および Zabbix エージェント 2 です。

環境によっては、このガイドの一部の手順が多少異なる場合があります。このガイドは Ubuntu を実行している環境を前提としています。

MySQL ユーザーの作成

MySQL サーバーを監視するには、Zabbix がそのサーバーとそのプロセスにアクセスできる必要があります。
MySQL のインストールには、必要なアクセス権限を持つユーザーがすでに存在します（Zabbix のインストール時に作成された”zabbix” ユーザー）が、
このユーザーには単純な監視には不要な権限も含まれています（データベースの DROP、テーブルからの DELETE などの権限）。
そのため、MySQL サーバーを 監視のみする目的の MySQL ユーザーを作成する必要があります。

1. MySQL クライアントに接続し、”zbx_monitor” ユーザーを作成します（”zbx_monitor” ユーザーの *<password>* を任意のパスワードに置き換えてください）。
そして、ユーザーに必要な権限を GRANT します。

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

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

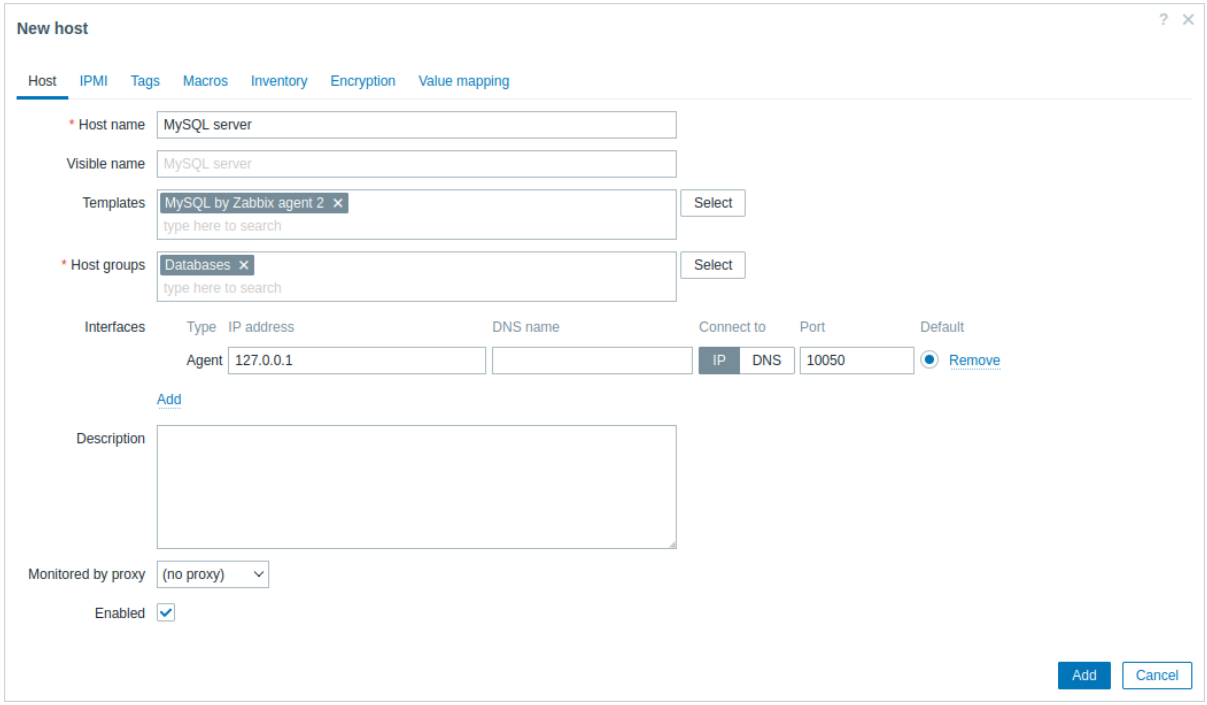
ユーザーの作成が完了したら、次の手順に進めます。

Zabbix Web インターフェースの設定

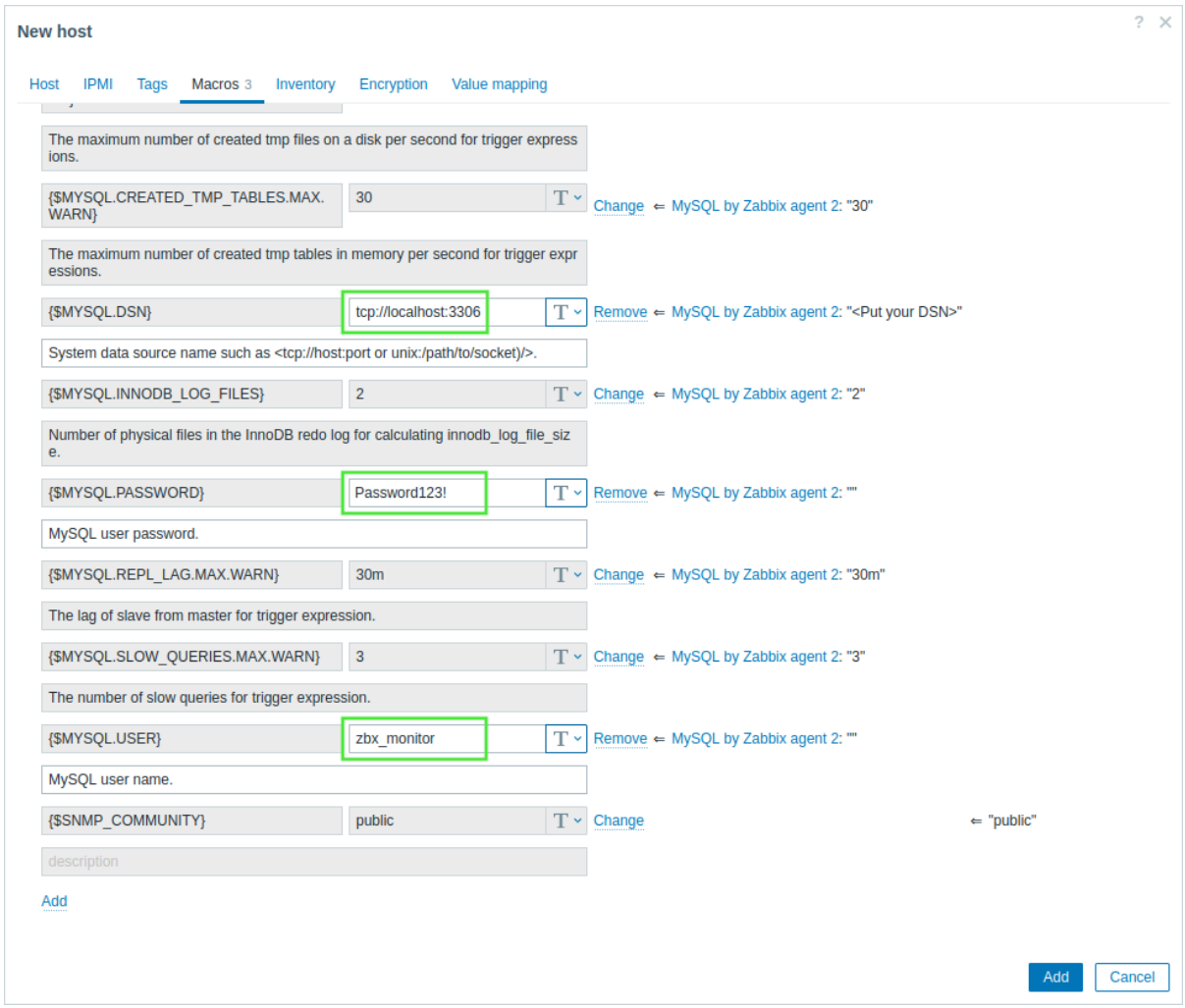
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成 します。

- ホスト名フィールドに、ホスト名を入力します（例: ”MySQL server”）。
- テンプレートフィールドで、ホストにリンク するテンプレート”MySQL by Zabbix agent 2” を入力または選択します。
- ホストグループフィールドで、ホストグループを入力または選択します（例: ”Databases”）。
- インターフェースフィールドで、”Agent” タイプのインターフェースを追加し、MySQL サーバーの IP アドレスを指定します。このガイドでは、Zabbix サーバーと Zabbix エージェント 2 が同じマシンにインストールされている MySQL サーバーを監視するために”127.0.0.1”（localhost）を使用します。



- *Macros* タブで *Inherited and host macros* に切り替え、次のマクロを探して、更新するにはマクロ値の横にある *Change* をクリックします。
  - {$MYSQL.DSN} - MySQL サーバーのデータソースを設定します（MySQL Zabbix agent 2 プラグイン設定ファイル内の名前付きセッションの接続文字列）。このガイドでは、Zabbix サーバーと Zabbix エージェント 2 が同じマシンにインストールされている MySQL サーバーを監視するために、デフォルトのデータソース”tcp://localhost:3306” を使用します。
  - {$MYSQL.PASSWORD} - 以前に作成した MySQL ユーザー ”zbx_monitor” のパスワードを設定します。
  - {$MYSQL.USER} - 以前に作成した MySQL ユーザー ”zbx_monitor” の名前を設定します。

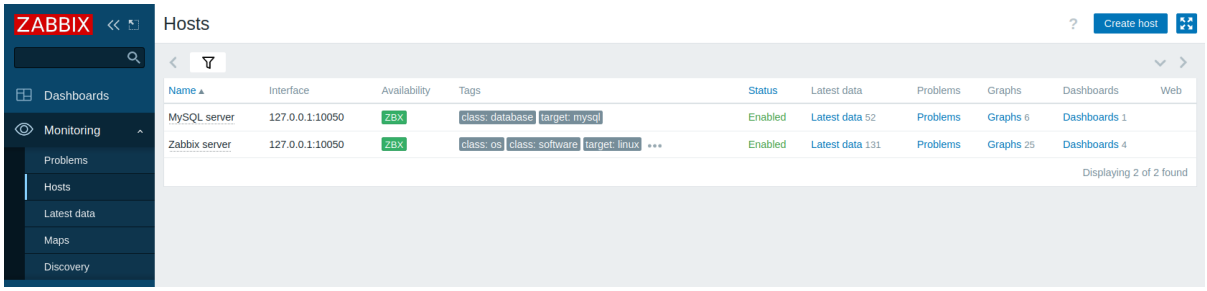


3. *Add* をクリックしてホストを追加します。このホストは MySQL サーバーを表します。

収集されたメトリックを表示する

おめでとうございます。ここまでで、Zabbix はすでに MySQL サーバーを監視しています。

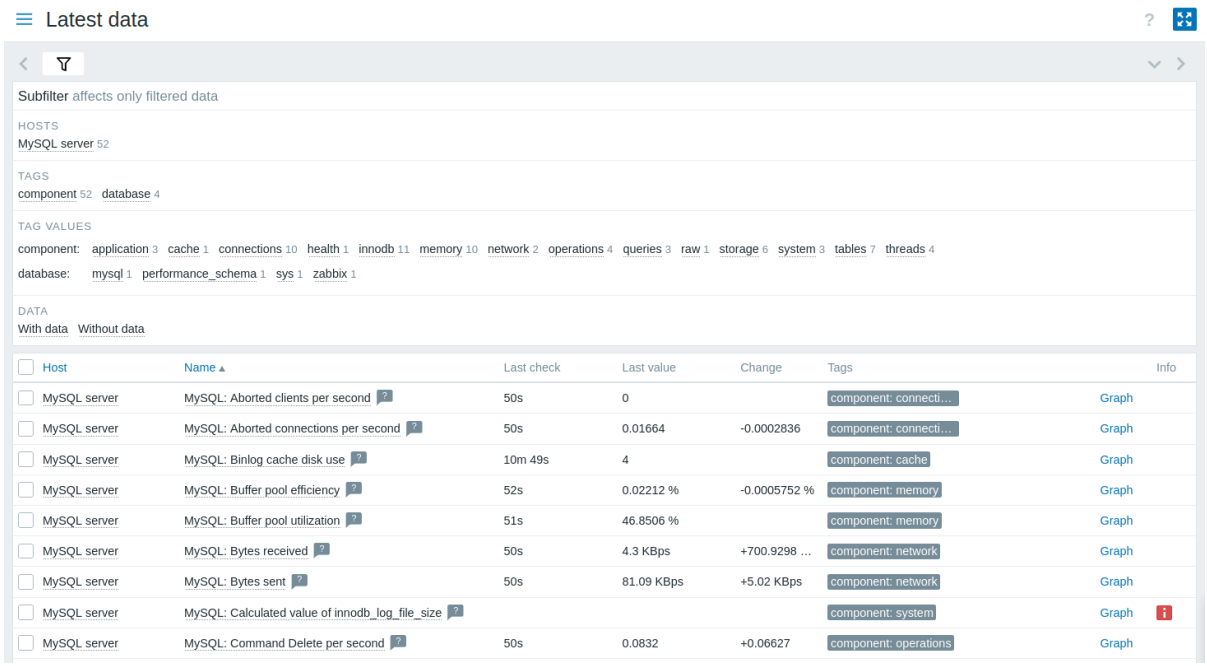
収集されたメトリックを表示するには、*Monitoring > Hosts* メニューセクションに移動し、ホストの横にある *Dashboards* をクリックします。



この操作により、MySQL サーバーから収集された最も重要なメトリックを含むホストダッシュボード（テンプレートレベルで設定済み）に移動します。



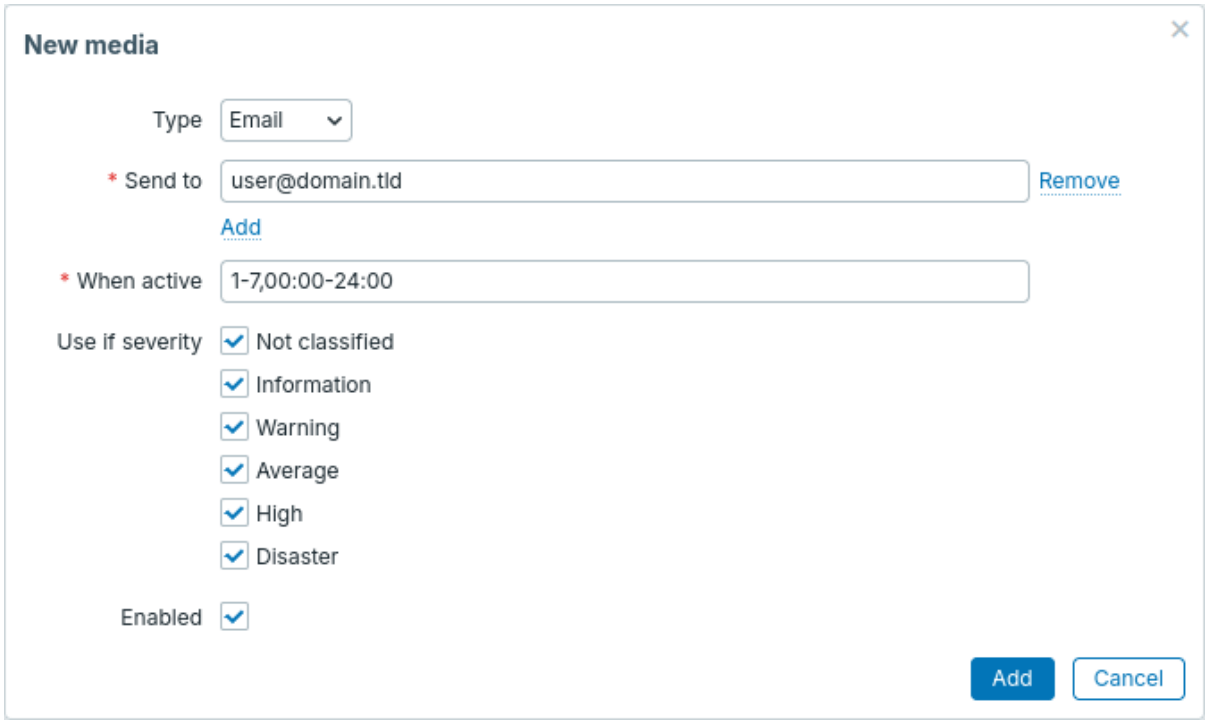
または、*Monitoring > Hosts* メニューセクションから*Latest data* をクリックすると、収集された最新のメトリックを一覧で表示できます。アイテム *MySQL: Calculated value of innodb_log_file_size* は、値が直近 1 時間のデータから計算されるため、データが表示されないことが想定されています。



障害アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの障害を通知できます。
このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロフィール に移動し、メディアタブに切り替えて、メールアドレスを追加 します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出すると、メールでアラートを受け取れるはずです。

設定をテストする

設定をテストするには、Zabbix Web インターフェースでホスト設定を更新し、実際の障害をシミュレートできます。

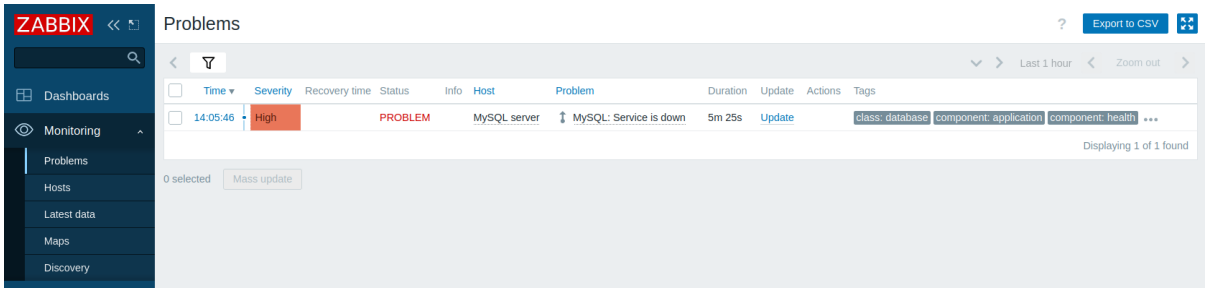
1. Zabbix で MySQL サーバーホストの設定を開きます。

2. *Macros* タブに切り替え、*Inherited and host macros* を選択します。

3. たとえば、以前に設定した {$MYSQL.USER} マクロ値の横にある *Change* をクリックし、別の MySQL ユーザー名を設定します。

4. *Update* をクリックしてホスト設定を更新します。

5. しばらくすると、MySQL サーバーに接続できなくなるため、Zabbix は”MySQL: Service is down” という障害を検出します。障害は*Monitoring > Problems* に表示されます。



アラートが設定されている 場合は、障害通知も受信します。

6. 障害を解消して MySQL サーバーの監視を続けるには、マクロ値を元の値に戻します。

MySQL を監視するための他の方法

Zabbix エージェント 2 を使用して MySQL サーバーを監視する代わりに、Zabbix エージェントまたは Open Database Connectivity (ODBC) 標準を使用することもできます。Zabbix エージェント 2 の使用が推奨されますが、Zabbix エージェント 2 をサポートしない構成や、独自の方法が必要な構成もあります。

Zabbix エージェントと ODBC の主な違いはデータ収集方法にあります。Zabbix エージェントは MySQL サーバーに直接インストールされ、組み込み機能を使用してデータを収集します。一方、ODBC は ODBC ドライバーに依存して MySQL サーバーへの接続を確立し、SQL クエリを使用してデータを取得します。

多くの設定手順は Zabbix エージェント 2 を使用して MySQL サーバーを監視する場合と似ていますが、いくつか重要な違いがあります。MySQL サーバーを監視できるようにするには、Zabbix エージェントまたは ODBC を設定する必要があります。以下の手順では、これらの違いについて説明します。

Zabbix エージェントで MySQL を監視する

Zabbix エージェントで MySQL サーバーを監視するには、OS に応じた手順に従って ダウンロードしてインストール し、Zabbix サーバー、Zabbix Web インターフェース、および Zabbix エージェントを導入する必要があります。

必要な Zabbix コンポーネントのインストールが正常に完了したら、*MySQL ユーザーの作成* セクションで説明されているとおりに MySQL ユーザーを作成する必要があります。

MySQL ユーザーを作成したら、Zabbix エージェントが MySQL サーバーとの接続を確立して監視できるように設定する必要があります。これには、カスタムのエージェントチェックを実行するための複数のuser parameters の設定に加え、前述の ”zbx_monitor” ユーザーとして MySQL サーバーに接続するために必要な認証情報を Zabbix エージェントに提供することが含まれます。

Zabbix エージェントの設定

1. Zabbix エージェントの追加設定ディレクトリに移動します。

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

> **Attention:**
> Zabbix エージェントの追加設定ディレクトリは、Zabbix のエージェント設定ファイルと同じディレクトリに配置する必要があります。デフォルトのパスは、OS と Zabbix のインストール方法によって異なります。

MySQL サーバーを監視するために必要なすべてのユーザーパラメータを Zabbix エージェントの設定ファイルに定義する代わりに、これらのパラメータを追加設定ディレクトリ内の別のファイルに定義します。

2. Zabbix エージェントの追加設定ディレクトリに `template_db_mysql.conf` ファイルを作成します。

```
vi template_db_mysql.conf
```

3. Zabbix リポジトリにある `template_db_mysql.conf` ファイルの内容を、作成した `template_db_mysql.conf` ファイルにコピーして保存します。

4. 設定を更新するため、Zabbix エージェントを再起動します。

```
systemctl restart zabbix-agent
```

Zabbix エージェントのユーザーパラメータを設定したら、Zabbix エージェントが MySQL サーバーにアクセスできるようにする認証情報の設定に進むことができます。

5. Zabbix エージェントのホームディレクトリに移動します（システムに存在しない場合は作成する必要があります。デフォルト: */var/lib/zabbix*）。

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

6. Zabbix エージェントのホームディレクトリに *.my.cnf* ファイルを作成します。

```
vi .my.cnf
```

7. 次の内容を *.my.cnf* ファイルにコピーします（*<password>* を”zbx_monitor” ユーザーのパスワードに置き換えます）。

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

Zabbix Web インターフェースを設定し、設定をテストする

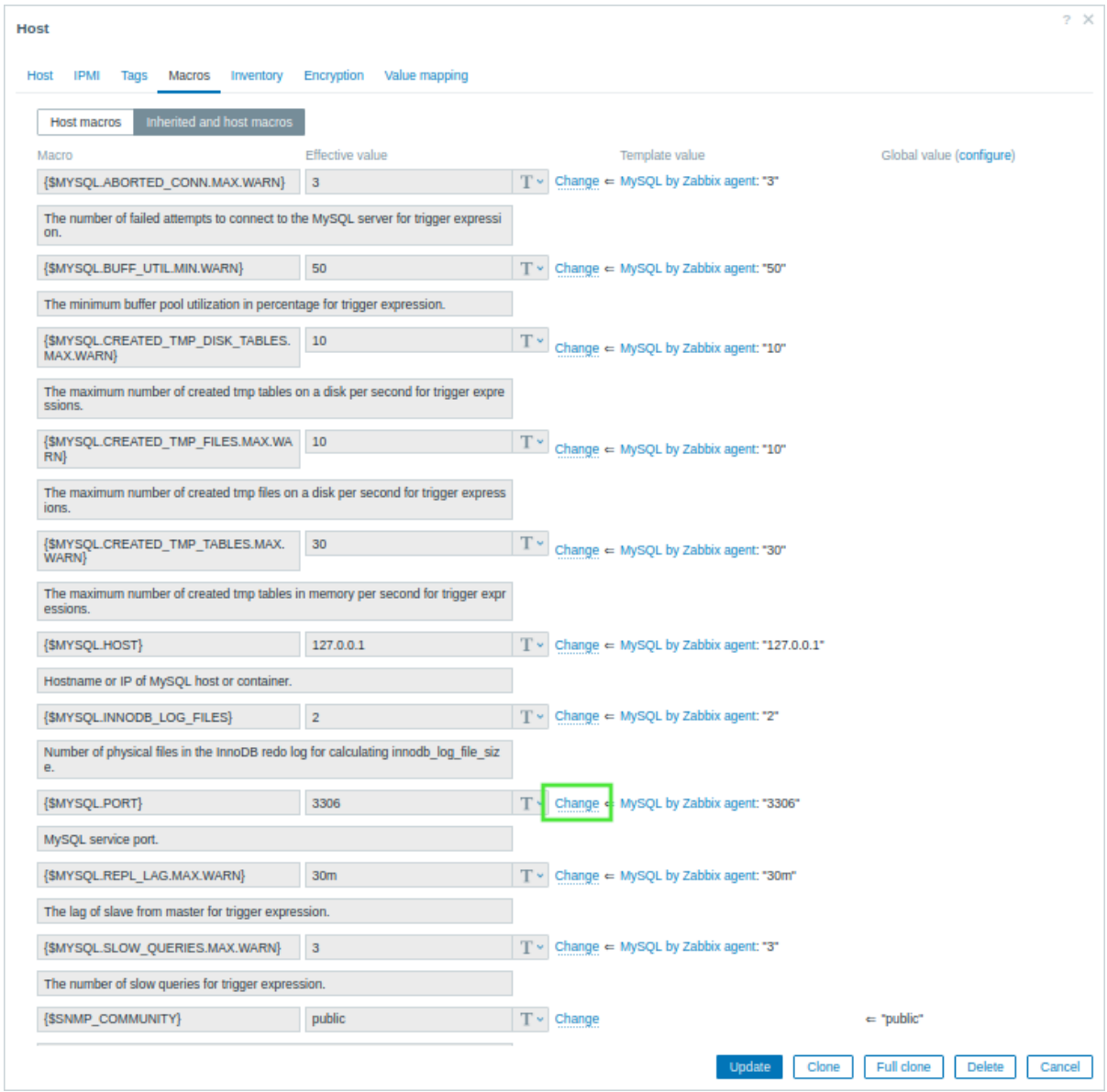
Zabbix Web インターフェースを設定するには、次の調整を加えたうえで、*Zabbix Web インターフェースを設定する* セクションの手順に従います。

- *Templates* フィールドで、ホストにリンク されるテンプレート”MySQL by Zabbix agent” を入力または選択します。
- *Macros* の設定は不要です。

Zabbix Web インターフェースを設定したら、収集したメトリクスを表示 し、障害アラートを設定 できます。

設定をテストするには、次の調整を加えたうえで、設定をテストする セクションの手順に従います。

- MySQL サーバーホストの設定の *Inherited and host macros* セクションで、{$MYSQL.PORT} マクロ値の横にある *Change* をクリックし、別のポート（例: ”6033”）を設定します。



ODBC で MySQL を監視する

ODBC で MySQL サーバーを監視するには、Zabbix サーバーと Zabbix Web インターフェースをダウンロードしてインストールする必要があります。

必要な Zabbix コンポーネントのインストールが正常に完了したら、*MySQL ユーザーの作成*セクションの説明に従って MySQL ユーザーを作成する必要があります。

MySQL ユーザーを作成したら、ODBC を設定する必要があります。これには、最も一般的に使用されているオープンソースの ODBC API 実装の 1 つである unixODBC と unixODBC ドライバーのインストール、および ODBC ドライバー設定ファイルの編集が含まれます。

ODBC の設定

1. unixODBC をインストールします。unixODBC の推奨されるインストール方法は、Linux オペレーティングシステムのデフォルトのパッケージリポジトリを使用することです。

```
apt install unixodbc
```

2. MariaDB の unixODBC データベースドライバーをインストールします。MySQL データベースを使用している場合でも、互換性のために MariaDB の unixODBC ドライバーを使用します。

```
apt install odbc-mariadb
```

3. ODBC 設定ファイル *odbcinst.ini* と *odbc.ini* の場所を確認します。

```
odbcinst -j
```

このコマンドの実行結果は、次のようになります。

```
unixODBC 2.3.9
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
...
```

4. MySQL データベースを監視するための ODBC ドライバーを設定するには、*odbcinst.ini* ファイルにあるドライバー名が必要です。次の *odbcinst.ini* の例では、ドライバー名は”MariaDB Unicode” です。

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

5. 次の内容を *odbc.ini* ファイルにコピーします（*<password>* は”zbx_monitor” ユーザーのパスワードに置き換えてください）。このガイドでは、ODBC ドライバーと同じマシンにインストールされた MySQL サーバーを監視するために、”127.0.0.1”（localhost）を MySQL サーバーアドレスとして使用します。データソース名（DSN）”test” に注意してください。これはZabbix Web インターフェースの設定 時に必要になります。

```
[test]
Driver=MariaDB Unicode
Server=127.0.0.1
User=zbx_monitor
Password=<password>
Port=3306
Database=zabbix
```

Zabbix Web インターフェースの設定と設定のテスト

Zabbix Web インターフェースを設定するには、次の調整を加えたうえで、*Zabbix Web インターフェースの設定* セクションの手順に従ってください。

- *Templates* フィールドで、ホストにリンク されるテンプレート”MySQL by ODBC” を入力または選択します。
- *Interfaces* の設定は不要です。
- MySQL サーバーホストの設定にある *Inherited and host macros* セクションで、{$MYSQL.DSN} マクロの値を *odbc.ini* ファイルの DSN 名に設定します。

Zabbix Web インターフェースを設定したら、収集したメトリクスを表示する、障害アラートを設定する、設定をテストする ことができます。

参照

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 障害エスカレーション - 複数ステップのアラートシナリオを作成する方法（例: まずシステム管理者にメッセージを送信し、次に障害が 45 分以内に解決されない場合はデータセンター管理者にメッセージを送信する）。
- ODBC 監視 - 他の Linux ディストリビューションで ODBC を設定する方法、および ODBC を使用して追加のデータベース関連メトリクスの監視を開始する方法。

- テンプレート *MySQL by Zabbix エージェント* - *MySQL by Zabbix* エージェントテンプレートに関する追加情報。
- テンプレート *MySQL by Zabbix エージェント 2* - *MySQL by Zabbix* エージェント *2* テンプレートに関する追加情報。
- テンプレート *MySQL by ODBC* - *MySQL by ODBC* テンプレートに関する追加情報。

Zabbix で **VMware** を監視する

はじめに

このページでは、VMware の基本的な監視を開始するために必要な手順を説明します。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、VMware の基本的な監視を有効にするために必要な最小限の手順をまとめています。より詳細なカスタマイズオプションを探している場合や、より高度な設定が必要な場合は、Zabbix マニュアルのVirtual machine monitoring セクションまたはConfiguration セクションを参照してください。

前提条件

このガイドを進める前に、OS に応じた手順に従って Zabbix サーバーと Zabbix Web インターフェースをダウンロードしてインストール する必要があります。

環境によっては、このガイドの一部の手順が多少異なる場合があります。このガイドは Ubuntu を実行している環境を前提としています。

VMware はすでに設定済みであることを前提としています。このガイドでは VMware の設定については扱いません。

Zabbix サーバーの設定

VMware を監視するには、*vmware collector* Zabbix プロセスを有効にする必要があります。VMware 監視の実行方法については、仮想マシンの監視を参照してください。

1. Zabbix サーバー設定ファイルを開きます。

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

2. Zabbix サーバー設定ファイルで、StartVMwareCollectorsパラメータを 2 以上に設定します（デフォルト値は 0 です）。

```
###### Option: StartVMwareCollectors
#####       Number of pre-forked vmware collector instances.
#####
##### Mandatory: no
##### Range: 0-250
##### Default:
##### StartVMwareCollectors=0

StartVMwareCollectors=2
```

3. Zabbix サーバーを再起動します。

```
systemctl restart zabbix-server
```

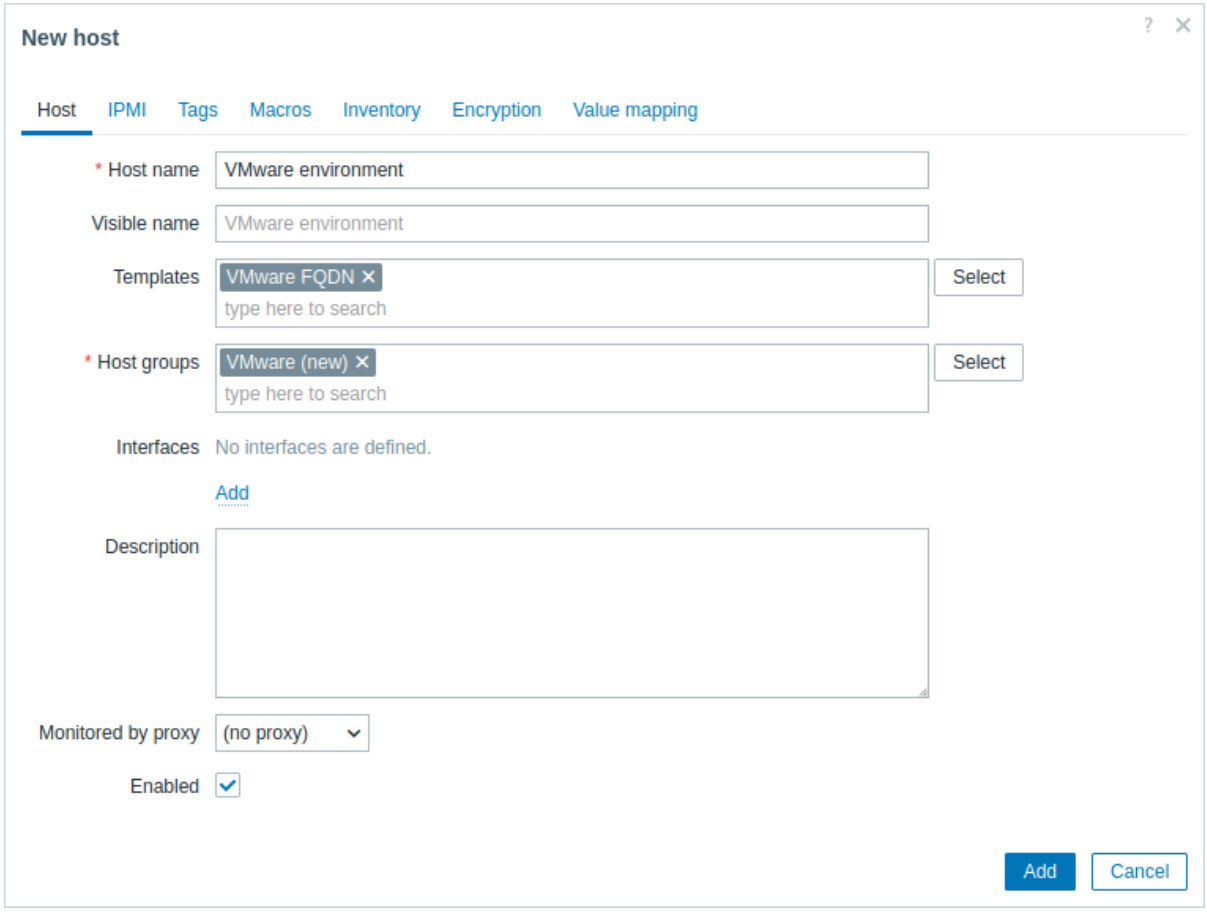
vmware collector プロセスが起動したら、次の手順に進みます。

Zabbix Web インターフェースの設定

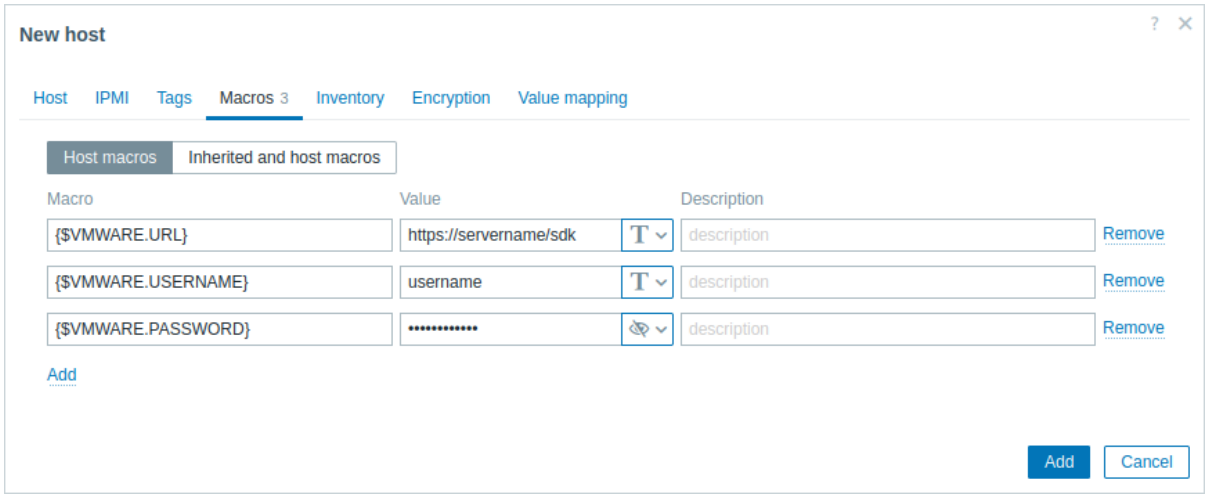
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成 します。

- *Host name* フィールドにホスト名を入力します（例: ”VMware environment”）。
- *Templates* フィールドに”VMware FQDN” テンプレート（または”VMware” テンプレート）を入力または選択します。これらのテンプレートの詳細については、仮想マシン監視 を参照してください。
- *Host groups* フィールドにホストグループを入力または選択します（例: 新しいホストグループ”VMware”）。



- *Macros* タブで、次のホストマクロを設定します:
  - {$VMWARE.URL} - VMware サービス（vCenter または ESXi ハイパーバイザー）の SDK URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - VMware サービスのユーザー名
  - {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーのパスワード



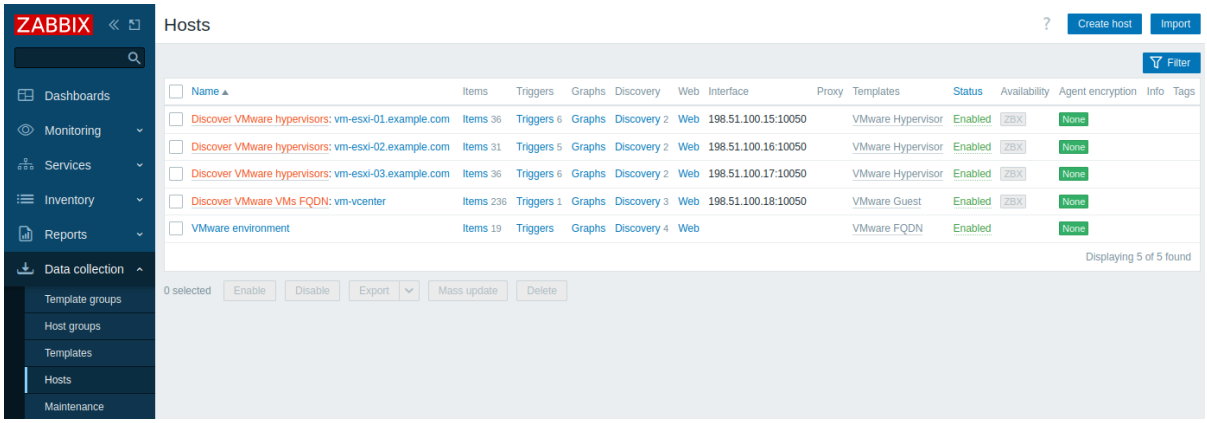
3. *Add* ボタンをクリックしてホストを作成します。このホストは VMware 環境を表します。

収集されたメトリックを表示する

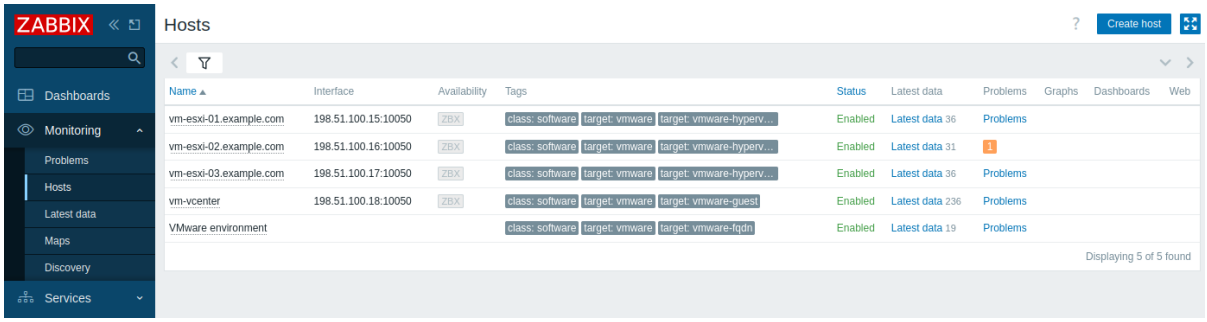
おめでとうございます。ここまでで、Zabbix はすでに VMware 環境の監視を開始しています。

VMware 環境の構成によっては、Zabbix が検出 を行い、その後、検出されたエンティティに対してホストを作成する場合があります。必要に応じて、ホストの検出と作成は手動で実行 することもできます。

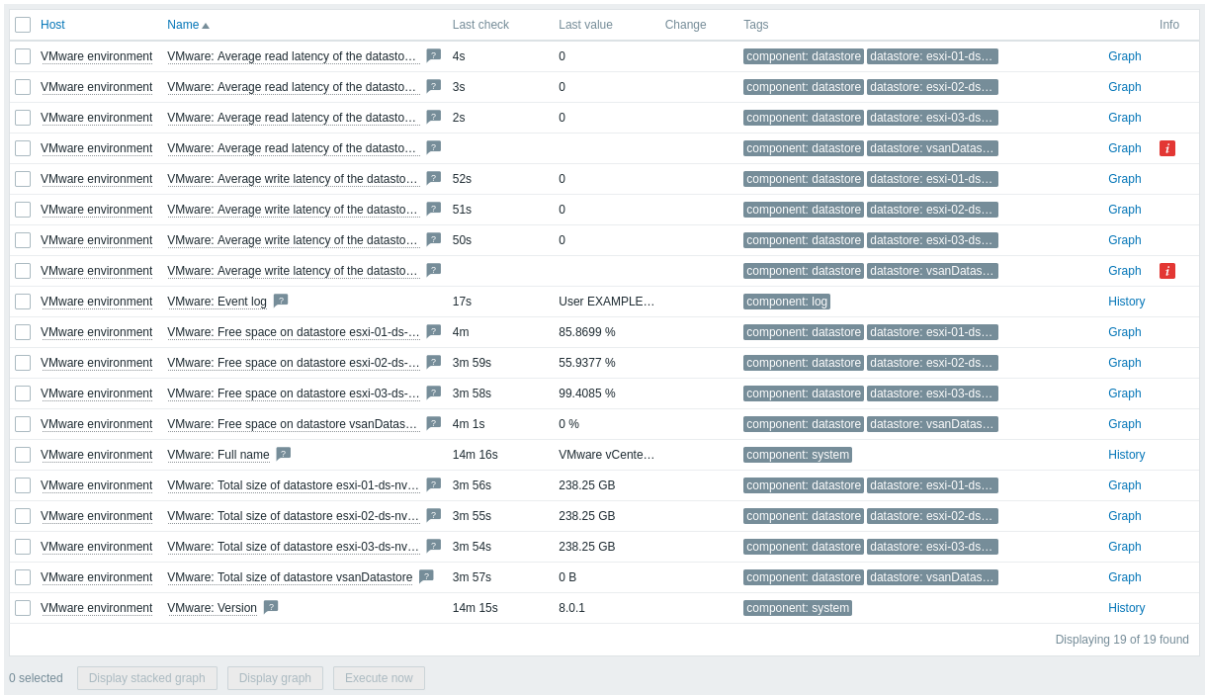
作成されたホストを表示するには、データ収集 > ホスト メニューセクションに移動します。



収集されたメトリックを表示するには、監視 > ホスト メニューセクションに移動し、作成された”VMware environment” ホスト、または検出されたエンティティ用に作成されたホストのいずれかの横にある 最新データをクリックします。



この操作により、選択したホストから収集されたすべての最新メトリックの一覧が開きます。

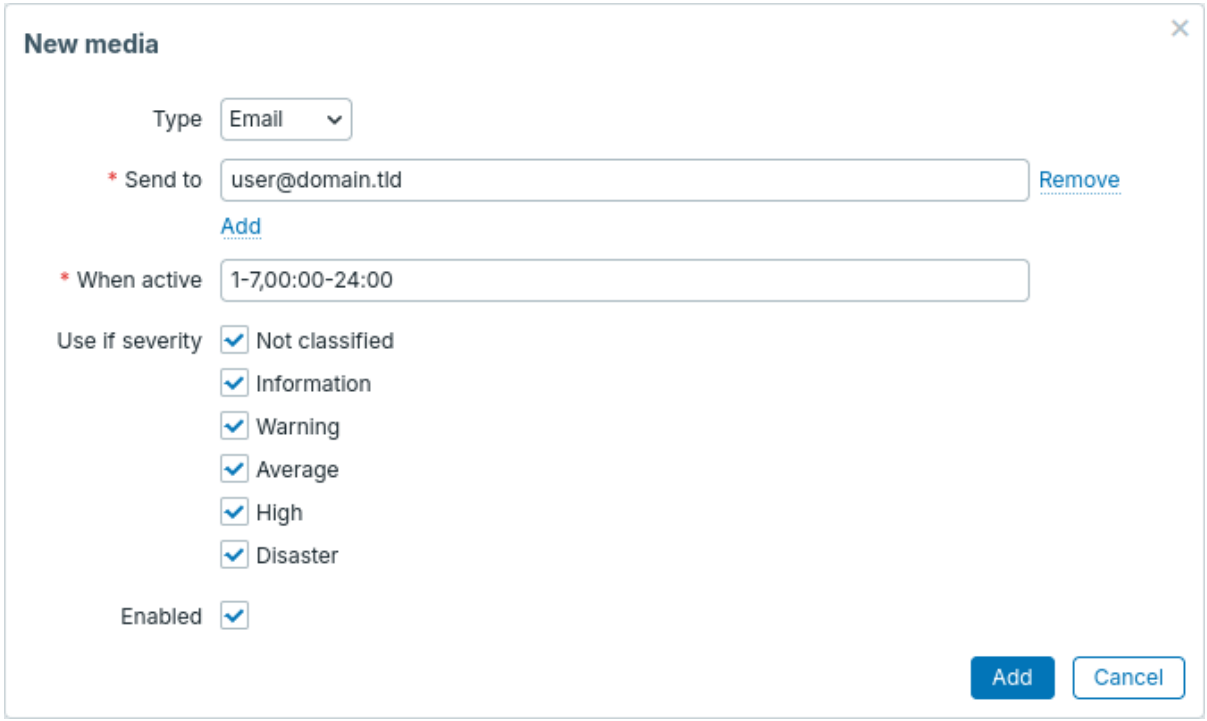


一部のアイテムにはデータがなく、サポートされていませんの状態になっている点に注意してください。これは、監視対象の VMware 環境で有効になっていないため、Zabbix が特定のデータストア上で有効なパフォーマンスカウンター を見つけられないためです。

障害アラートの設定

Zabbix は、さまざまな方法でインフラストラクチャの障害を通知できます。
このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロフィール に移動し、メディアタブに切り替えて、メールアドレスを追加 します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出すると、メールでアラートを受け取れるはずです。

関連情報

- アイテムの作成 - 追加の監視メトリクスの監視を開始する方法。
- 障害エスカレーション - 複数段階のアラートシナリオを作成する方法（例: まずシステム管理者にメッセージを送信し、次に、45 分以内に障害が解決されない場合はデータセンター管理者にメッセージを送信する）。
- 仮想マシン監視 - VMware 監視に関する追加情報（データ収集プロセス、サーバー設定オプション、トラブルシューティングガイダンスなど）。
- VMware 監視のアイテムキー - Zabbix で監視できる VMware メトリクスの完全な一覧。
- テンプレート *VMware* - *VMware* テンプレートに関する追加情報。
- テンプレート *VMware FQDN* - *VMware FQDN* テンプレートに関する追加情報。

Zabbix でネットワークトラフィックを監視する

はじめに

このページでは、Zabbix を使用してネットワークトラフィックの基本的な監視を開始するために必要な手順を説明します。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、ネットワークトラフィックの基本的な監視を有効にするために必要な最小限の手順をまとめています。より高度なカスタマイズオプションや、さらに詳細な設定が必要な場合は、Zabbix マニュアルのConfigurationセクションを参照してください。

前提条件

このガイドを進める前に、OS の手順に従って ダウンロードしてインストール した Zabbix サーバー、Zabbix Web インターフェース、および Zabbix エージェントが必要です。
Zabbix エージェントは、トラフィック監視が必要なマシンにインストールする必要があることに注意してください。
これは、Zabbix サーバーがインストールされている同じホストでも、別のホストでもかまいません。

このガイドでは、*Remote host* という名前の別のマシン上で、*eth0* インターフェースのネットワークトラフィック監視を設定する手順を説明します。

Zabbix の監視設定

Zabbix エージェントは、アクティブモードまたはパッシブモードでメトリクスを収集できます（同時に使用可能です）。詳細はパッシブおよびアクティブなエージェントチェックを参照してください。このガイドでは、パッシブチェックによる監視について説明します。

Zabbix エージェントの設定

1. エージェントがインストールされているマシンで、エージェント設定ファイルを開きます。

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

2. Zabbix サーバーの IP アドレスまたは DNS 名を *Server* パラメータに追加します。例:

```
Server=192.0.2.22
```

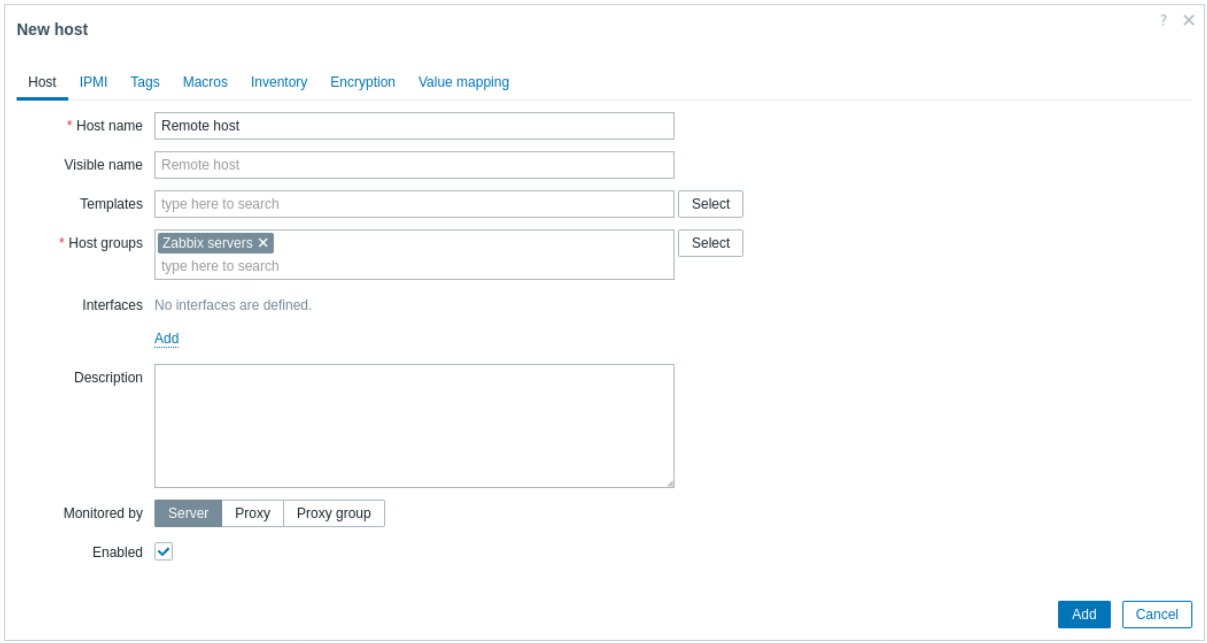
3. Zabbix エージェントを再起動します。

```
systemctl restart zabbix-agent
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. エージェントがインストールされているマシンの IP アドレスまたは DNS 名を指定して、Zabbix Web インターフェースでホストを作成します。

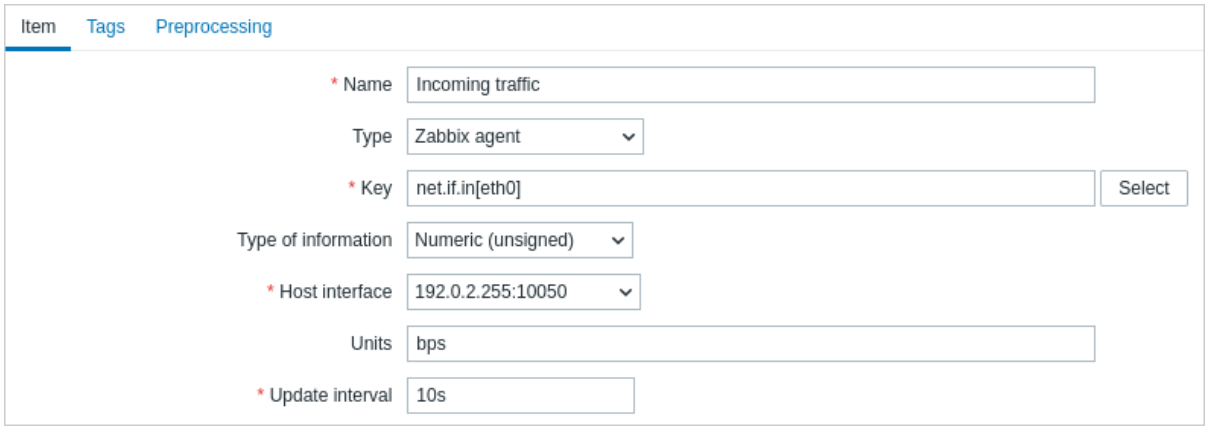


アイテムの作成

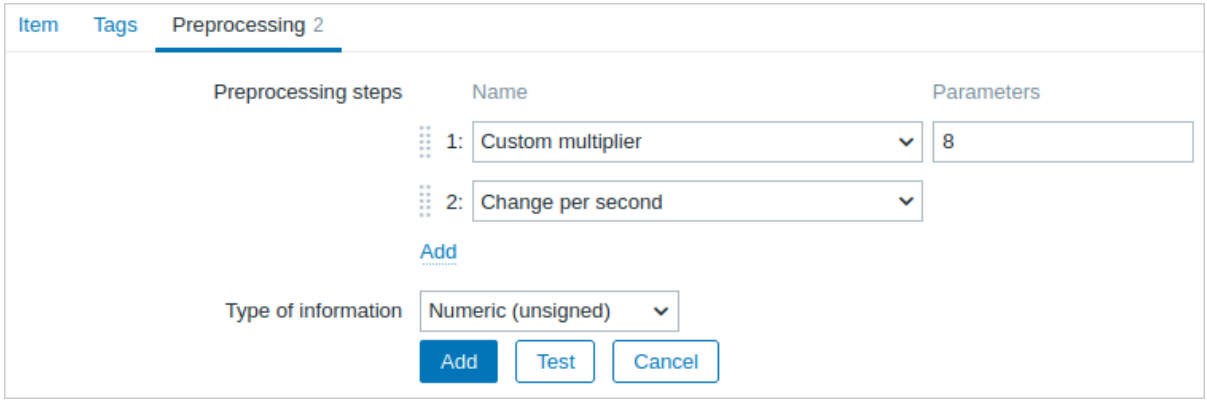
アイテムの作成 の手順に従って、トラフィック監視用のアイテムを追加します。具体的には、次のとおりです。

- 受信トラフィック
- 送信トラフィック
- 合計トラフィック

受信トラフィック監視アイテムの簡単な設定例は、次のようになります。



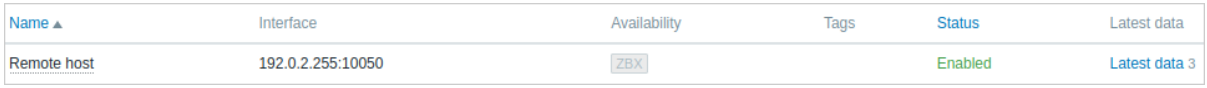
収集したデータを実用的に使えるようにするには、アイテムを作成する際にいくつかの前処理 ステップを設定できます。この場合は、8 倍する処理（バイトをビットに変換するため）と、1 秒あたりの変化量として表示する処理が該当します。



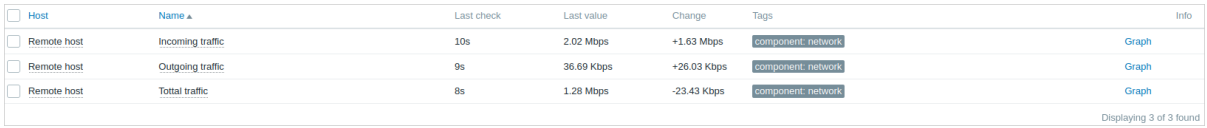
収集したデータの表示

これで、Zabbix によるネットワークトラフィックの監視が開始されています。

収集したメトリクスを表示するには、監視 > ホスト メニューセクションを開き、ホストの行にある 最新データをクリックします。

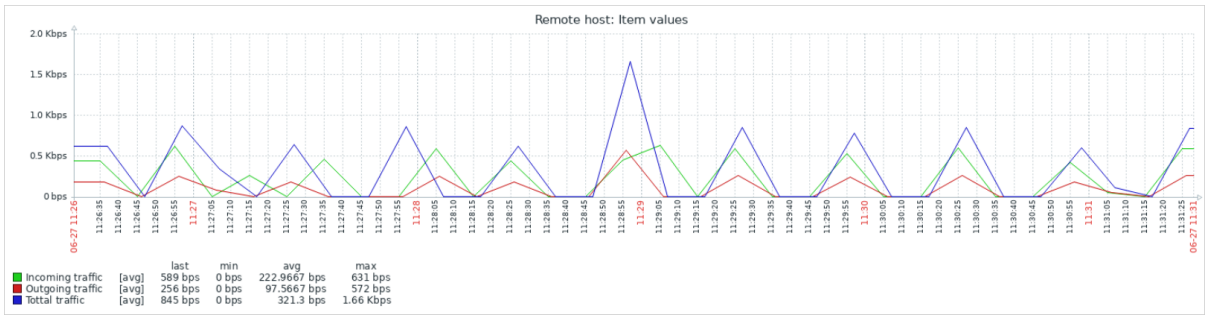


トラフィックデータが表示されます。



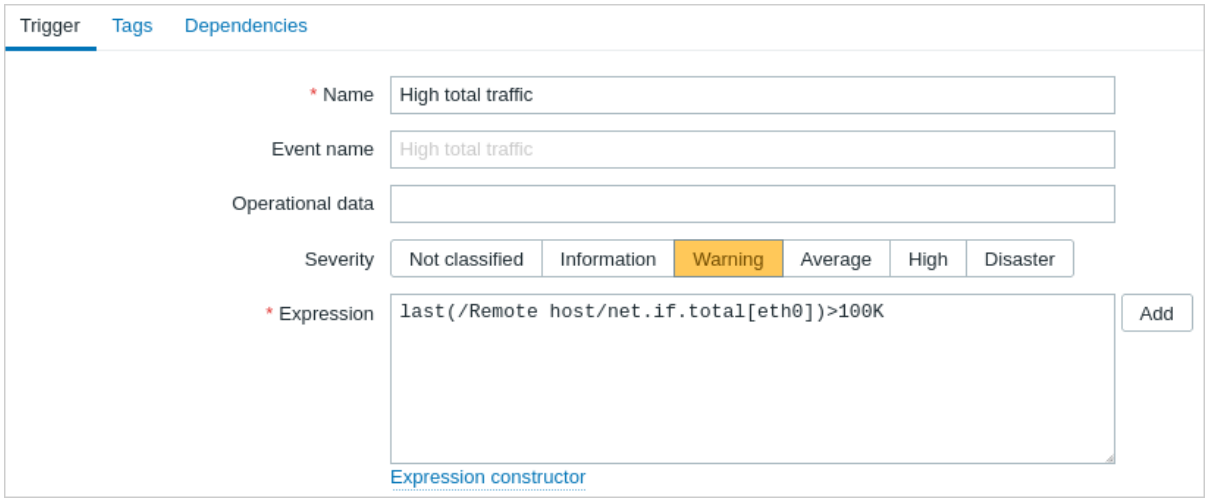
グラフの表示

収集されたデータはグラフとして表示できます。表示するには、最新データセクションでアイテムの行にある グラフをクリックするか、必要なアイテムを選択して下部の グラフを表示をクリックします。

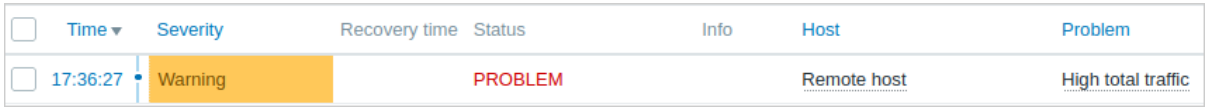


トリガーの設定

トリガーを設定して、異常なネットワークトラフィックを検出できます。トリガーの設定方法の手順に従い、たとえば総トラフィックが高すぎることを示すトリガーを追加します。



次に、トリガー式で設定したしきい値をトラフィックが超えるようにし、*Monitoring* > 障害に移動して、その障害が一覧に表示されていることを確認します。



Time ▾	Severity	Recovery time	Status	Info	Host	Problem
17:36:27	Warning		PROBLEM		Remote host	High total traffic

障害アラートの設定

障害に関する通知を受け取る方法はいくつかあります。
最も一般的なのはメールです。メールによる障害通知の設定手順に従ってください。
また、通知の配信に使用する他のメディアタイプを選択することもできます。

関連情報:

- 障害エスカレーション - 複数ステップのアラートシナリオを作成する方法
  (例: まずシステム管理者にメッセージを送信し、次に、45 分以内に障害が解決されない場合は、データセンター管理者にメッセージを送信する)。
- 障害の承認 - 障害が既知であることを示す方法、解決に関するコメントを追加する方法、障害を抑制またはクローズする方法。
- Zabbix エージェントで Linux を監視する - あらかじめ設定されたテンプレートをリンクして、最も重要なアイテムの基本監視を開始する方法。

アクティブチェックを使用してネットワークトラフィックを監視する

はじめに

このページでは、アクティブチェックを使用して Zabbix でネットワークトラフィックの基本監視を開始するために必要な手順を説明します。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、アクティブチェックを使用してネットワークトラフィックの基本監視を有効にするために必要な最小限の手順をまとめています。より詳細なカスタマイズオプションを探している場合や、さらに高度な設定が必要な場合は、Zabbix マニュアルのConfigurationセクションを参照してください。

前提条件

このガイドを進める前に、OS 向けの手順に従って Zabbix server、Zabbix frontend、Zabbix agent をダウンロードしてインストールする必要があります。
なお、トラフィック監視が必要なマシンには Zabbix agent をインストールしてください。
これは、Zabbix server がインストールされている同じホストでも、別のホストでもかまいません。

このガイドでは、*Remote host* という名前の別のマシン上で *eth0* インターフェースのネットワークトラフィック監視を設定する手順を説明します。

Zabbix の監視設定

Zabbix エージェントは、アクティブモードまたはパッシブモードでメトリクスを収集できます（同時に使用可能です）。詳細は、パッシブおよびアクティブなエージェントチェックを参照してください。このガイドでは、アクティブチェックによる監視について説明します。

Zabbix エージェントの設定

1. エージェントがインストールされているマシンで、エージェント設定ファイルを開きます。

Zabbix エージェントを使用している場合:

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

Zabbix エージェント 2 を使用している場合:

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（オプションでポートも指定可能）を、ServerActive パラメータに追加します。例:

```
ServerActive=192.0.2.1:10051
```

Zabbix エージェントは、このアドレスを使用して Zabbix サーバーのトラッパーポート（デフォルト: 10051）に接続し、アクティブチェックの設定データを要求します。

3. Hostname パラメータを定義します。この値は、Zabbix Web インターフェースで定義するホスト名と一致している必要があります。この例では、次のようになります。

```
Hostname=Remote host
```

アクティブチェックでは、Zabbix エージェントが Hostname の値を使用してサーバーから正しいホスト設定を取得するため、Hostname の値は一致している必要があります。具体的には、エージェントがサーバーへの接続を開始し、Hostname の値を使用して自身を識別します。その後、サーバーはそのホストの監視設定を提供します。これらの値が異なる場合、エージェントは適切な設定を受信できず、メトリクスが欠落したり、監視に問題が発生したりします。

4. Zabbix エージェントを再起動します。

Zabbix エージェントを使用している場合:

```
systemctl restart zabbix-agent
```

Zabbix エージェント 2 を使用している場合:

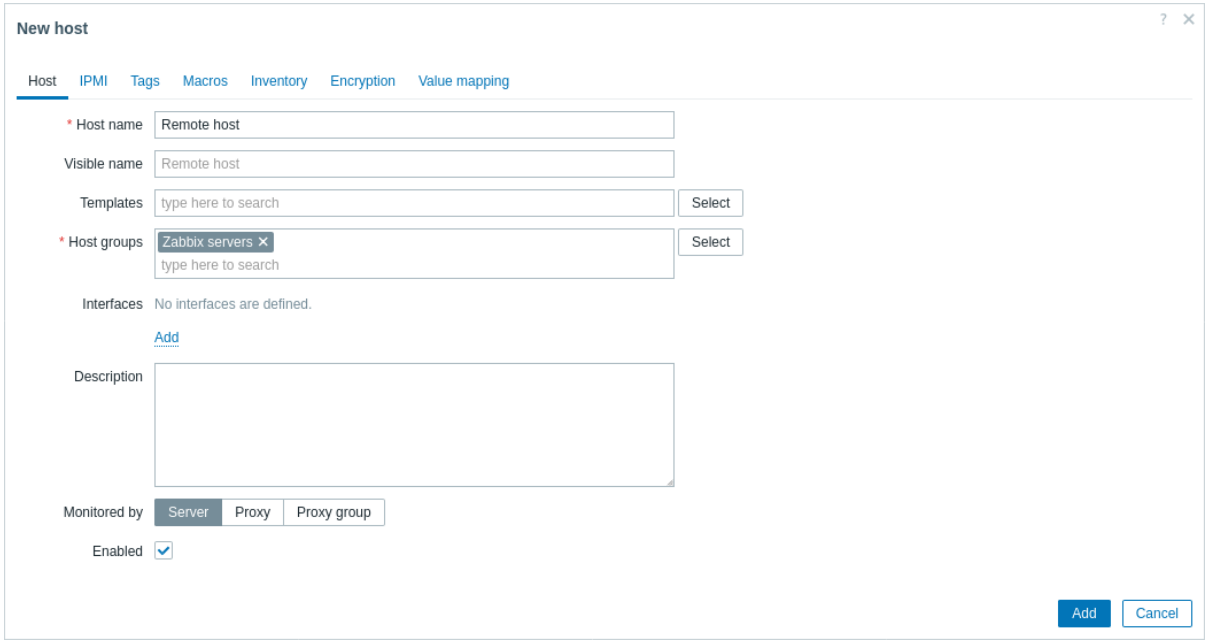
```
systemctl restart zabbix-agent2
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

- *Host name* フィールドに、以前エージェント設定ファイルで定義した Hostname パラメータの値と一致するホスト名（例: "Remote host"）を入力します。
- *Host groups* フィールドに、ホストグループ（例: "Zabbix servers"）を入力または選択します。
- *Interfaces* は未定義のままにします。アクティブチェックでは、エージェントがサーバーからの接続を待ち受けるのではなく、サーバーへ接続を開始するため、インターフェースは必要ありません。



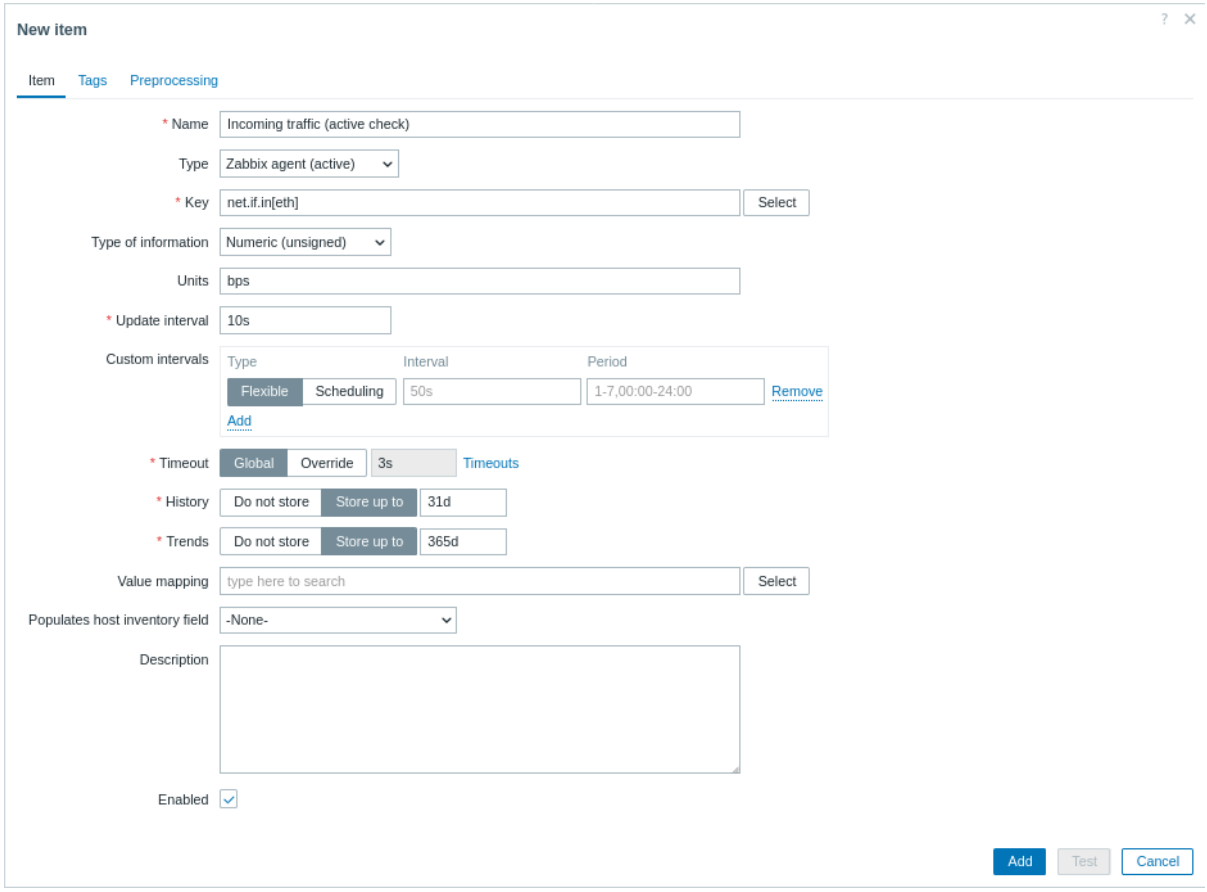
3. *Add* をクリックしてホストを追加します。このホストは監視対象の Linux マシンを表します。

アイテムの作成

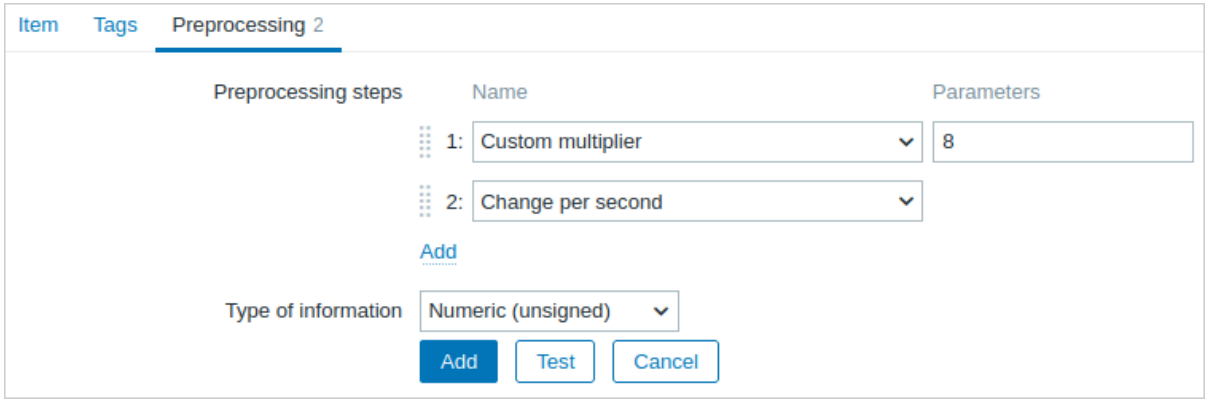
トラフィック監視用のアイテムを追加するには、アイテムの作成 の手順に従って、次のアイテムを作成します。

- 受信トラフィック
- 送信トラフィック
- 総トラフィック

アクティブチェックを使用した受信トラフィック監視アイテムの簡単な設定は、次のようになります。



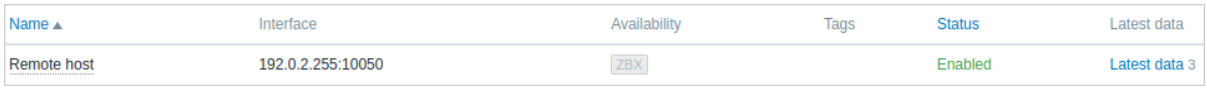
収集したデータを実用的に使えるようにするには、アイテムを作成する際にいくつかの前処理 ステップを設定できます。この場合は、8 倍する処理（バイトをビットに変換するため）と、1 秒あたりの変化量として表示する処理を設定できます。



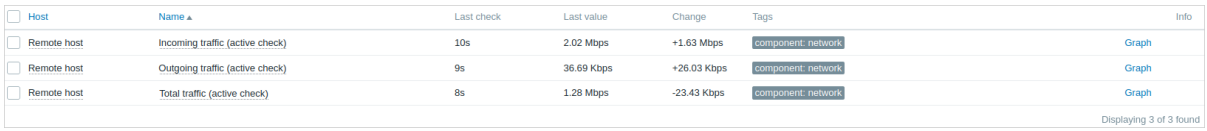
収集されたデータを表示する

おめでとうございます！この時点で、Zabbix はすでにネットワークトラフィックを監視しています。

収集されたメトリクスを表示するには、*Monitoring > Hosts* メニューセクションを開き、ホストの行にある *Latest data* をクリックします。



トラフィックデータが表示されます。



こちらも参照:

- グラフを表示する - 収集したデータをグラフとして表示する方法。
- トリガーを設定する - 異常なネットワークトラフィックを検知するためのトリガーを設定する方法。
- 障害アラートを設定する - 障害状況に関する通知を設定する方法。

- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法 (例: まずシステム管理者にメッセージを送信し、次に、45 分以内に障害が解決しない場合は、データセンター管理者にメッセージを送信する)。
- 障害の承認 - 障害が既知であることを示し、解決に関するコメントを付け、障害を抑制またはクローズする方法。
- Zabbix エージェントで Linux を監視する - あらかじめ設定されたテンプレートをリンクして、最も重要なアイテムの基本監視を開始する方法。
- パッケージからのインストール - さまざまな Linux ディストリビューション向けの公式 RPM および DEB パッケージを使用して Zabbix コンポーネントをインストールし、最新機能とバグ修正を利用できるようにする方法。

Browser アイテムで **Web** サイトを監視する

はじめに

このページでは、Browser アイテムを使用して Web サイトの基本的な監視を開始するために必要な手順を説明します。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、Browser アイテムを使用して Web サイトの基本的な監視を有効にするために必要な最小限の手順をまとめています。より詳細なカスタマイズオプションや、さらに高度な設定が必要な場合は、Zabbix マニュアルのBrowser アイテムページまたは設定セクションを参照してください。

前提条件

このガイドを進める前に、OS 向けの手順に従って ダウンロードしてインストール した Zabbix サーバーと Zabbix Web インターフェースが必要です。

環境によっては、このガイドの一部の手順が多少異なる場合があります。このガイドは Ubuntu を実行している環境を前提としています。

WebDriver の設定

ブラウザーアイテムには、ブラウザーを制御して操作し、ボタンのクリックやテキスト入力などのテストコマンドを実行する Web テスト用エンドポイントとして、オートメーションフレームワーク（Selenium Server または通常の WebDriver、たとえば ChromeDriver のいずれか）が必要です。
このガイドでは例として、Docker コンテナ内の Chrome を使用した Selenium Server を使います。

Docker はすでに設定済みであることを前提としています。このガイドでは Docker の設定については扱いません。
インストール手順については、Ubuntu に Docker Engine をインストールするを参照してください。

1. 次のオプションを指定して、Docker コンテナ内で Chrome を使用した Selenium Server を起動します。

- **docker run --name browser** - ”browser” という名前の新しい Docker コンテナを実行します。
- **-p 4444:4444** - ホストマシンのポート 4444 をコンテナのポート 4444 にマッピングします（これは Selenium Server がコマンドを受け付けるために使用するポートです）。
- **-p 7900:7900** - ホストマシンのポート 7900 をコンテナのポート 7900 にマッピングします（これは Virtual Network Computing (VNC) サーバーが使用するポートで、VNC クライアントが必要ですが、ブラウザーの GUI をリモートで表示できます）。
- **--shm-size=”2g”** - コンテナに 2GB の共有メモリを割り当てます（Chrome を正常に動作させるために重要です。クラッシュを避けるには、かなりの共有メモリが必要になる場合があります）。
- **-d** - コンテナをデタッチモードで実行します。つまり、バックグラウンドで動作します。
- **selenium/standalone-chrome:latest** - 使用する Docker イメージを指定します。この場合は、Chrome を使用した Selenium Server の最新バージョンです。

```
docker run --name browser \
-p 4444:4444 \
-p 7900:7900 \
--shm-size="2g" \
-d selenium/standalone-chrome:latest
```

2. browser Docker コンテナが実行中で、アクセス可能であることを確認します。

- コンテナの IP アドレスを取得します（この例では 192.0.2.1）。

```
ip addr

##### 1: lo: <LOOPBACK,UP,LOWER_UP>
#####     ...
##### 3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> ...
#####     inet 192.0.2.1/16 brd 192.0.255.255 scope global docker0
#####     ...
```

- Ncat を使用してコンテナへの接続をテストします。

```
nc -zv 192.0.2.1 4444

##### Connection to 192.0.2.1 4444 port [tcp/*] succeeded!
```

- curl を使用して Selenium Server から Web ページの内容を取得します。

```
curl -L 192.0.2.1:4444

##### <!DOCTYPE html>
##### <html lang="en">
#####
##### <head>
#####   <meta charset="utf-8"/>
#####   <link href="favicon.svg" rel="icon" type="image/svg">
#####   <meta content="width=device-width, initial-scale=1" name="viewport"/>
#####   <link href="logo192.png" rel="apple-touch-icon"/>
#####   <link href="manifest.json" rel="manifest"/>
#####   <title>Selenium Grid</title>
##### </head>
#####
##### <body>
##### ...
```

Note:
トラブルシューティングについては、Docker のドキュメントを参照してください。

Zabbix サーバーの設定

ブラウザーアイテムは、StartBrowserPollers サーバー設定パラメータを調整して有効にする必要がある、ブラウザーポーラー Zabbix プロセスによって実行および処理されます。また、WebDriverURL パラメータには、事前に設定した Web テストエンドポイントを指定する必要があります。

デフォルトでは、StartBrowserPollers パラメータは 1 に設定されているため、Web テストエンドポイントを指定するだけで済みます。

1. Zabbix サーバー設定ファイルを開きます。

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

2. Zabbix サーバー設定ファイルでWebDriverURLパラメータを見つけて設定します。

```
###### Option: WebDriverURL
#####WebDriver インターフェースの HTTP(S) URL。例: Selenium WebDriver スタンドアロンサーバーで使用する http://locali
#####
##### Mandatory: no
##### Default:
##### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Zabbix サーバーを再起動します。

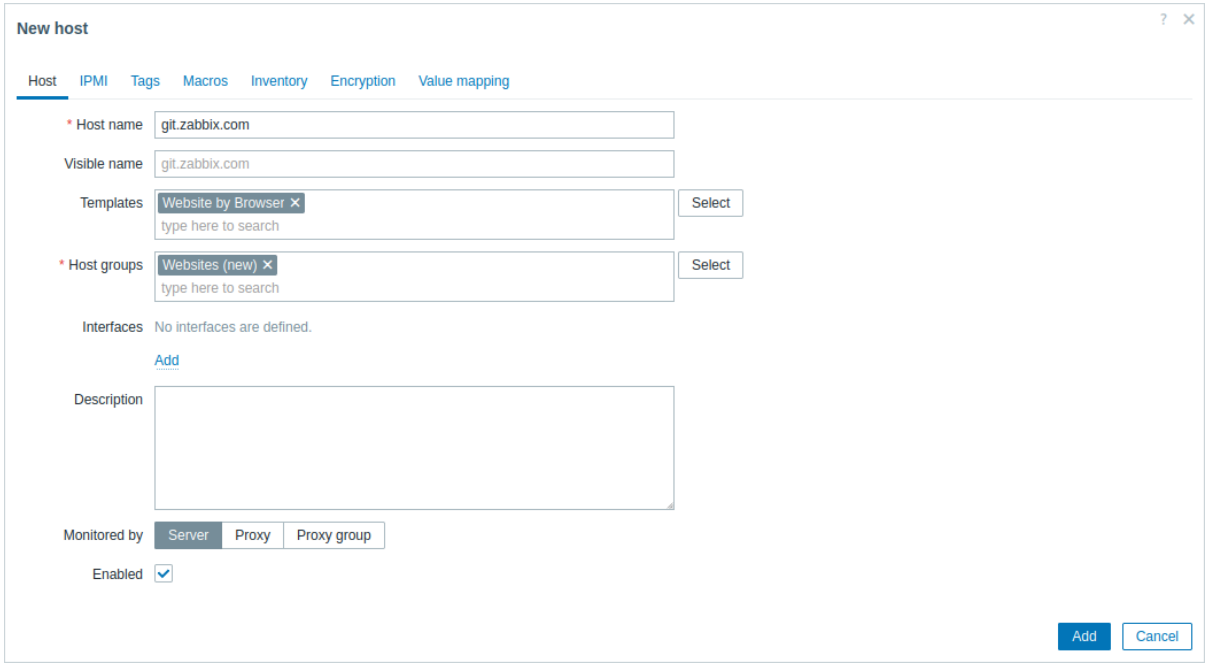
```
systemctl restart zabbix-server
```

Zabbix Web インターフェースの設定

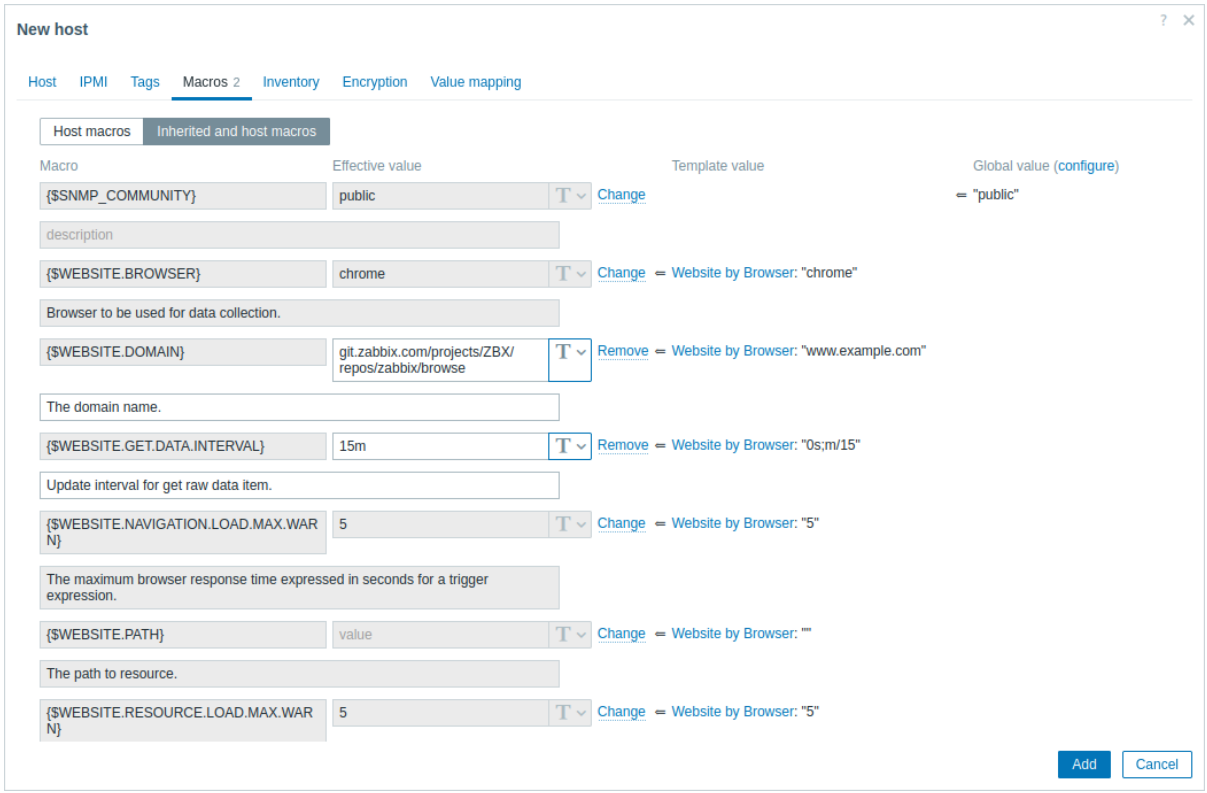
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

- ホスト名フィールドに、ホスト名を入力します（例: ”git.zabbix.com”）。
- テンプレートフィールドに、”Website by Browser” テンプレートを入力または選択します。このテンプレートの詳細については、Website by Browser を参照してください。
- ホストグループフィールドに、ホストグループを入力または選択します（例: 新しいホストグループ”Websites”）。



- マクロタブで、継承およびホストマクロに切り替え、次のマクロを探して、更新するマクロ値の横にある 変更をクリックします。
  - {$WEBSITE.DOMAIN} - ドメイン名（例: git.zabbix.com/projects/ZBX/repos/zabbix/browse）
  - {$WEBSITE.GET.DATA.INTERVAL} - アイテムデータの更新間隔（例: 15m）

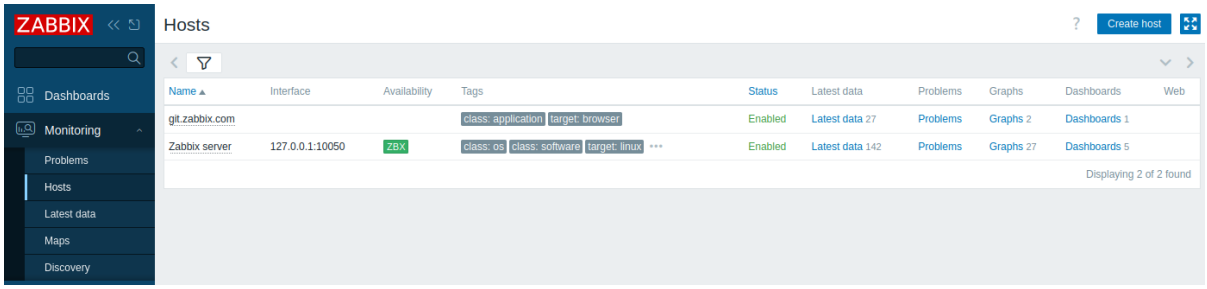


3. 追加ボタンをクリックしてホストを作成します。このホストは、監視したい Web サイトを表します。

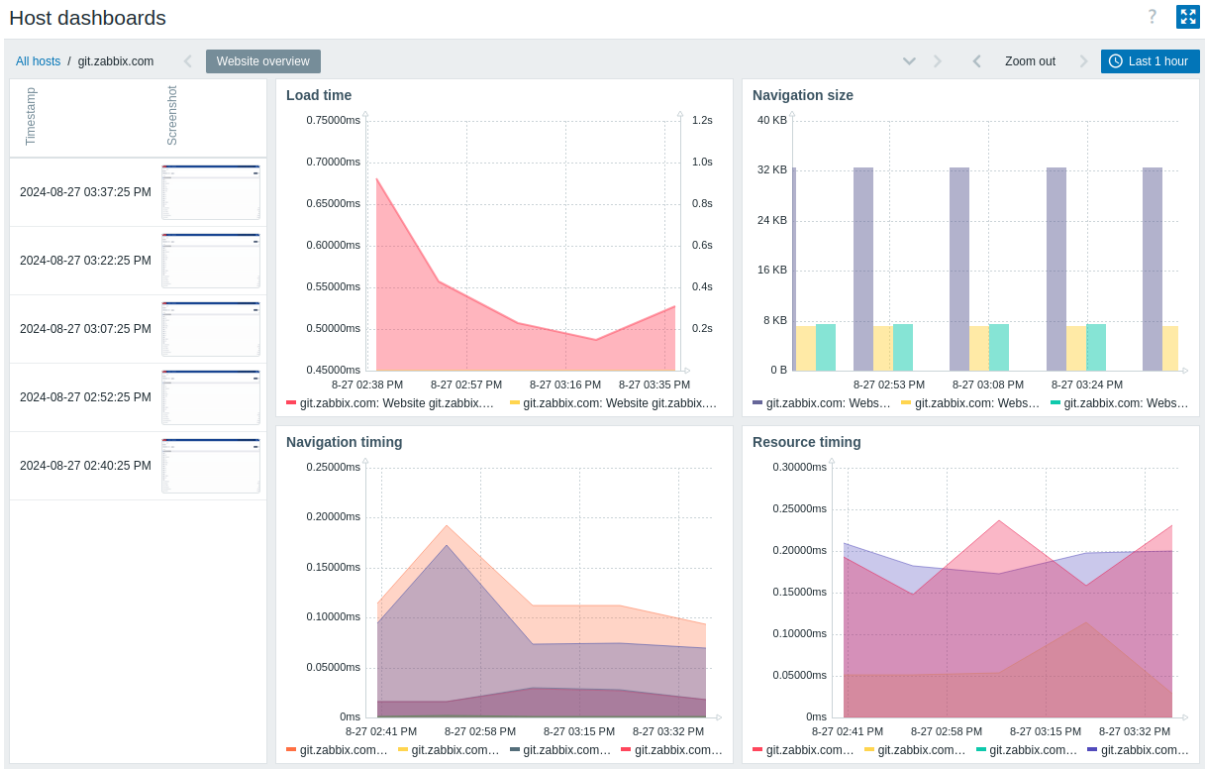
収集されたメトリクスを表示する

おめでとうございます。ここまでで、Zabbix はすでに指定した Web サイトを監視しています。

収集されたメトリクスを表示するには、*Monitoring > Hosts* メニューセクションに移動し、ホストの横にある *Dashboards* をクリックします。



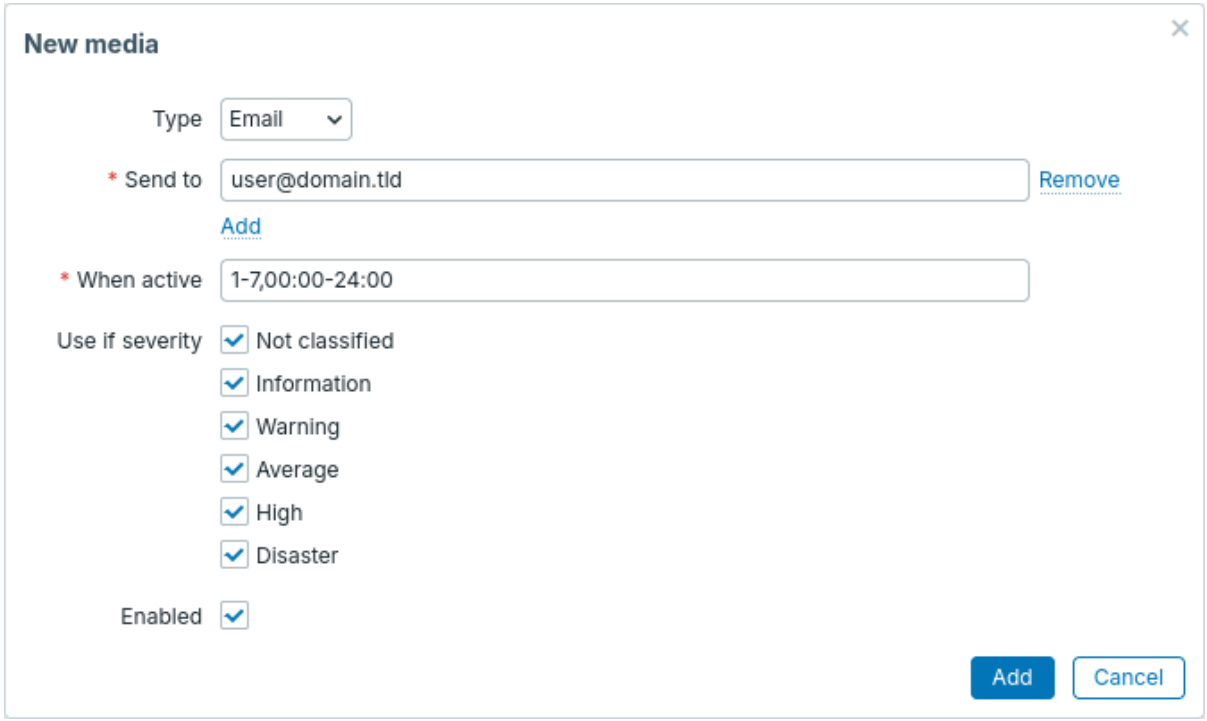
この操作により、Web サイトから収集された最も重要なメトリクスを含むホストダッシュボード（テンプレートレベルで設定済み）に移動します。



障害アラートの設定

Zabbix は、さまざまな方法を使用してインフラストラクチャの障害を通知できます。
このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロフィール に移動し、*Media* タブに切り替えて、メールアドレスを追加 します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出すると、メールでアラートを受け取るはずです。

設定をテストする

設定をテストするには、Zabbix Web インターフェースでホスト設定を更新し、実際の障害をシミュレートできます。

1. Zabbix で対象の Web サイトのホスト設定を開きます。

2. *Macros* タブに切り替え、*Inherited and host macros* を選択します。

3. たとえば、以前に設定した {$WEBSITE.DOMAIN} マクロ値の横にある *Change* をクリックし、誤ったドメイン名を設定します（例: /git.zabbix.com/projects/ZBX/repos/zabbix/browse）。

4. *Update* をクリックしてホスト設定を更新します。

5. しばらくすると、Zabbix は「Failed to get JSON of the requested website」という障害を検出します。これは、指定した Web サイトに接続できないためです。この障害は*Monitoring > Problems* に表示されます。



アラートが設定されている 場合は、障害通知も受信します。

6. マクロ値を元の値に戻して障害を解消し、Web サイトの監視を続行します。

関連項目

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 障害のエスカレーション - 複数ステップのアラートシナリオを作成する方法（例: まずシステム管理者にメッセージを送信し、次に障害が 45 分以内に解決されない場合はデータセンター管理者にメッセージを送信する）。
- Browser アイテム - Browser アイテムを設定する方法。
- テンプレート Website by Browser - *Website by Browser* テンプレートに関する追加情報。

Zabbix エージェント **2**（パッシブ）で **Web** サイトの証明書を監視する

はじめに

このガイドでは、Zabbix エージェント 2 の web.certificate.get キーを使用して SSL/TLS 証明書を設定および監視する方法を包括的に解説します。単一または複数の Web サイトの証明書監視を簡素化し、管理者が期限切れや無効な証明書などの潜在的な問題を迅速に特定できるようにすることを目的としています。

このガイドの対象者

このガイドは、Zabbix の新規ユーザー向けに作成されており、Web サイト証明書の基本的な監視を有効にするために必要な最小限の手順をまとめています。より詳細なカスタマイズオプションを探している場合や、より高度な設定が必要な場合は、Zabbix マニュアルのConfiguration セクションを参照してください。

前提条件

このガイドを進める前に、OS 向けの手順に従って ダウンロードしてインストール する必要があります。Zabbix サーバー、Zabbix Web インターフェース、および Zabbix エージェント 2 をインストールしてください。このチュートリアルでは、Zabbix サーバーとエージェントの両方が同じマシンにインストールされていることを前提としているため、設定では 127.0.0.1 を使用しています。

Zabbix エージェント 2 の設定

1. Zabbix エージェント設定ファイルを開きます。

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

2. エージェントとサーバーが同じマシン上で実行されているため、Server パラメータを 127.0.0.1 に設定します。

```
Server=127.0.0.1
```

3. ファイルを保存し、Zabbix エージェント 2 サービスを再起動します。

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

4. Zabbix エージェント 2 のセットアップと設定が完了したら、次のコマンドで利用可能かどうかをテストします。

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

Zabbix エージェント 2 には WebCertificate プラグインがデフォルトで含まれているため、個別のインストールや設定は必要ありません。

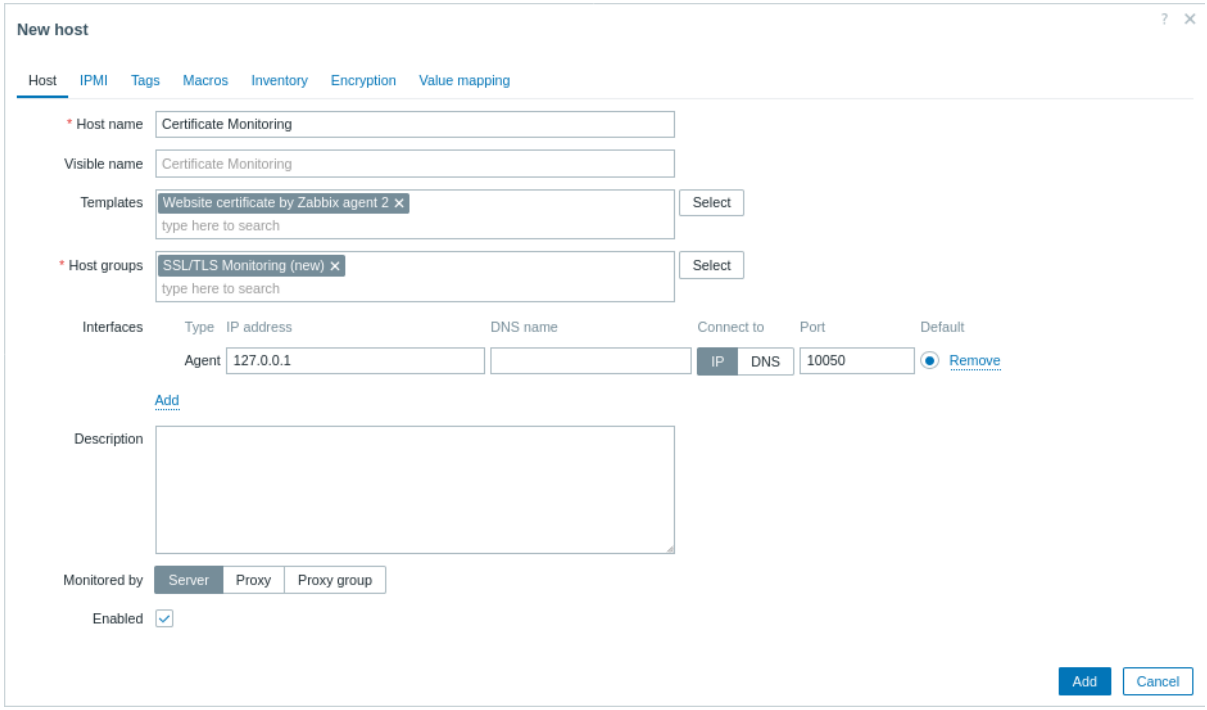
Zabbix Web インターフェースの設定

1. Zabbix Web インターフェースにログインします。

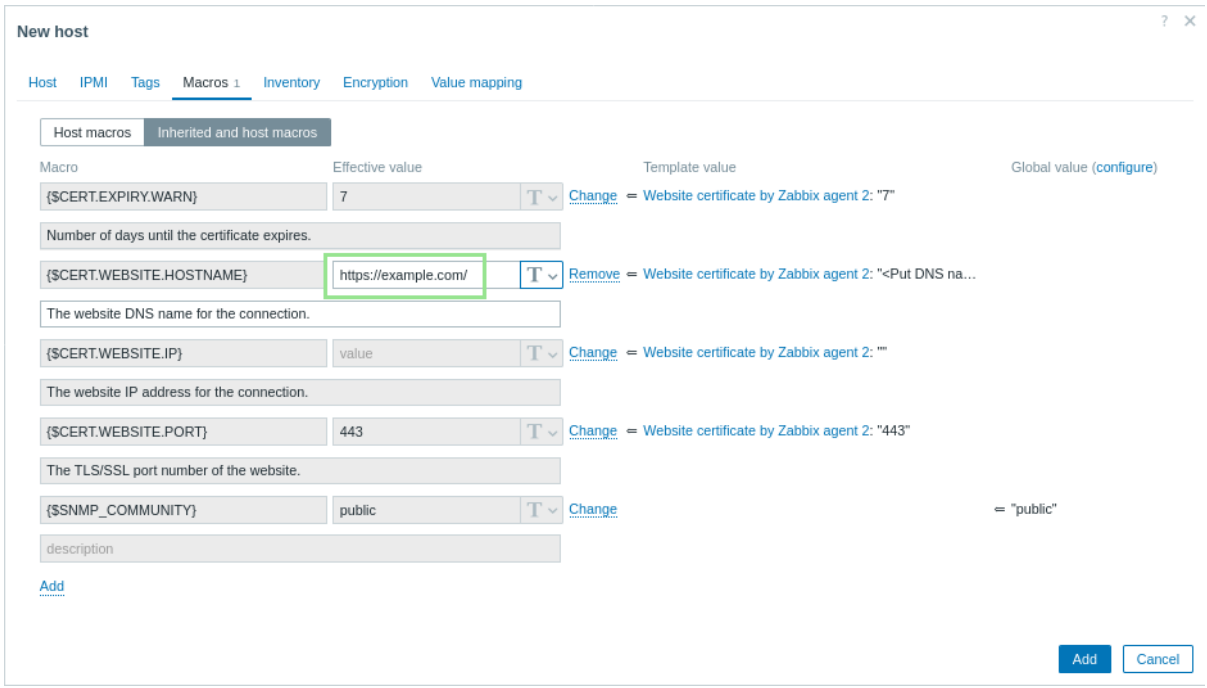
2. *Monitoring* > *Hosts* に移動します。

3. Web サイト証明書を監視したい既存のホストをクリックするか、必要に応じてホストを作成 します。

- *Host name* フィールドに、ホスト名を入力します（例: ”Certificate Monitoring”）。
- *Templates* フィールドで、ホストにリンク するテンプレート ”Website certificate by Zabbix agent 2” を入力または選択します。
- *Host groups* フィールドで、ホストグループを入力または選択します（例: ”SSL/TLS Monitoring”）。
- *Interfaces* フィールドで、”Agent” タイプのインターフェースを追加し、IP アドレスを指定します。この例では”127.0.0.1” を使用します。



- *Macros* タブで *Inherited and host macros* に切り替え、次のマクロを探して、値の横にある *Change* をクリックして更新します。
  - `{$CERT.WEBSITE.HOSTNAME}` - 値として、監視対象の Web サイトの DNS 名を入力します。



4. *Add* をクリックしてホストを追加します。

複数の Web サイトを監視するには、`{$CERT.WEBSITE.HOSTNAME}` マクロに、それらのホスト名をカンマ区切りで指定できます。必要に応じて、`{$CERT.WEBSITE.PORT}` マクロにポート番号のカンマ区切りリストも指定できます。この場合、各ポートは順番に対応するホスト名に対応します。たとえば、example.com と example.org をポート 8443 で監視するには、次のように指定します。

- `{$CERT.WEBSITE.HOSTNAME}`: example.com,example.org
- `{$CERT.WEBSITE.PORT}`: 443,8443

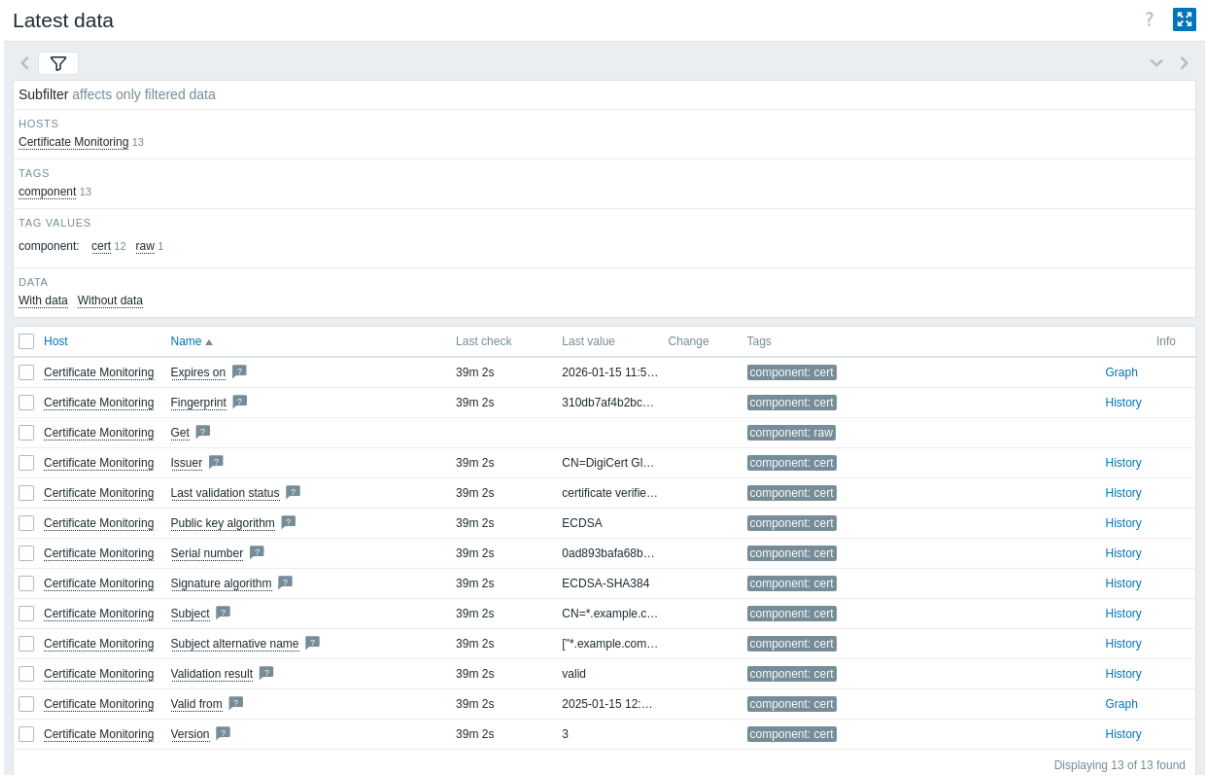
`{$CERT.WEBSITE.HOSTNAME}` マクロで指定した各 Web サイトごとに、Zabbix は対応するアイテムとトリガーのセットを作成します。これにより、各 Web サイトの SSL 証明書を個別に監視し、障害通知を行えます。

Host	Name ▲	Last check	Last value
Certificate Monitoring	Get data	11m 16s	{"error":{"message…
Certificate Monitoring	[example.com]: Expires on	11m 6s	2026-01-15 11:59:…
Certificate Monitoring	[example.com]: Fingerprint	11m 6s	310db7af4b2bc90…
Certificate Monitoring	[example.com]: Get		
Certificate Monitoring	[example.com]: Issuer	11m 6s	CN=DigiCert Glob…
Certificate Monitoring	[example.com]: Last validation status	11m 6s	certificate verified …
Certificate Monitoring	[example.com]: Public key algorithm	11m 6s	ECDSA
Certificate Monitoring	[example.com]: Serial number	11m 6s	0ad893bafa68b0b…
Certificate Monitoring	[example.com]: Signature algorithm	11m 6s	ECDSA-SHA384
Certificate Monitoring	[example.com]: Subject	11m 6s	CN=*.example.co…
Certificate Monitoring	[example.com]: Subject alternative name	11m 6s	["*.example.com","…
Certificate Monitoring	[example.com]: Validation result	11m 6s	valid
Certificate Monitoring	[example.com]: Valid from	11m 6s	2025-01-15 12:00:…
Certificate Monitoring	[example.com]: Version	11m 6s	3
Certificate Monitoring	[example.org]: Expires on	10m 52s	2026-01-15 11:59:…
Certificate Monitoring	[example.org]: Fingerprint	10m 52s	3b451cfce915637…
Certificate Monitoring	[example.org]: Get		
Certificate Monitoring	[example.org]: Issuer	10m 52s	CN=DigiCert Glob…
Certificate Monitoring	[example.org]: Last validation status	10m 52s	certificate verified …
Certificate Monitoring	[example.org]: Public key algorithm	10m 52s	ECDSA
Certificate Monitoring	[example.org]: Serial number	10m 52s	0722a749b558476…
Certificate Monitoring	[example.org]: Signature algorithm	10m 52s	ECDSA-SHA384
Certificate Monitoring	[example.org]: Subject	10m 52s	CN=*.example.org…
Certificate Monitoring	[example.org]: Subject alternative name	10m 52s	["*.example.org","e…
Certificate Monitoring	[example.org]: Validation result	10m 52s	valid
Certificate Monitoring	[example.org]: Valid from	10m 52s	2025-01-15 12:00:…

収集されたメトリクスを表示する

おめでとうございます。ここまでで、Zabbix は目的の Web 証明書をすでに監視しています。

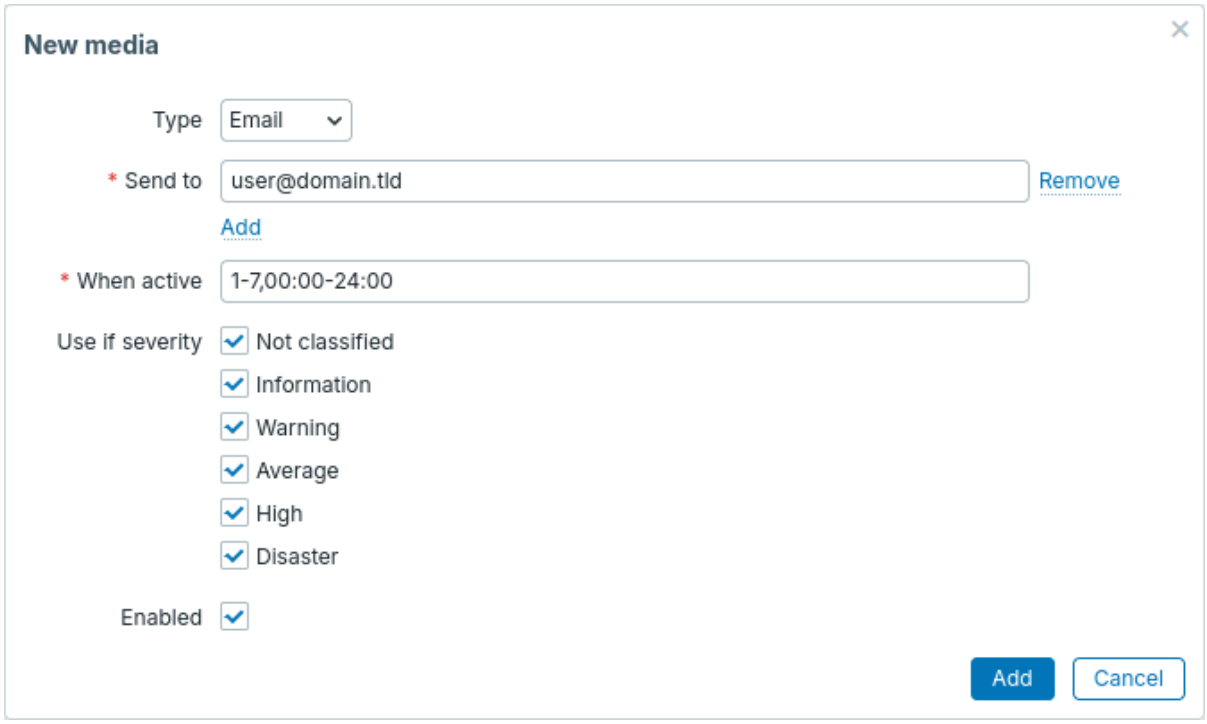
収集されたメトリクスを表示するには、*Monitoring > Hosts* メニューセクションに移動し、ホストの横にある*Latest data* をクリックして、失効日、発行者、サブジェクトなど、収集された最新のメトリクスを一覧で表示します。



障害アラートの設定

Zabbix はインフラストラクチャの問題を通知できます。このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロファイル に移動し、メディアタブに切り替えて、メールアドレスを追加します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出すると、メールでアラートを受け取るはずです。

設定をテストする

設定をテストするには、Zabbix Web インターフェースでホスト設定を更新し、実際の障害をシミュレートできます。

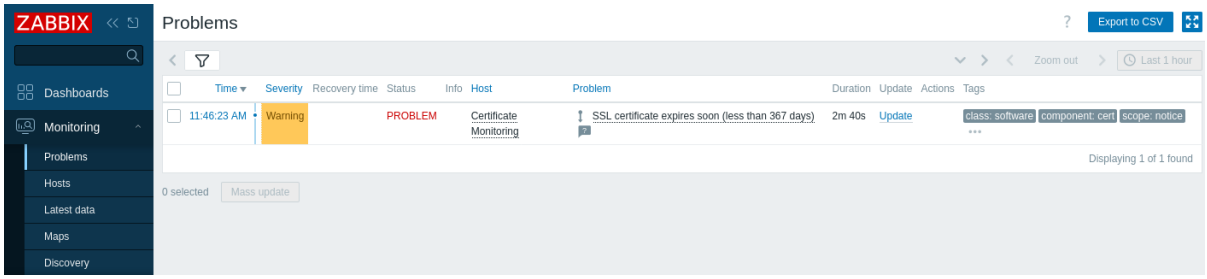
1. Zabbix で「Certificate Monitoring」ホストの設定を開きます。

2. *Macros* タブに切り替え、*Inherited and host macros* を選択します。

3. 以前に設定した {$CERT.EXPIRY.WARN} マクロ値の横にある *Change* をクリックし、証明書の有効期限が切れる前に警告を受け取るため、非常に大きい日数を設定します（365 日を超える値で十分です）。

4. *Update* をクリックしてホスト設定を更新します。

5. しばらくすると、Zabbix は有効期限までの日数とともに「SSL certificate expires soon」という障害を検出します。障害は*Monitoring > Problems* に表示されます。



アラートが設定されている 場合は、障害通知も受信します。

6. 障害を解消して証明書の値の監視を続けるため、マクロ値を元の値に戻します。

関連項目

- Zabbix エージェント 2 - アイテムキーの一覧です。
- テンプレート *Website certificate by Zabbix agent 2* - *Website certificate by Zabbix agent 2* テンプレートに関する追加情報です。
- テンプレート *Website certificate by Zabbix agent 2 active* - *Website certificate by Zabbix agent 2 active* テンプレートに関する追加情報です。
- Browser アイテムで Web サイトを監視する - Browser アイテムを使用して Web サイトの基本監視を開始する方法です。

Zabbix でネットワークスイッチまたはルーターを監視する

はじめに

このガイドでは、Zabbix を使用してネットワークスイッチまたはルーターの基本監視を開始するために必要な手順を説明します。
例として Cisco ルーターを使用しますが、この手順は SNMP 対応のあらゆるネットワークデバイスに適用できます。

このガイドの対象者

このガイドは、ネットワークデバイスの基本監視をすばやく有効化したい新規 Zabbix ユーザーおよびネットワーク管理者向けに作成されています。
詳細なカスタマイズや高度な設定オプションが必要な場合は、Zabbix マニュアルのSNMP agent ページまたはConfiguration セクションを参照してください。

前提条件

このガイドを進める前に、以下が準備されていることを確認してください。

- Zabbix server と Zabbix frontend がインストールされていること: お使いの OS の手順に従ってインストールしてください (パッケージからのインストール およびWeb interface のインストール を参照)。
- ローカルネットワークのメトリクスを監視する場合は Zabbix agent がインストールされていること。
- SNMP 対応デバイス: SNMP が有効なネットワークスイッチまたはルーター (たとえば Cisco ルーター)。
- MIB files がインストールされていること: MIB files をインストールすると、Zabbix は数値の OID を人間が読める名前や説明に変換できるようになります。
  適切な MIB サポートがない場合、数値のみが表示され、アイテムの設定や問題のトラブルシューティングが難しくなることがあります。

Ubuntu に MIB files をインストールするには:

1. MIB ダウンローダーパッケージをインストールします。

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

ベンダー固有の MIBs (例: Cisco、Juniper) を追加する必要がある場合は、適切な MIB ディレクトリに配置してください。

- Linux ベースのシステムでは、一般的な配置先は `/usr/share/snmp/mibs/` または `/usr/local/share/snmp/mibs/` です。
- Zabbix のインストール環境では、MIB files は `/var/lib/zabbix/mibs/` に保存できます。

MIBDIRS 環境変数または snmp.conf ファイルに正しいパスが含まれていることを確認してください。

システムが新しい MIBs を認識していることを確認するには、次を使用します。

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

詳細な手順については、SNMP ライブラリのドキュメントを参照してください。

- Cisco MIBs
- Juniper MIBs

2. /etc/snmp/snmp.conf を編集し、mibs : で始まる行をコメントアウトして、システムが利用可能なすべての MIBs を読み込めるようにします。

3. snmpwalk を実行して確認します (たとえば、snmpwalk -v 2c -c <your_community_string> <device_IP>)。OID が説明的な名前で表示されることを確認してください。

環境によっては、このガイドの一部の手順が多少異なる場合があります。
このガイドは、Ubuntu 環境で Cisco Catalyst 3750V2-24FS ネットワークデバイスを監視している構成を前提としています。

ネットワークデバイスはすでに物理的に設置され、接続済みであることを前提としています。

ネットワークデバイスを設定する（Cisco ルーターの例）

SNMP による監視を行うには、ネットワークデバイスで SNMP クエリを許可するように設定する必要があります。以下の例は SNMPv2 向けであり、既存の設定は考慮していません。注意: これらのコマンドを適用すると、現在の SNMP 設定が上書きされる場合があります。

Cisco ルーターの場合、通常は以下に示す手順で設定します。

SNMPv2 の例

1. SNMP を有効にし、community string を設定します。

Cisco ルーターのコンソールにログインし、設定モードに入ります:

```
configure terminal
```

次に、読み取り専用の community string を指定してSNMP を有効化します。例:

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

<your_community_string> を、セキュアな community string に置き換えてください。注: RO (Read-Only) オプションを使用すると、SNMP はデバイスからデータを取得できますが、設定変更はできません。

セキュリティ上の理由から、SNMP アクセスは必要なデバイスのみに制限することを推奨します。アクセス制御リスト (ACL) の設定方法については、Cisco の公式ドキュメントを参照してください。

2. 設定を保存します。

再起動後も SNMP 設定が保持されるよう、変更を保存します:

```
write memory
```

SNMPv3 の例

SNMPv3 は、認証と暗号化によって強化されたセキュリティを提供します。その設定は SNMPv2 よりも安全であり、デバイス固有のドキュメントと照らし合わせて確認する必要があります。

1. SNMP グループを作成します。

プライバシー（暗号化）を有効にした SNMPv3 グループを設定します。

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

2. SNMP ユーザーを作成します。

認証とプライバシーを有効にした SNMPv3 ユーザーを追加します。プレースホルダーは、必要な値に置き換えてください。

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

3. 設定を保存します。

```
write memory
```

詳細や機種固有の手順については、外部の Cisco SNMP configuration tutorials を参照してください。ただし、このガイドでは SNMP 監視を有効にするための基本手順を説明しています。

Zabbix Web インターフェースの設定

Zabbix Web インターフェースでホストを作成する

1. Zabbix Web インターフェースにログインします。

2. 新しいホストを追加します。

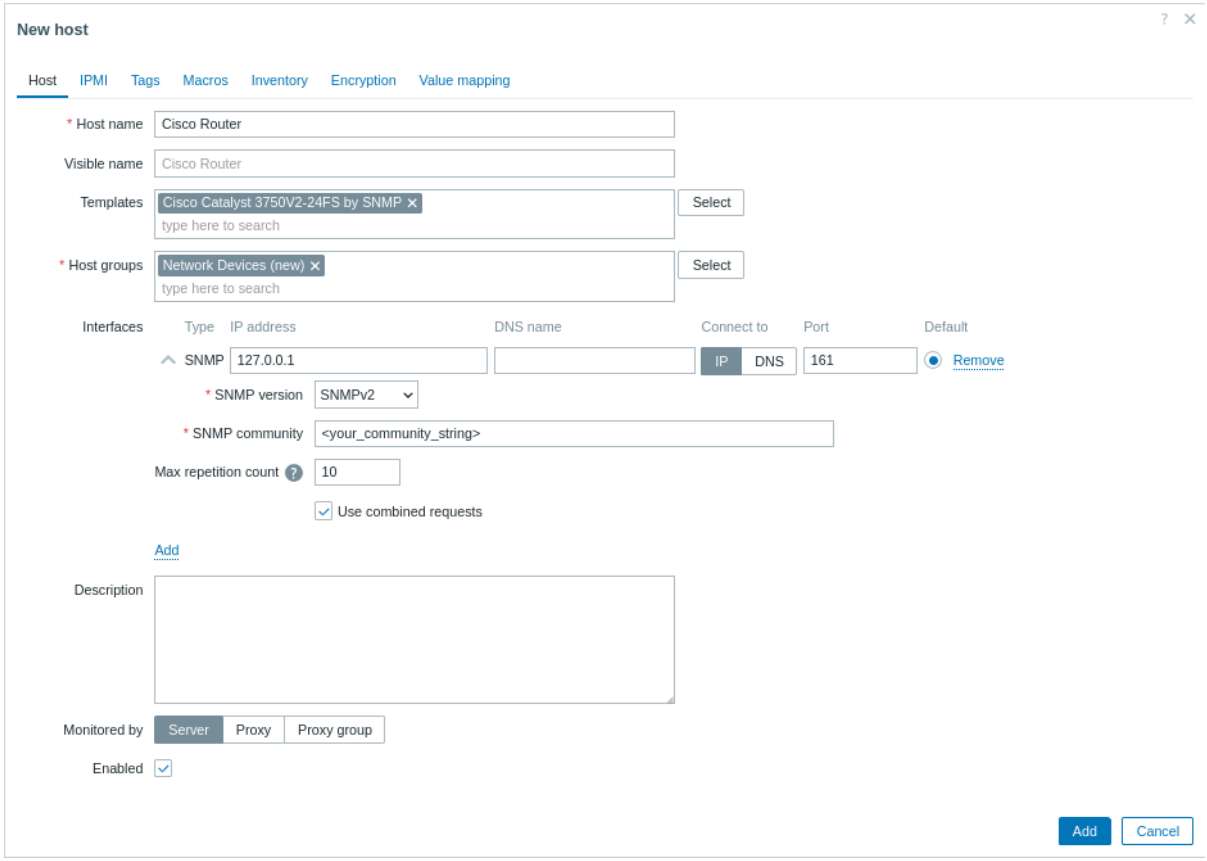
データ収集 > ホストに移動し、ホストの作成をクリックします。

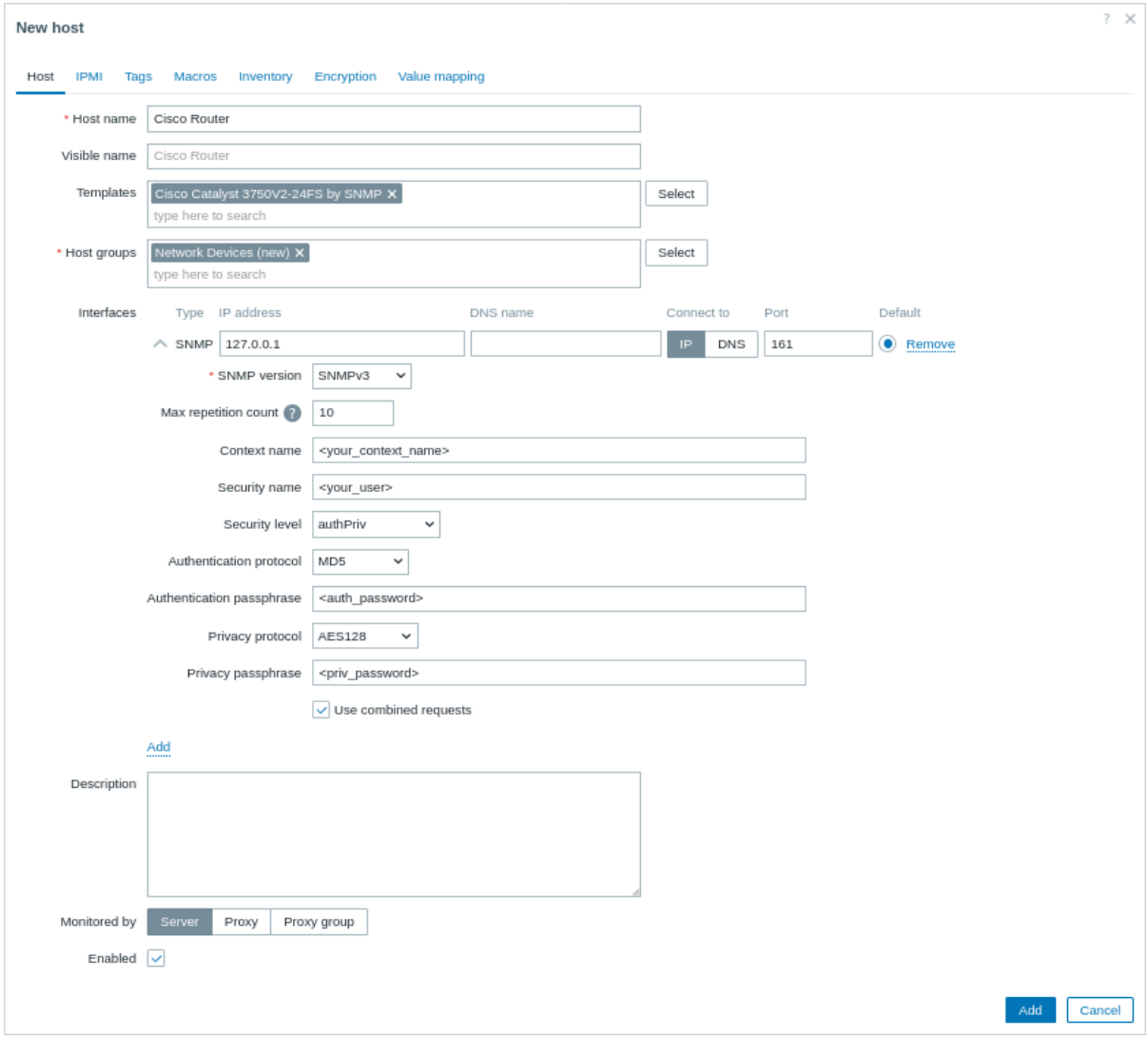
- ホスト名: デバイスの名前を入力します（例: ”Cisco Router”）。
- ホストグループ: 既存のグループを選択するか、”Network Devices” のような新しいグループを作成します。
- インターフェース:
  - インターフェースの下にある 追加をクリックします。
  - インターフェースタイプとして *SNMP* を選択します。
  - Cisco ルーターの IP アドレスまたは DNS 名を入力します。
  - デフォルトの SNMP ポートを設定します（通常は 161）。
  - ドロップダウンメニューを使用して、適切な SNMP バージョンを選択します（例: SNMPv2）。
  - SNMPv1/v2 の場合は、*SNMP community* フィールドに community string を入力します。SNMPv3 の場合は、追加の認証情報（*Context name*、*Security name*、*Security level* など）の入力が求められます。

3. テンプレートをリンクする

テンプレートフィールドで、デバイスに最も適した SNMP テンプレートを選択します。Zabbix には、多くのデバイスファミリー向けにあらかじめ用意されたSNMP templates が多数あります。たとえば、Cisco デバイスを監視する場合は、デバイスの OS またはモデルに対応するテンプレート（Cisco IOS SNMP や Cisco Catalyst 3750<device model> SNMP など）を選択します。

4. 追加をクリックしてホストを保存します。



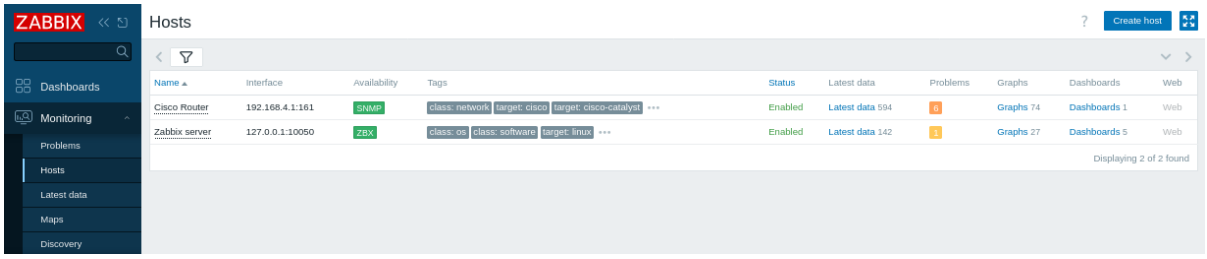


収集されたメトリクスを表示する

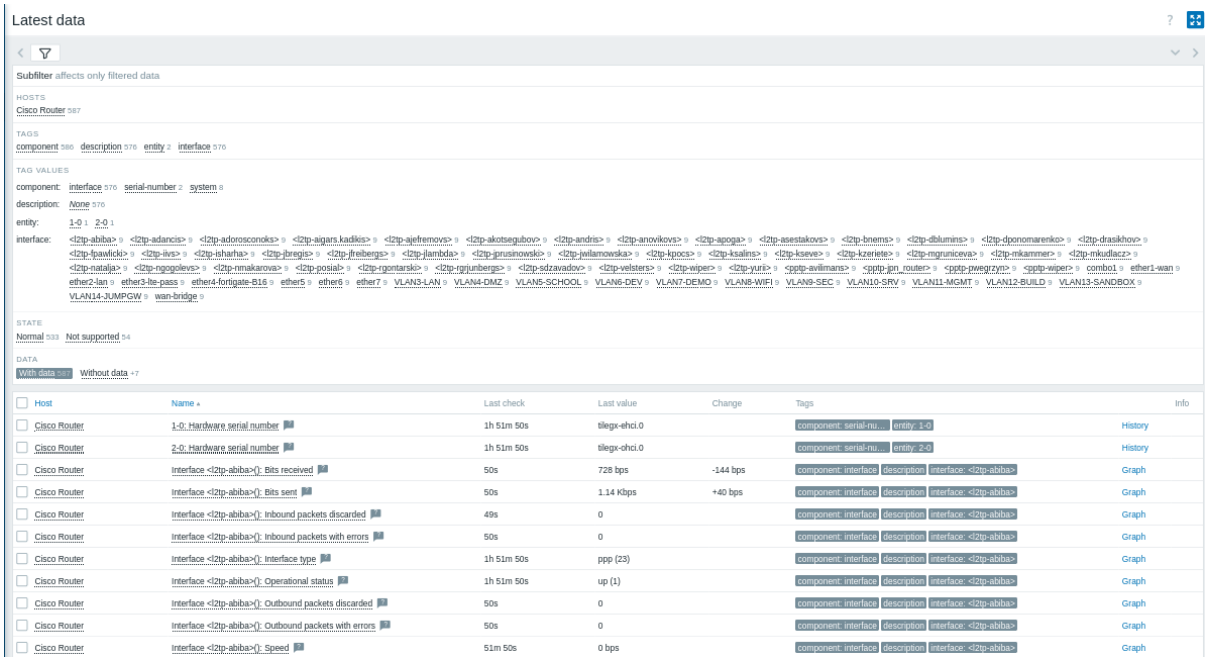
おめでとうございます! Zabbix の設定が完了し、ネットワークデバイスの監視を開始できるようになりました。

最新データ:

- Zabbix Web インターフェースで、Monitoring > Latest data に移動します。



- ”Cisco Router” ホスト（または検出されたホスト）を選択して、ハードウェアおよびネットワークの稼働時間、ICMP ロス、ping、応答時間などのメトリクスを表示します。



- グラフと画面:

パフォーマンスデータを可視化するには、SNMP アイテムの横にある *Graphs* をクリックして詳細なメトリクスを表示します。

次の手順として、以下を行えます:

- カスタム SNMP アイテムを追加する ことで、追加のメトリクスを監視します。
- 障害アラートを設定する ことで、潜在的な問題に関する通知を受け取ります。

SNMP アイテムの作成

ホストの設定が完了したら、特定のメトリックを監視するためのアイテムを作成できます。
注: テンプレートを使用している場合、この手順は省略できます。テンプレートには既定のアイテムセットがすでに含まれているためです。

1. SNMP OID を特定します:

`snmpwalk` コマンドを使用して、デバイスで利用可能な OID の一覧を取得します。
例:

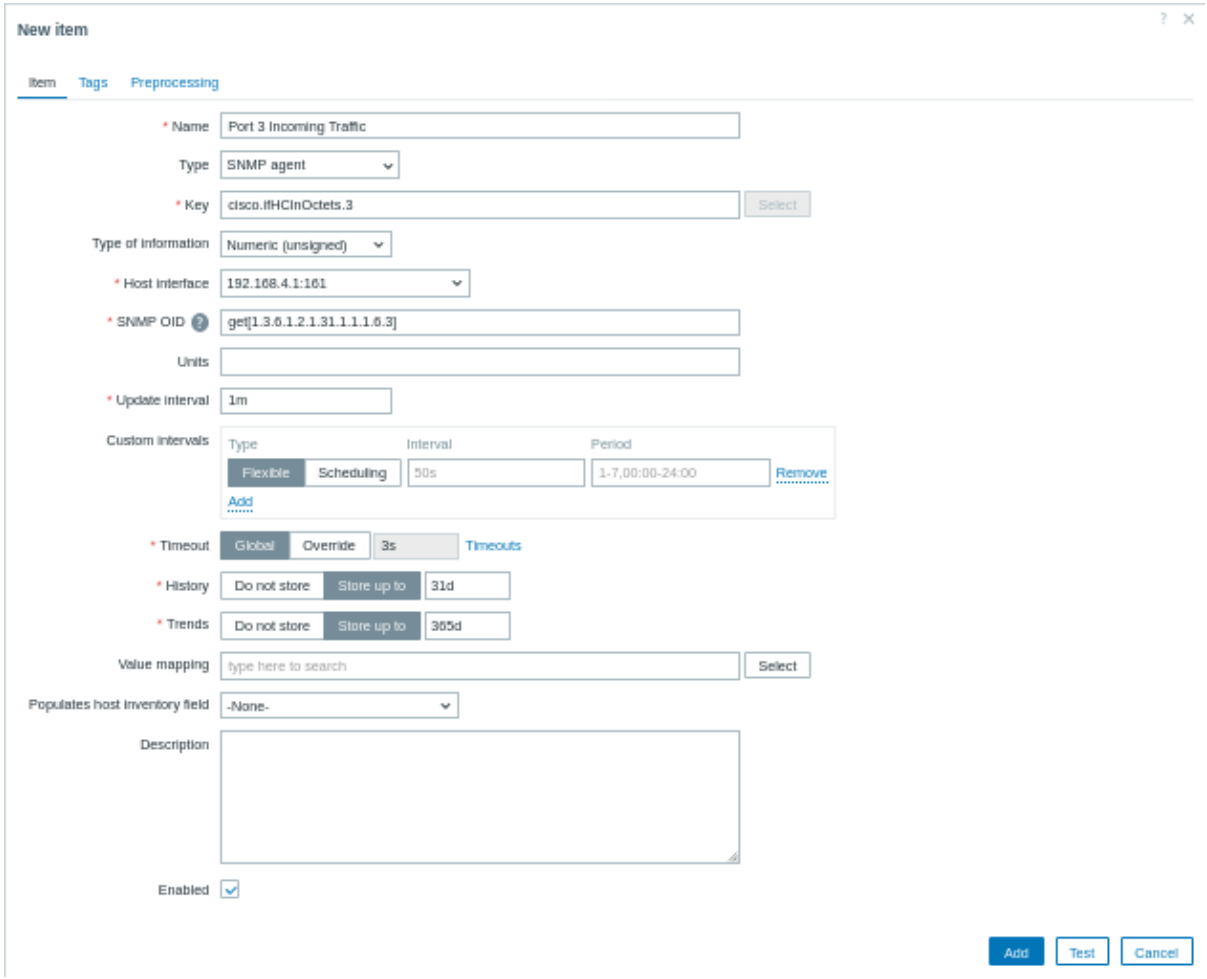
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

監視したいメトリックの OID を見つけます（たとえば、ポート 3 の受信トラフィックであれば IF-MIB::ifHCInOctets.3）。
数値 OID を取得するには、次を使用できます:

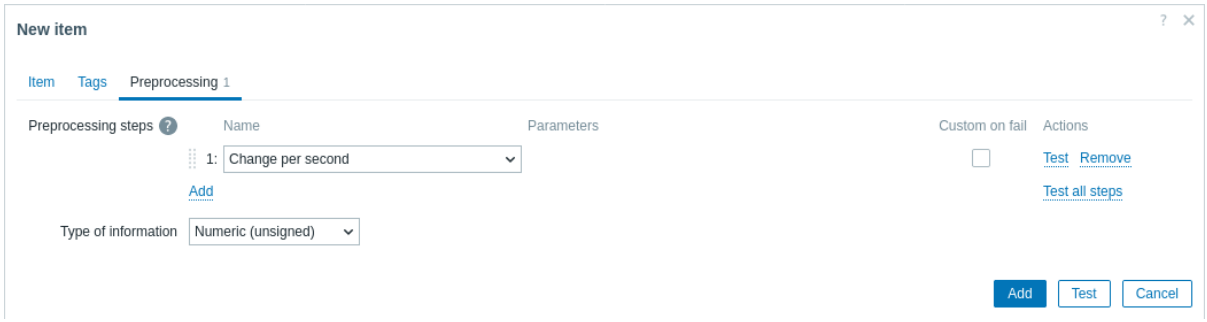
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. SNMP アイテムを作成します:

- データ収集 > ホストに移動し、SNMP ホストの アイテムタブをクリックして、アイテムの作成をクリックします。
- 名前: 説明的な名前を入力します（例: "Port 3 Incoming Traffic"）。
- タイプ: *SNMP agent* を選択します。
- キー: 意味のあるキーを指定します（例: `cisco.ifHCInOctets.3`）。
- ホストインターフェース: SNMP インターフェースが選択されていることを確認します。
- *SNMP OID*: サポートされている形式のいずれかを使用して OID を入力します。例:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` は単一の値用;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` は値のサブツリーを非同期で取得する場合。



- 前処理（必要な場合）：アイテムが累積カウンター（インターフェーストラフィックなど）を返す場合は、前処理タブに移動し、レートを計算するために”Change per second” のような前処理ステップを追加します。



1 回の SNMP トランザクションで複数の値を取得するには、`walk[OID1,OID2,...]` の構文を使用して複数の OID を指定できます。

数値 OID と MIB 名の間で OID を変換する

SNMP を扱う際には、数値 OID とそれに対応する MIB 名の間で変換する必要がある場合があります。
この変換により、メトリクスの特定やトラブルシューティングがより簡単になります。

- MIB 名を数値 OID に変換するには、`snmptranslate` コマンドに `-On` オプションを指定します。
  たとえば、MIB 名 `IF-MIB::ifHCInOctets.3` を数値 OID に変換するには、次を実行します:

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

このコマンドの出力例は次のとおりです:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- 数値 OID を MIB 名に変換するには、`snmptranslate` コマンドに `-IR`（または `-m ALL`）オプションを指定して逆変換します。
  たとえば、数値 OID `.1.3.6.1.2.1.31.1.1.1.6.3` を MIB 名に戻すには、次を実行します:

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

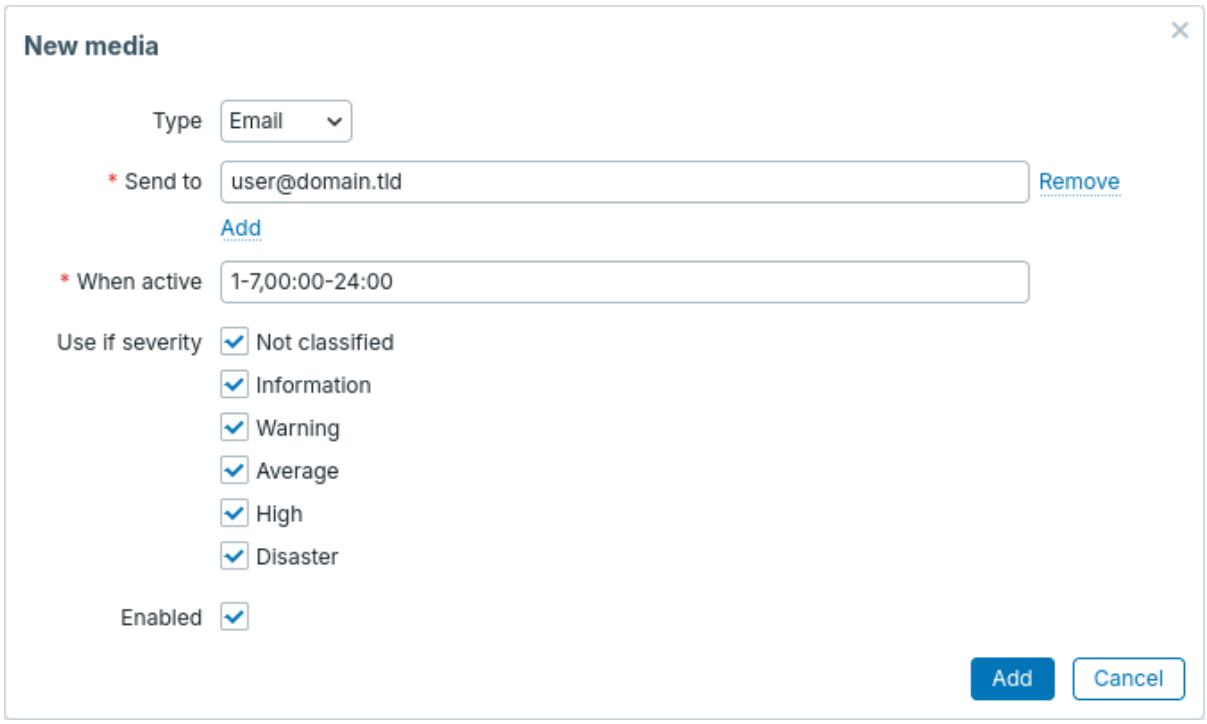
このコマンドの出力例は次のとおりです:

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

障害アラートの設定

このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. ユーザー設定 > プロファイル に移動し、*Media* タブに切り替えて、メールアドレスを追加します。



2. 障害通知の受信 のガイドに従います。

次回 Zabbix が障害を検出したときに、メールでアラートを受信できるはずです。

設定をテストする

Zabbix がネットワークパフォーマンスの問題を正しく検出できるようにするには、ICMP ping の応答時間しきい値を上げて、実際の障害をシミュレートします。

1. Zabbix で”Cisco Router” ホストの設定を開きます。

2. *Macros* タブに移動し、*Inherited and host macros* を選択します。

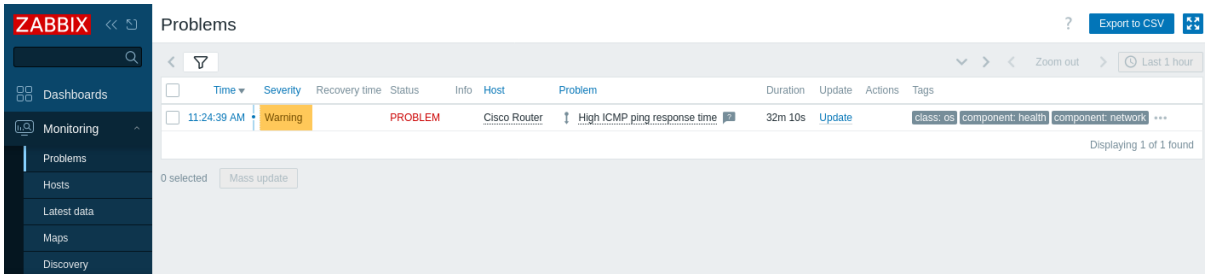
3. `{$ICMP_RESPONSE_TIME_WARN}` マクロ（または同様の応答時間しきい値マクロ）を見つけます。

4. 非常に低い値（例: 0.001）を設定し、ping 応答がこの値を超えたときにアラートが発生するようにします。

5. *Update* をクリックして変更を適用します。

6. Zabbix がシミュレートされた問題を検出するまで、しばらく待ちます。

7. *Monitoring* > *Problems* に移動し、アラートが表示されることを確認します（例: ”High ICMP ping response time”）。



アラートが設定されている 場合は、障害アラートも受信します。

8. マクロの値を元の設定に戻し、*Update* をクリックして変更を保存します。

9. 障害が解消され、*Problems* セクションから消えることを確認します。

SNMP 監視のトラブルシューティング

Zabbix Web インターフェースで SNMP アイコンが RED になっている、またはデータが収集されない場合は、次の手順を試してください。

1. SNMP 接続を確認します。

SNMPv2 の場合は、Zabbix サーバーから次のコマンドを実行します。

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

このコマンドは、デバイスが SNMP クエリに応答することを確認します。

SNMPv3 の場合は、適切な SNMPv3 認証情報を指定します。

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

これにより、SNMPv3 の認証情報が正しく、デバイスが安全に応答していることを確認できます。

2. 前提条件で説明されているとおり、MIB ファイルがインストールされ、有効になっていることを確認します。
そのためには、ネットワークデバイスを問い合わせたときに次のコマンドでエラーが出ないことを確認してください。

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

これにより、エラーなしで変換された OID が返されるはずです。

3. Zabbix で設定されている SNMP バージョンと認証情報が、デバイス側の設定と一致していることを確認します。
たとえば、Zabbix のホスト設定にある SNMP 設定を確認し、デバイスの設定と照合してください。
Cisco デバイスでは、次のコマンドで SNMP 設定を確認できます。

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

これにより、コミュニティ文字列（SNMPv2 の場合）または SNMPv3 のユーザー情報が正しいことを確認できます。

4. ネットワークデバイスで SNMP が正しく有効になっていることを確認します。
Cisco ルーターでは、コンソールにログインして次のコマンドを実行します。

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

このコマンドは、現在有効な SNMP 設定を表示し、SNMP が正しく構成されていることの確認に役立ちます。

5. Zabbix サーバーとデバイス間で、ファイアウォールやネットワークの問題により SNMP トラフィック（通常はポート 161）がブロックされていないことを確認します。
次のコマンドで接続をテストできます。

nc -zv <device_IP> 161

`nc -zv` は、ポート 161 がデバイス上で開いて待ち受けているかどうかを確認します。

また、Ubuntu で UFW を使用している場合は、ファイアウォールの状態を確認します。

```
sudo ufw status
```

または、iptables の場合は次を実行します。

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

6. 問題の特定に役立つよう、SNMP 関連のエラーがないか Zabbix サーバーのログファイルを確認します。

```
tail -f /tmp/zabbix_server.log
```

`tail -f` を使うと、ログの更新をリアルタイムで監視できます。

関連情報**:**

- アイテムの作成 - 追加のメトリクスを追加する方法を学びます。
- SNMP エージェント - Zabbix による SNMP 監視に関する追加情報です。
- ネットワークデバイス向けの標準テンプレート - 利用可能な SNMP テンプレートに関する情報です。
- SNMP OID の検出 - スイッチ上での SNMP 検出に関する追加情報です。
- ネットワーク検出ルールの設定 - Zabbix がホストとサービスを検出するために使用するネットワーク検出ルールの設定方法に関する追加情報です。

アクティブチェックを使用して **Windows** イベントログを監視する

はじめに

このガイドでは、アクティブチェックを使用して Zabbix で Windows のイベントログを監視する方法を説明します。Zabbix の Windows 固有のアイテムキーを使用すると、失敗したログオン試行、システムエラーなどの重要なイベントをリアルタイムで収集および分析できます。

このガイドの対象者

このガイドは、Windows のイベントログを監視したい Zabbix の新規ユーザーおよびネットワーク管理者向けに作成されています。詳細な設定オプションについては、Windows 固有のアイテムキー のドキュメントを参照してください。

前提条件

このガイドを進める前に、OS の手順に従って Zabbix サーバーと Zabbix Web インターフェースをダウンロードしてインストール する必要があります。また、監視対象の Windows マシンに Zabbix エージェントを ダウンロードしてインストール する必要もあります。

Windows イベントログ監視用の Zabbix エージェントを設定する

1. Windows ホストで Zabbix エージェント設定ファイルを開き、ServerActiveパラメータが Zabbix サーバーの IP アドレスに設定されていること、およびHostnameパラメータがZabbix Web インターフェースで定義するホスト名と一致していることを確認します。これにより、エージェントは自身のホストおよび指定された Zabbix サーバーからアクティブチェックを要求できるようになります。例:

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

2. 変更を適用するため、Zabbix エージェントサービスを再起動します。

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

3. Windows ホストが稼働していることを確認します。

- Windows ホストで Zabbix エージェントサービスが実行されていることを確認します。
- Windows ホストがポート 10051 で Zabbix サーバーに接続できることを確認します。Windows ホストから接続をテストするには、PowerShell を開き、次のコマンドを実行します。

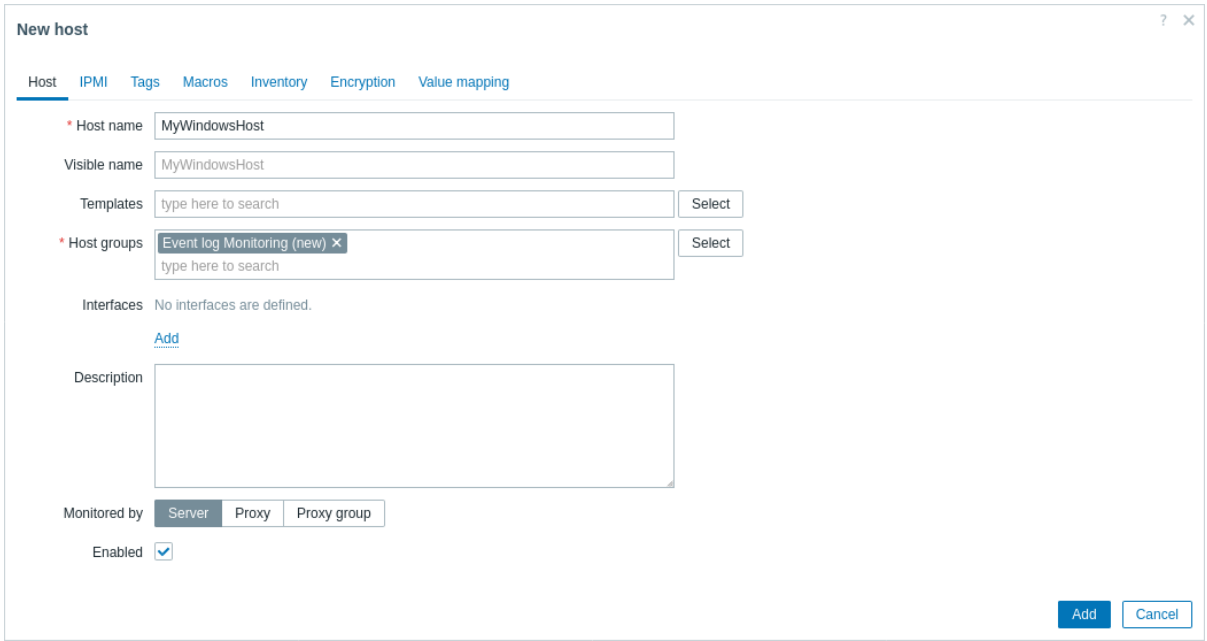
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Zabbix Web インターフェースの設定

1. データ収集 > ホストに移動し、ホストを作成します。

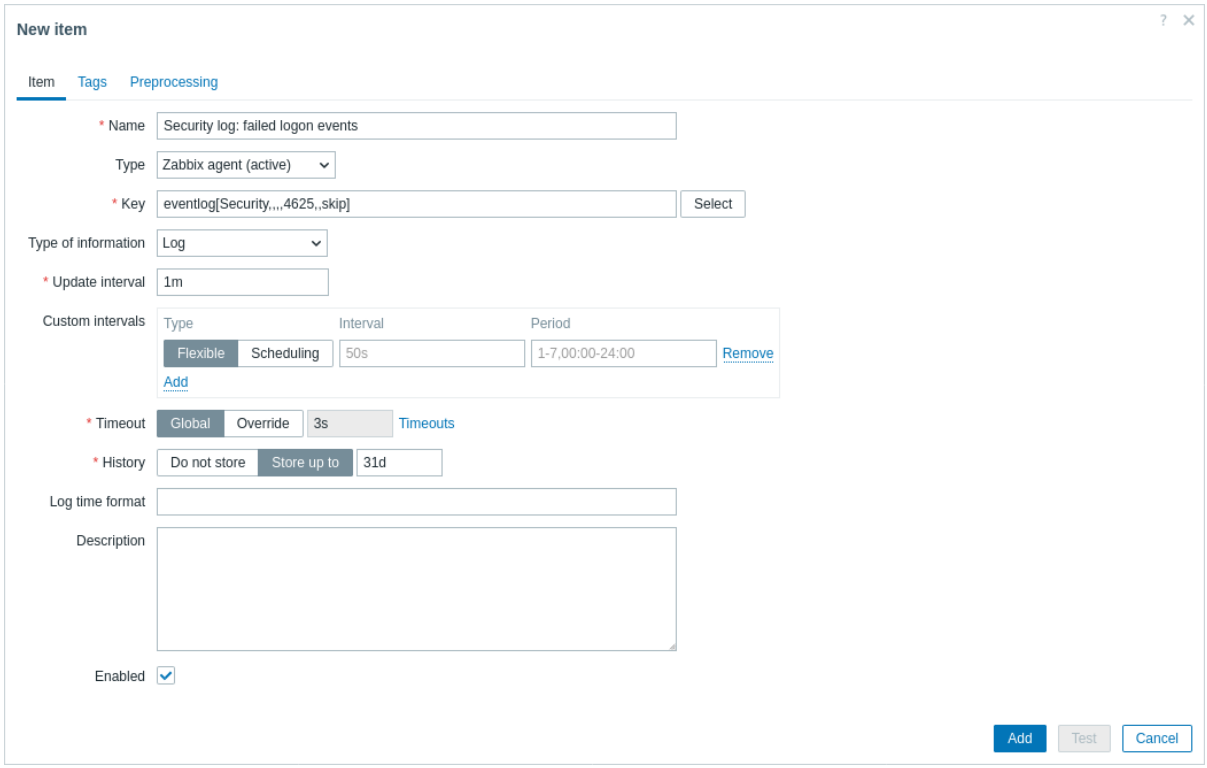
- ホスト名フィールドに、ホスト名を入力します（例: ”MyWindowsHost”）。
- ホストグループフィールドに、ホストグループを入力または選択します（例: ”Event log Monitoring”）。
- 追加をクリックして、設定したホストを保存します。

Note:
テンプレートフィールドでは、同じホスト上の他のアクティブなアイテムが更新されているかどうかを確認することでトラブルシューティングに役立つように、”Windows by Zabbix agent active” テンプレートを追加できます。



2. 次のパラメーターで新しいアイテムを作成します。

- 名前フィールドに、説明的なアイテム名を入力します（例: ”Security log: failed logon events”）。
- タイプドロップダウンで、”Zabbix agent (active)” を選択します（Event log monitoring に必要です）。
- キーフィールドに、eventlog アイテムキーを使用します。たとえば、Security log の失敗したログオン試行（Event ID: 4625）を監視し、アイテムの直前のチェックより古いエントリを無視する（`skip` パラメーターを使用する）には、次のアイテムキーを入力します: `eventlog[Security,,,,4625,,skip]`
- 情報の種類ドロップダウンで、”Log” を選択します。



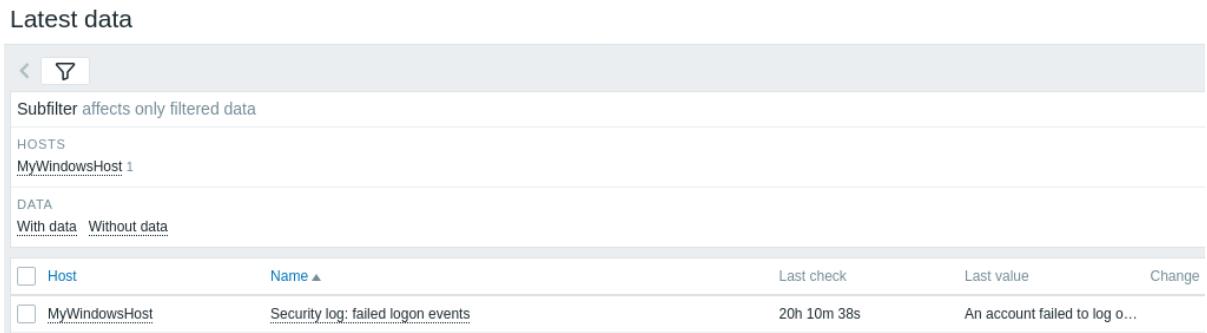
3. 追加をクリックして、アイテムを保存します。

収集したメトリックのテストと表示

おめでとうございます。Zabbix は Windows のイベントログを収集するように設定されました。イベントログが収集されていることを確認するには、Windows アカウントからログアウトし、誤った認証情報でログインを試行して、「Security log: failed logon events」アイテムをテストできます。

次に、Zabbix Web インターフェースで収集されたログを表示します。

1. Zabbix Web インターフェースで *Monitoring* > *Latest data* に移動します。



2. *Name* フィールドで、”MyWindowsHost” ホストによってフィルターします。

3. *History* をクリックして、記録されたログ値を表示します。

MyWindowsHost: Security log: failed logon events

Timestamp	Local time	Source	Severity	Event ID	Value
2025-04-07 11:13:11 PM	2025-04-07 11:12:19 PM	Microsoft-Windows-Security-Auditing	Failure Audit	4625	An account failed to log on. Subject: Security ID: NT AUTHORITY\SYSTEM Account Name: DESKTOP-NNEM26U$ Account Domain: WORKGROUP Logon ID: 0x3E7 Logon Type: 2 Account For Which Logon Failed: Security ID: NULL SID Account Name: Admin Account Domain: DESKTOP-NNEM26U Failure Information: Failure Reason: Unknown user name or bad password. Status: 0xC000006D Sub Status: 0xC000006A

4. ログ値が表示されない場合は、ガイドのTroubleshooting セクションに進んでください。

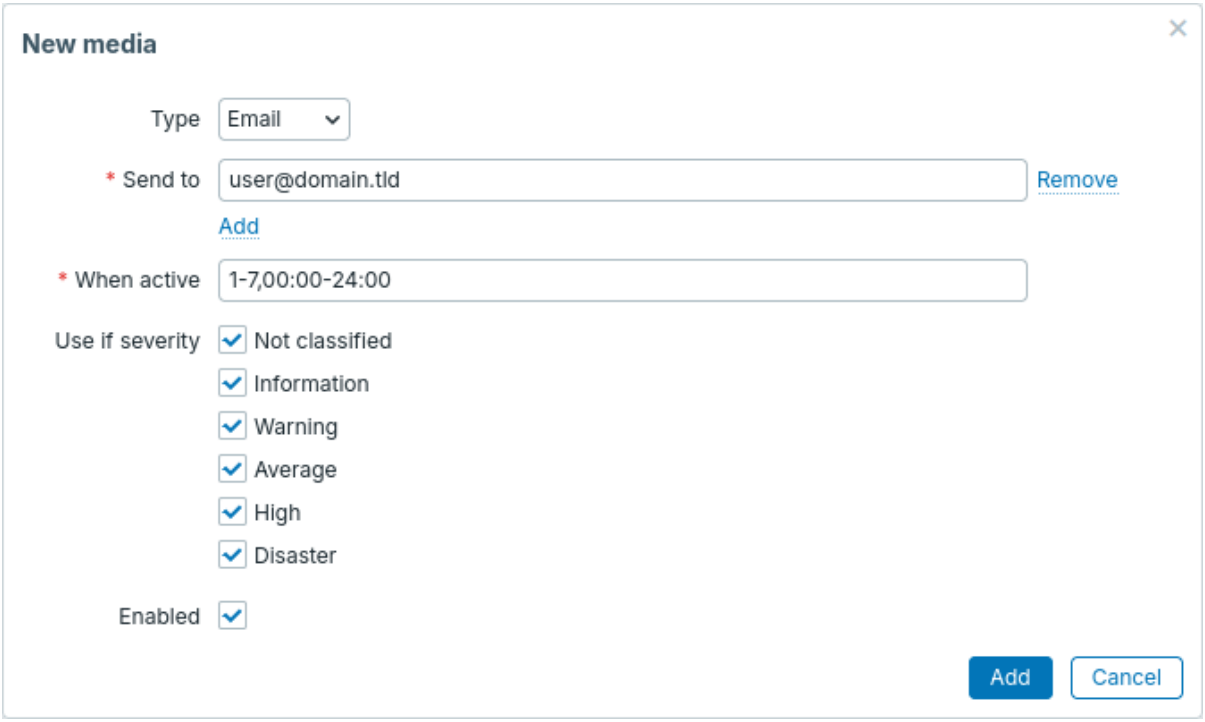
障害アラートの設定

このガイドでは、メールアラートを送信するための基本的な設定手順を説明します。

1. データ収集 > ホストに移動し、イベントログアイテムが監視対象のパターンを記録したときに発火するトリガーを定義します。たとえば、Security ログでログオン失敗を検出するには、find()関数を使用します:

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

2. ユーザー設定 > プロファイル に移動し、メディアタブを開いてメールアドレスを追加します。



3. 障害通知の受信 のガイドに従います。

次回、Zabbix が障害を検出すると、メールでアラートを受け取れるはずです。

トラブルシューティング

Windows イベントログの収集または表示で問題が発生した場合は、以下のヒントを使用して、一般的な問題を特定し解決してください。

1. Zabbix サーバー (Linux) で、次のコマンドを使用して iptables ルールを一覧表示します。

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

そして、TCP ポート 10051 に対する ACCEPT ルールがあることを確認します。

2. `eventlog[...]` キーで、Windows 固有のアイテムキー に示されているとおり、ログ名（大文字小文字を区別）、イベント ID、モード（例: skip）、およびその他のパラメータが正確に使用されていることを確認します。

関連情報:

- アイテムの作成 - 追加のメトリクスを追加する方法を学びます。
- Microsoft Windows 上の Zabbix エージェント - 詳細なインストール手順です。
- Zabbix エージェントで Windows を監視する - Zabbix エージェントを使用して Windows マシンの基本監視を設定するための包括的なガイドです。
- Windows 固有のアイテムキー - イベントログ監視用のものを含む、Zabbix エージェントでサポートされる Windows 固有のアイテムキーに関する詳細情報です。
- ログファイルの監視 - Windows のイベントログにも適用できる、ログファイルの集中監視と分析のために Zabbix を設定する手順です。

インストール: 追加ガイド

Zabbix を最も簡単にインストールする方法については、インストールと最初のステップを参照してください。

以下のガイドでは、Zabbix またはそのコンポーネントをインストールするための追加の方法や、システム固有の方法について説明します。

- コンテナからのインストール
- ソースからのインストール
- パッケージからのインストールに関する注意事項
- Windows MSI からの Zabbix エージェントのインストール
- Windows での Zabbix エージェント 2 のビルド
- Windows での Zabbix エージェントのビルド
- PKG からの macOS エージェントのインストール
- IBM AIX 上の Zabbix エージェント
- macOS での Zabbix エージェントのビルド
- Zabbix アプライアンス
- Zabbix サーバーの AWS デプロイメントガイド
- Zabbix プロキシの AWS デプロイメントガイド
- Zabbix サーバーの Azure デプロイメントガイド
- Zabbix プロキシの Azure デプロイメントガイド
- Google Cloud デプロイメントガイド

コンテナからのインストール

概要

このページでは、Docker または Docker Compose を使用して Zabbix をデプロイする方法について説明します。

Docker Compose は、Zabbix をデプロイする最も簡単な方法です。設定ファイルを読み込み、完全な Zabbix 環境を構成するすべてのコンテナを、正しい順序で自動的に起動します。

Docker（手動デプロイ） では、各コンポーネントを段階的にデプロイすることで、同じ結果を実現します。

前提条件

開始する前に、システムに Docker（1.12.0 以降）がインストールされていることを確認してください。インストールされていない場合は、Docker のインストールガイドに従ってください。Docker Compose でデプロイする場合は、Docker Compose（2.24.0 以降）も必要です。

一部の Zabbix コンポーネントでは、Docker を実行しているホストで特定のポートを開放する必要があります（例: Zabbix サーバーの場合は 10051/TCP、SNMP トラップの場合は 162/UDP）。Zabbix コンポーネントで使用されるポートの一覧については、要件を参

照してください。Zabbix サーバーおよびエージェントでは、それぞれのコンテナで ZBX_LISTENPORT 環境変数を設定することで、デフォルトポートを変更できます。

利用可能な Docker イメージ

Zabbix では、各 Zabbix コンポーネント用の Docker イメージを提供しており、すべてDocker Hubで公開しています。各イメージを使用して、そのコンポーネントを実行するコンテナを作成できます。

Zabbix コンポーネント	Docker イメージ
エージェント	zabbix/zabbix-agent
エージェント 2	zabbix/zabbix-agent2
サーバー（MySQL）	zabbix/zabbix-server-mysql
サーバー（PostgreSQL）	zabbix/zabbix-server-pgsql
Web インターフェース（Apache + MySQL）	zabbix/zabbix-web-apache-mysql
Web インターフェース（Apache + PostgreSQL）	zabbix/zabbix-web-apache-pgsql
Web インターフェース（Nginx + MySQL）	zabbix/zabbix-web-nginx-mysql
Web インターフェース（Nginx + PostgreSQL）	zabbix/zabbix-web-nginx-pgsql
プロキシ（SQLite3）	zabbix/zabbix-proxy-sqlite3
プロキシ（MySQL）	zabbix/zabbix-proxy-mysql
Java ゲートウェイ	zabbix/zabbix-java-gateway
Web サービス	zabbix/zabbix-web-service
SNMP トラップ	zabbix/zabbix-snmptraps
MCP サーバー	zabbix/zabbix-mcp-server

Note:
SNMP トラップを使用するには、SNMP トラップコンテナが Zabbix サーバーまたはプロキシコンテナとボリュームを共有する必要があります（Docker Hub のこのイメージの使用方法および以下の例を参照してください）。

Docker Hub 上のすべての Zabbix コンポーネントイメージは、サポート対象のオペレーティングシステムの最新メジャーバージョンをベースにしています。これらのイメージは、基盤となる OS イメージが更新されると自動的に再ビルドされます。

イメージタグ

各イメージは、ベースとなるオペレーティングシステムと Zabbix バージョンを選択するためのタグをサポートしています。

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

サポートされている <os> の値:

- alpine - Alpine Linux
- ubuntu - Ubuntu
- centos - CentOS Stream
- ol - Oracle Linux
- ltsc2022 - Windows 11 LTSC 2022（Zabbix エージェントのみ）

サポートされている <version> の値:

- latest - Alpine Linux 上の最新の安定版 Zabbix バージョン
- <os>-latest - 選択した OS 上の最新の安定版 Zabbix バージョン
- <os>-trunk - 選択した OS 上の最新の開発版（nightly）ビルド
- <os>-X.X-latest - 選択した OS 上の特定の Zabbix メジャーバージョンにおける最新の Zabbix マイナーリリース
- <os>-X.X.* - 選択した OS 上の特定の Zabbix マイナーリリース

例:

```
#### Alpine Linux 上の最新の安定版 Zabbix プロキシ（SQLite3）:
docker pull zabbix/zabbix-proxy-sqlite3:latest

#### Ubuntu 上の最新の安定版 Zabbix プロキシ（SQLite3）:
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

#### Ubuntu 上の Zabbix サーバー（MySQL）の最新の開発版（nightly）ビルド:
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk

#### Alpine Linux 上の Zabbix サーバー（MySQL）の最新の 8.0 マイナーリリース:
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest
```

```
#### Alpine Linux 上の Zabbix サーバー（MySQL）バージョン 8.0.1:
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

Docker Compose は、Zabbix をデプロイする最も迅速な方法です。設定ファイル（Compose ファイル）を読み込み、Docker イメージの取得、コンテナ間の内部ネットワークの作成、ストレージの設定、データベースの初期化、すべてのコンポーネントの正しい順序での起動まで、セットアップ全体を処理します。

公式 Zabbix Dockerfilesリポジトリには、すぐに使用できる Docker Compose ファイルと、さまざまなオペレーティングシステム、データベースバックエンド、Zabbix コンポーネントの設定に対応した.env ベースの設定システムが用意されています。

1. リポジトリをクローンし、リポジトリのディレクトリに移動して、バージョン 8.0 に切り替えます。

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

2. （オプション）環境変数を使用してデプロイをカスタマイズします。デフォルト設定を使用する場合は、この手順をスキップできます。

3. デフォルト設定（Zabbix サーバー、NGINX 上で実行される Web インターフェース、MySQL または PostgreSQL のデータベースバックエンド）をデプロイします。各コンポーネントは Alpine Linux 上の独自のコンテナで実行されます。

```
#### データベースに MySQL を使用する場合:
docker compose -f ./compose.yaml up -d

#### データベースに PostgreSQL を使用する場合:
docker compose -f ./compose_pgsql.yaml up -d
```

コンテナが起動して実行状態になると（通常は 1～3 分以内）、Zabbix サーバーが起動し、Web インターフェースでログインできるようになります。

> **Note:**
> docker compose ps を使用して、コンテナのステータスを確認します。zabbix-docker-server-db-init-1 コンテナを除くすべてのコンテナが Up ステータスになっている必要があります。Exited と表示されるコンテナがある場合は、docker logs -f <container-name> を使用してログを確認し、エラーを調べてください。

環境変数

Docker Compose の動作と Zabbix コンポーネントの設定は、環境変数を使用してカスタマイズできます。

Compose レベルの変数（.env ファイルで定義）は、使用する Docker イメージ、ポート、ネットワーク IP 範囲を制御します。これらの変数は、docker compose コマンドの前にインラインで指定するか、.env ファイルを編集して設定できます。

たとえば、次のコマンドは、デフォルトの Alpine Linux ではなく Ubuntu ベースのイメージを使用し、カスタム HTTP（8282）および HTTPS（8443）ポートで NGINX 経由の Web インターフェースを公開する、完全なマルチコンテナ Zabbix 環境をデプロイします。

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

コンポーネントレベルの変数（env_vars/.env_<component>ファイルで定義）は、Zabbix コンポーネントの設定を制御します。docker compose コマンドを実行する前に、該当する.env_<component> ファイルを編集してください。

たとえば、次の変数を編集することで、Zabbix サーバーのパッシブポーラー数とキャッシュ設定を調整し、Web インターフェースのタイムゾーンを設定できます。

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

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

Note:
コンポーネントレベルの環境変数は、異なる命名形式を使用して Zabbix コンポーネントの設定パラメータに対応します（例：ZBX_STARTPOLLERS はStartPollersに対応します）。一部の変数は Docker 固有のものであり、一部の設定パラメータは変更できません（例：PIDFile および LogType）。環境変数を使用する場合は、Docker Hub にある各コンポーネントのDocker イメージの説明に記載された環境変数セクションを参照してください。

ボリューム

Docker Compose は、Compose ファイルと同じ場所に作成される zbx_env/ ディレクトリに永続データを保存します。
このディレクトリにより、コンテナの再起動や更新後もコンポーネントのデータが保持されます。

zbx_env/ の内容は、各コンポーネントのイメージごとに事前定義されています。例えば、次のとおりです。

- zbx_env/etc/ - Zabbix Web インターフェースコンテナ で使用されます
- zbx_env/usr/ および zbx_env/var/ - Zabbix サーバーコンテナ で使用されます

各ボリュームの詳細については、Docker Hub にある各コンポーネントのDocker イメージ の説明内の *Allowed volumes* セクションを参照してください。

Makefile

Zabbix Dockerfiles リポジトリには、一般的な Docker Compose タスクのショートカットとして使用できる Makefile も用意されています。完全な `docker compose` コマンドを使用する代わりに、より短い `make` コマンドを使用できます（利用可能なオプションを確認するには `make help` を実行します）。

```
#### デフォルトのデプロイ（Zabbix サーバー、Web インターフェース、MySQL、すべて Alpine Linux 上）
make up

#### カスタムデプロイ（Zabbix サーバー、カスタム NGINX ポートを使用する Web インターフェース、PostgreSQL、すべて Ubuntu
make up \
  OS=ubuntu \
  DB=pgsql \
  ZABBIX_WEB_NGINX_HTTP_PORT=8282 \
  ZABBIX_WEB_NGINX_HTTPS_PORT=8443
```

Attention:
コンテナを停止または削除する場合は、デプロイ時と同じデータベースタイプを必ず指定してください（例: `make down DB=pgsql`）。

デフォルトでは、`make up` は最小限のサービス（Zabbix サーバー、Web インターフェース、データベース）のみを起動します。これにより、セットアップを軽量に保ち、不要なコンポーネントが起動されるのを防ぎます。追加のコンポーネントを含めるには、次の Compose プロファイル を使用できます。

```
#### デフォルトのデプロイ + Zabbix エージェント、Java ゲートウェイ、Web サービス、SNMP トラップ
make up COMPOSE_PROFILES=full

#### COMPOSE_PROFILES=full + Zabbix プロキシ（MySQL および SQLite3）
make up COMPOSE_PROFILES=all
```

Docker (手動デプロイ)

Zabbix を段階的にデプロイし、個々のコンポーネントを実行し、既存の環境と統合する場合や、Podman などの代替コンテナランタイムを使用する場合は、手動デプロイを使用します。

たとえば、Zabbix プロキシをデプロイするには、次のコマンドを実行します。

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

このコマンドは次の処理を行います。

- zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest Docker イメージ を取得します。

- Docker イメージを基に zabbix-proxy-sqlite3 コンテナを作成して起動します（--init および -d フラグを使用）。
- ZBX_SERVER_HOST 環境変数を使用して、Zabbix プロキシのServer パラメータを設定します。このパラメータは、プロキシが設定データを取得し、収集したデータを送信する Zabbix サーバーの IP アドレスを指定します。アクティブプロキシの動作に必要な他の変数はデフォルト値を使用するため、省略できます。

> **Note:**
> 環境変数は、Zabbix コンポーネントの設定パラメータに対して異なる命名規則を使用します（例: ZBX_SERVER_HOST はServer に対応します）。一部の変数は Docker 固有であり、変更できない設定パラメータもあります（例: PIDFile と LogType）。環境変数を使用する場合は、Docker Hub 上の各コンポーネントのDocker イメージ の説明にある *Environment variables* セクションを参照してください。

- Docker によって管理される保存先（例: /var/lib/docker/volumes/zabbix-proxy-data/）をコンテナの /var/lib/zabbix ディレクトリに接続し、コンテナが削除されても Zabbix プロキシのデータが永続的に保存されるようにします。

> **Note:**
> 各ボリュームの詳細については、Docker Hub 上の各コンポーネントのDocker イメージ の説明にある *Allowed volumes* セクションを参照してください。

Zabbix プロキシコンテナをデプロイした後は、Zabbix Web インターフェースでプロキシを追加し、このプロキシで監視するホストを設定 できます。

以下の例では、さらに 3 つのデプロイシナリオを紹介します。

- Zabbix サーバー (MySQL) と Java ゲートウェイ
- Zabbix サーバー (PostgreSQL) と SNMP トラップ
- RHEL 8-10 上の Zabbix サーバー (MySQL) と Java ゲートウェイ

その他の例については、Docker Hub 上の各コンポーネントのDocker イメージ の説明を参照してください。

Java ゲートウェイを使用した Zabbix サーバー（MySQL）のデプロイ

この例では、MySQL バックエンド、JMX 監視用の Java ゲートウェイ、および NGINX ベースの Web インターフェースを使用して Zabbix サーバーをデプロイする方法を説明します。

1. 専用の Docker ネットワークを作成し、すべての Zabbix コンポーネントコンテナがコンテナ名で相互に通信できるようにします。

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

2. 空の MySQL サーバーコンテナを起動します（zabbix_pwd と root_pwd を強力なパスワードに置き換え、以降の手順ではその値を使用します）。

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

3. MySQL のログを追跡し、MySQL が接続を受け付ける状態になるまで待ちます（ログを終了するには Ctrl+C を使用します）。

```
docker logs -f mysql-server

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

4. Zabbix Java ゲートウェイコンテナを起動します。

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

5. MySQL サーバーコンテナで log_bin_trust_function_creators を有効にします。これにより、ストアドファンクションに対するセキュリティ制限が一時的に緩和されます。この設定は後の手順で無効にします。MySQL でbinary loggingが有効な場合

(MySQL 8.0 以降のデフォルト)、Zabbix データベースユーザーが SUPER 権限なしでストアドファンクションを作成できるようになります。次のコマンドを実行します。

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

6. Zabbix サーバーコンテナを起動し、Java ゲートウェイおよび MySQL サーバーコンテナに接続します。

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

7. Zabbix サーバーのログを追跡し、Zabbix サーバーによるデータベーススキーマの初期化が完了するまで待ちます（ログを終了するには Ctrl+C を使用します）。

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

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

8. MySQL サーバーコンテナで log_bin_trust_function_creators を無効にします。これにより、SUPER 権限を持たないユーザーがストアドファンクションを作成できないようにするセキュリティ制限が再び適用されます。次のコマンドを実行します。

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

9. Zabbix Web インターフェースコンテナを起動し、Zabbix サーバーおよび MySQL サーバーコンテナに接続します。

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

コンテナが起動して実行状態になると、Zabbix サーバーが起動し、Web インターフェースからログインできるようになります。

> **Note:**
> docker ps を使用してコンテナの状態を確認します。すべてのコンテナが Up 状態になっている必要があります。Exited と表示されるコンテナがある場合は、docker logs -f <container-name> でログを確認し、エラーを調査してください。

SNMP トラップを使用した Zabbix サーバー（PostgreSQL）のデプロイ

この例では、PostgreSQL バックエンド、SNMP トラップ、および NGINX ベースの Web インターフェースを使用して Zabbix サーバーをデプロイする方法を説明します。

1. 専用の Docker ネットワークを作成し、すべての Zabbix コンポーネントコンテナがコンテナ名で相互に通信できるようにします。

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

2. 空の PostgreSQL サーバーコンテナを起動します（zabbix_pwd を強力なパスワードに置き換え、以降の手順ではその値を使用します）。

```
docker run --name postgres-server -t \
  -e POSTGRES_USER="zabbix" \
```

```
  -e POSTGRES_PASSWORD="zabbix_pwd" \
  -e POSTGRES_DB="zabbix" \
  --network=zabbix-net \
  --restart unless-stopped \
  -d postgres:latest
```

3. Zabbix SNMP トラップコンテナを起動します。

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

4. Zabbix サーバーコンテナを起動し、PostgreSQL サーバーおよび SNMP トラップコンテナに接続します。

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

5. Zabbix サーバーのログを追跡し、Zabbix サーバーによるデータベーススキーマの初期化が完了するまで待ちます（ログを終了するには Ctrl+C を使用します）。

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

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

6. Zabbix Web インターフェースコンテナを起動し、Zabbix サーバーおよび PostgreSQL サーバーコンテナに接続します。

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

コンテナが起動して実行状態になると、Zabbix サーバーが起動し、Web インターフェースからログインできるようになります。

> **Note:**
> docker ps を使用してコンテナの状態を確認します。すべてのコンテナが Up 状態になっている必要があります。Exited と表示されるコンテナがある場合は、docker logs -f <container-name> でログを確認し、エラーを調査してください。

RHEL 8～10 で Java ゲートウェイを使用した Zabbix サーバー（MySQL）のデプロイ

この例では、Red Hat Enterprise Linux 8、9、または 10 上で、MySQL バックエンド、JMX 監視用の Java ゲートウェイ、および NGINX ベースの Web インターフェースを使用して Zabbix サーバーをデプロイする方法を説明します。

Red Hat Enterprise Linux では、Docker ではなく Podman を使用することが推奨されます。Podman は Docker と同様に動作しますが、root としてバックグラウンドサービスを実行する必要がないため、Red Hat 環境により適しています。

1. zabbix という名前の新しい Pod を作成し、Zabbix Web インターフェースおよび Zabbix サーバーのトラッパー用にポートを公開します。

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

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

```
podman run --name zabbix-agent \
  -e ZBX_SERVER_HOST="127.0.0.1,localhost" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-agent-80:latest
```

3. RHEL ホスト上に./mysql/ディレクトリを作成します。

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

4. 空の MySQL サーバーコンテナを起動します（zabbix_pwd と root_pwd を強力なパスワードに置き換え、以降の手順ではその値を使用します）。

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

5. MySQL のログを追跡し、MySQL が接続を受け付ける状態になるまで待ちます（ログを終了するには Ctrl+C を使用します）。

```
podman logs -f mysql-server

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

6. Zabbix Java ゲートウェイコンテナを起動します。

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

7. MySQL サーバーコンテナで log_bin_trust_function_creators を有効にします。これにより、ストアドファンクションに対するセキュリティ制限が一時的に緩和されます。この設定は後の手順で無効にします。MySQL でbinary loggingが有効な場合（MySQL 8.0 以降のデフォルト）、Zabbix データベースユーザーが SUPER 権限なしでストアドファンクションを作成できるようになります。次のコマンドを実行します。

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

8. Zabbix サーバーコンテナを起動し、Java ゲートウェイおよび MySQL サーバーコンテナに接続します。

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

9. Zabbix サーバーのログを追跡し、Zabbix サーバーによるデータベーススキーマの初期化が完了するまで待ちます（ログを終了するには Ctrl+C を使用します）。

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

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

10. MySQL サーバーコンテナで log_bin_trust_function_creators を無効にします。これにより、SUPER 権限を持たないユーザーがストアドファンクションを作成できないようにするセキュリティ制限が再び適用されます。次のコマンドを実行します。

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

11. Zabbix Web インターフェースコンテナを起動し、Zabbix サーバーおよび MySQL サーバーコンテナに接続します。

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

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

コンテナが起動して実行状態になると、Zabbix サーバーが起動し、Web インターフェースからログインできるようになります。

Note:
docker ps を使用してコンテナの状態を確認します。すべてのコンテナが Up 状態になっている必要があります。Exited と表示されるコンテナがある場合は、docker logs -f <container-name> でログを確認し、エラーを調査してください。

ソースからのインストール

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

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

Zabbix デーモンのインストール

1 ソースアーカイブのダウンロード

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

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

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

2 ユーザーアカウントの作成

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

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

RedHat 系システムの場合:

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

Debian 系システムの場合:

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

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

セキュリティ推奨

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

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

ホームディレクトリ（オプション）

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

RedHat 系システムの場合:

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

Debian 系システムの場合:

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

3 Zabbix データベースの作成

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

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

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

4 ソースを設定する

Zabbix サーバー、Zabbix プロキシ、または Zabbix エージェントのビルドには、GNU 拡張を含む C99 が必要です。このバージョンは、CFLAGS="-std=gnu99" を設定して明示的に指定できます。

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

Note:
Zabbix Git リポジトリからインストールする場合は、最初に以下を実行する必要があります。
`./bootstrap.sh`

Zabbix サーバーまたはプロキシのソースを設定する場合は、使用するデータベースの種類を指定する必要があります。サーバーまたはプロキシのプロセスでコンパイルできるデータベースの種類は、一度に 1 つだけです。

サポートされている設定オプションをすべて表示するには、展開した Zabbix ソースディレクトリ内で以下を実行します。

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

Zabbix サーバーおよびエージェントのソースを設定するには、次のように実行します。

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

Zabbix サーバー（PostgreSQL などを使用）のソースを設定するには、次のように実行します。

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

Zabbix プロキシ（SQLite などを使用）のソースを設定するには、次のように実行します。

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

Zabbix エージェントのソースを設定するには、次のように実行します。

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

または、Zabbix エージェント 2 の場合は次のように実行します。

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

Note:
Zabbix エージェント 2 をビルドするには、サポートされている Go のバージョンをインストールする必要があります。

コンパイルオプションに関する注意事項：

- `--enable-agent` - Zabbix エージェントに加えて、Zabbix getおよびZabbix senderのコマンドラインユーティリティをコンパイルします。
- `--with-libcurl` - 仮想マシンの監視、SMTP 認証、`web.page.*` Zabbix エージェントのアイテム、およびテレメトリクエリアイテムに必要です。詳細は要件（libcurl）も参照してください。
- `--with-libxml2` - 仮想マシンの監視に必要です。
- `--with-libpcre2[=DIR]` - Zabbix は常に PCRE2 ライブラリを使用してコンパイルされます。このオプションでは、カスタム PCRE2 インストールパスのみを指定できます。
- `--with-mysql=/path/to/mysql_config` - 特定の MySQL クライアントライブラリ設定へのパスを指定します。複数のバージョンの MySQL または MariaDB がインストールされている場合に便利です。
- `--enable-static` - ライブラリを静的にリンクします（Solarisではサポートされていません）。必要なライブラリがないシステムにコンパイル済みバイナリを配布する予定がある場合に使用します。Zabbix サーバーのビルド時には推奨されません。サーバーを静的にビルドするには、すべての外部ライブラリの静的バージョンが必要です。configure スクリプトはこれを自動的には確認しません。
- `--with-stacksize=<value>` - スレッドごとのスタックサイズをキロバイト単位で設定します(例：`--with-stacksize=512`)。スタックオーバーフローが原因で Zabbix がクラッシュまたはフリーズする場合（デフォルトのスレッドスタック制限が低いシステムでの前処理中など）は、この値を増やすことができます。
- `--with-apm`（`--enable-proxy` との併用時のみ）- Zabbix プロキシでのアプリケーションパフォーマンス監視（APM）に必要です。詳細は要件（g++、gRPC、protobuf、Abseil）も参照してください。

Attention:
ライブラリが見つからないなどの理由で`./configure` が失敗した場合は、`config.log` ファイルで詳細なエラー情報を確認してください。
たとえば、`libssl` が見つからない場合、最初に表示されるエラーメッセージは誤解を招く可能性があります。
```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```
この場合、`config.log` には実際の原因が示されます。
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

関連項目：

- 暗号化サポート付きで Zabbix をコンパイルする
- 既知のコンパイルに関する問題

5 すべてをビルドしてインストールする

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

```
make install
```

この手順は、十分な権限を持つユーザー（通常は `root`、または `sudo` を使用）で実行してください。

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

Note:
`/usr/local` 以外の場所を指定するには、ソースの設定を行う前の手順で `--prefix` キーを使用します。たとえば `--prefix=/home/zabbix` です。この場合、デーモンのバイナリは `<prefix>/sbin` に、ユーティリティは `<prefix>/bin` にインストールされます。マニュアルページは `<prefix>/share` にインストールされます。

6 設定ファイルの確認と編集

- Zabbix エージェント設定ファイル（`zabbix_agentd.conf`）を編集します。

`zabbix_agentd` がインストールされているすべてのホストで、このファイルを設定する必要があります。

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

- Zabbix サーバー設定ファイル（`zabbix_server.conf`）を編集します。

データベース名、ユーザー、パスワード（使用する場合）を指定する必要があります。

小規模なインストール（監視対象ホストが 10 台以下）の場合は、その他のパラメータをデフォルトのまま使用できます。ただし、Zabbix サーバー（またはプロキシ）のパフォーマンスを最大限に高めたい場合は、デフォルトのパラメータを変更してください。

- Zabbix プロキシをインストールした場合は、プロキシ設定ファイル（`zabbix_proxy.conf`）を編集します。

サーバーの IP アドレスとプロキシのホスト名（サーバーが認識できるもの）、およびデータベース名、ユーザー、パスワード（使用する場合）を指定する必要があります。

> **Note:**
> SQLite を使用する場合は、データベースファイルの完全なパスを指定する必要があります。DB ユーザーとパスワードは必要ありません。

7 デーモンを起動する

サーバー側で zabbix_server を実行します。

```
zabbix_server
```

> **Note:**
> システムで 36MB（またはそれより少し多い）の共有メモリを割り当てられることを確認してください。そうでない場合、サーバーが起動しない可能性があり、サーバーのログファイルに”Cannot allocate shared memory for <type of cache>.” と表示されます。これは FreeBSD、Solaris 8 で発生する場合があります。

監視対象のすべてのマシンで zabbix_agentd を実行します。

```
zabbix_agentd
```

> **Note:**
> システムで 2MB の共有メモリを割り当てられることを確認してください。そうでない場合、エージェントが起動しない可能性があり、エージェントのログファイルに”Cannot allocate shared memory for collector.” と表示されます。これは Solaris 8 で発生する場合があります。

Zabbix プロキシをインストールしている場合は、zabbix_proxy を実行します。

```
zabbix_proxy
```

Zabbix Web インターフェースのインストール

PHP ファイルのコピー

Zabbix Web インターフェースは PHP で記述されているため、実行するには PHP をサポートする Web サーバーが必要です。
インストールは、ui ディレクトリから PHP ファイルを Web サーバーの HTML ドキュメントディレクトリに単純にコピーするだけで完了します。

Apache Web サーバーでの HTML ドキュメントディレクトリの一般的な場所は次のとおりです。

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

HTML ルートの代わりにサブディレクトリを使用することを推奨します。
サブディレクトリを作成し、Zabbix Web インターフェースのファイルをそこにコピーするには、<htdocs> を実際のディレクトリに置き換えて、次のコマンドを実行します。

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

英語以外の言語を使用する予定がある場合は、手順について追加の Web インターフェース言語のインストール を参照してください。

フロントエンドのインストール

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

Zabbix エージェント 2 のロード可能なプラグインのインストール

Zabbix エージェント 2 のロード可能なプラグインのインストールは、組み込みプラグインで対応していない監視対象（例: MongoDB サーバーまたはクラスター、PostgreSQL およびその派生版など）を監視したい場合にのみ必要です。
ロード可能なプラグイン と組み込みプラグイン の一覧を参照してください。

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

ソースからインストールするには、まず ダウンロード して、ロード可能なプラグインのソースアーカイブを展開します。

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

```
make
```

Note:
Zabbix エージェント 2 のロード可能なプラグインをビルドするには、サポートされている Go のバージョン がインストールされている必要があります。

プラグインの実行ファイルは、Zabbix エージェント 2 から読み込める限り、任意の場所に配置できます。
プラグインの設定ファイル内でプラグインバイナリへのパスを指定します。たとえば、PostgreSQL プラグインでは postgresql.conf に次のように記述します。

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

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

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

プラグインの設定方法の詳細については、セットアップ を参照してください。

Zabbix が提供するロード可能なプラグインは、次のビルドターゲットを持つシンプルな makefile を使用します。

- make - プラグインをビルドする
- make clean - プラグインのビルドによって作成されたすべてのファイルを削除する
- make check - 自己テストを実行する（実際の監視対象、たとえば PostgreSQL データベースが必要です）
- make style - golangci-lint で Go コードのスタイルを確認する
- make format - go fmt で Go コードを整形する
- make dist - すべての依存関係を含むソースアーカイブを作成する

Java ゲートウェイのインストール

JMX アプリケーションを監視したい場合にのみ、Java ゲートウェイをインストールする必要があります。Java ゲートウェイは軽量で、データベースを必要としません。

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

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

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

Java ゲートウェイをコンパイルして JAR ファイルにパッケージ化するには、make を実行します。この手順では、javac と jar の実行ファイルがパス上にある必要があります。

```
make
```

これで、src/zabbix_java/bin に zabbix-java-gateway-$VERSION.jar ファイルが作成されます。配布ディレクトリ内の src/zabbix_java から Java ゲートウェイを実行することに問題がなければ、Java ゲートウェイ の設定と実行に関する手順に進めます。そうでない場合は、十分な権限があることを確認して make install を実行してください。

```
make install
```

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

Zabbix web service のインストール

Zabbix web service のインストールは、スケジュールレポートを使用したい場合にのみ必要です。

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

Zabbix web service をコンパイルするには、./configure スクリプトを --enable-webservice オプション付きで実行します。

Note:
Zabbix web service をビルドするには、サポートされている Go のバージョンがインストールされている必要があります。

web service がインストールされているマシンで zabbix_web_service を実行します。

zabbix_web_service

スケジュールレポート生成の設定について詳しくは、セットアップに進んでください。

Zabbix MCP サーバーのインストール

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

Zabbix MCP サーバーをコンパイルするには、MCP サーバーのソースディレクトリで次のコマンドを実行します。

```
make build
```

> **Note:**
> Zabbix MCP サーバーをビルドするには、サポートされている Go のバージョンをインストールする必要があります。

zabbix_mcp_server -V の出力および Zabbix API リクエストの User-Agent で特定のバージョンを報告するには、VERSION 変数を設定します。例:

```
make build VERSION=8.0.0
```

バイナリを/usr/local/sbin に、サンプル設定ファイルを/etc/zabbix にインストールするには、次を実行します。

```
sudo make install
```

設定ファイル/etc/zabbix/zabbix_mcp_server.conf を編集し、少なくとも次のパラメータを設定します。

- FrontendURL - api_jsonrpc.php で終わる Zabbix API の URL。
- AllowedIP - 接続を許可する MCP クライアントのアドレス。

MCP クライアントからの接続を暗号化するには、TLSAccept=cert を設定して TLSCertFile および TLSKeyFile を指定するか、TLS を処理する信頼できるリバースプロキシの背後で MCP サーバーを実行します。

zabbix ユーザーが存在しない場合（たとえば、マシンに MCP サーバーのみがインストールされている場合）は、ユーザーアカウントの作成の説明に従って作成します。

設定ファイルを検証し、zabbix_mcp_server を起動します。

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

サーバーの設定および MCP クライアントの接続についての詳細は、MCP サーバーを参照してください。

Zabbix ソースコードの取得

Zabbix ソースコードを取得する方法はいくつかあります。

- 公式 Zabbix Web サイトから、リリース済みの安定版をダウンロードできます
- 公式 Zabbix Web サイトの開発者ページから、nightly build をダウンロードできます
- Git ソースコードリポジトリシステムから、最新の開発版を取得できます。
  - 完全なリポジトリの主要な場所は次のとおりです。https://git.zabbix.com/scm/zbx/zabbix.git
  - master およびサポート対象のリリースは、Github にもミラーされています。https://github.com/zabbix/zabbix

リポジトリをクローンするには、Git クライアントをインストールする必要があります。公式のコマンドライン Git クライアントパッケージは、ディストリビューションでは通常 **git** と呼ばれます。たとえば、Debian/Ubuntu でインストールするには、次を実行します。

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

Zabbix のすべてのソースコードを取得するには、コードを配置したいディレクトリに移動して、次を実行します。

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

コンパイル時の問題

以下は、ソースからの Zabbix のコンパイルに関して既知の問題です。その他のすべてのケースについては、既知の問題ページを参照してください。

非標準の場所にあるライブラリ

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

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

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

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

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

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

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

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

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

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

一部のシステムでスタックサイズが小さすぎる

スタックオーバーフローが原因で Zabbix がクラッシュまたはフリーズする場合は、ソースを設定する際に --with-stacksize オプションを使用して、スレッドごとのスタックサイズを増やすことができます。
この問題は、特に複数のスレッドが作成される前処理 中に、デフォルトのスレッドスタック制限が低いシステムで発生することがあります。

次の例では、スレッドごとのスタックサイズを 512 KB に設定しています。

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

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

パッケージからのインストールに関する注意

概要

Zabbix 8.0 の公式 プレリリースパッケージは、Zabbix Web サイト から入手できます。ここでオペレーティングシステムと Zabbix コンポーネントを選択すると、インストール手順とコマンドを生成できます。

以下のパッケージのインストールに関する注意事項に加えて、インストールとセットアップのページも参照してください。

パッケージは、以下の Linux ディストリビューションで利用できます。

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

Attention:
一部の OS ディストリビューション（特に Debian ベースのディストリビューション）では、独自の Zabbix パッケージが提供されています。これらのパッケージは Zabbix ではサポートされておらず、古い場合や、最新の機能およびバグ修正が含まれていない場合があります。Zabbix 公式リポジトリの公式パッケージのみを使用することを推奨します。以前にオペレーティングシステムのリポジトリから Zabbix をインストールした場合は、OS リポジトリから Zabbix パッケージをアップグレードする手順を参照してください。

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

必要に応じて、Zabbix エージェント/エージェント 2、Zabbix get、および Zabbix sender 用の個別パッケージをZabbix 公式リポジトリから入手できます。

Zabbix では、Linux 以外のオペレーティングシステム向けに、コンパイル済みの Zabbix エージェントバイナリも提供しています。詳細は以下を参照してください。

- MSI からの Windows エージェントのインストール

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

パッケージのインストールに関する注意事項

以下の注意事項は、すべてのシステムに適用されます。

- PostgreSQL を使用する場合、Zabbix サーバー/プロキシの設定で DBHost=localhost（または IP アドレス）を指定すると、PostgreSQL はローカル UNIX ソケットではなくネットワークソケットを使用します。関連する設定手順については、SELinux の設定を参照してください。
- TimescaleDB を使用する場合は、追加のTimescaleDB のセットアップを参照してください。
- Zabbix Java ゲートウェイ (JMX アプリケーションの監視用) をインストールする場合は、RHEL ベースのシステムおよびDebian ベースのシステム向けの追加のセットアップ手順を参照してください。
- Zabbix MCP サーバーをインストールする場合は、zabbix-mcp-server パッケージをインストールし、サービスを起動する前に設定ファイル（少なくとも FrontendURL と AllowedIP）を更新してください。
- Zabbix エージェントを root として実行する場合は、エージェントを root として実行するを参照してください。

以下の注意事項は、RHEL およびその派生ディストリビューションに適用されます。

- EL9 用の EPEL リポジトリを有効にしている場合、そこからも Zabbix パッケージが提供されるため、公式の Zabbix パッケージをインストールする前に、パッケージ解決の対象から除外する必要があります。EPEL の Zabbix パッケージが誤ってインストールされる場合を参照してください。
- Red Hat UBI 環境に Zabbix パッケージをインストールする場合は、Red Hat UBI 環境向けの RHEL 用 Zabbix パッケージを参照してください。
- ICMP ping アイテムを使用する場合、fping 用のパッケージもZabbix 公式リポジトリから入手できます。

SELinux の設定

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

SELinux が enforcing モードで有効になっている場合、Zabbix Web インターフェースとサーバー間の通信を有効にするには、次のコマンドを実行します。

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

```
setsebool -P httpd_can_connect_zabbix on
```

データベースにネットワーク経由でアクセスする場合（PostgreSQL の localhost を含む）、Zabbix Web インターフェースがデータベースに接続できるようにもします。

```
setsebool -P httpd_can_network_connect_db on
```

RHEL 7 より前の場合:

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

SELinux の設定を適用した後、Apache を再起動します。

```
systemctl restart httpd
```

必要に応じて、Zabbix Official Repository から事前定義済みの zabbix-selinux-policy パッケージをインストールできます。
このパッケージは、設定の複雑さが原因でユーザーが SELinux を無効化してしまうことを防ぎ、Zabbix の導入を簡素化するために、サポート対象のすべての OS バージョン向けに提供されています。
 AlmaLinux、CentOS Stream、RHEL、Rocky Linux、Oracle Linux のバージョン 10 では、SELinux ポリシーは zabbix-selinux-policy-main パッケージによって提供されます。
zabbix-selinux-policy パッケージは zabbix-selinux-policy-main または selinux-policy-targeted-extra のいずれかに依存します。
zabbix-selinux-policy-main パッケージは selinux-policy-targeted-extra と競合するため、両方のパッケージを同時にインストールすることはできません。

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

zabbix-selinux-policy(AlmaLinux、CentOS Stream、RHEL、Rocky Linux、Oracle Linux のバージョン 10 では zabbix-selinux-policy-ma
パッケージには、Zabbix がソケットを作成および使用できるようにし、HTTPd が PostgreSQL（Web インターフェースで使用）へ接

続できるようにする基本的な SELinux ポリシーが含まれています。

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

```
module zabbix_policy 1.2;

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

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

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

Debuginfo パッケージ

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

Debuginfo パッケージは、以下のバージョンで利用可能です: Red Hat Enterprise Linux 7 以前（派生版を含む）、および SUSE Linux Enterprise Server 15 以前（派生版を含む）。

zabbix-debuginfo リポジトリを有効にするには:

- RHEL 7 では、/etc/yum.repos.d/zabbix.repo を編集し、zabbix-debuginfo セクションで enabled=1 に設定します:

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

- SUSE では、/etc/zypp/repos.d/zabbix.repo を編集し、zabbix-debuginfo セクションで enabled=1 に設定します:

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

有効化したら、パッケージをインストールします:

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

```
dnf install zabbix-debuginfo
```

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

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

不安定リリースリポジトリの有効化

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

まず、最新の zabbix-release パッケージをインストールするか、更新してください。システムで rc パッケージを有効にするには、次の操作を行います:

Red Hat Enterprise Linux

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

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

Debian/Ubuntu

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

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

SUSE

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

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

MSI からの **Windows** エージェントのインストール

概要

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

MSI インストールの最小 OS 要件は以下の通りです。

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

32 ビットパッケージは 64 ビットシステムにはインストールできません。

パッケージには以下が含まれます。

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

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

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

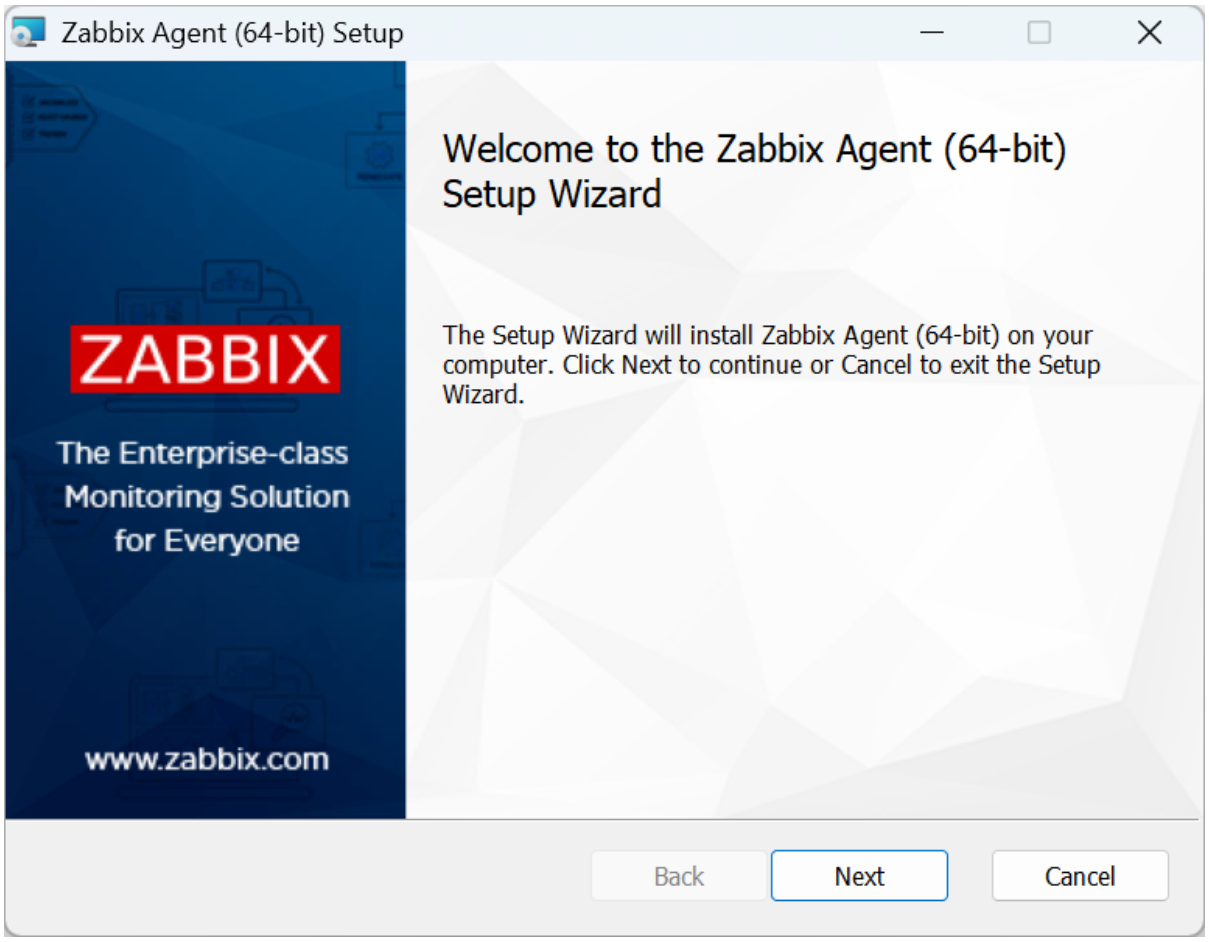
MSI パッケージによるインストールは完全にサポートされていますが、適切なエラー処理のために少なくともMicrosoft .NET Framework 2のインストールを推奨します。

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

セットアップウィザードからのインストール

以下のインストール手順は、Zabbix エージェントと Zabbix エージェント 2 の両方に適用されます。

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

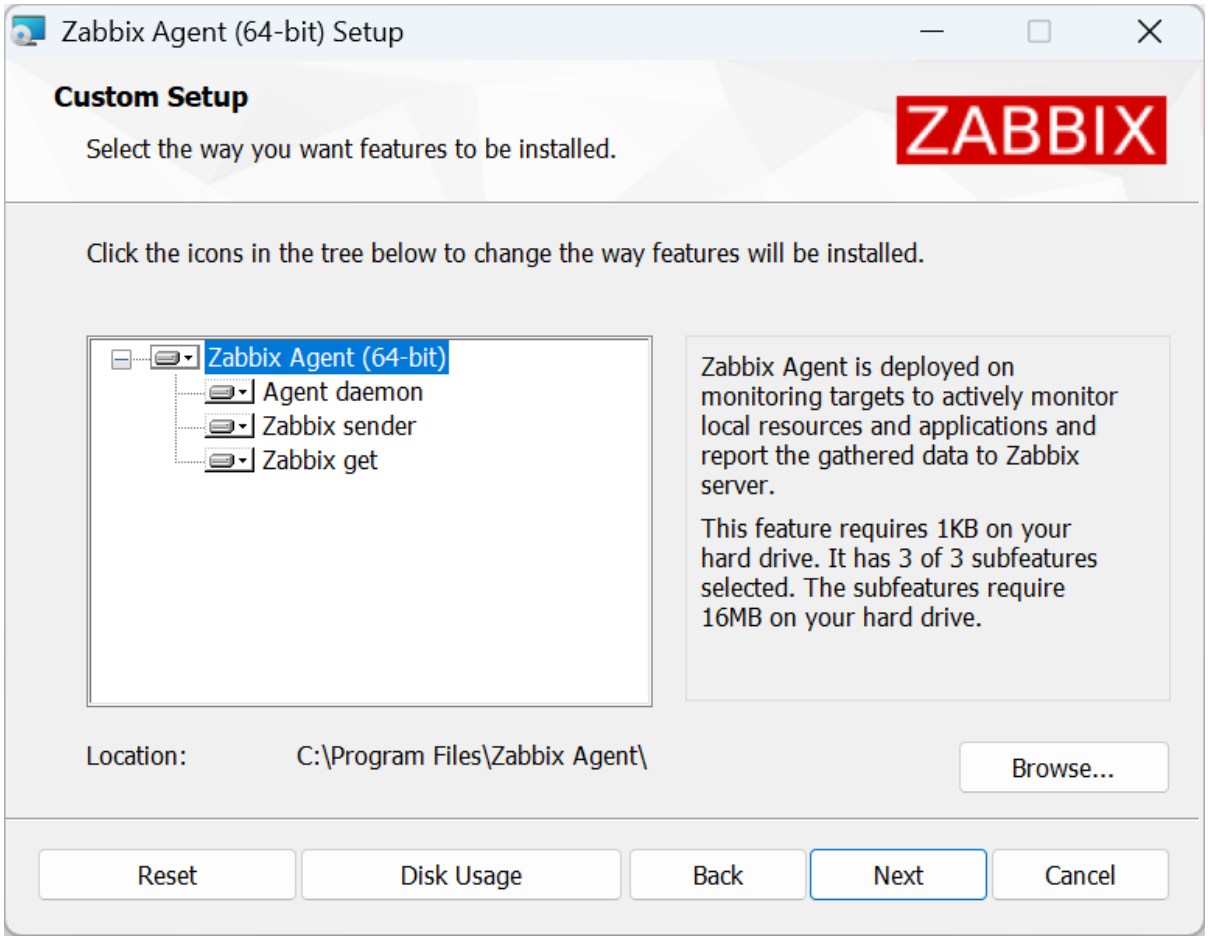


2. エンドユーザーライセンス契約に同意します。



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

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

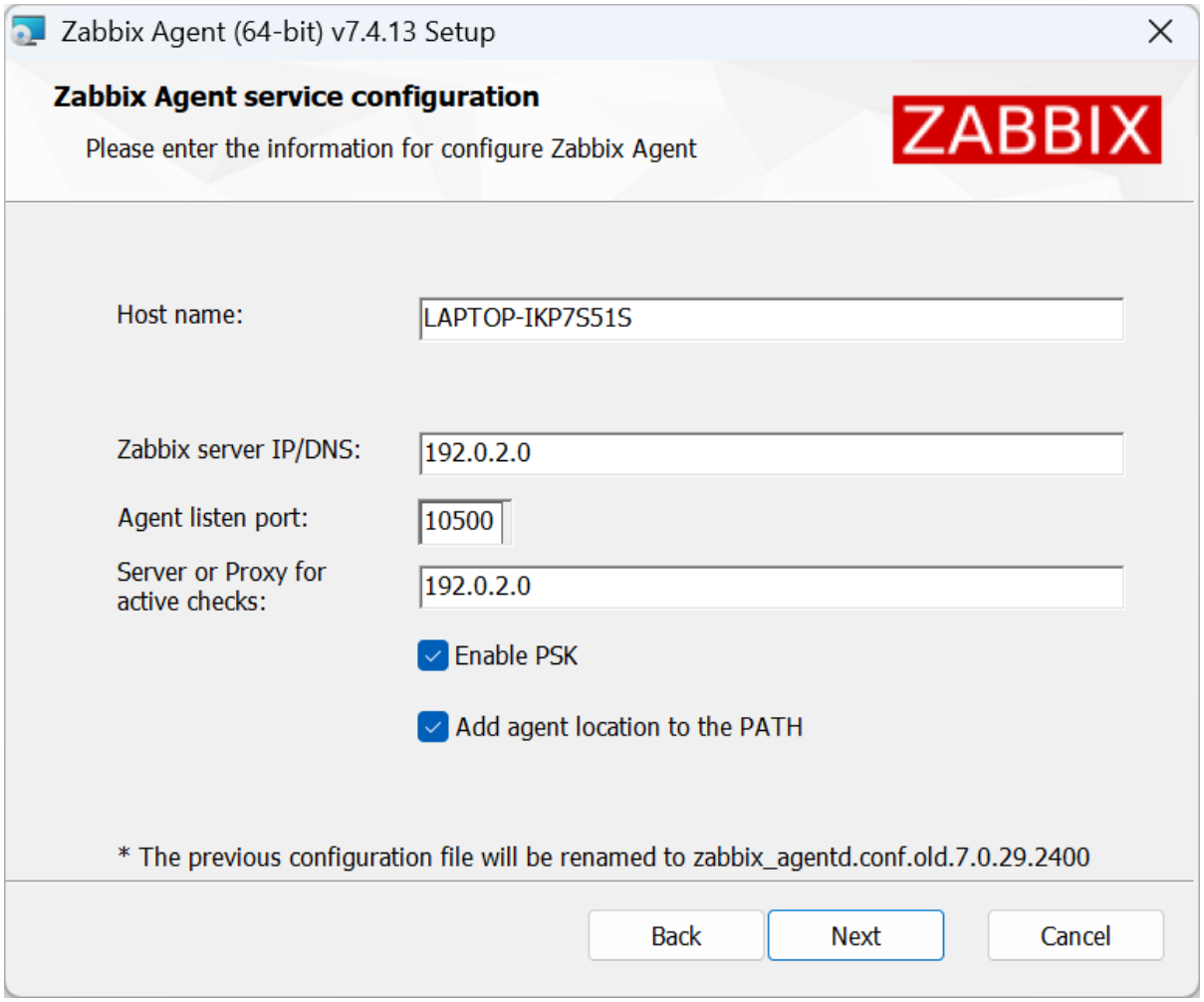


Note:
`Program Files` ディレクトリ外のカスタムの場所を使用し、保存先フォルダーがすでに存在する場合、インストーラーはその権限を変更しません。新規インストール時には、適切な権限を手動で設定するか、権限を変更せずに続行するかを選択するよう求められます。

4. 以下のパラメータを設定します。これらの値は、Zabbix エージェント設定ファイルに設定されます。

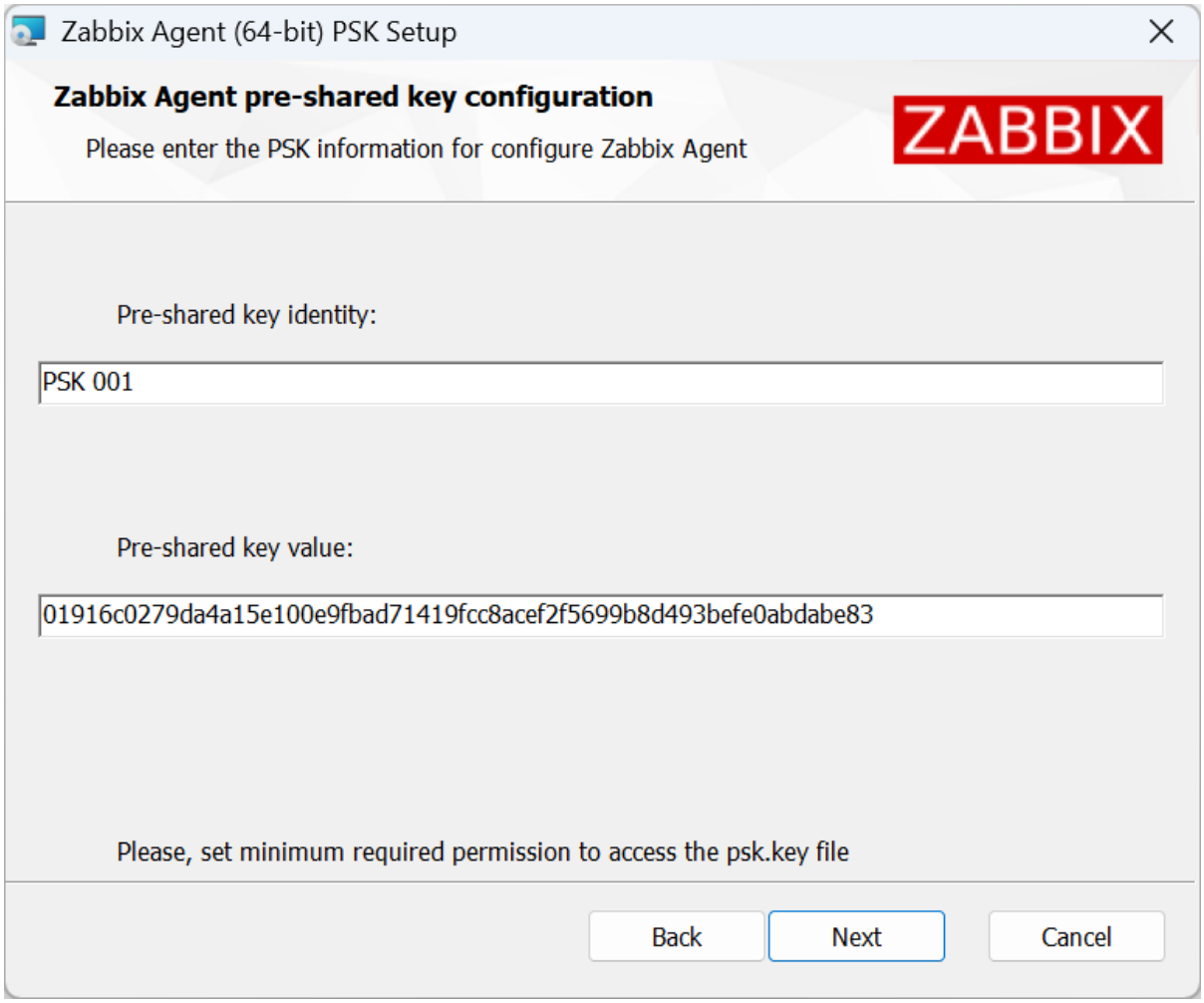
パラメータ	説明
ホスト名	Zabbix エージェントをインストールするマシンのホスト名。Hostname パラメータを設定します。
Zabbix サーバーの *IP/DNS*	CIDR 表記を使用できる、カンマ区切りの IP アドレスのリスト、または Zabbix サーバーや Zabbix プロキシの DNS 名。このパラメータは必須です。Server パラメータを設定します。
エージェントの待ち受けポート	エージェントがサーバーからの接続を待ち受けるポート。ListenPort パラメータを設定します。
アクティブチェック用のサーバーまたはプロキシ	アクティブチェックを取得する Zabbix サーバーまたはプロキシのアドレス、あるいはクラスタ設定。サーバーまたはプロキシのアドレスは、IP アドレスまたは DNS 名に、コロンで区切った任意のポートを付加したものです。ServerActive パラメータを設定します。
PSK を有効にする	事前共有鍵を使用して TLS サポートを有効にするには、チェックボックスを選択します。TLSConnect および TLSAccept パラメータを `psk` に設定します。
エージェントの場所を *PATH* に追加	Zabbix エージェントの場所をシステムの PATH 変数に追加するには、チェックボックスを選択します。

Note:
既存の Zabbix エージェントが検出された場合、その設定ファイルのパラメータ値が表示されます。また、インストール中に既存の設定ファイルの名前が変更され、新しい設定ファイルが作成されます。



5. 前の手順で *PSK* を有効にするチェックボックスを選択した場合は、PSK パラメータを設定します。これらのパラメータも Zabbix エージェント設定ファイルに設定されます。

パラメータ	説明
事前共有鍵の識別子	事前共有鍵の識別子文字列。TLSPSKIdentity パラメータを設定します。
事前共有鍵の値	事前共有鍵の文字列値。鍵を含む `psk.key` ファイルを作成し、TLSPSKFile パラメータを鍵の場所（デフォルト: `C:\Program Files\Zabbix Agent\psk.key`）に設定します。事前共有鍵ファイルのセキュリティ設定を調整し、Zabbix エージェント（またはエージェントを実行するユーザー）のみが読み取れるようにアクセスを制限することを推奨します。

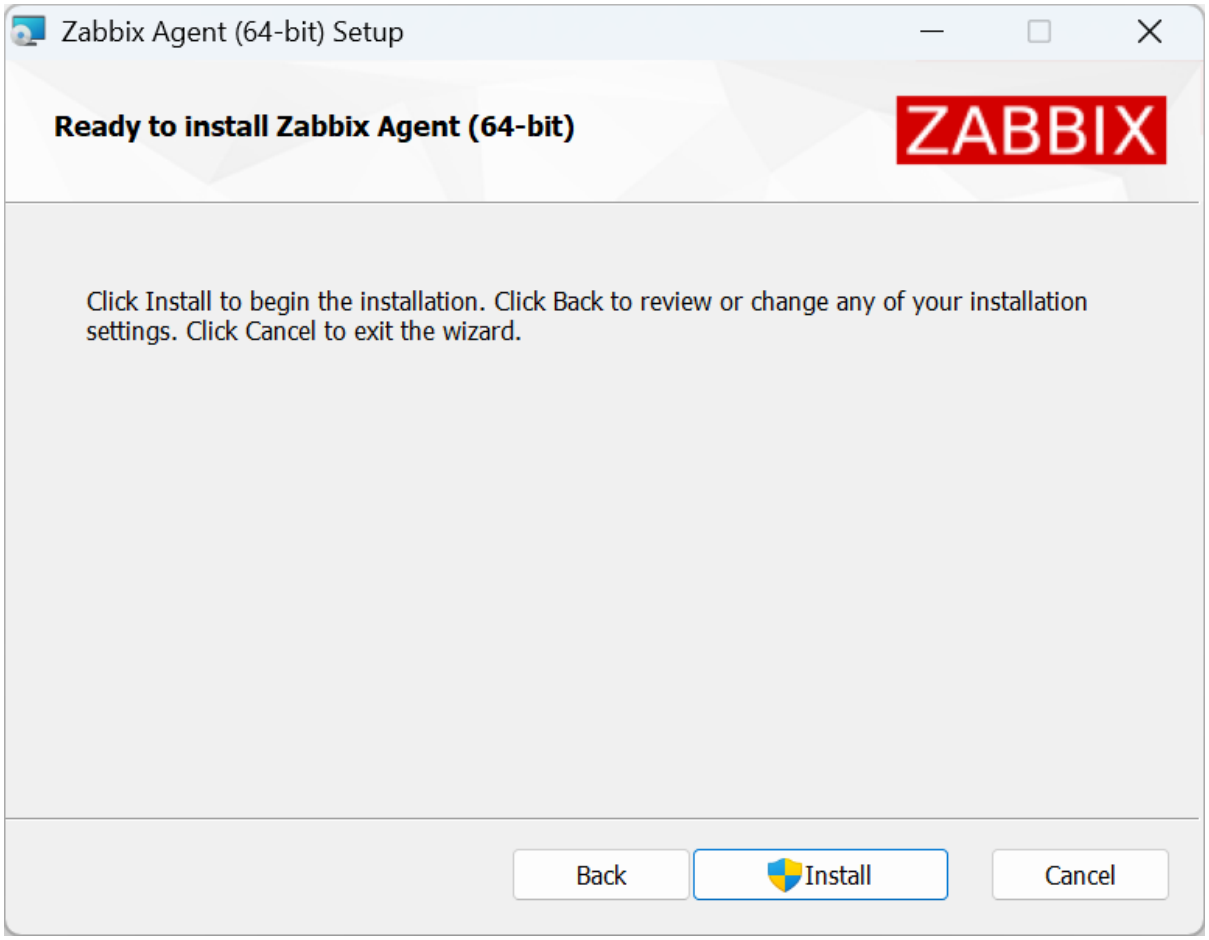


6. *Install* をクリックして、インストールを開始します。

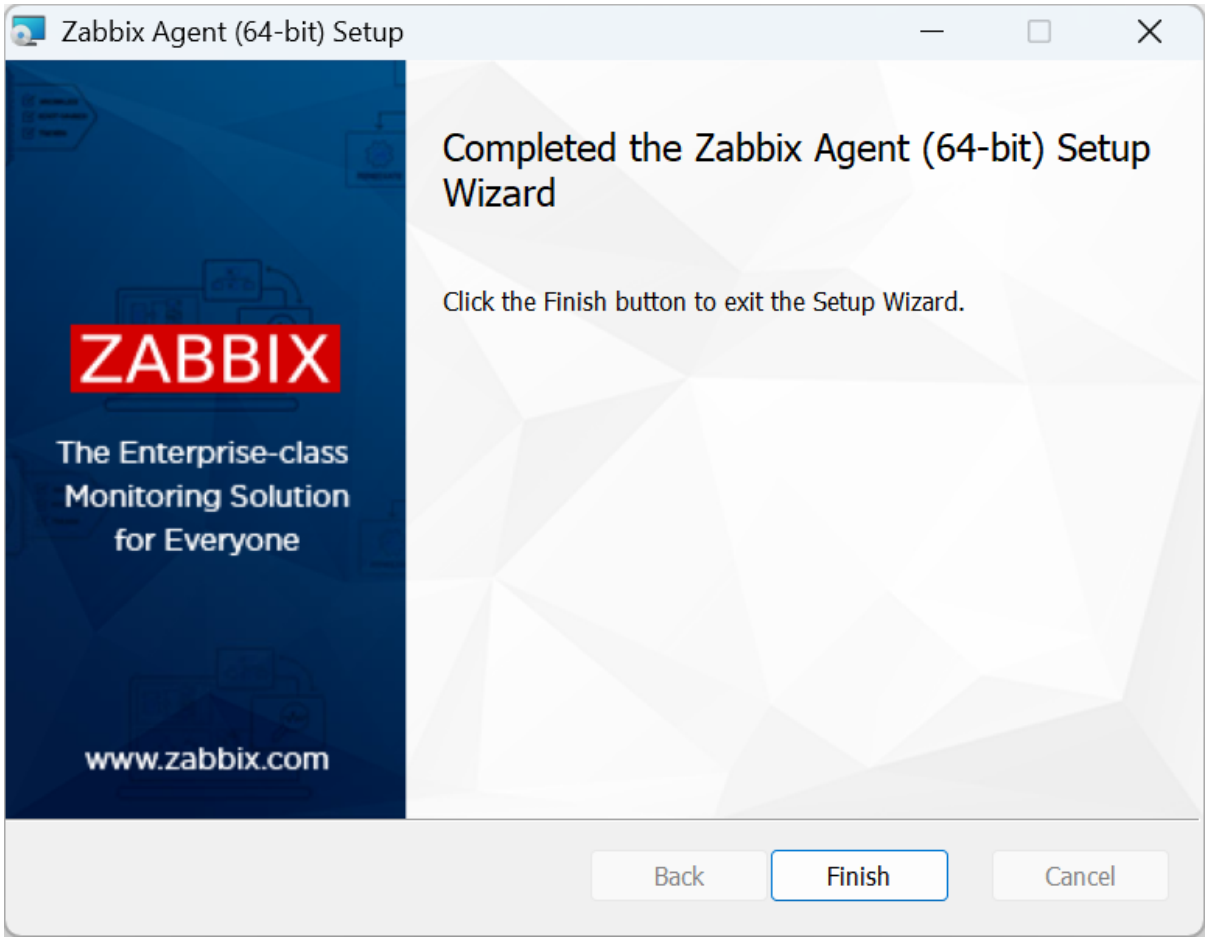
選択したすべての Zabbix コンポーネントと Zabbix エージェント設定ファイルが、指定した場所（デフォルト: `C:\Program Files\Zabbix Agent`）にインストールされます。Zabbix エージェント 2 も同様ですが、組み込みプラグイン用の追加設定ファイルが `zabbix_agent2.d\plugins.d` サブフォルダーにインストールされます。

さらに、`zabbix_agentd.exe`（または `zabbix_agent2.exe`）が、遅延自動起動（Windows Vista/Server 2008 より前の Windows バージョンでは自動起動）に設定された Windows サービスとして登録されます。

インストール中に別のバージョンの Zabbix エージェントが実行されている場合は、アプリケーションを終了して再起動を試みるか、開いたままにするかを選択するよう求められます。後者を選択した場合は、再起動が必要になります。



7. *Finish* ボタンをクリックして、セットアップウィザードを終了します。



コマンドラインからのインストール

Zabbix エージェントは、msiexec を使用して MSI インストーラを実行することで、コマンドラインからインストールできます。たとえば、次のようにします。

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

この方法では、パラメータを使用した無人インストールとカスタム設定が可能です。

サポートされるパラメーター

Zabbix agent MSI インストーラーパッケージは、Zabbix エージェントと Zabbix agent 2 の両方で次のパラメーターをサポートします。

Note:
Zabbix agent/agent2 のパラメーターは、インストール時に設定ファイルで設定されます。パラメーター名をクリックすると、Zabbix agent (Windows) ページで詳細な説明と設定例を確認できます。Zabbix agent 2 については、Zabbix agent 2 (Windows) ページを参照してください。

Parameter	Description
ADDDEFAULT	既定の構成でインストールする、カンマ区切りのコンポーネント一覧です。詳細は ADDDEFAULT property を参照してください。 指定可能な値: AgentProgram, GetProgram, SenderProgram, ALL 例: ADDDEFAULT=AgentProgram,GetProgram
ADDLOCAL	ローカルにインストールする、カンマ区切りのコンポーネント一覧です。詳細は ADDLOCAL property を参照してください。 指定可能な値: AgentProgram, GetProgram, SenderProgram, ALL 例: ADDLOCAL=AgentProgram,SenderProgram
ALLOWDENYKEY	Zabbix agent のチェックを制限する ための、セミコロン区切りの AllowKey または DenyKey パラメーターの一覧です。必要に応じて、区切り文字はバックスラッシュでエスケープします (\;)。エージェントの設定ファイルにAllowKey およびDenyKey パラメーターを設定します。 例: ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"
CONF	Zabbix agent 用のテンプレート設定ファイル への完全なパス名です。インストール時に、このファイルはエージェントの設定ファイルになります。ファイルには少なくともServer とLogFile パラメーターが含まれている必要があります。 例: CONF="C:\full\path\to\example.conf"
DONOTSTART	DONOTSTART=1 を使用すると、MSI インストーラーが Zabbix agent サービスを開始しないようにできます。
ENABLEPATH	ENABLEPATH=1 を使用すると、Zabbix agent の場所をシステムの PATH 変数に追加できます。
ENABLEPERSISTENTBUFFER	Zabbix agent 2 のみ。アクティブアイテムに対してローカルの永続ストレージの使用を有効にします。
HOSTINTERFACE	ホストインターフェースを定義するオプションのパラメーターです。
HOSTMETADATA	ホストメタデータを定義するオプションのパラメーターです。
HOSTMETADATAITEM	ホストメタデータを取得するために使用するアイテムを定義するオプションのパラメーターです。
HOSTNAME	ホスト名を定義するオプションのパラメーターです。
HOSTNAMEITEM	ホスト名を取得するために使用するアイテムを定義するオプションのパラメーターです。
INCLUDE	Zabbix agent の設定ファイルに含める、セミコロン区切りの個別ファイルまたはディレクトリ内のすべてのファイルの一覧です。
IGNOREALREADYEXISTSFOLDER	新規インストール時に、IGNOREALREADYEXISTSFOLDER=1 を使用すると、Program Files ディレクトリ外の既存の配置先フォルダーへのインストールを許可できます。このパラメーターを指定しない場合、そのような配置先フォルダーが既に存在するとインストールは失敗します。
INSTALLFOLDER	Zabbix コンポーネントと Zabbix agent の設定ファイルをインストールするフォルダーへの完全なパス名です。Zabbix agent 2 では、組み込みプラグイン 用の追加設定ファイルが zabbix_agent2.d\plugins.d サブフォルダーにインストールされます。 例: INSTALLFOLDER="C:\Program Files\Zabbix Agent"
LISTENIP	エージェントが待ち受ける、カンマ区切りの IP アドレス一覧です。
LISTENPORT	エージェントは、このポートでサーバーからの接続を待ち受けます。
LOGFILE	Zabbix agent のログファイル名です。
LOGTYPE	ログ出力の種類です。
NONMSICONFNAME	Zabbix agent 用のカスタム設定ファイル への完全なパス名です。インストール時に、このファイルに含まれる有効なエージェント設定パラメーター（この表に記載されているものに限定されます）は、新しく作成されるエージェント設定ファイルに書き込まれます。ファイルには少なくともServer パラメーターが含まれている必要があります。 例: NONMSICONFNAME="C:\full\path\to\example.conf"

Parameter	Description
PERSISTENTBUFFERFILE	Zabbix agent 2 のみ。Zabbix agent 2 が SQLite データベースを保持するファイルです。
PERSISTENTBUFFERPERIOD	Zabbix サーバーまたはプロキシへの接続がない場合に、データを保存する期間です。
SERVER	Zabbix サーバーまたは Zabbix プロキシの IP アドレス（必要に応じて CIDR 表記可）または DNS 名の、カンマ区切りの一覧です。このパラメーターは、STARTAGENTS が 0 に設定されている場合を除き、必須です。
SERVERACTIVE	アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレス、またはクラスタ構成です。
SKIP	SKIP=fw を使用すると、MSI インストーラーが Zabbix agent 用の Windows Firewall 例外ルールを追加しないようにできます。
SOURCEIP	Zabbix サーバーまたは Zabbix プロキシへの送信接続、または一部のアイテム（web.page.get、net.tcp.port など）を実行する際の接続に使用する送信元 IP アドレスです。
STARTAGENTS	パッシブチェックを処理する zabbix_agentd の事前フォーク済みインスタンス数です。 0 に設定すると、パッシブチェックは無効になり、エージェントはどの TCP ポートでも待ち受けしません。
STARTUPTYPE	Zabbix agent サービスの起動種類です。指定可能な値: **automatic** - Windows 起動時にサービスを自動的に開始します。 **delayed** - (既定) 自動起動サービスの起動完了後にサービスの開始を遅延します（Windows Vista/Server 2008 以降で利用可能）。 **manual** - サービスを手動で開始します（ユーザーまたはアプリケーションによる）。 **disabled** - サービスを無効にし、ユーザーまたはアプリケーションが開始できないようにします。 例: STARTUPTYPE=disabled
STATUSPORT	Zabbix agent 2 のみ。設定されている場合、エージェントは HTTP ステータス要求 (http://localhost:<port>/status) をこのポートで待ち受けます。
TIMEOUT	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。
TLSACCEPT	受け入れる受信接続です（パッシブチェックで使用します）。psk に設定すると、TLSCONNECT も psk に設定されます（別途指定されていない場合）。
TLSCAFILE	ピア証明書検証用の上位 CA 証明書を含むファイルへの完全なパス名です。
TLSCERTFILE	エージェント証明書または証明書チェーンを含むファイルへの完全なパス名です。
TLSCONNECT	Zabbix サーバーまたはプロキシへエージェントがどのように接続するかです（アクティブチェックで使用します）。psk に設定すると、TLSACCEPT も psk に設定されます（別途指定されていない場合）。
TLSCRLFILE	失効した証明書を含むファイルへの完全なパス名です。
TLSKEYFILE	Zabbix agent の秘密鍵を含むファイルへの完全なパス名です。
TLSPSKFILE	Zabbix agent の事前共有鍵を含むファイルへの完全なパス名です。TLSPSKFILE と TLSPSKVALUE の両方が設定されている場合、TLSPSKVALUE の値は TLSPSKFILE で指定されたファイルに書き込まれます。事前共有鍵ファイルへのアクセスは、ファイルのセキュリティ設定を調整して Zabbix agent (またはエージェントを実行するユーザー) のみが読み取れるように制限することを推奨します。
TLSPSKIDENTITY	事前共有鍵の識別文字列です。
TLSPSKVALUE	事前共有鍵の文字列値です。TLSPSKFILE と TLSPSKVALUE の両方が設定されている場合、TLSPSKVALUE の値は TLSPSKFILE で指定されたファイルに書き込まれます。 例: TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
TLSSERVERCERTISSUER	許可されるサーバー（プロキシ）証明書の発行者です。
TLSSERVERCERTSUBJECT	許可されるサーバー（プロキシ）証明書のサブジェクトです。
UNSAFEUSERPARAMETERS	ユーザー定義パラメーターの引数にすべての文字を渡せるようにします。

例

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

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
```

```
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

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

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

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

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

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

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

Zabbix エージェント 2 のロード可能なプラグイン

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

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

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

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

セットアップウィザードからのインストール

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

2. エンドユーザーライセンス契約に同意します。

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

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

4. インストールをクリックしてインストールを開始します。

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

5. 完了ボタンをクリックしてセットアップウィザードを終了します。

コマンドラインからのインストール

Zabbix エージェント 2 のロード可能なプラグインは、msiexec を使用して MSI インストーラーを実行することで、コマンドラインからインストールできます。例:

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

Zabbix エージェント 2 のロード可能なプラグイン MSI インストーラーパッケージは、次のパラメーターをサポートしています。

Parameter	Description
ADDDEFAULT	既定の構成でインストールする、カンマ区切りのコンポーネント一覧です。詳細は、ADDDEFAULT property を参照してください。 指定可能な値: ALL, CephPlugin, EmberplusPlugin, MongodbPlugin, MssqlPlugin, NvidiagpuPlugin, PostgresqlPlugin 例: ADDDEFAULT=MongodbPlugin,PostgresqlPlugin
ADDLOCAL	ローカルにインストールする、カンマ区切りのコンポーネント一覧です。詳細は、ADDLOCAL property を参照してください。 指定可能な値: ALL, CephPlugin, EmberplusPlugin, MongodbPlugin, MssqlPlugin, NvidiagpuPlugin, PostgresqlPlugin 例: ADDLOCAL=MongodbPlugin,MssqlPlugin
INSTALLFOLDER	Zabbix コンポーネントをインストールするフォルダの完全なパス名です。設定ファイルは zabbix_agent2.d サブフォルダにインストールされます。 例: INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"

Windows での Zabbix agent 2 のビルド

概要

このページでは、Windows 10（64 ビットまたは 32 ビット）上でソースから Zabbix エージェント 2 をビルドする方法を説明します。

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

Zabbix エージェント 2 のビルドには、以下が必要です。

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

以下のいずれかの方法で Zabbix エージェント 2 をビルドできます。

- vcpkg を使用する - C++ パッケージマネージャーを使用して依存関係の管理を簡素化する自動化された方法です。
- 手動でビルドする - エージェントをコンパイルする前に、すべての依存関係をインストールする必要がある手動の方法です。

Attention:
ビルドプロセスを開始する前に、以下の点に注意してください。

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

vcpkg を使用した Zabbix エージェント 2 のビルド

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

1. Build Tools for Visual Studio 2022 をダウンロードしてインストールします。インストール中に、vcpkg パッケージマネージャーを含む C++ によるデスクトップ開発ワークロードを必ず選択してください。

2. Go をダウンロードしてインストールします（MSI インストーラーを利用できます）。インストール中に、インストール先として C:\Zabbix\Go を必ず指定してください。

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

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

Attention:
MinGW のバージョンは、最小要件を満たしている必要があります。

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

4. vcpkg を初期化し、Zabbix エージェント 2 のビルドに必要な依存関係をインストールします（完了までに時間がかかる場合があります）。

```
cd C:\Zabbix

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

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

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

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

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

- 64 ビットビルドの場合:

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

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

:: Crypt32 ライブラリのリンカーフラグを設定します:
SET CGO_LDFLAGS="-lCrypt32"

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

- 32 ビットビルドの場合:

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

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

:: Crypt32 ライブラリのリンカーフラグを設定します:
SET CGO_LDFLAGS="-lCrypt32"

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

7. 次のスクリプトを実行して Zabbix エージェント 2 をコンパイルします。

```
build.bat
```

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

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

```
mkdir C:\Zabbix\agent2

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

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

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

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

必要に応じて、Zabbix エージェント 2 のロード可能モジュールのコンパイルを続行します。

Zabbix エージェント 2 のロード可能なプラグインのコンパイル

1. ご利用の Zabbix エージェント 2 のバージョンに一致する Zabbix plugin source をダウンロードし、C:\Zabbix に展開します。

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

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

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

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

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

コンパイル後、Zabbix エージェント 2 のプラグインバイナリとその設定ファイルは、同じプラグインディレクトリに配置されます。

プラグインの実行ファイルは、Zabbix エージェント 2 から読み込める限り、任意の場所に配置できます。たとえばPostgreSQL プラグインの場合は、プラグイン設定ファイル内でプラグインバイナリへのパスを指定します。

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

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

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

プラグインの設定方法の詳細については、セットアップ を参照してください。

Zabbix agent 2 の手動ビルド

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

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

ビルドツールのセットアップ

1. MSYS2 をダウンロードしてインストールします（MSI インストーラーとして提供されています）。
インストール時には、インストール先フォルダーとして C:\Zabbix\msys64 を指定してください。

2. Go をダウンロードしてインストールします（MSI インストーラーとして提供されています。現在サポートされているGo のバージョン を参照してください）。
インストール時には、インストール先フォルダーとして C:\Zabbix\Go を指定してください。

3. Microsoft Visual C ランタイムライブラリを使用する MinGW ディストリビューション をダウンロードします。たとえば、次のとおりです。

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

Attention:
MinGW のバージョンは 最小要件 を満たしている必要があります。

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

OpenSSL のインストール

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

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

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

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

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

- 64 ビットビルドの場合:

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

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

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

make
make install
```

- 32 ビットビルドの場合:

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

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

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

make
make install
```

Attention:
管理者以外のユーザーから C:\Zabbix\x64\OpenSSL-Win64-350-static ディレクトリへの書き込みアクセスを必ず取り消してください。そうしないと、エージェントは権限のないユーザーが変更できるパスから SSL 設定を読み込むことになり、セキュリティ上の脆弱性につながる可能性があります。

- no-shared オプションを指定すると、libcrypto.lib と libssl.lib の OpenSSL 静的ライブラリが自己完結型になるため、Zabbix バイナリに OpenSSL が含まれ、外部 DLL が不要になります。これにより、OpenSSL ライブラリなしで Zabbix バイナリを他の Windows マシンにコピーできます。ただし、新しい OpenSSL のバグ修正版がリリースされた場合は、Zabbix エージェントを再コンパイルする必要があります。
- no-shared オプションを指定しない場合、Zabbix は実行時に OpenSSL DLL に依存します。この場合、OpenSSL の更新時に Zabbix エージェントを再コンパイルする必要がない可能性があります。ただし、他のマシンにコピーする際には、OpenSSL DLL もコピーする必要があります。

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

4. スクリプトを実行して OpenSSL を設定およびインストールします（完了までに時間がかかる場合があります）。

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

PCRE2 のインストール

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

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

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

3. PCRE2 を構成します:

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

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

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

4. PCRE2 をインストールします:

```
make install
```

Zabbix エージェント 2 のコンパイル

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

raw ソースリポジトリからソースアーカイブを生成する必要がある場合（カスタムパッチを適用する場合や、最新のソースコードからビルドする場合など）は、Go（Zabbix エージェント 2 の設定に必要）がインストールされた **Linux** マシンで次のコマンドを実行します。

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

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

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

- 64 ビットビルドの場合:

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

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

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

- 32 ビットビルドの場合:

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

```
#### TLS サポートあり:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win32-350-static"

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

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

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

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

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

必要に応じて、Zabbix エージェント 2 のロード可能プラグインのコンパイルに進みます。

Windows での Zabbix エージェントのビルド

概要

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

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

Zabbix エージェントのビルドには、以下が必要です。

- C コンパイラー（Visual Studio 2022 用ビルドツールに含まれています）
- OpenSSL（Zabbix の暗号化機能用）
- PCRE2（Perl 互換正規表現。Zabbix の正規表現パターンマッチング機能用）

Zabbix エージェントは、以下のいずれかの方法でビルドできます。

- vcpkg を使用する - C++ パッケージマネージャーを使用して依存関係の管理を簡素化する自動化された方法です。
- 手動でビルドする - エージェントをコンパイルする前に、すべての依存関係をインストールする必要がある手動の方法です。

監視の要件によっては、追加のライブラリが必要になる場合があります。詳細については、要件を参照してください。

> **Attention:**
> ビルドプロセスを開始する前に、以下の点に注意してください。

> - コマンドを実行するには、十分な権限を持つユーザーで起動した「x64 Native Tools Command Prompt」(Visual Studio 2022 用ビルドツールに含まれています）を使用してください。保護されたフォルダーへの書き込み権限が必要です。
> - すべてのソースファイルとビルドフォルダー用に、C:\Zabbix に作業ディレクトリを作成することを推奨します。ただし、コンパイルされたコンポーネントは C:\Program Files\Zabbix\x64 にインストールしてください。

vcpkg を使用した Zabbix エージェントのビルド

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

1. Build Tools for Visual Studio 2022をダウンロードしてインストールします。インストール中に、*Desktop development with C++* ワークロードを選択してください。このワークロードには、vcpkg を使用してエージェントをビルドするために必要なツールが含まれています。

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

2. 管理者権限で cmd.exe を開き、vcpkg を初期化して、Zabbix エージェントのビルドに必要な依存関係をインストールします（完了までに時間がかかる場合があります）。

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

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

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

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

:: ビルドプロセスを実行:
nmake -f Makefile CPU=AMD64 ^
    PCRE2INCDIR="%vcpkg%\include" ^
    PCRE2LIBDIR="%vcpkg%\lib" ^
    TLS=openssl ^
    TLSINCDIR="%vcpkg%\include" ^
    TLSLIBDIR="%vcpkg%\lib" ^
    LIBS="$(LIBS) Crypt32.lib" ^
    all
```

5. 「x64 Native Tools Command Prompt」を開き、次のスクリプトを実行して Zabbix エージェントをコンパイルします。

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

別のコマンドプロンプトを開いた場合は、最初に Visual Studio のビルド環境を初期化します。

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

スペースを含むパスは引用符で囲んでください。

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

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

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

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

Zabbix エージェントの手動ビルド

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

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

ビルドツールのインストール

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

- C コンパイラー（Microsoft Visual C++）
- NMake コマンドラインツール
- 「x64 Native Tools Command Prompt」

OpenSSL のインストール

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

1. Strawberry Perl（MSI インストーラーとして提供されています）をダウンロードしてインストールします。インストール中に、インストール先フォルダーとして C:\Zabbix\Strawberry を指定してください。

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

```
cpanm Text::Template
```

3. Strawberry Perl のインストール中に、Netwide Assembler（NASM、OpenSSL のコンパイルに必要）がコンパイルされたことを確認します。

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

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

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

5. 展開したディレクトリに移動し、次の例のように OpenSSL を設定します。

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

Attention:
Windows で Zabbix エージェントをコンパイルする際に OpenSSL 用のカスタムディレクトリ（例: C:\zabbix または C:\openssl-64bit）を選択する場合は、管理者以外のユーザーからこのディレクトリへの書き込みアクセスを必ず取り消してください。そうしないと、エージェントが権限のないユーザーによって変更可能なパスから SSL 設定を読み込むため、セキュリティ上の脆弱性が発生する可能性があります。

- no-shared オプションを指定すると、libcrypto.lib と libssl.lib が自己完結型の OpenSSL 静的ライブラリになるため、Zabbix バイナリに OpenSSL が含まれ、外部 DLL が不要になります。これにより、OpenSSL ライブラリがインストールされていない他の Windows マシンに Zabbix バイナリをコピーできます。ただし、新しい OpenSSL のバグ修正版がリリースされた場合は、Zabbix エージェントを再コンパイルする必要があります。
- no-shared オプションを指定しない場合、Zabbix は実行時に OpenSSL DLL に依存します。この場合、OpenSSL の更新時に Zabbix エージェントを再コンパイルする必要がない可能性があります。ただし、他のマシンにコピーする際には、OpenSSL DLL もコピーする必要があります。

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

6. OpenSSL をコンパイルしてテストを実行します（完了までに時間がかかる場合があります）。

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

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

7. OpenSSL をインストールします。

```
nmake install
```

ソフトウェアコンポーネント（ライブラリ、ヘッダーファイル。ただしドキュメントは除く）のみをインストールするには、nmake install_sw を使用できます。

PCRE2 のインストール

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

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

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

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

4. PCRE2 を構成します。

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

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

5. NMake を使用して PCRE2 をビルドします。

```
nmake
```

6. PCRE2 をインストールします。

```
cmake --install .
```

Zabbix エージェントのコンパイル

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

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

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

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

2. Zabbix のビルドディレクトリに移動し、「x64 Native Tools Command Prompt」を開いて、Zabbix エージェント（またはその他のコンポーネント）をコンパイルします。OpenSSL と PCRE2 がインストールされているディレクトリを正しく指定してください。

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

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

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

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

スペースを含むパスは引用符で囲んでください。

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

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

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

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

PKG からの **macOS** エージェントのインストール

概要

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

Zabbix エージェントのパッケージは、暗号化 の有無を選択できます。

エージェントのインストール

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

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

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

エージェントの実行

エージェントはインストールまたは再起動後に自動的に起動します。

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

エージェントを手動で起動するには、次のコマンドを実行します:

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

エージェントを手動で停止するには:

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

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

トラブルシューティングとエージェントの削除

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

Zabbix エージェントが実行中かどうかを確認します:

```
ps aux | grep zabbix_agentd
```

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

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

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

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

launchctl で起動した Zabbix エージェントを停止します:

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

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

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

Zabbix エージェントがインストールされたことを忘れさせます:

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

macOS での Zabbix エージェントのビルド

概要

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

前提条件

コマンドライン開発ツール（Xcode は不要）、Automake、pkg-config、PCRE（v8.x）または PCRE2（v10.x）が必要です。TLS 対応のエージェントバイナリをビルドする場合は、OpenSSL または GnuTLS も必要です。

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

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

次に、Automake と pkg-config をインストールします。

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

PCRE、OpenSSL、GnuTLS ライブラリの準備方法は、それらをエージェントにどのようにリンクするかによって異なります。

これらのライブラリがすでに存在する macOS マシンでエージェントバイナリを実行する予定であれば、Homebrew が提供するプリコンパイル済みライブラリを使用できます。これは通常、Zabbix エージェントバイナリのビルドやその他の目的で Homebrew を使用している macOS マシンです。

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

共有ライブラリを使用したエージェントバイナリのビルド

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

```
brew install pcre2
```

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

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

Zabbix ソースをダウンロード。

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

TLS なしでエージェントをビルド。

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

OpenSSL を使用してエージェントをビルド。

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

GnuTLS を使用してエージェントをビルド。

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

TLS を使用しない静的ライブラリを使用したエージェントバイナリのビルド

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

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

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

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

Zabbix ソースとビルドエージェントをダウンロードします。

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

OpenSSL と静的ライブラリを使用してエージェントバイナリをビルドする

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

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

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

static-libs-source で静的ライブラリをビルドします。

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

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

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

OpenSSL をダウンロードしてビルドする。

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

Zabbix ソースとビルドエージェントをダウンロードする。

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

GnuTLS と静的ライブラリを使用してエージェントバイナリをビルドする

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

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

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

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

`static-libs-source` で静的ライブラリをビルドします。

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

Nettle をダウンロードしてビルドします。

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

GnuTLS をダウンロードしてビルドします。

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

Zabbix ソースとビルドエージェントをダウンロードをします。

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

Zabbix サーバーの AWS デプロイガイド

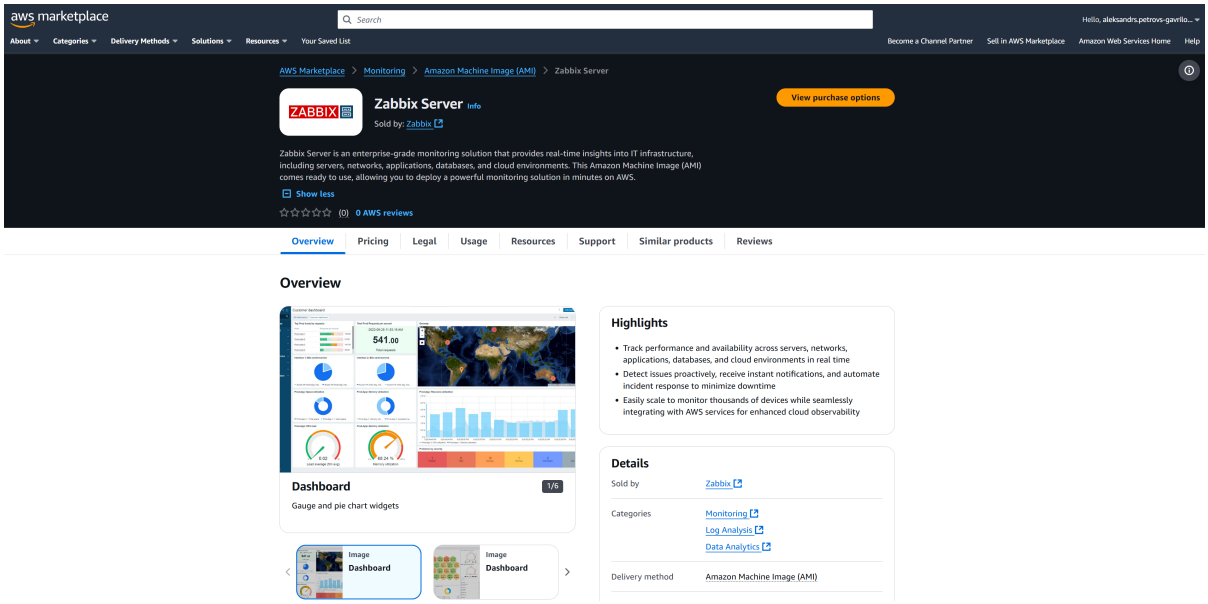
概要

このページでは、AWS を使用して Zabbix サーバーをデプロイする方法について説明します。

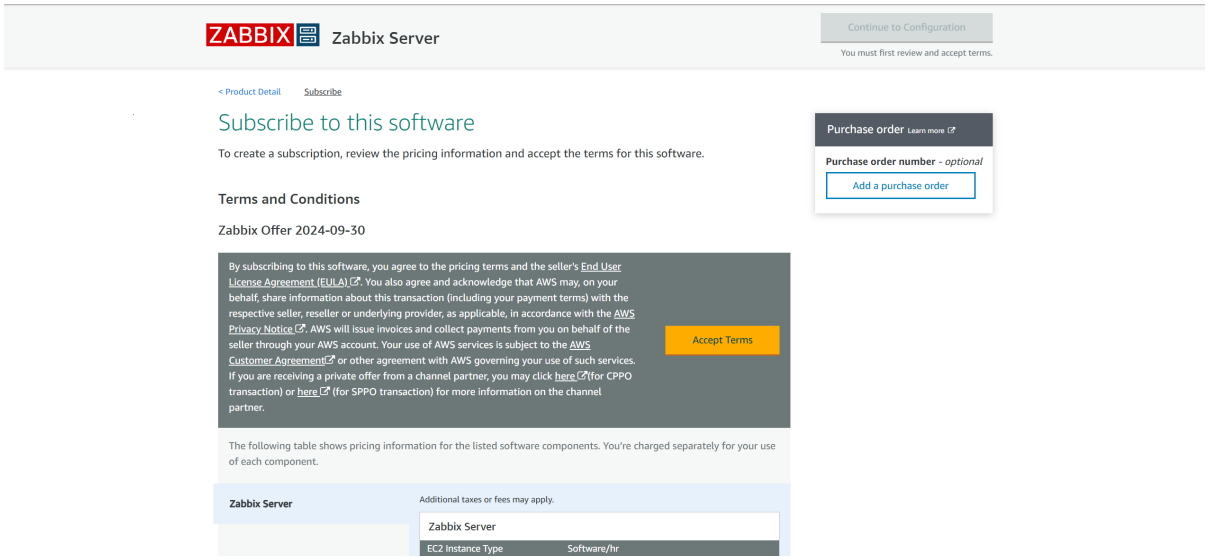
> **Attention:**
> このプラットフォームでは、複数の Zabbix バージョンをデプロイできます。デプロイした Zabbix と同じバージョンのドキュメントを必ずお読みください。異なる Zabbix バージョンのドキュメントは、バージョンのドロップダウンメニューから選択できます。

アプライアンスのデプロイ

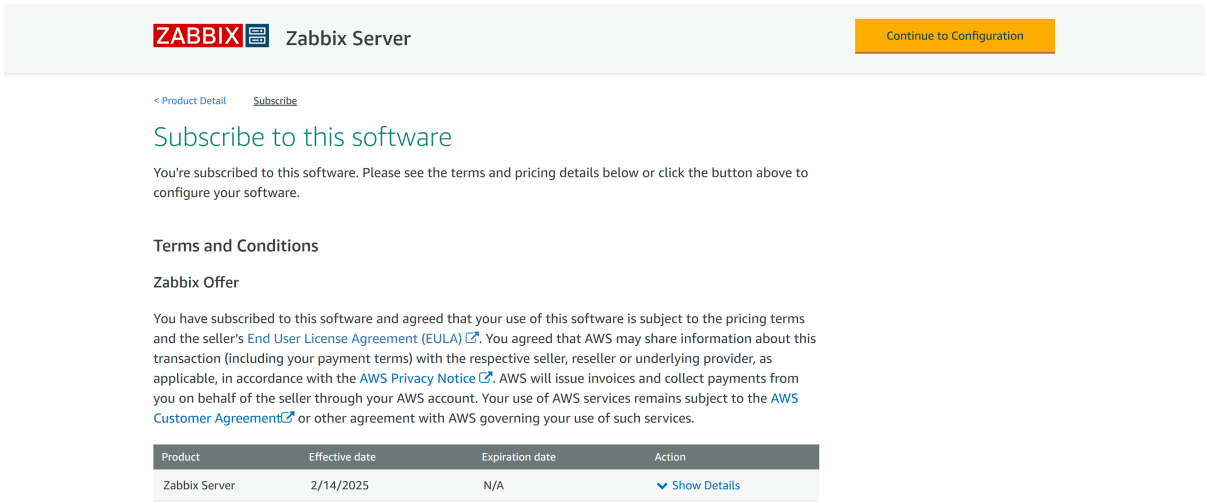
Zabbix アプライアンスをデプロイするには、まずZabbix AWS Marketplace ページに移動します。次に、*View purchase options* をクリックしてイメージをデプロイし、サブスクリプションページに進みます。



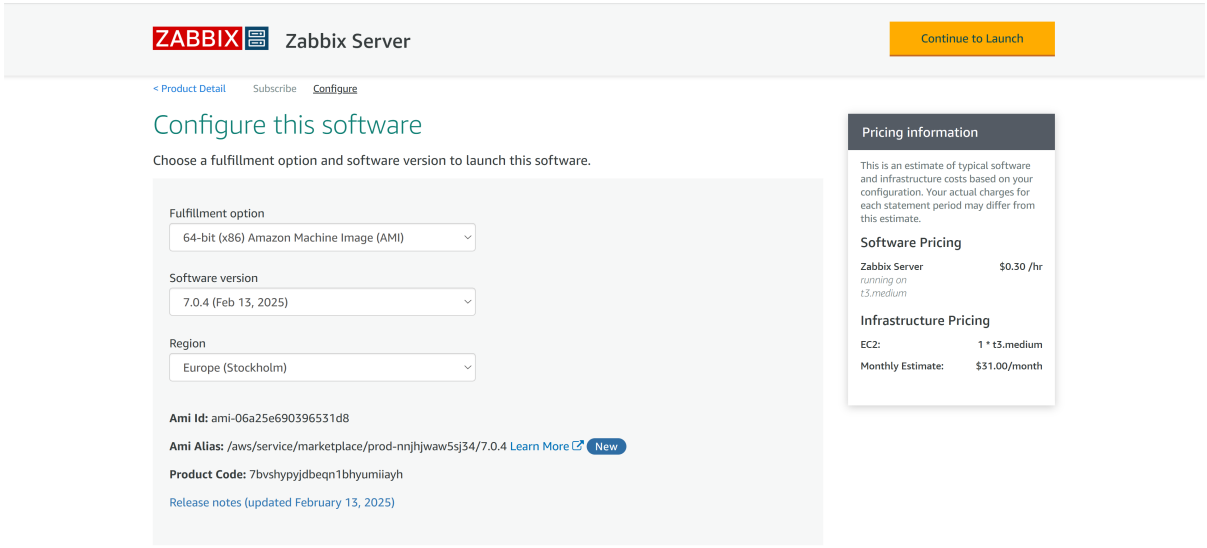
Accept terms ボタンをクリックして、*Terms and Conditions* に同意します。



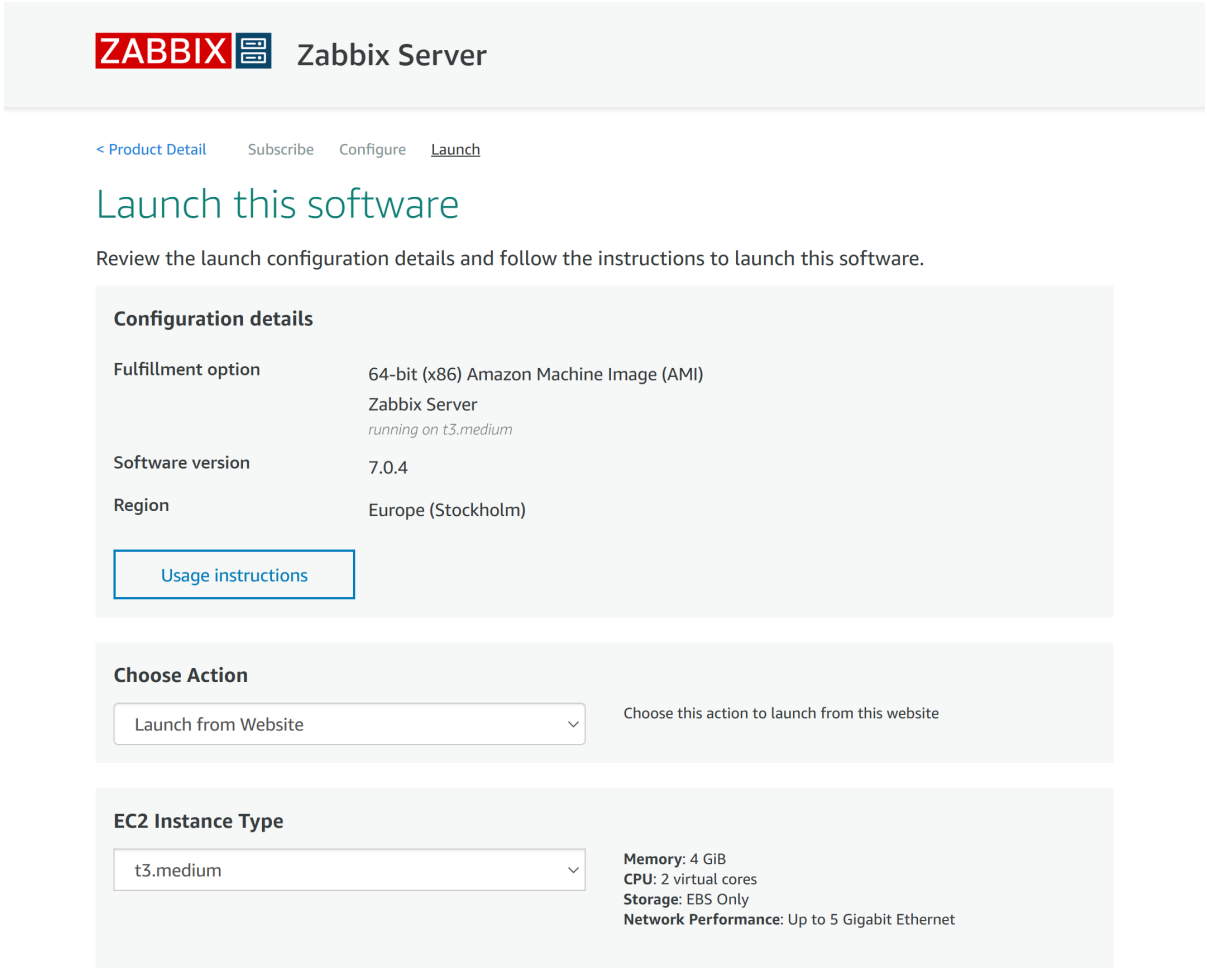
Continue to Configuration ボタンをクリックして、Zabbix アプライアンスの Configuration セクションに移動します。



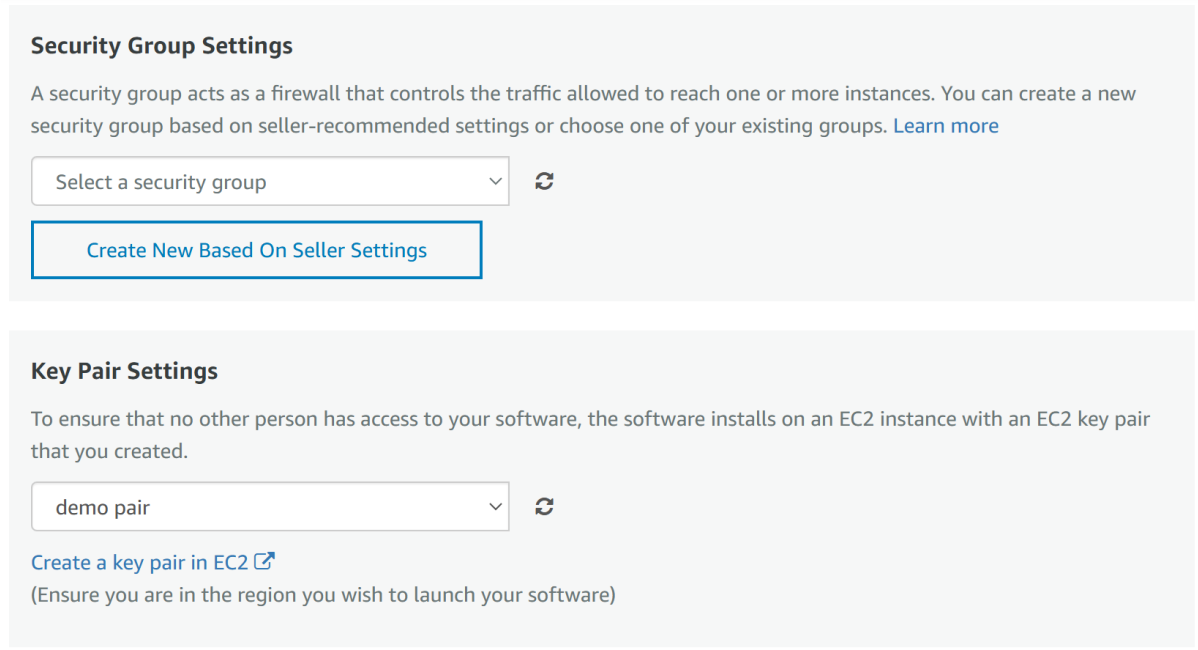
Fulfilment オプション、Zabbix のバージョン、および希望するリージョンを選択します。



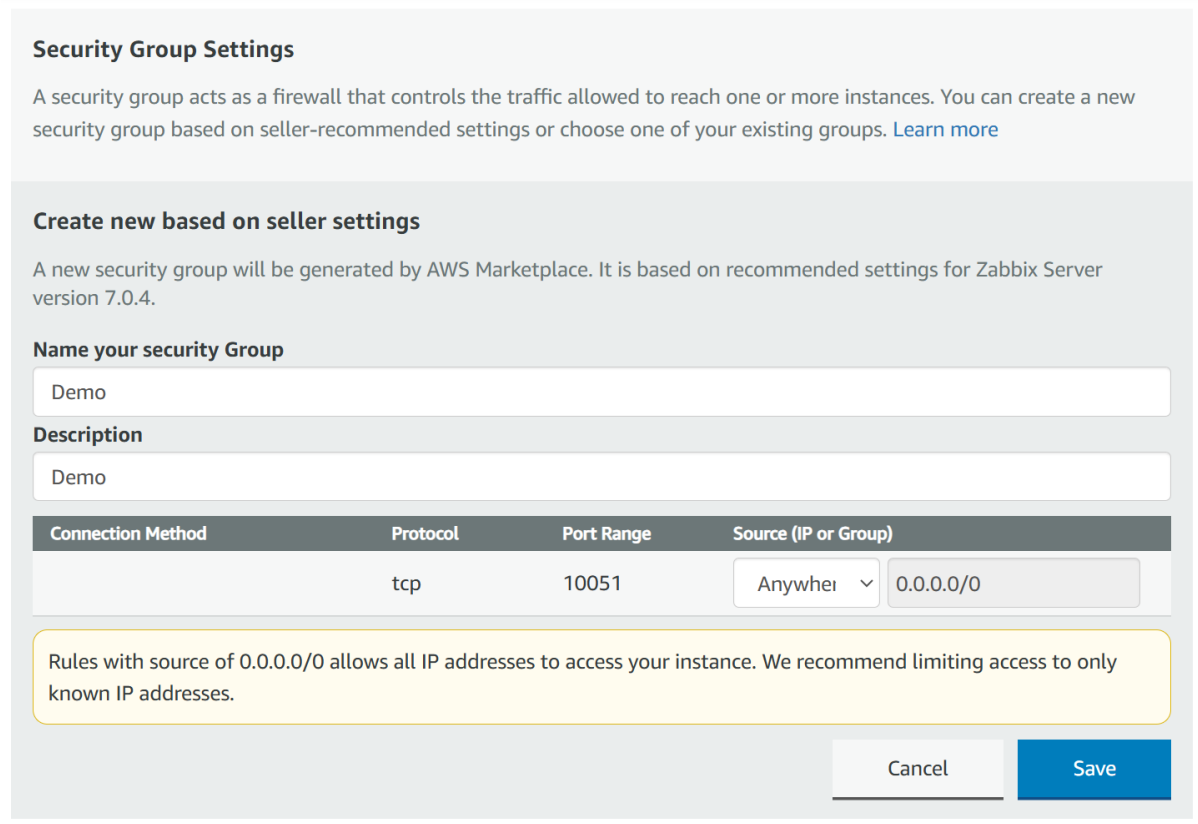
Continue to Launch ボタンをクリックします。Launch ページで、希望する Launch オプション、EC2 インスタンスタイプ、VPC、およびサブネットの設定を選択します。



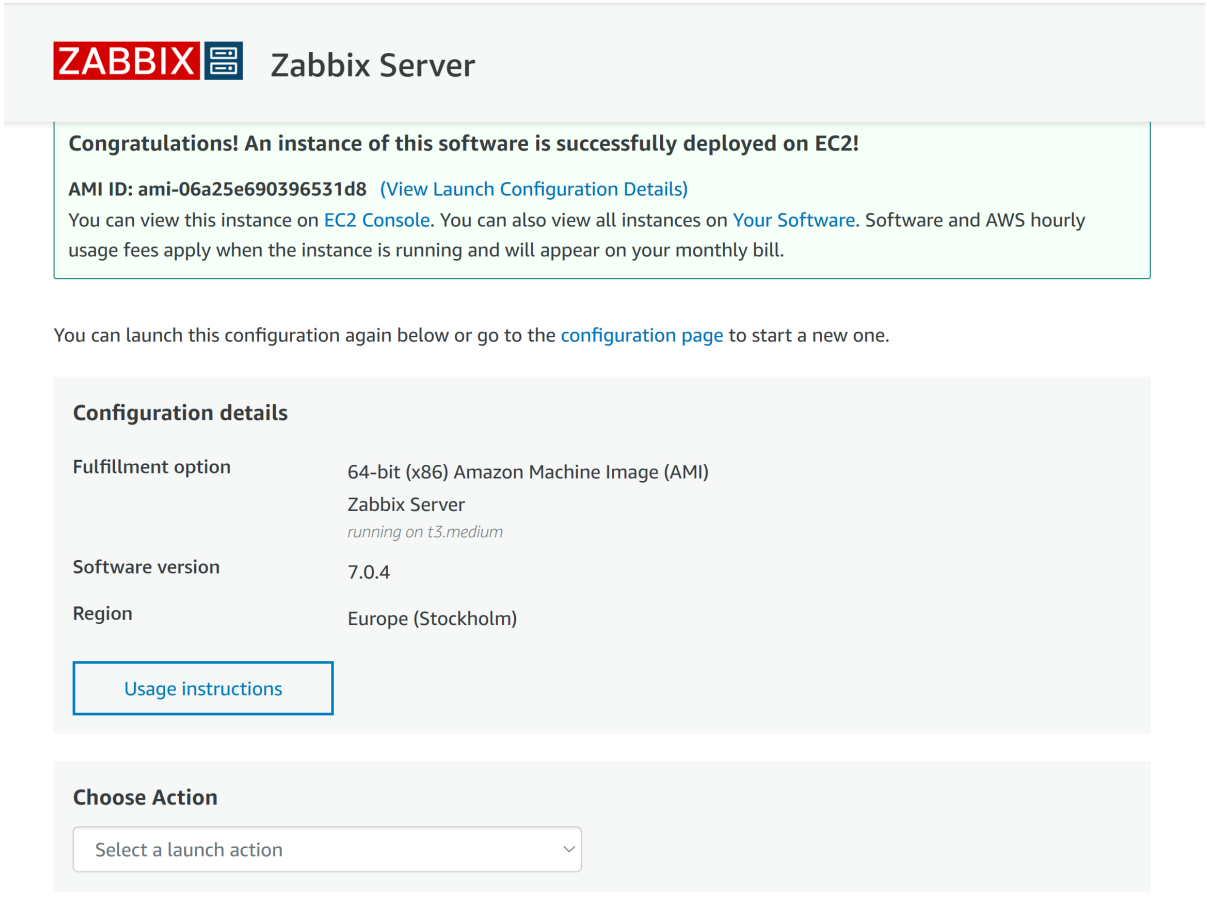
Launch ページの下部で、セキュリティグループの設定を指定し、EC2 キーペアを選択または作成します。秘密鍵は安全な場所に保存してください。



Create a new Security Group based on seller settings を選択すると、TCP ポート 10051（Zabbix サーバー）が許可されます。注: SSH および HTTP/HTTPS（Web インターフェース）経由でインスタンスにアクセスする場合は、セキュリティグループの設定で SSH および HTTP/HTTPS 接続も許可する必要があります。

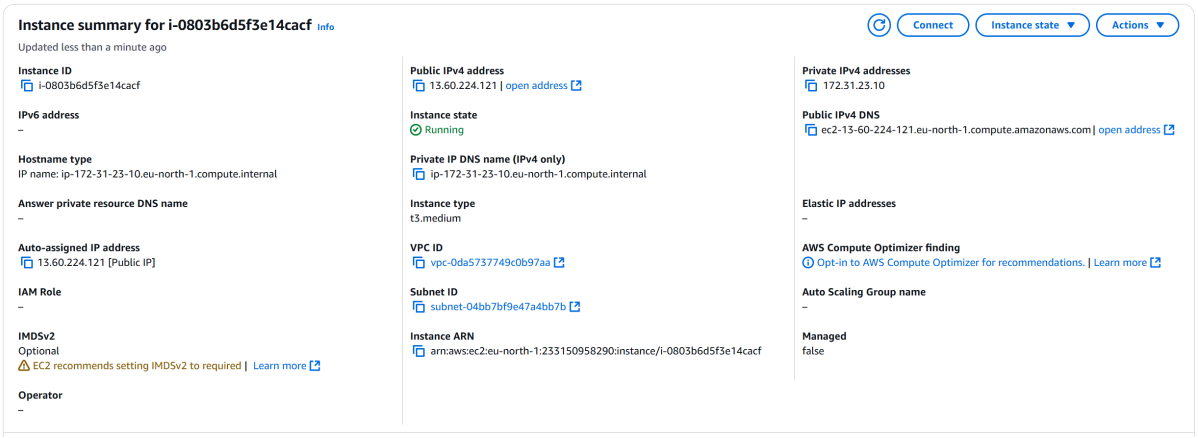


これで、Zabbix アプライアンスのデプロイが正常に完了しました。

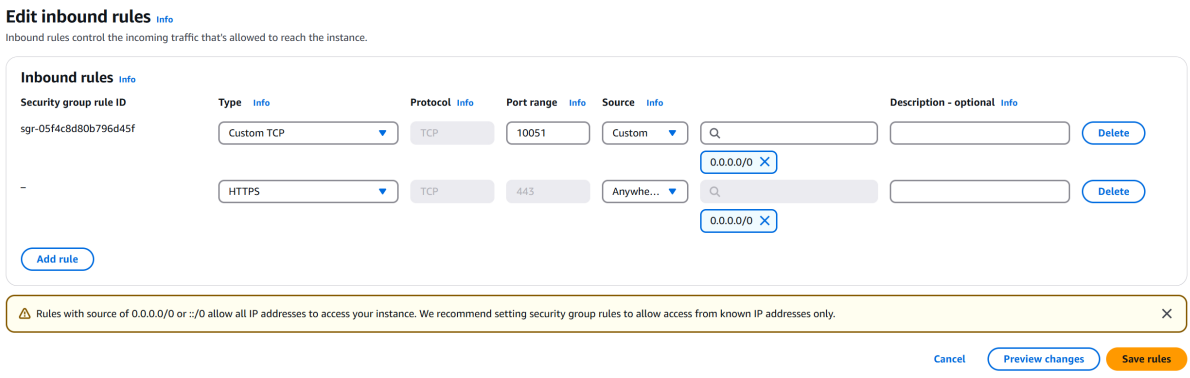


アプライアンスへのアクセス

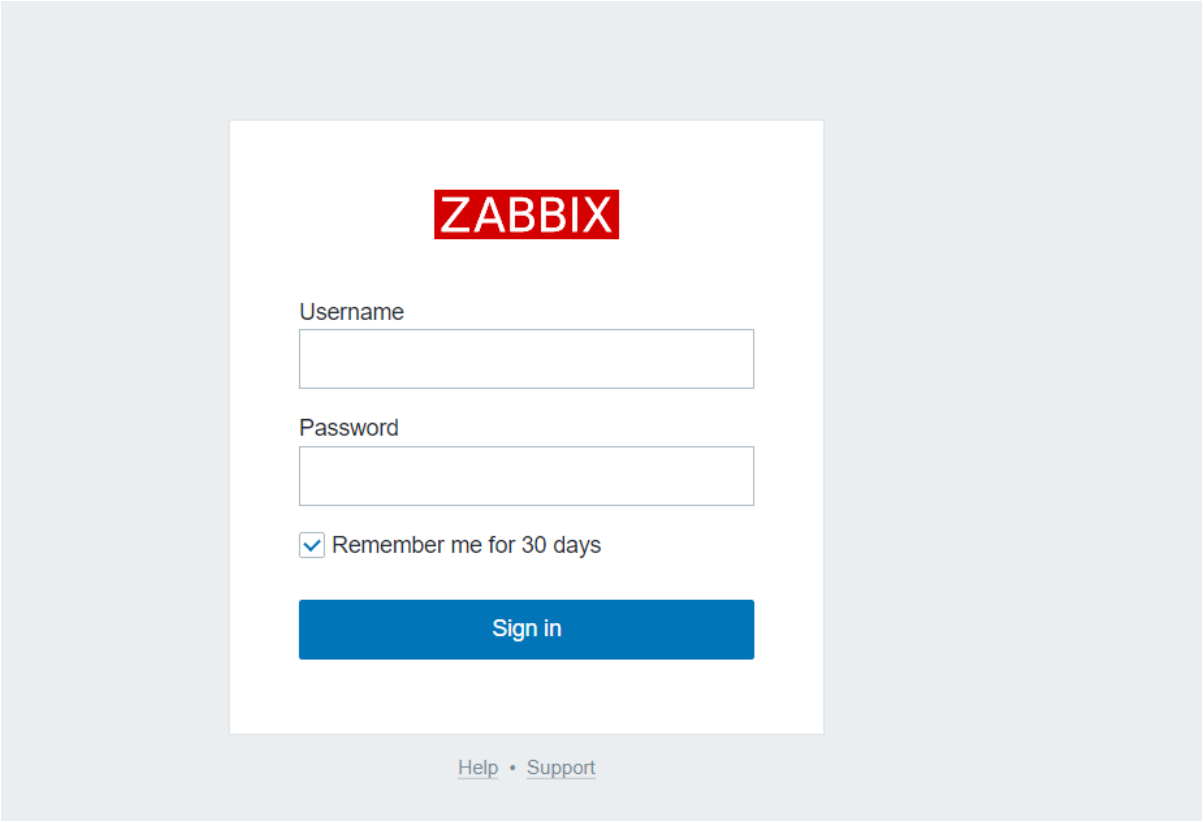
Instances セクションを開き、新しくデプロイした Zabbix アプライアンスのインスタンスを選択します。



インスタンスの Public IPv4 アドレスまたは Public IPv4 DNS を使用して Zabbix の Web インターフェースにアクセスするには、Security Group でポート 443 の受信 HTTPS トラフィックを許可します。



これで、Zabbix アプライアンスの Public IPv4 アドレスまたは Public IPv4 DNS を開いて、Zabbix の Web インターフェースにアクセスできます。



パスワードを取得するには、シリアルコンソールまたは SSH を使用してマシンに接続する必要があります。デフォルトの SSH ユーザー名は rocky です。以前に選択または生成したキーペアを使用してください。認証情報の生成には時間がかかる場合があります。認証情報の生成には数分かかることがあります。接続を試みる前に、数分待ってください。

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

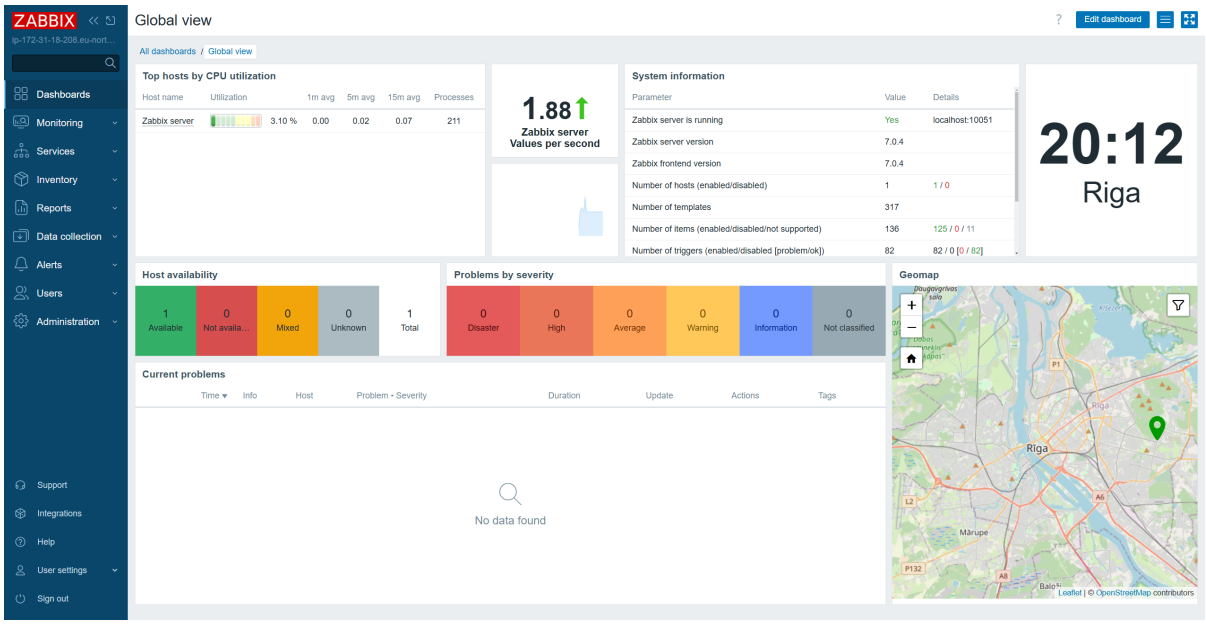
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

********************************************************************************
```

これで、この情報を使用して Zabbix の Web インターフェースにログインできます。



データベースへのアクセス

パスワードを入力せずに、root ユーザーを使用して Zabbix データベースにアクセスできます。

```
[root@ip-172-31-29-138 ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

認証情報は /root/.my.cnf 設定ファイルに保存されています。

デフォルトの Zabbix Web インターフェースのタイムゾーンは、管理 > 一般設定 > *GUI* で変更できます。

Zabbix プロキシの **AWS** デプロイガイド

概要

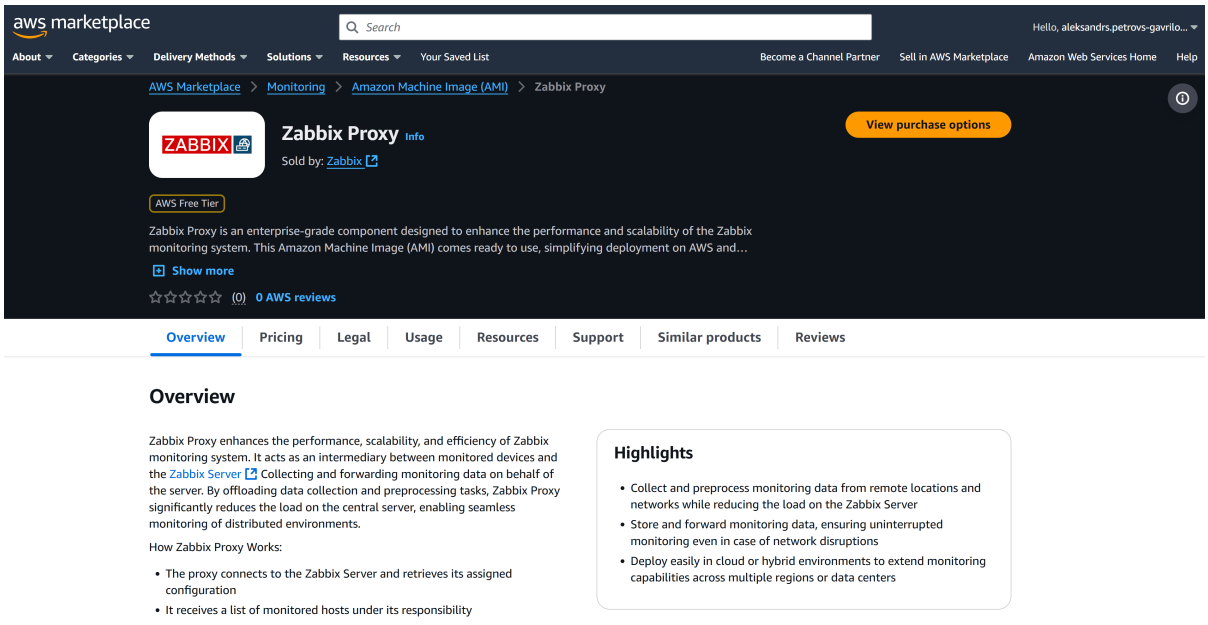
このページでは、AWS を使用して Zabbix プロキシをデプロイする方法について説明します。

> **Attention:**
> プラットフォームには、複数の Zabbix バージョンをデプロイできます。デプロイした Zabbix と同じバージョンのドキュメントを必ず参照してください。異なる Zabbix バージョンのドキュメントは、バージョンのドロップダウンメニューから選択できます。

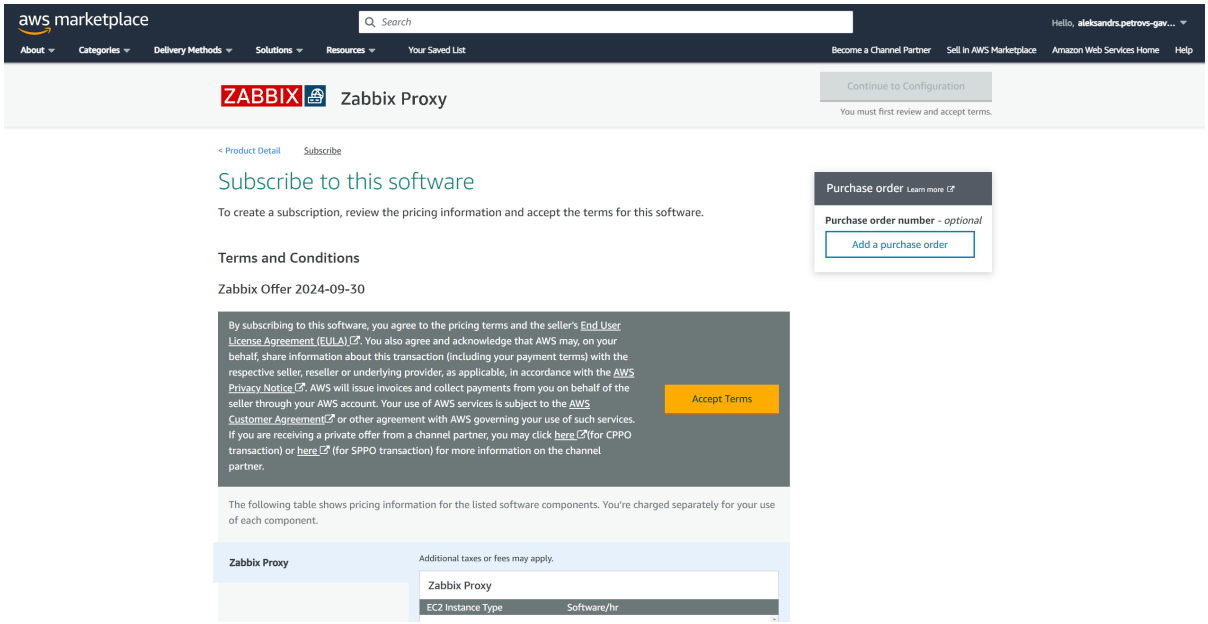
Zabbix プロキシのデプロイ

> **Attention:**
> Zabbix プロキシコンポーネントを使用するには、稼働中の Zabbix サーバーが必要です。

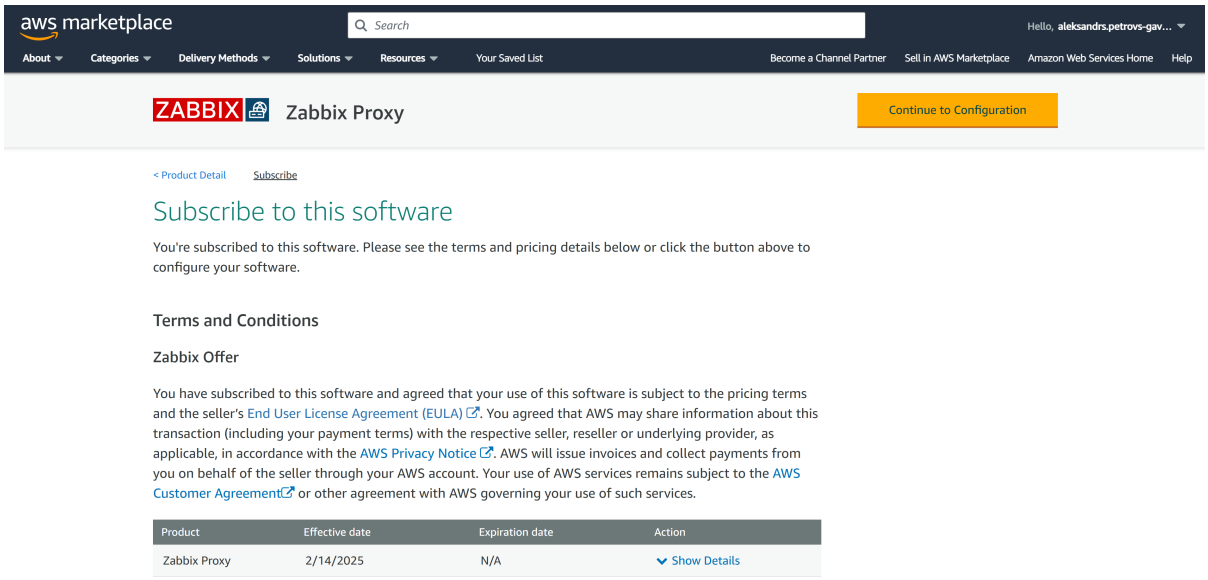
Zabbix プロキシをデプロイするには、まずZabbix AWS マーケットプレイスページに移動します。次に、*View purchase options* をクリックしてイメージをデプロイし、サブスクリプションページに進みます。



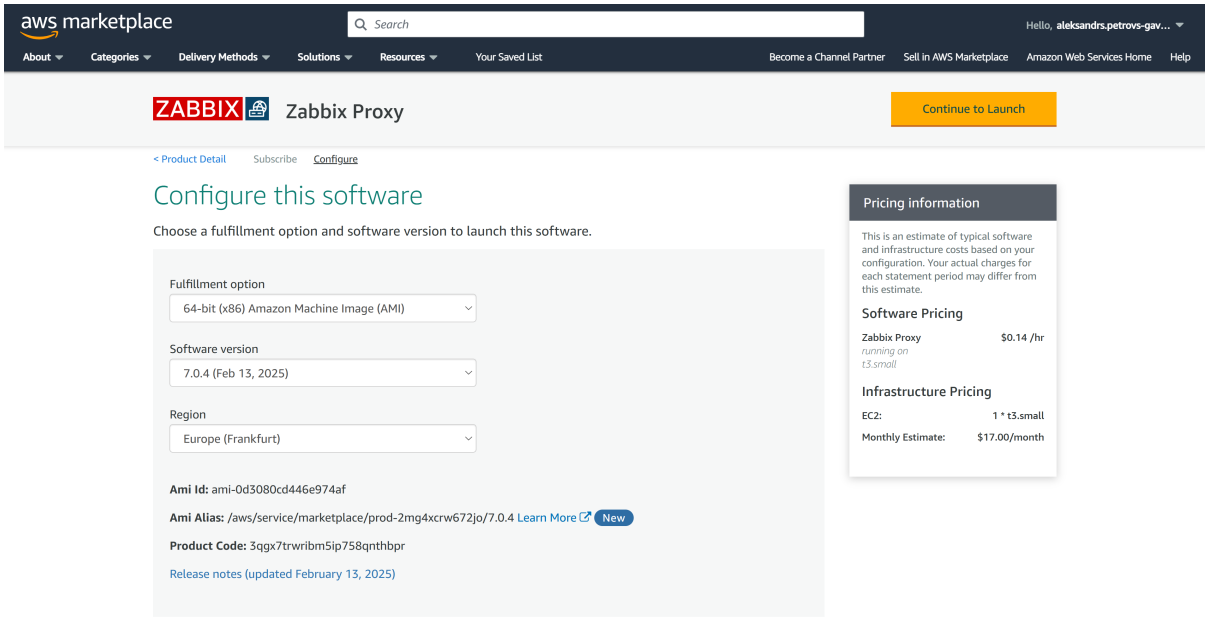
Terms and Conditions に同意するには、*Accept terms* ボタンをクリックします。



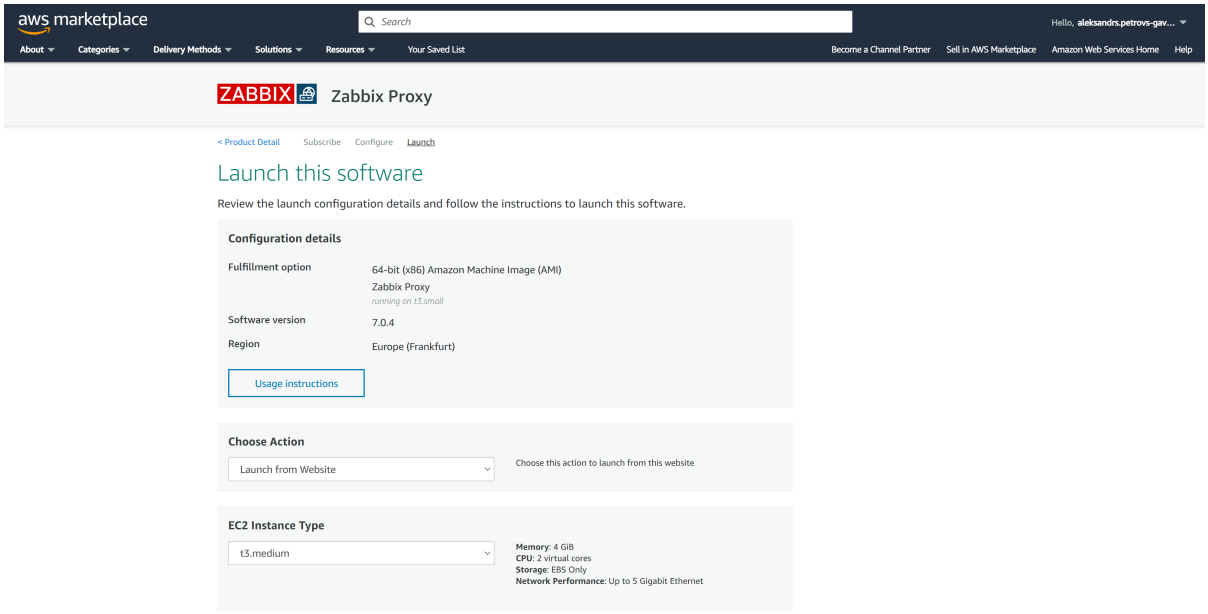
Continue to Configuration ボタンをクリックして、Zabbix プロキシの Configuration セクションに移動します。



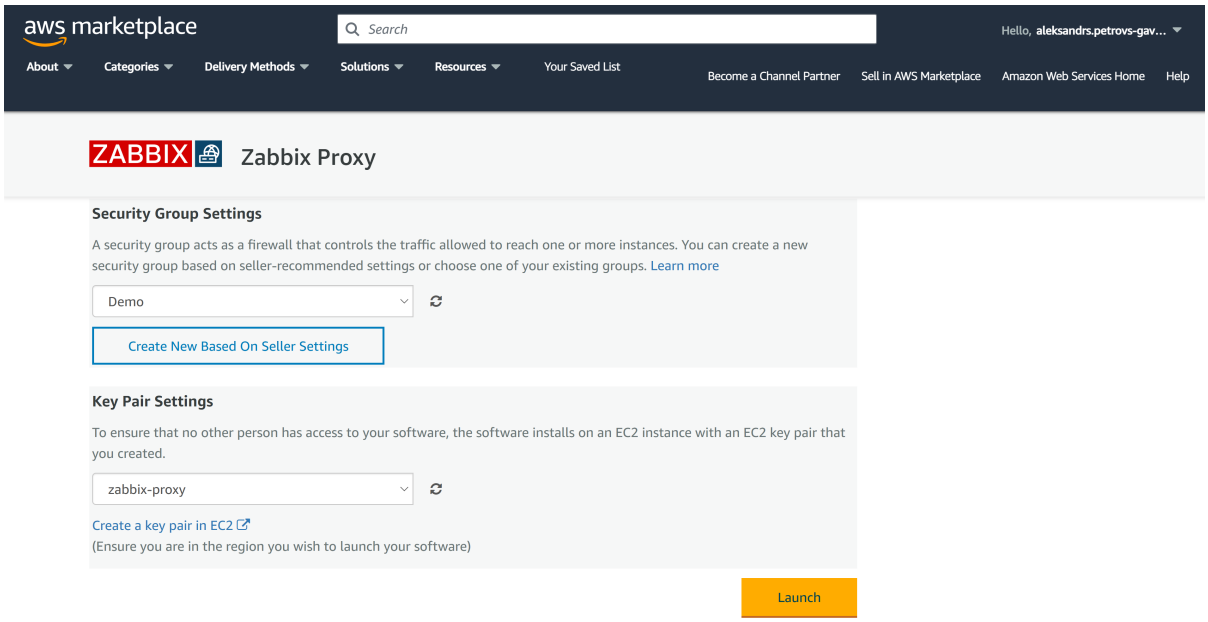
Fulfilment オプション、Zabbix のバージョン、および希望するリージョンを選択します。



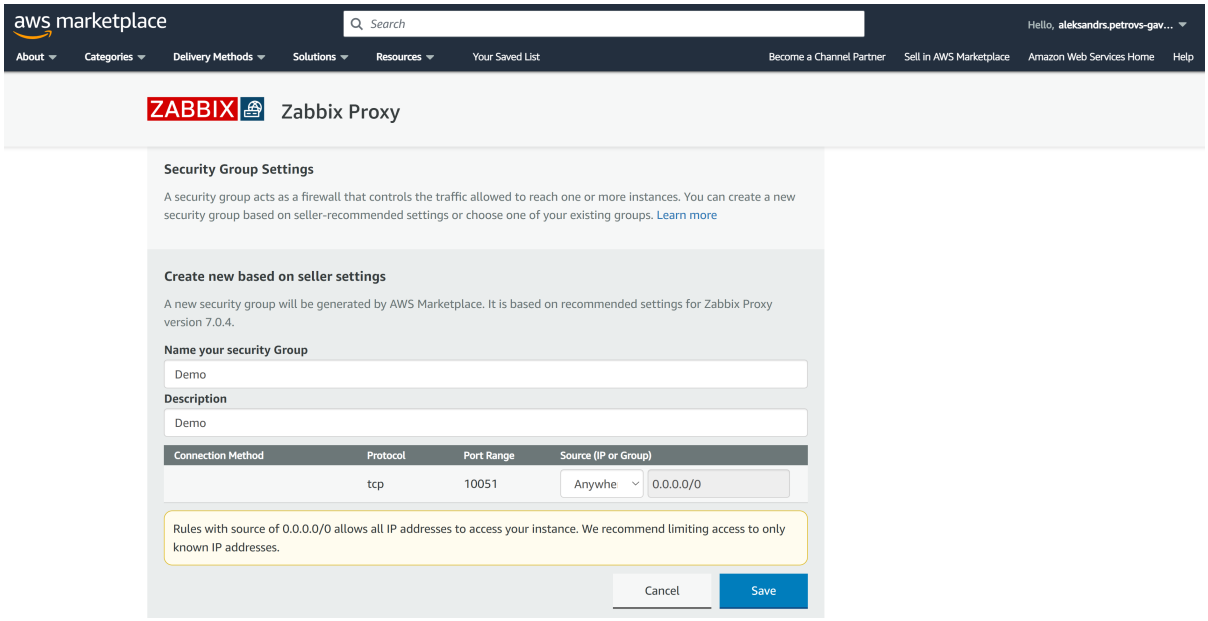
Continue to Launch ボタンをクリックします。Launch ページで、希望する Launch オプション、EC2 Instance Type、VPC、および Subnet の設定を選択します。



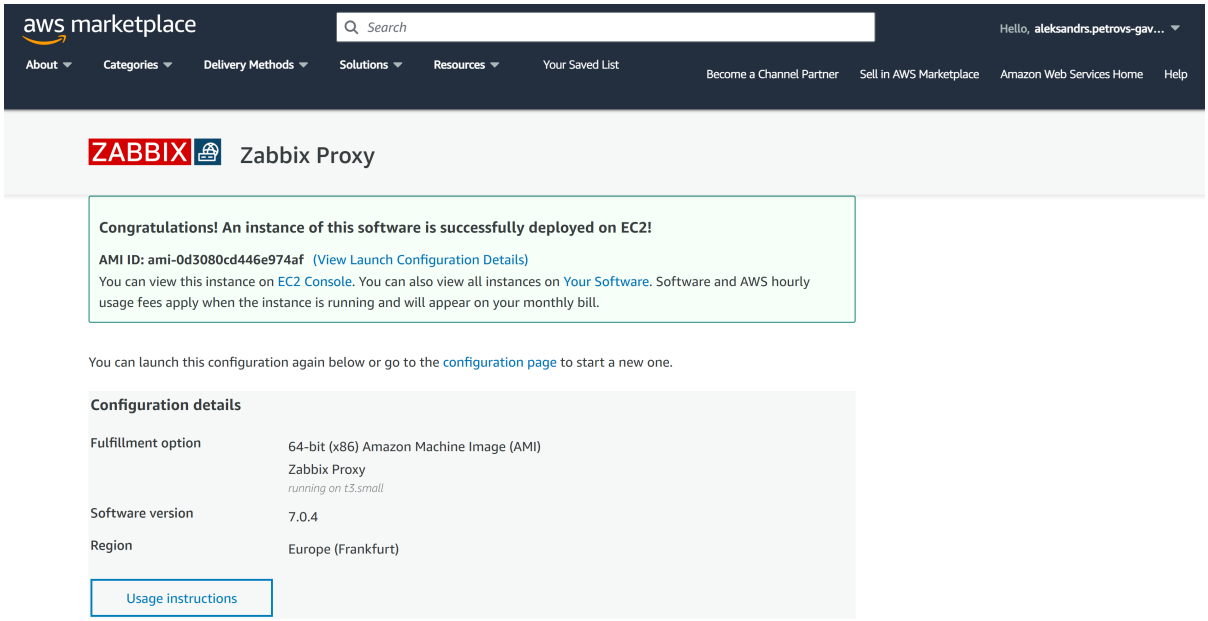
Launch ページの下部で、Security Group の設定を指定し、EC2 キーペアを選択または作成します。秘密鍵は安全な場所に保存してください。



Create a new Security Group based on the Seller settings を選択すると、TCP ポート 10051（Zabbix サーバー）が許可されます。
注: SSH 経由でインスタンスにアクセスする場合は、Security Group の設定で SSH 接続も許可する必要があります。

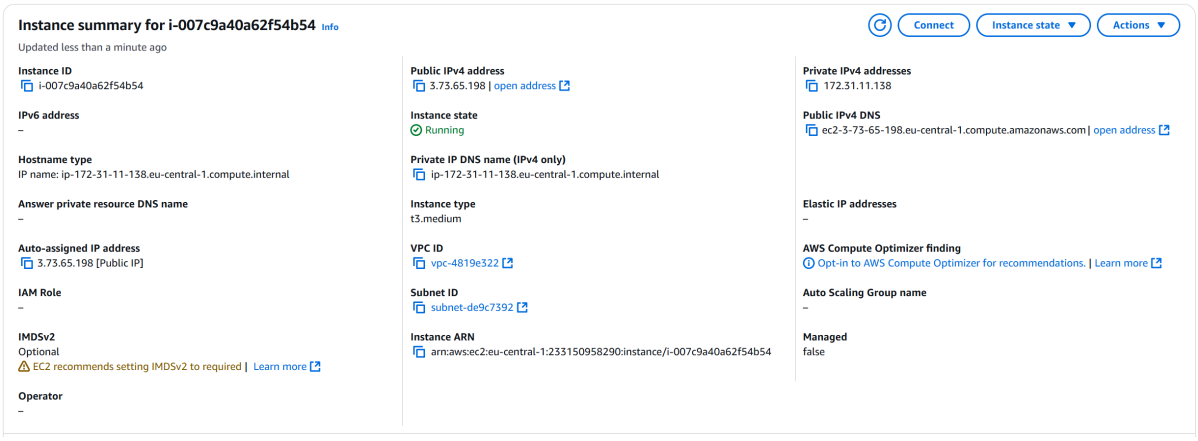


最後に、*Launch* ボタンをクリックします。これで Zabbix プロキシのデプロイが開始されます。



Zabbix プロキシの設定とアクセス

EC2 コンソールで「Instances」セクションを開き、新しくデプロイされた Zabbix プロキシインスタンスを選択します。



SSH を使用して、Zabbix プロキシの Public IPv4 アドレスまたは Public IPv4 DNS に接続します。デフォルトの SSH ユーザー名は `rocky` です。事前に生成したキーペアを使用することを忘れないでください。

認証情報の生成には数分かかる場合があります。接続を試みる前に、数分待ってください。

```
login as: ec2-user
Authenticating with public key "Demo"
*******************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

*******************************************************************************
[ec2-user@ip-172-31-11-215 ~]$
```

Zabbix プロキシの設定ファイルは/etc/zabbix/zabbix_proxy.conf にあります。

設定ファイルは、vi など任意のテキストエディターで開くことができます。

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

初期プロキシ設定では、Zabbix プロキシの設定ファイルにある ProxyMode、Server、Hostname パラメータを変更します。

プロキシモードは、ProxyModeパラメータを編集して変更できます。アクティブモードの場合は ProxyMode=0、パッシブモードの場合は ProxyMode=1 を設定します。

Serverパラメータには、Zabbix サーバーの IP アドレスを指定します。例:

```
Server=192.168.50.141
```

Hostnameパラメータには、Zabbix プロキシのホスト名を指定します。ここで定義するホスト名は、Zabbix Web インターフェースでのプロキシ設定時に指定したプロキシ名と一致している必要があります。例:

```
Hostname=EC2-Frankfurt
```

設定が完了したら、次のコマンドで Zabbix プロキシを再起動します。

```
sudo systemctl restart zabbix-proxy
```

注: 設定ファイルの編集やコマンドラインからのコマンド実行には sudo を使用してください。

```
### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name of Zabbix server to get configuration data from and send data to.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix
server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated
 equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

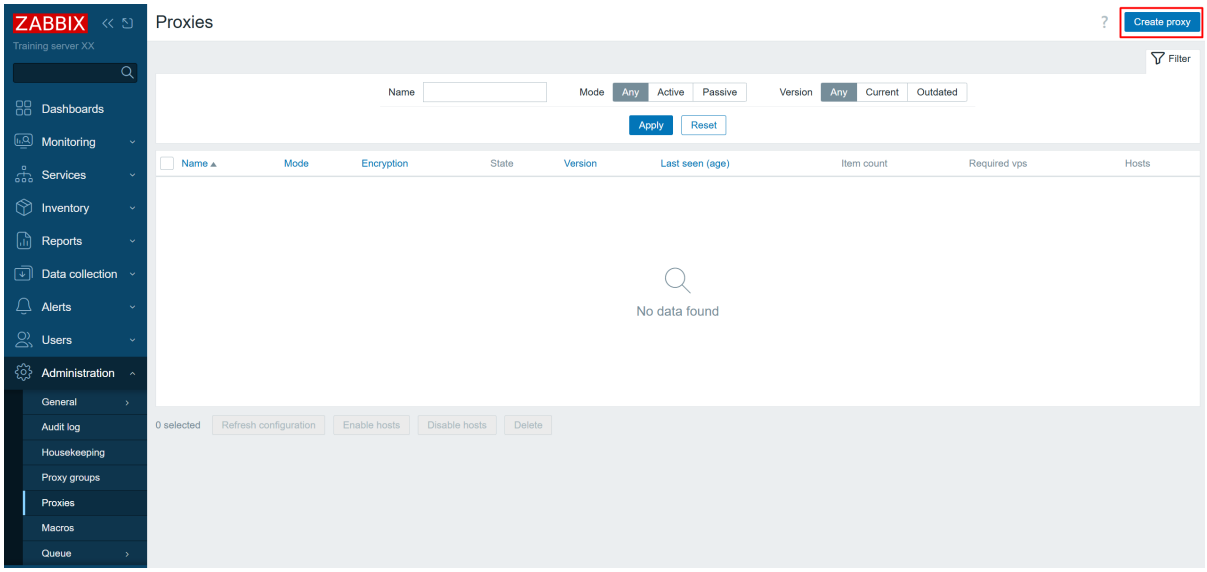
Hostname=EC2-Frankfurt
```

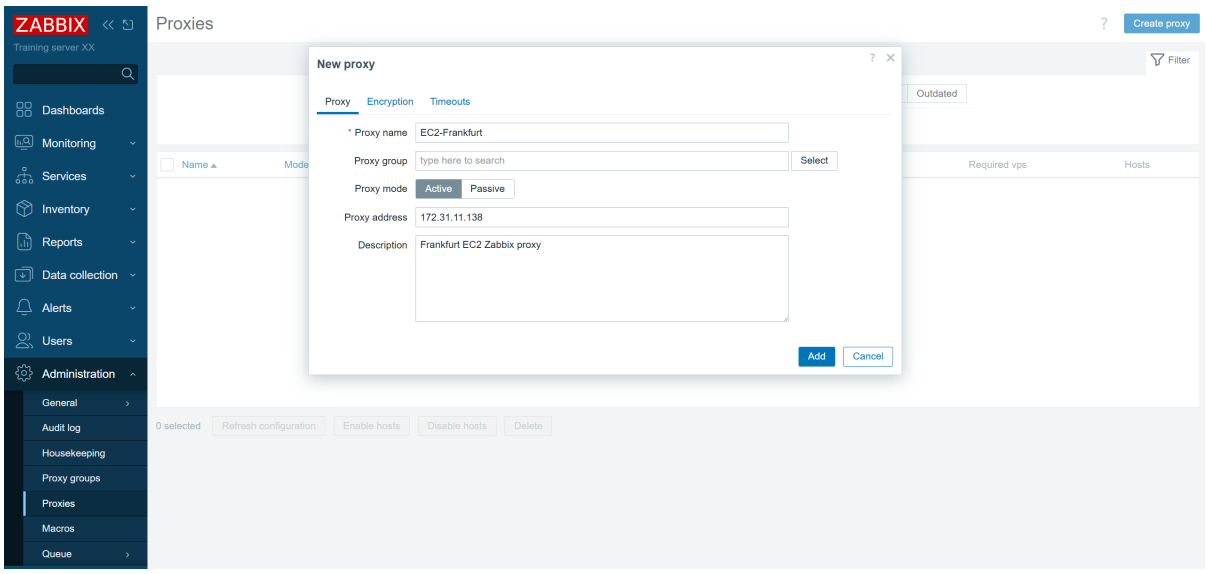
Zabbix Web インターフェースを開き、*Administration* > *Proxies* に移動します。*Create proxy* ボタンをクリックします。

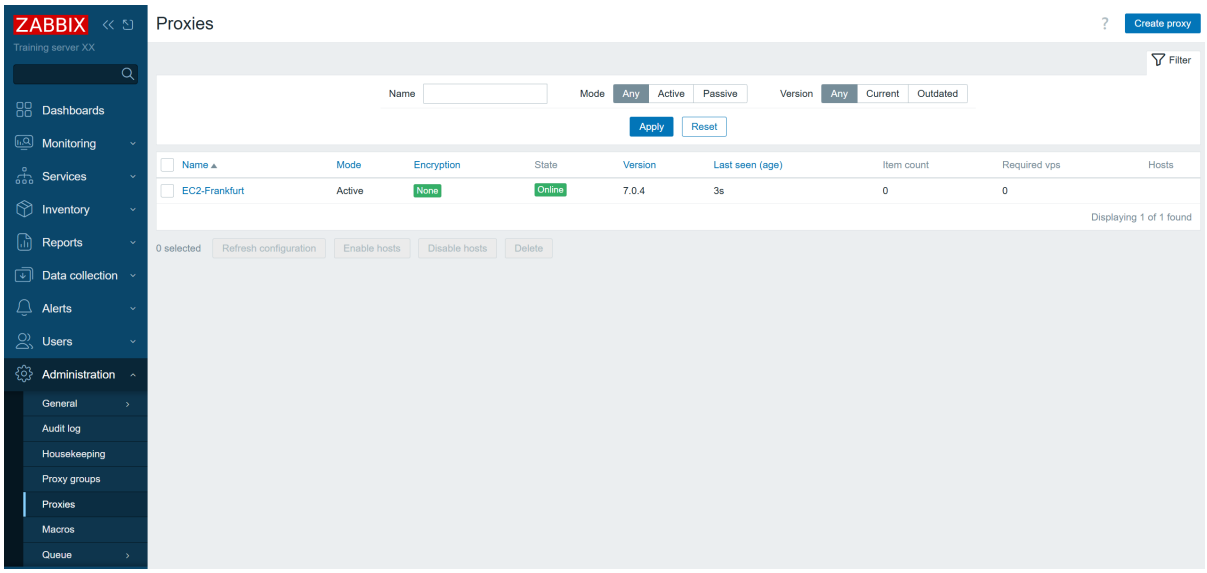
プロキシの設定ファイルにある Hostname パラメータの値と同じプロキシ名を入力します。

プロキシモードを選択し、プロキシの IP アドレスを入力します（アクティブプロキシの場合は任意）。

Create proxy をクリックします。







プロキシデータベースの場所

SQLite プロキシデータベースは `/tmp/zabbix_proxy.sqlite3` に保存され、Zabbix プロキシ設定ファイルの DBName パラメータで指定されます。

Zabbix サーバーの **Azure** デプロイガイド

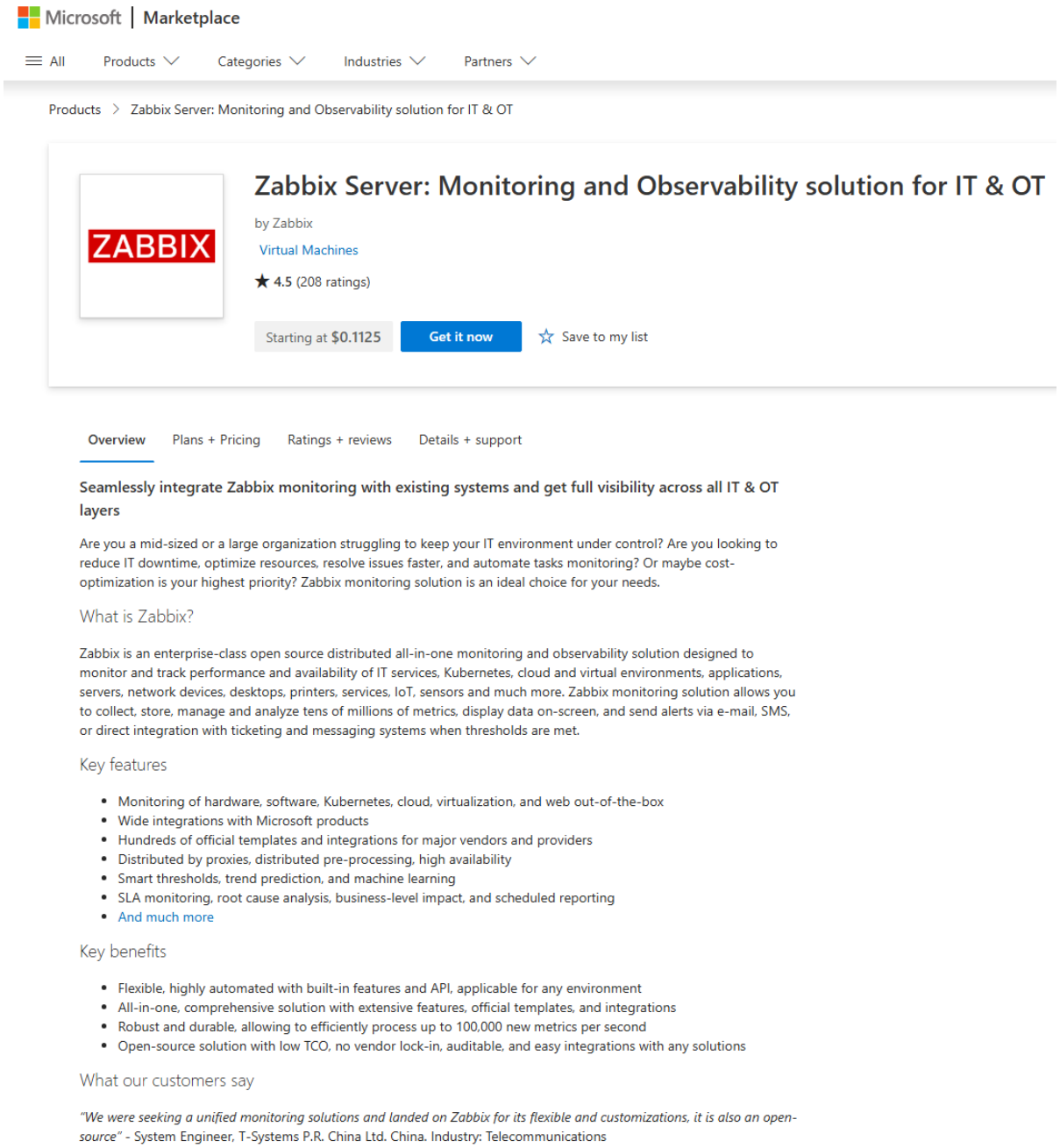
概要

このページでは、Azure を使用して Zabbix サーバーをデプロイする方法について説明します。

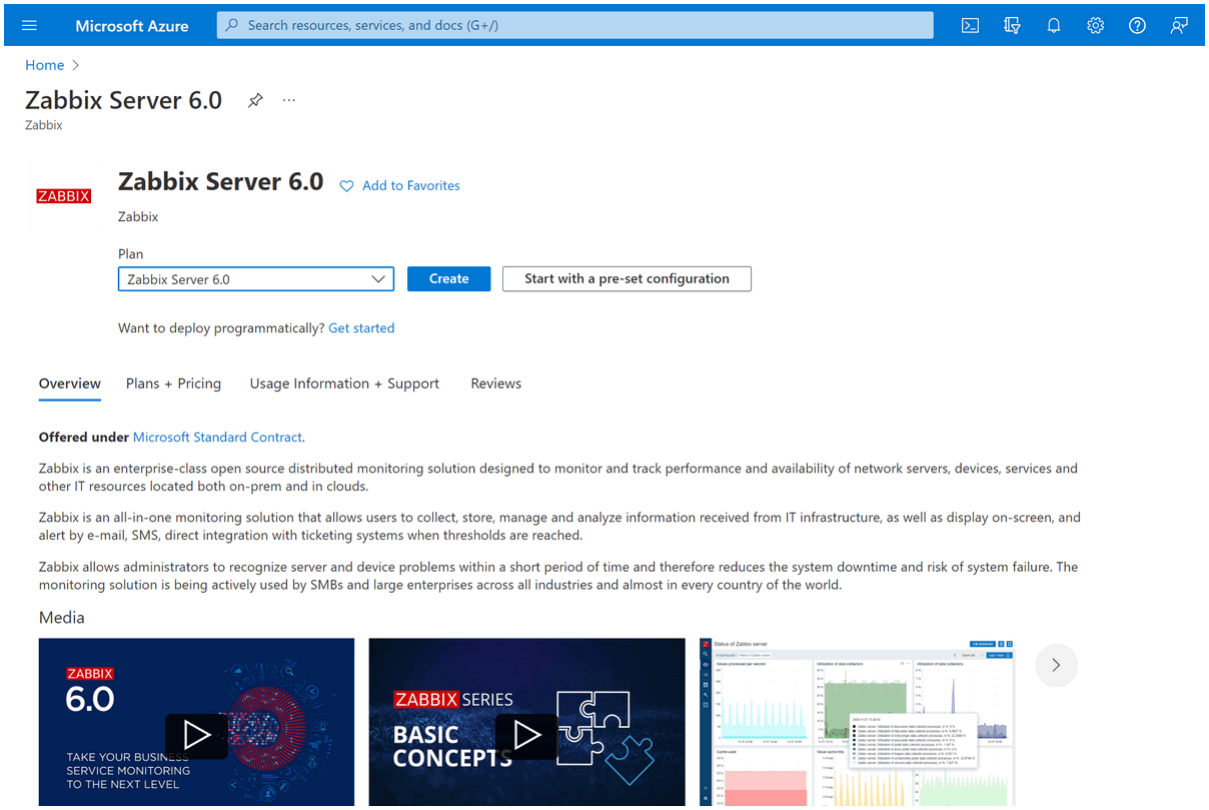
> **Attention:**
> このプラットフォームでは、複数の Zabbix バージョンをデプロイできます。デプロイした Zabbix と同じバージョンのドキュメントを必ずお読みください。異なる Zabbix バージョンのドキュメントは、バージョンのドロップダウンメニューから選択できます。

アプライアンスのデプロイ

Zabbix アプライアンスをデプロイするには、まず Zabbix Azure Marketplace ページ に移動します。概要を確認し、*Get it now* をクリックします。



デプロイの概要、料金プラン、その他の詳細を確認します。次に、Zabbix アプライアンスをデプロイするため、*Create* または *Start with a pre-set configuration* をクリックします。

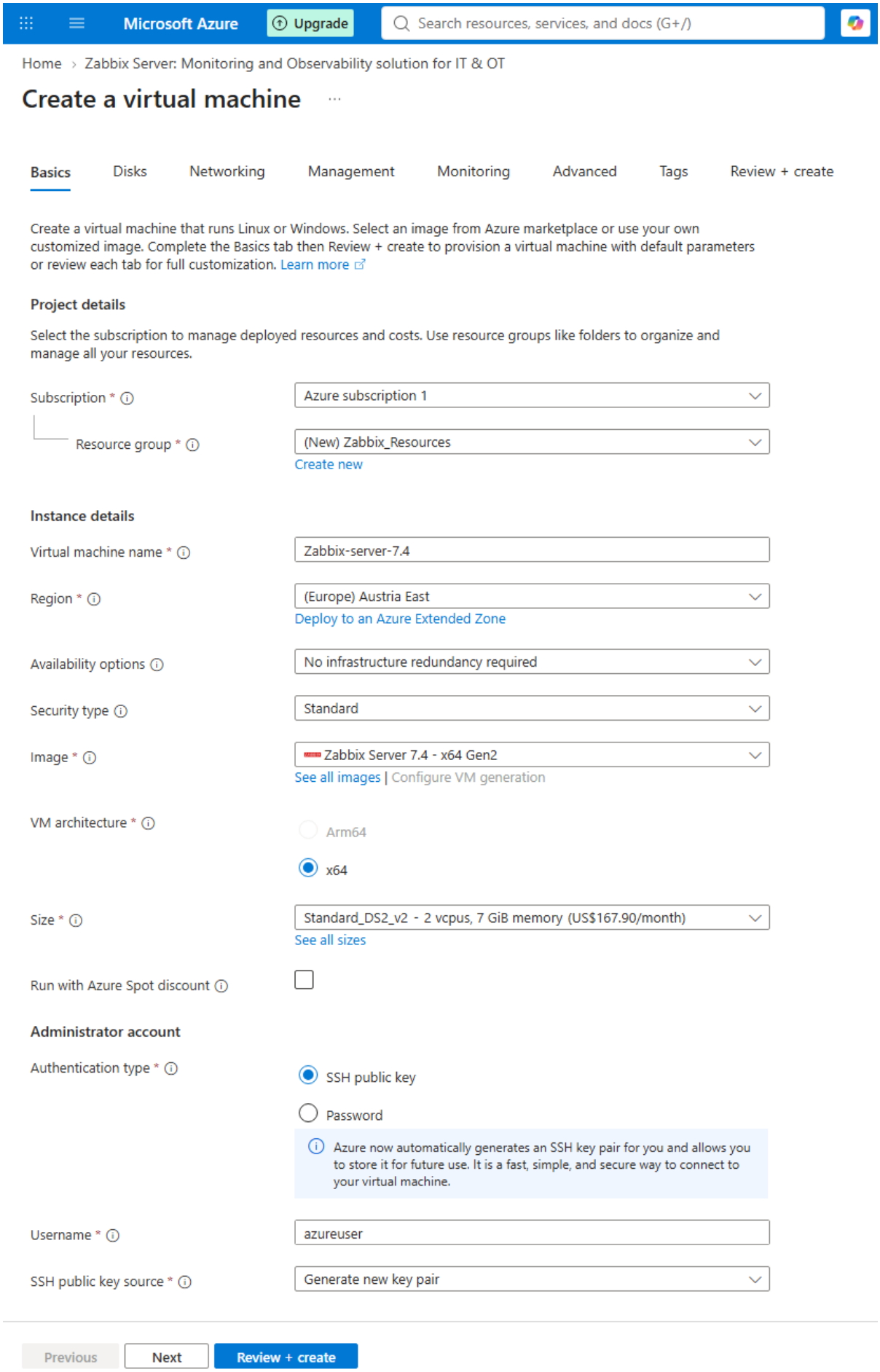


VM サイズを選択し、管理者アカウントを作成して、ディスク、ネットワーク、管理、その他の設定を定義することで VM を設定します。これらの設定は、インフラストラクチャの要件によって異なる場合があります。

次の設定を例として使用できます。

Basics タブ:

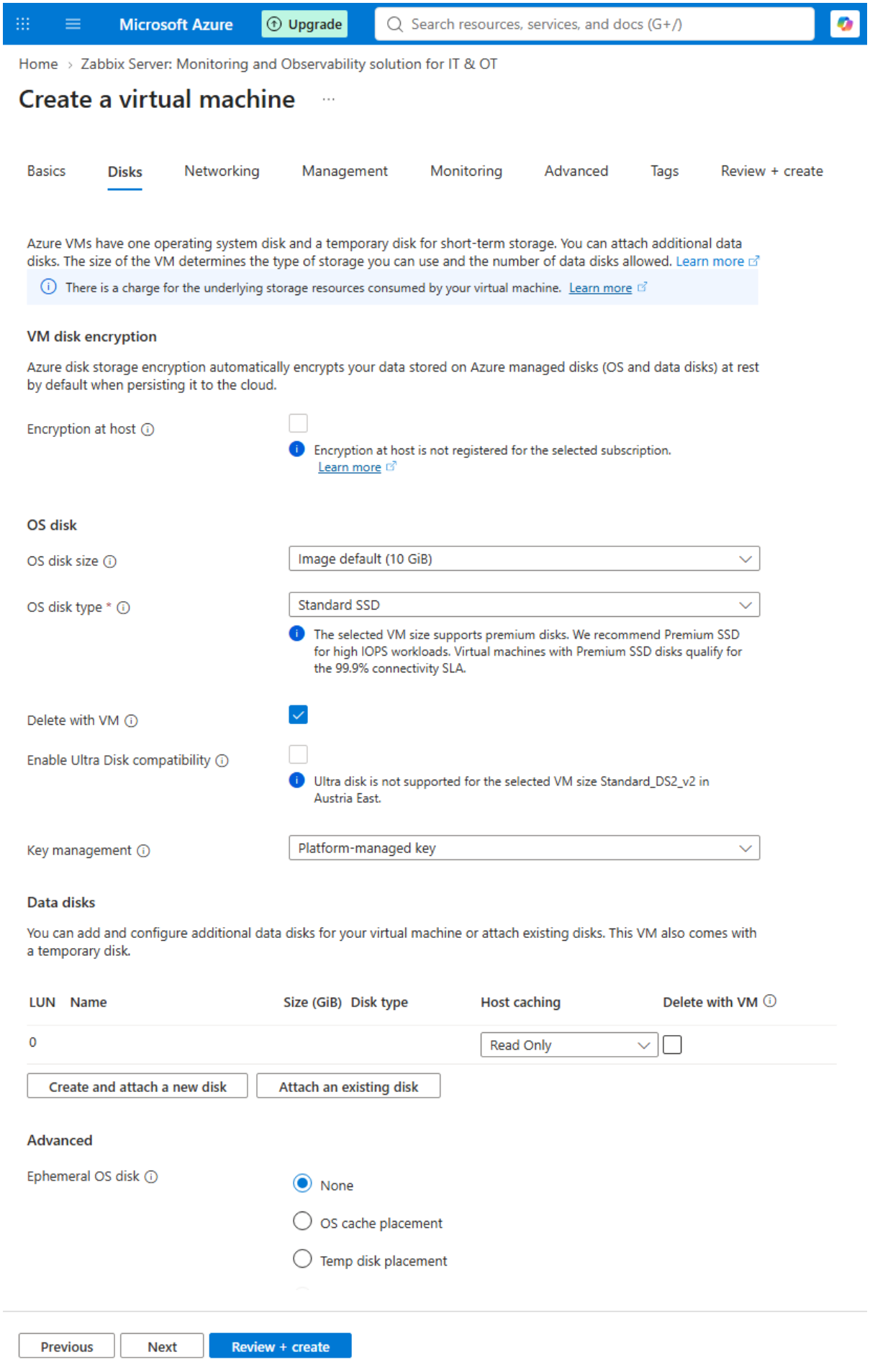
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-server--7.4
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Server 7.4 - x64 Gen2
- *Size*: Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Disks タブ:

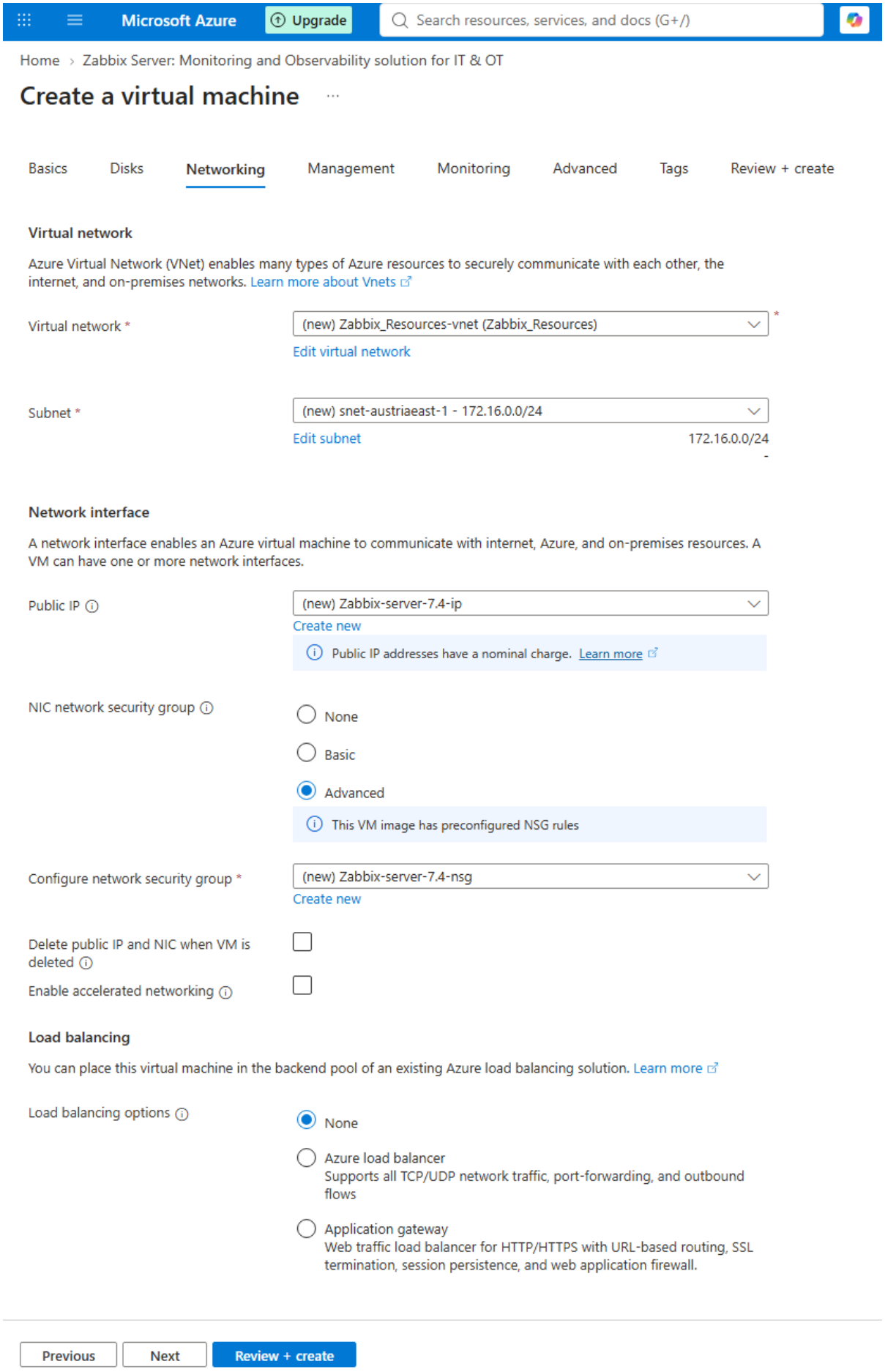
- *Encryption at host*: 未選択
- *OS disk type*: Standard SSD

- *Delete with VM*: 選択
- *Enable Ultra Disk compatibility*: 未選択

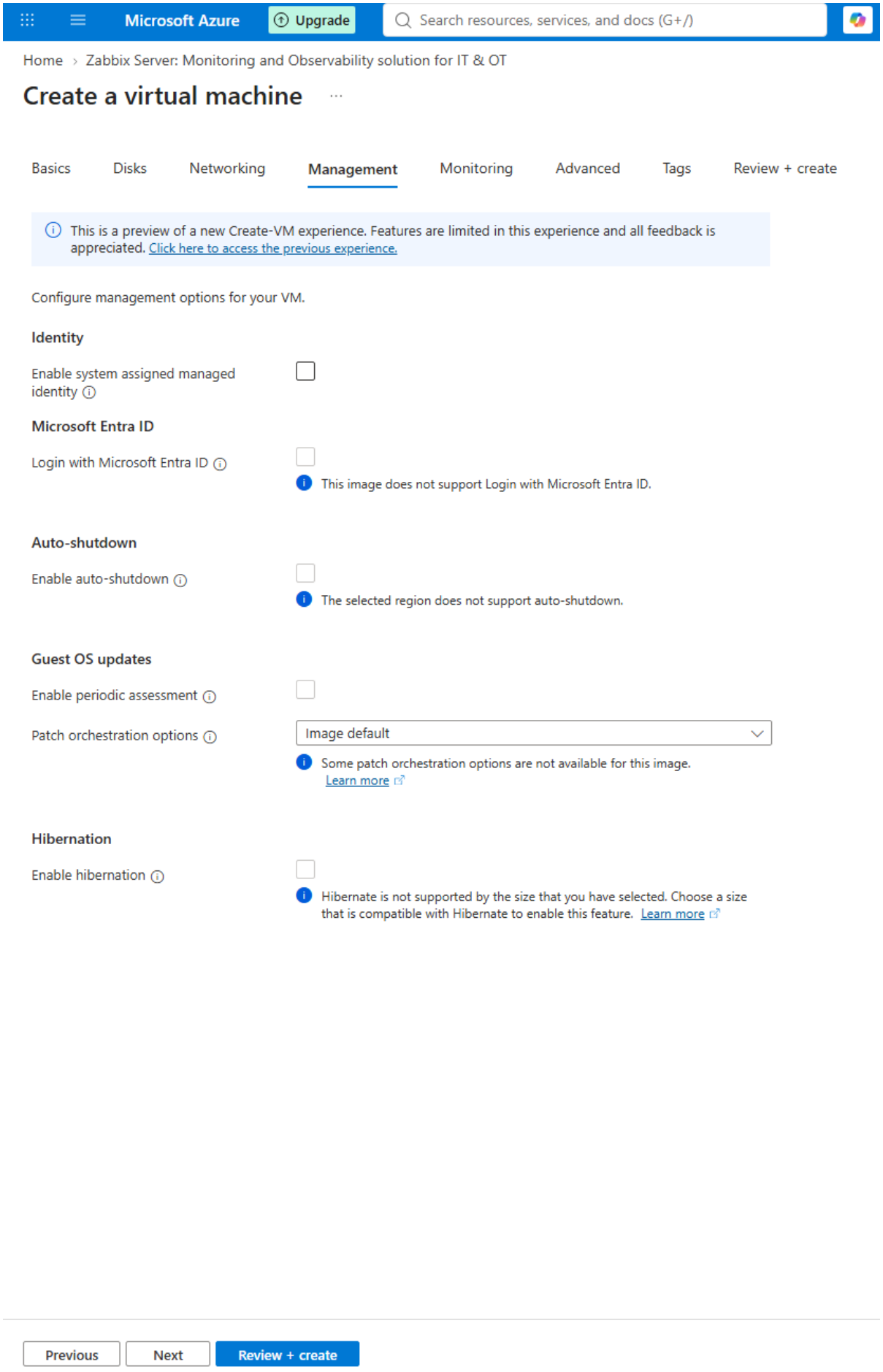


Networking タブ:

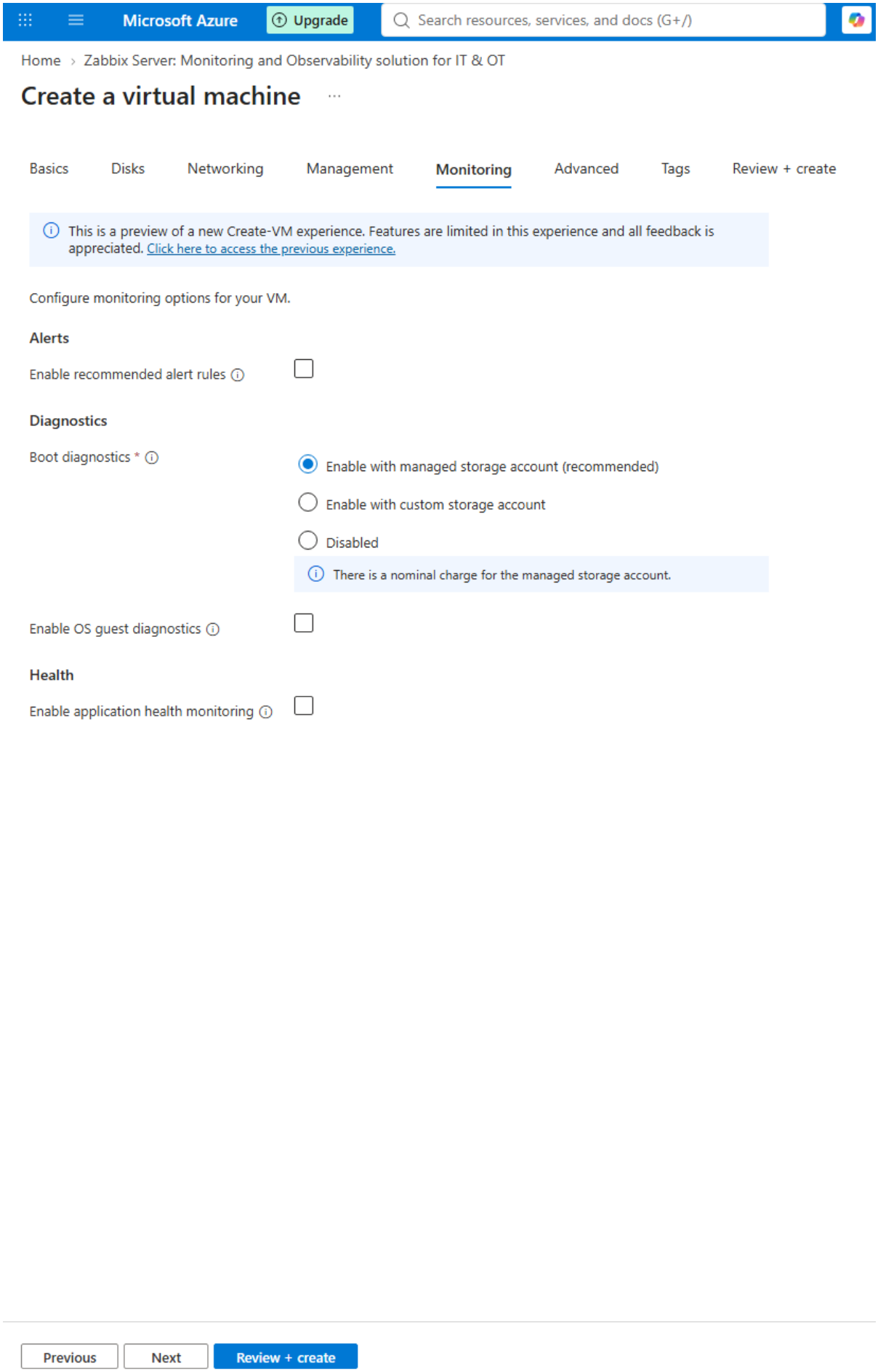
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-server-7.4-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-server-7.4-nsg



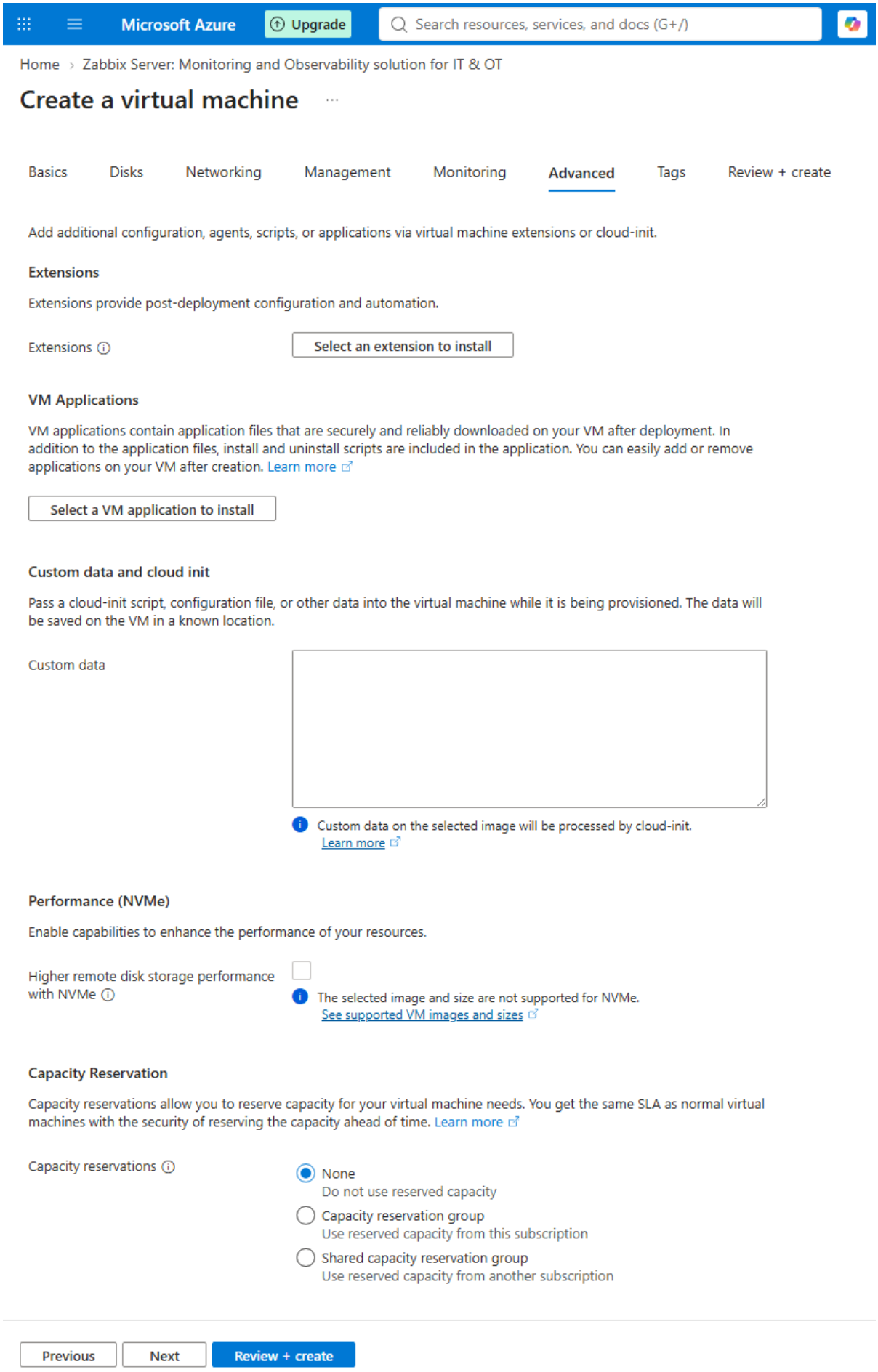
Management タブ: パラメータはデフォルトのままにできます。



Monitoring タブ: パラメータはデフォルトのままにできます。



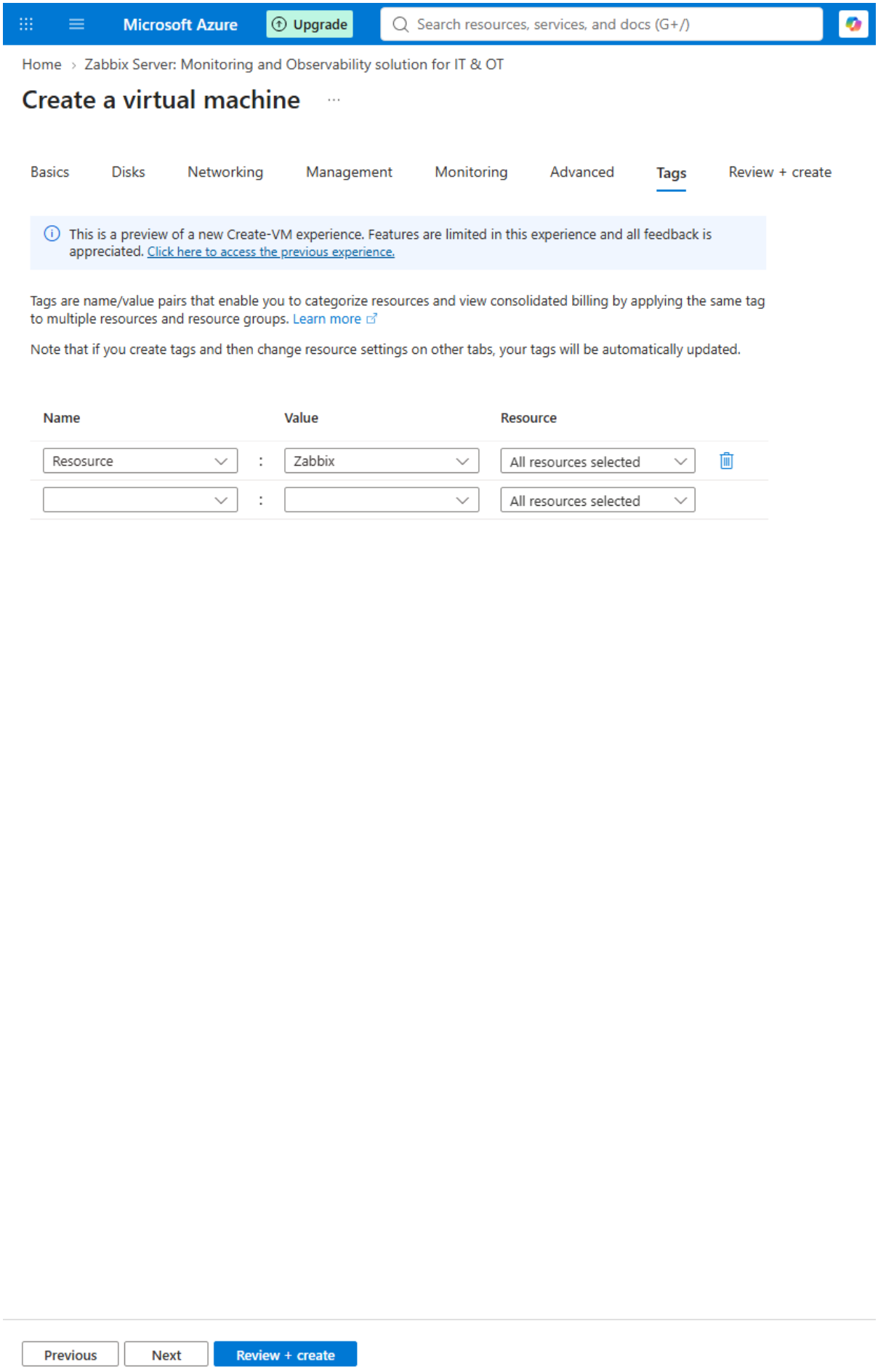
Advanced タブ: パラメータはデフォルトのままにできます。



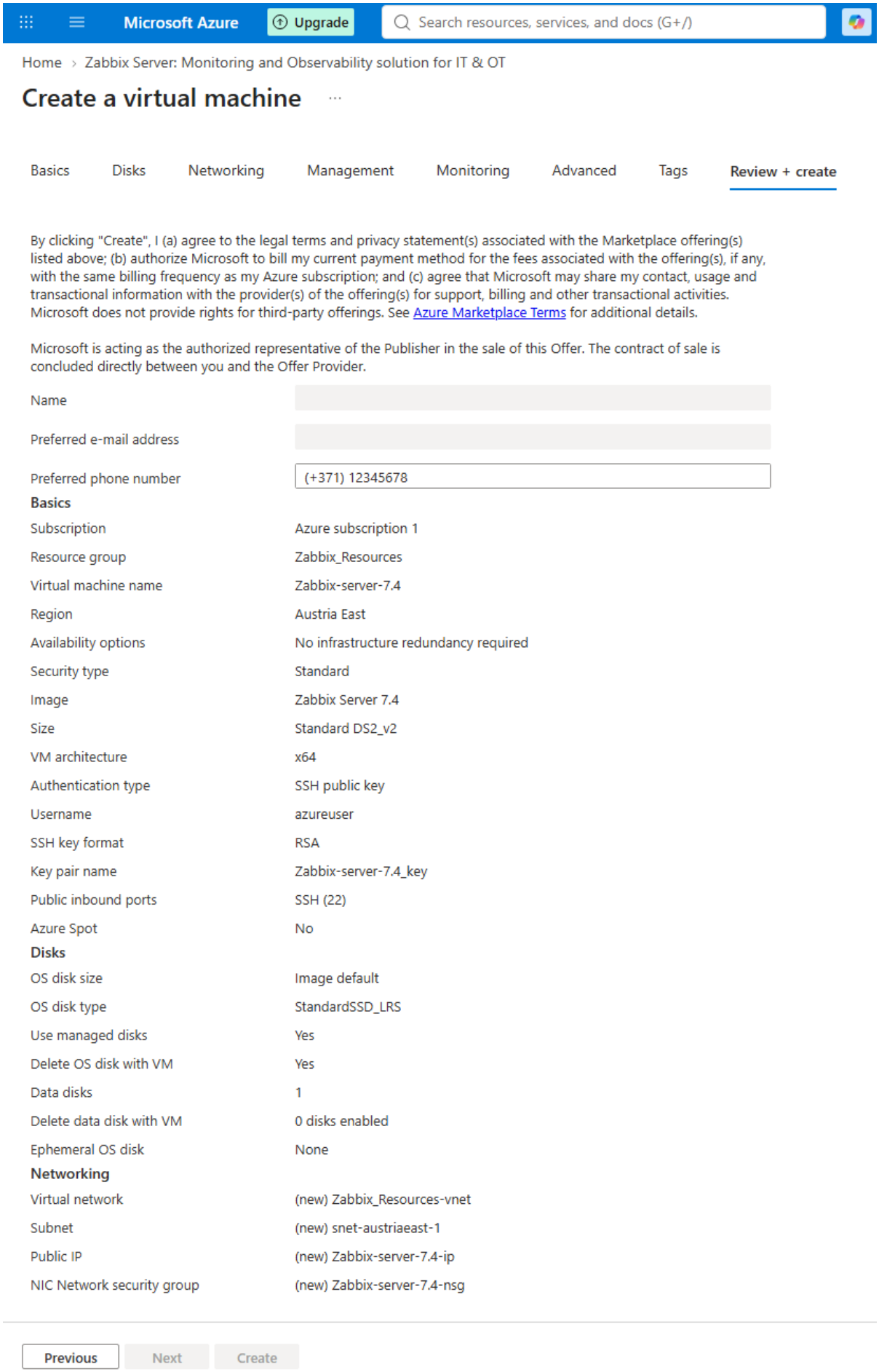
Tags タブ:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



VM の詳細を確認し、VM を作成します。



VM のデプロイステータスページにリダイレクトされます。デプロイが完了するまで待ちます。

デプロイが完了したら、リソースを開き、そのパブリック IP アドレスを確認します。次に、SSH（または利用可能な別の方法）を使用して Zabbix サーバーに接続し、Zabbix Web インターフェースのログイン認証情報を取得します。

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

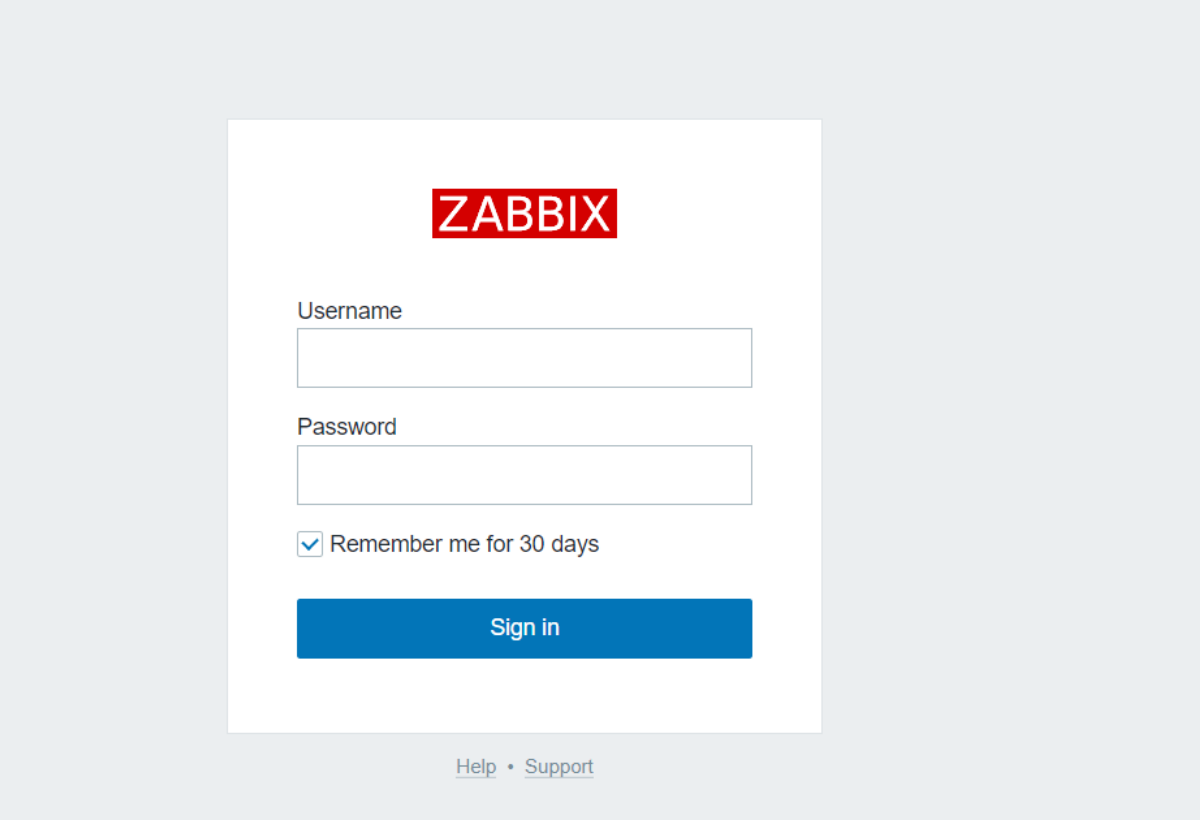

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Web ブラウザーで Zabbix アプライアンスのパブリック IP アドレスを開き、Zabbix Web インターフェースにアクセスします。ログインするには、前の手順で取得したユーザー名とパスワードを使用します。



データベースへのアクセス

パスワードを入力せずに、`root` ユーザーを使用して Zabbix データベースにアクセスできます。認証情報は `/root/.my.cnf` 設定ファイルに保存されています。

Zabbix プロキシの **Azure** デプロイガイド

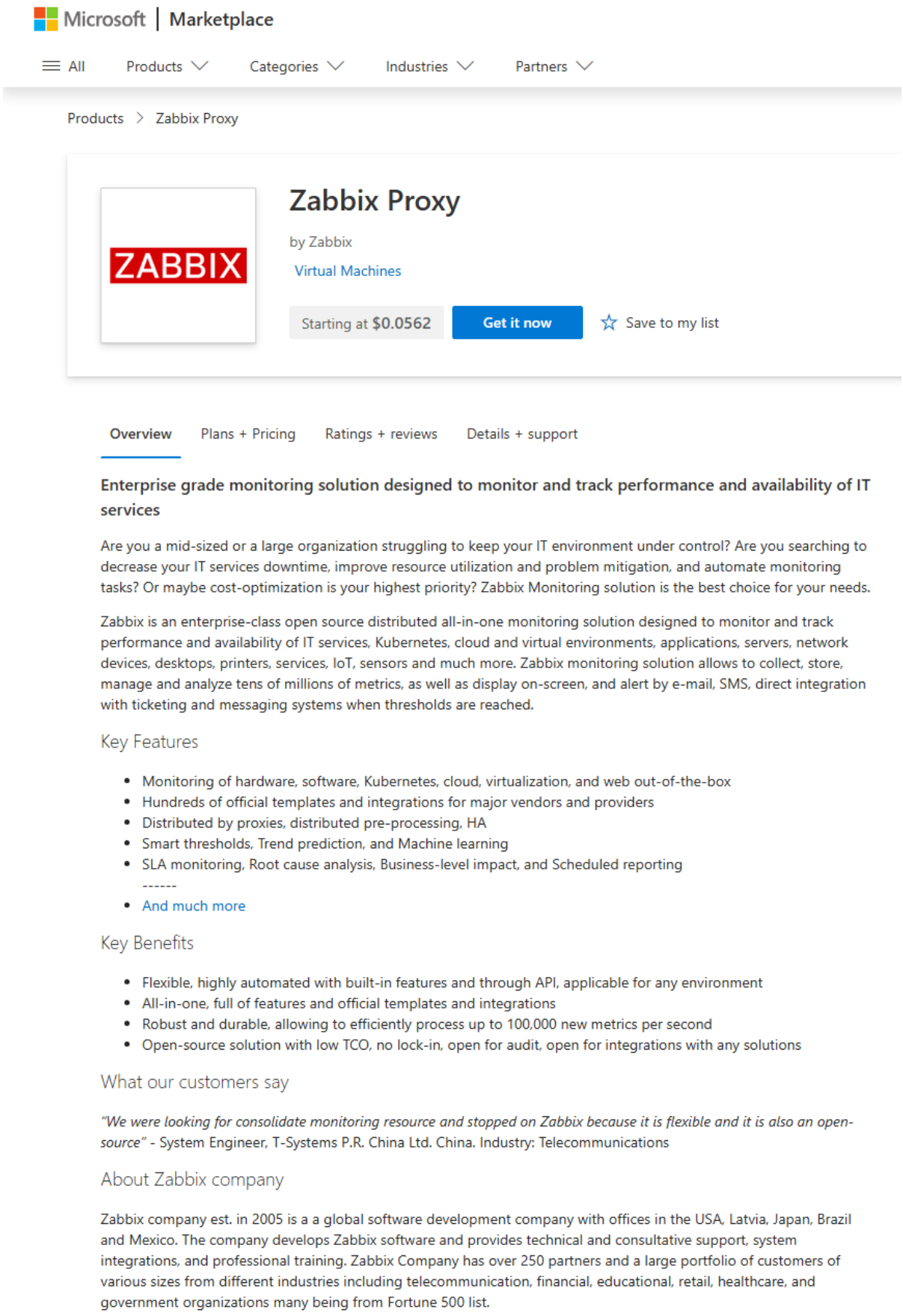
概要

このページでは、Azure を使用して Zabbix プロキシをデプロイする方法について説明します。

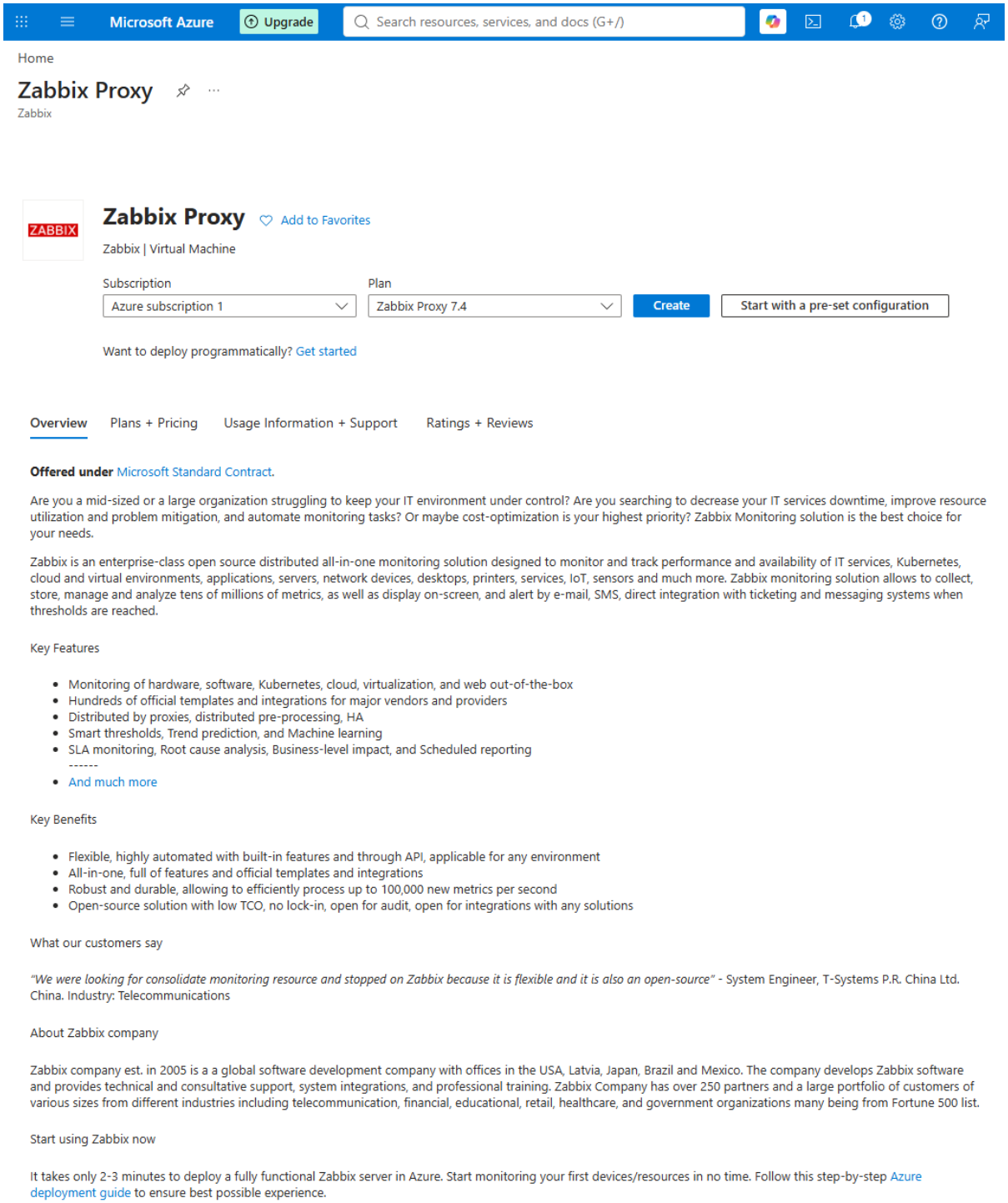
Attention:
このプラットフォームでは、複数の Zabbix バージョンをデプロイできます。デプロイした Zabbix と同じバージョンのドキュメントを必ずお読みください。異なる Zabbix バージョンのドキュメントは、バージョンのドロップダウンメニューから選択できます。

Zabbix プロキシのデプロイ

Zabbix プロキシをデプロイするには、まず Zabbix Azure マーケットプレースページ に移動します。*Get it now* をクリックします。



ここでは、現在のデプロイに関する一般的な概要、料金プランなどの追加情報を確認できます。*Create* または *Start with a pre-set configuration* をクリックします。

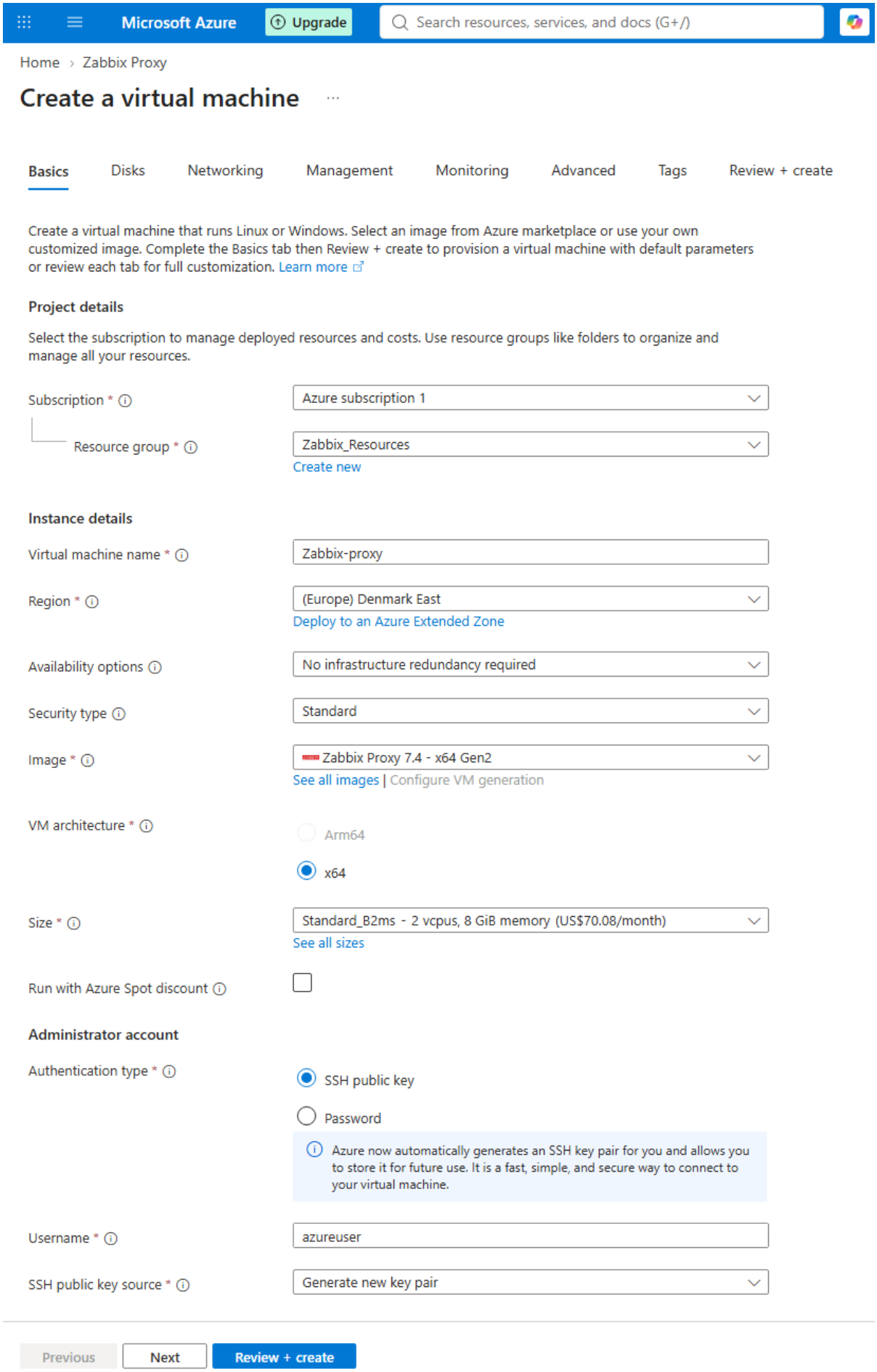


詳細な VM 設定に進みます。VM サイズ、管理者アカウント、ディスク、ネットワーク、管理、その他の設定を選択します。これらの設定は、インフラストラクチャの要件によって異なる場合があります。

以下の設定を例として使用できます。

Basics タブ:

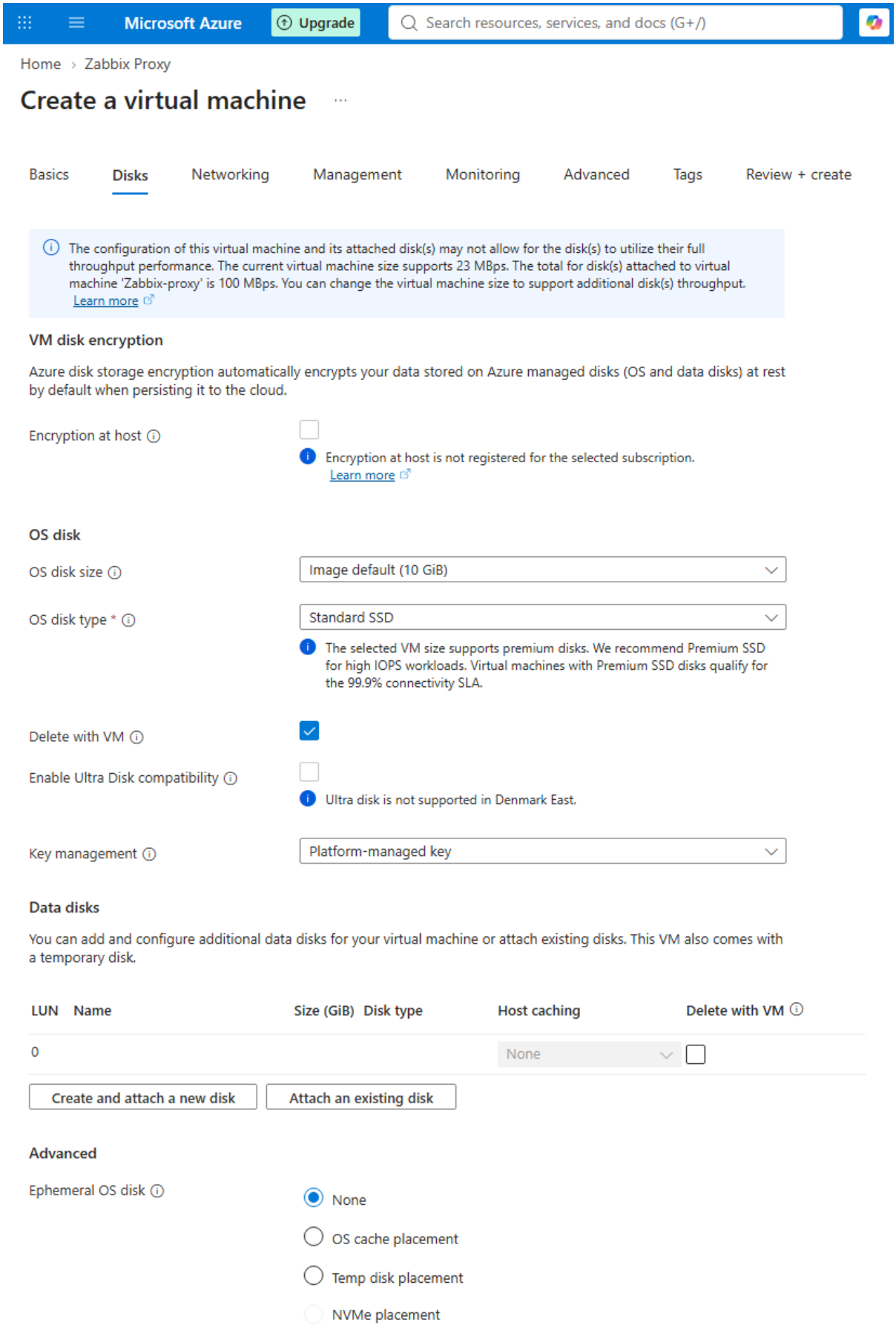
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-proxy
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Proxy 7.4 - x64 Gen2
- *Size*: Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Disks タブ:

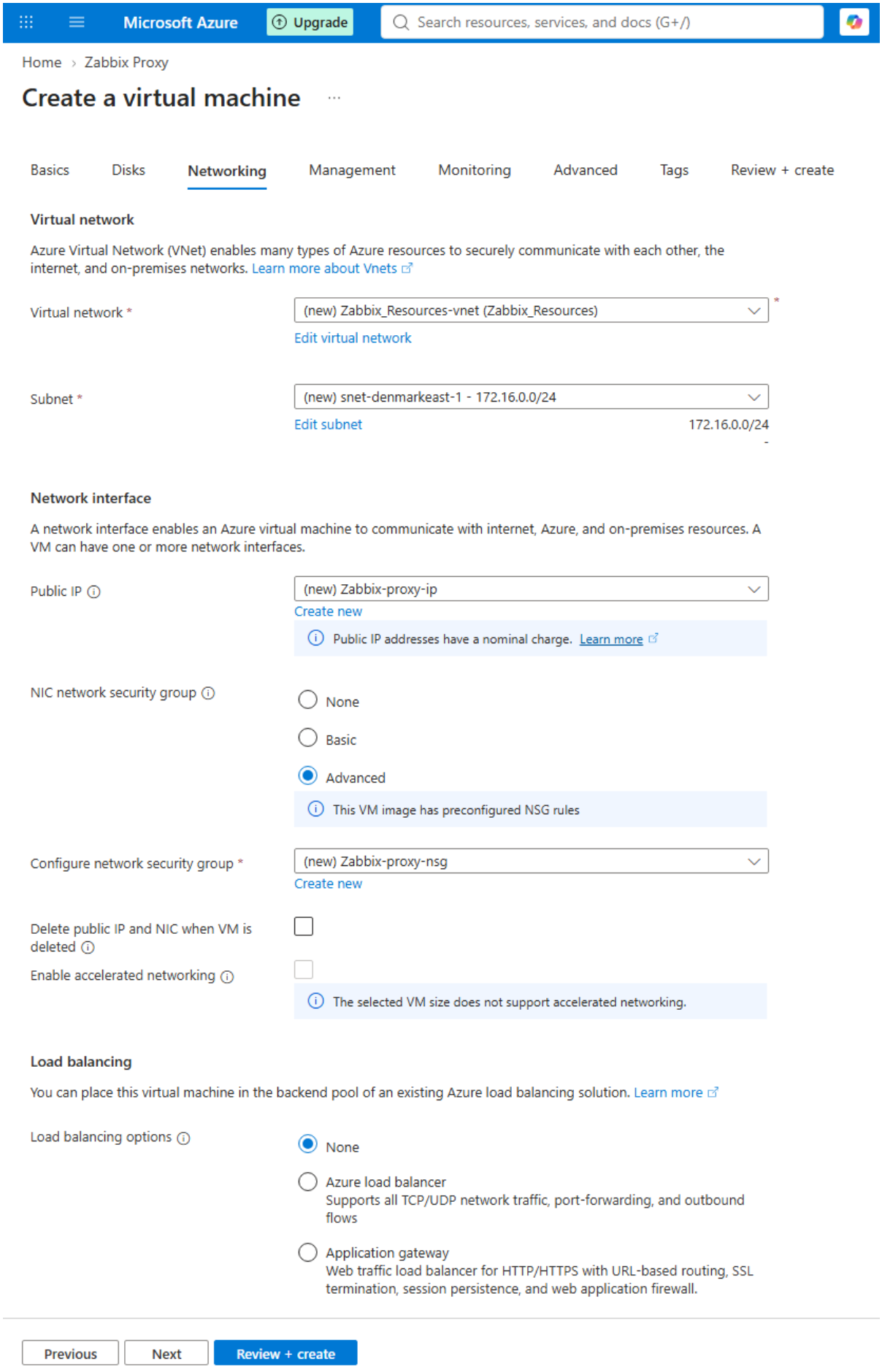
- *Encryption at host*: unchecked

- *OS disk type*: Standard SSD
- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

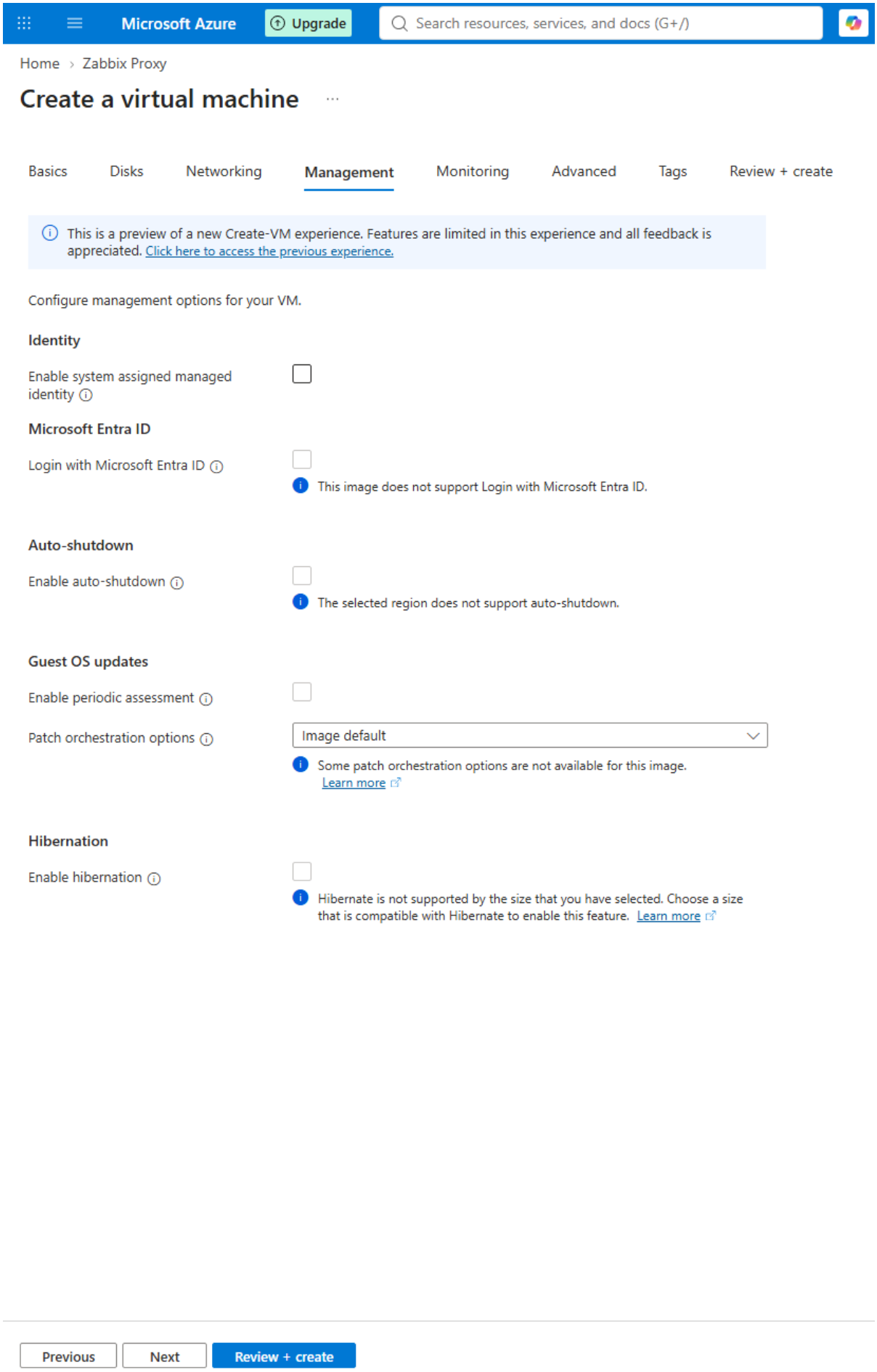


Networking タブ:

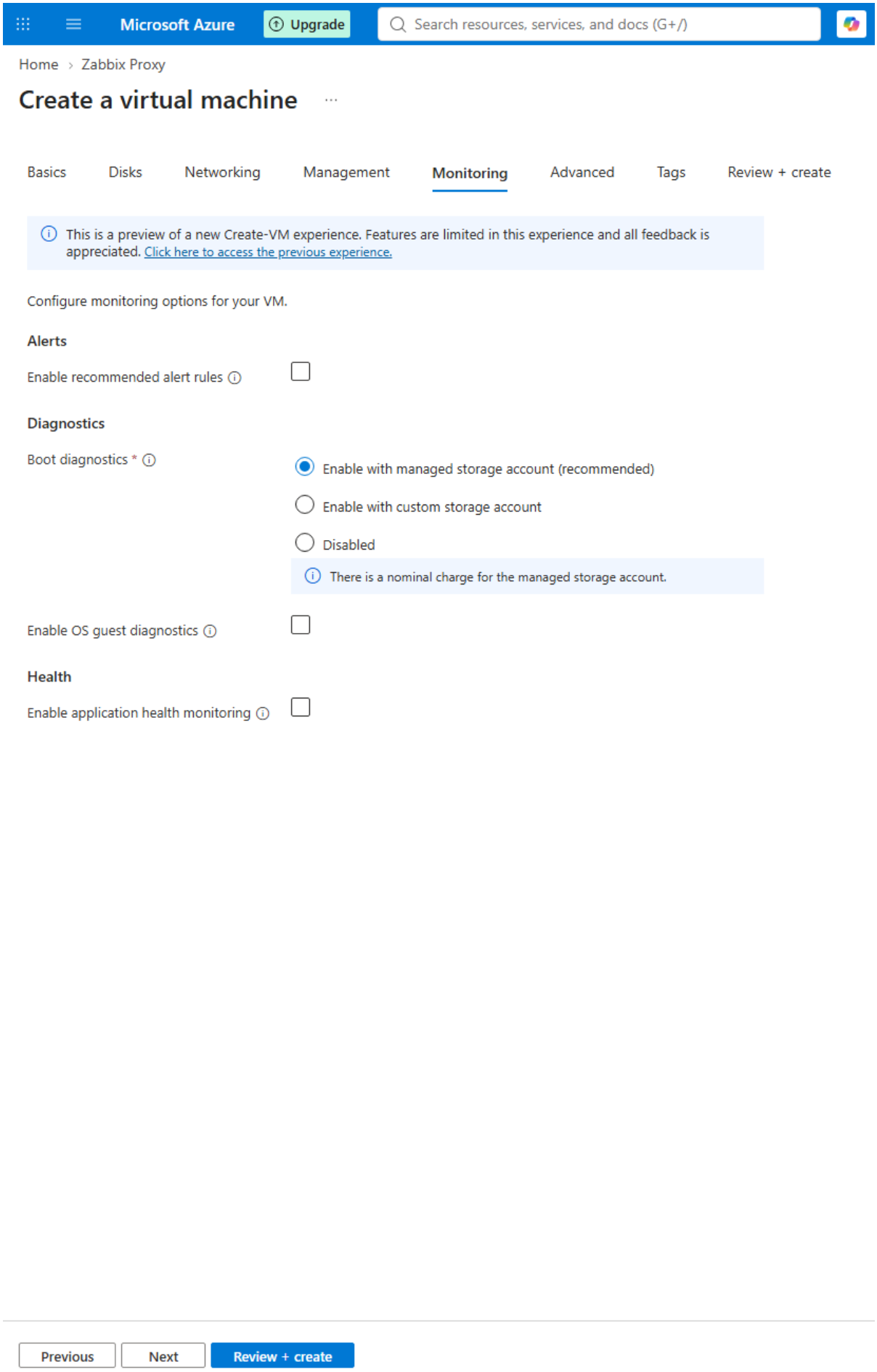
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-proxy-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-proxy-nsg



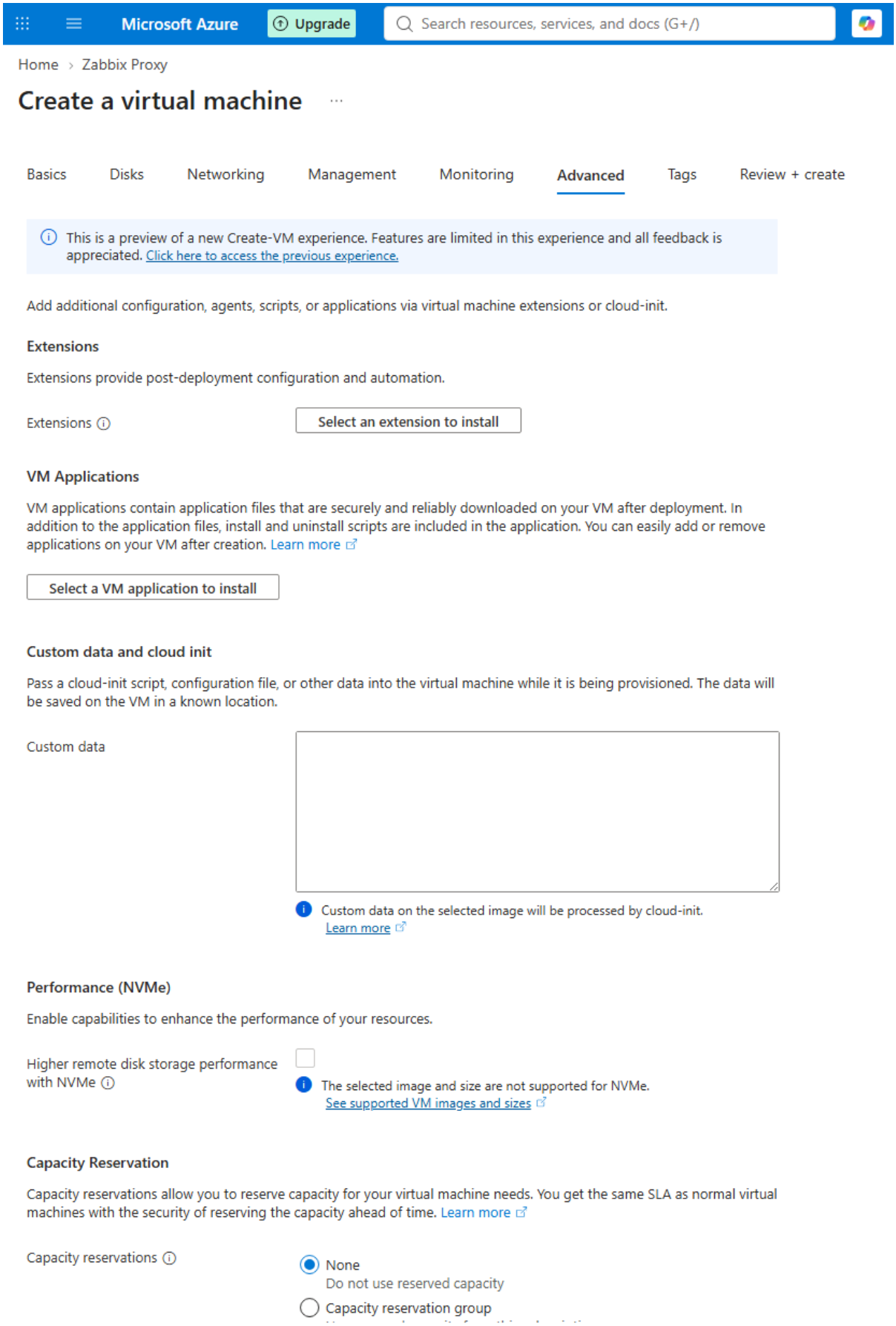
Management タブ: パラメータはデフォルトのままにできます。



Monitoring タブ: パラメータはデフォルトのままにできます。



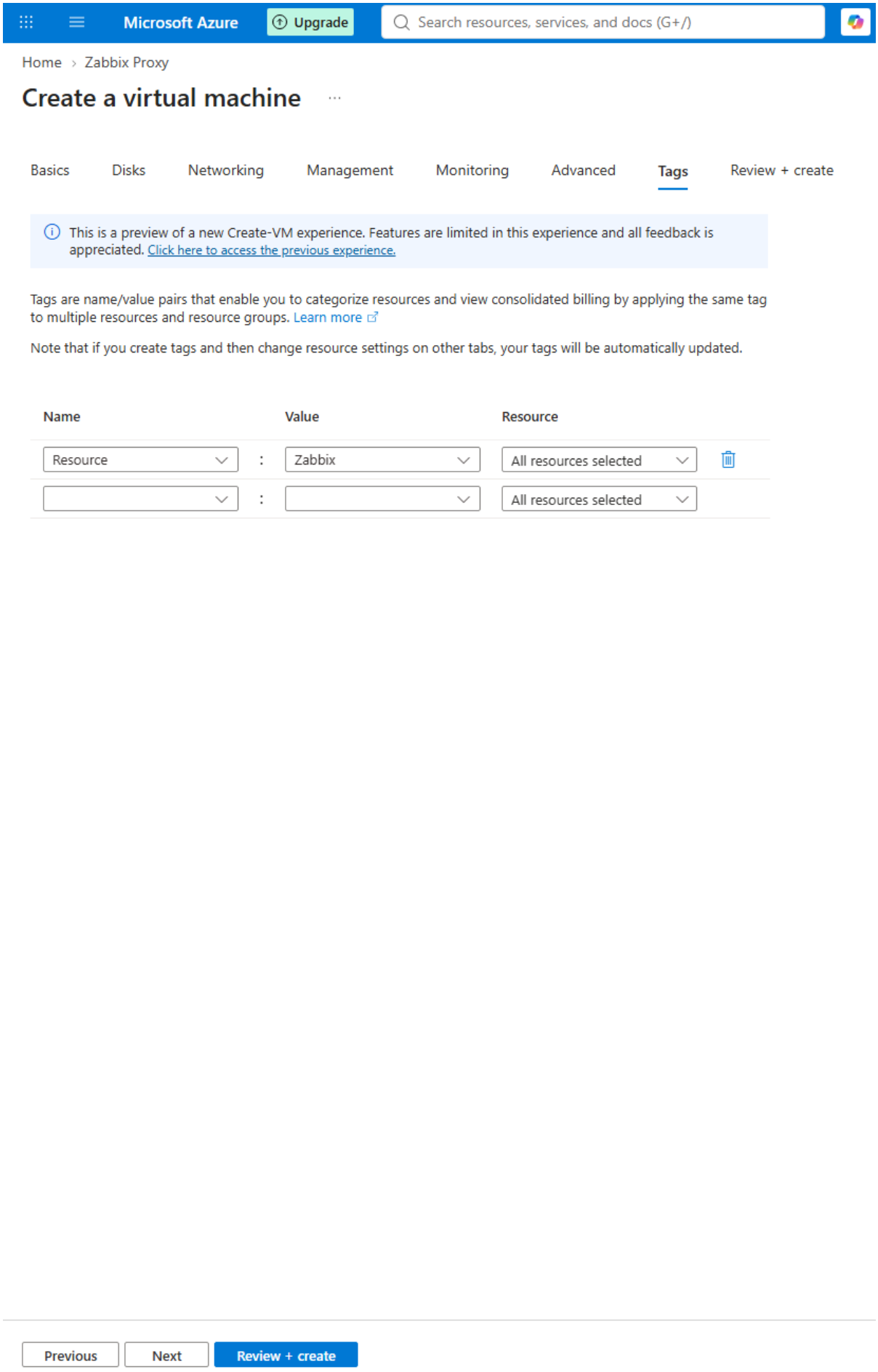
Advanced タブ: パラメータはデフォルトのままにできます。



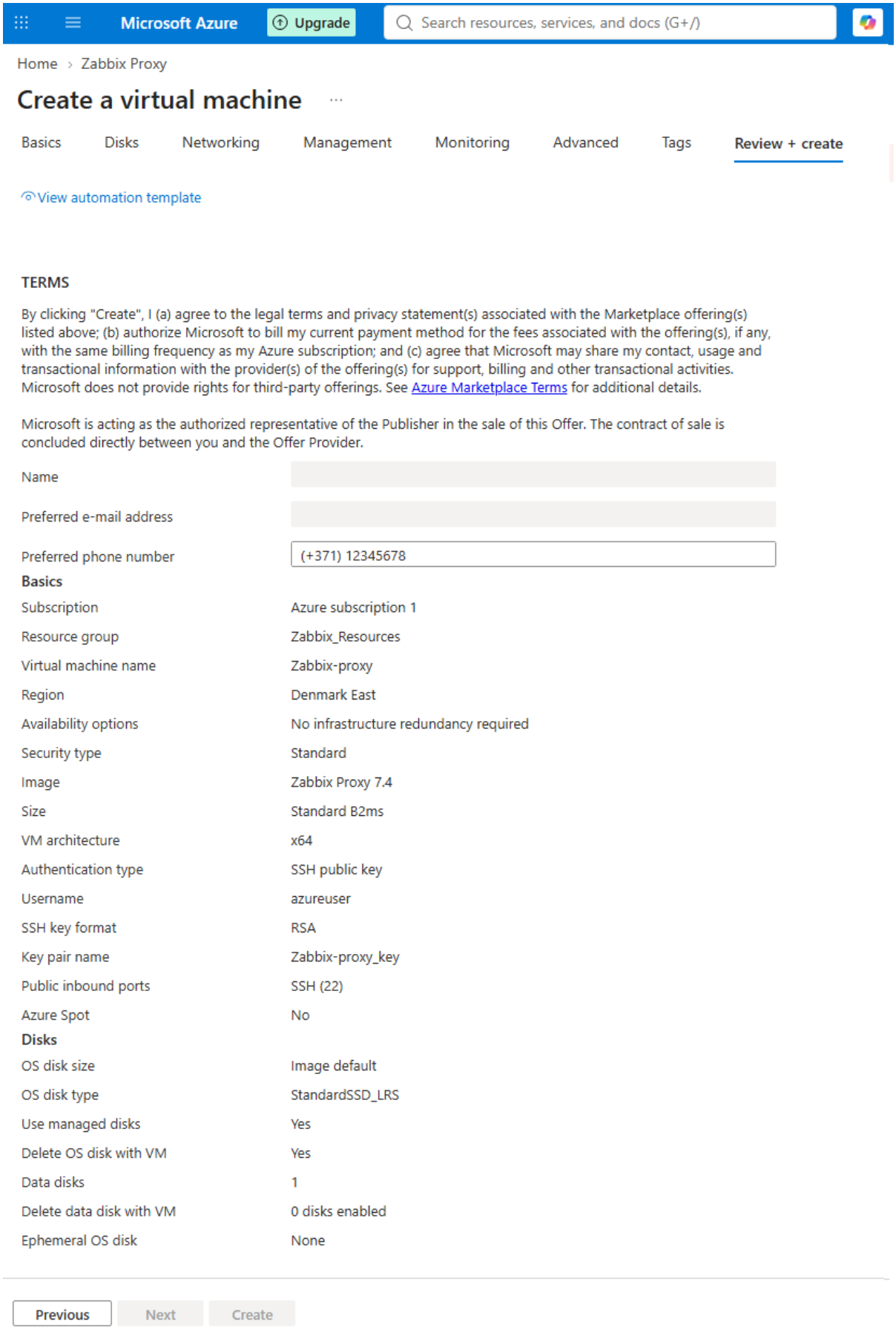
Tags タブ:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



VM の詳細を確認し、VM を作成します。



VM のデプロイステータスページにリダイレクトされます。デプロイが完了するまで待ちます。

デプロイが完了したら、リソースを開き、そのパブリック IP アドレスを確認します。VM のデプロイ時に作成した SSH および管理者認証情報を使用して Zabbix プロキシに接続し、プロキシを設定します。

```
******************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

******************************************************************************
[azureuser@Zabbix-proxy ~]$
```

Zabbix プロキシの設定ファイルは `/etc/zabbix/zabbix_proxy.conf` にあります。

初期プロキシ設定では、Zabbix プロキシの設定ファイルにある `ProxyMode`、`Server`、`Hostname` パラメータを変更します。

プロキシモードは、`ProxyMode` パラメータを編集して変更できます。アクティブモードの場合は `ProxyMode=0`、パッシブモードの場合は `ProxyMode=1` を設定します。

`Server` パラメータには、Zabbix サーバーの IP アドレスを指定します。

`Hostname` パラメータには、Zabbix プロキシのホスト名を指定します。ここで定義するホスト名は、Zabbix の Web インターフェースでプロキシを設定する際に指定するプロキシ名と一致している必要があります。

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Zabbix の Web インターフェースを開き、*Administration* > *Proxies* に移動します。*Create proxy* ボタンをクリックします。

プロキシ設定ファイルの *Hostname* パラメータの値と同じプロキシ名を入力します。

プロキシモードを選択し、プロキシの IP アドレスを入力します（アクティブプロキシの場合は任意）。

Create proxy ボタンをクリックしてプロキシを作成します。

プロキシデータベースの場所

SQLite プロキシデータベースは `/tmp/zabbix_proxy.sqlite3` にあり、Zabbix プロキシ設定ファイルの `DBName` パラメータで指定されています。

Google Cloud デプロイガイド

概要

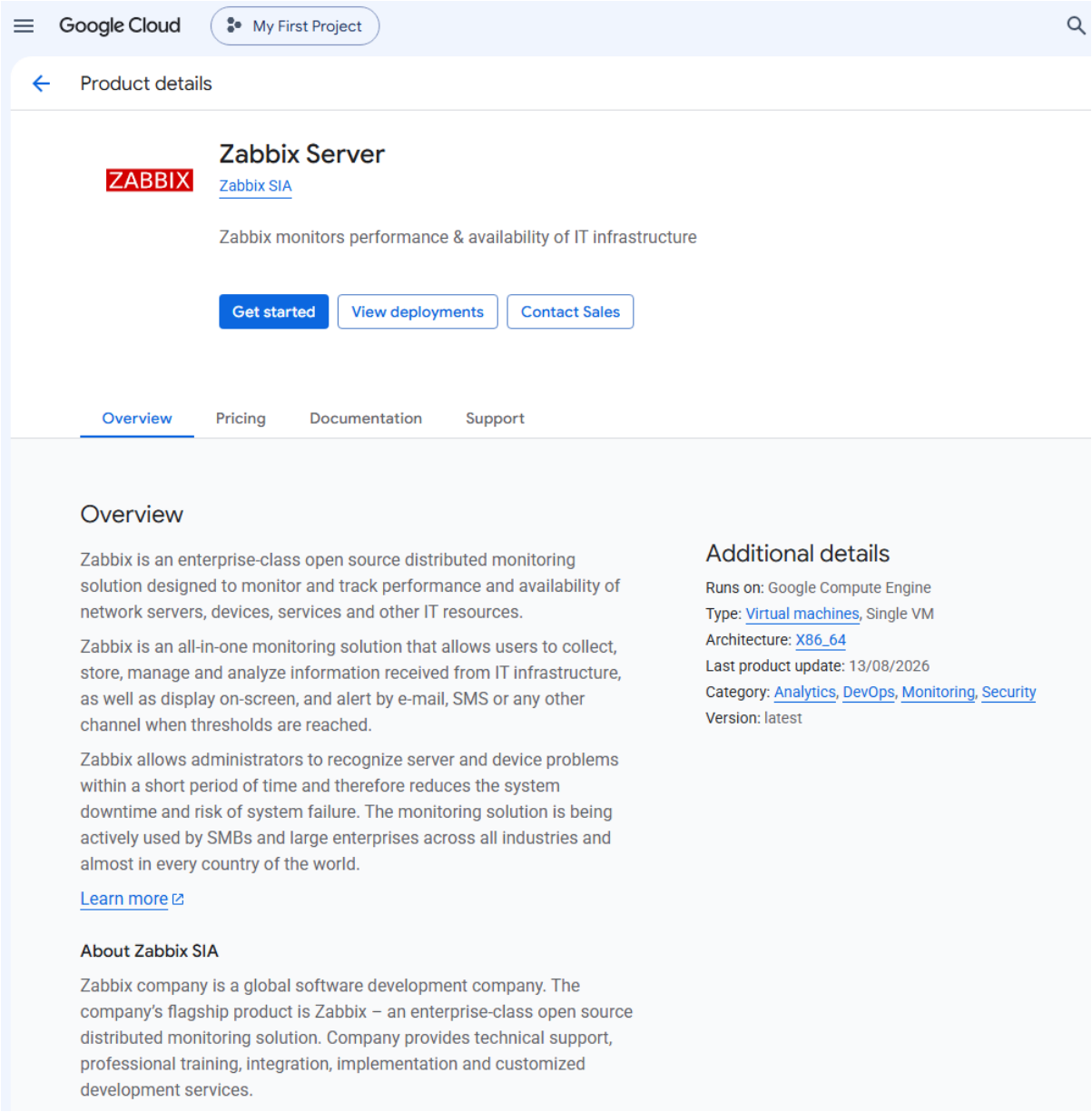
このページでは、Google Cloud Marketplace を使用して Zabbix をデプロイする方法について説明します。

Attention:
このプラットフォームでは、複数の Zabbix バージョンをデプロイできます。デプロイした Zabbix と同じバージョンのドキュメントを必ずお読みください。異なる Zabbix バージョンのドキュメントは、バージョンのドロップダウンメニューから選択できます。

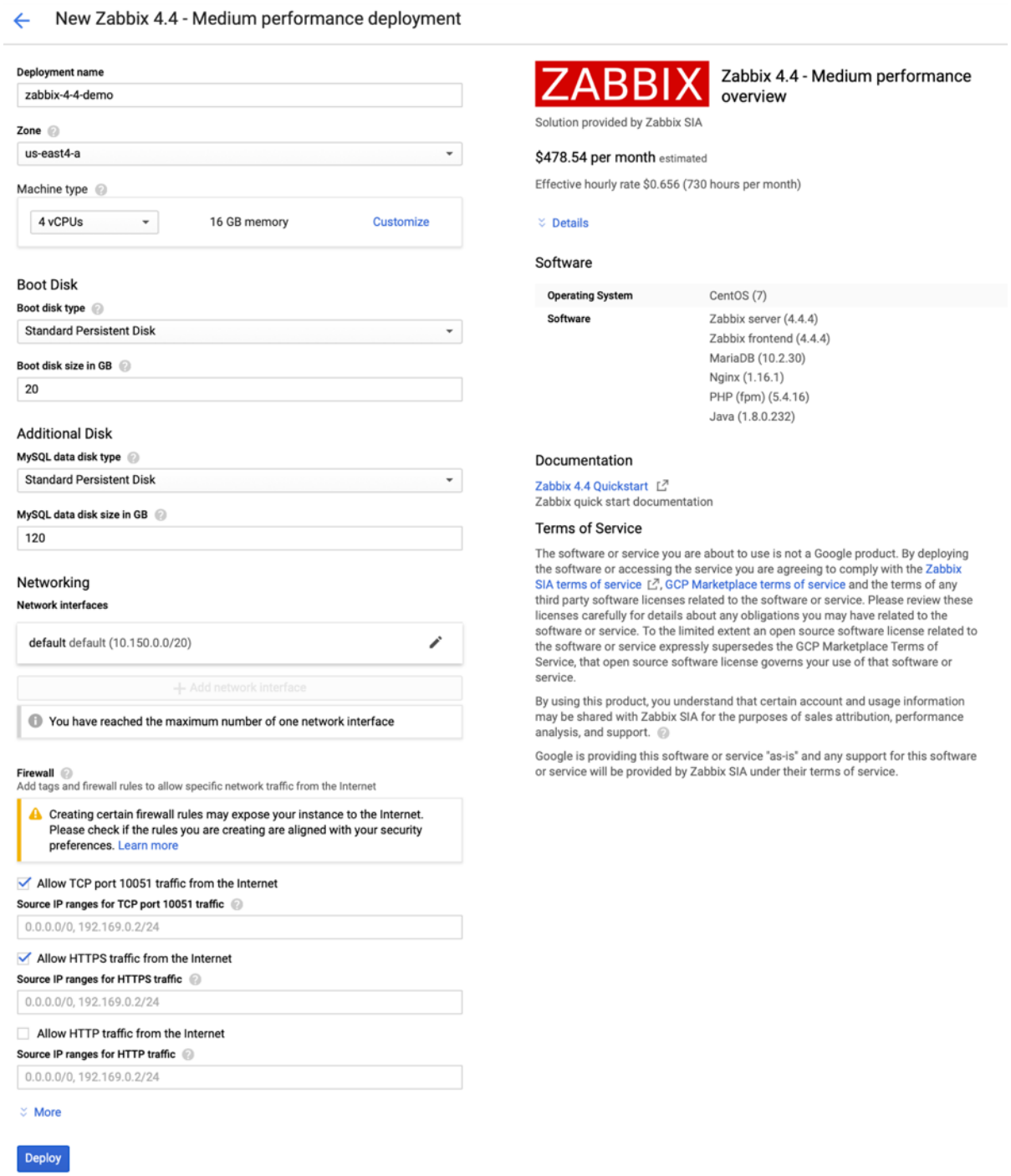
アプライアンスのデプロイ

まず、Zabbix アプライアンスをデプロイします。Google Cloud Marketplace を使用すれば、非常に簡単です。Marketplace でZabbix Server製品に移動します。

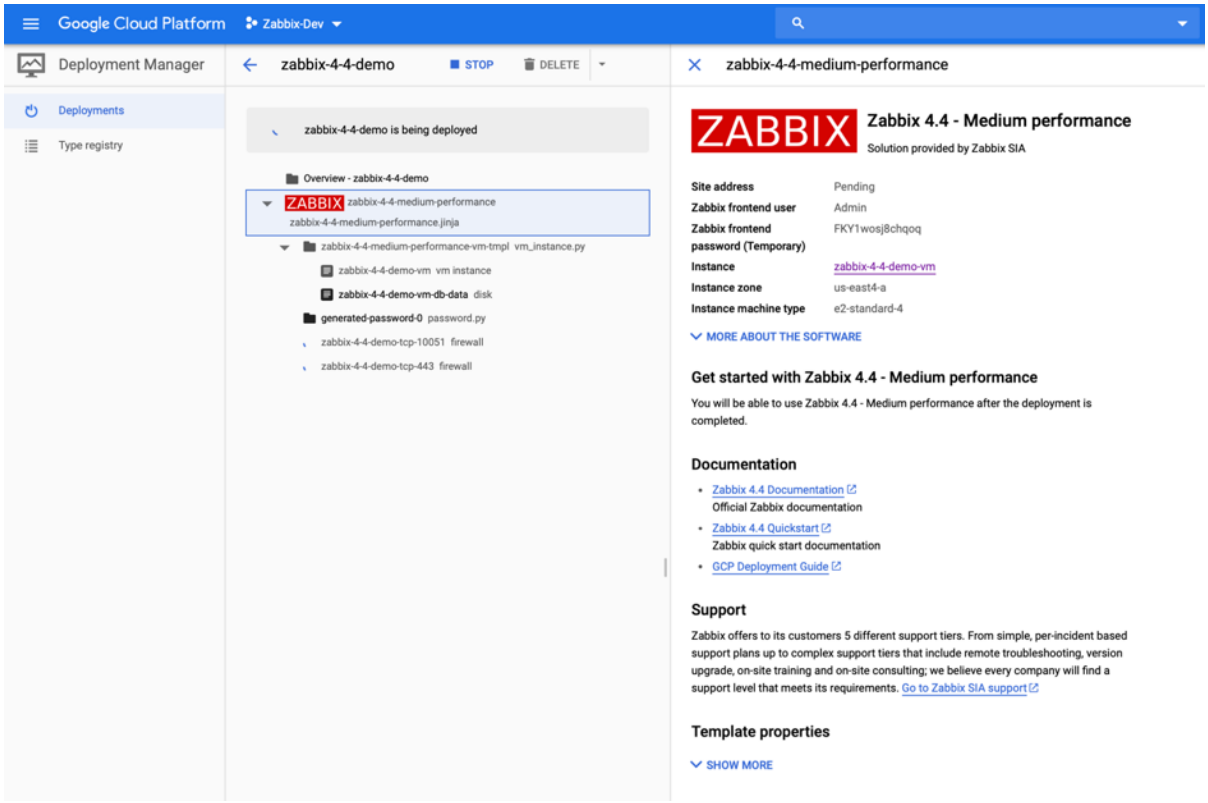
Get started をクリックします。



デフォルト設定を使用することも、カスタマイズすることもできます。完了したら、*Deploy* をクリックします。



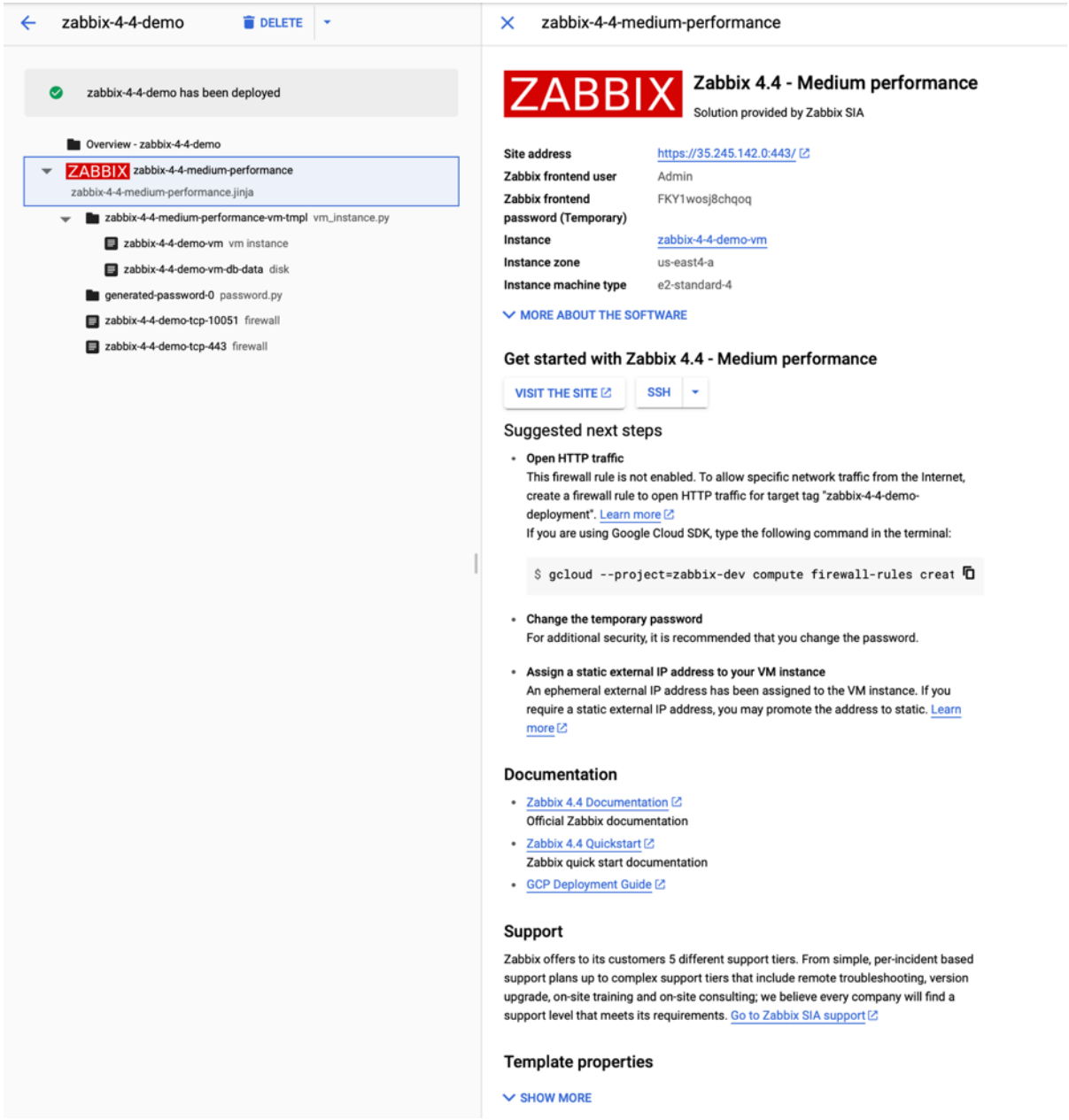
デプロイが進行中です。完了するまで待ちます。



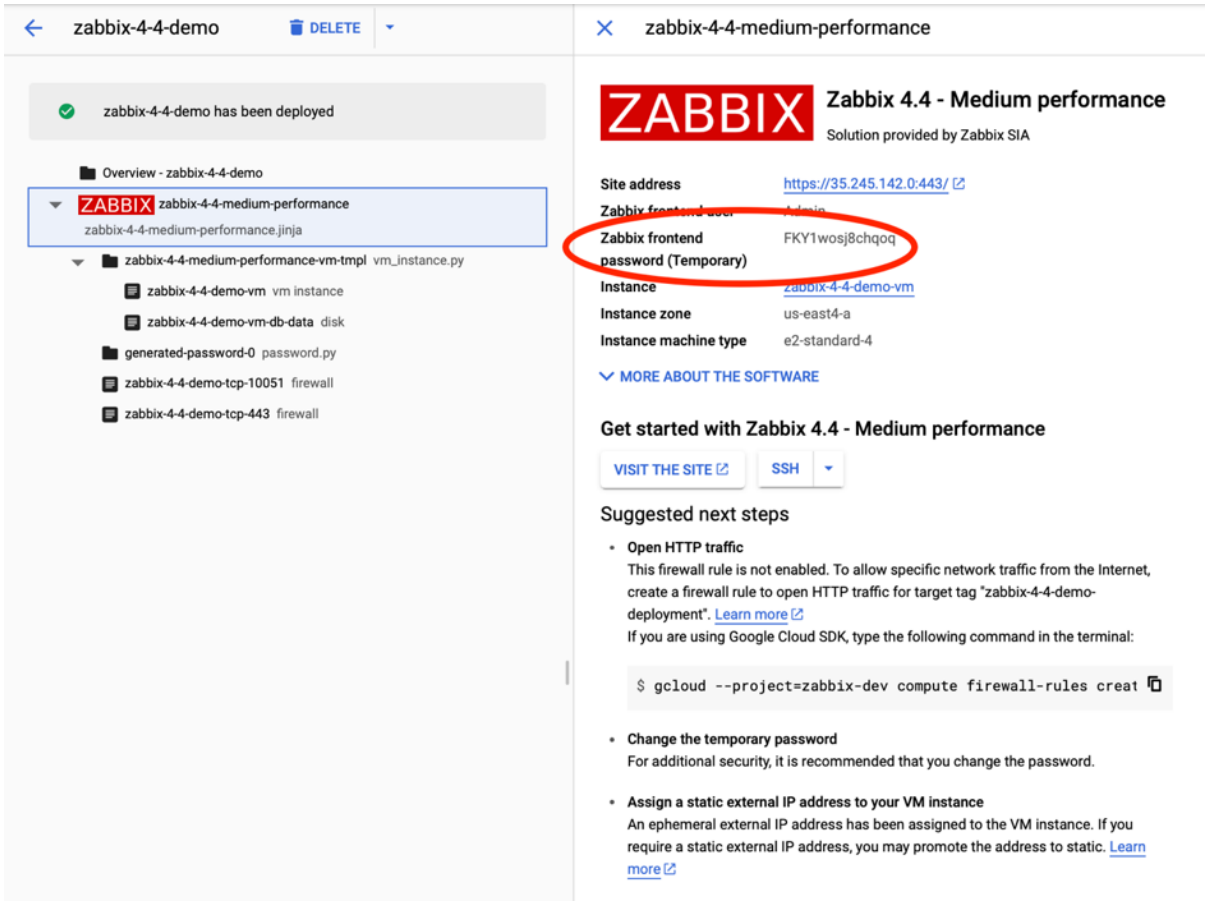
これで完了です。デプロイが開始されました。

アプライアンスへのアクセス

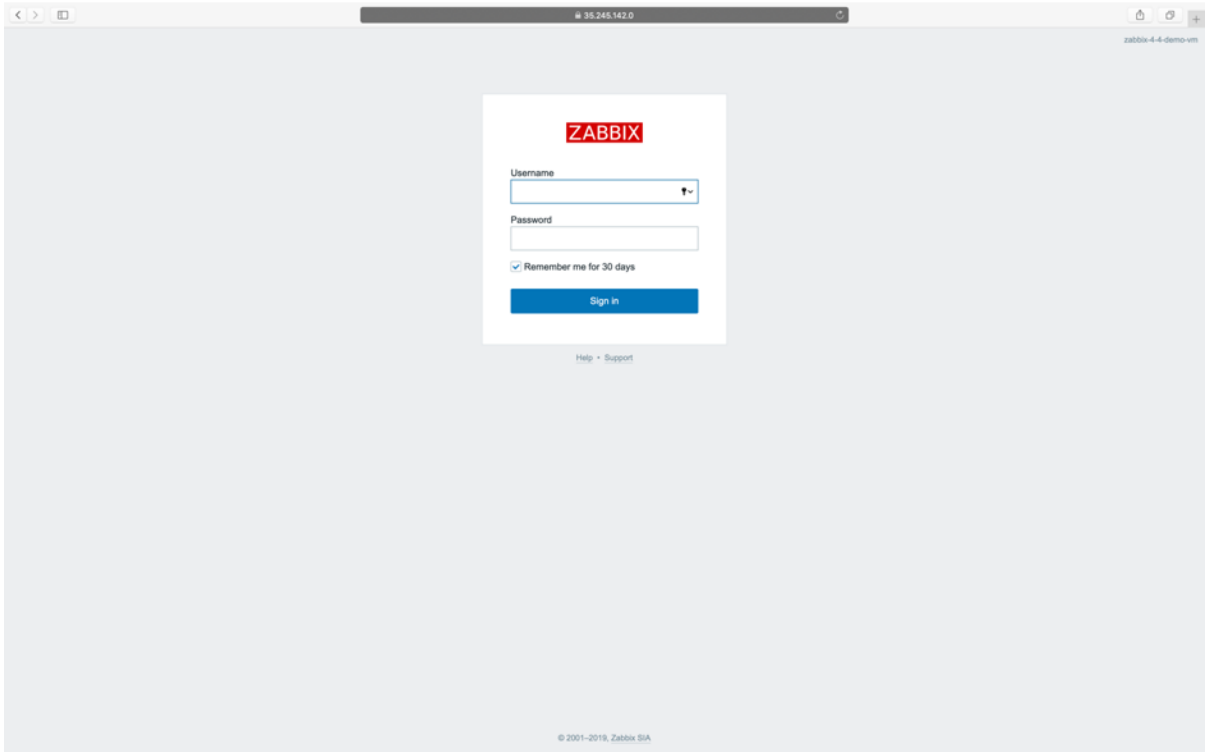
デプロイが完了すると、アプライアンスがデプロイ中であることを示すメッセージが表示されます。

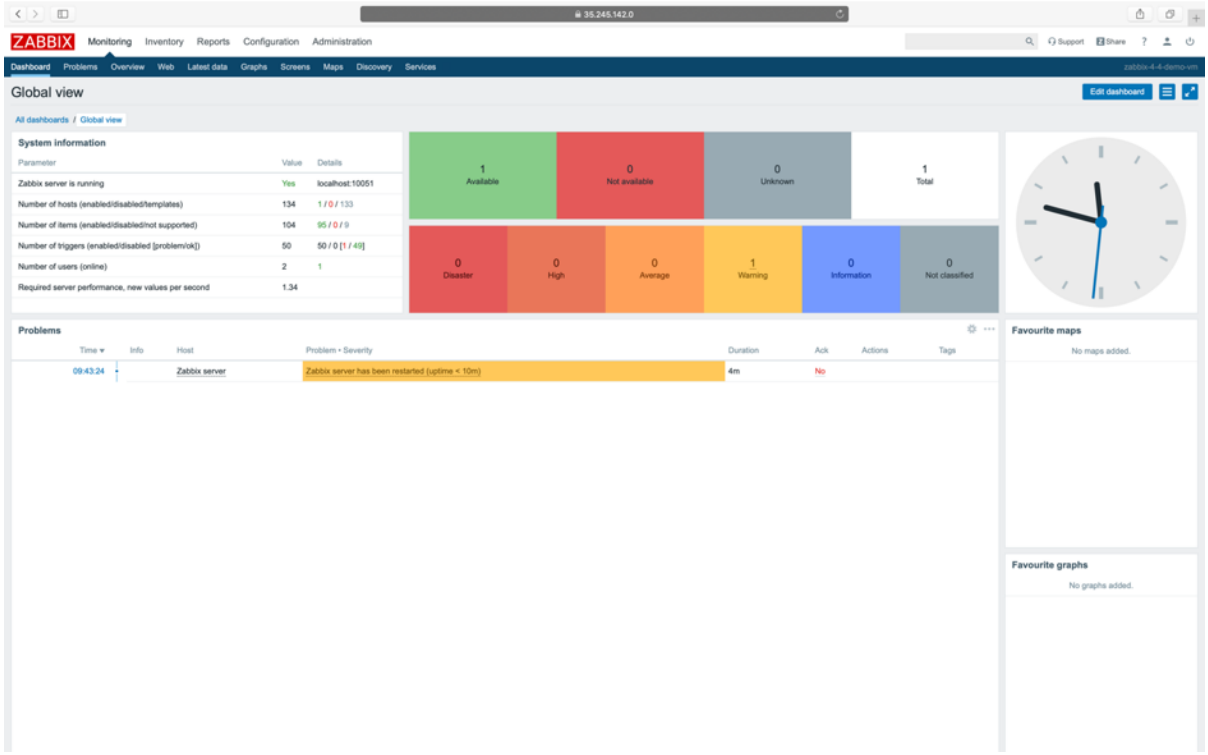


Web ブラウザーで *Site Address* を開き、Zabbix の Web インターフェースを表示します。その前に、Zabbix アプライアンスにログインするため、Zabbix の *Admin* ユーザーのパスワードを取得します。

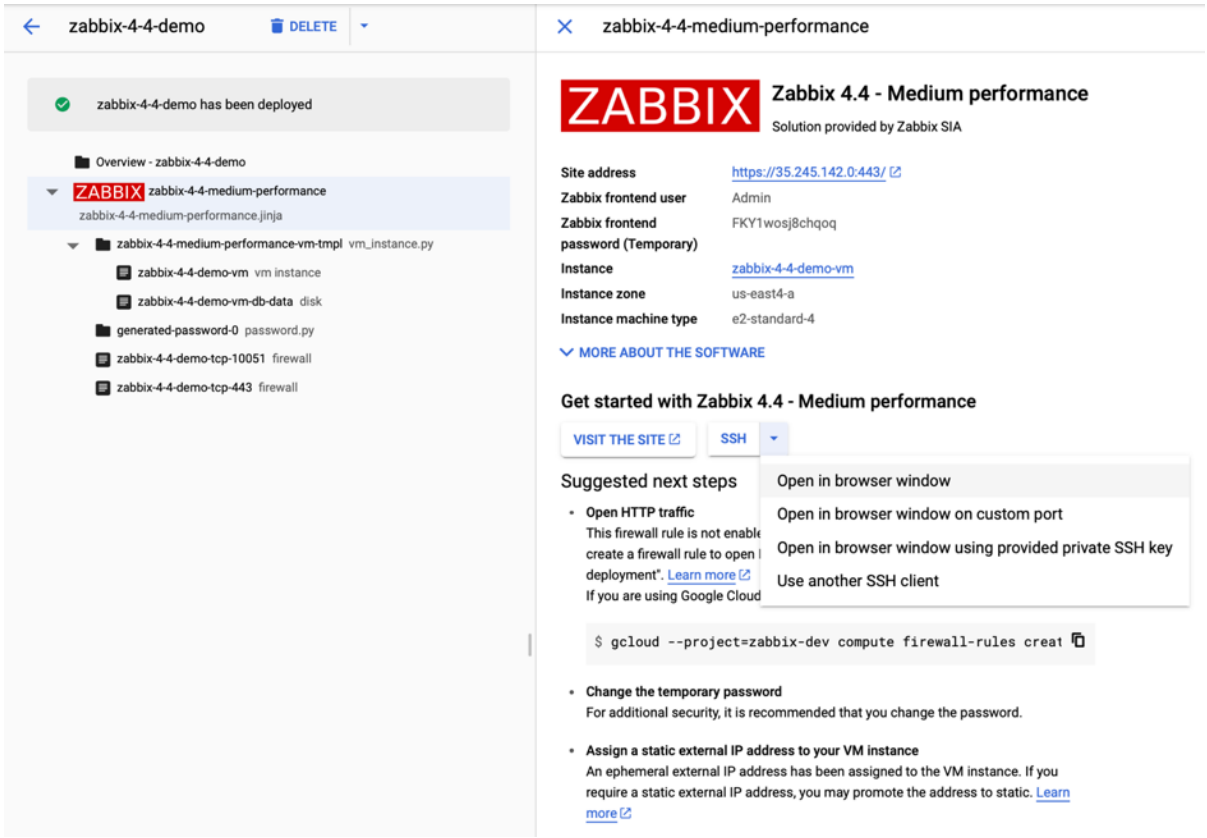


これで、Zabbix アプライアンスを使用できるようになりました。





インスタンスにログインして、データベースを操作することもできます。そのためには、Google コンソールに戻り、以下に示す赤い矢印に従って、*Open in browser window* オプションを使用して SSH セッションを開始します。



MySQL は `root` パスワードなしで使用できます。パスワードは `/root/.my.cnf` 設定ファイルに保存されています。デフォルトでは、Zabbix アプライアンスは UTC タイムゾーンを使用します。変更する必要がある場合は、`/etc/php.d/99-zabbix.ini` の `date.timezone` PHP 設定ディレクティブを編集し、php-fpm プロセスを再起動します: `systemctl restart php-fpm`。

アップグレード

アップグレードする前に、関連するすべてのアップグレードノートを確認することを強く推奨します。

また、どのコンポーネントをアップグレードする必要があるかを把握しておくと役立ちます。

概要

以下の環境について、ステップごとのアップグレード手順を利用できます。

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

関連する手順：

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

アップグレード対象のコンポーネント

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

Zabbix プロキシのアップグレードは強く推奨されます。
Zabbix サーバーは、サーバーと同じメジャーバージョンのプロキシを完全にサポートします。
また、Zabbix サーバーは、Zabbix サーバーの以前の LTS リリースバージョンより古くないプロキシもサポートしますが、機能は制限されます（データ収集、リモートコマンドの実行、アイテム値の即時チェック）。

設定の更新も無効になり、古いプロキシは古い設定でのみ動作します。

Attention:
Zabbix サーバーの前回の LTS リリースバージョンより古い、または Zabbix サーバーのメジャーバージョンより新しいプロキシはサポートされません。Zabbix サーバーはサポート対象外のプロキシからのデータを無視し、Zabbix サーバーとのすべての通信は警告とともに失敗します。詳細については、バージョンの互換性を参照してください。

アップグレード中のダウンタイムとデータ損失を最小限に抑えるため、Zabbix サーバーを停止、アップグレード、起動してから、Zabbix プロキシを 1 台ずつ順番に停止、アップグレード、起動することを推奨します。サーバーのダウンタイム中も、稼働中のプロキシはデータ収集を継続します。サーバーが稼働すると、古いバージョンのプロキシは新しいサーバーにデータを送信し（ただし、プロキシの設定は更新されません）、一部の機能を引き続き使用できます。Zabbix サーバーのダウンタイム中に発生した障害に関する通知は、アップグレードされたサーバーが起動した後にのみ生成されます。

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

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

データベースのサイズによっては、データベースをバージョン 8.0 にアップグレードするのに長い時間がかかる場合があります。

アップグレードに関する注意事項

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

Note:
アップグレード後、外部ソフトウェアがアップグレード後の Zabbix バージョンと互換性がない場合、一部のサードパーティ製ソフトウェア統合に影響が生じる可能性があります。

以下のアップグレードノートを利用できます。

アップグレード元	アップグレードノート 全文を読む	バージョン間の主な変更点
7.4.x	対象: Zabbix 8.0	1.6 より前のバージョンの Zabbix エージェントはサポートされなくなりました。 必要なデータベースの最小バージョンが引き上げられました。 必要な PHP の最小バージョンが 8.0.0 から 8.2.0 に引き上げられました。 % 文字が、Zabbix エージェントおよびZabbix エージェント 2の `UnsafeUserParameters` リストに追加されました。

アップグレード元	アップグレードノート 全文を読む	バージョン間の主な変更点
7.2.x	対象: Zabbix 7.4 Zabbix 8.0	PCRE ライブラリのサポートが廃止されました。
7.0.x	対象: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Oracle DB のサポートが廃止されました。
6.4.x	対象: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な PHP の最小バージョンが 7.4.0 から 8.0.0 に引き上げられました。 エージェント、HTTP エージェント、SNMP walk[OID]、SNMP get[OID] チェック用の非同期ポーラー。 プロキシ用の個別のデータベーステーブル。 Windows エージェント設定ファイルのデフォルトの場所が変更されました。 Oracle DB が非推奨になりました。 旧数値（浮動小数点）値タイプが廃止されました。
6.2.x	対象: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な MySQL の最小バージョンが 8.0.0 から 8.0.30 に引き上げられました。 Zabbix サーバー/プロキシには `libevent_pthreads` ライブラリが必要です。 アップグレード後の初回起動時に、SQLite3 を使用する Zabbix プロキシは、古いバージョンのデータベース（すべての履歴を含む）を自動的に削除し、新しいデータベースを作成します。
6.0.x LTS	対象: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な PHP の最小バージョンが 7.2.5 から 7.4.0 に引き上げられました。 サービス監視が大幅に再設計されました。 アップグレード中に決定論的トリガーを作成する必要があります。 MySQL/MariaDB でバイナリロギングが有効になっている場合、スーパーユーザー権限または変数/設定パラメータ *log_bin_trust_function_creators = 1* の設定が必要です。変数の設定方法については、データベース作成スクリプトを参照してください。
5.4.x	対象: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要なデータベースの最小バージョンが引き上げられました。 データベースが古い場合、サーバー/プロキシは起動しません。 データベース構造の変更により、監査ログの記録が失われました。
5.2.x	対象: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要なデータベースの最小バージョンが引き上げられました。 集計アイテムが個別のタイプとしては削除されました。
5.0.x LTS	対象: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な PHP の最小バージョンが 7.2.0 から 7.2.5 に引き上げられました。 パスワードのハッシュアルゴリズムが MD5 から bcrypt に変更されました。

アップグレード元	アップグレードノート 全文を読む	バージョン間の主な変更点
4.4.x	対象: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	IBM DB2 のサポートが廃止されました。 必要な PHP の最小バージョンが 5.4.0 から 7.2.0 に引き上げられました。 必要なデータベースの最小バージョンが引き上げられました。 Zabbix PHP ファイルのディレクトリが変更されました。
4.2.x	対象: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Jabber、Ez Texting のメディアタイプが削除されました。
4.0.x LTS	対象: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	古いプロキシは、アップグレードされたサーバーにデータを報告できなくなりました。 新しいエージェントは、古い Zabbix サーバーで動作できなくなりました。
3.4.x	対象: Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	`libpthread` および `zlib` ライブラリが必須になりました。 プレーンテキストプロトコルのサポートが廃止され、ヘッダーが必須になりました。 1.4 より前のバージョンの Zabbix エージェントはサポートされなくなりました。 パッシブプロキシ設定の Server パラメータが必須になりました。

アップグレード元	アップグレードノート 全文を読む	バージョン間の主な変更点
3.2.x	対象: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Zabbix サーバー/Web インターフェースのバックエンドデータベースとしての SQLite サポートが廃止されました。 POSIX 拡張正規表現の代わりに、Perl 互換正規表現（PCRE）がサポートされます。 Zabbix サーバーには `libpcre` および `libevent` ライブラリが必須です。 ユーザーパラメータ、リモートコマンド、「nowait」フラグのない system.run[] アイテム、および Zabbix サーバーで実行されるスクリプトに対して、終了コードのチェックが追加されました。 新しい機能をサポートするには、Zabbix Java ゲートウェイをアップグレードする必要があります。
3.0.x LTS	対象: Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	データベースのアップグレードは、履歴テーブルのサイズによっては時間がかかる場合があります。
2.4.x	対象: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な PHP の最小バージョンが 5.3.0 から 5.4.0 に引き上げられました。 LogFile エージェントパラメータを指定する必要があります。

アップグレード元	アップグレードノート 全文を読む	バージョン間の主な変更点
2.2.x LTS	対象: Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	ノードベースの分散監視が削除されました。
2.0.x	対象: Zabbix 2.2 Zabbix 2.4 Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	必要な PHP の最小バージョンが 5.1.6 から 5.3.0 に引き上げられました。 サーバーを正常に動作させるには、大文字と小文字を区別する MySQL データベースが必要です。Zabbix サーバーが MySQL データベースで正常に動作するには、文字セット utf8 および utf8_bin 照合順序が必要です。 データベース作成スクリプトを参照してください。 `mysql` の代わりに `mysqli` PHP 拡張機能が必要です。

パッケージからのアップグレード

概要

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

- Red Hat Enterprise Linux
- Debian/Ubuntu

OS リポジトリからの Zabbix パッケージ

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

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

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

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

Debian/Ubuntu

概要

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

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

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

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

アップグレード手順

1 Zabbix プロセスの停止

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

```
systemctl stop zabbix-server
```

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

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

2 Zabbix データベースのバックアップ

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

3 Zabbix の設定ファイル、PHP ファイル、および Zabbix バイナリをバックアップする

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

設定ファイルについては、次を実行します。

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
##### Apache の場合:
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
##### Nginx の場合:
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.conf /opt/zabbix-backup/
```

PHP ファイルと Zabbix バイナリについては、次を実行します。

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

4 リポジトリ設定パッケージを更新する

アップグレードを続行する前に、現在の Zabbix リポジトリパッケージをアンインストールします。

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

また、アップグレード中にパッケージマネージャーが古いバージョンを再利用しないように、新しいパッケージをダウンロードする前に、作業ディレクトリ内の古い Zabbix パッケージを手動で削除する必要がある場合があります（例: `rm zabbix-release_latest+debian12_all.deb`）。

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

Debian 13 では、次を実行します。

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian13_all.deb
```

Debian 12 では、次を実行します。

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

> **Note:**
> 古い Debian バージョンでは、上記のリンクを Zabbix repository の正しいものに置き換えてください。ただし、これらのバージョン向けのパッケージにはすべての Zabbix コンポーネントが含まれていない場合があるため、パッケージでそれらのコンポーネントをアップグレードするには、OS のアップグレードを検討してください。含まれるコンポーネントの一覧については、Zabbix packages を参照してください。

Ubuntu 26.04 では、次を実行します。

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu26.04_all.deb
```

Ubuntu 24.04 では、次を実行します。

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

Ubuntu 22.04 では、次を実行します。

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

> **Note:**
> 古い Ubuntu バージョンでは、上記のリンクを Zabbix repository の正しいものに置き換えてください。ただし、これらのバージョン向けのパッケージにはすべての Zabbix コンポーネントが含まれていない場合があるため、パッケージでそれらのコンポーネントをアップグレードするには、OS のアップグレードを検討してください。含まれるコンポーネントの一覧については、Zabbix packages を参照してください。

Zabbix リポジトリ設定に関するプロンプトが表示される場合があります。

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

Y（または I）を入力して、Zabbix リポジトリ設定のパッケージメンテナー版をインストールします。

その後、リポジトリ情報を更新します。

```
apt update
```

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次のコマンドを実行します。

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

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

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

Zabbix サーバー（またはプロキシ）の設定について、次のようなプロンプトが表示される場合があります。

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

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

次に、Zabbix Web インターフェースをアップグレードし、Web サーバーを再起動します。

Apache の場合は、次を実行します。

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

Nginx の場合は、次を実行します。

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Nginx の場合、nginx.conf 設定ファイルについて、次のようなプロンプトが表示される場合があります。状況に最も適したオプションを入力します。

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

6 コンポーネントの設定パラメータを確認する

必要に応じて設定パラメータの変更が必要かどうかを確認するため、関連するアップグレードノートを確認してください。

新しいオプションパラメータについては、What's newページを参照してください。

Web サーバーとして Nginx を使用しており、アップグレード中に /etc/zabbix/nginx.conf 設定ファイルが再インストールされた場合は、ファイルを確認し、listen および server_name ディレクティブを Zabbix Web インターフェースの設定に合わせて更新してください。

7 Zabbix プロセスの起動

アップグレードされた Zabbix コンポーネントを起動します。

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

8 Web ブラウザのクッキーとキャッシュのクリア

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

マイナーバージョン間のアップグレード

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

まず、リポジトリ情報を更新します:

```
apt update
```

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

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

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

Red Hat Enterprise Linux

概要

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

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

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

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

アップグレード手順

1 Zabbix プロセスの停止

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

```
systemctl stop zabbix-server
```

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

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

2 Zabbix データベースのバックアップ

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

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

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

設定ファイルの場合は、次を実行します。

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

PHP ファイルと Zabbix バイナリの場合は、次を実行します。

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

4 リポジトリ設定パッケージの更新

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

RHEL 10 では、次を実行します:

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

RHEL 9 では、次を実行します:

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

> **Note:**
> 古い RHEL バージョンまたはその派生ディストリビューションでは、上記のリンクを Zabbix repository にある正しいものに置き換えてください。ただし、これらのバージョン向けパッケージには、すべての Zabbix コンポーネントが含まれていない場合があります。これらのコンポーネントをパッケージからアップグレードするには、OS のアップグレードを検討してください。含まれているコンポーネントの一覧については、Zabbix packages を参照してください。

次に、dnf パッケージマネージャーのキャッシュをクリーンアップします（以前のインストールまたは更新時にダウンロードされたヘッダー、メタデータ、パッケージファイルを含む）:

```
dnf clean all
```

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

関連項目: RHEL でリポジトリ設定パッケージを更新する方法については、Known issues を参照してください。

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

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

- PostgreSQL を使用している場合は、コマンド内の `mysql` を `pgsql` に置き換えてください。
- プロキシをアップグレードする場合は、コマンド内の `server` を `proxy` に置き換えてください。
- エージェント 2 をアップグレードする場合は､コマンド内の `zabbix-agent` を `zabbix-agent2 zabbix-agent2-plugin-*` に置き換えてください。
- 他の Zabbix パッケージがインストールされている場合は、それらもコマンドに含めてアップグレードすることを検討してください。

> **Attention:**
> `dnf install zabbix-agent2` コマンドで Zabbix agent 2 をアップグレードすると、エラーが発生する可能性があります。詳細については、既知の問題を参照してください。

次に、Apache を使用して Zabbix Web インターフェースをアップグレードし、Apache を再起動するには、次を実行します。

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

6 コンポーネントの設定パラメータを確認する

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

新しい任意パラメータについては、新機能ページを参照してください。

7 Zabbix プロセスの起動

アップグレードされた Zabbix コンポーネントを起動します。

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

8 Web ブラウザのクッキーとキャッシュのクリア

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

マイナーバージョン間のアップグレード

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

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

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

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

アップグレードの問題

エスケープ関連　アップグレード後の関数パラメータの文字数制限超過

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

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

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

セットアップ

1. Zabbix 6.0 バージョン。
2. 末尾に'a' が付いた 114 個のバックスラッシュを含む /tmp/ONE ファイル:

```
\\\
```

```
  in trigger: 'TRIG_228'.
Current parameter value:
'$,1m:now,"regexp","\\\
```

Calc_228:

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

Calc_232:

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

コンテナからのアップグレード

概要

このセクションでは、Zabbix **8.0**.x コンテナへの正常なアップグレードに必要な手順について説明します。

以下については、それぞれ個別の手順が用意されています。

- Zabbix コンポーネントイメージのアップグレード
- Docker Compose ファイルのアップグレード

Warning:
アップグレードを行う前に、関連するアップグレードに関する注意事項を必ず確認してください。

Attention:
アップグレードを開始する前に、アップグレード作業のために、ユーザーがデータベースに対する必要な権限を持っていることを確認してください。

 Zabbix 6.0 以前からアップグレードする場合、アップグレード中に deterministic trigger を作成する必要があります。MySQL/MariaDB でバイナリログが有効になっている場合、これにはスーパーユーザー権限、または変数/設定パラメータ `log_bin_trust_function_creators = 1` の設定が必要です。変数の設定方法については、データベース作成スクリプトを参照してください。

 コンソールから実行した場合、この変数は一時的にのみ設定され、Docker が再起動されると失われることに注意してください。この場合は、SQL サービスは実行したままにし、`docker compose down zabbix-server` を実行して `zabbix-server` サービスのみを停止し、その後 `docker compose up -d zabbix-server` を実行してください。

 または、この変数を設定ファイルで設定することもできます。

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

Zabbix コンポーネントイメージのアップグレード

この手順では、Zabbix コンポーネントのコンテナのベースとして使用している Docker イメージをアップグレードします。

以下の手順では、Zabbix プロキシを例として使用します。
`zabbix-proxy-sqlite3` を、対象コンポーネントのコンテナ名に置き換えてください。

1. すべてのコンテナを確認して、名前と現在のイメージバージョンを特定します:

```
docker ps -a

#### CONTAINER ID   IMAGE                                              (...) NAMES
#### 5a0c2e3d02b4   zabbix/zabbix-proxy-sqlite3:alpine-7.4-latest      (...) zabbix-proxy-sqlite3
```

2. 新しいイメージバージョンを取得します:

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Zabbix コンポーネントのコンテナを停止します:

```
docker stop zabbix-proxy-sqlite3
```

4. Zabbix コンポーネントのコンテナを削除します:

Attention:
続行する前に、元の `docker run` コマンドで使用したすべての設定フラグとボリュームマウントを必ず記録しておいてください。コンテナを再作成する際に必要になります。

```
docker rm zabbix-proxy-sqlite3
```

5. 更新されたイメージを使用して新しいコンテナを起動します:

```
docker run --name zabbix-proxy-sqlite3 \
  -e ZBX_SERVER_HOST=192.0.2.1 \
```

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

Note:
Zabbix コンテナの実行方法の詳細については、Docker (manual deployment) を参照してください。

6. コンテナログを確認して、アップグレードが成功したことを検証します:

```
docker logs -f zabbix-proxy-sqlite3
```

Docker Compose ファイルのアップグレード

これらの手順では、Zabbix コンポーネントのデプロイに使用する Docker Compose ファイルをアップグレードします。
Zabbix をDocker Compose を使用してインストールした場合は、これらの手順に従ってください。

1. すべてのコンテナを確認して、名前と現在のイメージバージョンを特定します:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest   (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest (...) zabbix-docker-zabbix-web-ngina
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest   (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                               (...) zabbix-docker-mysql-server-1
```

2. 以前にクローンした Zabbix Docker リポジトリに移動し、最新の更新を取得して、8.0 ブランチに切り替えます:

Attention:
ローカルリポジトリ内の .env、compose.yaml、またはその他のファイルを変更している場合、git pull によってそれらが上書きされる可能性があります。
この手順を実行する前に、ローカルの変更をバックアップしてください。

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. 更新された Docker Compose ファイルを使用して Zabbix コンポーネントをデプロイします:

```
#### データベースとして MySQL を使用する場合:
docker compose -f ./compose.yaml up -d

#### データベースとして PostgreSQL を使用する場合:
docker compose -f ./compose_pgsql.yaml up -d
```

Note:
すべてのコンテナが再作成されて起動した後、データ収集が再開されるまで 1〜3 分かかる場合があります。
Zabbix コンテナの実行に関する詳細は、Docker Compose を参照してください。

4. コンテナログを確認して、アップグレードが成功したことを検証します（zabbix-docker-zabbix-server-1 は対象コンポーネントのコンテナ名に置き換えてください）:

```
docker logs -f zabbix-docker-zabbix-server-1
```

ソースからのアップグレード

概要

このセクションでは、Zabbix の公式ソースを使用して、Zabbix **7.4**.x から Zabbix **8.0**.x への正常なアップグレードに必要な手順を説明します。

Warning:
アップグレードする前に、関連するアップグレードに関する注意事項を必ずお読みください。

8.0 の要件も確認することをお勧めします。

Note:
アップグレード中は、2 つの SSH セッションを並行して実行すると便利です。一方でアップグレード手順を実行し、もう一方でサーバー/プロキシのログを監視します。例えば、2 つ目の SSH セッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログエントリや発生したエラーをリアルタイムで確認できます。これは本番環境のインスタンスでは非常に重要です。

サーバーのアップグレードプロセス

1 サーバーの停止

データベースに新しいデータが挿入されないように、Zabbix サーバーを停止します。

2 既存の Zabbix データベースをバックアップする

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

3 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

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

4 新しいサーバーバイナリをインストールする

これらの手順に従って、ソースから Zabbix サーバーをコンパイルします。

5 サーバー設定パラメータの確認

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

新しい任意パラメータについては、新機能ページを参照してください。

6 新しい Zabbix バイナリを起動する

新しいバイナリを起動します。
バイナリが正常に起動したことを確認するには、ログファイルを確認します。

Zabbix サーバーはデータベースを自動的にアップグレードします。
起動時に、Zabbix サーバーは現在のデータベースバージョン(必須およびオプション)と必要なデータベースバージョンを報告します。
現在の必須バージョンが必要なバージョンより古い場合、Zabbix サーバーは必要なデータベースアップグレードパッチを自動的に実行します。
データベースアップグレードの開始と進行状況(パーセンテージ)は、Zabbix サーバーのログファイルに記録されます。
アップグレードが完了すると、「database upgrade fully completed」というメッセージがログファイルに記録されます。
アップグレードパッチのいずれかが失敗した場合、Zabbix サーバーは起動しません。
現在の必須データベースバージョンが必要なバージョンより新しい場合も、Zabbix サーバーは起動しません。
現在の必須データベースバージョンが必要な必須バージョンと一致する場合にのみ、Zabbix サーバーは起動します。

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

サーバーを起動する前に、次の点を確認してください。

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

7 新しい Zabbix Web インターフェースをインストールする

必要な PHP の最小バージョンは 8.2.0 です。必要に応じて更新し、インストール手順に従ってください。

8 Web ブラウザーの Cookie とキャッシュをクリアする

アップグレード後、Zabbix Web インターフェースが正常に動作するように、Web ブラウザーの Cookie とキャッシュをクリアする必要がある場合があります。

プロキシのアップグレードプロセス

1 プロキシのを停止

Zabbix プロキシを停止します。

2 設定ファイルと Zabbix プロキシバイナリのバックアップ

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

3 新しいプロキシバイナリをインストールする

これらの手順に従って、ソースから Zabbix プロキシをコンパイルします。

4 プロキシ設定パラメータの確認

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

新しい任意パラメータについては、新機能ページを参照してください。

5 新しい Zabbix プロキシを起動する

新しい Zabbix プロキシを起動します。ログファイルを確認し、プロキシが正常に起動したことを確認します。

Zabbix プロキシはデータベースを自動的にアップグレードします。データベースのアップグレードは、Zabbix サーバーの起動時と同様に実行されます。

エージェントのアップグレード手順

> **Attention:**
> エージェントのアップグレードは必須ではありません。新しい機能にアクセスする必要がある場合にのみ、エージェントをアップグレードしてください。

このセクションで説明するアップグレード手順は、Zabbix エージェントと Zabbix エージェント 2 の両方のアップグレードに使用できます。

1 エージェントの停止

Zabbix エージェントを停止します。

2 設定ファイルと Zabbix エージェントバイナリのバックアップ

Zabbix エージェントバイナリと設定ファイルのバックアップコピーを作成します。

3 新しいエージェントバイナリをインストールする

手順に従って、ソースから Zabbix エージェントをコンパイルします。

または、Zabbix ダウンロードページからコンパイル済みの Zabbix エージェントをダウンロードできます。

4 エージェント設定パラメータの確認

設定パラメータに変更が必要かどうかを確認するため、Upgrade notes を必ず確認してください。

新しい任意パラメータについては、What's new ページを参照してください。

5 新しい Zabbix エージェントを起動する

新しい Zabbix エージェントを起動します。
エージェントが正常に起動したかどうかを確認するには、ログファイルを確認します。

マイナーバージョン間のアップグレード

8.0.x のマイナーバージョン間（例: 8.0.1 から 8.0.3）でアップグレードする場合、メジャーバージョン間のアップグレード時と同じ操作をサーバー/プロキシ/エージェントに対して実行する必要があります。唯一の違いは、マイナーバージョン間のアップグレードではデータベースに変更が加えられないことです。

既知の問題

参考: コンパイルの問題

アップグレード

正常なアップグレードのための SQL モード設定

MySQL/MariaDB の `sql_mode` 設定には、"STRICT_TRANS_TABLES" モードが設定されている必要があります。これが設定されていない場合、Zabbix データベースのアップグレードは失敗します（ZBX-19435も参照してください）。

MariaDB 10.2.1 以前でのアップグレード

MariaDB 10.2.1 以前でデータベーステーブルが作成された場合、Zabbix のアップグレードが失敗することがあります。これらのバージョンではデフォルトの行フォーマットが compact であるためです。この問題は、行フォーマットを dynamic に変更することで修正できます（ZBX-17690も参照）。

MySQL データベースの文字セットと照合順序

utf8mb3 文字セットおよび utf8mb3_bin 照合順序で作成された MySQL データベースを使用する Zabbix インストールでは、問題が発生する可能性があります。たとえば、既知の MySQL の問題により、データベーススキーマのアップグレードやその他の DDL 操作中に `Row size too large (> 8126)` エラーが発生する場合があります。
 Zabbix データベースの文字セットと照合順序の修復で説明されているように、utf8mb3 文字セットおよび utf8mb3_bin 照合順序を使用するデータベースを、utf8mb4 文字セットおよび utf8mb4_bin 照合順序に変換することを強く推奨します。

libcurl 8.21.0 におけるメモリおよびファイルディスクリプタのリーク

libcurl 8.21.0 の 既知の問題 により、TLS で保護された接続を処理する際に、libcurl を使用する Zabbix の機能 でメモリおよびファイルディスクリプタのリークが発生する可能性があります。
 この問題の回避策としては、修正を含むリリースが利用可能になるまで libcurl を 8.20.0 にダウングレードしてください。

テンプレート

デュアルスタック（IPv4/IPv6）環境におけるテンプレートの互換性

デュアルスタック環境（IPv4 と IPv6 の両方をサポートするように構成されたシステム）では、ホスト名 `localhost` は通常、IPv4 アドレスと IPv6 アドレスの両方に解決されます。多くのオペレーティングシステムや DNS リゾルバが IPv4 よりも IPv6 を優先するという一般的な優先順位付けのため、監視対象のサービスが IPv4 のみでリッスンするように構成されている場合、Zabbix テンプレートが正しく動作しないことがあります。

IPv6 アドレスでリッスンするように構成されていないサービスはアクセスできなくなり、監視が失敗する可能性があります。ユーザーは IPv4 で正しくアクセスを構成していても、IPv6 がデフォルトで優先される動作により接続の問題に直面することがあります。

この回避策としては、サービス（Nginx、Apache、PostgreSQL など）が IPv4 と IPv6 の両方のアドレスでリッスンするように構成されており、Zabbix サーバー/エージェントが IPv6 経由でアクセスできるようにすることです。さらに、Zabbix のテンプレートや設定では、`127.0.0.1` の代わりに `localhost` を明示的に使用して、IPv4 と IPv6 の両方との互換性を確保してください。

例えば、PostgreSQL by Zabbix agent 2 テンプレートで PostgreSQL を監視する場合、`zbx_monitor` ユーザーの接続を許可するために `pg_hba.conf` ファイルを編集する必要があるかもしれません。デュアルスタック環境で IPv6 が優先されている場合（システムが localhost を `::1` に解決する）に `localhost` を構成し、IPv4 エントリ（`127.0.0.1/32`）のみを追加した場合、対応する IPv6 エントリがないため接続は失敗します。

次の `pg_hba.conf` ファイルの例は、`zbx_monitor` ユーザーがローカルマシンから IPv4 および IPv6 アドレスの両方を使用して、異なる認証方式で任意のデータベースに接続できることを保証します。

```
### TYPE     DATABASE     USER            ADDRESS          METHOD
  host     all          zbx_monitor     localhost        trust
  host     all          zbx_monitor     127.0.0.1/32     md5
  host     all          zbx_monitor     ::1/128          scram-sha-256
```

必要に応じて、接続文字列のテンプレートマクロを構成する際に、PostgreSQL by Zabbix agent 2 テンプレートで IPv4 アドレス（`127.0.0.1`）を直接使用することもできます。

EPEL Zabbix パッケージの誤インストール

EPEL リポジトリがインストールされ有効になっている場合、Zabbix パッケージのインストール時に公式の Zabbix パッケージではなく EPEL バージョンがインストールされることがあります。これを解決するには:

1. EPEL からインストールされた Zabbix パッケージを削除します:

```
dnf remove zabbix-server-mysql
```

2. `/etc/yum.repos.d/epel.repo` ファイルに以下の行を追加して、EPEL から Zabbix パッケージを除外します:

```
[epel]
...
excludepkgs=zabbix*
```

3. 公式の Zabbix サーバーパッケージを再インストールします:

```
dnf install zabbix-server-mysql
```

インストール時､公式の Zabbix パッケージはバージョン文字列に `release` という単語が含まれており (例: `7.0.0-release1.el8`)、EPEL パッケージと区別できます。

Red Hat UBI 環境の RHEL 用 Zabbix パッケージ

Red Hat Universal Base Image環境で Red Hat Enterprise Linux パッケージから Zabbix をインストールする場合は、必要なリポジトリと依存関係にアクセスできることを確認してください。Zabbix パッケージは libOpenIPMI.so および libOpenIPMIposix.so ライブラリに依存していますが、これらは UBI システムで有効になっているデフォルトのパッケージマネージャーリポジトリのどのパッケージでも提供されていないため、インストールが失敗します。

libOpenIPMI.so および libOpenIPMIposix.so ライブラリは、redhat-#-for-<arch>-appstream-rpms リポジトリによって提供される OpenIPMI-libs パッケージで使用できます。このリポジトリへのアクセスはサブスクリプションによって管理され UBI 環境の場合、RHEL ホストのリポジトリ構成およびシークレットディレクトリをコンテナーファイルシステムの名前空間にマウントすることによって伝播されます。

詳細についてはZBX-24291を参照してください。

RHEL パッケージの署名鍵の有効期限切れ

Red Hat Enterprise Linuxまたはその派生ディストリビューションで Zabbix をアップグレードする際、Zabbix リポジトリのパッケージに対して署名鍵の有効期限切れの問題が発生する場合があります。署名鍵が有効期限切れになると、パッケージ署名の検証を試みても、証明書または鍵が有効でないことを示すエラーが発生します。例:

```
error: Verifying a signature using certificate D9AA84C2B617479C6E4FCF4D19F2475308EFA7DD (Zabbix LLC (Jul 2
  1. Certificate 19F2475308EFA7DD invalid: certificate is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
  2. Key 19F2475308EFA7DD invalid: key is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
```

このような問題を解決するには、RHEL のバリアントに合わせて最新の `zabbix-release` パッケージを手動で再インストールしてください（下記のリンクはZabbix リポジトリから正しいものに置き換えてください）。

例えば、**RHEL 10** の場合は次のコマンドを実行します:

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

その後、リポジトリ情報を更新します:

```
dnf update
```

詳細はZBX-24761を参照してください。

MariaDB を使用したデータベース TLS 接続

MariaDB を使用している場合、DBTLSConnect パラメーター の `verify_ca` オプションではデータベース TLS 接続はサポートされていません。

MySQL/MariaDB で発生する可能性のあるデッドロック

高負荷で動作し、複数の LLD ワーカーが関与している場合、行ロック戦略に関連する InnoDB エラーが原因でデッドロックが発生する可能性があります (上流バグを参照)。このエラーは MySQL 8.0.29 以降で修正されていますが、MariaDB では修正されていません。詳細はZBX-21506を参照してください。

グローバルイベントの相関

最初のイベントと 2 番目のイベントの間隔が非常に短い場合、つまり 0.5 秒以下の場合、イベントが正しく相関されないことがあります。

NetBSD 8.0 以降

NetBSD 8.X および 9.X バージョンでは、さまざまな Zabbix プロセスが起動時にランダムにクラッシュすることがあります。これは、デフォルトのスタックサイズ（4MB）が小さすぎるためであり、次のコマンドを実行して増やす必要があります。

```
ulimit -s 10240
```

詳細については、関連する問題レポートをご覧ください: ZBX-18275。

Zabbix エージェント 2 の正規表現の制限

Zabbix エージェント 2 は、標準の Go 正規表現ライブラリの制限により、正規表現の先読みと後読みをサポートしていません。

IPMI チェック

Debian 9（stretch）以前および Ubuntu 16.04（xenial）以前の標準 OpenIPMI ライブラリパッケージでは、IPMI チェックは動作しません。これを修正するには、ZBX-6139で説明されているように、OpenSSL を有効にして OpenIPMI ライブラリを再コンパイルしてください。

IPMI — 信頼できないホストによって OpenIPMI がクラッシュする可能性

Zabbix が IPMI データのポーリングに使用する OpenIPMI ライブラリには、信頼できないデバイスから特別に細工された応答を受信することで引き起こされるバグがあります。信頼できない IPMI デバイスが細工されたデータを送信すると、OpenIPMI ライブラリがクラッシュし、その結果、IPMI ポーリングを実行する Zabbix サーバープロセスが終了する可能性があります。

SSH チェック

- Debian、Ubuntu などの一部の Linux ディストリビューションでは、libssh2 ライブラリがパッケージからインストールされている場合、暗号化された秘密鍵（パスフレーズ付き）はサポートされません。詳細については、ZBX-4850をご覧ください。
- 一部の Linux ディストリビューションで OpenSSH 8 と libssh 0.9.x を使用している場合、SSH チェックで「Cannot read data from SSH server」と報告されることがあります。これは libssh の問題（詳細なレポート）が原因です。このエラーは、安定版 libssh 0.9.5 リリースで修正されているはずです。詳細については、ZBX-17756もご覧ください。
- SSH スクリプトでパイプ「|」を使用すると、「Cannot read data from SSH server」エラーが発生する場合があります。この場合は、libssh ライブラリのバージョンアップを推奨します。詳細については、ZBX-21337もご覧ください。

ODBC チェック

- 可能であれば、MariaDB コネクターライブラリに対してコンパイルされた Zabbix サーバーまたは Zabbix プロキシでは MySQL unixODBC ドライバーを使用せず、その逆も避けてください。また、upstream bug により、ドライバーとして同じコネクターを使用することも避けたほうがよいです。
  推奨構成:

  PostgreSQL、SQLite または Oracle コネクター → MariaDB または MySQL unixODBC ドライバー
MariaDB コネクター → MariaDB unixODBC ドライバー
MySQL コネクター → MySQL unixODBC ドライバー

  詳細および利用可能な回避策については、ZBX-7665 を参照してください。
- Microsoft SQL Server から照会した XML データは、Linux および UNIX システム上でさまざまな形で切り詰められる場合があります。
- Linux 向けのさまざまなバージョンの Oracle Instant Client を使用して Oracle データベースを監視するために ODBC チェックを使用すると、Zabbix サーバーがクラッシュすることが確認されています。
 関連情報: ZBX-18402, ZBX-20803。
- FreeTDS UnixODBC ドライバーを使用する場合は、SQL クエリの先頭に'SET NOCOUNT ON' 文を追加する必要があります（例: `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`）。
  そうしないと、Zabbix のデータベースモニターアイテムは次のエラーで情報の取得に失敗します。
  "SQL query returned empty result".
 詳細については、ZBX-19917 を参照してください。

アイテムのリクエストメソッドパラメータが正しくありません

HTTP チェックでのみ使用されるリクエストメソッドパラメータが、Zabbix 4.0 以前のバージョンからのアップグレードの結果として、すべてのアイテムでデフォルト以外の値「1」に誤って設定されている場合があります。この状況を修正する方法の詳細については、ZBX-19308を参照してください。

Web 監視と HTTP エージェント

Zabbix サーバーは、Web シナリオまたは HTTP エージェントで「SSL verify peer」が有効になっている場合、一部の Linux ディストリビューションで上流のバグによりメモリリークが発生します。詳細および利用可能な回避策については、ZBX-10486を参照してください。

シンプルチェック

fping v3.10 より前のバージョンには、重複したエコーリプレイパケットを誤って処理するバグがあります。これにより、`icmpping`、`icmppingloss`、`icmppingsec` アイテムで予期しない結果が発生する可能性があります。**fping** の最新バージョンを使用することを推奨します。詳細はZBX-11726をご覧ください。

rootless コンテナでの fping 実行時のエラー

コンテナが rootless モードまたは特定の制限環境で実行されている場合、ICMP チェックを実行する際に fping の実行に関連するエラー（例：`fping: Operation not permitted` や、すべてのリソースへのパケットが失われる）に直面することがあります。

この問題を修正するには、"docker run" または"podman run" コマンドに`--cap-add=net_raw` を追加してください。

さらに、非 root 環境で fping を実行するには sysctl の変更が必要な場合があります。例：

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

ここで"1995" は zabbix の GID です。詳細については、ZBX-22833を参照してください。

SNMP チェック

OpenBSD オペレーティングシステムを使用している場合、Net-SNMP ライブラリの 5.7.3 バージョンまでの use-after-free バグにより、Zabbix サーバーの設定ファイルで SourceIP パラメータが設定されていると、Zabbix サーバーがクラッシュする可能性があります。回避策として、SourceIP パラメータを設定しないでください。同じ問題は Linux にも当てはまりますが、Zabbix サーバーの動作

停止は引き起こしません。OpenBSD では、net-snmp パッケージにローカルパッチが適用されており、OpenBSD 6.3 でリリースされる予定です。

SNMP データのスパイク

SNMP データにスパイクが発生することがあり、これは主電源の電圧スパイクなどの物理的要因に関連している可能性があります。詳細はZBX-14318を参照してください。

SNMP トラップ

SNMP トラップに必要な”net-snmp-perl” パッケージは、RHEL 8.0-8.2 で削除され、RHEL 8.3 で再追加されました。

そのため、RHEL 8.0-8.2 を使用している場合は、RHEL 8.3 へのアップグレードが最善の解決策です。

詳細については、ZBX-17192 も参照してください。

RHEL 7 でのアラートプロセスのクラッシュ

RHEL 7 で Zabbix サーバーのアラートプロセスがクラッシュする事例が報告されています。詳細については、ZBX-10461 を参照してください。

Zabbix agent 2 (6.0.5 以前) のアップグレード

パッケージから Zabbix agent 2 (バージョン 6.0.5 以前) をアップグレードする際、プラグイン関連のファイル競合エラーが発生する場合があります。このエラーを修正するには、必要に応じて agent 2 の設定をバックアップし、agent 2 をアンインストールして再インストールしてください。

RHEL 系システムでは、以下を実行します:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Debian 系システムでは、以下を実行します:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

詳細はZBX-23250を参照してください。

Web インターフェースのロケールが切り替わる

Web インターフェースのロケールが明確な理由なく切り替わることがあります。つまり、一部のページ（またはページの一部）がある言語で表示される一方、別のページ（またはページの一部）が異なる言語で表示されます。通常、この障害は複数のユーザーが存在し、その一部があるロケールを使用し、他のユーザーが別のロケールを使用している場合に発生することがあります。

この問題の既知の回避策は、PHP と Apache でマルチスレッドを無効にすることです。

この障害は、PHPでのロケール設定の仕組みに関連しています。ロケール情報はスレッドごとではなく、プロセスごとに保持されます。そのため、マルチスレッド環境で同じ Apache プロセスによって複数のプロジェクトが実行されている場合、別のスレッドでロケールが変更され、その変更によって Zabbix のスレッドでデータが処理される方法が変わる可能性があります。

詳細については、関連する障害レポートを参照してください。

- ZBX-10911（Web インターフェースのロケールが切り替わる障害）
- ZBX-16297（BC Math 関数の `bcdiv` 関数を使用するグラフで数値処理に関する障害）

グラフ

グラフ（クラシック）の問題

クラシックグラフで問題が発生する場合は、GD ライブラリ（libgd）をバージョン 2.3.3-13 以降に、PHP をバージョン 8.0.19、8.1.33、8.2.29、8.3.25、8.4.12 以降に更新することを推奨します。

サマータイム

サマータイム（DST）の変更により、X 軸ラベルの表示に不規則性（日付の重複、日付の欠落など）が生じます。

合計集計

1 時間未満の期間のグラフで合計集計を使用すると、データがトレンドから取得される場合に、グラフに誤った（増幅された）値が表示されます。

テキストの重なり

一部の Web インターフェース言語（例: 日本語）では、ローカルフォントが原因でグラフの凡例内のテキストが重なることがあります。これを回避するには、PHP GD 拡張モジュールのバージョン 2.3.0 以降を使用してください。

ログファイルの監視

ファイルシステムが 100% フルで、ログファイルが追記されている場合、log[] および logrt[] アイテムはログファイルの先頭から繰り返し再読み込みします（詳細は ZBX-10884 を参照してください）。

遅い MySQL クエリ

Zabbix サーバーは、アイテムの値が存在しない場合に遅い SELECT クエリを生成します。この問題は MySQL 5.6/5.7 バージョンで発生することが知られています（詳細な議論についてはZBX-10652を参照）。また、特定の場合にはそれ以降の MySQL バージョンでも発生する可能性があります。これに対する回避策としては、MySQL でindex_condition_pushdownまたはprefer_ordering_indexオプティマイザを無効にすることです。ただし、この回避策ですべての遅いクエリに関連する問題が解決するわけではないことに注意してください。

リンクからのフィルター設定の永続化

フィルター設定（時間セレクターを含む）を含む Zabbix フロントエンドページへのリンクを開くと、そのフィルターは自動的にデータベースにユーザーごとに保存され、そのページの以前に保存されたフィルターや時間セレクターの設定が置き換えられます。これらの設定は、ユーザーが手動で更新またはリセットするまで有効です。

SNMPv3 トラップにおける IPv6 アドレスの問題

net-snmp のバグにより、SNMP トラップで SNMPv3 を使用する場合、IPv6 アドレスが正しく表示されないことがあります。詳細および回避策については、ZBX-14541を参照してください。

失敗したログイン情報での長い IPv6 IP アドレスの切り捨て

失敗したログイン試行メッセージでは、データベースフィールドの文字数制限が 39 文字のため、格納された IP アドレスの最初の 39 文字のみが表示されます。つまり、39 文字を超える IPv6 IP アドレスは不完全に表示されます。

Windows 上の Zabbix エージェントチェック

Zabbix エージェントの設定ファイル (zabbix_agentd.conf) の Server パラメータに存在しない DNS エントリが含まれていると、Windows 上で Zabbix エージェントの応答時間が長くなる場合があります。これは、Windows の DNS キャッシュデーモンが IPv4 アドレスに対する負の応答をキャッシュしないためです。ただし、IPv6 アドレスについては負の応答がキャッシュされるため、これを回避する方法として、ホストで IPv4 を無効にすることが考えられます。

YAML エクスポート/インポート

YAML エクスポート/インポート には、いくつか既知の問題があります。

- エラーメッセージは翻訳されません。
- .yaml ファイル拡張子を持つ有効な JSON が、インポートできない場合があります。
- 引用符で囲まれていない人間が読みやすい日付は、自動的に Unix タイムスタンプに変換されます。

SUSE で NGINX と php-fpm を使用するセットアップウィザード

Web インターフェースのセットアップウィザードは、SUSE で NGINX + php-fpm を使用している場合に設定ファイルを保存できません。これは、/usr/lib/systemd/system/php-fpm.service ユニット内の設定が原因で、Zabbix が /etc に書き込めないようになっているためです（PHP 7.4 で導入されました）。

回避策は 2 つあります。

- php-fpm の systemd ユニットで、ProtectSystem オプションを'full' ではなく'true' に設定します。
- /etc/zabbix/web/zabbix.conf.php ファイルを手動で保存します。

認証ヘッダーの転送

Apache または NGINX が、API リクエスト内の認証ヘッダーを Zabbix に転送できない場合があります。これにより、Zabbix API や Okta を使用した SAML といったシングルサインオン (SSO) サービスの使用時に認証の問題が発生する可能性があります。

この問題を解決するには、Web サーバーの設定を更新してください。

Apache をリバースプロキシ (CGI 以外の設定) として使用する場合は、(RHEL ベースシステムの場合) /etc/httpd/conf/httpd.conf または (Debian/Ubuntu の場合) /etc/apache2/apache2.conf に次のディレクティブを追加してください。

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Apache がリクエストを処理するためにスクリプトを直接実行する場合 (例: mod_cgi を使用)、代わりに次のディレクティブを追加してください。

```
CGIPassAuth On
```

一方、**NGINX** は Authorization ヘッダーを自動的に処理します。ただし、NGINX がリバースプロキシとして動作している場合は、/etc/nginx/nginx.conf (Zabbix フロントエンドの場所) に以下のディレクティブを追加することで、Authorization ヘッダーを明示的に転送できます。

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

設定を更新したら、Web サーバーを再起動してください。

詳細については、以下を参照してください:

- ZBX-22952
- Apache 2.4 + PHP-FPM および Authorization ヘッダー
- SetEnvIfNoCaseおよびCGIPassAuthディレクティブ
- NGINX リバースプロキシ

Ubuntu 20 での Zabbix Web サービス用 Chromium

ほとんどの場合、Zabbix Web サービスは Chromium で動作しますが、Ubuntu 20.04 で Chromium を使用すると、次のエラーが発生します。

```
Cannot fetch data: chrome failed to start:cmd_run.go:994:
WARNING: cannot create user data directory: cannot create
"/var/lib/zabbix/snap/chromium/1564": mkdir /var/lib/zabbix: permission denied
Sorry, home directories outside of /home are not currently supported. See https://forum.snapcraft.io/t/112
```

このエラーは、ユーザー'zabbix' のホームディレクトリとして/var/lib/zabbix が使用されているために発生します。

MySQL のカスタムエラーコード

Zabbix はバックエンドデータベースにアクセスできないことを検出すると、通知を送信し、接続を試み続けます。特定のデータベースエンジンでは、特定のエラーコードが認識されます。MySQL では、これらの認識されるエラーコードには以下が含まれます:

- CR_CONN_HOST_ERROR
- CR_SERVER_GONE_ERROR
- CR_CONNECTION_ERROR
- CR_SERVER_LOST
- CR_UNKNOWN_HOST
- ER_SERVER_SHUTDOWN
- ER_ACCESS_DENIED_ERROR
- ER_ILLEGAL_GRANT_FOR_TABLE
- ER_TABLEACCESS_DENIED_ERROR
- ER_UNKNOWN_ERROR

さらに、Azure 上の MySQL インストールで Zabbix を使用している場合、Zabbix ログに一般的なエラーメッセージ *[9002] Some errors occurred* が表示されることがあります。このメッセージは、データベースから Zabbix サーバーまたはプロキシに送信されます。エラーの原因を特定するには、Azure のログを参照してください。

PCRE2 への切り替え後に正規表現が無効になる場合

Zabbix 6.0 では PCRE2 のサポートが追加されました。PCRE は引き続きサポートされていますが、RHEL 7 以降、SLES（すべてのバージョン）、Debian 9 以降、Ubuntu 16.04 以降向けの Zabbix インストールパッケージは、PCRE2 を使用するように更新されています。多くの利点がある一方で、PCRE2 への切り替えにより、既存の一部の PCRE 正規表現パターンが無効になったり、動作が変わったりする場合があります。特に、これはパターン [1] に影響します。意味を変えずにこの正規表現を再び有効にするには、式を [2] に変更してください。これは、PCRE2 ではダッシュ記号が区切り文字として扱われ、文字クラス内に範囲が作成されるためです。

Geomap ウィジェットのエラー

NGINX を使用して古い Zabbix バージョンからアップグレードし、アップグレード時に新しい NGINX 設定ファイルへ切り替えなかった場合、Geomap ウィジェットの地図が正しく読み込まれないことがあります。

この問題を修正するには、古い設定ファイルを破棄し、現在のバージョンのパッケージに含まれる設定ファイルを使用して、ダウンロード手順 の *e. Configure PHP for Zabbix frontend* セクションに記載されているとおりに再設定します。

または、既存の NGINX 設定ファイル（通常は /etc/zabbix/nginx.conf）を手動で編集することもできます。
その場合は、ファイルを開いて次のブロックを見つけます。

[1] \w-\.
[2] -\w\.

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

次に、このブロックを以下に置き換えます。

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

前処理 — グローバル変数は安全ではありません

前処理の JavaScript はリクエストごとに実行されますが、宣言されていない識別子への代入（例: `secret = value`）は暗黙的なグローバル変数を作成し、現在の実行を超えて持続する可能性があります。暗黙的なグローバル変数に機密データ（トークン、パスワードなど）を保存すると、後続の前処理実行や同じ環境で実行される他の統合によって、偶発的な漏洩や再利用のリスクが高まります。

暗黙的なグローバル変数に依存しないでください。常に `var` や `const` で変数を宣言し、グローバルオブジェクト（例: `globalThis` や `window`）に秘密情報を付与しないでください。前処理内から組み込みのグローバルオブジェクトを上書きするサポートされた方法はありません。

安全な例:

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

Windows のプロセッサグループ

Microsoft のドキュメントによると、64 個未満の論理プロセッサを持つシステムには常に 1 つのプロセッサグループ（グループ 0）しかありません。しかし、Zabbix ユーザーからは、64 個以下の論理プロセッサを持つシステムで 2 つのプロセッサグループが存在するという稀なバグZBX-20260が報告されています。この結果、2 つのプロセッサグループのうち 1 つのプロセッサグループに対してのみ”\Processor(n)” パフォーマンスカウンタが存在することになります。このバグの実際の根本原因は不明です。しかし、stackoverflow.comで同様のケースが説明されており、そこでは BIOS と Windows の相互運用に根本原因がありました。

utf8mb4 照合順序によるフィルタリングの制限

フィルタ（例：データ収集 > メンテナンス）は、特定の Unicode 文字（例：𝒸、𝓅）を含むエンティティに適用した場合、正しく機能しないことがあります。この問題は、MySQL または MariaDB データベースのデフォルトの utf8mb4_bin 照合順序が、Unicode 文字のソートや比較をどのように処理するかによって発生します。

この制限に対処するには、データベースのカラムの照合順序を utf8mb4_0900_bin、utf8mb4_0900_ai_ci、utf8mb4_unicode_520_ci などの代替案に変更することができます。ただし、照合順序を変更すると、空白の扱いや他の文字のソートやフィルタリングで予期しない動作が発生する可能性があることに注意してください。

照合順序の変更についての詳細は、MySQL ドキュメントまたはMariaDB ドキュメントを参照してください。照合順序の違いについては、MySQL ドキュメントのUnicode 文字セットを参照してください。

tcmalloc を使用した過剰なメモリ使用量のプロファイリング

Zabbix のインストールでメモリを使いすぎている疑いがある場合は、tcmalloc のメモリプロファイリング機能を使用して、Zabbix サーバー/プロキシのメモリ消費を調査できます。

1. Zabbix をソースから インストールする際に、追加のフラグを設定します:

```
export CFLAGS="-std=gnu99 -g -O0"
```

`-std=gnu99` フラグは、Zabbix サーバー、Zabbix プロキシ、または Zabbix エージェントのビルドに必要です。
`-g` フラグは追加のデバッグ情報を付与し、`-O0` は最適化を無効にします。これは tcmalloc のプロファイリングに干渉する可能性があります。

2. Zabbix サーバーを起動する前に、次の環境変数を設定します。
これらの変数は、tcmalloc にメモリ使用量の追跡と報告方法を指示します:

```
LD_PRELOAD="/usr/lib/aarch64-linux-gnu/libtcmalloc.so" \
HEAPPROFILE=./heap_profile \
HEAP_PROFILE_ALLOCATION_INTERVAL=0 \
HEAP_PROFILE_INUSE_INTERVAL=4294967296 \
HEAPPROFILESIGNAL=5 \
MALLOCSTATS=1 \
./sbin/zabbix_server -f -c /etc/zabbix/zabbix_server.conf
```

3. 対象プロセスに signal 5 を送信して、プロファイルダンプを生成します。
1234 は実際のプロセス ID (PID) に置き換えてください:

```
kill -5 1234
```

4. 生成されたプロファイルを表示します:

```
pprof-symbolize -text ./sbin/zabbix_server ./heap_profile.0001.heap

Using local file ./sbin/zabbix_server.
Using local file ./heap_profile.0001.heap.
Total: 1078.1 MB
  1076.8  99.9%  99.9%   1076.8  99.9% zbx_malloc2
     1.0   0.1% 100.0%      1.0   0.1% __GI___strdup
     0.2   0.0% 100.0%      0.2   0.0% CRYPTO_zalloc@@OPENSSL_3.0.0
     0.1   0.0% 100.0%      0.1   0.0% OPENSSL_LH_insert@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% zbx_realloc2
     0.0   0.0% 100.0%      0.1   0.0% PKCS7_decrypt@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% find_best_tree_node
     0.0   0.0% 100.0%      0.0   0.0% CRYPTO_strndup@@OPENSSL_3.0.0
     ...
     0.0   0.0% 100.0%      0.0   0.0% preprocessing_flush_value
     0.0   0.0% 100.0%   1074.0  99.6% preprocessor_add_request
```

この例では、zbx_malloc2 がメモリ割り当てのほぼすべてを担当しています。

関連情報:

- 関連する障害報告については ZBX-25050 および ZBX-25584 を参照してください。
- コンパイルオプション (-std=gnu99, -g, -O0 など) については GCC Option Summary を参照してください。
- tcmalloc プロファイリング用の環境変数については Gperftools Heap Profiler のドキュメントを参照してください。

マルチプライマリモードでの MySQL 8.0 Group Replication

MySQL 8.0 Group Replication をマルチプライマリモードで使用している場合、トランザクションのコミット中に次のようなエラーが発生することがあります。

```
1531697:20250128:064734.697 query [txnlev:1] [update alerts set status=1,retries=0,error='' where alertid=
1531697:20250128:064734.713 query [txnlev:1] [commit;]
1531697:20250128:064734.753 [Z3005] query failed: [3101] Plugin instructed the server to rollback the curr
```

このエラーは、外部キー制約を伴うロールバック処理の問題によって発生しているようです。

関連情報:

- 関連する問題レポートについては、ZBX-26060 を参照してください。
- アップストリームの問題については、MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" を参照してください。

ログインと systemd

- zabbix * ユーザーをシステムユーザーとして、つまりログインできないユーザーとして作成することを推奨します。一部のユーザーはこの推奨事項を無視し、Zabbix を実行しているホストにログインする際（SSH など）にも同じアカウントを使用しています。これにより、ログアウト時に Zabbix デーモンがクラッシュする可能性があります。この場合、Zabbix サーバーのログに次のようなメッセージが表示されます。

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

また、Zabbix エージェントのログには次のように表示されます。

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

これは、/etc/systemd/logind.conf で設定されている systemd のデフォルト設定 RemoveIPC=yes が原因です。システムからログアウトすると、以前に Zabbix が作成したセマフォが削除されるため、クラッシュが発生します。

systemd のドキュメントからの引用:

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

この障害には 2 つの解決策があります。

1. (推奨) *zabbix* アカウントを Zabbix プロセス以外の目的で使用しないでください。その他の用途には専用のアカウントを作成します。
2. (非推奨) /etc/systemd/logind.conf で RemoveIPC=no を設定し、システムを再起動します。RemoveIPC はシステム全体に適用されるパラメータであるため、変更するとシステム全体に影響します。

要件

ハードウェア要件

メモリ

Zabbix には物理メモリとディスクメモリの両方が必要です。必要なディスクメモリの量は、監視するホスト数やパラメータ数によって異なります。監視パラメータの履歴を長期間保存する予定がある場合は、データベースに履歴を保存するために少なくとも数ギガバイトの空き容量を確保する必要があります。各 Zabbix デーモンプロセスは、データベースサーバーへの複数の接続を必要とします。接続に割り当てられるメモリの量は、データベースエンジンの設定によって異なります。

> **Note:**
> 物理メモリが多いほど、データベース（および Zabbix）の動作が高速になります。

CPU

Zabbix、特に Zabbix データベースは、監視するパラメータの数や選択したデータベースエンジンによっては、かなりの CPU リソースを必要とする場合があります。

その他のハードウェア

Zabbix で SMS 通知サポートを使用するには、シリアル通信ポートとシリアル GSM モデムが必要です。USB-シリアル変換器も動作します。

ハードウェア構成の例

この表は、**Linux/BSD/Unix** プラットフォームを前提としたハードウェア構成の例を示しています。

以下は、開始時点でのサイズおよびハードウェア構成の例です。Zabbix の各インストール環境はそれぞれ異なります。Zabbix のインストール環境を本番環境にデプロイする前に、ステージング環境または開発環境で Zabbix システムのパフォーマンスをベンチマークし、要件を十分に把握してください。

インストール規模	監視メトリクス数[1]	CPU/vCPU コア数	メモリ (GiB)	データベース	Amazon EC2[2]
小規模	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
中規模	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge

インストール規模	監視メトリクス数[1]	CPU/vCPU コア数	メモリ (GiB)	データベース	Amazon EC2[2]
大規模	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
非常に大規模	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] サイジングの目的では、監視メトリクス 1 つは、おおよそ「1 メトリクス = 1 アイテム + 1 トリガー + 1 グラフ」という式で表されます。これは推定の目安であり、監視メトリクス数を計算するための式ではありません。
 [2] Amazon の汎用 EC2 インスタンスを使用した例です。ARM64 または x86_64 アーキテクチャを使用しています。Zabbix を本番環境にインストールする前に、インストール環境の評価およびテストを行い、Compute/Memory/Storage 最適化など、適切なインスタンスタイプを選択してください。

Note:
実際の構成は、アクティブなアイテム数と更新間隔に大きく依存します（詳細については、このページのデータベースサイズセクションを参照してください）。大規模なインストール環境では、データベースを別のサーバーで実行することを強く推奨します。

サポートされているプラットフォーム

セキュリティ要件と監視サーバーのミッションクリティカルな性質により、UNIX は必要なパフォーマンス、フォールトトレランス、およびレジリエンスを一貫して提供できる唯一のオペレーティングシステムです。Zabbix は市場をリードするバージョンで動作します。

Zabbix コンポーネントは、以下のプラットフォームで利用可能でテストされています。

プラットフォーム	サーバー	エージェント	エージェント 2	コメント
Linux	x	x	x	
Windows	-	x	x	Zabbix エージェントは、Windows XP (64-bit)/Server 2003 以降のすべてのデスクトップおよびサーバーバージョンでサポートされています。 Zabbix エージェント 2 は、サポートされている Go バージョンでのみコンパイルされ、重大なセキュリティ脆弱性を防ぐため、Windows 10 (32-bit)/Server 2016 以降のすべてのデスクトップおよびサーバーバージョンでサポートされています。Go 1.21 以降、必要な最小 Windows バージョンが引き上げられ、Windows 10/Server 2016 が Zabbix エージェント 2 の最小バージョンとなります。
macOS	x	x	-	
IBM AIX	x	x	-	Zabbix エージェントは、AIX 6.1 TL07 / 7.1 TL01 未満のプラットフォームでは動作しません。
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
Zabbix サーバー/エージェントは、他の Unix 系オペレーティングシステムでも動作する場合があります。

Attention:
Zabbix は、暗号化を有効にしてコンパイルされた場合、コアダンプを無効にし、コアダンプの無効化を許可しないシステムでは起動しません。

必要なソフトウェア

Zabbix は、最新の Web サーバー、主要なデータベースエンジン、PHP スクリプト言語を中心に構築されています。

サードパーティ製の外部周辺ソフトウェア

必須と記載されている場合、該当するソフトウェアまたはライブラリは必ず必要です。オプションのソフトウェアまたはライブラリは、特定の機能をサポートする場合に必要です。

ソフトウェア	必須ステータス	サポート対象バージョン	コメント
MySQL/Percona	いずれか 1 つ	8.4.0-9.7.X	MySQL（または Percona）を Zabbix のバックエンドデータベースとして使用する場合に必要です。InnoDB エンジンが必要です。 サーバー/プロキシのビルドには、C API (libmysqlclient) ライブラリの使用を推奨します。 本番環境では、LTS リリースのみを使用することを強く推奨します。「rolling release」、「early access」などのラベルが付いたリリースは推奨しません。
MariaDB		10.11.00-12.3.X	InnoDB エンジンが必要です。 サーバー/プロキシのビルドには、MariaDB Connector/C ライブラリの使用を推奨します。 詳細は、MariaDB で発生するデッドロックの可能性およびMariaDB 10.5.1-10.5.9 での UI 要素へのアクセスも参照してください。 本番環境では、LTS リリースのみを使用することを強く推奨します。「rolling release」、「early access」などのラベルが付いたリリースは推奨しません。
PostgreSQL		15.0-18.X	PostgreSQL を Zabbix のバックエンドデータベースとして使用する場合に必要です。 インストール規模によっては、特定の操作でデータベースが使用するメモリ量を十分に確保し、クエリの実行に時間がかかりすぎないように、PostgreSQL の *work_mem* 設定プロパティ（デフォルト値は 4MB）を増やす必要があります。
PostgreSQL 用 *TimescaleDB*		2.20.X-2.29.X	TimescaleDB を PostgreSQL のデータベース拡張として使用する場合に必要です。圧縮をサポートする TimescaleDB Community Edition をインストールしてください。 PostgreSQL と TimescaleDB のバージョン互換性については、TimescaleDB のドキュメントを参照してください。
Elasticsearch		7.X	Elasticsearch を Zabbix のバックエンドデータベースとして使用する場合に必要です。 Elasticsearch は履歴データの保存にのみ使用でき、Zabbix サーバーでのみサポートされます。Elasticsearch のサポートは現在試験的です。サーバー/プロキシに必要なソフトウェアも参照してください。
ClickHouse		26.3.0.0-26.9.X.X	履歴データの保存（Zabbix サーバーでのみサポート）に使用する Zabbix のバックエンドデータベースとして ClickHouse を使用する場合、またはテレメトリデータソースとして使用する場合に必要です。 ClickHouse をテレメトリデータソースとして使用するには、ClickHouse 26.8.17.4 以降が必要です。また、Zabbix サーバーと ClickHouse サーバーのシステム時刻を同期する必要があります。同期しない場合、テレメトリクエリでデータが欠落したり、同じデータが 2 回返されたりする可能性があります。
SQLite	オプション	3.3.5-3.53.X	SQLite は Zabbix プロキシでのみサポートされます。SQLite を Zabbix プロキシのデータベースとして使用する場合に必要です。
smartmontools		7.1 以降	Zabbix エージェント 2 に必要です。
who			ユーザー数プラグインに必要です。
dpkg			system.sw.packages プラグインに必要です。
pkgtool			system.sw.packages プラグインに必要です。
rpm			system.sw.packages プラグインに必要です。
pacman			system.sw.packages プラグインに必要です。

ソフトウェア	必須ステータス	サポート対象バージョン	コメント
q applets			Gentoo Linux で system.sw.packages プラグインを使用するには、q applets の一部である `qlist` と `qsize` が必要です。

Note:
Zabbix はオペレーティングシステムで利用可能なデータベースでも動作しますが、最適な環境を実現するため、データベース開発元の公式リポジトリからインストールしたデータベースを使用することを推奨します。

Web インターフェース

Zabbix Web インターフェースでサポートされる画面の最小幅は 1200px です。

必須と記載されている場合、必要なソフトウェア/ライブラリは厳密に必要です。オプションのものは、特定の機能をサポートするために必要です。

ソフトウェア	必須ステータス	サポートされるバージョン	コメント
PHP	はい	8.2.0 - 8.5.X	
Apache	いずれか	2.4 以降	
Nginx		1.20 以降	
MySQL	いずれか	サードパーティの外部周辺ソフトウェアを参照	
PostgreSQL			
PHP 拡張機能			
mysqli	はい		MySQL を Zabbix バックエンドデータベースとして使用する場合に必要です。
pgsql			PostgreSQL を Zabbix バックエンドデータベースとして使用する場合に必要です。
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*)。ユーザースクリプトのサポートに必要です。
gd		2.0.28 以降	php-gd（ディストリビューターが個別のパッケージとして提供している場合）。PHP GD 拡張機能は、PNG 画像 (*--with-png-dir*)、JPEG 画像 (*--with-jpeg-dir*)、および FreeType 2 (*--with-freetype-dir*) をサポートしている必要があります。一部の Web インターフェース言語で、グラフ内のテキストの重なりを回避するために、バージョン 2.3.0 以降が必要になる場合があります。
libxml		2.6.15 以降	php-xml（ディストリビューターが個別のパッケージとして提供している場合）
xmlwriter			php-xmlwriter（ディストリビューターが個別のパッケージとして提供している場合）
xmlreader			php-xmlreader（ディストリビューターが個別のパッケージとして提供している場合）
ctype			php-ctype (*--enable-ctype*)
session			php-session（ディストリビューターが個別のパッケージとして提供している場合）
ldap	いいえ		php-ldap。LDAP 認証に必要です。
openssl			php-openssl。SAML 認証に必要です。
gettext			php-gettext (*--with-gettext*)。翻訳に必要です。
cURL		7.19.4 以降	php-curl。Duo Universal Prompt MFA、SMTP 認証、およびテレメトリデータソースのテストに必要です。

Zabbix に付属するサードパーティの Web インターフェースライブラリ:

ライブラリ	必須ステータス	付属バージョン	コメント
jQuery JavaScript Library	はい	3.6.0	クロスブラウザー開発のプロセスを簡素化する JavaScript ライブラリです。
jQuery UI		1.12.1	jQuery 上に構築された、ユーザーインターフェースの操作、エフェクト、ウィジェット、およびテーマのセットです。
SAML PHP Toolkit		4.3.1	Zabbix へのサインインで SAML 2.0 認証をサポートする PHP ツールキットです。
Duo Universal PHP library		1.1.2	Zabbix で Duo Universal Prompt の多要素認証をサポートする PHP ライブラリです。
Symfony Yaml Component		5.1.0	Zabbix の設定要素を YAML 形式でエクスポートおよびインポートするためのサポートを追加します。

Note:
Zabbix は、以前のバージョンの Apache、MySQL、および PostgreSQL でも動作する場合があります。

Attention:
デフォルトの DejaVu 以外のフォントを使用する場合、PHP 関数imagerotateが必要になることがあります。この関数がない場合、グラフの表示時にこれらのフォントが正しく描画されないことがあります。この関数は、PHP がバンドルされた GD を使用してコンパイルされている場合にのみ利用できます。Debian やその他のディストリビューションでは該当しません。

Zabbix Web インターフェースコードの作成およびデバッグに使用されるサードパーティライブラリ:

ライブラリ	必須ステータス	最小バージョン	説明
Composer	いいえ	2.4.1	PHP ソフトウェアと必要なライブラリの依存関係を管理するための標準形式を提供する、アプリケーションレベルの PHP パッケージマネージャーです。
PHPUnit		8.5.29	Zabbix Web インターフェースをテストするための PHP ユニットテストフレームワークです。
SASS		3.4.22	Cascading Style Sheets（CSS）に解釈およびコンパイルされるプリプロセッサスクリプト言語です。

クライアント側の Web ブラウザ

Cookie と JavaScript を有効にする必要があります。

Google Chrome、Mozilla Firefox、Microsoft Edge、Apple Safari、Opera の最新の安定バージョンがサポートされています。

Warning:
IFrame の同一生成元ポリシーが実装されているため、Zabbix を異なるドメインのフレームに配置することはできません。

 ただし、Zabbix フレームに配置されたページは、フレームに配置されたページと Zabbix フロントエンドが同じドメインにある場合、Zabbix フロントエンドに（JavaScript を介して）アクセスできます。`http://secure-zabbix.com/cms/page.html` のようなページを `http://secure-zabbix.com/zabbix/`のダッシュボードに配置すると、Zabbix への完全な JS アクセスが可能になります。

サーバー/プロキシ

必須と記載されている場合、必要なソフトウェア/ライブラリは厳密に必要です。
オプションのものは、特定の機能をサポートするために必要です。

要件	必須	説明
libpcre2	はい	Perl 互換正規表現（PCRE）をサポートするために、PCRE2 ライブラリが必要です。PCRE2 v10.x がサポートされています。
libevent		プロセス間通信に必要です。バージョン 2.0.10 以降。
libevent-pthreads		プロセス間通信に必要です。
libpthread		mutex および読み取り/書き込みロックのサポートに必要です（libc の一部である場合があります）。
libresolv		DNS 名前解決に必要です（libc の一部である場合があります）。

要件	必須	説明
libiconv		テキストのエンコーディング/形式変換に必要です（libc の一部である場合があります）。Linux 上の Zabbix サーバーでは必須です。
libz		圧縮のサポートに必要です。
libm		数学ライブラリ。Zabbix サーバーでのみ必要です。
libmysqlclient	いずれか	MySQL を使用する場合に必要です。
libmariadb		MariaDB を使用する場合に必要です。
libpq5		PostgreSQL を使用する場合に必要です。*libpq5* のバージョンは、使用する PostgreSQL データベースのバージョンと同じか、それより新しい必要があります。
libsqlite3		SQLite を使用する場合に必要です。Zabbix プロキシでのみ必要です。
libOpenIPMI	いいえ	IPMI のサポートに必要です。Zabbix サーバーでのみ必要です。
libssh2 または *libssh*		SSH チェックに必要です。バージョン 1.8.0 以降（libssh2）、0.9.0 以降（libssh）。
libcurl		以下の機能に必要です。 - Web 監視、VMware 監視、およびHTTP エージェントアイテム（すべてバージョン 7.19.1 以降）。 - Zabbix エージェントのweb.page.*アイテム（バージョン 7.19.1 以降。エージェントの要件も参照）。 - SMTP 認証（Basic：バージョン 7.20.0 以降、OAuth：バージョン 7.33 以降。Web インターフェースの要件も参照）。 - Elasticsearch（バージョン 7.28.0 以降）。 - Telemetry クエリアイテム（バージョン 7.28.0 以降）。 すべての機能でバージョン 7.28.0 以降を使用することを推奨します。 `web.page.*` アイテムでアップグレードされた cURL 機能を使用するには、Zabbix サーバー/プロキシを再起動してください。 SMTP 認証では、実行時に `libcurl-full` パッケージを使用してください。
libxml2		VMware 監視および XML XPath 前処理に必要です。
net-snmp		SNMP のサポートに必要です。バージョン 5.3.0 以降。 強力な暗号化プロトコル（AES192/AES192C、AES256/AES256C）のサポートは、net-snmp ライブラリ 5.8 以降で利用できます。RHEL 8 以降ベースのシステムでは、net-snmp 5.8.15 以降を使用することを推奨します。
libunixodbc		データベース監視に必要です。
libgnutls または *libopenssl*		暗号化を使用する場合に必要です。 最小バージョン：*libgnutls* - 3.1.18、*libopenssl* - 1.0.1
libldap		LDAP のサポートに必要です。
fping		ICMP ping アイテムに必要です。
c-ares		Zabbix が`--with-ares` オプションを指定して設定されている場合、非同期 DNS 名前解決に必要です。それ以外の場合は、*libevent* が使用されます。 最小バージョン：1.16.0。DNS クエリのキャッシュおよびフェイルオーバー機能（クエリキャッシュはデフォルトで有効）には、c-ares 1.26.0 以降が必要です。c-ares が OS パッケージによって提供されるシステムでは、インストールされたランタイムライブラリが、DNS キャッシュおよびフェイルオーバーに必要なリゾルバー機能を公開していることを確認してください。
g++		C++ コンパイラ。アプリケーションパフォーマンス監視（APM）のために、`--with-apm` オプションを指定して Zabbix プロキシをコンパイルする場合に必要です。
gRPC		`--with-apm` オプションを指定してコンパイルした Zabbix プロキシでAPMを使用する場合に必要です。 必要なライブラリ：*grpc++*、*grpc*、*gpr*。 Zabbix プロキシのコンパイルには、`grpc_cpp_plugin` ユーティリティも必要です。 必要なバージョンは、以下のオペレーティングシステムの標準パッケージリポジトリで利用できます。 - Ubuntu 23.04 以降。 - Debian 12（Bookworm）以降。 - Alpine Linux 3.15 以降。 - RHEL、Rocky Linux、CentOS Stream 9 以降（EPELリポジトリからのみ）。 - Fedora X 以降。 古いバージョンのオペレーティングシステムでは、*gRPC* をソースからビルドする必要があります。
protobuf		`--with-apm` オプションを指定してコンパイルした Zabbix プロキシでAPMを使用する場合に必要です。 Zabbix プロキシのコンパイルには、`protoc` コンパイラも必要です。

要件	必須	説明
Abseil		`--with-apm` オプションを指定してコンパイルした Zabbix プロキシでAPMを使用する場合に必要です。 必要なライブラリ：*absl_synchronization*、*absl_base*。 通常、*gRPC* と一緒にインストールされます。

エージェント

Requirement	Mandatory status	Description
libpcre2	Yes	Perl 互換正規表現 (Perl Compatible Regular Expression) (PCRE) のサポートに必要な PCRE2 ライブラリです。 PCRE2 v10.x がサポートされています。 ログ監視に必要です。Windows でも必要です。
libpthread		mutex および read-write lock のサポートに必要です (libc の一部である場合があります)。Windows では不要です。
libresolv		DNS 解決に必要です (libc の一部である場合があります)。Windows では不要です。
libiconv		ログアイテム、ファイルコンテンツ、ファイル regex、および regmatch アイテムで UTF-8 へのテキストエンコーディング/形式変換に必要です (libc の一部である場合があります)。Windows では不要です。
libgnutls or *libopenssl*	No	暗号化 を使用する場合に必要です。 最小バージョン: *libgnutls* - 3.1.18、*libopenssl* - 1.0.1 Microsoft Windows では OpenSSL 1.1.1 以降が必要です。
libldap		LDAP を使用する場合に必要です。Windows ではサポートされていません。
libcurl		Zabbix エージェントのweb.page.* アイテムの拡張サポートに必要です。 libcurl がない場合でも、基本機能は利用できます (例: web.page.get[http://example.com])。libcurl がある場合、エージェントは認証情報付きの HTTP URL (例: http://user:password@example.com) や HTTPS URL などの追加機能をサポートします。 バージョン 7.19.1 以降が必要です (7.28.0 以降を推奨します)。 更新された cURL 機能を使用するには、Zabbix エージェントを再起動してください。
libmodbus		Modbus 監視を使用する場合のみ必要です。 バージョン 3.0 以降。

エージェント 2

Requirement	Mandatory status	Description
Go	Yes	Zabbix エージェント 2 とそのプラグインをソースからビルドするために必要です。 Go 1.25 以降がサポートされています。インストール手順については go.dev を参照してください。 Zabbix エージェント 2 とそのプラグインで使用される Go ライブラリは Zabbix の Git リポジトリに一覧されています（リポジトリ内で `indirect` とマークされているライブラリは、他の必須ライブラリの依存関係です）。 - Zabbix エージェント 2 - プラグインサポート - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Yes	PCRE2 ライブラリは Perl Compatible Regular Expression (PCRE) のサポートに必要です。 PCRE2 v10.x がサポートされています。 ログ監視に必要です。Windows でも必要です。

Requirement	Mandatory status	Description
libopenssl	No	暗号化を使用する場合に必要です。 UNIX プラットフォームでは OpenSSL 1.0.1 以降が必要です。 OpenSSL ライブラリでは PSK サポートを有効にする必要があります。LibreSSL はサポートされていません。 Microsoft Windows システムでは OpenSSL 1.1.1 以降が必要です。

Web サービス

Zabbix Web サービスを使用して定期レポートを生成するのに、Google Chrome の最新の安定バージョンがサポートされています。

Web サービスの構築に必要な Go のバージョンは、Zabbix エージェント 2で使用されているバージョンと同じです。

Java ゲートウェイ

ソースリポジトリまたはアーカイブから Zabbix を入手した場合、必要な依存関係はすでにソースツリーに含まれています。

ディストリビューションのパッケージから Zabbix を入手した場合、必要な依存関係はすでにパッケージングシステムによって提供されています。

上記のどちらの場合も、ソフトウェアは使用できる状態になっており、追加のダウンロードは必要ありません。

ただし、これらの依存関係について独自のバージョンを提供したい場合（たとえば、Linux ディストリビューション用のパッケージを準備している場合）は、以下に Java ゲートウェイで動作することが確認されているライブラリバージョンの一覧を示します。Zabbix は、これらのライブラリの他のバージョンでも動作する可能性があります。

次の表は、現在、元のコードで Java ゲートウェイにバンドルされている JAR ファイルの一覧です。

ライブラリ	必須	バンドルされているバージョン	コメント
android-json	はい	4.3r1	JSON（JavaScript Object Notation）は、軽量なデータ交換形式です。これは、Android SDK から抽出された、`org.json` 互換の Android 実装です。
logback-classic		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

Java ゲートウェイは、Oracle Java またはオープンソースの OpenJDK（バージョン 8 以降）を使用してビルドできます。Zabbix が提供するパッケージは、OpenJDK を使用してコンパイルされています。次の表は、各ディストリビューションで Zabbix パッケージのビルドに使用される OpenJDK パッケージの一覧です。

ディストリビューション	OpenJDK パッケージ
AlmaLinux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)

ディストリビューション	OpenJDK パッケージ
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

ブリッジアダプター

Zabbix ブリッジアダプターに特別な要件はありません。

デフォルトポート

以下のリストには、デフォルト設定で Zabbix コンポーネントが使用する可能性のあるポートが含まれています。

Zabbix コンポーネント	プロトコルとポート	接続を開始するコンポーネント
Zabbix エージェント	TCP 10050	Zabbix サーバーまたは Zabbix プロキシ（パッシブチェックのみ）
Zabbix エージェント 2	TCP 10050	Zabbix サーバーまたは Zabbix プロキシ（パッシブチェックのみ）
Zabbix サーバー	TCP 10051	アクティブな Zabbix エージェント、アクティブな Zabbix プロキシ、Zabbix sender、Zabbix Web インターフェース
Zabbix プロキシ	TCP 10051	アクティブな Zabbix エージェント、Zabbix sender、Zabbix サーバー（パッシブプロキシのみ）
Zabbix Java ゲートウェイ	TCP 10052	Zabbix サーバーまたはプロキシ
Zabbix Web サービス	TCP 10053	Zabbix サーバー
Zabbix MCP サーバー	TCP 8443	MCP クライアント（AI エージェント）
Zabbix Web インターフェース	TCP 80 TCP 443	Web ブラウザー、Zabbix Web サービス（スケジュールレポート）、API クライアント
SNMP トラップ（snmptrapd）	UDP 162	SNMP デバイス
Zabbix ブリッジアダプター	TCP 10054	Zabbix サーバー

Warning:
実際に実行するコンポーネントに必要なポートのみを開き、各ファイアウォールルールを接続が必要な特定のホストに限定してください（たとえば、ポート 10050 は任意のアドレスからではなく、Zabbix サーバーまたはプロキシからのみ許可します）。Zabbix のポートをインターネットに直接公開しないでください。
 ポートを開く前に、インストール環境のセキュリティを強化してください。コンポーネント間の暗号化を有効にし、コンポーネントの設定ファイルで許可するホストを制限し、セキュリティのベストプラクティスを確認してください。

データベースサイズ

Zabbix の設定データは、一定量のディスク容量を必要とし、あまり増加しません。

Zabbix のデータベースサイズは主に次の変数に依存し、これらが保存される履歴データ量を決定します。

- 1 秒あたりの処理済み値の数

これは、Zabbix サーバーが毎秒受信する新しい値の平均数です。
たとえば、監視対象のアイテムが 3000 個あり、更新間隔が 60 秒の場合、1 秒あたりの値の数は 3000/60 = **50** と計算されます。

つまり、毎秒 50 個の新しい値が Zabbix データベースに追加されます。

- 履歴の Housekeeper 設定

Zabbix は値を一定期間保持します。通常は数週間または数か月です。
新しい値ごとに、データとインデックスのために一定量のディスク容量が必要になります。

そのため、30 日分の履歴を保持し、1 秒あたり 50 個の値を受信する場合、値の総数はおよそ (**30***24*3600)* **50** = 129.600.000、つまり約 130M 個の値になります。

使用するデータベースエンジンや受信する値の種類（浮動小数点、整数、文字列、ログファイルなど）によって、1 つの値を保持するために必要なディスク容量は 40 バイトから数百バイトまで変動します。
通常、数値アイテムの場合は 1 値あたり約 90 バイトです[2]。
この例では、130M 個の値には 130M * 90 バイト = **10.9GB** のディスク容量が必要になります。

Note:
テキスト/ログアイテムの値のサイズを正確に予測することはできませんが、1 値あたり約 500 バイトと見積もることができます。

- トレンドの Housekeeper 設定

Zabbix は、各アイテムについて 1 時間分の max/min/avg/count の値を **trends** テーブルに保持します。
このデータはトレンド表示や長期間のグラフに使用されます。
1 時間の期間はカスタマイズできません。

Zabbix データベースは、データベースの種類に応じて、各 total あたり約 90 バイトを必要とします。
トレンドデータを 5 年間保持したいとします。
3000 個のアイテムの値には、3000*24*365* **90** = **2.2GB** / 年、または 5 年で **11GB** が必要になります。

- イベントの Housekeeper 設定

各 Zabbix イベントには、約 250 バイトのディスク容量が必要です[1]。
Zabbix が 1 日に生成するイベント数を見積もるのは困難です。
最悪のケースでは、Zabbix が 1 秒あたり 1 件のイベントを生成すると仮定できます。

回復した各イベントについて、event_recovery レコードが作成されます。
通常、ほとんどのイベントは回復するため、イベントごとに 1 件の event_recovery レコードが作成されると考えられます。
つまり、イベントごとに追加で 80 バイトが必要です。

必要に応じて、イベントにはタグを付けることができ、各タグレコードには約 100 バイトのディスク容量が必要です[1]。
イベントごとのタグ数（#tags）は設定によって異なります。
そのため、各イベントには追加で #tags * 100 バイトのディスク容量が必要になります。

つまり、3 年分のイベントを保持したい場合、3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B のディスク容量が必要になります[2]。

Note:
[1] イベント名、タグ、値に非 ASCII 文字が含まれる場合は、さらに多くなります。
 [2] この概算値は MySQL に基づいており、他のデータベースでは異なる場合があります。

次の表には、Zabbix システムに必要なディスク容量を計算するために使用できる式が示されています。

Parameter	必要なディスク容量の式（バイト）
Zabbix configuration	固定サイズ。通常は 10MB 以下。
History	days*(items/refresh rate)*24*3600*bytes items：アイテム数 days：履歴を保持する日数 refresh rate：アイテムの平均更新間隔 bytes：1 つの値を保持するために必要なバイト数。データベースエンジンに依存し、通常は約 90 バイト。
Trends	days*(items/3600)*24*3600*bytes items：アイテム数 days：履歴を保持する日数 bytes：1 つのトレンドを保持するために必要なバイト数。データベースエンジンに依存し、通常は約 90 バイト。
Events	days*events*24*3600*bytes events：1 秒あたりのイベント数。最悪ケースでは 1（1）秒あたり 1 件のイベント。 days：履歴を保持する日数 bytes：1 つのイベントを保持するために必要なバイト数。データベースエンジンに依存し、通常は約 330 + イベントごとの平均タグ数 * 100 バイト。

したがって、必要なディスク容量の合計は次のように計算できます。

Configuration + History + Trends + Events

ディスク容量は、Zabbix のインストール直後にすぐ使用されるわけではありません。
データベースサイズはその後増加し、Housekeeper 設定に依存するある時点で増加が止まります。

時刻同期

Zabbix が稼働しているサーバーで正確なシステム時刻を持つことは非常に重要です。ntpdは、ホストの時刻を他のマシンの時刻と同期させる最も一般的なデーモンです。Zabbix コンポーネントが稼働しているすべてのシステムで、システム時刻を同期させておくことを強く推奨します。

新機能とアップグレードに関する注意事項: 8.0

Zabbix 8.0.0 は Zabbix 7.4.0 の後継として、新機能と機能強化を追加しています。

このページには、次の内容も記載されています。

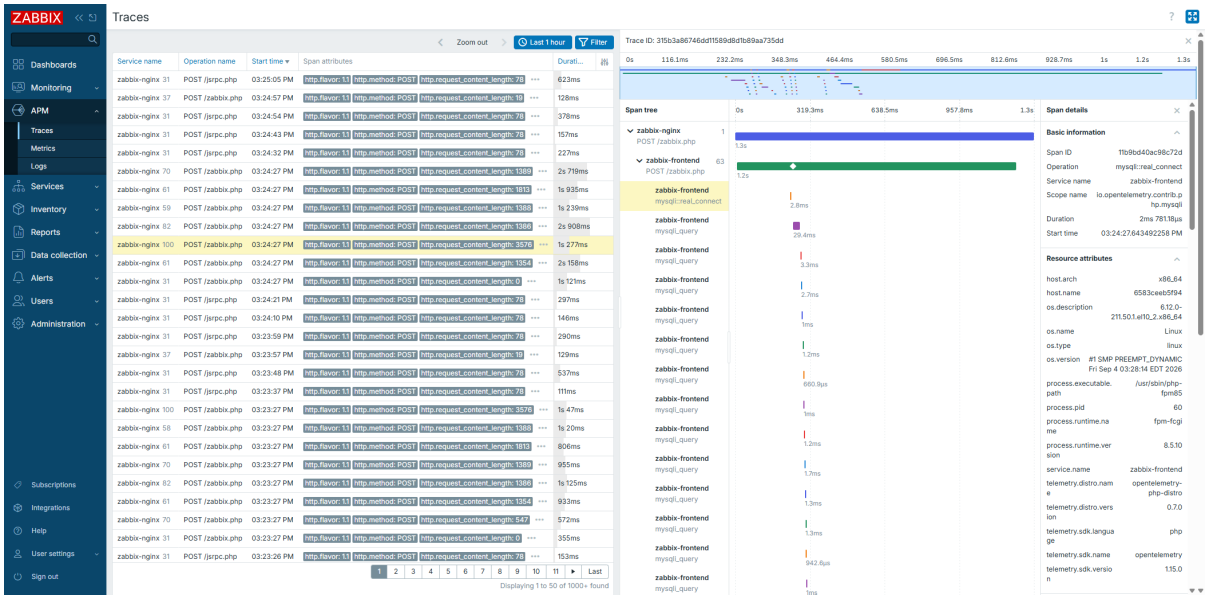
- テンプレートの新機能: 新しいテンプレートや既存のテンプレートへの変更など。
- アップグレードに関する注意事項: 互換性を損なう変更など。

関連項目: API の変更。

アプリケーションパフォーマンス監視　Zabbix は、アプリケーションやサービスからテレメトリデータを収集するためのオープン標準であるOpenTelemetryに基づくアプリケーションパフォーマンス監視（APM）をサポートするようになりました。このデータにより、アプリケーションやサービスを通過するリクエストの流れを確認し、パフォーマンスを測定して、処理が遅い、または失敗する原因を特定できます。

APM では、次のことができます。

- Zabbix プロキシでテレメトリデータを収集できます。Zabbix プロキシは、OpenTelemetry プロトコル（OTLP/gRPC）を介してアプリケーションまたは OpenTelemetry コレクターからデータを受信し、データソースに書き込めるようになりました。プロキシの設定に新しく追加された *APM* タブを参照してください。
- 新しい管理 > データソースセクションでテレメトリデータソース（ClickHouse）を指定し、新しい*APM*メニューでデータを表示できます。
- 新しいテレメトリクエリアイテムタイプを使用して、トリガーやグラフでテレメトリデータを利用できます。



次のパラメータとアイテムも追加されました。

- Zabbix サーバーのパラメータ: TelemetryProviderおよびStartTelemetryQueryPollers。
- Zabbix プロキシのパラメータ: TelemetryProvider、StartTelemetryQueryPollers、StartAPMCollectors、APMListenIP、APMListenPort、APMTLSCAFile、APMTLSCertFile、およびAPMTLSKeyFile。
- Zabbix Web インターフェースのパラメータ: $TELEMETRY_PROVIDERS[]。
- Zabbix プロキシのコンパイルオプション: --with-apm。公式の Zabbix プロキシパッケージは、このオプションを使用してビルドされています。
- テレメトリデータの収集を監視する内部アイテム: zabbix[apm,<type>]。

ClickHouse のサポート　アイテムの値の履歴を保存する Zabbix のバックエンドデータベースとして、またアプリケーションパフォーマンス監視のテレメトリデータソースとして、ClickHouse を使用できるようになりました。

履歴保存の設定方法と要件については、ClickHouse のセットアップを参照してください。既存の Zabbix データベース（MySQL または PostgreSQL）から ClickHouse に履歴データを移行する場合は、ClickHouse のスキーマおよび履歴の移行スクリプトを参照してください。

この機能により、Zabbix の設定ファイルで履歴保存プロバイダーを設定する方法も変更されます。この変更は、ClickHouse と Elasticsearch の両方のセットアップに影響します。詳細については、アップグレードに関する注意事項を参照してください。

複合イベント処理　複合イベント処理（CEP）は、Zabbix イベントに対してルールベースの処理を実行します。

たとえば、次のような処理を行えます。

- 関連するイベントをグループ化し、一連のイベントの最初のイベントを原因としてマークする（後続のイベントは症状として扱う）
- イベントの有効期間を指定する
- 特定のイベントの最大数を指定する
- 場所ごとに類似したイベントの通知を受け取る
- 一定期間内に特定のイベントが頻繁に発生した場合（またはまったく発生しなかった場合）に通知を受け取る

複合イベント処理では、イベント分析のためのスライディング時間ウィンドウが導入されます。
たとえば、10 分間のウィンドウでは、そのウィンドウ内に入るすべてのイベントが 1 つのコンテキストを形成します。
ウィンドウが進むと、古いイベントはウィンドウから退出（「追い出し」）し、新しいイベントが入る場合があります。

時間ウィンドウの概念は、操作を実行するための論理的なポイントを提供します。
たとえば、イベントがウィンドウに入ったときに、イベントを変換または「エンリッチ」できます。重要度を上げ下げしたり、タグを追加または調整したり、抑制を有効にしたりできます。

イベントがウィンドウから退出したときや、パターンが見つかったときにも、同様の操作を実行できます。

イベントストリームの分析は、CEP ルールによって定義されます。
CEP ルールは通常、時間ウィンドウ、条件、操作で構成されます。
操作は実行ポイントに関連付けられます。

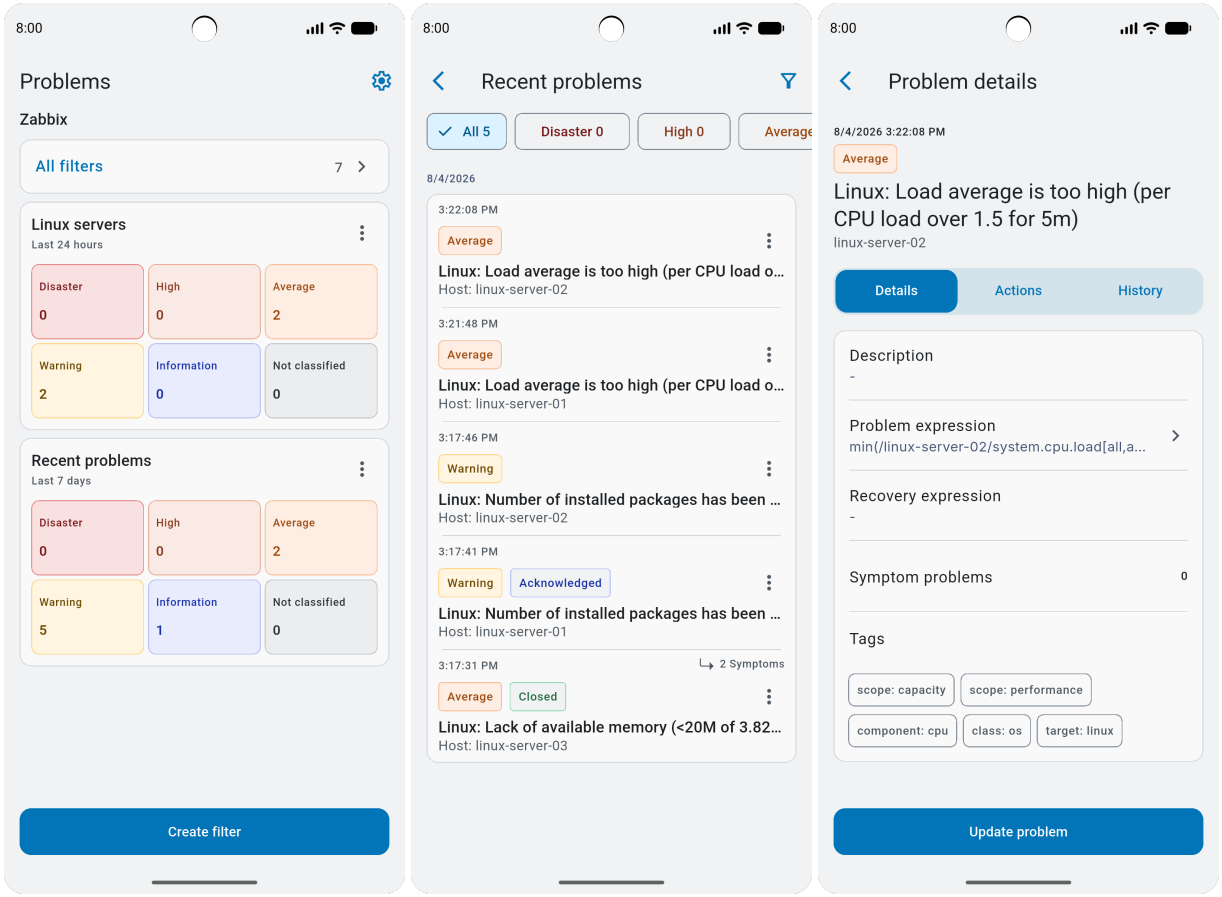
詳細なドキュメントは、安定版 8.0 がリリースされた時点で提供される予定です。

モバイルアプリ　Zabbix モバイルアプリ（近日公開）を使用すると、モバイルデバイスから Zabbix 環境を監視できます。障害の表示、詳細の確認、障害の更新、対応が必要な場合のプッシュ通知の受信が可能です。

このアプリでは、Zabbix 8.0 で次の機能が新たに追加されました。

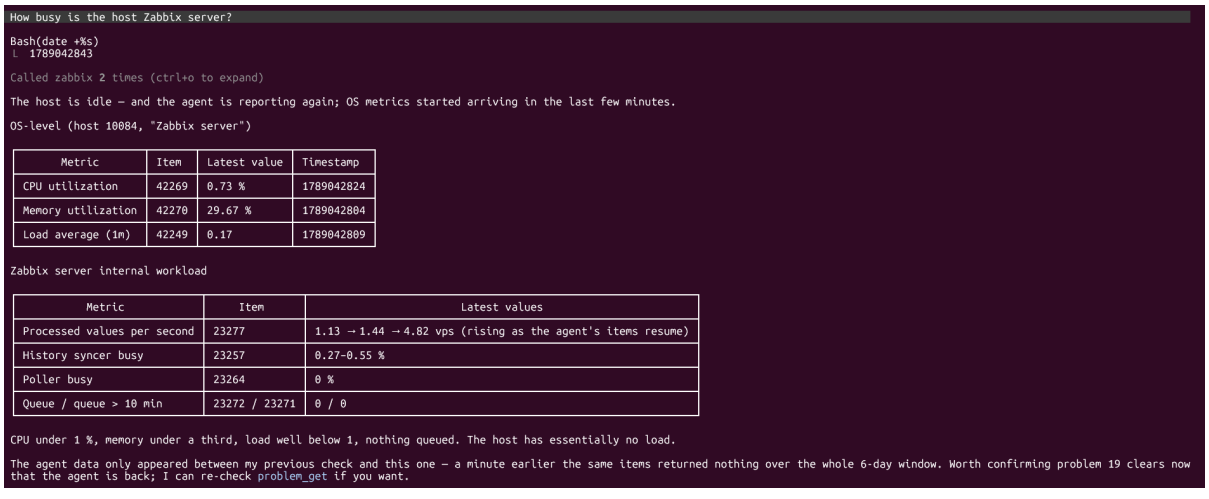
- モバイルデバイスを管理するための Web インターフェースのセクション: ユーザー > デバイスおよびユーザー設定 > デバイス。
- Zabbix サーバーからモバイルアプリに障害通知を送信するためのメディアタイプ: プッシュ通知。
- Zabbix サーバーとモバイルアプリ間の通信を可能にするコンポーネント: ブリッジアダプター。
- Zabbix サーバーの設定パラメータ: BridgeAdapterURL、EnableMobileDevices。

モバイルアプリで使用するための Zabbix の設定方法については、モバイルアプリセクションを参照してください。



MCP サーバー　新しいZabbix MCP サーバーは、Model Context Protocol（MCP）を介して AI エージェントを Zabbix に接続します。これにより、監視について AI アシスタントに自然言語で質問できるようになりました。たとえば、次のような質問ができます。

- データベースサーバーで、まだ認識されていない障害はどれですか？
- *web-01* では、今朝から障害が報告されています。原因は何ですか？
- *Linux servers* グループのすべてのホストを、今夜 *22:00* から *02:00* までメンテナンスに設定してください。



エージェントは Zabbix API トークンを使用して接続するため、トークン所有者に許可された範囲に限ってアクセスできます。また、

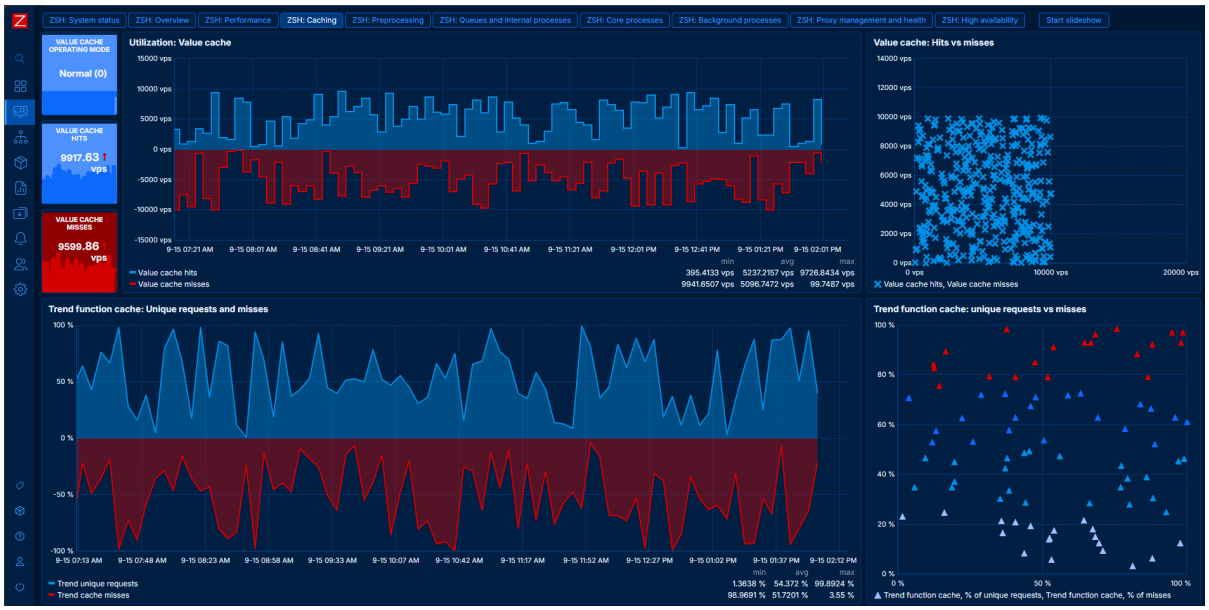
すべての変更は、そのユーザーの監査ログに記録されます。利用可能な操作の一覧については、ツールとリソースを参照してください。

トリガーのオンデマンドメンテナンス　メンテナンス期間を、トリガーまたはイベント名に対して定義できるようになりました。

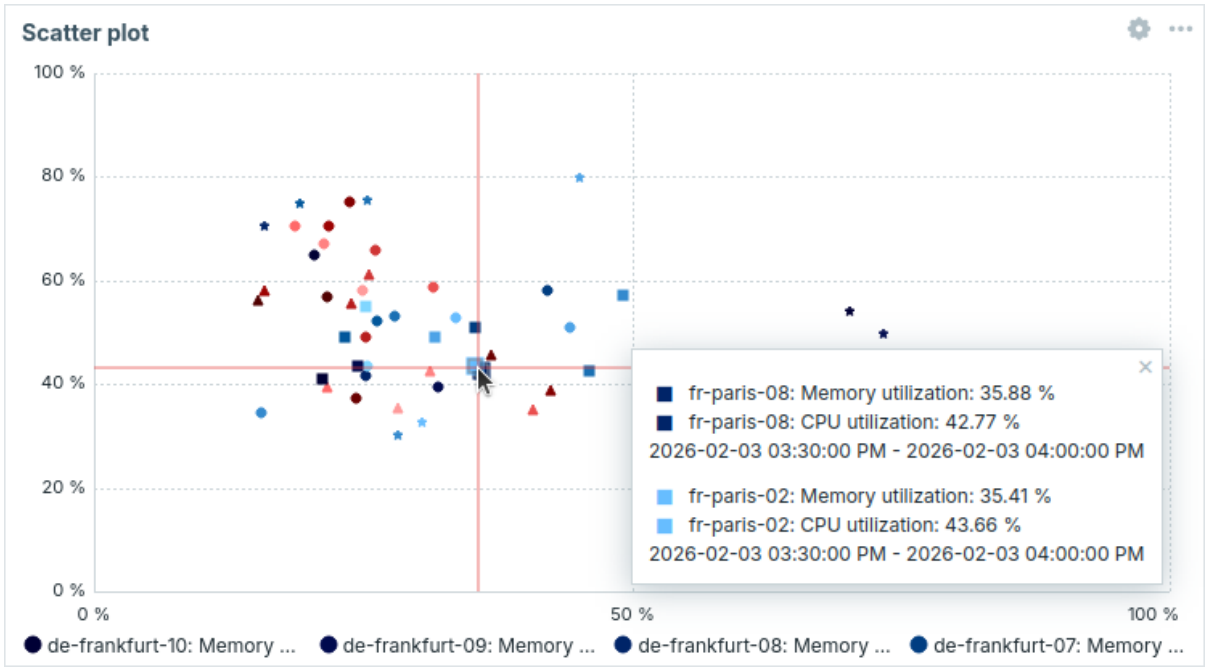
イベントメニューから、オンデマンドでメンテナンス期間を作成することもできます。オンデマンドメンテナンスは、ホスト、トリガー、イベント名、またはタグに対して開始できます。

トリガーのメンテナンスを作成するもう 1 つの方法は、障害の更新フォームから作成することです。

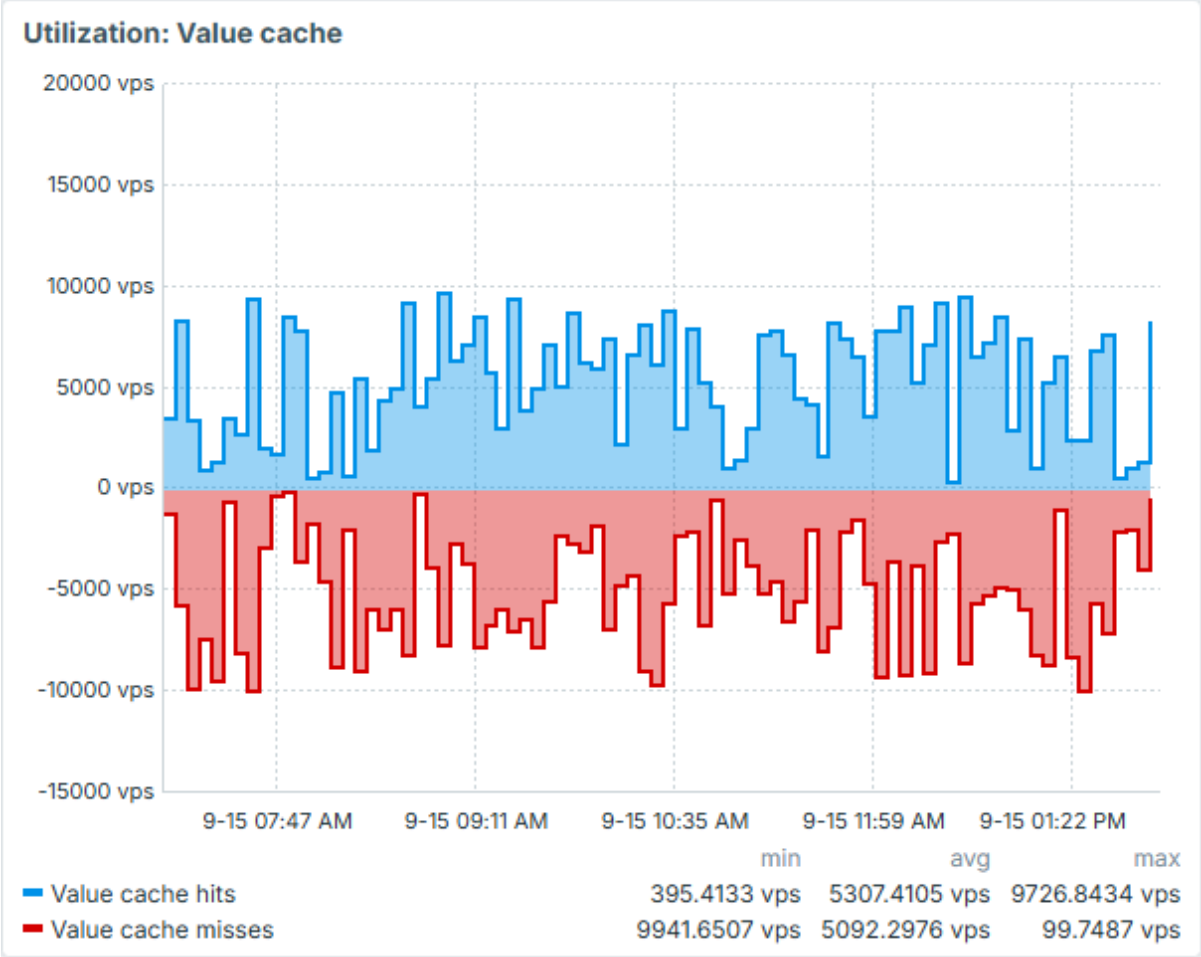
ダークブルーテーマ　Zabbix の Web インターフェースで、新しいダークブルーテーマを利用できるようになりました。ユーザープロファイルおよびデフォルトテーマとして設定できます。



散布図ウィジェット　散布図ウィジェットがダッシュボードウィジェットに追加されました。
これは、X 軸と Y 軸に沿って個々のデータポイントをプロットすることで、2 つのメトリクス間の関係を表示します。
これにより、データセット内のパターン、クラスター、相関関係、外れ値を明らかにするのに役立ちます。



グラフウィジェットの **Y** 軸反転　グラフウィジェットのグラフでは、Y 軸の値を反転して表示できるようになりました。
新しい 値を反転設定を使用すると、元のデータを変更せずに Y 軸の値に −1 を掛けることができます。



関連項目: グラフウィジェットの機能強化。

JSON データ型　Zabbix は、アイテム値のデータ型として JSON をサポートするようになりました。

以前は、JSON 値はテキストアイテムで収集され、64KB の制限付きで文字列として保存されていました。
現在、Zabbix は JSON 値をネイティブに 128MiB の制限付きで保存でき、無効な JSON 値（例: クォートされていないキー、末尾のカンマ、括弧の不一致）も拒否できます。

JSON データ型は、*Calculated* を除くすべてのアイテムおよびアイテムプロトタイプのタイプでサポートされており、リアルタイムデータエクスポートおよびconnectorsで利用できます。
JSON 値は、すべてのサポートされるデータベースおよびElasticsearchに保存できます。
TimescaleDB を使用している場合は、アップグレードノートを参照してください。

JSON データ型のアイテムはトリガーでは使用できない点に注意してください。ただし、JSON 以外のデータ型を持つ依存アイテムを使用して JSON フィールドを抽出し、それをトリガーで使用できます。

JSON 文字列（`net.if.discovery`、`vfs.file.get` など）を返すアイテムは、引き続きテキストアイテムです。ただし、必要に応じて JSON に変更できます。

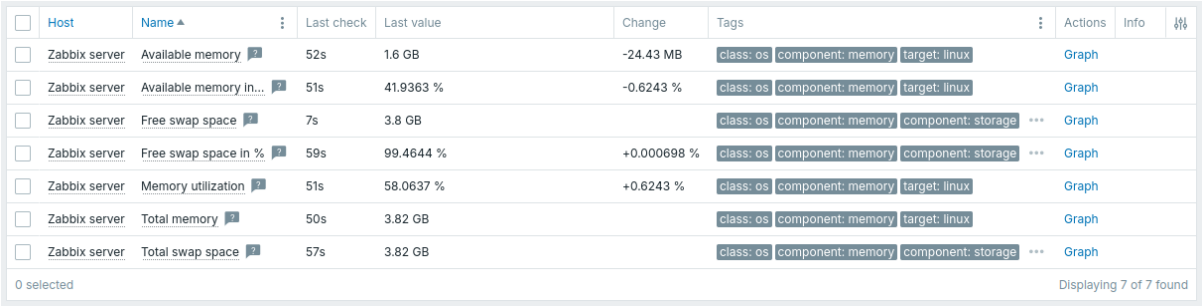
詳細および JSON データの制限については、アイテム設定を参照してください。

テーブルのカスタムレイアウト　いくつかの Web インターフェースのテーブルでは、列の表示/非表示、並べ替え、幅の変更ができるようになりました。また、カスタムテキスト、マクロ値、またはその両方を含む カスタムテキスト列を追加することも可能です。

カスタムレイアウトは次の場所で利用できます。

- *Monitoring* > ホスト、最新データ、および 障害
- *Data collection* > テンプレート および ホスト

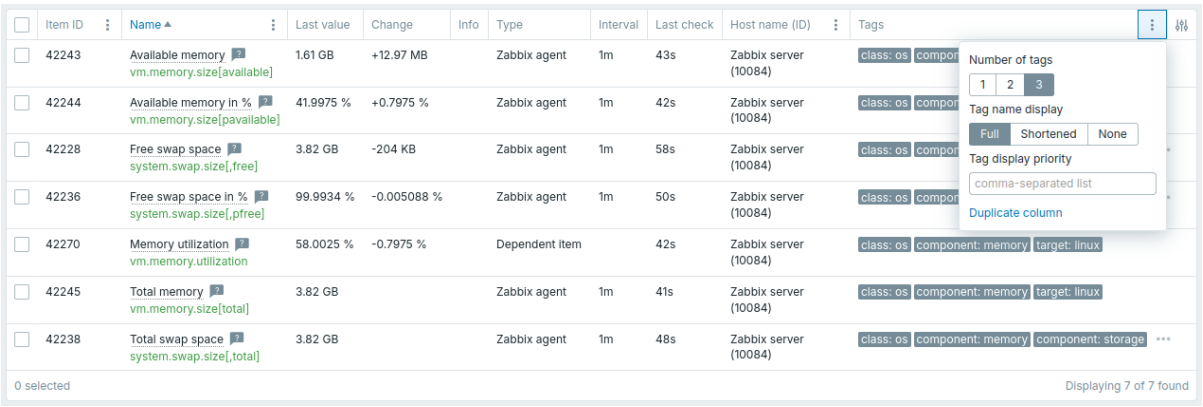
たとえば、*Monitoring* > 最新データのデフォルトのテーブルレイアウトは次のとおりです。



Host	Name	Last check	Last value	Change	Tags	Actions	Info
Zabbix server	Available memory	52s	1.6 GB	-24.43 MB	class: os, component: memory, target: linux	Graph	
Zabbix server	Available memory in...	51s	41.9363 %	-0.6243 %	class: os, component: memory, target: linux	Graph	
Zabbix server	Free swap space	7s	3.8 GB		class: os, component: memory, component: storage	Graph	
Zabbix server	Free swap space in %	59s	99.4644 %	+0.000698 %	class: os, component: memory, component: storage	Graph	
Zabbix server	Memory utilization	51s	58.0637 %	+0.6243 %	class: os, component: memory, target: linux	Graph	
Zabbix server	Total memory	50s	3.82 GB		class: os, component: memory, target: linux	Graph	
Zabbix server	Total swap space	57s	3.82 GB		class: os, component: memory, component: storage	Graph	

0 selected — Displaying 7 of 7 found

カスタムレイアウトの例:



Item ID	Name	Last value	Change	Info	Type	Interval	Last check	Host name (ID)	Tags
42243	Available memory vm.memory.size[available]	1.61 GB	+12.97 MB		Zabbix agent	1m	43s	Zabbix server (10084)	class: os, compon...
42244	Available memory in % vm.memory.size[pavailable]	41.9975 %	+0.7975 %		Zabbix agent	1m	42s	Zabbix server (10084)	class: os, compon...
42228	Free swap space system.swap.size[,free]	3.82 GB	-204 KB		Zabbix agent	1m	58s	Zabbix server (10084)	class: os, compon...
42236	Free swap space in % system.swap.size[,pfree]	99.9934 %	-0.005088 %		Zabbix agent	1m	50s	Zabbix server (10084)	class: os, compon...
42270	Memory utilization vm.memory.utilization	58.0025 %	-0.7975 %		Dependent item		42s	Zabbix server (10084)	class: os, component: memory, target: linux
42245	Total memory vm.memory.size[total]	3.82 GB			Zabbix agent	1m	41s	Zabbix server (10084)	class: os, component: memory, target: linux
42238	Total swap space system.swap.size[,total]	3.82 GB			Zabbix agent	1m	48s	Zabbix server (10084)	class: os, component: memory, component: storage

0 selected — Displaying 7 of 7 found

一部の列には、データの表示方法をさらに制御できる *Options* ボタンもあります。そのため、一部のフィルターパラメータはフィルタータブから列オプションへ移動しました。お気に入りフィルターのタブ を使用している場合は、アップグレードノート を参照してください。

グローバル正規表現　グローバル正規表現に、いくつかの使いやすさに関する改善が追加されました。

- グローバル正規表現の設定フォームがモーダルウィンドウで開くようになりました。
- 表現の種類の名称が変更されました。
- フォームに説明フィールドが追加されました。
- テストが別のタブで表示されなくなりました。
- グローバル正規表現の一覧を、名前と説明でフィルターできるようになりました。
- グローバル正規表現をエクスポートおよびインポートできるようになりました。

テンプレート

新しいテンプレートが追加され、多くのテンプレートが更新されました。

HTTPS エンドポイントに接続するテンプレートでは、デフォルトでエンドポイントの TLS 証明書を検証するようになりました。これは、*Claude API by HTTP*、*Cloudflare by HTTP*、*Kubernetes Cluster by HTTP* など、すべてのクラウドおよび API 監視テンプレートに影響します。HTTP ベースのテンプレートをアップグレードする前に、アップグレードに関する注意事項を確認してください。

Zabbix をアップグレードしても、カスタム変更が上書きされないように、既存のテンプレートは自動的には更新されません。テンプレートをアップグレードするか、新しいテンプレートを追加するには、テンプレートのアップグレードを参照してください。

> **Note:**
> Zabbix 6.0 以降、すべてのテンプレートは更新された形式に従っているため、6.0 より前のテンプレートのインポートに影響する可能性があります。詳細については、6.0 でのテンプレートの変更を参照してください。

新しいテンプレート　**AI** 使用状況の監視

- Claude API by HTTP: Claude API の使用状況とコストを監視するテンプレート。
- OpenAI Platform by HTTP: OpenAI の開発者向けプラットフォームを監視するテンプレート。

クラウド

- AWS by HTTPテンプレートセットに、*AWS Backup Vault by HTTP* テンプレートが追加されました。
- Azure by HTTPテンプレートセットに、*Azure Sentinel by HTTP* および *Azure Container Apps by HTTP* テンプレートが追加されました。
- GCP by HTTPテンプレートセットに、*GCP Cloud Run Service by HTTP*、*GCP Cloud Storage Bucket by HTTP*、*GCP Application Load Balancer by HTTP*、および *GCP Cost Monitoring by HTTP* テンプレートが追加されました。

- Oracle Cloud by HTTPテンプレートセットに、*Oracle Cloud Load Balancer by HTTP* および *Oracle Cloud Costs by HTTP* テンプレートが追加されました。

仮想化とコンテナ

- Kubernetes Cluster by HTTP: Kubernetes クラスターを監視するテンプレート。ノードは自動的に検出され、専用のホストを通じて監視されます。このテンプレートは、以前の Kubernetes テンプレートに置き換わるものです。
- Microsoft Hyper-V Failover Cluster by SSHおよびMicrosoft Hyper-V Standalone by SSH: Microsoft Hyper-V クラスターおよびスタンドアロンホストを SSH 経由で監視するテンプレート。
- Podman のZabbix エージェントによる監視、Zabbix エージェント (アクティブ) による監視、およびHTTP による監視: Podman API を通じて Podman コンテナとサービスの状態を監視するテンプレート。Podman API は認証方式をサポートしていないことに注意してください。

データベース

- MariaDB by ODBC: ODBC 経由で MariaDB データベースを監視するテンプレート。
- MariaDB by Zabbix agentおよびMariaDB by Zabbix agent active: Zabbix エージェントおよび Zabbix エージェント（アクティブ）経由で MariaDB データベースを監視するテンプレート。
- Percona by ODBC: ODBC 経由で Percona データベースを監視するテンプレート。
- Percona by Zabbix agentおよびPercona by Zabbix agent active: Zabbix エージェントおよび Zabbix エージェント（アクティブ）経由で Percona データベースを監視するテンプレート。

ネットワーク、セキュリティ、通信

- Aruba CX 8300s by SNMP: Aruba CX 8300 スイッチシリーズを SNMP ベースで監視するテンプレート。
- Ciena 3906 by SNMP: ファイルシステムや CPU 負荷を含む、Ciena 3906 デバイスを監視するテンプレート。
- Cisco Secure Firewall Threat Defense by HTTP: REST API を使用して Cisco Secure Firewall Threat Defense デバイスを監視する機能を提供するテンプレート。
- Cradlepoint NCM v2 by HTTP: Cradlepoint NCM v2 およびそのデバイスを HTTP 経由で監視するテンプレートセット。
- Huawei AR600 by SNMP: CoS および NQA メトリクスを含む、Huawei AR600 ルーターを監視するテンプレート。
- Ribbon SBC Edge by HTTP: Ribbon SBC Edge（旧称 SWe Lite）デバイスを HTTP 経由で監視するテンプレート。
- Ribbon SBC SWe Core by HTTP: SBC SWe Core デバイスを HTTP 経由で監視するテンプレート、およびRibbon SBC SWe CE by HTTP: Ribbon SBC SWe Call Engine（CE）インスタンスを HTTP 経由で監視するテンプレート。
- Stormshield SNS by SNMP: SNMP 経由で Stormshield Network Security（SNS）デバイスを監視するテンプレート。
- VeloCloud SD-WAN Edge by HTTP: VeloCloud SD-WAN Edge デバイスを HTTP 経由で監視するテンプレート。
- Vyatta Virtual Router by SNMP: Vyatta 1908e 仮想ルーターを監視するテンプレート。

アプリケーションとサービス

- Domain RDAP by HTTP: RDAP プロトコルを通じてドメイン登録データを監視するテンプレート。
- Envoy Proxy by OTLP: *Telemetry query* アイテムを通じて Envoy Proxy の OpenTelemetry トレースとログを収集するテンプレート。
- Generic OpenTelemetry by OTLP: *Telemetry query* アイテムを通じて OpenTelemetry のトレース、メトリクス、ログを収集するテンプレート。
- GLPI by HTTP: OAuth2 認証を使用し、REST API v2 経由で GLPI を HTTP 監視するテンプレート。
- OpenRouter traces by OTLP: OpenRouter のトレースデータを収集するテンプレート。
- Proxmox VE by OTLP: *Telemetry query* アイテムを通じて、OpenTelemetry メトリクスストリームを使用して Proxmox VE ノードを監視するテンプレート。

更新されたテンプレート

- SNMP テンプレートのデバイス稼働時間トリガーを改善し、SNMP 稼働時間カウンターのリセットに対応するとともに、誤った再起動アラートを防止しました。

オペレーティングシステム

- AIX、FreeBSD、HP-UX、Linux、macOS、OpenBSD、Solaris、Windows のテンプレートで、ファイルシステムの監視に新しい `vfs.fs.get` パラメータを使用するようになりました（軽量ファイルシステムディスカバリを参照）。詳細については、テンプレートの変更を参照してください。
- Linux by Zabbix agentおよびLinux by Zabbix agent activeを更新し、ダッシュボードを改善するとともに、ネットワークインターフェースおよびブロックデバイスの監視を強化しました。
- Linux by PromおよびLinux by SNMPを更新し、ダッシュボードを改善しました。
- Windows by Zabbix agentおよびWindows by Zabbix agent activeテンプレートを強化し、Windows のスケジュールされたタスク、Windows ファイアウォールプロファイル、BitLocker で暗号化可能なボリュームのサポートを含む、新しいメトリクス、ディスカバリルール、プロトタイプ、ユーザーマクロ、ダッシュボードウィジェットを追加しました。

AI 使用状況の監視

- GitHub organization by HTTPを更新し、Microsoft Copilot を監視するアイテムを追加しました。

- Microsoft 365 reports by HTTPを更新し、Microsoft Copilot を監視するアイテムを追加しました。

クラウド

- AWS by HTTPテンプレートセットの *AWS Cost Explorer by HTTP* およびAzure by HTTPテンプレートセットの *Azure Cost Management by HTTP* を更新し、ダッシュボードを改善しました。

仮想化

- Proxmox VE by HTTPを更新し、ネストされた LLD 機能を追加しました。さらに、パーセンテージを表示するアイテムの単位形式を調整し、明確性を向上させました。
- VMwareおよびVMware FQDNテンプレートセットで、VMware アラームディスカバリのアイテムおよびトリガープロトタイプ名を更新し、アラームエンティティ名を含めるようにしました。

データベース

- MySQL by Zabbix agent、MySQL by Zabbix agent 2、MySQL by Zabbix agent active、MySQL by Zabbix agent 2 active、MySQL by ODBCを更新し、`SHOW SLAVE STATUS`（旧構文）と `SHOW REPLICA STATUS`（新構文）の両方をサポートしました。さらに、MySQL by Zabbix agent、MySQL by Zabbix agent active、MySQL by ODBCを更新し、新しいメトリクス、改善されたダッシュボード、テーブルの拡張されたディスカバリルール（*MySQL by ODBC* ではレプリカも含む）を追加しました。
- PostgreSQL by ODBCを更新し、新しいメトリクス、ディスカバリルール、ダッシュボードを追加するとともに、パフォーマンスと可観測性を全般的に改善しました。
- PostgreSQL by Zabbix agentおよびPostgreSQL by Zabbix agent activeを更新し、追加のメトリクス、ディスカバリルール、アイテムおよびトリガープロトタイプ、強化された監視機能を追加しました。
- Redis by Zabbix agent 2を更新し、ユーザーおよびパスワードマクロによる認証と、改善されたアイテムの説明をサポートしました。

ネットワーク、セキュリティ、通信

- 以前は *VMWare SD-WAN VeloCloud by HTTP* だったVeloCloud SD-WAN by HTTPの名前を変更し、最新の VeloCloud SD-WAN プラットフォームに合わせて更新しました。

ストレージとバックアップ

- Ceph by Zabbix agent 2でネイティブ接続モードをサポートするようになりました。詳細については、Ceph プラグインを参照してください。
- Veeam Backup and Replication by HTTPおよびVeeam Backup Enterprise Manager by HTTPを更新し、新しい監視機能と Veeam Backup & Replication v13.1.1.18 をサポートしました。これには、REST API revision 1.3-rev2 および更新されたライセンス有効期限の監視が含まれます。

OS テンプレートでのファイルシステム監視　AIX、FreeBSD、HP-UX、Linux、macOS、OpenBSD、Solaris、および Windows のテンプレート（Linux および Windows テンプレートのアクティブチェック版を含む）では、ファイルシステム監視で新しい `vfs.fs.get` パラメータを使用するようになりました。

- *Get filesystems* アイテム（`vfs.fs.get`）が削除されました。
- *Mounted filesystem discovery* ルールは、従属ディスカバリールールではなく、キー `vfs.fs.get[short]` を持つ Zabbix エージェントアイテムになりました（更新間隔：1 時間）。
- *FS [{#FSNAME}]: Get data* アイテムのプロトタイプ（Windows テンプレートでは *FS [{#FSLABEL}({#FSNAME})]: Get data*）は、キー `vfs.fs.get[full,{#FSNAME}]` を持つ Zabbix エージェントアイテムになり、1 つのファイルシステムのみの統計情報を収集します。
  その他のファイルシステムアイテムのプロトタイプでは、このアイテムをマスターアイテムとして使用します。

これにより、統計情報はファイルシステムごとに個別のエージェントチェックで収集され、新しくマウントされたファイルシステムは、ディスカバリールールの更新間隔（デフォルトでは 1 時間）以内に検出されます。

これらのテンプレートでファイルシステム監視を使用するには、Zabbix エージェント 8.0 以降が必要です。

メディアタイプ

次のメディアタイプが追加されました。

- IBM Maximo Service Request webhook メディアタイプにより、Zabbix と IBM Maximo を連携できます。

次のメディアタイプが更新されました。

- GLPI webhook メディアタイプが更新され、OAuth2 認証（デフォルトで有効）を使用した RESTful API v2、設定フラグによる従来の REST API v1 のオプションサポート、障害イベント発生時のチケット作成、およびユーザーが設定可能な緊急度マッピングをサポートしました。
- MS Teams Workflow webhook メディアタイプが、Microsoft Teams との連携に推奨される方法になりました。Microsoft が 2026 年 5 月 22 日をもって廃止した Office 365 Connector（Incoming Webhook）機能をベースとする MS Teams webhook メ

ディアタイプは、利用できなくなりました。既存の MS Teams webhook 設定は、MS Teams Workflow webhook メディアタイプに移行する必要があります。

また、HTTPS エンドポイントに接続するメディアタイプでは、デフォルトでエンドポイントの TLS 証明書が検証されるようになりました。これは、*Slack*、*Jira*、*Telegram* など、すべての webhook メディアタイプに影響します。webhook メディアタイプをアップグレードする前に、アップグレードに関する注意事項を確認してください。

アイテム

トラッパーアイテムの許可ホストのデフォルト値　Zabbix トラッパーアイテムおよび トラッピングを有効にするが有効なHTTP エージェントアイテムの 許可ホストフィールドは、既定でユーザーマクロ {$TRAPPER.ALLOWED_HOSTS} を使用するようになりました。この値が空の場合、すべてのホストからの受信接続は拒否されます。データの送信を許可するには、1 つ以上の IP アドレスまたは DNS 名を指定してください。

Trend function cache utilization monitoring　paccessed パラメーターが`zabbix[tcache,cache,<parameter>]` 内部アイテムに追加されました。
これは、過去 24 時間以内にアクセスされた trend function cache エントリの割合を返します。

web.page.* アイテムのリダイレクト　Zabbix エージェントおよび Zabbix エージェント 2 のweb.page.get、web.page.perf、web.page.regexp アイテムで、リダイレクトを追跡できるようになりました。リダイレクトの最大数は、新しいオプションの `redirect_limit` パラメータで設定します。デフォルトでは、リダイレクトは追跡されません。

プラグイン

Ceph プラグイン　Zabbix エージェント 2 のCeph プラグイン は、現在 2 つのモードで動作します。

- **native** - このモードでは go-ceph ライブラリを使用し、ネイティブ Ceph API（msgr2 プロトコル）を介して Ceph クラスタと直接通信します。
  これは最新の Ceph 環境に推奨されるモードですが、**Linux** 上でのみ、かつ Ceph 16 以降でサポートされます。
- **restful**（非推奨）- このモードでは、Ceph RESTful API を使用して通信します。
  後方互換性のためのデフォルトモードですが、`mgr/restful` モジュールが削除されたため、Ceph バージョン 20（Tentacle）以降では動作しません。

重要: Zabbix エージェント 2 の Ceph プラグインで Ceph 20（Tentacle）以降を監視する場合は、restful モードはサポートされなくなっているため、native モードを使用してください。

どのモードを使用するかは、mode パラメータの値（native/restful）によって決まります。

- `Plugins.Ceph.Default.Mode=native` - プラグインに native モードを設定します
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - 指定したセッションに native モードを設定します

ユーザー認証情報のセットはモードごとに異なり、互換性はありません。
`Plugins.Ceph.InsecureSkipVerify` パラメータは native モードでは無視されます。これは、接続のセキュリティが Ceph クラスタ側で msgr2 プロトコルによって定義されるためです（デフォルトで安全）。

Zabbix エージェント 2 の Ceph プラグインは現在loadable plugin であり、追加のインストール手順が必要です。
これは librados パッケージが必要なためです（native モード用）。
詳細は Ceph プラグインの readme を参照してください。

MongoDB プラグイン　Zabbix エージェント 2 のMongoDB プラグイン には、次のような複数の機能強化があります。- `mongodb://` と `mongodb+srv://` の両方のスキームに対応し、従来の MongoDB URI の解析を完全にサポートします。- MongoDB レプリカセットのノードを自動検出し、MongoDB クラスターの監視機能を強化します。- x509 認証をサポートし、クライアント証明書を使用して MongoDB へ安全に接続できます。

Oracle プラグイン　Zabbix エージェント 2 のOracle plugin は、TCPS (TLS) プロトコルを使用した Oracle データベースへの暗号化接続をサポートするようになりました。これにより、TLS で保護されたソケット経由で Oracle インスタンスを監視できるようになり、リモート監視のセキュリティが向上します。

Redis プラグイン **— TLS** サポートと起動時検証　Zabbix エージェント 2 のRedis プラグイン に TLS サポートが追加されました。

プラグインの TLS 設定に対する起動時検証が実装され、検証/エラーメッセージも改善されました。
不正な設定ロジック（たとえば、`TLSCAFile` を指定せずに接続タイプ `verify_full` を使用する場合）は、Zabbix エージェント 2 の起動を妨げる可能性があります。

ロード可能なプラグインのテスト実行モード　ロード可能なプラグイン は、アイテムキーを引数として渡し、`-t` (`--test`) フラグを使用してテストモードで起動できるようになりました。
このモードでは、プラグインはデバッグおよび開発目的で実行され、プラグインの設定ファイルは無視されます。

ディスカバリと自動登録

プロキシグループの条件　プロキシグループの条件が、ディスカバリおよび自動登録アクションでサポートされるようになりました。

ローレベルディスカバリ

ディスカバリーフォームの **JSON** に変換チェックボックス　ローレベルディスカバリーのディスカバリールールフォームおよびディスカバリープロトタイプフォームに、新しい *JSON* に変換チェックボックスが追加されました。このチェックボックスは、タイプドロップダウンで「HTTP エージェント」を選択した場合に表示されます。このオプションを使用すると、取得したデータを後続の処理の前に自動的に JSON に変換するよう指定できます。

ネストされたローレベルディスカバリのマクロサポート　ローレベルディスカバリマクロは、以下のネストされたローレベルディスカバリルールでサポートされるようになりました:

- JSONPath の前処理パラメータ
- カスタム LLD マクロの JSONPath フィールド

ローレベルディスカバリで作成されたトリガーのタグの編集　トリガープロトタイプから作成されたトリガーに、手動でタグを追加できるようになりました。トリガープロトタイプから継承されたタグは、引き続き自動的に適用されます。手動で追加されたタグは、検出されたトリガーで変更でき、イベントタグ配列に含まれ、フィルター、ダッシュボードウィジェット、通知マクロなどのタグ対応機能で利用できます。

Linux と **AIX** におけるブロックデバイスの検出　Linux と AIX のブロックデバイス検出用に、新しい`vfs.dev.get` アイテムキーが追加されました。
これは、ブロックデバイスの低レベルディスカバリ で使用できます。
`vfs.dev.discovery` アイテムキーは、互換性のため引き続き利用可能です。

Linux でのネットワークインターフェースの検出　新しい`net.if.get`アイテムキーが、Linux 上の Zabbix エージェントおよび Zabbix エージェント 2 に追加されました。
これは、ネットワークインターフェースの低レベルディスカバリで使用でき、詳細情報を含むネットワークインターフェースの一覧を返します。
`net.if.discovery`アイテムキーは、互換性のため引き続き利用できます。

軽量なファイルシステムディスカバリ　Zabbix エージェントおよび Zabbix エージェント 2 の `vfs.fs.get` アイテムで、ファイルシステムのディスカバリを軽量化し、信頼性を高める 2 つのオプションパラメータがサポートされるようになりました。

- **mode** - `short` モードでは、ディスク容量および inode 統計を含めず、マウントされたファイルシステムのリストと基本プロパティのみを返します。ファイルシステム自体に対してクエリを実行しないため、ディスカバリによって automount がトリガーされることはなく、応答しないネットワークファイルシステムの影響も受けません。`full` モード（デフォルト）では、従来と同じデータを返します。
- **mountpoint** - 指定したマウントポイントのデータのみを返します。これにより、検出された各ファイルシステムの統計を個別に収集できます（例: `vfs.fs.get[full,{#FSNAME}]`）。

OS テンプレートでは、この方法が使用されるようになりました。詳細については、テンプレートの変更を参照してください。

詳細については、vfs.fs.getおよびマウントされたファイルシステムのディスカバリを参照してください。

データベース

複数の **PostgreSQL** データベースホストのサポート　Zabbix では、PostgreSQL の複数のデータベースホストを、カンマ区切りの `host:port` 値のリストとして指定できるようになりました。これにより、PostgreSQL の接続フェイルオーバー機能を活用できます。ホストは、読み書き可能な接続が確立されるまで順番に試行されます。
この動作は、Web インターフェースのインストール の *Database host* フィールド、および `zabbix_server.conf` と `zabbix_proxy.conf` のDBHost で利用できます。

最適化された **Zabbix** サーバーの起動　Zabbix サーバーは、これまでより大幅に高速に起動するようになりました。
これは、履歴データを 1 件ずつではなく、複数のアイテムについてデータベースから一度に読み込むことで実現しています。
この改善は、Zabbix サーバーがサポートするすべてのデータベース種別に適用されます。

Elasticsearch のパフォーマンスの最適化　Zabbix サーバーは、Elasticsearchとの連携がより効率的になりました。
サーバーは Elasticsearch の *scroll* 機能を使用せずにデータをより効率的に取得し、各リクエストごとに新しい接続を開く代わりに、既存の Elasticsearch 接続を再利用します。
また、cURL の *multi-handle* を使用して Elasticsearch に複数のリクエストを同時に送信した際に、サーバーが停止することもなくなりました。

プロセス

SNMPv3 EngineID のキャッシュと再利用　Zabbix は、SNMPv3 EngineID → IP のマッピングをキャッシュし、以降の SNMPv3 チェックでキャッシュされた EngineID を再利用しようとします。これにより、プローブトラフィックが削減され、ポーラーのパフォーマンスが向上します。再利用された EngineID が応答しない場合、ポーラーは EngineID プローブにフォールバックし、インターフェースの変更や永続的な障害の後に古いエントリを削除することがあります。

履歴キャッシュ復旧時のプロキシのスロットリングの改善　プロキシのスロットリングロジックが改善され、履歴キャッシュ復旧中のサーバーの安定性が向上しました。
履歴キャッシュの使用率がスロットリングしきい値に達すると、サーバーは従来どおりプロキシデータの受け入れを停止し続けます。キャッシュ使用率が 60% まで低下すると、サーバーはスロットリングリストの処理を開始しますが、キャッシュへの負荷がさらに下がるまでは、非常に大きなバッチ（およそ 1 万レコード超）を含むプロキシのアップロードを引き続き拒否する場合があります。この変更により、サーバーの復旧中にキャッシュが繰り返し過負荷になるリスクが低減されます。

zabbix_get と **zabbix_js** の最大タイムアウト時間を延長　zabbix_get およびzabbix_js コマンドラインユーティリティの `timeout` パラメータの最大値が 600 秒に引き上げられました。

最適化されたハウスキーピング　ハウスキーピング は、ハウスキータスクの作成をデータベーストリガーに移行することで最適化されました。アイテム（ローレベルディスカバリルールを含む）、トリガー、サービス、またはネットワークディスカバリルールが削除されると、データベーストリガーによって（アプリケーションレベルでの明示的な呼び出しの代わりに）`housekeeper` テーブルにクリーンアップタスクが登録されるようになりました。詳細については、ハウスキーピング手順を参照してください。

ハウスキーパーは現在、削除済みのディスカバリ済みホストまたはサービスから生成されたネットワークディスカバリイベントに加え、削除済みのトリガーの障害に関連付けられたすべてのイベントも削除します（以前は障害自体のみが削除され、関連するイベントはハウスキーパーの トリガーデータ保存期間が期限切れになった後にのみ削除されていました）。

手動クローズ ― 回復イベントに継承されるトリガータグ　手動クローズ の後に作成された回復イベントには、アイテムタグおよびホストタグに加えてトリガータグが継承されます。これらのタグはイベントタグ配列に含まれ、`{EVENT.RECOVERY.TAGS}` や`{EVENT.RECOVERY.TAGSJSON}` などの通知マクロで利用できます。

Zabbix デーモンの **DNS** クエリキャッシュ　Zabbix サーバー、Zabbix プロキシ、Zabbix エージェントは、`--with-ares` でビルドされた場合、すべての DNS リクエストに c-ares リゾルバーを使用することができるようになりました。これにより、DNS クエリキャッシュとリゾルバーのフェイルオーバーが改善されます。DNS クエリキャッシュには c-ares 1.26.0 以降が必要です。

Windows での **c-ares** サポートのビルド　Zabbix エージェントは、Microsoft Windows 上で c-ares リゾルバーを使用してビルドできるようになりました。c-ares は vcpkg 経由でインストールでき、エージェントのビルドでは、インクルードディレクトリとライブラリディレクトリのパスとして、`ARES=<vcpkg prefix>` または個別の `ARESINCDIR`/`ARESLIBDIR` のいずれかをサポートしています。

カウントベースのトリガー関数の改善　トリガー式におけるカウントベースの関数（count()、item_count()、changecount() など）の動作が改善され、より多くの値をより高速に処理できるようになりました。

ランタイム制御対象の更新　Zabbix サーバーおよびプロキシのヘルプメッセージとmanpagesに表示される、利用可能なruntime control対象の一覧が、サポートされているprocess typesに合わせて更新されました。

`supervisor` 対象が追加され、サポートされていない `discovery worker` および `preprocessing worker` 対象は削除されました。

セキュリティ

プロキシとサーバーのアクセス制御　user group configuration に新しい プロキシアクセスリストタブが追加されました。このタブでは、個別のプロキシまたはプロキシグループを通じて、ユーザーグループにプロキシへのアクセス権を付与できます。

user role configuration に新しい 監視とディスカバリに「サーバー」を選択オプションが追加されました。このオプションを使用すると、ユーザーがサーバーによって監視されるホストや、サーバーによって実行されるネットワークディスカバリルールを作成したり、サーバーでアイテムをテストしたりすることを禁止できます。

Zabbix エージェントおよび **Zabbix** エージェント **2** の設定における新しい **AllowKeyRegexp** および **DenyKeyRegexp** パラメータ　新しい `AllowKeyRegexp` および `DenyKeyRegexp` パラメータが、Zabbix エージェント および Zabbix エージェント 2 の設定に追加されました。これらのパラメータを使用すると、正規表現ベースのパターンを定義して、キーとそのパラメータをより正確かつ柔軟に照合できます。

認証

シングルサインオン用のインポート可能な **SAML** 証明書　Super admin ユーザーは、SAML設定のために、証明書と秘密鍵を Web インターフェース上で直接インポートできるようになりました。*Users* > *Authentication* > *SAML settings* に、次の 3 つの新しいフィールドが追加されました。

- *IdP certificate* - ID プロバイダーによって提示される X.509 証明書
- *SP certificate* - SAML 通信に使用されるサービスプロバイダー証明書
- *SP private key* - SP 証明書に対応する秘密鍵

これらのコントロールにより、SAML 設定タブから新しい値を追加したり、既存の値を変更したりできます。証明書と秘密鍵は、選択したストレージバックエンドに保存される前に検証されます。無効または不正な形式の値は、説明付きのエラーとともに拒否されます。

Post-login URL per user role　It is now possible to enforce a specific URL after login based on the user role. This is handy for enforcing a specific page after login for provisioned users.

For more details, see User profile settings in user role configuration.

AppRole メソッドを使用した **HashiCorp vault** 認証　HashiCorp vault 認証で AppRole メソッドがサポートされるようになりました。

- サーバー/プロキシの設定では、新しい `VaultAppRoleID` および `VaultAppSecretID` パラメータを使用します（例を参照）。
- Web インターフェースの設定では、認証用のロール ID とシークレット ID の値を指定します。

ウィジェット

クロックウィジェットのタイムゾーンオプションの変更　クロックウィジェットの設定フォームのタイムゾーンオプションが、詳細設定セクションからウィジェットのメイン設定セクションに移動され、タイムゾーンを上書きに名称変更されました。

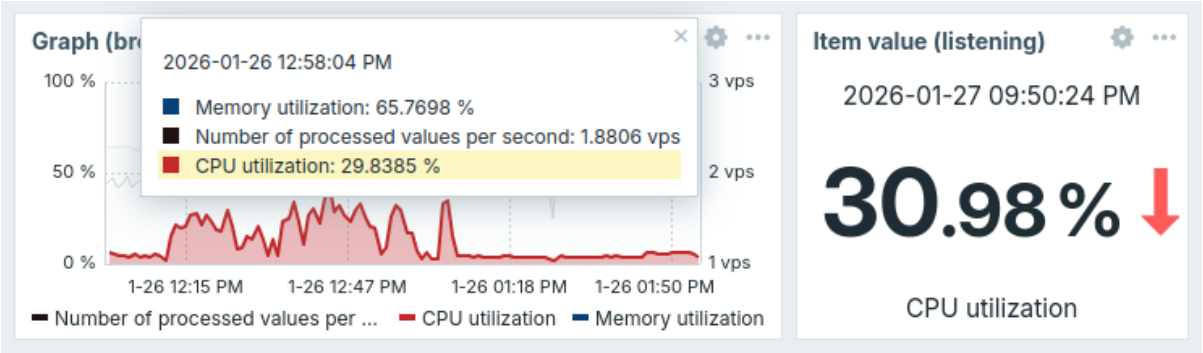
このオプションは、時刻タイプがローカル時刻およびサーバー時刻の場合に使用できます。

ジオマップホストマーカーのクラスタリング　ジオマップウィジェットは、近くのホストマーカーをカウント付きの単一マーカーにまとめる方法を制御するクラスタリングパラメータをサポートするようになりました。クラスタリングのためのマップズームレベルを設定することで、大きなマップの可読性を保ち、ズームイン時の正確な可視性を維持できます。

Top items での列の集計　Top itemsウィジェットで、アイテムのパターンを 1 つの集計列または行にグループ化できるようになりました。これにより、結合したメトリクス（例：ホストのすべてのネットワークインターフェースにおける受信トラフィックの合計）を簡単に表示できます。アイテム列の設定時に、次の 3 つのオプションが追加されました：列の集計、列の集計関数、結合列名。

グラフウィジェットの機能強化　グラフウィジェットに、複数の使いやすさに関する機能強化が追加されました。

- グラフ上にカーソルを合わせたときに表示されるグラフデータのツールチップ内のアイテムが、値の降順で並べ替えられます。
- ツールチップ内のアイテムにカーソルを合わせると、そのグラフが強調表示され、他のグラフが薄く表示されます。
- ツールチップ内のアイテムを選択すると、そのデータが、データの受信に対応している他のウィジェットにブロードキャストされます。
- ツールチップに表示されるアイテム数の上限がなくなりました。
- 新しいラベル内のホスト名設定により、ツールチップとグラフの凡例にホスト名を表示するかどうかを選択できます。
- 新しい値を反転設定により、グラフを反転できます。



ウィジェットの説明フィールドの文字数制限を拡大　Gauge およびItem value ウィジェットの *Description* フィールドの文字数制限が、2048 文字から 65535 文字に拡大されました。

フロントエンド

インポート可能なダッシュボード　新しいインポート可能なダッシュボードがダッシュボードリポジトリで利用できるようになりました。

- マルチクラウドコストダッシュボード - AWS、Azure、Google Cloud、Oracle Cloud 全体のクラウドコストを監視するためのダッシュボードです。
- マルチクラウド仮想マシンダッシュボード - AWS、Google Cloud、Oracle Cloud、Azure で実行されている仮想マシンを監視するためのダッシュボードです。

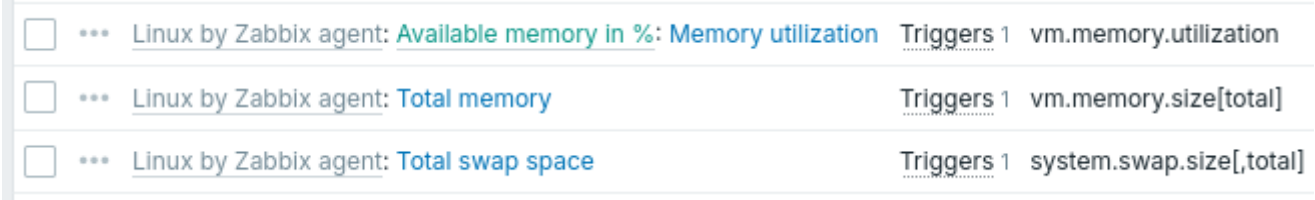
これらのダッシュボードは、複数のホストおよび監視対象からのデータを統合されたグローバルダッシュボードにまとめることで、既存の Zabbix インテグレーションと連携するように設計されています。

ダッシュボードのエクスポートとインポート　ダッシュボードは、エクスポートおよびインポートできるようになり、Zabbix インスタンス間でダッシュボードを移行したり、異なる環境で再利用したりしやすくなりました。

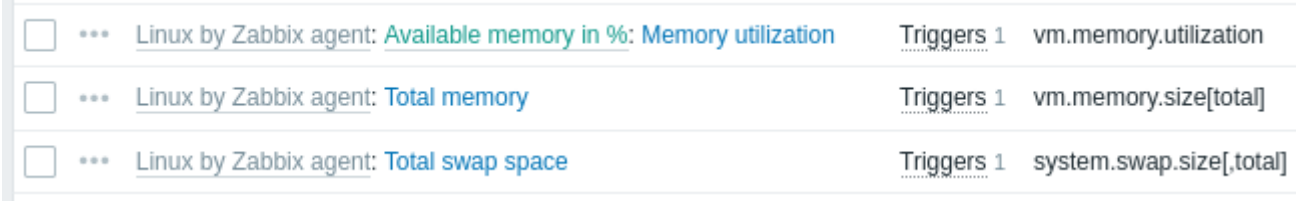
新しいエクスポートオプションにより、選択したダッシュボードを YAML、JSON、または XML ファイルとして保存できます。これらのファイルは、後でインポート機能を使用して取り込み、ページ、ウィジェット、設定を含むダッシュボードを再作成できます。

新しい埋め込みフォント　Zabbix には、可読性を向上させ、読み込みを高速化し、レイアウトへの影響を最小限に抑えながら Web インターフェースの追加言語を表示できる、新しい埋め込みフォントが含まれるようになりました。
埋め込みフォントであるため、これらのフォントはシステム間で一貫した表示も保証します。

新しいフォント:



従来のフォント:

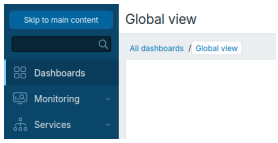


新しいフォントは、ほぼすべてのテーマで使用されています。
必要に応じて、従来のフォントを使用する、新たに追加された *Blue (classic)* および *Dark (classic)* テーマを利用できます。

等幅コンテンツおよびグラフのフォントファミリーは変更されていません。

Zabbix UI の機能強化　アクセシビリティと使いやすさを向上させるため、Zabbix のユーザーインターフェースが刷新されました。

アクセシビリティの向上　キーボード操作とアクセシビリティを改善するために、メインコンテンツへスキップ リンクが追加されました。



このリンクを使用すると、キーボード利用者やスクリーンリーダー利用者は、繰り返し表示されるサイドバーのナビゲーションを飛ばして、直接メインコンテンツへ移動できます。通常の使用時には非表示のままで、ページ読み込み後に Tab キーを押した場合や、スクリーンリーダーでアクセスした場合など、フォーカスが当たったときにのみ表示されます。

インライン検証　Web インターフェースの以下のフォームが、インライン検証をサポートするフォームのセットに追加されました。

- API token
- 監査ログ
- 認証
- 自動登録

- Connector
- ディスカバリプロトタイプ
- Low-level ディスカバリルール
- Network ディスカバリルール
- イベント相関
- 地理マップ
- グラフプロトタイプ
- カスタムグラフ
- GUI
- ホストグループ
- ホストプロトタイプ
- ハウスキーピング
- アイコンマッピング
- 画像
- メンテナンス
- メディアタイプ
- 通知
- その他の設定パラメータ
- プロフィール
- プロキシ
- プロキシグループ
- 正規表現
- 定期レポート
- スクリプト
- サービス
- SLA
- テンプレートグループ
- テストアイテム
- テストメディアタイプ
- タイムアウト
- トリガー表示オプション
- トリガー式コンストラクタ
- 障害の更新
- ユーザーグループ
- グローバルユーザーマクロ
- ユーザーロール
- ユーザー

入力エラーはフィールドへの入力直後に表示されるため、使いやすさが向上し、設定ミスを減らせます。

モーダルフォーム　以下のフォームは、現在モーダルウィンドウで開かれます。

- ディスカバリプロトタイプ の設定フォーム
- 低レベルディスカバリルール の設定フォーム
- ホストプロトタイプ の設定フォーム
- グローバル正規表現 の設定フォーム

ホスト、テンプレート、アイテム、トリガーで継承されたタグを表示　継承されたタグが、テンプレート、ホスト、アイテム、Web シナリオ、トリガー全体で一貫して表示および返されるようになりました。
テンプレートやホストのチェーンから継承されたタグは、監視 > 最新データ に表示されます。
継承されたタグによるフィルタリングは、タグフィルタリングがサポートされているすべての場所で利用できます。これには、監視 > 最新データ、データ収集 セクション、およびホスト、アイテム、トリガー、Web シナリオのタグによるフィルタリングに対応するすべてのダッシュボードウィジェット が含まれます。これにより、タグが定義された場所に関係なく、タグに基づく選択とサブフィルタリングが同じように動作します。

テンプレート、ホスト、ホストプロトタイプの設定フォームにある タグタブに、タグの表示方法を選択するラジオボタンが追加されました。テンプレート では テンプレートのタグ / 継承されたタグとテンプレートのタグを、ホスト とホストプロトタイプでは ホストのタグ / 継承されたタグとホストのタグを選択できます。
継承されたタグは、タグラベルの横に表示される新しい枠線付きドキュメントアイコンによって視覚的に区別されます。

グラフ ウィジェットと円グラフ ウィジェットに、新しい アイテムのタグ設定が追加されました。

ツールチップの配置　ツールチップは、ドラッグして新しい位置に移動することで再配置できるようになりました。
これは、たとえばグラフのツールチップや、最新データまたは障害セクション内の説明付きツールチップに適用されます。

Web インターフェース設定の新しい **$ZBX_FEATURE_FLAGS** 変数　ユーザーの HTTP 認証、モジュール、メディアタイプの設定へのアクセスを制御するため、Web インターフェース設定に新しい $ZBX_FEATURE_FLAGS[] 変数が実装されました。以下の機能フラグが zabbix.conf.php 設定ファイルに追加されました。

- $ZBX_FEATURE_FLAGS['http_auth_enabled'] = false は、ユーザーの HTTP 認証設定へのアクセスを無効にします（$ALLOW_HTTP_AUTH 変数を置き換えます）。
- $ZBX_FEATURE_FLAGS['modules_config_enabled'] = false は、モジュール設定へのアクセスを無効にします。
- $ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]　は、指定した配信方法（'email'、'script'、'sms'、または 'webhook'。1 つ以上の値を指定可能）を使用するメディアタイプの編集を制限します。

zabbix.conf.php 設定ファイルが存在する場合、setup.php へのアクセスは制限されます。

イベント詳細に抑制理由を表示　*Monitoring* > *Problems* から開くイベント詳細 ページでは、イベントがホストのメンテナンスによって抑制されたのか、ユーザーによって手動で抑制されたのかが表示されるようになりました。

オレンジ色のアイコン  と  はメンテナンスに関連する抑制/抑制解除を示し、灰色のアイコン  と  は手動の抑制/抑制解除を示します。

障害 およびイベント の履歴では、削除されたユーザーによって実行された操作が *Inaccessible user* として表示されるようになりました。

システム情報レポートの拡張　システム情報 レポートにサーバー ID (UUIDv7) が表示されるようになりました。
 システム情報レポートは、Zabbix の *Admin* ユーザーでも利用できるようになりました。

Zabbix SELinux ポリシーパッケージの変更　AlmaLinux、CentOS Stream、RHEL、Rocky Linux、および Oracle Linux のバージョン 10 では、SELinux ポリシーは新しい zabbix-selinux-policy-main パッケージによって提供されます。このパッケージには、上流の SELinux リポジトリの SELinux ポリシーが含まれています。

既存の zabbix-selinux-policy パッケージは､現在 zabbix-selinux-policy-main または selinux-policy-targeted-extra のいずれかに依存します。
zabbix-selinux-policy-main パッケージは selinux-policy-targeted-extra と競合するため、両方のパッケージを同時にインストールすることはできません。

すでに zabbix-selinux-policy をインストールしている場合、手動での変更は不要です。
パッケージのアップグレード時に、zabbix-selinux-policy-main が自動的にインストールされます。
selinux-policy-targeted-extra がすでにインストールされている場合は、追加のパッケージはインストールされません。

ダッシュボードのスライドショー設定　ダッシュボードを作成する際、スライドショーを開始チェックボックスはデフォルトで無効になっているため、明示的に有効にする必要があります。
slideshow URL パラメータでは on または off の値を指定でき、ダッシュボードのスライドショー設定をすばやく上書きできます。
slideshow=1 パラメータは非推奨であり、今後サポートされなくなります。

Zabbix Cloud プロキシへのリンク　*Administration* > *Proxies* のプロキシリスト に、*Create proxy in Cloud* ボタンが追加されました。このボタンをクリックすると、新しいブラウザータブで Zabbix Cloud が開き、自身のインフラストラクチャにインストールせずにプロキシをデプロイできます。詳細については、Zabbix Cloud プロキシのセットアップを参照してください。

ドキュメント

マイナーリリース向けの統合ドキュメントページ　メジャーな Zabbix リリースのマイナー版に関するリリースドキュメントは、今後、新機能 とアップグレードノートの両方を含む 1 つのドキュメントページにまとめられます。

アップグレードノート

これらのノートは、Zabbix 7.4.x から Zabbix 8.0.0 へのアップグレード向けです。

すべてのノートは次のように分類されています。

- 破壊的変更 - 既存のインストールを壊す可能性のある変更、およびアップグレード手順に関連するその他の重要な情報
- その他 - Zabbix の機能変更に関する残りのすべての情報

関連情報:

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

破壊的変更

ユーザーロールからホストインターフェース **API** メソッドを削除　Zabbix のアップグレード後、`hostinterface.create`、`hostinterface.update`、`hostinterface.delete`、`hostinterface.massadd`、`hostinterface.massremove` API メソッドは、既存のユーザーロールの *API* へのアクセスリストから削除されます。

カスタムテーマとメディアタイプ　Zabbix には、3 つの新しい定義済み Web インターフェーステーマ（*Blue (classic)*、*Dark (classic)*、*Dark blue*）と、新しいPush notificationメディアタイプが追加されます。これらの名前（`blue-classic-theme`、`dark-classic-theme`、`dark-blue-theme`）のいずれかを使用するカスタムテーマがある場合、または `Push notification` という名前のメディアタイプがある場合は、アップグレード前にテーマまたはメディアタイプの名前を変更してください。変更しない場合、アップグレード後に次のようになります。

- 定義済みテーマによってカスタムテーマのスタイルシートが置き換えられます。一方、カスタムグラフの色は引き続き使用されるため、両方のテーマが混在した状態になります。
- 既存のメディアタイプは変更されませんが、新しいメディアタイプは作成されません。

その他の名前のカスタムテーマおよびメディアタイプは、アップグレードの影響を受けません。

サーバー監視ホストに制限された内部プロキシアイテム　内部アイテム `zabbix[proxy,*]` および `zabbix[proxy group,*]` は、プロキシまたはプロキシグループによって監視されるホストではサポートされなくなりました。

1.6 未満の **Zabbix** エージェントのサポート終了　Zabbix サーバー/プロキシのトラッパーから従来のプレーンテキストおよび XML プロトコルのサポートが削除されたため、1.6 未満の Zabbix エージェントはサポートされなくなりました。

Logback ライブラリの最小バージョン　Logback ライブラリの最小バージョンが 1.5.16 から 1.5.25 に引き上げられました。

インライン検証　Web インターフェースの以下のフォームが、インライン検証をサポートするフォームのセットに追加されました。

- API token
- Audit log
- Authentication
- Autoregistration
- Connector
- Event correlation
- Geographical maps
- Graph prototypes
- Custom graphs
- GUI
- Host group
- Host prototypes
- Housekeeping
- Icon mapping
- Images
- Maintenance
- Media type
- Notifications
- Other configuration parameters
- Profile
- Proxy
- Proxy group
- Regular expressions
- Scheduled reports
- Script
- Service
- SLA
- Template group
- Test item
- Test media type
- Timeouts
- Trigger displaying options
- Trigger expression constructor
- Update problem
- User groups
- Global user macros

- User roles
- Users

入力エラーはフィールド入力後すぐに表示されるため、使いやすさが向上し、設定ミスを減らせます。

プラグイン　Zabbix エージェント 2 用の Ceph プラグインは、現在ロード可能なプラグイン であり、追加のインストール手順が必要です。詳細については、Ceph プラグインの readme を参照してください。

MongoDB プラグイン では `tcp://` スキームは非推奨ですが、既存の設定との後方互換性のために保持されています。

Oracle プラグインの設定ファイル にある `Plugins.Oracle.CallTimeout` パラメータは、もはや効果を持ちません。最大待機時間は、アイテムのタイムアウト によって決定されます。

MongoDB プラグインのバージョンサポート　このプラグインバージョンでは、MongoDB 2.6～4.0 はサポートされなくなりました。MongoDB Go Driver v2を使用しており、MongoDB 4.2 以降が必要です。

MongoDB 2.6～4.0 を監視するには、以前のバージョンのプラグインを使用してください。
以前のバージョンは MongoDB 2.6～5.3 をサポートしており、Zabbix エージェント 2 バージョン 6.0.0 以降で使用できます。

必要な最小 **PHP** バージョン　必要な PHP の最小バージョンは、8.0.0 から 8.2.0 に引き上げられました。

拡張 **UnsafeUserParameters** リスト　% 文字が、Zabbix エージェント とZabbix エージェント 2 の `UnsafeUserParameters` リストに追加されました。

フィルターパラメータの移動　カスタマイズ可能なテーブル機能の一環として、一部のフィルターパラメータがフィルタータブから列オプションに移動されました。以下のフィルターパラメータを使用するお気に入りのフィルター用タブを保存している場合、アップグレード後にリセットされるため、手動で再設定する必要があります。

- 監視 > ホスト: 抑制された障害を表示
- 監視 > 最新データ: タグを表示, タグ名, タグ表示の優先度, 詳細を表示
- 監視 > 障害: タグを表示, タグ名, タグ表示の優先度, 運用データを表示, コンパクト表示, 詳細を表示, タイムラインを表示, 行全体を強調表示

JavaScript の前処理**:** 組み込みオブジェクトのメソッドは読み取り専用になりました　前処理スクリプト内の JavaScript 組み込みオブジェクトのメソッドは、読み取り専用になりました。アイテムの前処理スクリプトで JavaScript のプロトタイプ上の組み込みメソッドをすでに変更している場合、組み込みのプロトタイプメソッドの変更はもはや許可されないため、問題が発生する可能性があります。

MySQL データベースの文字セットと照合順序　Zabbix の MySQL データベースが utf8mb3 文字セットおよび utf8mb3_bin 照合順序で作成されている場合は、Zabbix データベースの文字セットと照合順序の修復 に記載されている手順に従って、Zabbix をアップグレードする前に utf8mb4 文字セットおよび utf8mb4_bin 照合順序へ変換することを強く推奨します。

その他

SAML 署名がデフォルトで有効化　セキュリティを強化するため、ユーザー > 認証でSAML 認証を設定する際、すべての SAML 署名オプション（メッセージ、アサーション、*AuthN* リクエスト、ログアウトリクエスト、ログアウトレスポンス）がデフォルトで有効になりました。その結果、署名オプションを無効にしない限り、新しい SAML 設定では SP 秘密鍵と SP 証明書が必須になりました。

既存の SAML 設定は、アップグレードの影響を受けません。

非推奨の **Kubernetes** テンプレート　Kubernetes のテンプレートは非推奨となり、*Kubernetes Cluster by HTTP* テンプレートに置き換えられました。古いテンプレートは、既存の設定との後方互換性のために保持されています。

非推奨マクロの廃止　以下の組み込みマクロのサポートは廃止されました。

廃止	代替
{ACK.DATE}	{EVENT.UPDATE.DATE}
{ACK.MESSAGE}	{EVENT.UPDATE.MESSAGE}
{ACK.TIME}	{EVENT.UPDATE.TIME}
{EVENT.ACK.HISTORY}	{EVENT.UPDATE.HISTORY}
{HOSTNAME<1-9>}	{HOST.HOST}
{IPADDRESS<1-9>}	{HOST.IP}
{PROFILE.*}	{INVENTORY.*}
{TRIGGER.COMMENT}	{TRIGGER.DESCRIPTION}

廃止	代替
{TRIGGER.KEY}	{ITEM.KEY}
{STATUS}	{TRIGGER.STATUS}
{USER.ALIAS}	{USER.USERNAME}

履歴ストレージプロバイダーの非推奨パラメータ　以下のパラメータは非推奨になりました。

- Zabbix サーバー設定ファイル: HistoryStorageDateIndex、HistoryStorageURL、HistoryStorageTypes。代わりに、新しいHistoryProvider パラメータを使用してください。
- Zabbix Web インターフェース設定ファイル: $HISTORY。代わりに、新しい$HISTORY_PROVIDERS パラメータを使用してください。

どちらの新しいパラメータでも、履歴ストレージプロバイダーとして、Elasticsearch、ClickHouse、または SQL（デフォルト）を選択できます。これらの新しいパラメータと非推奨のパラメータを同時に使用することはできません。

HTTP エージェントアイテムから **JSON** および **XML** の検証を削除　HTTP エージェントアイテムの *Request body* フィールドから、JSON および XML の検証が削除されました。このフィールドでは値が有効な XML または JSON かどうかを確認しなくなり、*Request body type* として *XML data* を選択しても libxml2 ライブラリは不要になりました。

監視対象ホストで無効なリンクを非表示　無効な *Graphs*、*Dashboards*、および *Web* リンクは、監視対象ホスト の一覧に表示されなくなりました。

JSON データ型　Zabbix では、アイテムの値のデータ型として JSON がサポートされるようになりました。TimescaleDB を使用している場合は、JSON 値の保存に使用される新しい history_json ハイパーテーブルを手動で設定する必要があります。Elasticsearch を使用している場合、非推奨となった Zabbix サーバー設定パラメータHistoryStorageTypesのデフォルト値に json が含まれるようになりました。

新しいタグフィルター形式からラジオボタンを削除　新しいタグフィルターを作成する際に、*All tags* と *Tag list* のどちらかを選択する必要があったラジオボタンは削除されました。

iframe を使用して **HTML** として表示されるアイテム値　Item history ウィジェット内のアイテムデータは、HTML 形式のテキストとして表示される場合、現在は iframe 内に分離されています。

HTTP ベースのテンプレートおよびメディアタイプでの **TLS** 検証　HTTPS エンドポイントに接続するテンプレートおよびメディアタイプでは、デフォルトでエンドポイントの TLS 証明書を検証するようになりました。

この変更は、すべてのクラウドおよび API 監視テンプレート（*Claude API by HTTP*、*Cloudflare by HTTP*、*Kubernetes Cluster by HTTP* など）と、すべての webhook メディアタイプ（*Slack*、*Jira*、*Telegram* など）に適用されます。この動作は、これらのテンプレートおよびメディアタイプに追加された新しい {$HTTP.TLS.VERIFY} ユーザーマクロで変更できます。検証を無効にするにはマクロの値を none に設定し、証明書チェーンと有効期限のみを検証するには peer に設定し、完全な検証を行うには full に設定します。詳細については、各HTTP テンプレートまたはwebhookの README ファイルを参照してください。

Zabbix をアップグレードしても、テンプレートおよびメディアタイプは自動的にアップグレードされません。これは、ユーザーが行ったカスタム変更が上書きされるのを防ぐためです。テンプレートおよびメディアタイプを手動でアップグレードまたは追加するには、テンプレートのアップグレードの手順を参照してください。

テンプレートのホストウィザードとの互換性　すべての標準テンプレートは、現在Host Wizard と互換性があります。それらをアップグレードするには、Template upgrade を参照してください。

設定フィールドの自動サイズ調整　UI のほとんどの設定フィールドは、内容に合わせて自動的にサイズが調整されるようになりました。

Web インターフェース設定変数の置き換え　$ALLOW_HTTP_AUTH 変数は、Web インターフェース設定ファイル (zabbix.conf.php) 内の $ZBX_FEATURE_FLAGS['http_auth_enabled'] フィーチャーフラグに置き換えられました。

SNMPv1 と **SNMPv2** チェックの並列処理　SNMPv1 および SNMPv2 の非同期サービスチェックは、利用可能なディスカバリーワーカーによって並列に処理されるようになりました。

Zabbix SELinux ポリシーパッケージの変更　AlmaLinux、CentOS Stream、RHEL、Rocky Linux、および Oracle Linux のバージョン 10 では、既存の zabbix-selinux-policy パッケージが zabbix-selinux-policy-main または selinux-policy-targeted-extra のいずれかに依存するようになりました。
 既存の zabbix-selinux-policy のインストールがある場合、手動での変更は不要です。パッケージのアップグレード時に、zabbix-selinux-policy-main が自動的にインストールされます。selinux-policy-targeted-extra がすでにインストールされている場合は、追加のパッケージはインストールされません。

APM support in Zabbix proxy packages To use application performance monitoring after upgrading Zabbix proxy, choose the option for your operating system:

- RHEL 10 and its derivatives - install the EPEL repository, enable the CRB repository, and install the Zabbix proxy APM package (see details).
- Debian 12 or later, Raspberry Pi OS (Raspbian) 12 or later, Ubuntu 24.04 or later - upgrade Zabbix proxy packages as usual; APM is included.
- Other systems - compile Zabbix proxy from sources with the `--with-apm` option.

メンテナンス期間の状態の名称変更　メンテナンス期間の状態は、*Approaching* から *Upcoming* に名称変更されました。

非推奨のダッシュボードスライドショーパラメータ　`slideshow=1` パラメータのサポートは削除されます。スライドショーを開始および停止するために、`slideshow` パラメータに新しい値が導入されました。

- `slideshow=on` - スライドショーを開始する
- `slideshow=off` - スライドショーを停止する

グローバル正規表現の変更

- 正規表現の設定フォームに *Description* フィールドが追加されました。
- *Expression type* のオプション名が変更されました:

Before	Now
Character string included	*Contains string*
Any character string included	*Contains any substring from list*
Character string not included	*Does not contain string*
Result is TRUE	*Matches regular expression*
Result is FALSE	*Does not match regular expression*

API へのデフォルトアクセス　新規インストールでは、新しく作成されたロールやデフォルトのユーザーロールを含む、すべてのロールで API へのアクセスがデフォルトで無効になっています。

既存のインストールでは、API へのアクセスがデフォルトで無効になるのは、新しく作成されたロールのみです。

Empty API allow list In user roles, an *Allow list* with no API methods specified now denies access to all API methods. Previously, the empty list was ignored, and all API methods were allowed.

To preserve existing access, the upgrade adds the * wildcard (any method) to all user roles that have *Allow list* selected but no methods specified. After the upgrade, review these roles in *Users* > *User roles*: replace * with the methods the role actually needs, or clear the *Enabled* checkbox in the *Access to API* section if the role does not need API access.

新機能とアップグレードに関する注意事項**: 8.0.x**

このページでは、Zabbix メジャーバージョンのマイナーリリースに含まれるすべての新機能を紹介します。テンプレートの変更点とアップグレードに関する注意事項も記載しています。

メジャーバージョンの新機能も参照してください。

Zabbix 8.0.1 の新機能　このバージョンはまだリリースされていません。

このバージョンのアップグレードノートを参照してください。

アップグレードノート

Zabbix アプライアンス

概要　Zabbix アプライアンスを使用すると、手動でセットアップしたり、既存のサーバーを Zabbix 用に再利用したりする代わりに、Zabbix サーバーと Web インターフェースをすぐにデプロイできます。

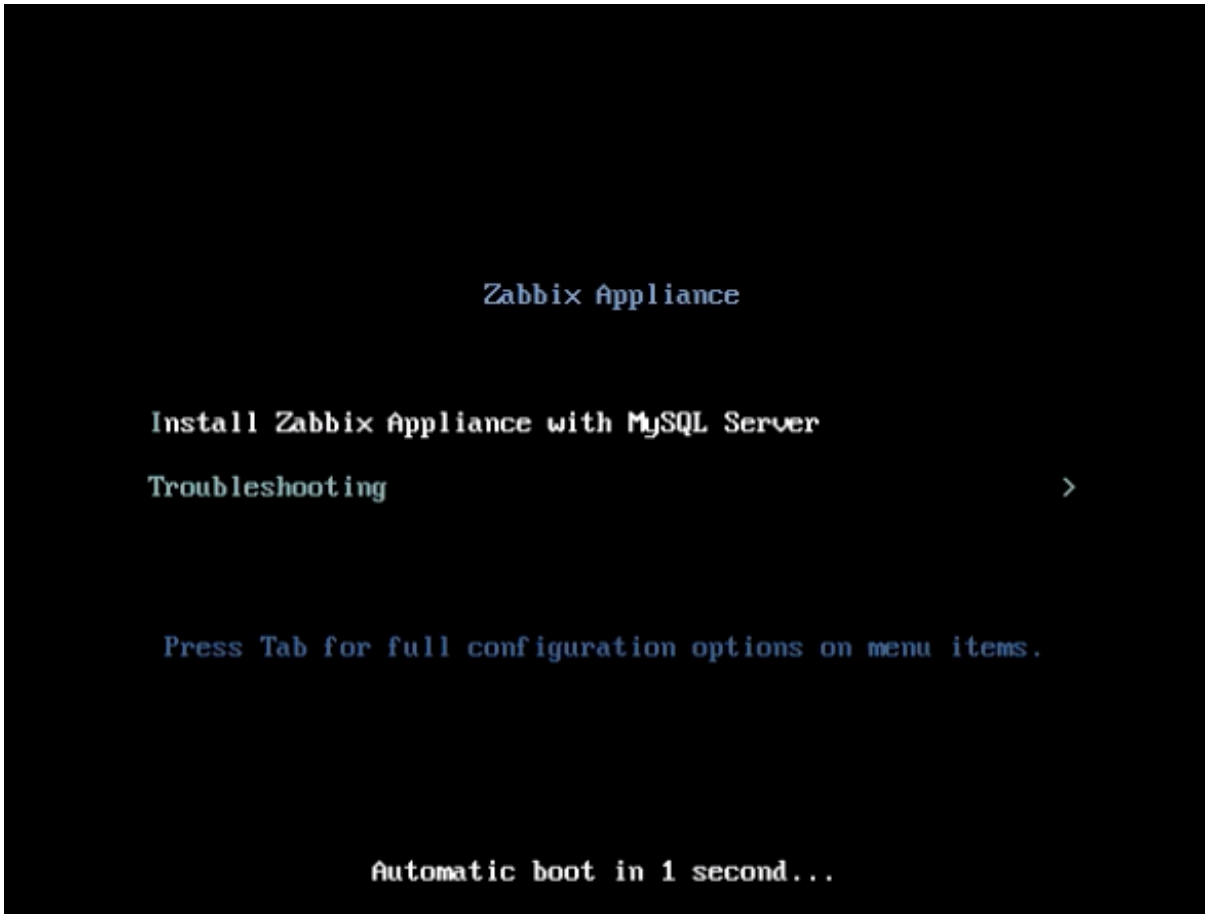
Zabbix 8.0 用のアプライアンスは、AlmaLinux や Rocky Linux などの Red Hat Enterprise Linux 10 派生ディストリビューションをベースとしており、MySQL 上で動作する事前設定済みの Zabbix サーバーと、NGINX Web サーバー上で動作する Web インターフェースが含まれています。

Attention:
このアプライアンスは、Zabbix の評価用に設計されています。本番環境での使用は推奨されません。

アプライアンスイメージは、以下の形式でダウンロードできます。

- インストール CD/DVD（.iso）
- VMware（.vmx）- 注記を参照
- Open Virtualization Format（.ovf）
- Microsoft Hyper-V（.vhd/.vhdx）- 注記を参照
- KVM、Parallels、QEMU、USB メモリ、VirtualBox、Xen（.raw）- 注記を参照
- KVM、QEMU（.qcow2）

Zabbix インストール CD/DVD のブートメニュー:



クイックスタート　前提条件

1. ホストマシンに、仮想マシンのシステム要件を満たすのに十分なリソースがあることを確認してください。

- *RAM*: 4 GB
- ディスク容量: 仮想マシンに最低 8GB
- *CPU*: 最低 2 コア

2. まだインストールされていない場合は、アプライアンスイメージを起動するための仮想化ソフトウェアをインストールします (例: VirtualBox)。
3. 仮想化ソフトウェアでサポートされている形式でアプライアンスをダウンロードします。
4. ネットワーク設定をチェックして、ホストマシンで DHCP が有効になっていることを確認します。

インストール

1. ダウンロードしたイメージからアプライアンスの仮想マシンを起動します。手順については、使用している仮想化ソフトウェアのドキュメントを参照してください。たとえば、VirtualBox documentation をご覧ください。
2. ホストマシン上のブラウザーからアクセスできるように、仮想マシンのネットワーク設定を構成します。これは、ブリッジモードを有効にすることで実現できます。
3. デフォルトのシステムcredentials を使用して仮想マシンにログインします。
4. IP アドレスを取得するには、仮想マシン上で次のコマンドを実行します。

```
ip addr show
```

5. ホストマシン上でブラウザーを開き、アプライアンスが DHCP で取得した IP アドレスにアクセスします。
6. デフォルトのcredentials を使用して Zabbix にログインし、監視を開始します。

設定　このセクションでは、頻繁に必要となるデフォルトの設定と、利用可能なカスタマイズオプションについて説明します。

認証情報

システム

- ユーザー名: `root`
- パスワード: `zabbix`

Zabbix Web インターフェース

- ユーザー名: `Admin`
- パスワード: `zabbix`

ログイン後、ユーザープロファイル設定でデフォルトパスワードを変更するか、新しいユーザーを作成してデフォルトユーザーを削除できます。

データベース

すべてのデータベースユーザーのパスワードは、インストールプロセス中にランダムに生成されます。データベースには、次のユーザーが定義されています。

Root:

- ユーザー名: `root`
- パスワード: `/root/.my.cnf` ファイルに保存されています root アカウントでは、パスワードを入力する必要はありません。

Zabbix サーバー*:*

- ユーザー名: `zabbix_srv`
- パスワード: サーバー設定ファイルに保存されています

Zabbix Web インターフェース*:*

- ユーザー名: `zabbix_web`
- パスワード: Web インターフェース設定ファイルに保存されています

データベースユーザーのパスワードを変更するには、MySQL および対応する設定ファイルで変更してください。

Web インターフェースへのアクセス

Zabbix の Web インターフェースには、*http://*< 仮想マシンの *IP*> からアクセスできます。

デフォルトでは、どこからでもアクセスできます。アクセスを制限するには、`/etc/nginx/conf.d/zabbix.conf` を変更します。
編集したファイルを保存した後、root ユーザーとして SSH 経由でログインし、次のコマンドを実行して NGINX を再起動します。

```
systemctl restart nginx
```

静的 IP アドレス

デフォルトでは、アプライアンスは DHCP を使用して IP アドレスを取得します。
静的 IP アドレスを設定するには、次の手順を実行します。

1. root user としてログインします。
2. カスタムの IP アドレスに値を置き換えて、次のコマンドを実行します。

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # アプライアンスの IP アドレス/CIDR プレフィックス
nmcli con mod eth0 ipv4.gateway 192.168.1.1      # ゲートウェイの IP アドレス
nmcli con mod eth0 ipv4.dns 8.8.8.8    # DNS サーバーの IP アドレス
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

または、ファイル /etc/NetworkManager/system-connections/eth0.nmconnection を開き、次の行を追加します。

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

編集したファイルを保存した後、次のコマンドを実行します。

```
nmcli con reload
nmcli con up eth0
```

ネットワーク設定の変更が適用されたことを確認するには、次のコマンドを実行します。

```
nmcli dev show eth0
```

ファイアウォールの構成

ファイアウォール設定を管理するために、アプライアンスは事前定義されたルールを持つ iptables を使用します。

- SSH (22 TCP) ポートを開放
- Zabbix エージェント (10050 TCP) と Zabbix トラッパー (10051 TCP) ポートを開放
- HTTP (80 TCP) と HTTPS (443 TCP) ポートを開放
- SNMP trap (162 UDP) ポートを開放
- NTP (123 UDP) ポートの発信接続を開放
- ICMP パケットを 1 秒あたり 5 パケットに制限
- 他のすべての着信接続はドロップ

追加のポートを開くには、/etc/sysconfig/iptables ファイルを変更し、ファイアウォールルールをリロードします。

```
systemctl reload iptables
```

リポジトリ

Zabbix アプライアンスは、Zabbix リポジトリの *zabbix-release* パッケージを使用します。リポジトリは/etc/yum.repos.d/* ディレクトリで設定されています。

タイムゾーン

デフォルトではアプライアンスはシステムクロックに UTC を使用します。タイムゾーンを変更するには適切なファイルを/usr/share/zoneinfo から/etc/localtime にコピーします。例:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

Note:
Zabbixフロントエンドのタイムゾーンは個別に設定され、フロントエンドの設定で変更できます。Zabbix フロントエンドのデフォルトのタイムゾーンは Europe/Riga です。

ファイルの場所

- Zabbix サーバーとエージェントの設定ファイルは /etc/zabbix にあります。
- Zabbix サーバーとエージェントのログファイルは /var/log/zabbix にあります。
- Zabbix Web インターフェースの設定ファイルは /etc/zabbix/web にあります。
- Zabbix Web インターフェースは /usr/share/zabbix にあります。
- *zabbix* ユーザーのホームディレクトリは /var/lib/zabbix です。

システムサービス

Systemd サービスが利用可能です。Zabbix サービスのリストを表示するには、仮想マシンで次のコマンドを実行します。

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

イメージフォーマット固有の注意点　VMware

vmdk 形式のイメージは、VMware Player、Server、Workstation 製品で直接使用できます。ESX、ESXi、vSphere で使用する場合は、VMware vCenter Converter（ダウンロードには認証が必要）を使用して変換する必要があります。VMware vCenter Converter を使用すると、ハイブリッドネットワークアダプターで問題が発生する場合があります。その場合は、変換プロセス中に E1000 アダプターを指定してみてください。または、変換の完了後に既存のアダプターを削除し、E1000 アダプターを追加することもできます。

HDD/フラッシュイメージ (raw)

イメージを起動するには、次のコマンドを実行します。

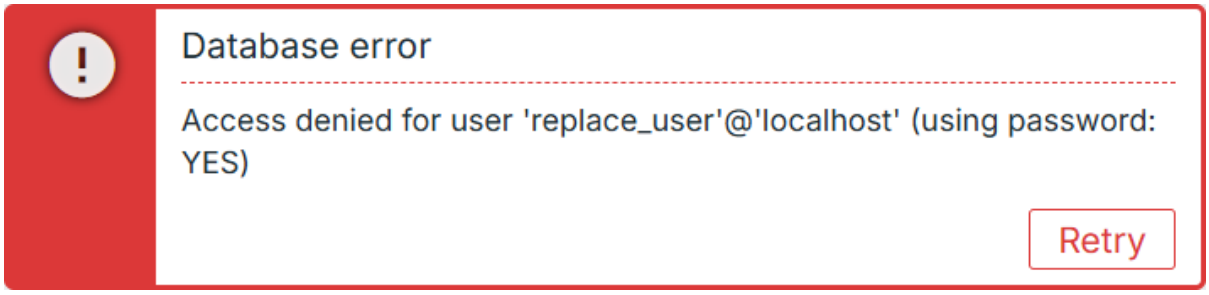
```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

/dev/sdc をフラッシュ/HDD ディスクデバイスのパスに置き換えてください。

Hyper-V

アプライアンスが Hyper-V で起動しない場合は、**Ctrl+Alt+F2** を押して TTY セッションを切り替えてみてください。

トラブルシューティング　Web インターフェースにログインしようとした際に、*Access denied for user 'replace_user'@'localhost' (using password: YES)* というエラーメッセージが表示される場合、インストールがまだ進行中であることを示している可能性があります。



数分待ってもエラーが解消されない場合、またはその他の予期しない動作が見られる場合は、インストール処理が正常に完了していない可能性があります。
この場合は、現在のアプライアンスを削除し、同じインストール手順に従って再デプロイすることを推奨します。
通常、この手順で問題は解決します。

なお、壊れたインストールを手動で修復しようとすることは推奨されません。さらなる問題を引き起こす可能性があるためです。

プロセス

概要　Zabbix は、システムを監視し、通知を送信するために連携して動作する複数のプロセスで構成されています。各プロセスは、監視インフラストラクチャ内で特定の役割を担っています。

コアプロセス

- サーバー - データのポーリングとトラッピングを実行し、トリガーを計算して、ユーザーに通知を送信する中心的なプロセスです。
- エージェント - 監視対象のホストからデータを収集します。
- エージェント 2 - プラグインアーキテクチャと拡張された機能を備えた次世代のエージェントです。
- プロキシ - Zabbix サーバーに代わって監視対象デバイスから監視データを収集し、分散構成で情報を一元管理する Zabbix サーバーに送信します。
- Java ゲートウェイ - Java アプリケーションの監視を可能にします。
- MCP サーバー - AI エージェントが Model Context Protocol を使用して Zabbix と連携できるようにします。

どこからでも、またどのプラットフォームからでも Zabbix に簡単にアクセスできるように、Web ベースのインターフェースが提供されています。このインターフェースは Zabbix サーバーの一部であり、通常は（必須ではありません）サーバーが実行されている物理マシンと同じマシン上で動作します。

すべての設定情報と、Zabbix が収集したデータはデータベースに保存されます。

ヘッダー　ヘッダーは、Zabbix コンポーネント間のすべてのリクエストおよびレスポンスメッセージに含まれます。
これは、メッセージ長、圧縮されているかどうか、大きなパケットかどうかを判定するために必要です。

Zabbix 通信プロトコルでは、1 接続あたりのパケットサイズ上限は 1GB です。この 1GB の上限は、受信したパケットデータ長と非圧縮データ長の両方に適用されます。

Zabbix プロキシに設定を送信する場合、サイズの大きい設定を同期できるように、パケットサイズ上限は 4GB に引き上げられます。圧縮前のデータ長が 4GB を超えると、Zabbix サーバーは自動的に大きなパケット形式（0x04 フラグ）の使用を開始し、パケットサイズ上限は 16GB に拡張されます。

大きなパケット形式は任意のデータ送信に使用できますが、現時点で 1GB を超えるパケットを処理できるのは Zabbix プロキシの設定同期機能のみです。

ヘッダー構造　ヘッダーは 4 つのフィールドで構成されます。ヘッダー内のすべての数値はリトルエンディアン形式で表されます。

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" または 5A 42 58 44
<FLAGS>	1	1	プロトコルフラグ: 0x01 - Zabbix 通信プロトコル 0x02 - 圧縮 0x04 - 大きなパケット
<DATALEN>	4	8	データ長。
<RESERVED>	4	8	圧縮を使用する場合（0x02 フラグ）- 非圧縮データの長さ 圧縮を使用しない場合 - 00 00 00 00

ヘッダーの例　以下は、送信したいデータに Zabbix プロトコルヘッダーを追加して、Zabbix に正しく解釈させるために送信すべきパケットを取得する方法を示すコードスニペットです。これらのコードスニペットでは、データサイズが 1GB を超えないことを前提としているため、大きなパケット形式は使用していません。

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

または

```
def zbx_create_header(plain_data_size, compressed_data_size=None):
    protocol = b"ZBXD"
    flags = 0x01
    if compressed_data_size is None:
        datalen = plain_data_size
        reserved = 0
    else:
        flags |= 0x02
        datalen = compressed_data_size
        reserved = plain_data_size
    return protocol + struct.pack("<BII", flags, datalen, reserved)

packet = zbx_create_header(len(data)) + data
```

Perl

```
my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;
```

または

```
sub zbx_create_header($;$)
{
    my $plain_data_size = shift;
    my $compressed_data_size = shift;

    my $protocol = "ZBXD";
    my $flags = 0x01;
    my $datalen;
    my $reserved;

    if (!defined($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
```

```
    return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);
}

my $packet = zbx_create_header(length($data)) . $data;
```

PHP

```
$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;
```

または

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

サーバー

概要　Zabbix サーバーは、Zabbix ソフトウェアの中心となるプロセスです。

サーバーはデータのポーリングとトラッピングを行い、トリガーを計算し、ユーザーに通知を送信します。
また、Zabbix エージェントとプロキシがシステムの可用性と整合性に関するデータを報告する中心的なコンポーネントでもあります。
サーバー自身も、簡単なサービスチェックを使用して、ネットワーク上のサービス（Web サーバーやメールサーバーなど）をリモートで確認できます。

サーバーは、すべての設定、統計、および運用データが保存される中央リポジトリであり、監視対象のいずれかのシステムで問題が発生した際に、管理者へ積極的に警告を発する Zabbix の主体です。

基本的な Zabbix サーバーの動作は、次の 3 つの明確なコンポーネントに分かれています。Zabbix サーバー、Web インターフェース、およびデータベースストレージです。

Zabbix のすべての設定情報はデータベースに保存され、サーバーと Web インターフェースの両方がこれにアクセスします。
たとえば、Web インターフェース（または API）を使用して新しいアイテムを作成すると、そのアイテムはデータベースの items テーブルに追加されます。
その後、およそ 1 分ごとに Zabbix サーバーは items テーブルを照会して、アクティブなアイテムの一覧を取得し、それを Zabbix サーバー内のキャッシュに保存します。
このため、Zabbix frontend で行った変更が最新データのセクションに反映されるまで、最大で 2 分かかることがあります。

関連情報:

- Zabbix サーバーの高可用性構成
- 設定パラメーター: Zabbix サーバー

サーバーの実行　パッケージとしてインストールされている場合

Zabbix サーバーはデーモンプロセスとして動作します。
サーバーは次のコマンドを実行して起動できます。

```
systemctl start zabbix-server
```

これはほとんどの GNU/Linux システムで動作します。
それ以外のシステムでは、次のコマンドを実行する必要がある場合があります。

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

同様に、停止/再起動/状態の確認には、次のコマンドを使用します。

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

手動で起動する

上記の方法で動作しない場合は、手動で起動する必要があります。
zabbix_server バイナリへのパスを見つけて、次を実行します。

```
zabbix_server
```

Zabbix サーバーでは、次のコマンドラインパラメータを使用できます。

Parameter	Description
-c --config <config-file>	設定ファイルへのパス。
-f --foreground	Zabbix サーバーをフォアグラウンドで実行します。
-R --runtime-control <option>	管理機能を実行します。
-T --test-config	設定ファイルを検証して終了します。
-h --help	ヘルプ情報を表示して終了します。
-V --version	バージョン番号を表示します。

コマンドラインパラメータを指定して Zabbix サーバーを実行する例:

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

ランタイム制御

ランタイム制御オプション:

オプション	説明	対象
config_cache_reload	設定キャッシュを再読み込みします。キャッシュの読み込み中の場合は無視されます。	
history_cache_clear[=<target>]	指定したアイテムの履歴キャッシュをクリアします。 アイテムの最初と最後の値を除く、すべての値に影響します。	**target** - アイテムの ID
diaginfo[=<section>]	診断情報をサーバーのログファイルに収集します。	historycache - 履歴キャッシュの統計情報; valuecache - value cache の統計情報; preprocessing - 前処理マネージャーの統計情報; alerting - アラートマネージャーの統計情報; lld - LLD マネージャーの統計情報; locks - mutex の一覧（*BSD* システムでは空）; connector - キューが最も大きいコネクターの統計情報。
ha_status	高可用性（HA）クラスタのステータスをログに記録します。	
ha_remove_node=<target>	名前または ID で指定した高可用性（HA）ノードを削除します。 アクティブ/スタンバイノードは削除できないことに注意してください。	**target** - ノードの名前または ID（ha_status を実行して取得できます）。

オプション	説明	対象
ha_set_failover_delay=<delay>	高可用性ノードのフェイルオーバー遅延を設定します。 時間のサフィックスがサポートされています（例: 10s、1m）。	
proxy_config_cache_reload[=<target>]	プロキシの設定キャッシュを再読み込みします。	**target** - プロキシ名をカンマ区切りで指定したリスト。 対象を指定しない場合、すべてのプロキシの設定を再読み込みします。
secrets_reload	Vault からシークレットを再読み込みします。	
service_cache_reload	サービスマネージャーのキャッシュを再読み込みします。	
snmp_cache_reload	SNMP キャッシュを再読み込みします。すべてのホストについて、SNMP エンジンのプロパティ（エンジン時間、エンジンブート、エンジン ID、認証情報）をクリアします。SNMP の問題をトラブルシューティングする際に、グローバルキャッシュを強制的にクリアするために使用します。	
housekeeper_execute	ハウスキーピング手順を開始します。 ハウスキーピング手順の実行中の場合は無視されます。	
trigger_housekeeper_execute	トリガーのハウスキーピング手順を開始します。 トリガーのハウスキーピング手順の実行中の場合は無視されます。	
log_level_increase[=<target>]	ログレベルを上げます。対象を指定しない場合、すべてのプロセスに影響します。 *BSD* システムではサポートされません。	**process type** - 指定したタイプのすべてのプロセス（例: poller）。 すべてのサーバープロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号（例: poller,3）。 **pid** - プロセス識別子（1 から 65535）。これより大きい値を指定する場合は、対象を「process type,N」として指定してください。
log_level_decrease[=<target>]	ログレベルを下げます。対象を指定しない場合、すべてのプロセスに影響します。 *BSD* システムではサポートされません。	
prof_enable[=<target>]	プロファイリングを有効にします。 対象を指定しない場合、すべてのプロセスに影響します。 プロファイリングを有効にすると、関数名ごとにすべての rwlock/mutex の詳細が提供されます。	**process type** - 指定したタイプのすべてのプロセス（例: history syncer） プロファイリング対象としてサポートされるプロセスタイプ: alerter、alert manager、availability manager、configuration syncer、discovery manager、escalator、history poller、history syncer、housekeeper、http poller、icmp pinger、ipmi manager、ipmi poller、java poller、lld manager、lld worker、odbc poller、poller、preprocessing manager、proxy poller、self-monitoring、service manager、snmp trapper、task manager、timer、trapper、unreachable poller、vmware collector。 **process type,N** - プロセスタイプと番号（例: history syncer,1）。 **pid** - プロセス識別子（1 から 65535）。これより大きい値を指定する場合は、対象を「process type,N」として指定してください。 **scope** - rwlock、mutex、processing を、プロセスタイプと番号（例: history syncer,1,processing）またはそのタイプのすべてのプロセス（例: history syncer,rwlock）とともに使用できます。

オプション	説明	対象
prof_disable[=<target>]	プロファイリングを無効にします。 対象を指定しない場合、すべてのプロセスに影響します。	**process type** - 指定したタイプのすべてのプロセス（例: `history syncer`）。 プロファイリング対象としてサポートされるプロセスタイプ: `prof_enable` を参照してください。 **process type,N** - プロセスタイプと番号（例: `history syncer,1`）。 **pid** - プロセス識別子（1 から 65535）。これより大きい値を指定する場合は、対象を「process type,N」として指定してください。

実行時制御の対象としてプロセス種別を指定した場合、そのコマンドはその種別の実行中プロセスにのみ適用されます。指定した種別のプロセスが現在実行されていない場合、コマンドはエラーを返します。

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

サーバーの設定キャッシュを再読み込みするために実行時制御を使用する例:

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

プロキシの設定を再読み込みするために実行時制御を使用する例:

```
### すべてのプロキシの設定を再読み込み:
zabbix_server -R proxy_config_cache_reload

### Proxy1 と Proxy2 の設定を再読み込み:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

アイテムの履歴キャッシュをクリアするために実行時制御を使用する例:

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

診断情報を収集するために実行時制御を使用する例:

```
### サーバーのログファイルに、利用可能なすべての診断情報を収集:
zabbix_server -R diaginfo

### サーバーのログファイルに、履歴キャッシュの統計情報を収集:
zabbix_server -R diaginfo=historycache
```

SNMP キャッシュを再読み込みするために実行時制御を使用する例:

```
zabbix_server -R snmp_cache_reload
```

Attention:
SNMPv3 インターフェースが Zabbix UI から更新された場合、ほとんどのケースでは Zabbix がそのインターフェースの新しい SNMPv3 認証情報を自動的に再読み込みします。`-R snmp_cache_reload` は、認証情報の変更後もポーリングが失敗し続ける場合（たとえば、engineBoots/engineID の不整合や RFC 非準拠デバイスが原因の場合）や、トラブルシューティングのためにグローバルな SNMP キャッシュのクリアを強制したい場合にのみ使用してください。

housekeeper の実行をトリガーするために実行時制御を使用する例:

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

ログレベルを変更するために実行時制御を使用する例:

```
### すべてのプロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### 2 番目の poller プロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### PID 1234 のプロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234
```

```
### すべての http poller プロセスのログレベルを下げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

HA フェイルオーバー遅延を最小の 10 秒に設定する例:

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

プロセスユーザー

Zabbix サーバーは、root 以外のユーザーとして実行されるように設計されています。
起動した root 以外のユーザーで実行されます。
そのため、サーバーは任意の root 以外のユーザーで問題なく実行できます。

'root' として実行しようとすると、ハードコードされた'zabbix' ユーザーに切り替わります。このユーザーは、システム上に存在 している必要があります。
サーバーを'root' として実行できるのは、サーバー設定ファイルの'AllowRoot' パラメーターを適切に変更した場合のみです。

Zabbix サーバーとエージェント を同じマシンで実行する場合は、サーバーの実行ユーザーをエージェントとは別のユーザーにすることを推奨します。
同じユーザーで両方を実行すると、エージェントがサーバー設定ファイルにアクセスできてしまい、Zabbix の Admin 権限を持つユーザーであれば、たとえばデータベースパスワードなどを簡単に取得できてしまいます。

設定ファイル

サーバー設定ファイルのデフォルトパスは、オペレーティングシステムと Zabbix のインストール方法によって異なります。

- ソースからインストールした場合: `/usr/local/etc/zabbix_server.conf`、またはコンパイル時の変数 `--sysconfdir` または `--prefix` で設定したパス。
- パッケージおよびアプライアンスからインストールした場合: `/etc/zabbix/zabbix_server.conf`。

`zabbix_server` の設定方法については、設定ファイルのオプションを参照してください。

起動スクリプト

これらのスクリプトは、システムの起動/シャットダウン時に Zabbix プロセスを自動的に開始/停止するために使用されます。スクリプトは misc/init.d ディレクトリにあります。

サーバープロセスの種類とスレッド

- `agent poller` - パッシブチェック用のワーカースレッドを持つ非同期ポーラープロセス
- `alert manager` - アラートキューの管理プロセス
- `alert syncer` - アラートのデータベース書き込みプロセス
- `alerter` - 通知を送信するプロセス
- `availability manager` - ホストの可用性を更新するプロセス
- `browser poller` - ブラウザーアイテムのチェック用ポーラー
- `configuration syncer` - 設定データのメモリ内キャッシュを管理するプロセス
- `configuration syncer worker` - アイテム名内のユーザーマクロ値を解決および同期するプロセス
- `connector manager` - コネクターの管理プロセス
- `connector worker` - コネクターマネージャーからのリクエストを処理するプロセス
- `discovery manager` - デバイス検出の管理プロセス
- `discovery worker` - ディスカバリーマネージャーからの検出タスクを処理するプロセス
- `escalator` - アクションをエスカレーションするプロセス
- `ha manager` - 高可用性を管理するプロセス
- `history poller` - データベース接続を必要とする計算チェックを処理するプロセス
- `history syncer` - 履歴のデータベース書き込みプロセス
- `housekeeper` - 古いデータ（アイテムの履歴とトレンド、ユーザーセッション、イベントなど）および削除されたオブジェクトによって残されたデータを削除するプロセス
- `http agent poller` - HTTP チェック用のワーカースレッドを持つ非同期ポーラープロセス
- `http poller` - Web 監視用ポーラー
- `icmp pinger` - icmpping チェック用ポーラー
- `internal poller` - 内部チェック用ポーラー
- `ipmi manager` - IPMI ポーラーの管理プロセス
- `ipmi poller` - IPMI チェック用ポーラー
- `java poller` - Java チェック用ポーラー
- `lld manager` - ローレベルディスカバリータスクの管理プロセス
- `lld worker` - ローレベルディスカバリータスクのワーカープロセス
- `odbc poller` - ODBC チェック用ポーラー
- `poller` - パッシブチェック用の通常のポーラー

- preprocessing manager - 前処理ワーカースレッドを持つ前処理タスクの管理プロセス
- preprocessing worker - データを前処理するスレッド
- proxy poller - パッシブプロキシ用ポーラー
- proxy group manager - プロキシの負荷分散と高可用性の管理プロセス
- report manager- スケジュールレポート生成タスクの管理プロセス
- report writer - スケジュールレポートを生成するプロセス
- self-monitoring - 内部サーバー統計を収集するプロセス
- service manager - history syncer、task manager、alert manager から障害、障害タグ、障害の復旧に関する情報を受け取り、サービスを管理するプロセス
- snmp poller - SNMP チェック用のワーカースレッドを持つ非同期ポーラープロセス（walk[OID] および get[OID] アイテムのみ）
- snmp trapper - SNMP トラップ用トラッパー
- supervisor - サーバープロセスとスレッドの起動およびシャットダウンを調整し、それらの準備状態を監視するプロセス
- task manager - 他のコンポーネントから要求されたタスクをリモート実行するプロセス（例: 障害のクローズ、障害の認識、アイテムの値の即時チェック、リモートコマンド機能）
- telemetry query poller - テレメトリークエリアイテムのチェック用ポーラー
- timer - メンテナンスを処理するタイマー
- trapper - アクティブチェック、トラップ、プロキシ通信に使用するトラッパー
- trigger housekeeper - 削除されたトリガーによって生成された障害とイベントを削除するプロセス
- unreachable poller - 到達不能なデバイス用ポーラー
- vmware collector - VMware サービスからデータを収集する VMware データコレクター

これらのプロセスの種類は、サーバーログファイルで確認できます。

サーバーログファイルは、ファイル所有者だけが読み書きできる権限で作成されます。また、所有者グループはファイルを読み取ることができます。その他のすべての権限は拒否されます。

さまざまな種類の Zabbix サーバープロセスは、zabbix[process,<type>,<mode>,<state>] 内部アイテムを使用して監視できます。

History syncer トランザクション統計

history syncer プロセスのタイトルには、history syncer トランザクションに関する詳細な統計が表示されます。

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

「A+B triggers」では、次の意味になります。

- A - history values に基づいて処理されたトリガー
- B - タイマーに基づいて処理されたトリガー

processed...in N（<timings>）sec の各時間は、次のとおりです。

- アイテム値をデータベースに書き込むのにかかった時間
- アイテムデータ（状態、エラー、ホストインベントリなど）を更新するのにかかった時間
- トレンドをデータベースにフラッシュするのにかかった時間
- トリガーを計算するのにかかった時間
- イベントとアクションを処理するのにかかった時間

Housekeeping procedure

housekeeper プロセスは、定期的に古いデータ（アイテム履歴やトレンド、ユーザーセッション、イベントなど）および削除されたオブジェクトに残されたデータを削除します。この処理はサイクル単位で実行され、実行頻度と 1 サイクルあたりの削除上限はHousekeepingFrequency とMaxHousekeeperDelete によって決まります。1 サイクルで削除しきれなかったデータは、次のサイクルに持ち越されます。自動 housekeeping は *Administration* > *Housekeeping* で有効化および設定できます。

housekeeper プロセスが削除されたオブジェクトに残されたデータを削除する際は、オブジェクトが削除されるたびに作成される housekeeper テーブルのタスクを参照します。たとえば、ホストを削除すると、Zabbix はそのアイテムを削除しますが、履歴、トレンド、障害は削除しません。代わりに、データベーストリガーが housekeeper テーブルに次のフィールドからなるタスクを作成します。

- housekeeperid - タスク ID
- object - オブジェクト種別 (0 - アイテム; 1 - トリガー; 2 - サービス; 3 - 発見されたホスト; 4 - 発見されたサービス)
- objectid - オブジェクト ID（housekeeper がオブジェクト関連データを見つけるのに役立ちます）

たとえば、2 つのアイテムと 1 つのトリガーを持つホストを削除すると、housekeeper テーブルは次のようになります。

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
```

```
+---------------+--------+----------+
|             1 |      1 |    28724 |
|             2 |      0 |    59396 |
|             3 |      0 |    59397 |
+---------------+--------+----------+
```

データベーストリガーは、オブジェクト関連データの有無を確認せずに housekeeper テーブルへタスクを作成します。その確認は、housekeeper プロセスがデータを削除する際に行います。

- 削除されたアイテム（LLD ルールを含む）については、すべての履歴およびトレンドテーブル（history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint）からそのデータを削除します。また、problem テーブルから内部イベントも削除します。
- 削除されたトリガーについては、イベント関連テーブル（problem, event_symptom, events; データベース自体も event_recovery テーブルからイベントを削除します）からそのイベントを削除します。また、削除されたイベントについて service manager プロセスに通知します。

Note:
別個の trigger housekeeper プロセスは、より限定されたタスク、つまり、既知のソーストリガーを持たない障害とイベントの削除を担当します。実行頻度はProblemHousekeepingFrequency によって制御されます。
トリガー housekeeping procedure が開始されるまで、すでに削除されたトリガーによって作成された障害が、引き続きサービス障害を生成し、サービスに割り当てられる場合があります。構成に、頻繁に発見/未発見となるトリガーに基づく多数のサービスstatus calculation rules が含まれる場合は、ProblemHousekeepingFrequency サーバー設定パラメータを調整して、housekeeping procedure の頻度を上げることを検討してください。

- 削除されたサービスについては、problem テーブルからその障害を削除します。events テーブルからのイベント削除は、設定されたdata storage period の後にのみ行われます。
- 削除された発見済みホスト/サービス（network discovery）については、events テーブルからそのイベントを削除します。

housekeeper プロセスは、障害に関連付けられていないイベントのみを削除します。たとえば、古い障害/復旧イベントでも、オープンな障害に関連付けられている場合は削除されません。housekeeper が古いエンティティを削除する際は、最初に障害を削除し、その後にイベントを削除します。

Note:
partition モード（TimescaleDB のパーティション分割テーブル）を使用するテーブルはスキップされ、regular モードを使用するテーブルのみが処理されます。

サポート対象プラットフォーム　セキュリティ要件とサーバー運用のミッションクリティカルな性質により、必要な性能、フォールトトレランス、およびレジリエンスを一貫して提供できるのは UNIX のみです。
Zabbix は、市場をリードする各バージョンで動作します。

Zabbix サーバーは、以下のプラットフォームでテストされています。

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

Note:
Zabbix は、他の Unix 系オペレーティングシステムでも動作する場合があります。

ロケール　サーバーでは、一部のテキストアイテムを正しく解釈できるように、UTF-8 ロケールが必要です。多くの最新の Unix 系システムでは UTF-8 ロケールがデフォルトになっていますが、システムによっては明示的に設定する必要があります。

Zabbix サーバーの高可用性構成

概要

高可用性（HA）は、ほぼダウンタイムを許容できない重要なインフラストラクチャで通常必要とされます。そのため、障害が発生する可能性のあるサービスには、現在のサービスが停止した場合に引き継ぐフェイルオーバーオプションを用意しておく必要があります。

Zabbix は、設定が簡単で、事前の HA に関する知識を必要としない ネイティブの高可用性ソリューションを提供します。Zabbix のネイティブ HA は、Zabbix サーバーのソフトウェア/ハードウェア障害に対する追加の保護層として、またはメンテナンスによるダウンタイムを減らすために役立ちます。

Zabbix の高可用性モードでは、複数の Zabbix サーバーがクラスタ内のノードとして実行されます。クラスタ内の 1 台の Zabbix サーバーがアクティブである一方、他のサーバーは待機状態となり、必要に応じて引き継げるよう準備されています。



Zabbix HA への切り替えは非拘束的です。いつでもスタンドアロン運用に戻すことができます。

関連情報: 実装の詳細

高可用性の有効化

Zabbix サーバーをクラスターノードとして起動する

Zabbix サーバーをクラスターノードとして起動するには、サーバー設定で 2 つのパラメータが必要です。

- HA クラスターノードとなる各 Zabbix サーバーに、HANodeName パラメータを指定する必要があります。

これは一意のノード識別子（例: zabbix-node-01）で、エージェントおよびプロキシの設定でサーバーを識別するために使用されます。HANodeName を指定しない場合、サーバーはスタンドアロンモードで起動します。

- 各ノードに NodeAddress パラメータを指定する必要があります。

NodeAddress パラメータ（address:port）は、Zabbix の Web インターフェースがアクティブなサーバーノードに接続するために使用されます。NodeAddress は、該当する Zabbix サーバーの IP または FQDN 名と一致している必要があります。

設定ファイルを変更した後、すべての Zabbix サーバーを再起動します。これで、サーバーはクラスターノードとして起動します。サーバーの新しいステータスは、レポート > システム情報で確認できます。また、次のコマンドを実行して確認することもできます。

```
zabbix_server -R ha_status
```

このランタイムコマンドを実行すると、現在の HA クラスターのステータスが Zabbix サーバーのログ（および標準出力）に記録されます。

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

Web インターフェースの準備

Zabbix Web インターフェース設定ファイル（zabbix.conf.php）で、Zabbix サーバーの address:port が定義されていないことを確認してください。

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

Zabbix Web インターフェースは、Zabbix データベースの nodes テーブルから設定を読み取り、アクティブなノードを自動検出します。アクティブなノードのノードアドレスが、Zabbix サーバーのアドレスとして使用されます。

プロキシ設定

HA クラスタノード（サーバー）は、パッシブまたはアクティブ Zabbix プロキシのいずれかの設定に一覧表示する必要があります。

パッシブプロキシの場合、ノード名はプロキシの Server パラメータ に カンマ区切りで記載する必要があります。

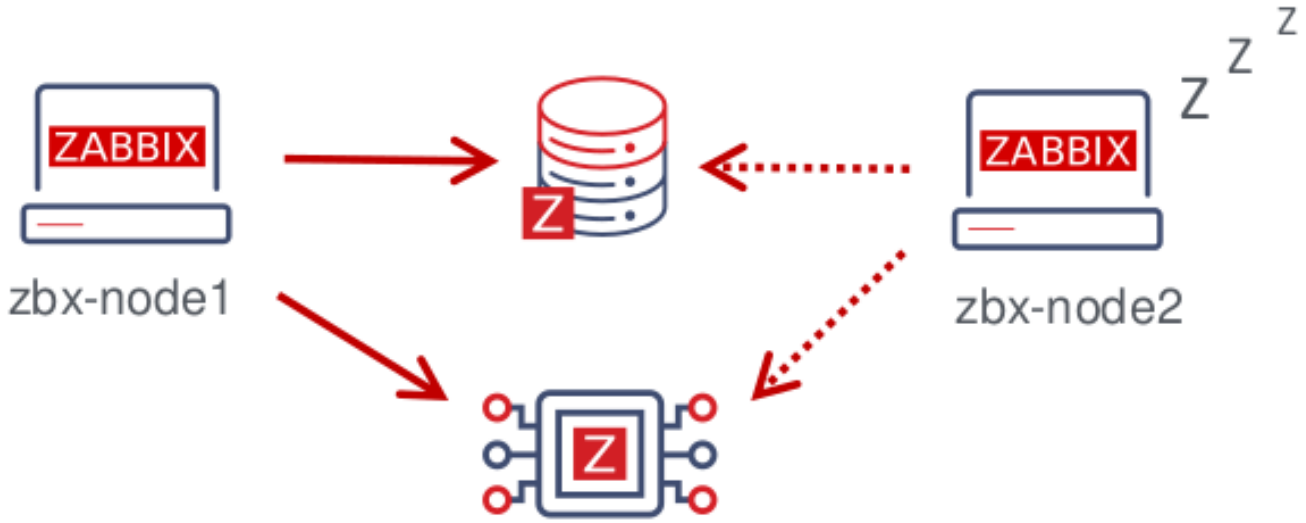
```
Server=zabbix-node-01,zabbix-node-02
```

アクティブプロキシの場合、ノード名はプロキシの Server パラメータ に セミコロン区切りで記載する必要があります。

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

エージェント設定

HA クラスタノード（サーバー）は、Zabbix エージェントまたは Zabbix エージェント 2 の設定に記載する必要があります。



パッシブチェックを有効にするには、ノード名を Server parameter に カンマ区切りで記載する必要があります。

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

アクティブチェックを有効にするには、ノード名を ServerActive parameter に記載します。アクティブチェックでは、ノードは他のサーバーとはカンマで区切り、ノード同士は セミコロンで区切る必要があることに注意してください。例:

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

スタンバイノードへのフェイルオーバー

アクティブノードが停止すると、Zabbix は自動的に別のノードへフェイルオーバーします。
フェイルオーバーを発生させるには、少なくとも 1 つのノードがスタンバイ状態である必要があります。

フェイルオーバーはどのくらい速く行われますか?
すべてのノードは、最後のアクセス時刻（および変更があった場合はステータス）を 5 秒ごとに更新します。
そのため、次のようになります。

- アクティブノードが停止し、その状態を”stopped” として報告できた場合、別のノードが **5** 秒以内に引き継ぎます。
- アクティブノードが停止した、または利用不可になったものの、ステータスを更新できなかった場合、スタンバイノードは引き継ぎまで フェイルオーバー遅延 + 5 秒待機します。

フェイルオーバー遅延は設定可能で、対応範囲は 10 秒から 15 分です（デフォルトは 1 分）。
フェイルオーバー遅延を変更するには、次を実行します。

```
zabbix_server -R ha_set_failover_delay=5m
```

HA クラスターの管理

HA クラスターの現在のステータスは、専用のランタイムコントロールオプションを使用して管理できます。

- `ha_status` - Zabbix サーバーのログ（および標準出力）に HA クラスターのステータスを記録します。
- `ha_remove_node=target` - `<target>` で識別される HA ノードを削除します。`<target>` にはノードの名前または ID を指定します（名前/ID は `ha_status` の実行結果から取得できます）。例：

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

アクティブ/スタンバイノードは削除できないことに注意してください。

- `ha_set_failover_delay=delay` - HA フェイルオーバーの遅延を設定します（10 秒から 15 分の間。時間のサフィックスに対応しています（例：30s、1m））。

ノードのステータスは、次の方法で監視できます。

- レポート > システム情報
- システム情報ダッシュボードウィジェット
- サーバーの `ha_status` ランタイムコントロールオプション（上記を参照）

内部アイテム `zabbix[cluster,discovery,nodes]` は、高可用性ノードの情報を含む JSON を返すため、ノードのディスカバリに使用できます。

HA クラスタの無効化

高可用性クラスタを無効にするには、次の手順を実行します。

1. 設定ファイルのバックアップを作成します。
2. スタンバイノードを停止します。
3. アクティブなプライマリーサーバーから HANodeName パラメーターを削除します。
4. プライマリーサーバーを再起動します（スタンドアロンモードで起動します）。

HA クラスタのアップグレード

HA ノードのメジャーバージョンアップグレードを実行するには、次の手順に従います。

1. すべてのノードを停止します。
2. データベースの完全バックアップを作成します。
3. データベースでレプリケーションを使用している場合は、すべてのノードが同期されており、問題がないことを確認します。レプリケーションが壊れている場合はアップグレードしないでください。
4. データベースのアップグレードを実行する単一のノードを選択し、`HANodeName` をコメントアウトして設定をスタンドアロンモードに変更し、そのノードをアップグレードします。
5. データベースのアップグレードが完全に完了していることを確認します（システム情報に Zabbix サーバーが稼働中であることが表示されている必要があります）。
6. ノードを HA モードで再起動します。
7. 残りのノードをアップグレードして起動します（この時点ではデータベースはすでにアップグレードされているため、スタンドアロンモードに変更する必要はありません）。

マイナーバージョンアップグレードでは、最初のノードをアップグレードし、それがアップグレード済みで稼働していることを確認してから、次のノードのアップグレードを開始すれば十分です。

実装の詳細

高可用性（HA）クラスタはオプトイン方式のソリューションであり、Zabbix サーバーでサポートされています。
ネイティブの HA ソリューションは、使いやすさを重視して設計されており、サイトをまたいで動作し、Zabbix が認識するデータベースに特別な要件はありません。
ユーザーは、環境の高可用性要件に最も適したものに応じて、ネイティブの Zabbix HA ソリューションまたはサードパーティの HA ソリューションを自由に使用できます。

このソリューションは、複数の `zabbix_server` インスタンスまたはノードで構成されます。
各ノードは次のとおりです。

- 個別に設定されます。
- 同じデータベースを使用します。
- active、standby、unavailable、stopped の複数のモードを持つ場合があります。

同時にアクティブ（稼働中）になれるノードは 1 つだけです。standby ノードは 1 つのプロセス、つまり HA マネージャーのみを実行します。
standby ノードは、データ収集、処理、その他の通常のサーバー活動を行いません。ポートをリッスンせず、データベース接続数も最小限です。

active ノードと standby ノードの両方が、5 秒ごとに最終アクセス時刻を更新します。
各 standby ノードは、active ノードの最終アクセス時刻を監視します。
active ノードの最終アクセス時刻が'failover delay' 秒を超えると、standby ノードは自分自身を active ノードに切り替え、以前 active だったノードに `unavailable` ステータスを割り当てます。

active ノードは、自身のデータベース接続を監視します。接続が `failover delay`-5 秒を超えて失われた場合、すべての処理を停止し、standby モードに切り替える必要があります。
active ノードは standby ノードのステータスも監視します。standby ノードの最終アクセス時刻が'failover delay' 秒を超えると、その standby ノードには `unavailable` ステータスが割り当てられます。

これらのノードは、Zabbix のマイナーバージョン間で互換性を持つように設計されています。

設定パラメーター: **Zabbix** サーバー

概要

設定パラメータを使用すると、Zabbix サーバーの動作をカスタマイズできます。

パラメータ値は、Zabbix のサーバー設定ファイル（zabbix_server.conf）で指定します。パラメータが必須であることが明示されていない限り、すべてのパラメータは任意です。

このページは次の構成になっています。

- パラメータの概要（パラメータの詳細を表示するには、その名前をクリックします）
- パラメータの詳細

パラメータ	説明
AlertScriptsPath	アラートスクリプトの場所を指定します。
AllowRoot	root としてサーバーを実行できるようにします。
AllowSoftwareUpdateCheck	zabbix.com からソフトウェア更新に関する情報を受信できるようにします。
AllowUnsupportedDBVersions	サポート対象外のデータベースバージョンでサーバーを動作できるようにします。
BridgeAdapterURL	Zabbix ブリッジアダプターの HTTP(S) URL を設定します。
CacheSize	設定キャッシュのサイズ（バイト単位）を設定します。
CacheUpdateFrequency	設定キャッシュの更新頻度（秒単位）を設定します。
DBHost	データベースのホスト（またはソケットディレクトリ）を指定します。
DBName	データベース名を指定します。必須です。
DBPassword	データベースのパスワードを指定します。
DBPort	データベースホストのポートを指定します。
DBSchema	データベーススキーマ名を指定します。PostgreSQL で使用します。
DBSocket	MySQL ソケットファイルへのパスを指定します。
DBUser	データベースユーザーを指定します。
DBTLSConnect	データベースへの TLS 接続を強制する値を指定します。
DBTLSCAFile	データベース証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。
DBTLSCertFile	データベースへの認証に使用する Zabbix サーバー証明書を含むファイルの完全なパス名を指定します。
DBTLSKeyFile	データベースへの認証に使用する秘密鍵を含むファイルの完全なパス名を指定します。
DBTLSCipher	TLS v1.2 までの TLS プロトコルで Zabbix サーバーが許可する暗号化暗号のリストを指定します。MySQL でのみサポートされます。
DBTLSCipher13	TLS v1.3 プロトコルで Zabbix サーバーが許可する暗号スイートのリストを指定します。MySQL 8.0.16 以降でのみサポートされます。
DebugLevel	デバッグレベルを設定します。
EnableGlobalScripts	Zabbix サーバーでグローバルスクリプトを有効にします。
EnableMobileDevices	Zabbix サーバーでモバイルデバイスのサポートを有効にします。
ExportDir	イベント、ヒストリ、トレンドのリアルタイムエクスポートを有効にするディレクトリを指定します。
ExportFileSize	エクスポートファイルの最大サイズ（バイト単位）を設定します。
ExportType	リアルタイムエクスポートのエンティティタイプ（イベント、ヒストリ、トレンド）を指定します。
ExternalScripts	外部スクリプトの場所を指定します。
Fping6Location	fping6 の場所を指定します。
FpingLocation	fping の場所を指定します。
FrontendAllowedIP	Web インターフェースからの接続を許可する IP アドレスまたは CIDR 範囲を指定します。
HANodeName	高可用性クラスター内のノード名を指定します。
HistoryCacheSize	ヒストリキャッシュのサイズを設定します。
HistoryIndexCacheSize	ヒストリインデックスキャッシュのサイズを設定します。
HistoryProvider	ヒストリの保存先プロバイダー（Elasticsearch、ClickHouse、または SQL）を設定します。
HistoryStorageDateIndex	このパラメータは非推奨です（代わりに HistoryProvider を使用してください）。 日付に基づいて異なるインデックスにヒストリ値を保存できるようにします。
HistoryStorageURL	このパラメータは非推奨です（代わりに HistoryProvider を使用してください）。 ヒストリの保存先 URL を指定します。
HistoryStorageTypes	このパラメータは非推奨です（代わりに HistoryProvider を使用してください）。 ヒストリの保存先に送信する値のタイプを指定します。
HousekeepingFrequency	ハウスキーピング処理の頻度（時間単位）を設定します。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
JavaGateway	Zabbix Java ゲートウェイの IP アドレス（またはホスト名）を指定します。
JavaGatewayPort	Zabbix Java ゲートウェイのポートを指定します。

パラメータ	説明
ListenBacklog	TCP キューで保留できる接続数の最大値を設定します。
ListenIP	トラッパーが待ち受ける IP アドレスを指定します。
ListenPort	トラッパーが待ち受けるポートを指定します。
LoadModule	サーバー起動時にロードするモジュールを指定します。
LoadModulePath	サーバーモジュールの場所への完全なパスを指定します。
LogFile	サーバーログファイルを指定します。
LogFileSize	ログファイルの最大サイズを設定します。
LogSlowQueries	ログに記録するまでにデータベースクエリに許可する最大実行時間（ミリ秒単位）を設定します。
LogType	ログ出力タイプを指定します。
MaxConcurrentChecksPerPoller	各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが一度に実行できる非同期チェック数の最大値を設定します。
MaxHousekeeperDelete	1 回のハウスキーピングサイクルで、タスクごとに削除できる行数の最大値を設定します。
NodeAddress	Web インターフェースからサーバーへの接続方法を上書きする IP またはホスト名を、任意のポートとともに指定します。
PidFile	PID ファイルを指定します。
ProblemHousekeepingFrequency	削除されたトリガーの障害を削除する頻度を設定します。
ProxyConfigFrequency	サーバーが Zabbix プロキシに設定データを送信する頻度を設定します。
ProxyDataFrequency	サーバーが Zabbix プロキシからヒストリデータを要求する頻度を設定します。
ServiceManagerSyncFrequency	サービスマネージャーの設定を同期する頻度を設定します。
SMSDevices	Zabbix サーバーでの使用を許可するモデムファイルを指定します。
SNMPTrapperFile	SNMP トラップデーモンからサーバーにデータを渡すために使用する一時ファイルを指定します。
SocketDir	内部 Zabbix サービスが使用する IPC ソケットを保存するディレクトリを指定します。
SourceIP	送信元 IP アドレスを指定します。
SSHKeyLocation	SSH チェックおよびアクション用の公開鍵と秘密鍵の場所を指定します。
SSLCertLocation	クライアント認証用の SSL クライアント証明書ファイルの場所を指定します。
SSLKeyLocation	クライアント認証用の SSL 秘密鍵ファイルの場所を指定します。
SSLCALocation	SSL サーバー証明書の検証に使用する認証局（CA）ファイルの場所を指定します。
StartAgentPollers	非同期 Zabbix エージェントポーラーの fork 済みインスタンス数を設定します。
StartAlerters	アラート送信プロセスの fork 済みインスタンス数を設定します。
StartBrowserPollers	ブラウザーアイテムポーラーの fork 済みインスタンス数を設定します。
StartConnectors	コネクタワーカーの fork 済みインスタンス数を設定します。
StartDBSyncers	ヒストリ同期プロセスの fork 済みインスタンス数を設定します。
StartDiscoverers	ディスカバリーワーカーの fork 済みインスタンス数を設定します。
StartEscalators	エスカレーターの fork 済みインスタンス数を設定します。
StartHistoryPollers	ヒストリポーラーの fork 済みインスタンス数を設定します。
StartHTTPAgentPollers	非同期 HTTP エージェントポーラーの fork 済みインスタンス数を設定します。
StartHTTPPollers	HTTP ポーラーの fork 済みインスタンス数を設定します。
StartIPMIPollers	IPMI ポーラーの fork 済みインスタンス数を設定します。
StartJavaPollers	Java ポーラーの fork 済みインスタンス数を設定します。
StartLLDProcessors	ローレベルディスカバリ（LLD）ワーカーの fork 済みインスタンス数を設定します。
StartODBCPollers	ODBC ポーラーの fork 済みインスタンス数を設定します。
StartPingers	ICMP ping プロセスの fork 済みインスタンス数を設定します。
StartPollersUnreachable	到達不能なホスト（IPMI および Java を含む）用ポーラーの fork 済みインスタンス数を設定します。
StartPollers	ポーラーの fork 済みインスタンス数を設定します。
StartPreprocessors	前処理ワーカーの事前起動インスタンス数を設定します。
StartProxyPollers	パッシブプロキシ用ポーラーの fork 済みインスタンス数を設定します。
StartReportWriters	レポートライターの fork 済みインスタンス数を設定します。
StartSNMPPollers	非同期 SNMP ポーラーの fork 済みインスタンス数を設定します。
StartSNMPTrapper	SNMP トラッパープロセスを起動するには「1」に設定します。
StartTelemetryQueryPollers	非同期 Telemetry クエリポーラーの fork 済みインスタンス数を設定します。
StartTimers	タイマーの fork 済みインスタンス数を設定します。
StartTrappers	トラッパーの fork 済みインスタンス数を設定します。
StartVMwareCollectors	VMware コレクターインスタンスの fork 済みインスタンス数を設定します。
StatsAllowedIP	統計情報の要求を許可する外部 Zabbix インスタンスの IP アドレスまたは DNS 名を指定します。
TelemetryProvider	テレメトリデータソースを設定します。
Timeout	Zabbix プロキシ、エージェント、Web サービスとの接続確立およびデータ交換、ならびに SNMP チェック（SNMP `walk[OID]` および `get[OID]` アイテムを除く）のために待機する時間（秒単位）を指定します。
TLSCAFile	Zabbix コンポーネント間の暗号化通信で使用する、ピア証明書の検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
TLSCertFile	Zabbix コンポーネント間の暗号化通信で使用する、サーバー証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

パラメータ	説明
TLSCipherAll	GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 で OpenSSL 1.1.1 以降に使用する暗号文字列を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 で OpenSSL 1.1.1 以降に使用する暗号文字列を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 で OpenSSL 1.1.1 以降に使用する暗号文字列を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCRLFile	失効した証明書を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSFrontendAccept	Web インターフェースからの受信接続に使用する暗号化レベルを指定します。
TLSFrontendCertIssuer	許可する Web インターフェース証明書の発行者を指定します。
TLSFrontendCertSubject	許可する Web インターフェース証明書のサブジェクトを指定します。
TLSKeyFile	Zabbix コンポーネント間の暗号化通信で使用する、サーバー秘密鍵を含むファイルの完全なパス名を指定します。
TLSListen	トラッパーソケットで TLS 接続のみを要求します。
TmpDir	一時ディレクトリを指定します。
TrapperTimeout	次の処理におけるトラッパーのタイムアウトを秒単位で指定します。 - Zabbix プロキシからのヒストリデータの取得 - Zabbix プロキシへの設定データの送信 - Zabbix サーバーでのグローバルスクリプトの実行またはリモートコマンドの実行
TrendCacheSize	トレンドキャッシュのサイズを設定します。
TrendFunctionCacheSize	トレンド関数キャッシュのサイズを設定します。
UnavailableDelay	利用不可期間中にホストの可用性を確認する頻度を設定します。
UnreachableDelay	到達不能期間中にホストの可用性を確認する頻度を設定します。
UnreachablePeriod	到達不能になってから何秒後にホストを利用不可として扱うかを設定します。
User	システム上に存在する特定のユーザーに権限を降格します。
ValueCacheSize	ヒストリ値キャッシュのサイズを設定します。
Vault	Vault プロバイダーを指定します。
VaultAppRoleID	HashiCorp Vault の AppRole ID を指定します。
VaultAppSecretID	HashiCorp Vault の AppRole シークレット ID を指定します。
VaultDBPath	キーによってデータベース認証情報を取得する場所を指定します。
VaultPrefix	Vault パスまたはクエリのカスタムプレフィックスを指定します。
VaultTLSCertFile	クライアント認証に使用する SSL 証明書ファイルの名前を指定します。
VaultTLSKeyFile	クライアント認証に使用する SSL 秘密鍵ファイルの名前を指定します。
VaultToken	HashiCorp Vault の認証トークンを指定します。
VaultURL	Vault サーバーの URL を指定します。
VMwareCacheSize	VMware データの保存に使用する共有メモリのサイズを設定します。
VMwareFrequency	単一の VMware サービスからデータを収集する頻度を秒単位で設定します。
VMwarePerfFrequency	単一の VMware サービスからパフォーマンスカウンター統計を取得する頻度を秒単位で設定します。
VMwareTimeout	`vmware collector` が VMware サービスからの応答を待機する最大時間を秒単位で設定します。
WebDriverURL	WebDriver インターフェースの URL を指定します。
WebServiceURL	`http[s]://host:port/report` 形式で Web サービスの URL を指定します。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、デーモンのデフォルト値を反映しています。
- 値では環境変数を使用できます。
- Zabbix は、BOMなしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみサポートされます。

Warning:
プロセス数が多すぎると、前処理マネージャーのプロセスごとのファイルディスクリプター制限を使い果たす可能性があります。

このようなプロセスには、ポーラー、到達不能ポーラー、ODBC ポーラー、HTTP ポーラー、Java ポーラー、ping プロセス、トラッパ

ー、プロキシポーラー、IPMI マネージャー、SNMP トラッパー、前処理ワーカー、ディスカバリーワーカーによるデータ収集が含まれます。ファイルディスクリプター制限を使い果たすと、Zabbix サーバーは停止します。通常は起動直後に発生しますが、時間がかかる場合もあります。

このような問題を避けるには、Zabbix サーバー設定ファイルを確認し、同時実行するチェック数とプロセス数を最適化してください。また、必要に応じて、システムの制限を確認および調整し、ファイルディスクリプター制限が十分に高く設定されていることを確認してください。

パラメーターの詳細

AlertScriptsPath

アラートスクリプトの場所を指定します。datadir のコンパイル時インストール変数に依存します。

デフォルト: /usr/local/share/zabbix/alertscripts

AllowRoot

サーバーが root として実行されることを許可します。無効にされていて、サーバーが root によって起動された場合、サーバーは代わりに zabbix ユーザーへ切り替えようとします。通常ユーザーで起動された場合は影響しません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

AllowSoftwareUpdateCheck

Zabbix UI が zabbix.com からソフトウェア更新に関する情報を受信することを許可します。

デフォルト: 1
 値: 0 - 許可しない; 1 - 許可する

AllowUnsupportedDBVersions

サーバーがサポートされていないデータベースバージョンで動作することを許可します。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

BridgeAdapterURL

Zabbix ブリッジアダプターの HTTP(S) URL を設定します。

例:

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

設定キャッシュサイズを指定します（バイト単位）。ホスト、アイテム、トリガーのデータを格納するための共有メモリサイズです。

デフォルト: 32M
 範囲: 128K-64G

CacheUpdateFrequency

設定キャッシュの更新頻度を秒単位で設定します。runtime control オプションも参照してください。

デフォルト: 10
 範囲: 1-3600

DBHost

データベースのホスト（またはソケットディレクトリ）を指定します。

MySQL の場合:

- localhost または空文字列は、デフォルトの UNIX ドメインソケットを使用します。

PostgreSQL の場合:

- localhost は DNS 経由で解決されます（通常は 127.0.0.1）。
- 空文字列は、デフォルトの UNIX ドメインソケットを使用します。
- パス（たとえば /var/run/pgbouncer）を指定すると、そのパスにある UNIX ソケットを使用します。
- カンマ区切りのリストには、複数の host:port 値を含めることができます。各ホストは、読み書き可能な接続が確立されるまで順番に試行されます。

例: DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]

デフォルト: localhost

DBName

データベース名を指定します。

必須: はい

DBPassword

データベースのパスワードを指定します。パスワードを使用しない場合は、この行をコメントアウトしてください。

DBPort

Zabbix データベースホストのポートを指定します（DBHost を参照）。
 DBSocket と DBPort は、サーバー設定において排他的です。どちらか一方のみを指定するか、両方とも未定義のままにしてください。

MySQL のデフォルト: 3306

PostgreSQL のデフォルト: 5432

範囲: 1024-65535

DBSchema

データベーススキーマ名を指定します。PostgreSQL で使用されます。

DBSocket

MySQL ソケットファイルへのパスを指定します。
 DBSocket と DBPort はサーバー設定において排他的です。どちらか一方のみを指定するか、両方とも未定義のままにしてください。

DBUser

データベースユーザーを指定します。

DBTLSConnect

データベースへの TLS 接続を強制する値を指定します。- *required* - TLS を使用して接続します。- *verify_ca* - TLS を使用して接続し、証明書を検証します。- *verify_full* - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース識別子がその証明書と一致することを検証します。

MySQL では 5.7.11 以降、PostgreSQL では次の値がサポートされています: `required`, `verify_ca`, `verify_full`。
 MariaDB では 10.2.6 以降、`required` と `verify_full` の値がサポートされています。
 デフォルトではどのオプションにも設定されず、動作はデータベースの設定に依存します。

DBTLSCAFile

データベース証明書の検証に使用する最上位 CA 証明書を含むファイルの完全なパス名を指定します。

必須: いいえ（DBTLSConnect が `verify_ca` または `verify_full` に設定されている場合は必須）

DBTLSCertFile

データベースへの認証に使用する Zabbix サーバー証明書を含むファイルの完全なパス名を指定します。

DBTLSKeyFile

データベースへの認証に使用する秘密鍵を含むファイルの完全なパス名を指定します。

DBTLSCipher

Zabbix サーバーが TLS v1.2 までの TLS プロトコルで許可する暗号化方式の一覧を指定します。MySQL でのみサポートされます。

DBTLSCipher13

Zabbix サーバーが TLS v1.3 プロトコルで許可する暗号スイートの一覧を指定します。MySQL では、バージョン 8.0.16 以降でのみサポートされます。

DebugLevel

デバッグレベルを設定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。
 ランタイム制御 オプションも参照してください。

デフォルト: 3
 範囲: 0-5

EnableGlobalScripts

Zabbix サーバーでグローバルスクリプトを有効にします。
 注: グローバルスクリプトの実行はデフォルトで有効です。ただし、Zabbix 7.0 以降の新規インストールでは、EnableGlobalScripts は明示的に 0（無効）に設定されます。

デフォルト: 1
 値: 0 - 無効; 1 - 有効

EnableMobileDevices

Zabbix サーバーでモバイルデバイスのサポートを有効にします。

有効にすると、Zabbix サーバーはモバイルデバイスの追加または削除要求を処理し、それらのデバイスにプッシュ通知を送信します。

デフォルト: 1
 値: 0 - 無効; 1 - 有効

ExportDir

イベント、履歴、トレンドのリアルタイムエクスポートを有効にするためのディレクトリを指定します。

ExportFileSize

最大エクスポートファイルサイズをバイト単位で設定します。ExportDir が設定されている場合、ローテーションに使用されます。

デフォルト: 1G
 範囲: 1M-1G

ExportType

リアルタイムエクスポート のエンティティタイプ（events、history、trends）を指定します。カンマ区切りのリストです。デフォルトではすべてのタイプがエクスポートされます。ExportDir が設定されている場合にのみ有効です。
 注 ExportType が指定されているにもかかわらず ExportDir が設定されていない場合、これは設定エラーとなり、サーバーは起動しません。

history と trends のエクスポートの例:

```
ExportType=history,trends
```

event のみをエクスポートする例:

```
ExportType=events
```

ExternalScripts

外部スクリプトの配置場所を指定します。datadir のコンパイル時インストール変数に依存します。

デフォルト: /usr/local/share/zabbix/externalscripts

Fping6Location

fping6 の場所を指定します。fping6 バイナリの所有者が root であり、SUID フラグが設定されていることを確認してください。fping ユーティリティが IPv6 アドレスを処理できる場合は、空にしてください (Fping6Location=)。

デフォルト: /usr/sbin/fping6

FpingLocation

fping の場所を指定します。fping バイナリの所有者が root であり、SUID フラグが設定されていることを確認してください。

デフォルト: /usr/sbin/fping

FrontendAllowedIP

Web インターフェースからの接続を許可する IP アドレスまたは CIDR 範囲を指定します。カンマ区切りのリストです。デフォルトでは、Web インターフェースのリクエストに対してすべての接続が受け入れられます。IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 は任意の IPv4 または IPv6 アドレスを許可します。0.0.0.0/0 を使用すると、任意の IPv4 アドレスを許可できます。

例:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

高可用性クラスター内のノード名を指定します。空の場合、サーバーはスタンドアロンモードで動作し、空の名前を持つノードが作成されます。

HistoryCacheSize

履歴キャッシュサイズをバイト単位で設定します。履歴データを保存するための共有メモリサイズです。

デフォルト: 16M
 範囲: 128K-16G

HistoryIndexCacheSize

history-index キャッシュサイズを設定します（バイト単位）。history cache に保存される履歴データをインデックス化するための共有メモリサイズです。インデックスキャッシュサイズは、1 アイテムをキャッシュするのにおおよそ 100 バイト必要です。

デフォルト: 4M
 範囲: 128K-16G

HistoryProvider

<provider>[;<options>] 形式で履歴ストレージプロバイダーを設定します。ここで、次のようになります。

- <provider> - サポートされている履歴ストレージプロバイダーのいずれか: sql（デフォルト）、elasticsearch、clickhouse。
- <options> - カンマ区切りの key=value ペア。値は引用符で囲むことができます。

すべての履歴ストレージプロバイダーに対するオプション:

- value_types - 保存する値の種類を設定します（uint、dbl、str、log、text、json）。各値は 1 回だけ使用でき、かつ単一のプロバイダーに対してのみ使用できます。
- precache - 1（デフォルト）を設定すると、履歴ストレージプロバイダーからデータを一括で value cache に事前読み込みします。これにより、再起動後の value cache のウォームアップを大幅に改善できる場合があります。

Elasticsearch およびClickHouse に対するオプション:

- url - 履歴ストレージプロバイダーへの接続に使用する HTTP または HTTPS の URL を設定します。
- log_slow_queries - 履歴ストレージプロバイダーのクエリがログに記録されるまでに許容される最大時間（ミリ秒）を設定します。このオプションは DebugLevel=3 から有効になります。
- source_ip - 履歴ストレージプロバイダーへの送信接続に使用する送信元 IP アドレスを設定します。

Elasticsearch に対するオプション:

- date_index - 日付に基づいて異なるインデックスに履歴値を保存するようにするには 1 を設定します。

ClickHouse に対するオプション:

- db - データベース名を設定します。
- username - データベースにログインするために使用するユーザー名を設定します。
- password - データベースにログインするために使用するパスワードを設定します。
- ssl_cert_file, ssl_key_file, ssl_key_password, ssl_verify_peer, ssl_verify_host, ssl_ca_location, ssl_cert_location, ssl_key_location - データベースへの安全な（TLS/SSL）接続を設定するために、これらのオプションを設定します。

ClickHouse の例:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Elasticsearch の例:

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

ClickHouse と Elasticsearch の例:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

このパラメーターは非推奨です（代わりにHistoryProvider を使用してください）。
 日付に基づいて履歴値を異なるインデックスに保存することを有効にします。このパラメーターはElasticsearch のセットアップで使用されます。

デフォルト: 0
 値: 0 - 無効; 1 - 有効

HistoryStorageURL

このパラメーターは非推奨です（代わりにHistoryProvider を使用してください）。
 履歴ストレージの URL を指定します。このパラメーターはElasticsearch のセットアップで使用されます。

HistoryStorageTypes

このパラメーターは非推奨です（代わりにHistoryProvider を使用してください）。
 履歴ストレージに送信する値のタイプを指定します。カンマ区切りのリストです。このパラメーターはElasticsearch のセットアップで使用されます。

デフォルト: uint,dbl,str,log,text,json

HousekeepingFrequency

ハウスキーピング処理の頻度（時間単位）を設定します。ハウスキーピングとは、データベースから古くなった情報を削除する処理です。
 注: ハウスキーピング処理が過負荷にならないようにするため（たとえば、history と trend の保持期間を大幅に短縮した場合など）、各アイテムごとに、1 回のハウスキーピングサイクルで削除される古い情報は HousekeepingFrequency 時間の 4 倍までに制限されます。したがって、HousekeepingFrequency が 1 の場合、1 回のサイクルで削除される古い情報は最大 4 時間分（最も古いエントリから）です。
 注: サーバー起動時の負荷を下げるため、ハウスキーピングはサーバー起動後 30 分間は延期されます。したがって、HousekeepingFrequency が 1 の場合、サーバー起動後最初のハウスキーピング処理は 30 分後に実行され、その後は 1 時間遅れで繰り返されます。
HousekeepingFrequency を 0 に設定すると、自動ハウスキーピングを無効にできます。この場合、ハウスキーピング処理は housekeeper_execute ランタイム制御オプションでのみ開始でき、1 回のハウスキー

ピングサイクルで削除される古い情報の期間は、前回のハウスキーピングサイクルからの経過時間の 4 倍になりますが、4 時間未満でも 4 日を超えることもありません。
runtime control オプションとhousekeeping procedure の詳細も参照してください。

デフォルト: 1
 範囲: 0-24

Include

設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
 Zabbix のインストール時、このパラメータは /usr/local/etc に設定されますが、コンパイル時に変更した場合はその設定が反映されます。

Include パラメータに関する一般的なルール:

- パスは zabbix_server.conf ファイルの場所からの相対パスにできます。
- 1 つの設定ファイル内で複数の Include パラメータを指定できます。
- Zabbix は Include パラメータを読み取るとすぐに処理するため、設定ファイルの末尾に配置することを検討してください。

含まれるファイルに関するルール:

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルは正しい構文でなければなりません。そうでない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンを含める際のルール:

- 含める順序を前提にしないでください（たとえば、ファイルはアルファベット順には含まれません）。
- 1 つの設定を上書きする目的などで、複数の含まれるファイルに同じパラメータを定義しないでください。どの値が使用されるか予測できないためです。
- 一部のテキストエディタが自動的に作成するバックアップコピーに注意してください(たとえば､include/my_specific.conf を編集すると include/my_specific.conf.BAK が作成される場合があります)。両方のファイルが含まれてしまいます。

ディレクトリから関連するファイルのみを含めるには、ワイルドカード (*) を使ってパターンマッチングを行います。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java ゲートウェイの IP アドレスまたはホスト名を指定します。Java poller が起動している場合にのみ必要です。

JavaGatewayPort

Zabbix Java ゲートウェイのポートを指定します。

デフォルト: 10052
 範囲: 1024-32767

ListenBacklog

TCP キュー内で保留中の接続の最大数を設定します。
 デフォルト値はハードコードされた定数で、システムによって異なります。
 サポートされる最大値もシステムに依存し、値が高すぎる場合は'implementation-specified maximum' に静かに切り詰められることがあります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

トラッパーの待ち受け IP アドレスを指定します。カンマ区切りのリストです。
 このパラメーターが指定されていない場合、トラッパーはすべてのネットワークインターフェースで待ち受けます。

デフォルト: 0.0.0.0

ListenPort

トラッパーの待ち受けポートを指定します。

デフォルト: 10051
 範囲: 1024-32767

LoadModule

サーバーの起動時にロードするモジュールを指定します。モジュールは、サーバーの機能を拡張するために使用されます。モジュールは LoadModulePath で指定されたディレクトリに配置する必要があります。または、モジュール名の前にパスを指定する必要があります。前に指定したパスが絶対パス（/ で始まる）の場合、LoadModulePath は無視されます。
 形式:

- LoadModule=<module.so>
- LoadModule=<path/module.so>
- LoadModule=</abs_path/module.so>

複数の LoadModule パラメータを指定できます。

LoadModulePath

サーバーモジュールの配置場所への完全なパスを指定します。デフォルト値はコンパイルオプションによって異なります。

LogFile

サーバーのログファイルを指定します。

必須: はい。LogType が file に設定されている場合。そうでない場合は不要です

LogFileSize

最大ログファイルサイズを設定します（MB 単位）。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新しく作成されます。

デフォルト: 1
 範囲: 0-1024
 必須: はい。LogType が file に設定されている場合。そうでない場合は不要

LogSlowQueries

データベースクエリがログに記録されるまでの最大実行時間を設定します（ミリ秒）。
0 - 遅いクエリをログに記録しません。
 このオプションは DebugLevel=3 から有効になります。

デフォルト: 0
 範囲: 0-3600000

LogType

ログ出力タイプを指定します: - *file* - LogFile パラメータで指定されたファイルにログを書き込みます; - *system* - syslog にログを書き込みます; - *console* - 標準出力にログを書き込みます。

デフォルト: file

MaxConcurrentChecksPerPoller

各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが一度に実行できる非同期チェックの最大数を設定します。StartHTTPAgentPollers、StartAgentPollers、およびStartSNMPPollers を参照してください。

デフォルト: 1000
 範囲: 1-1000

MaxHousekeeperDelete

history、history_str、history_log、history_uint、history_text、history_bin、history_json、trends、trends_uint および problem テーブルから、1 回の housekeeping サイクルでタスクごとに削除する行の最大数を設定します。
Housekeeping タスクは housekeeper テーブルに [housekeeperid]、[object]、[objectid] の形式で存在します。
 このパラメータは、すでに削除されたアイテムによって残されたデータの削除にのみ適用されます。
 0 に設定すると、制限は一切使用されません。
 この場合は、データベースに過負荷をかけないよう、何をしているかを十分に理解しておく必要があります。
 多数のアイテムが削除されると、housekeeper がそれらのアイテムに関連するすべての履歴データを削除する必要があるため、データベースへの負荷が増加します。
 たとえば、テンプレートから削除する必要があるのが 1 つのアイテムのプロトタイプだけであっても、そのテンプレートが 50 個のホストにリンクされており、各ホストでそのプロトタイプが 100 個の実アイテムに展開される場合、合計 5000 個のアイテムを削除する必要があります (1*50*100)。
 MaxHousekeeperDelete に 500 を設定した場合 (MaxHousekeeperDelete=500)、housekeeper プロセスは 1 サイクルで、削除されたアイテムに対する最大 2500000 個の値 (5000*500) を history および trends テーブルから削除する必要があります。

詳細はhousekeeping procedure も参照してください。

Default: 5000
 Range: 0-1000000

NodeAddress

Web インターフェースがサーバーへ接続する方法を上書きするための、オプションのポート付き IP またはホスト名を指定します。
 形式: <address>[:<port>]

IP またはホスト名が設定されていない場合、ListenIP の値が使用されます。ListenIP が設定されていない場合、localhost の値が使用されます。
 ポートが設定されていない場合、ListenPort の値が使用されます。ListenPort が設定されていない場合、10051 の値が使用されます。

 このオプションは、Web インターフェースの設定で指定されたアドレスによって上書きできます。

 関連項目: HANodeName パラメーター; 高可用性の有効化。

デフォルト: localhost:10051

PidFile

PID ファイルを指定します。

デフォルト: /tmp/zabbix_server.pid

ProblemHousekeepingFrequency

削除されたトリガーに対する問題を削除する頻度を設定します（秒単位）。
 housekeeping 手順の詳細も参照してください。

デフォルト: 60
 範囲: 1-3600

ProxyConfigFrequency

サーバーが Zabbix プロキシに設定データを送信する頻度を設定します。パッシブモードのプロキシでのみ使用されます。

デフォルト: 10
 範囲: 1-604800

ProxyDataFrequency

サーバーが Zabbix プロキシから履歴データを要求する頻度を設定します。パッシブモードのプロキシにのみ使用されます。

デフォルト: 1
 範囲: 1-3600

ServiceManagerSyncFrequency

サービスマネージャーの設定を同期する頻度を指定します（秒）。

デフォルト: 60
 範囲: 1-3600

SMSDevices

Zabbix サーバーで使用を許可するモデムファイルを指定します。カンマ区切りのリストです。
 このパラメータが設定されていない場合、SMS の送信はできません。

例:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

SNMP トラップデーモンからサーバーへデータを渡すために使用する一時ファイルを指定します。
zabbix_trap_receiver.pl または SNMPTT の設定ファイルと同じである必要があります。

デフォルト: /tmp/zabbix_traps.tmp

SocketDir

内部 Zabbix サービスで使用される IPC ソケットを保存するディレクトリを指定します。

デフォルト: /tmp

SourceIP

次の接続に使用する送信元 IP アドレスを指定します。

- Zabbix プロキシおよび Zabbix エージェントへの送信接続
- エージェントレス接続（VMware、SSH、JMX、SNMP、Telnet、およびシンプルチェック）
- HTTP エージェント接続
- スクリプトアイテムの JavaScript HTTP リクエスト
- 前処理の JavaScript HTTP リクエスト
- 通知メールの送信（SMTP サーバーへの接続）
- webhook 通知（JavaScript HTTP 接続）
- Vault への接続

SSHKeyLocation

SSH チェックとアクションで使用する公開鍵と秘密鍵の場所を指定します。

SSLCertLocation

クライアント認証に使用する SSL クライアント証明書ファイルの場所を指定します。
 このパラメータは、Web 監視、Vault との通信、履歴保存プロバイダーの ClickHouse、およびテレメトリデータソースの ClickHouse で使用されます。

SSLKeyLocation

クライアント認証用の SSL 秘密鍵ファイルの場所を指定します。
 このパラメータは、Web 監視、Vault との通信、履歴保存プロバイダーの ClickHouse、およびテレメトリデータソースの ClickHouse で使用されます。

SSLCALocation

SSL サーバー証明書の検証に使用する認証局（CA）ファイルの場所を指定します。設定されていない場合は、システム全体のディレクトリが使用されます。
 このパラメータの値は、libcurl オプション CURLOPT_CAPATH として設定されることに注意してください。7.42.0 より前の libcurl バージョンでは、libcurl が OpenSSL を使用するようにコンパイルされている場合にのみ有効です。詳細については、cURL の Web ページを参照してください。
 このパラメータは、Web 監視、SMTP 認証、HTTP エージェントアイテム、Vault との通信、履歴ストレージプロバイダーの ClickHouse、およびテレメトリデータソースの ClickHouse で使用されます。

StartAgentPollers

Zabbix エージェントの事前 fork 済みインスタンス数pollers を設定します。MaxConcurrentChecksPerPoller を参照してください。

デフォルト: 1
 範囲: 0-1000

StartAlerters

alerters の事前フォーク済みインスタンス数を設定します。

デフォルト: 3
 範囲: 1-100

StartBrowserPollers

ブラウザーアイテムのpollersの事前フォーク済みインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartConnectors

connector workers の事前 fork 済みインスタンス数を設定します。connector worker が開始されると、connector manager プロセスは自動的に起動します。

デフォルト: 0
 範囲: 0-1000

StartDBSyncers

history syncers の事前 fork されたインスタンス数を設定します。
 注: この値を変更する際は注意してください。増やしても、かえって悪影響を及ぼす場合があります。おおむね、デフォルト値で最大 4000 NVPS まで処理するのに十分です。

デフォルト: 4
 範囲: 1-100

StartDiscoverers

ディスカバリーワーカー[1] の事前フォークされたインスタンス数を設定します。

デフォルト: 5
 範囲: 0-1000

StartEscalators

escalators の事前フォークされたインスタンス数を設定します。

デフォルト: 1
 範囲: 1-100

StartHistoryPollers

history pollers の事前フォーク済みインスタンス数を設定します。
 計算チェックでのみ必要です。

デフォルト: 5
 範囲: 0-1000

StartHTTPAgentPollers

HTTP エージェントのpollers の事前 fork されたインスタンス数を設定します。MaxConcurrentChecksPerPoller を参照してください。

デフォルト: 1
 範囲: 0-1000

StartHTTPPollers

HTTP pollers[1] の事前フォークされたインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartIPMIPollers

IPMI pollers の事前フォーク済みインスタンス数を設定します。

デフォルト: 0
 範囲: 0-1000

StartJavaPollers

Java pollers[1] の事前フォークされたインスタンス数を設定します。

デフォルト: 0
 範囲: 0-1000

StartLLDProcessors

low-level discovery (LLD) のworker[1] の事前フォーク済みインスタンス数を設定します。
 LLD worker が開始されると、LLD マネージャープロセスは自動的に起動されます。

デフォルト: 2
 範囲: 1-100

StartODBCPollers

ODBC pollers[1] の事前フォークされたインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartPingers

ICMP pingers[1] の事前フォークされたインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartPollersUnreachable

到達不能ホスト用のポーラー（IPMI および Java を含む）の事前フォーク済みインスタンス数を設定します[1]。
 通常のポーラー、IPMI ポーラー、または Java ポーラーが起動している場合は、到達不能ホスト用のポーラーが少なくとも 1 つ実行されている必要があります。

デフォルト: 1
 範囲: 0-1000

StartPollers

pollers[1] の事前 fork 済みインスタンス数を設定します。

デフォルト: 5
 範囲: 0-1000

StartPreprocessors

前もって起動する前処理のworker[1] スレッド数を、利用可能な CPU コア数以上に設定します。前処理が CPU ボトルネックではなく、ネットワーク要求が多い場合は、worker をさらに増やしてください。

デフォルト: 16
 範囲: 1-1000

StartProxyPollers

パッシブプロキシ用の poller[1] の事前 fork されたインスタンス数を設定します。

デフォルト: 1
 範囲: 0-250

StartReportWriters

report writers の事前 fork 済みインスタンス数を設定します。
0 に設定すると、スケジュールされたレポート生成は無効になります。
report writer が起動すると、レポートマネージャープロセスは自動的に起動されます。

デフォルト: 0
 範囲: 0-100

StartSNMPPollers

SNMP pollers の事前フォークされたインスタンス数を設定します。`MaxConcurrentChecksPerPoller` を参照してください。

デフォルト: 1
 範囲: 0-1000

StartSNMPTrapper

1 に設定すると、SNMP trapper プロセスを開始します。

デフォルト: 0
 範囲: 0-1

StartTelemetryQueryPollers

非同期 Telemetry クエリpollerの、フォーク前に起動するインスタンス数を設定します。
MaxConcurrentChecksPerPollerを参照してください。

デフォルト: 1
 範囲: 0-1000

StartTimers

timers の事前フォークされたインスタンス数を設定します。
Timers はメンテナンス期間を処理します。

デフォルト: 1
 範囲: 1-1000

StartTrappers

trappers[1] の事前フォークされたインスタンス数を設定します。
 トラッパーは、Zabbix sender、アクティブエージェント、およびアクティブプロキシからの受信接続を受け付けます。

デフォルト: 5
 範囲: 0-1000

StartVMwareCollectors

事前に fork されたVMware collector インスタンスの数を設定します。

デフォルト: 0
 範囲: 0-250

StatsAllowedIP

統計情報の要求を許可する外部 Zabbix インスタンスの IP アドレス（必要に応じて CIDR 表記）または DNS 名を指定します。カンマ区切りのリストです。
 このパラメータが設定されていない場合、統計情報の要求は受け付けられません。
 IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 により任意の IPv4 または IPv6 アドレスが許可されます。0.0.0.0/0 は任意の IPv4 アドレスを許可するために使用できます。

例:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

テレメトリデータソースを <provider>[;<options>] の形式で設定します。

- <provider> - サポートされているテレメトリデータソースのいずれか: clickhouse
- <options> - カンマ区切りの key=value ペア。値は引用符で囲むことができます。

ClickHouse のオプション:

- url - テレメトリデータソースへの接続に使用する HTTP または HTTPS URL を設定します。
- db - データベース名を設定します。
- username - データベースへのログインに使用するユーザー名を設定します。
- password - データベースへのログインに使用するパスワードを設定します。
- ssl_cert_file、ssl_key_file、ssl_key_password、ssl_verify_peer、ssl_verify_host、ssl_ca_location、ssl_cert_location、ssl_key_location - データベースへのセキュアな（TLS/SSL）接続を設定するためのオプションを設定します。
- source_ip - テレメトリデータソースへの送信接続に使用する送信元 IP アドレスを設定します。
- vault_path - username と password の値を保持する Vault シークレットへのパスを設定します。username および password オプションと同時に使用しないでください。

ClickHouse の例:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Vault を使用する ClickHouse の例:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

タイムアウト

Zabbix プロキシ、エージェント、Web サービス、およびレガシー SNMP チェック（単一の OID 番号または文字列）との接続確立とデータ交換を待機する時間を、秒単位で指定します。

このパラメーターは、次の各種通信操作の継続時間を定義します。

- Zabbix エージェント上でのリモートコマンド実行
- SSH/Telnet コマンド実行
- Zabbix Web サービスへのリクエスト
- mediatype テストリクエストおよび history.push メソッドの通信タイムアウト
- IPMI インターフェースが利用できなくなった場合のアイテムの再スケジュール
- 権限または暗号化の問題によりデータ交換に失敗した場合の Zabbix プロキシへの応答送信
- IPC 非同期ソケットおよびランタイム制御オプションの期限
- JMX 接続
- リモートの Zabbix プロキシまたはサーバーからの統計情報の取得
- Zabbix Web インターフェースへの応答送信
- 非同期ポーラーの DNS リクエスト
- アクティブチェックのハートビートへの応答
- アクティブエージェントからの Zabbix エージェントデータ（値）の取得
- Zabbix sender からのデータ取得
- trapper がリクエストの処理に失敗した場合の応答
- Zabbix エージェントへのアクティブチェック一覧の送信

このタイムアウトは、Web インターフェースでflexible timeout 設定が構成されているチェック（グローバル、プロキシ、またはアイテムごとのレベル）には 使用されません。たとえば、SNMP の walk[OID] および get[OID] アイテムは Web インターフェースで設定されたタイムアウトを使用しますが、レガシー SNMP チェックは引き続きサーバーのタイムアウト値を使用します。

デフォルト: 3
 範囲: 1-30

TLSCAFile

ピア証明書の検証に使用する、最上位の CA 証明書を含むファイルのフルパスを指定します。これは Zabbix コンポーネント間の暗号化通信で使用されます。

TLSCertFile

Zabbix コンポーネント間の暗号化通信に使用する、サーバー証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

TLSCipherAll

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 で OpenSSL 1.1.1 以降に対する暗号スイート文字列を指定します。証明書ベースおよび PSK ベースの暗号化に対する既定の暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。
証明書ベースの暗号化に対する既定の暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 で OpenSSL 1.1.1 以降に対する暗号スイート文字列を指定します。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。
PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号スイート文字列を指定します。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

失効した証明書を含むファイルのフルパスを指定します。このパラメーターは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSFrontendAccept

Web インターフェースからの受信接続に対する暗号化レベルを指定します。

複数の値をカンマ区切りで指定できます。

- unencrypted - 暗号化なしの接続を受け入れます。
- cert - TLS と証明書で保護された接続を受け入れます。

デフォルト: `unencrypted`

TLSFrontendCertIssuer

許可する Web インターフェース証明書の発行者を指定します。

TLSFrontendCertSubject

許可する Web インターフェース証明書のサブジェクトを指定します。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用される、サーバーの秘密鍵を含むファイルの完全なパス名を指定します。

TLSListen

トラッパーソケットで TLS のみの接続を要求します。

サポートされる値:

- required - TLS 接続のみを受け入れます

TmpDir

一時ディレクトリを指定します。

デフォルト: `/tmp`

TrapperTimeout

次の処理に対する trapper のタイムアウトを秒単位で指定します。

- Zabbix プロキシからの履歴データの取得
- Zabbix プロキシへの設定データの送信
- Zabbix サーバー上でのグローバルスクリプトの実行またはリモートコマンドの実行

デフォルト: 300
 範囲: 1-300

TrendCacheSize

トレンドキャッシュサイズをバイト単位で設定します。
 トレンドを保存するための共有メモリサイズです。

デフォルト: 4M
 範囲: 128K-16G

TrendFunctionCacheSize

トレンド関数キャッシュのサイズをバイト単位で設定します。
 計算済みのトレンド関数データをキャッシュするための共有メモリサイズです。

デフォルト: 4M
 範囲: 128K-2G

UnavailableDelay

unavailability 期間中にホストの到達可能性を確認する頻度を秒単位で設定します。

デフォルト: 60
 範囲: 1-3600

UnreachableDelay

到達不能期間中にホストの到達可能性を確認する頻度を秒単位で設定します。

デフォルト: 15
 範囲: 1-3600

UnreachablePeriod

到達不能になってから何秒後にホストを利用不可として扱うかを設定します。

デフォルト: 45
 範囲: 1-3600

ユーザー

システム上に存在する特定のユーザーに権限を下げます。
'root' として実行され、AllowRoot が無効な場合にのみ有効です。

デフォルト: `zabbix`

ValueCacheSize

履歴値の value cache のサイズを設定します（バイト単位）。
 アイテムの履歴データ要求をキャッシュするための共有メモリサイズです。
 0 に設定すると value cache が無効になります（推奨されません）。
 value cache の共有メモリが不足すると、5 分ごとに警告メッセージがサーバーのログに書き込まれます。

デフォルト: 8M
 範囲: 0、128K-64G

Vault

vault プロバイダーを指定します: - *HashiCorp* - HashiCorp KV Secrets Engine version 2; - *CyberArk* - CyberArk Central Credential Provider
Web インターフェースで設定された vault プロバイダーと一致している必要があります。

デフォルト: `HashiCorp`

VaultAppRoleID

HashiCorp Vault の AppRole ID を指定します。
 設定すると、AppRole 認証が有効になります。VaultToken と同時には使用できません。
 HashiCorp Vault でのみサポートされています。

VaultAppSecretID

HashiCorp Vault の AppRole secret ID を指定します。
 HashiCorp Vault でのみサポートされています。

VaultDBPath

データベース認証情報をキーで取得する場所を指定します。これは、Vault に応じて Vault のパスまたはクエリです。

HashiCorp で使用されるキーは `password` と `username` です。

`VaultPrefix=/v1/secret/data/zabbix/` を指定した場合のパスの例:

```
database
```

VaultPrefix を指定しない場合のパスの例:

```
secret/zabbix/database
```

CyberArk で使用されるキーは `Content` と `UserName` です。

例:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

このオプションは、`DBUser` と `DBPassword` が指定されていない場合にのみ使用できます。

VaultPrefix

Vault の種類に応じて、vault パスまたはクエリのカスタムプレフィックスを指定します。指定しない場合は、最適なデフォルト値が使用されます。
 HashiCorp では、`VaultPrefix` を指定しない場合、`data` が mountpoint の後に自動的に追加されることに注意してください。

HashiCorp のプレフィックス例:

```
/v1/secret/data/zabbix/
```

Cyberark のプレフィックス例:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

クライアント認証に使用する SSL 証明書ファイルの名前を指定します。証明書ファイルは PEM1 形式である必要があります。
 証明書ファイルに秘密鍵も含まれている場合は、SSL キーファイルのフィールドは空のままにしてください。
 このファイルを含むディレクトリは、設定パラメータ SSLCertLocation で指定します。
 このオプションは省略できますが、CyberArkCCP vault では指定することを推奨します。

VaultTLSKeyFile

クライアント認証に使用する SSL 秘密鍵ファイルの名前を指定します。秘密鍵ファイルは PEM1 形式である必要があります。
 このファイルを含むディレクトリは、設定パラメータ SSLKeyLocation で指定します。
 このオプションは省略可能ですが、CyberArkCCP vault では指定することを推奨します。

VaultToken

HashiCorp Vault の認証トークンを指定します。HashiCorp Vault の認証トークンは、Vault macros で指定されたパスに対する読み取り専用権限と、任意の VaultDBPath 設定パラメータで指定されたパスに対する読み取り専用権限を持ち、Zabbix サーバー専用に生成されている必要があります。
 `VaultToken` と `VAULT_TOKEN` 環境変数が同時に定義されている場合はエラーになります。
 VaultAppRoleID と併用できません。
 HashiCorp Vault でのみサポートされます。

VaultURL

vault サーバーの URL を指定します。SSLCALocation が指定されていない場合は、システム全体の CA 証明書ディレクトリが使用されます。

デフォルト: `https://127.0.0.1:8200`

VMwareCacheSize

VMware データを保存するための共有メモリサイズを設定します。VMware 内部チェック zabbix[vmware,buffer,...] を使用して、VMware キャッシュの使用状況を監視できます（内部チェックを参照）。
 なお、起動するように設定された vmware collector インスタンスがない場合、共有メモリは割り当てられません。

デフォルト: 8M
 範囲: 256K-2G

VMwareFrequency

単一の VMware サービスからデータを収集する頻度を設定します（秒単位）。この頻度は、いずれかの VMware 監視アイテムの最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwarePerfFrequency

単一の VMware サービスからパフォーマンスカウンター統計を取得する頻度を設定します（秒単位）。
 この頻度は、VMware パフォーマンスカウンターを使用する VMware 監視アイテム の最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwareTimeout

vmware collector が VMware サービス（vCenter または ESX ハイパーバイザー）からの応答を待機する最大秒数を設定します。

デフォルト: 10
 範囲: 1-300

WebDriverURL

WebDriver インターフェースの URL を指定します。

例（Selenium WebDriver standalone server と併用する場合）:

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Web サービスの URL を http[s]://host:port/report の形式で指定します。

例:

```
WebServiceURL=http://localhost:10053/report
```

http:// スキームは、TLS を使用しない（HTTP）接続の場合にのみ省略できます。TLS が設定されている場合は、https:// を使用する必要があります。

脚注

[1] データ収集プロセス（pollers、unreachable pollers、ODBC pollers、HTTP pollers、Java pollers、pingers、trappers、proxypollers）が多すぎると、IPMI manager、SNMP trapper、preprocessing workers、discovery workers と合わせて、preprocessing manager のプロセスごとのファイルディスクリプタ上限を使い切る可能性があります。

> **Warning:**
> ファイルディスクリプタ上限を使い切ると、Zabbix サーバーは停止します。通常は起動直後に発生しますが、より時間がかかる場合もあります。このような問題を避けるには、Zabbix server configuration file を確認し、同時チェック数とプロセス数を最適化してください。さらに必要に応じて、システムの制限を確認・調整し、ファイルディスクリプタ上限が十分に高く設定されていることを □□□□□□□ してください。

プロキシ

概要　Zabbix プロキシは、1 台以上の監視対象デバイスから監視データを収集し、その情報を Zabbix サーバーに送信するプロセスです。基本的には、サーバーの代わりに動作します。収集したすべてのデータはローカルにバッファリングされた後、プロキシが所属する Zabbix サーバーに転送されます。

プロキシの導入は任意ですが、1 台の Zabbix サーバーの負荷を分散するうえで非常に有効です。プロキシのみがデータを収集する構成にすると、サーバーで必要となる CPU およびディスク I/O の処理負荷を軽減できます。

Zabbix プロキシは、リモート拠点、支店、ローカル管理者がいないネットワークを一元的に監視するための理想的なソリューションです。

Zabbix プロキシには個別のデータベースが必要です。

Attention:
Zabbix プロキシでサポートされているデータベースは、SQLite、MySQL、PostgreSQL です。

関連項目:

- プロキシを使用した分散監視
- プロキシの負荷分散と高可用性
- 監視設定の同期
- サーバーとプロキシ間のデータ交換プロトコル
- 設定パラメータ: Zabbix プロキシ
- Zabbix Cloud プロキシのセットアップ

プロキシの実行　パッケージとしてインストールした場合

Zabbix プロキシはデーモンプロセスとして実行されます。プロキシは次のコマンドで起動できます:

```
systemctl start zabbix-proxy
```

これはほとんどの GNU/Linux システムで動作します。他のシステムでは、次のコマンドを実行する必要があるかもしれません:

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

同様に、Zabbix プロキシの停止/再起動/ステータス表示には、以下のコマンドを使用します:

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

手動で起動する

上記でうまくいかない場合は、手動で起動する必要があります。
zabbix_proxy バイナリへのパスを見つけて、次を実行します。

```
zabbix_proxy
```

Zabbix プロキシでは、次のコマンドラインパラメーターを使用できます。

Parameter	Description
`-c --config <config-file>`	設定ファイルへのパス。
`-f --foreground`	Zabbix プロキシをフォアグラウンドで実行します。
`-R --runtime-control <option>`	管理機能を実行します。
`-T --test-config`	設定ファイルを検証して終了します。
`-h --help`	ヘルプ情報を表示して終了します。
`-V --version`	バージョン番号を表示します。

コマンドラインパラメーターを指定して Zabbix プロキシを実行する例:

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

ランタイム制御

ランタイム制御オプション:

Option	Description	Target
`config_cache_reload`	設定キャッシュを再読み込みします。現在キャッシュが読み込み中の場合は無視されます。 アクティブな Zabbix プロキシは Zabbix サーバーに接続して設定データを要求します。 パッシブな Zabbix プロキシは、次回サーバーがプロキシに接続したときに Zabbix サーバーから設定データを要求します。	

Option	Description	Target
history_cache_clear[=<target>]	指定されたアイテムの history cache をクリアします。 アイテムの最初と最後の値を除く、すべての値に影響します。	**target** - アイテムの ID。
diaginfo[=<section>]	プロセスのログファイルに診断情報を収集します。	historycache - history cache の統計情報; preprocessing - preprocessing manager の統計情報; locks - mutex の一覧（*BSD* システムでは空です）。
snmp_cache_reload	SNMP cache を再読み込みします - すべてのホストの SNMP エンジンプロパティ（engine time、engine boots、engine id、credentials）をクリアします。SNMP の問題をトラブルシュートする際に、グローバルなキャッシュクリアを強制するために使用します。	
housekeeper_execute	housekeeping 手順を開始します。housekeeping 手順が現在進行中の場合は無視されます。	
log_level_increase[=<target>]	ログレベルを上げます。target が指定されていない場合は、すべてのプロセスに影響します。 *BSD* システムではサポートされていません。	**process type** - 指定した種類のすべてのプロセス（例: poller）。 すべてのプロキシプロセスの種類 を参照してください。 **process type,N** - プロセスの種類と番号（例: poller,3） **pid** - プロセス識別子（1 から 65535）。より大きい値を指定する場合は、target を'process type,N' として指定します。
log_level_decrease[=<target>]	ログレベルを下げます。target が指定されていない場合は、すべてのプロセスに影響します。 *BSD* システムではサポートされていません。	
prof_enable[=<target>]	プロファイリングを有効にします。 target が指定されていない場合は、すべてのプロセスに影響します。 有効化されたプロファイリングでは、関数名ごとのすべての rwlock/mutex の詳細が提供されます。	**process type** - 指定した種類のすべてのプロセス（例: history syncer）。 すべてのプロキシプロセスの種類 を参照してください。 **process type,N** - プロセスの種類と番号（例: history syncer,1）。 **pid** - プロセス識別子（1 から 65535）。より大きい値を指定する場合は、target を'process type,N' として指定します。 **scope** - rwlock、mutex、processing は、プロセスの種類と番号（例: history syncer,1,processing）またはその種類のすべてのプロセス（例: history syncer,rwlock）と組み合わせて使用できます。
prof_disable[=<target>]	プロファイリングを無効にします。 target が指定されていない場合は、すべてのプロセスに影響します。	**process type** - 指定した種類のすべてのプロセス（例: history syncer）。 すべてのプロキシプロセスの種類 を参照してください。 **process type,N** - プロセスの種類と番号（例: history syncer,1）。 **pid** - プロセス識別子（1 から 65535）。より大きい値を指定する場合は、target を'process type,N' として指定します。

プロセスの種類がランタイム制御の target として指定されると、そのコマンドはその種類の実行中のプロセスにのみ適用されます。指定した種類のプロセスが現在実行されていない場合、コマンドはエラーを返します:

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

ランタイム制御を使用してプロキシの設定キャッシュを再読み込みする例:

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

ランタイム制御を使用してアイテムの history cache をクリアする例:

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

ランタイム制御を使用して診断情報を収集する例:

```
### プロキシのログファイルに利用可能なすべての診断情報を収集します:
zabbix_proxy -R diaginfo

### プロキシのログファイルに history cache の統計情報を収集します:
zabbix_proxy -R diaginfo=historycache
```

ランタイム制御を使用して SNMP cache を再読み込みする例:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
SNMPv3 インターフェースが Zabbix UI から更新された場合、ほとんどの場合、Zabbix はそのインターフェースの新しい SNMPv3 credentials を自動的に再読み込みします。-R snmp_cache_reload は、credentials の変更後もポーリングが失敗し続ける場合（たとえば engineBoots/engineID の不整合や RFC 非準拠デバイスが原因の場合）、またはトラブルシューティングのためにグローバルな SNMP cache クリアを強制する必要がある場合にのみ使用してください。

ランタイム制御を使用して housekeeper の実行を開始する例:

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

ランタイム制御を使用してログレベルを変更する例:

```
### すべてのプロセスのログレベルを上げます:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### 2 番目の poller プロセスのログレベルを上げます:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### PID 1234 のプロセスのログレベルを上げます:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### すべての http poller プロセスのログレベルを下げます:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

ユーザーとしてのプロセス実行

Zabbix プロキシは、root 以外のユーザーとして実行するように設計されています。起動した root 以外のユーザーとして実行されます。そのため、問題なく任意の root 以外のユーザーとしてプロキシを実行できます。

「root」として実行しようとすると、ハードコードされた「zabbix」ユーザーに切り替わります。このユーザーはシステム上に存在している必要があります。プロキシ設定ファイルの`AllowRoot` パラメータを適切に変更した場合にのみ、「root」としてプロキシを実行できます。

設定ファイル

設定ファイルのデフォルトパスは、オペレーティングシステムと Zabbix のインストール方法によって異なります。

- ソースからインストールした場合: /usr/local/etc/zabbix_proxy.conf、またはコンパイル時 に設定された変数 --sysconfdir または --prefix。
- パッケージからインストールした場合: /etc/zabbix/zabbix_proxy.conf。

zabbix_proxy の設定方法の詳細については、設定ファイル のオプションを参照してください。

プロキシのプロセスタイプとスレッド

- agent poller - パッシブチェック用の非同期ポーラープロセス（ワーカースレッド付き）;
- apm manager - アプリケーションパフォーマンスモニタリング（APM）用のマネージャープロセス;
- apm worker - apm manager からのアプリケーションパフォーマンスモニタリング（APM）タスクを処理するプロセス;
- availability manager - ホストの可用性を更新するプロセス;
- browser poller - ブラウザーアイテムのチェック用ポーラー;
- configuration syncer - 設定データのメモリ内キャッシュを管理するプロセス;

- data sender - プロキシデータ送信プロセス;
- discovery manager - デバイス検出用のマネージャープロセス;
- discovery worker - discovery manager からの検出タスクを処理するプロセス;
- history syncer - 履歴データベースライター;
- housekeeper - 古いアイテム履歴を削除するプロセス;
- http agent poller - HTTP チェック用の非同期ポーラープロセス（ワーカースレッド付き）;
- http poller - Web 監視ポーラー;
- icmp pinger - icmpping チェック用ポーラー;
- internal poller - 内部チェック用ポーラー;
- ipmi manager - IPMI ポーラーマネージャー;
- ipmi poller - IPMI チェック用ポーラー;
- java poller - Java チェック用ポーラー;
- odbc poller - ODBC チェック用ポーラー;
- poller - パッシブチェック用の通常のポーラー;
- preprocessing manager - 前処理ワーカースレッドを使用して前処理タスクを管理するマネージャー;
- preprocessing worker - データ前処理用のスレッド;
- self-monitoring - 内部サーバー統計を収集するプロセス;
- snmp poller - SNMP チェック用の非同期ポーラープロセス（ワーカースレッド付き。walk[OID] および get[OID] アイテムのみ）;
- snmp trapper - SNMP トラップ用のトラッパー;
- supervisor - サーバープロセスおよびスレッドの起動とシャットダウンを調整し、それらの準備状態を監視するプロセス;
- task manager - 他のコンポーネントから要求されたタスク（障害のクローズ、障害の認識、アイテム値の即時チェック、リモートコマンド機能など）をリモート実行するプロセス;
- telemetry query poller - テレメトリクエリアイテムのチェック用ポーラー;
- trapper - アクティブチェック、トラップ、プロキシ通信に使用するトラッパー;
- unreachable poller - 到達不能なデバイス用のポーラー;
- vmware collector - VMware サービスからデータを収集する VMware データコレクター。

これらのプロセスタイプは、プロキシのログファイルで確認できます。

プロキシのログファイルは、ファイル所有者だけが読み書きできる権限で作成されます。また、所有者グループはファイルを読み取ることができます。その他の権限はすべて拒否されます。

さまざまなタイプの Zabbix プロキシプロセスは、zabbix[process,<type>,<mode>,<state>] 内部itemを使用して監視できます。

History syncer トランザクション統計

history syncer プロセスのタイトルには、history syncer トランザクションに関する詳細な統計が表示されます。

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

processed...in N（<timings>）sec に含まれる時間は、次のとおりです。

- アイテムの値をデータベースに書き込むのにかかった時間。
- アイテムデータ（状態、エラー）を更新するのにかかった時間。

ハウスキーピング手順

Zabbix プロキシには、古くなったアイテムの履歴データとトレンドを削除する housekeeper プロセスがあります。このプロセスは、HousekeepingFrequency によって決まる頻度でサイクル実行され、各サイクルでの削除上限はProxyLocalBufferとProxyOfflineBuffer によって決まります。Zabbix サーバーのハウスキーピング手順とは異なり、プロキシの housekeeper プロセスは housekeeper テーブルを使用しません。代わりに、各ハウスキーピングサイクルで古くなったデータをすべて削除します。

サポートされるプラットフォーム　Zabbix プロキシは、Zabbix サーバーと同じサポートされるプラットフォーム の一覧で動作します。

メモリバッファ　メモリバッファを使用すると、新しいデータ（アイテムの値、ネットワーク検出、ホストの自動登録）をバッファに保存し、データベースにアクセスせずに Zabbix サーバーへアップロードできます。メモリバッファは、Zabbix 7.0 でプロキシに導入されました。

Zabbix 7.0 より前のインストールでは、収集したデータは Zabbix サーバーへアップロードする前にデータベースへ保存されていました。これらのインストールでは、Zabbix 7.0 へのアップグレード後も、これがデフォルトの動作です。

パフォーマンスを最適化するため、プロキシでメモリバッファを使用するように設定することを推奨します。これを行うには、ProxyBufferModeの値を disk（既存のインストールでハードコードされているデフォルト値）から hybrid（推奨）または memory に変更します。また、メモリバッファのサイズ（ProxyMemoryBufferSizeパラメータ）を設定する必要があります。

hybrid モードでは、プロキシが停止した場合、バッファがいっぱいになった場合、またはデータが古くなりすぎた場合に、未送信のデータをデータベースへフラッシュすることで、データ損失からバッファを保護します。すべての値がデータベースへフラッシュされると、プロキシはメモリバッファの使用に戻ります。

memory モードではメモリバッファが使用されますが、データ損失に対する保護はありません。プロキシが停止した場合、またはメモリがいっぱいになった場合、未送信のデータは破棄されます。

hybrid モード（ProxyBufferMode=hybrid）は、Zabbix 7.0 以降のすべての新規インストールに適用されます。

ProxyMemoryBufferSizeやProxyMemoryBufferAgeなどの追加パラメータでは、それぞれメモリバッファのサイズと、バッファ内のデータの最大経過時間を定義します。

設定に矛盾がある場合、プロキシはエラーを出力して起動に失敗します。たとえば、次の場合です。

- ProxyBufferMode が hybrid または memory に設定され、ProxyMemoryBufferSize が 0 の場合。
- ProxyBufferMode が hybrid または memory に設定され、ProxyLocalBuffer が 0 でない場合。

ロケール　プロキシでは、一部のテキストアイテムを正しく解釈できるように、UTF-8 ロケールが必要です。多くの最新の Unix 系システムでは UTF-8 ロケールがデフォルトになっていますが、システムによっては明示的に設定する必要があります。

メンテナンス中のキューの計算

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してく

プロキシの背後で **Zabbix Web** インターフェースを使用する　Zabbix Web インターフェースがプロキシサーバーの背後で実行されている場合、リバースプロキシされたパスに一致するよう、プロキシ設定ファイル内の Cookie パスを書き換える必要があります。以下の例を参照してください。Cookie パスを書き換えない場合、Zabbix Web インターフェースへのログイン時に、ユーザーが認証の問題に遭遇する可能性があります。

NGINX の設定例

```
### ..
location / {
### ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
### ..
```

Apache の設定例

```
### ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
### ..
```

プロキシを使用した分散監視

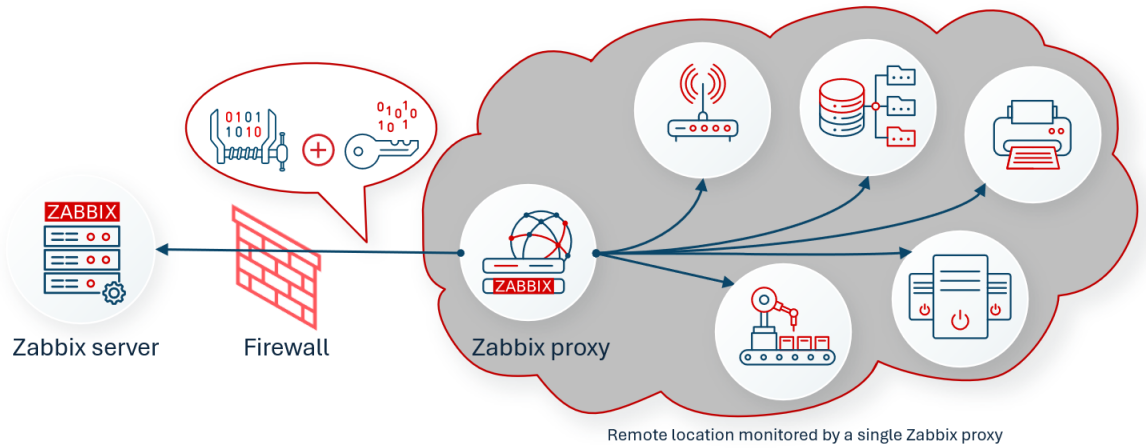
概要

Zabbix プロキシは、効果的で信頼性の高い分散監視環境を構築するために使用できます。

プロキシは、パフォーマンスデータと可用性データを収集するためにローカルに配置します。その後、データは一元管理されたサーバーに報告されます。

Zabbix プロキシは、次の用途に使用できます。

- リモートロケーションの監視
- 通信が不安定なロケーションの監視
- 数千台のデバイスを監視する際の Zabbix サーバーの負荷軽減
- アプリケーションパフォーマンス監視（APM）のためのテレメトリデータの収集



プロキシには、Zabbix サーバーへの TCP 接続が 1 つだけ必要です。このため、ファイアウォールルールを 1 つ設定するだけで済み、ファイアウォールを容易に通過できます。

Attention:
Zabbix プロキシは、専用のデータベースを使用する必要があります。Zabbix サーバーのデータベースを指定すると、設定が破損します。

プロキシが収集したすべてのデータは、サーバーに送信される前にローカルに保存されます。これにより、サーバーとの一時的な通信障害によるデータ損失を防止できます。プロキシ設定ファイルの `ProxyLocalBuffer` および `ProxyOfflineBuffer` パラメータで、データをローカルに保持する期間を制御します。

Attention:
Zabbix サーバーのデータベースから最新の設定変更を直接受信するプロキシは、Zabbix サーバーよりも新しい設定を持つ場合があります。これは、CacheUpdateFrequencyの値によって、Zabbix サーバーの設定更新が遅れる可能性があるためです。その結果、プロキシがデータの収集を開始して Zabbix サーバーに送信しても、Zabbix サーバーがそのデータを無視する場合があります。

Zabbix プロキシはデータコレクターです。トリガーの計算、イベントの処理、アラートの送信は行いません。プロキシの機能の概要については、次の表を参照してください。

機能		プロキシでのサポート
アイテム		
	Zabbix エージェントチェック	はい
	Zabbix エージェントチェック（アクティブ）	はい [1]
	シンプルチェック	はい
	トラッパーアイテム	はい
	SNMP チェック	はい
	SNMP トラップ	はい
	IPMI チェック	はい
	JMX チェック	はい
	ログファイル監視	はい
	内部チェック	はい
	SSH チェック	はい
	Telnet チェック	はい
	外部チェック	はい
	依存アイテム	はい
	スクリプトアイテム	はい
	ブラウザーアイテム	はい
	テレメトリクエリアイテム	はい
組み込み Web 監視		はい
アイテム値の前処理		はい
ネットワークディスカバリ		はい
アクティブエージェントの自動登録		はい
ローレベルディスカバリ		はい [2]
リモートコマンド		はい
テレメトリデータの収集（APM）		はい [3]

機能	プロキシでのサポート
トリガーの計算	いいえ
イベントの処理	いいえ
イベント相関	いいえ
アラートの送信	いいえ

Note:
[1] エージェントがアクティブチェックについてサーバーではなくプロキシに問い合わせるようにするには、エージェント設定ファイルの **ServerActive** パラメータにプロキシを指定する必要があります。
 [2] LLD では、Zabbix プロキシはデータの収集と前処理のみを行い、その後、さらなる処理のために Zabbix サーバーへ送信します。
 [3] テレメトリデータは、プロキシで APM が有効になっている場合にのみ収集されます（プロキシの設定の *APM* タブを参照）。Zabbix サーバーはテレメトリデータを受信できません。

プロキシの機能

プロキシを使用するかどうかを選択する際には、いくつかの考慮事項を踏まえる必要があります。

	プロキシ
軽量	はい
GUI	いいえ
独立して動作する	はい
保守が容易	はい
DB の自動作成	はい[1]
ローカル管理	いいえ
組み込みハードウェアに対応	はい
一方向の *TCP* 接続	はい
集中設定	はい
通知を生成する	いいえ

[1] DB の自動作成機能は SQLite でのみ動作します。その他の対応データベースでは手動セットアップが必要です。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

過負荷からの保護

Zabbix サーバーがしばらく停止しており、プロキシが大量のデータを収集していた場合、その後サーバーが起動すると、過負荷になることがあります（history cache の使用率がしばらく 95-100% のままになります）。この過負荷により、処理性能が低下し、本来よりもチェックの処理が遅くなる可能性があります。このような状況を防ぎ、history cache の過負荷によって発生する問題を回避するための保護機能が実装されています。

Zabbix サーバーの history cache がいっぱいになると、history cache への書き込みアクセスが制限され、サーバーのデータ収集処理が停止します。history cache 過負荷の最も一般的なケースは、サーバー停止後にプロキシが収集済みデータをアップロードする際に発生します。これを避けるためにプロキシのスロットリングが追加されました（現在は無効化できません）。

history cache の使用率が 80% に達すると、Zabbix サーバーはスロットリングモードに入ります。スロットリングモードでは、history cache の使用率が 60% 未満のときにのみプロキシデータを受け入れ、受け入れるプロキシを順番に切り替えます。history cache の使用率が 20% 未満になると、サーバーは通常モードに戻ります。

さらに、通常モードでは、history cache の使用率が 60% を超えると、Zabbix サーバーは非常に大きなパケット（10,000 レコード以上）を送信する個々のプロキシをスロットリングします。この判定は、サーバーがプロキシのアップロードを評価した瞬間に適用されるため、history cache 使用率のグラフ（内部アイテム `zabbix[wcache,history,pused]` とその更新間隔）には、短時間のスパイクがすぐには反映されない場合があります。

このスロットリングモードは、キャッシュ使用率が再び 80% に達するか、20% まで低下するか、またはスロットリングリストが空になるまで継続します。最初のケースでは、サーバーは再びプロキシデータの受け入れを停止します。他の 2 つのケースでは、サーバーは通常どおり動作し、すべてのプロキシからのデータを受け入れます。

上記の情報は、次の表で示すことができます。

History write cache usage	Zabbix server mode	Zabbix server action
80% に達する	待機	プロキシデータの受け入れを停止するが、後で連絡するための優先順位付きプロキシ一覧（*throttling list*）は保持する。
60% に達する	通常、ただしスロットリングされる準備あり	データを受け入れるかどうかを判断する際に、非常に大きなプロキシアップロード（10k レコード超）を拒否する場合がある。それ以外のプロキシデータは受け入れ続ける。
20% まで低下する	通常	スロットリングリストを破棄し、プロキシデータの受け入れを通常どおり開始する。

この動作をメトリックと関連付けるには、zabbix[wcache,history,pused] 内部アイテムを使用できます。

設定

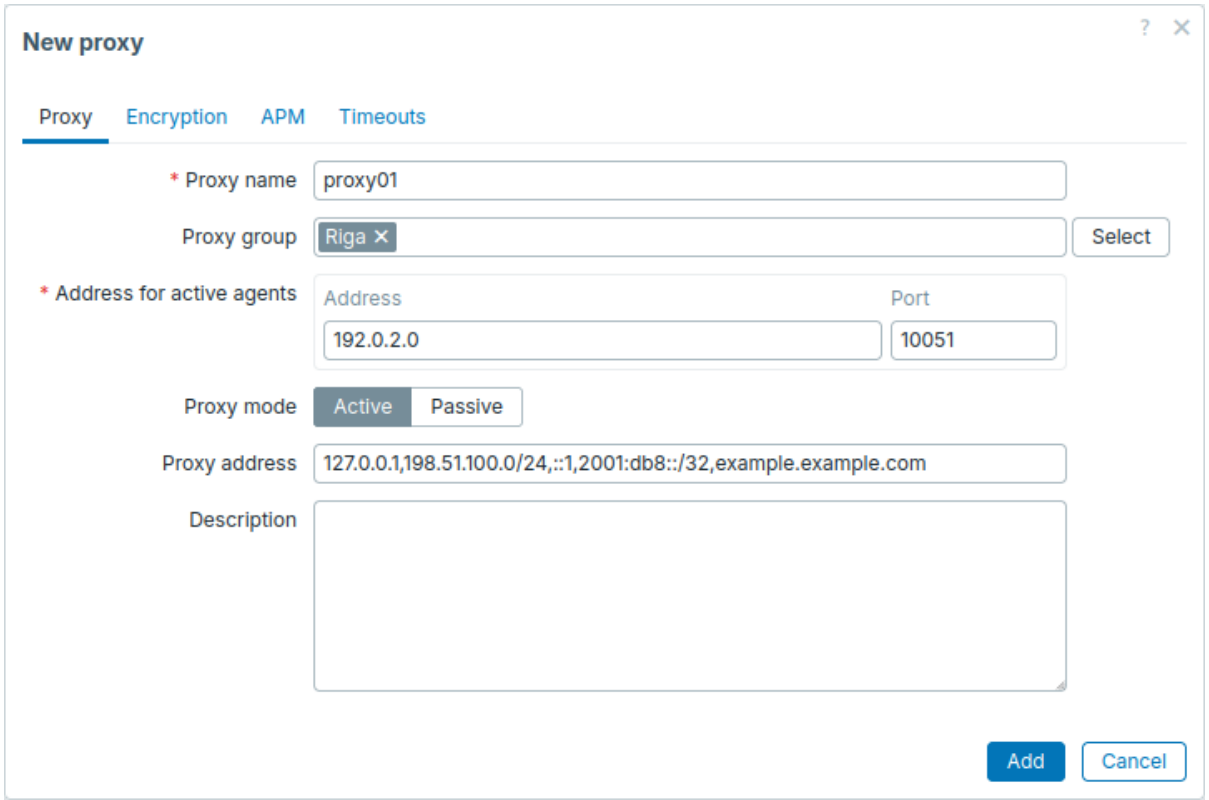
インストールし、設定したプロキシを、Zabbix Web インターフェースで設定します。

プロキシの追加

Note:
独自のインフラストラクチャにプロキシをインストールする代わりに、Zabbix Cloud にプロキシをデプロイできます。デプロイするには、管理 > プロキシで クラウドにプロキシを作成をクリックします。Zabbix Cloud が新しいブラウザータブで開きます。そこでプロキシノードを作成した後、以下の説明に従って Zabbix の Web インターフェースでプロキシを設定します。詳細については、Zabbix Cloud プロキシのセットアップを参照してください。

Zabbix の Web インターフェースでプロキシを設定するには、次の手順に従います。

1. 管理 > プロキシに移動します。
2. プロキシの作成をクリックします。
3. フォームにプロキシのパラメータを入力します。



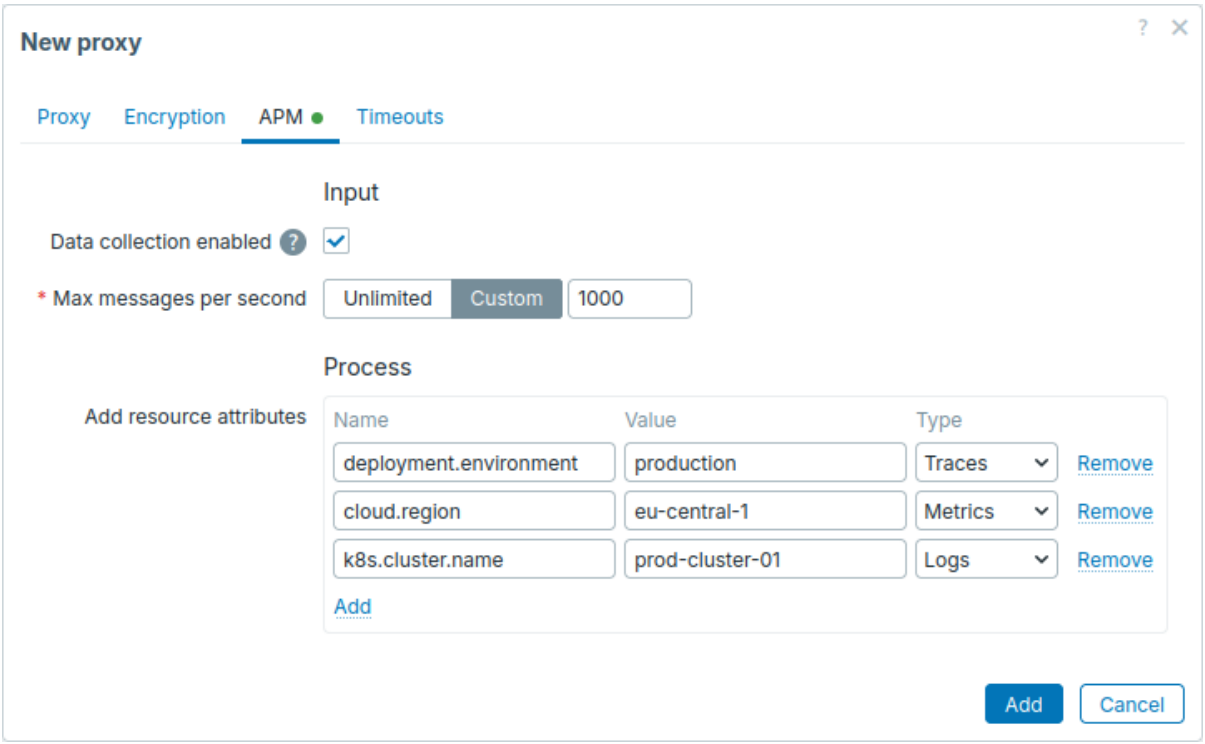
パラメータ		説明
プロキシ名		プロキシ名を入力します。プロキシ設定ファイルの *Hostname* パラメータと同じ名前である必要があります。
プロキシグループ		プロキシの負荷分散/高可用性に使用するプロキシグループを 1 つ選択します。
アクティブエージェント用アドレス		監視対象のアクティブエージェントまたは送信元が接続するアドレスを入力します。**Zabbix 7.0** 以降のエージェントでのみサポートされます。 このアドレスは、アクティブプロキシとパッシブプロキシの両方への接続に使用されます。このフィールドは、プロキシグループフィールドでプロキシグループが選択されている場合にのみ使用できます。
プロキシアドレスインターフェース	アドレス	接続先の IP アドレス/DNS 名。
	ポート	接続先の TCP ポート番号（デフォルトは 10051）。ユーザーマクロを使用できます。
プロキシモード		プロキシモードを選択します。 アクティブ - プロキシが Zabbix サーバーに接続し、設定データを要求します パッシブ - Zabbix サーバーがプロキシに接続します 注：暗号化通信を使用しない場合（機密性）、アクティブプロキシの使用時に、Zabbix サーバーのトラッパーポートにアクセスできる第三者がプロキシ設定データを取得できる可能性があります。認証が行われない場合、または プロキシアドレスフィールドでプロキシアドレスが制限されていない場合、誰でもアクティブプロキシになりすまして設定データを要求できるためです。
プロキシアドレス		指定した場合、アクティブプロキシからの要求は、カンマ区切りの IP アドレス（必要に応じて CIDR 表記）、またはアクティブな Zabbix プロキシの DNS 名で構成されたこのリストからのみ受け付けられます。 このフィールドは、プロキシモードフィールドでアクティブプロキシが選択されている場合にのみ使用できます。マクロはサポートされません。
プロキシアドレスインターフェース		パッシブプロキシのインターフェース情報を入力します。 このフィールドは、プロキシモードフィールドでパッシブプロキシが選択されている場合にのみ使用できます。
	アドレス	パッシブプロキシの IP アドレス/DNS 名。
	ポート	パッシブプロキシの TCP ポート番号（デフォルトは 10051）。ユーザーマクロを使用できます。

パラメータ	説明
説明	プロキシの説明を入力します。

暗号化タブでは、プロキシとの暗号化接続を必須にできます。

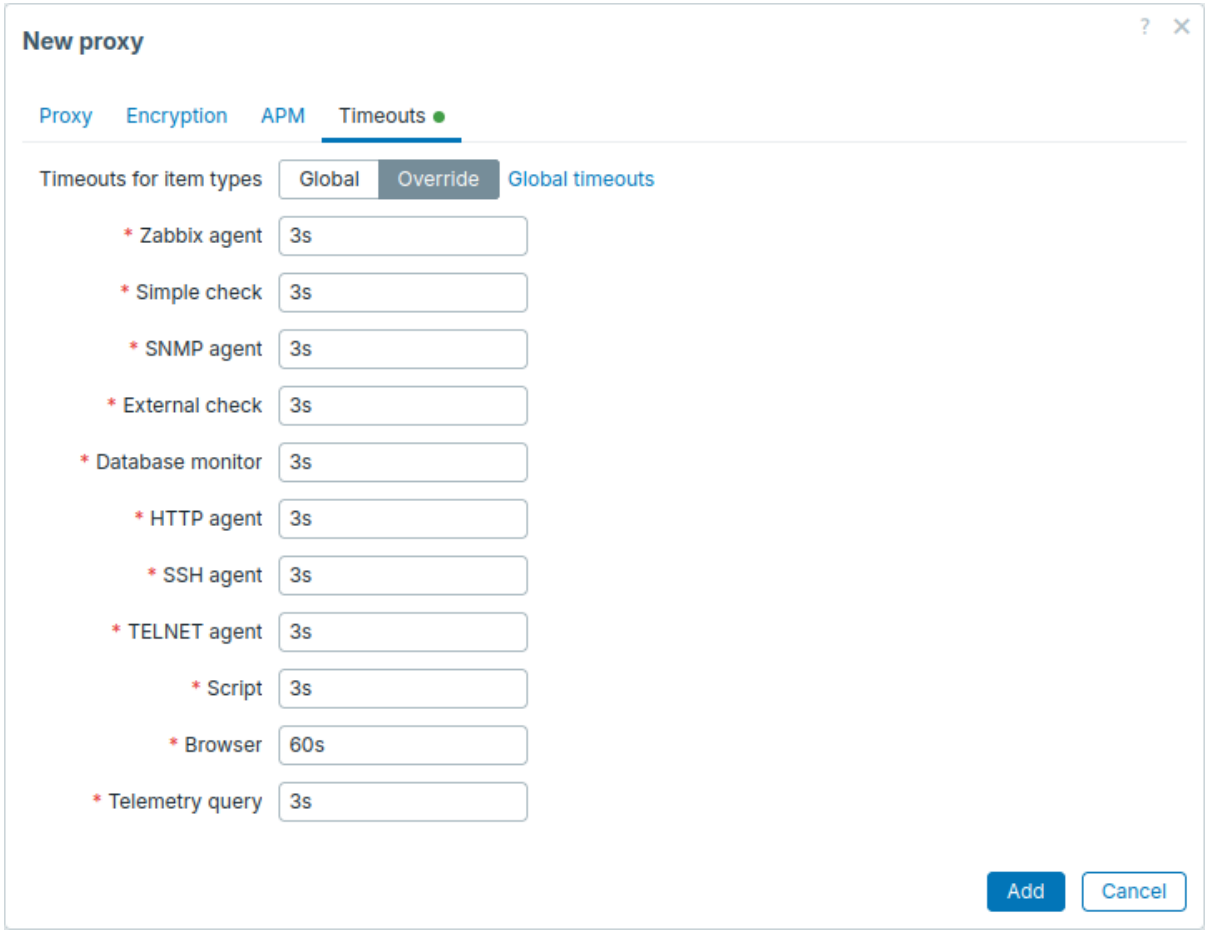
パラメータ	説明
プロキシへの接続	サーバーがパッシブプロキシに接続する方法：暗号化なし（デフォルト）、PSK（事前共有鍵）、または証明書を使用。
プロキシからの接続	アクティブプロキシから許可する接続の種類を選択します。複数の接続タイプを同時に選択できます（テストや別の接続タイプへの切り替えに便利です）。デフォルトは「暗号化なし」です。
発行者	証明書で許可する発行者。証明書はまず CA（認証局）で検証されます。有効で CA によって署名されている場合、発行者フィールドを使用して許可する CA をさらに制限できます。このフィールドは任意であり、Zabbix のインストール環境で複数の CA の証明書を使用する場合に使用します。
サブジェクト	証明書で許可するサブジェクト。証明書はまず CA で検証されます。有効で CA によって署名されている場合、サブジェクトフィールドを使用して、サブジェクト文字列の値を 1 つだけ許可できます。このフィールドが空の場合、設定された CA によって署名された有効な証明書はすべて受け入れられます。
PSK identity	事前共有鍵の識別文字列。 PSK identity には機密情報を含めないでください。使用する PSK を受信側に知らせるため、ネットワーク上で暗号化されずに送信されます。
PSK	事前共有鍵（16 進数文字列）。最大長：Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 桁（256 バイトの PSK）、Zabbix が mbed TLS（PolarSSL）ライブラリを使用する場合は 64 桁（32 バイトの PSK）。例： 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

APM タブでは、プロキシ上でアプリケーションパフォーマンス監視（APM）を設定できます。プロキシは、OpenTelemetry プロトコル（OTLP/gRPC）を介してアプリケーションまたは OpenTelemetry コレクターから送信されたテレメトリデータ（トレース、メトリクス、ログ）を収集し、データをテレメトリデータソースに書き込むことができます。



パラメータ		説明
データ収集を有効にする		プロキシでテレメトリデータの収集を有効にするには、チェックボックスをオンにします。 テレメトリデータの収集を有効にする前に、プロキシ設定ファイルで`StartAPMCollectors`パラメータを 1 に設定します。
1 秒あたりの最大メッセージ数		プロキシが受け入れる 1 秒あたりのメッセージ数の最大値を設定します。 無制限 - 受信するすべてのメッセージを受け入れます。 カスタム - 指定した 1 秒あたりのメッセージ数まで受け入れます。 メッセージとは、アプリケーション、OpenTelemetry コレクター、または別のテレメトリソースから送信される OTLP リクエストです。サイズは送信元によって異なります。たとえば、OpenTelemetry コレクターは数千のスパンを 1 つのメッセージにまとめることがありますが、単純なアプリケーションはより小さなメッセージ（例：トレースごとに 1 つ）を送信する場合があります。 制限を超えると、プロキシは gRPC ステータスコード `UNAVAILABLE`（一時的なエラー）で応答します。そのため、送信元は設定に応じて、後でメッセージの送信を再試行する場合があります。
リソース属性を追加		選択したタイプの受信テレメトリデータに追加するリソース属性を指定します（例：このプロキシが受信したすべてのデータに環境またはリージョンを示す属性を付加する）。これらの属性を使用してテレメトリデータをフィルターできます。たとえば、テレメトリクエリアイテムの条件で使用できます。
	名前	属性名を入力します（例：`deployment.environment`、`cloud.region`）。選択した タイプで一意である必要があります。
	値	属性値を入力します（例：`production`、`eu-central-1`）。空にすることもできます。
	タイプ	属性を適用する受信データのタイプを選択します：トレース、メトリクス、またはログ。

タイムアウトタブでは、対応するアイテムタイプのグローバルタイムアウトを上書きできます。



パラメータ	説明
アイテムタイプのタイムアウト	アイテムのタイムアウトをタイプに基づいて設定します。 グローバル - グローバルタイムアウトを使用します（各アイテムタイプのグレー表示された タイムアウトフィールドに表示されます）。 上書き - カスタムタイムアウトを設定します（各アイテムタイプの タイムアウトフィールド）。許可される範囲：1～600 秒（デフォルト：グローバルタイムアウトから継承）。時間のサフィックス（例：30s、1m）およびユーザーマクロがサポートされます。 グローバルタイムアウトリンクをクリックすると、グローバルタイムアウトを設定できます。グローバルタイムアウトリンクは、管理 > 一般設定 Web インターフェースセクションへの権限を持つスーパー管理者タイプのユーザーにのみ表示されます。 プロキシレベルのタイムアウトはグローバルタイムアウトを上書きしますが、個々のアイテムのタイムアウトが設定されている場合は、個々のアイテムのタイムアウトによって上書きされます。

Note:

プロキシのメジャーバージョンがサーバーのメジャーバージョンと一致しない場合、 アイコンが アイテムタイプのタイムアウトの横に表示され、ホバー時に「プロキシとサーバーのバージョンが一致しないため、タイムアウトが無効です」というメッセージが表示されます。この場合、プロキシはプロキシ設定ファイルの`Timeout` パラメータを使用します。

既存のプロキシの編集フォームには、次の追加ボタンがあります。

- 設定を更新 - プロキシの設定を更新します。
- 複製 - 既存のプロキシのプロパティに基づいて新しいプロキシを作成します。
- 削除 - プロキシを削除します。

ホスト設定

ホスト設定フォームの *Monitored by* フィールドを使用して、個別のホストをプロキシまたはプロキシグループで監視するように指定できます。



ホストの一括更新も、ホストをプロキシまたはプロキシグループで監視するように指定する別の方法です。

プロキシの負荷分散と高可用性

概要

Zabbix プロキシは、プロキシグループにまとめることで、プロキシの負荷分散と高可用性を実現できます。

プロキシの負荷分散と高可用性とは、プロキシグループ内でホストをプロキシ間に自動的に再配置することです。

- プロキシがオフラインになると、そのホストは他のプロキシへ移動され、高いプロキシ可用性が維持されます。
- あるプロキシのホスト数が他のプロキシより大幅に多い、または少ない場合、そのホストは他のプロキシへ移動され、プロキシの負荷が均等化されます。

ホストの再配置は、次の条件を満たすグループ内のプロキシ間でのみ機能します。

- プロキシが Zabbix 7.0 以降で動作していること。
- プロキシのバージョン が Zabbix サーバーのバージョンと一致していること。Zabbix エージェント（パッシブ）を使用している場合は、プロキシのバージョンがエージェントのバージョンと一致している必要があります。アクティブエージェントでは、Zabbix 7.0 以降であることのみが必要です。
- プロキシグループがオンライン状態 であること。
- ホストが個別のプロキシではなく、プロキシグループによって監視対象として設定 されていること。

Note:
プロキシグループの健全性は、プロキシグループに割り当てられた任意のホストから内部チェック で監視できます。ただし、グループ内の単一のプロキシの健全性を監視するには、そのプロキシにホストを割り当ててください。そうしないと、結果が一貫しない場合があります。

ホストの再分配

プロキシの負荷分散と高可用性は、Zabbix サーバーによってproxy group manager を通じて管理されます。proxy group manager は、各プロキシグループ内のすべてのプロキシの状態と、そのホスト分散を継続的に監視します。

グループ内のプロキシの高可用性は、プロキシのフェイルオーバーによって確保されます。プロキシがオフラインになると、そのホストは直ちに他のプロキシへ再分配されます。ホストは、割り当て済みホスト数が最も少ないプロキシへ再割り当てされるため、プロキシの負荷分散も行われます。

さらに、プロキシのホスト数がグループ平均から少なくとも 10 ホスト、かつ 2 倍の差（ホスト過剰またはホスト不足）がある場合にも、プロキシの負荷分散がトリガーされます。猶予期間（10 x failover delay）の後も不均衡が続く場合、プロキシグループはホスト再分配のキューに入れられます。

proxy group manager は、次のロジックでホストを再分配します。

1. プロキシごとの平均ホスト数を計算します。
2. ホスト過剰のプロキシについては、余分なホストを未割り当てのプロキシプールへ移動します。
3. ホスト不足のプロキシについては、バランスを取るために必要なホスト数を計算します。
4. 最も多くのホストを持つプロキシから、必要な数のホストを削除します。
5. 未割り当てのホストを、最も少ないホスト数のプロキシへ移動します。

ホスト再分配の例:

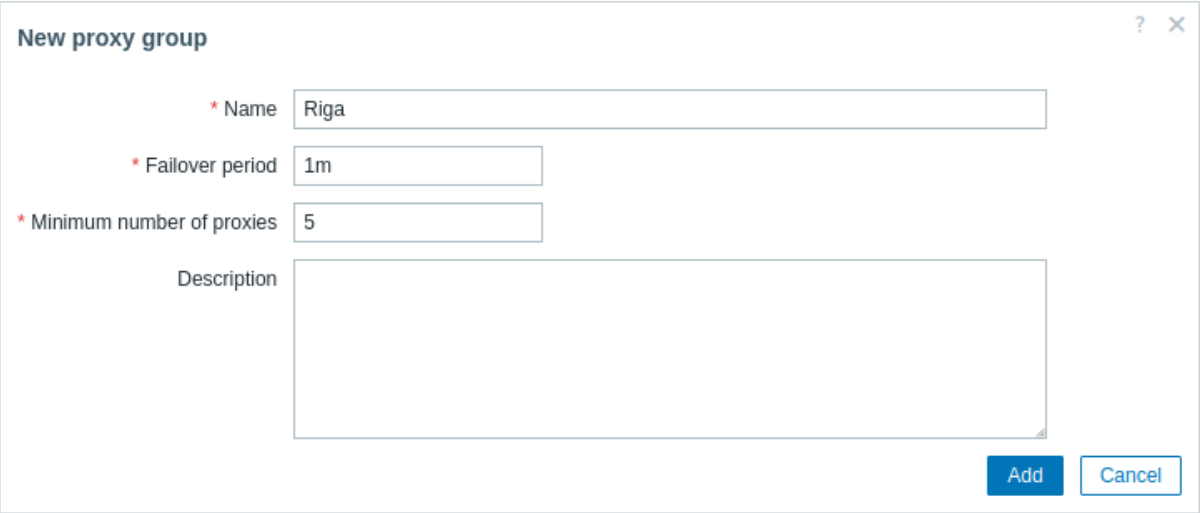
プロキシ上のホスト数	グループ平均	ホストの再割り当て
100	50	Yes
60	50	No
40	50	No
25	50	Yes
15	5	Yes
10	5	No

Attention:
監視対象 のホスト数が 10 未満のプロキシグループでは、グループ内のプロキシ間でホスト分散が不均一になる可能性があります。

プロキシグループの設定

Zabbix の Web インターフェースでプロキシグループを設定するには、次の手順に従います。

1. 管理 > プロキシグループに移動します。
2. プロキシグループの作成をクリックします。
3. フォームにプロキシグループのパラメータを入力します。



パラメータ	説明
名前	プロキシグループの名前。
フェイルオーバー期間	プロキシグループ内のプロキシがオンラインとみなされるために、Zabbix サーバーと通信する必要がある期間（秒単位）（デフォルト: 1m、範囲: 10s-15m）。この期間内にプロキシが通信しない場合、プロキシの状態はオフラインに変更され、そのホストは直ちに他のプロキシに再分配されます。プロキシの負荷分散は、この期間の 10 倍が経過した後に開始されます。 時間のサフィックス（例: 30s、1m）およびユーザーマクロをサポートします。
プロキシの最小数	プロキシグループをオンラインに維持するために必要なオンラインプロキシの最小数（デフォルト: 1、範囲: 1-1000）。 ユーザーマクロをサポートします。 この値は、グループ内のプロキシの総数より少なくする必要があります。たとえば、10 台のプロキシで構成されるグループで最小数を 10 に設定すると、いずれかのプロキシに障害が発生した場合にグループがオフラインになります。オフラインのグループ内にあるオンラインプロキシは通常どおり機能し続けますが、負荷分散や高可用性は機能しません。
説明	プロキシグループの説明。
プロキシ	プロキシを含むグループを編集する際に、最大 5 台のプロキシの一覧を表示します（プロキシに対するユーザー権限に応じて、リンクまたはプレーンテキストで表示されます）。

プロキシ負荷分散の設定

プロキシ負荷分散を使用するには、Zabbix Web インターフェースでプロキシグループを設定し（上記参照）、ホストが個別のプロキシではなくプロキシグループによって監視されるようにする必要があります（ホストをプロキシからプロキシグループへ移動するには、ホストの一括更新を使用できます）。

Zabbix エージェントを使用する場合は、次のようにも設定します。

- パッシブチェックでは、Server パラメーターにプロキシグループ内のすべてのプロキシを列挙します。
- アクティブチェックでは、ServerActive パラメーターにプロキシグループ内のすべてのプロキシ、または Zabbix サーバーを列挙することを推奨します。プロキシグループをアクティブモードで使用できるのは、Zabbix エージェント 7.0（以降）のみです。

ServerActive パラメーターにプロキシグループ内の 1 つのプロキシ（または Zabbix サーバー）だけが含まれている場合でも、エージェントは正しいプロキシに接続できます。エージェントサービスが起動して指定されたプロキシに接続すると、エージェントはプロキシ IP の完全な一覧と、グループ内での現在の負荷を受け取り、キャッシュします。その後、アクティブチェックは、プロキシグループ内の現在のプロキシ-ホスト割り当てに基づいて、ホストに対して正しいオンラインのプロキシへ転送されます。

Warning:
Zabbix エージェントの ServerActive パラメーターに 1 つのプロキシだけを指定すると、指定したプロキシがオフラインの状態でエージェントが起動または再起動された場合に、監視データが失われる可能性があります。

Note:
Zabbix sender を使用する場合も、データ要求はプロキシグループ内の現在のプロキシ-ホスト割り当てに基づいて、ホストに対して正しいオンラインのプロキシへ転送されます。ただし、入力ファイルから複数ホストの値を送信する場合は、誤ったプロキシにデータが送信されないように、-g オプションを使用してください。

Zabbix エージェントは、ファイアウォール経由でプロキシグループ内のすべてのプロキシにも接続できる必要があります。そうでない場合、アクティブチェックは転送やフェイルオーバー中に停止したり、失敗したりする可能性があります。たとえば、次のようなケースです。

- アクティブチェック中に、プロキシがエージェントを別のプロキシへ転送することがあります。そのプロキシがファイアウォールでブロックされていると、応答を待つ間に通信が停止します。
- 最近の再バランスが行われていない安定した高可用性構成では、エージェントがバックアッププロキシに接続しないままになることがあります。ファイアウォールルールが変更されていて検証されていない場合、フェイルオーバーが失敗する可能性があります。

プロキシ負荷分散のテスト

プロキシ負荷分散をテストするには、次の手順を実行します。

1. プロキシグループを設定します。
2. プロキシグループがオンライン状態 であることを確認します。
3. ホストが個別のプロキシではなく、プロキシグループによって監視されている ことを確認します（ホストをプロキシからプロキシグループへ移動するには、ホストの一括更新 を使用できます）。
4. 設定の更新と、プロキシグループ内のプロキシ間でのホストの分散が反映されるまで数秒待ちます。管理 > プロキシ でホスト一覧を更新し、変更を確認します。

重要な注意事項

- SNMP トラップ は、プロキシグループ内のプロキシではサポートされていません。
- 外部設定に依存するチェック（例: 外部チェック 用のスクリプトや、データベースチェック 用の ODBC 設定）は、プロキシグループ内のすべてのプロキシで同じ設定にする必要があります。
- データベースチェック には、データベースオブジェクト/サーバーに対する拡張権限が必要です。
- プロキシグループによって監視される VMware ホストは、グループ内のプロキシ間でランダムに分散されます。そのため、各プロキシがすべての VMware データをキャッシュすることになり、vCenter に追加の負荷がかかります。
- プロキシグループ内のプロキシからの自動登録データに基づいて作成されたホストは、そのプロキシグループによって監視される ように設定されます。ただし、プロキシグループ内のプロキシからのネットワークディスカバリデータに基づいて作成されたホストは、そのプロキシによって監視される ように設定されます。

監視設定の同期

概要

このページでは、プロキシの監視設定更新について詳しく説明します。つまり、サーバー上で監視設定に加えられた変更が、どのようにプロキシへ同期されるかを示します。

増分更新

プロキシ設定の更新は増分方式で行われます。設定の同期中は、変更されたエンティティのみが更新されます（そのため、変更されたエンティティがない場合は何も送信されません）。この方法により、リソースを節約し、プロキシ設定の更新間隔をより短く（ほぼ即時に）設定できます。

プロキシ設定の変更は、リビジョン番号を使用して追跡されます。プロキシ設定のリビジョンより大きいリビジョンを持つエンティティのみが、プロキシに送信される設定データに含まれます。

設定同期の対象となるエンティティは次のとおりです。

Entity	Details
自動登録 *TLS* データ	すべての自動登録 TLS データ。
エクスプレッション	すべてのエクスプレッション（正規表現、エクスプレッションテーブル）。
グローバル設定	'config' テーブルで定義されたグローバル設定。
ホスト	ホストのすべてのプロパティ、インターフェース、インベントリ、アイテム、アイテムの前処理、アイテムパラ
ホストマクロ	ホスト上で定義されたすべてのマクロ、およびそのホストにリンクされたすべてのテンプレート ID。

Entity	Details
プロキシディスカバリルール	プロキシに割り当てられたディスカバリルールおよびチェック。

つまり、次のようになります。

- ホスト上のアイテムが変更された場合、そのホストのすべての設定が同期されます。
- 正規表現が変更された場合、すべての正規表現が同期されます。

例外として、ホスト上で何らかの変更が行われた場合、ホストマクロも送信されます。

プロキシ上で `-R config_cache_reload` コマンドを実行すると、増分更新も開始されます。

プロキシの起動または再起動時、HA フェイルオーバー時、セッショントークンが変更された場合、またはプロキシで設定の更新に失敗した場合（たとえば、設定データの受信中に接続が切断された場合）は、完全な設定同期が実行されます。

設定パラメータ

ProxyConfigFrequency パラメータは、プロキシの設定がサーバーとどのくらいの頻度で同期されるかを決定します（デフォルトは 10 秒です）。

ProxyConfigFrequency は次のとおりです。

- パッシブプロキシ向けのサーバーパラメータ
- アクティブプロキシ向けのプロキシパラメータ

アクティブプロキシでは、ProxyConfigFrequency は Zabbix 6.4 以降の新しいパラメータであり、現在は非推奨となっている ConfigFrequency の代わりに使用する必要があります。

> **Attention:**
> ProxyConfigFrequency と ConfigFrequency の両方が使用されている場合、プロキシはエラーをログに記録して終了します。

メンテナンス中のキューの計算

> **Attention:**
> Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

サーバー-プロキシデータ交換プロトコル

サーバー - プロキシ間のデータ交換は JSON 形式に基づいています。

リクエストおよびレスポンスメッセージは、header and data length で始まる必要があります。

パッシブプロキシ

設定リクエスト

サーバーは最初に空の `proxy config` リクエストを送信します。このリクエストは、`ProxyConfigFrequency`（サーバー設定パラメータ）秒ごとに送信されます。

プロキシは、現在のプロキシバージョン、セッショントークン、設定リビジョンで応答します。サーバーは、更新が必要な設定データで応答します。

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	プロキシバージョン（<major>.<minor>.<build>）。
session	*string*	プロキシ設定のセッショントークン。
config_revision	*number*	プロキシ設定のリビジョン。
server→proxy:		
full_sync	*number*	完全な設定データが送信される場合は 1。それ以外の場合は存在しません
data	*array*	テーブルデータのオブジェクト。設定が変更されていない場合は存在しま
<table>	*object*	<table> データを含む 1 つ以上のオブジェクト（変更内容に応じてオプ
fields	*array*	フィールド名の配列。

name			value type	description
		-	*string*	フィールド名。
	data		*array*	行の配列。
		-	*array*	列の配列。
		-	*string,number*	データベーススキーマの列タイプに応じた型の列値。
macro.secrets			*object*	シークレットマクロの情報。Vault マクロに変更がない場合は存在しませ
config_revision			*number*	設定キャッシュのリビジョン。設定データとともに送信されます（オプ
del_hostids			*array*	削除された hostid の配列（オプション）。
	-		*number*	ホスト識別子。
del_macro_hostids			*array*	すべてのマクロが削除された hostid の配列（オプション）。
	-		*number*	ホスト識別子。
proxy→server:				
response			*string*	リクエストの成功情報（'success' または'failed'）。
version			*string*	プロキシバージョン（<major>.<minor>.<build>）。

例:

サーバー → プロキシ:

```
{
  "request":"proxy config"
}
```

プロキシ → サーバー:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

サーバー → プロキシ:

```
{
    "full_sync": 1,
    "data": {
        "hosts": {
            "fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
```

```
            [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
        ]
    },
    "item_rtdata": {
        "fields": ["itemid", "lastlogsize", "mtime"],
        "data": [
            [44161, 0, 0],
            [44162, 0, 0]
        ]
    },
    "item_preproc": {
        "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
        "data": []
    },
    "item_parameter": {
        "fields": ["item_parameterid", "itemid", "name", "value"],
        "data": []
    },
    "globalmacro": {
        "fields": ["globalmacroid", "macro", "value", "type"],
        "data": [
            [2, "{$SNMP_COMMUNITY}", "public", 0]
        ]
    },
    "hosts_templates": {
        "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
        "data": []
    },
    "hostmacro": {
        "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
        "data": [
            [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
        ]
    },
    "drules": {
        "fields": ["druleid", "name", "iprange", "delay"],
        "data": [
            [2, "Local network", "127.0.0.1", "10s"]
        ]
    },
    "dchecks": {
        "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
        "data": [
            [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
        ]
    },
    "regexps": {
        "fields": ["regexpid", "name"],
        "data": [
            [1, "File systems for discovery"],
            [2, "Network interfaces for discovery"],
            [3, "Storage devices for SNMP discovery"],
            [4, "Windows service names for discovery"],
            [5, "Windows service startup states for discovery"]
        ]
    },
    "expressions": {
        "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
        "data": [
            [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
            [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
            [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
```

```
                [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
                [7, 2, "^Software Loopback Interface", 4, ",", 1],
                [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
                [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
                [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
                [11, 2, "^[Ss]ystem$", 4, ",", 1],
                [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
            ]
        },
        "settings": {
            "fields": ["name", "type", "value_str", "value_int"],
            "data": [
                ["autoreg_tls_accept", 2, "", 1],
                ["hk_history_global", 2, "", 0],
                ["snmptrap_logging", 2, "", 1],
                ["proxy_secrets_provider", 2, "", 0],
                ["hk_history", 1, "31d", 0],
                ["timeout_db_monitor", 1, "3s", 0],
                ["timeout_external_check", 1, "3s", 0],
                ["timeout_http_agent", 1, "3s", 0],
                ["timeout_simple_check", 1, "3s", 0],
                ["timeout_snmp_agent", 1, "3s", 0],
                ["timeout_ssh_agent", 1, "3s", 0],
                ["timeout_telnet_agent", 1, "3s", 0],
                ["timeout_zabbix_agent", 1, "3s", 0],
                ["timeout_browser", 1, "30s", 0]
            ]
        },
        "httptest": {
            "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
            "data": []
        },
        "httptestitem": {
            "fields": ["httptestitemid", "httptestid", "itemid", "type"],
            "data": []
        },
        "httptest_field": {
            "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
            "data": []
        },
        "httpstep": {
            "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
            "data": []
        },
        "httpstepitem": {
            "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
            "data": []
        },
        "httpstep_field": {
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
```

```
        }
    },
    "config_revision": 2
}
```

プロキシ → サーバー:

```
{
  "response": "success",
  "version": "8.0.0"
}
```

データ要求

`proxy data` 要求は、ホストインターフェースの可用性、履歴データ、ディスカバリデータ、および自動登録データをプロキシから取得するために使用されます。この要求は、`ProxyDataFrequency`（サーバー設定パラメータ）秒ごとに送信されます。

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	データセッショントークン。
interface availability	*array*	*(optional)* インターフェース可用性データオブジェクトの配列。
interfaceid	*number*	インターフェース識別子。
available	*number*	インターフェースの可用性: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - 不明 **1**, *INTERFACE_AVAILABLE_TRUE* - 利用可能 **2**, *INTERFACE_AVAILABLE_FALSE* - 利用不可
error	*string*	インターフェースのエラーメッセージ、または空文字列。
history data	*array*	*(optional)* 履歴データオブジェクトの配列。
itemid	*number*	アイテム識別子。
clock	*number*	アイテム値のタイムスタンプ（秒）。
ns	*number*	アイテム値のタイムスタンプ（ナノ秒）。
value	*string*	*(optional)* アイテム値。
id	*number*	値の識別子（昇順のカウンター、1 つのデータセッション内で一意）。
timestamp	*number*	*(optional)* ログタイプアイテムのタイムスタンプ。
source	*string*	*(optional)* Eventlog アイテムの source 値。
severity	*number*	*(optional)* Eventlog アイテムの severity 値。
eventid	*number*	*(optional)* Eventlog アイテムの eventid 値。
state	*string*	*(optional)* アイテムの状態: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* ログタイプアイテムの最終ログサイズ。
mtime	*number*	*(optional)* ログタイプアイテムの更新時刻。
discovery data	*array*	*(optional)* ディスカバリデータオブジェクトの配列。
clock	*number*	ディスカバリデータのタイムスタンプ。
druleid	*number*	ディスカバリルール識別子。
dcheckid	*number*	ディスカバリチェック識別子、またはディスカバリルールデータの場合は null。

name	value type	description
type	*number*	ディスカバリチェックの種類: **-1** ディスカバリルールデータ **0**, *SVC_SSH* - SSH サービスチェック **1**, *SVC_LDAP* - LDAP サービスチェック **2**, *SVC_SMTP* - SMTP サービスチェック **3**, *SVC_FTP* - FTP サービスチェック **4**, *SVC_HTTP* - HTTP サービスチェック **5**, *SVC_POP* - POP サービスチェック **6**, *SVC_NNTP* - NNTP サービスチェック **7**, *SVC_IMAP* - IMAP サービスチェック **8**, *SVC_TCP* - TCP ポート可用性チェック **9**, *SVC_AGENT* - Zabbix エージェント **10**, *SVC_SNMPv1* - SNMPv1 エージェント **11**, *SVC_SNMPv2* - SNMPv2 エージェント **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 エージェント **14**, *SVC_HTTPS* - HTTPS サービスチェック **15**, *SVC_TELNET* - Telnet 可用性チェック
ip	*string*	ホストの IP アドレス。
dns	*string*	ホストの DNS 名。
port	*number*	*(optional)* サービスポート番号。
key_	*string*	*(optional)* 種類 **9** *SVC_AGENT* のディスカバリチェック用アイテムキー
value	*string*	*(optional)* サービスから受信した値。ほとんどのサービスでは空にできます。
status	*number*	*(optional)* サービスの状態: **0**, *DOBJECT_STATUS_UP* - サービス稼働中 **1**, *DOBJECT_STATUS_DOWN* - サービス停止中
auto reg- is- tra- tion	*array*	*(optional)* 自動登録データオブジェクトの配列。
clock	*number*	自動登録データのタイムスタンプ。
host	*string*	ホスト名。
ip	*string*	*(optional)* ホストの IP アドレス。
dns	*string*	*(optional)* IP アドレスから解決された DNS 名。
port	*string*	*(optional)* ホストのポート。
host_metadata	*string*	*(optional)* エージェントが送信したホストメタデータ（エージェント設定パラメータ HostMetadata または HostMetadataItem に基づく）。
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスクの種類: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - リモートコマンド結果
status	*number*	リモートコマンドの実行状態: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - リモートコマンドは正常に完了 **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - リモートコマンドは失敗
error	*string*	*(optional)* エラーメッセージ。
parent_taskid	*number*	親タスク ID。
more	*number*	*(optional)* 1 - 送信すべき履歴データがさらにある。
clock	*number*	*(optional)* データ転送のタイムスタンプ（秒）。
ns	*number*	*(optional)* データ転送のタイムスタンプ（ナノ秒）。
version	*string*	プロキシのバージョン（<major>.<minor>.<build>）。
server→proxy:		
response	*string*	要求成功情報（'success' または'failed'）。
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスクの種類: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - リモートコマンド

name	value type	description
clock	*number*	タスク作成時刻。
ttl	*number*	タスクの有効期限（秒）。
commandtype	*number*	リモートコマンドの種類: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - カスタムスクリプトを使用 **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI を使用 **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH を使用 **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet を使用 **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - グローバルスクリプトを使用（現在は機能的にカスタムスクリプトと同等）
command	*string*	実行するリモートコマンド。
execute_on	*number*	カスタムスクリプトの実行先: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - エージェント上でスクリプトを実行 **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - サーバー上でスクリプトを実行 **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - プロキシ上でスクリプトを実行
port	*number*	*(optional)* Telnet および SSH コマンド用のポート。
authtype	*number*	*(optional)* SSH コマンドの認証方式。
username	*string*	*(optional)* Telnet および SSH コマンドのユーザー名。
password	*string*	*(optional)* Telnet および SSH コマンドのパスワード。
publickey	*string*	*(optional)* SSH コマンドの公開鍵。
privatekey	*string*	*(optional)* SSH コマンドの秘密鍵。
parent_taskid	*number*	親タスク ID。
hostid	*number*	対象ホスト ID。

例:

server→proxy:

```
{
  "request": "proxy data"
}
```

proxy→server:

```
{
    "session": "12345678901234567890123456789012"
    "interface availability": [
        {
            "interfaceid": 1,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 2,
            "available": 2,
            "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection
    },
        {
            "interfaceid": 3,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 4,
            "available": 1,
            "error": ""
    }
    ],
    "history data":[
        {
            "itemid":"12345",
```

```
            "clock":1478609647,
            "ns":332510044,
            "value":"52956612",
            "id": 1
        },
        {
            "itemid":"12346",
            "clock":1478609647,
            "ns":330690279,
            "state":1,
            "value":"Cannot find information for this network interface in /proc/net/dev.",
            "id": 2
        }
    ],
    "discovery data":[
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":3,
            "type":12,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        },
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":null,
            "type":-1,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        }
    ],
    "auto registration":[
        {
            "clock":1478608371,
            "host":"Logger1",
            "ip":"10.3.0.1",
            "dns":"localhost",
            "port":"10050"
        },
        {
            "clock":1478608381,
            "host":"Logger2",
            "ip":"10.3.0.2",
            "dns":"localhost",
            "port":"10050"
        }
    ],
    "tasks":[
        {
            "type": 0,
            "status": 0,
            "parent_taskid": 10
        },
        {
            "type": 0,
            "status": 1,
            "error": "No permissions to execute task.",
            "parent_taskid": 20
        }
```

```
    ],
    "version":"8.0.0"
}
```

server→proxy:

```
{
  "response": "success",
  "tasks":[
      {
         "type": 1,
         "clock": 1478608371,
         "ttl": 600,
         "commandtype": 2,
         "command": "restart_service1.sh",
         "execute_on": 2,
         "port": 80,
         "authtype": 0,
         "username": "userA",
         "password": "password1",
         "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqGKukO1De7zhZj6+H0qtjTkVxwTCpvKe",
         "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u09lHNsj6tQ5QCqGKukO1De7zhd",
         "parent_taskid": 10,
         "hostid": 10070
      },
      {
         "type": 1,
         "clock": 1478608381,
         "ttl": 600,
         "commandtype": 1,
         "command": "restart_service2.sh",
         "execute_on": 0,
         "authtype": 0,
         "username": "",
         "password": "",
         "publickey": "",
         "privatekey": "",
         "parent_taskid": 20,
         "hostid": 10084
      }
  ]
}
```

タスク要求

`proxy tasks` の交換は、サーバーとパッシブプロキシ間でリモートコマンドタスクの配信と受領確認を行います。サーバーがタスク結果を取得するためにプロキシをポーリングすると、`ProxyDataFrequency` で定義された間隔で空の `proxy tasks` 要求を送信します。プロキシは、現在のバージョンと保留中のタスク結果（タイムスタンプを含む）を返します。サーバーはその後、`response` を返して受信を確認し、プロキシが実行する新しいタスクを含めることもできます。

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	プロキシのバージョン (<major>.<minor>.<build>).
clock	*number*	*(optional)* データ転送タイムスタンプ（秒）。
ns	*number*	*(optional)* データ転送タイムスタンプ（ナノ秒）。
server→proxy:		
response	*string*	要求成功情報（'success' または'failed'）。

例:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

アクティブプロキシ

設定リクエスト

`proxy config` リクエストは、アクティブプロキシがプロキシ設定データを取得するために送信します。このリクエストは、`ProxyConfigFrequency`（プロキシ設定パラメータ）秒ごとに送信されます。

name				value type	description
プロキシ → サーバー:					
request				*string*	'proxy config'
host				*string *	プロキシ名。
version				*string*	プロキシのバージョン（<major>.<minor>.<build>）。
session				*string*	プロキシ設定セッショントークン。
config_revision				*number*	プロキシ設定のリビジョン。
hostmap_revision				*number*	現在のホストとプロキシのマッピングリビジョン。
サーバー → プロキシ:					
full_sync				*number*	完全な設定データが送信される場合は 1。それ以外の場合は存在しません
data				*array*	テーブルデータのオブジェクト。設定が変更されていない場合は存在し
	<table>			*object*	<table> データを含む 1 つ以上のオブジェクト（変更内容に応じてオプ
		fields		*array*	フィールド名の配列。
			-	*string*	フィールド名。
		data		*array*	行の配列。
			-	*array*	列の配列。
			-	*string,number*	データベーススキーマの列タイプに応じた型の列値。
macro.secrets				*object*	秘密マクロの情報。Vault マクロに変更がない場合は存在しません（オプ
proxy_group				*string*	プロキシが所属するプロキシグループの名前。
config_revision				*number*	設定キャッシュのリビジョン。設定データとともに送信されます（オプ
del_hostids				*array*	削除された hostid の配列（オプション）。
	-			*number*	ホスト識別子。
del_macro_hostids				*array*	すべてのマクロが削除された hostid の配列（オプション）。
	-			*number*	ホスト識別子。

例:

プロキシ → サーバー:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

サーバー → プロキシ:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
```

```
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
```

```
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


データ要求

`proxy data` 要求は、ホストインターフェースの可用性、履歴、ディスカバリ、および自動登録データを提供するためにプロキシから送信されます。この要求は `DataSenderFrequency`（プロキシ設定パラメータ）秒ごとに送信されます。アクティブプロキシは、リモートコマンドタスクのために、引き続き毎秒 Zabbix サーバーをポーリングします（空の `proxy data` 要求を使用）。

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	プロキシ名。
session	*string*	データセッショントークン。
interface	*array*	*(optional)* インターフェース可用性データオブジェクトの配列。
avail-abil-ity		
interfaceid	*number*	インターフェース識別子。
available	*number*	インターフェースの可用性: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - 不明 **1**, *INTERFACE_AVAILABLE_TRUE* - 利用可能 **2**, *INTERFACE_AVAILABLE_FALSE* - 利用不可
error	*string*	インターフェースのエラーメッセージ、または空文字列。
history	*array*	*(optional)* 履歴データオブジェクトの配列。
data		
itemid	*number*	アイテム識別子。
clock	*number*	アイテム値のタイムスタンプ（秒）。
ns	*number*	アイテム値のタイムスタンプ（ナノ秒）。
value	*string*	*(optional)* アイテム値。
id	*number*	値の識別子（昇順カウンター、1 つのデータセッション内で一意）。

name	value type	description
timestamp	*number*	*(optional)* ログタイプアイテムのタイムスタンプ。
source	*string*	*(optional)* Eventlog アイテムの source 値。
severity	*number*	*(optional)* Eventlog アイテムの severity 値。
eventid	*number*	*(optional)* Eventlog アイテムの eventid 値。
state	*string*	*(optional)* アイテム状態: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* ログタイプアイテムの最終ログサイズ。
mtime	*number*	*(optional)* ログタイプアイテムの更新時刻。
discovery	*array*	*(optional)* ディスカバリデータオブジェクトの配列。
data		
clock	*number*	ディスカバリデータのタイムスタンプ。
druleid	*number*	ディスカバリルール識別子。
dcheckid	*number*	ディスカバリチェック識別子、またはディスカバリルールデータの場合は null。
type	*number*	ディスカバリチェックの種類: **-1** ディスカバリルールデータ **0**, *SVC_SSH* - SSH サービスチェック **1**, *SVC_LDAP* - LDAP サービスチェック **2**, *SVC_SMTP* - SMTP サービスチェック **3**, *SVC_FTP* - FTP サービスチェック **4**, *SVC_HTTP* - HTTP サービスチェック **5**, *SVC_POP* - POP サービスチェック **6**, *SVC_NNTP* - NNTP サービスチェック **7**, *SVC_IMAP* - IMAP サービスチェック **8**, *SVC_TCP* - TCP ポート可用性チェック **9**, *SVC_AGENT* - Zabbix エージェント **10**, *SVC_SNMPv1* - SNMPv1 エージェント **11**, *SVC_SNMPv2* - SNMPv2 エージェント **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 エージェント **14**, *SVC_HTTPS* - HTTPS サービスチェック **15**, *SVC_TELNET* - Telnet 可用性チェック
ip	*string*	ホストの IP アドレス。
dns	*string*	ホストの DNS 名。
port	*number*	*(optional)* サービスポート番号。
key_	*string*	*(optional)* 種類 **9** *SVC_AGENT* のディスカバリチェック用アイテムキー
value	*string*	*(optional)* サービスから受信した値。多くのサービスでは空にできます。
status	*number*	*(optional)* サービス状態: **0**, *DOBJECT_STATUS_UP* - サービス稼働中 **1**, *DOBJECT_STATUS_DOWN* - サービス停止中
autoregistration	*array*	*(optional)* 自動登録データオブジェクトの配列。
clock	*number*	自動登録データのタイムスタンプ。
host	*string*	ホスト名。
ip	*string*	*(optional)* ホストの IP アドレス。
dns	*string*	*(optional)* IP アドレスから解決された DNS 名。
port	*string*	*(optional)* ホストのポート。
host_metadata	*string*	*(optional)* エージェントが送信したホストメタデータ（HostMetadata または HostMetadataItem エージェント設定パラメータに基づく）。
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスク種別: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - リモートコマンド結果
status	*number*	リモートコマンド実行ステータス: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - リモートコマンドは正常に完了 **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - リモートコマンドは失敗

name	value type	description
error	*string*	*(optional)* エラーメッセージ。
parent_taskid	*number*	親タスク ID。
more	*number*	*(optional)* 1 - 送信する履歴データがさらにある
clock	*number*	*(optional)* データ転送タイムスタンプ（秒）。
ns	*number*	*(optional)* データ転送タイムスタンプ（ナノ秒）。
version	*string*	プロキシバージョン（<major>.<minor>.<build>）。
server→proxy:		
response	*string*	要求成功情報（'success' または'failed'）。
upload	*string*	履歴データ（history、自動登録、ホスト可用性、ネットワークディスカバリ）のアップロード制御。 可能な値: **enabled** - 通常動作 **disabled** - サーバーがデータを受け付けていない（内部キャッシュの上限超過が原因の可能性あり）
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスク種別: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - リモートコマンド
clock	*number*	タスク作成時刻。
ttl	*number*	タスクが期限切れになるまでの秒数。
commandtype	*number*	リモートコマンドの種類: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - カスタムスクリプトを使用 **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI を使用 **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH を使用 **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet を使用 **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - グローバルスクリプトを使用（現在は機能的にカスタムスクリプトと同等）
command	*string*	実行するリモートコマンド。
execute_on	*number*	カスタムスクリプトの実行先: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - エージェント上でスクリプトを実行 **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - サーバー上でスクリプトを実行 **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - プロキシ上でスクリプトを実行
port	*number*	*(optional)* Telnet および SSH コマンド用のポート。
authtype	*number*	*(optional)* SSH コマンドの認証タイプ。
username	*string*	*(optional)* Telnet および SSH コマンドのユーザー名。
password	*string*	*(optional)* Telnet および SSH コマンドのパスワード。
publickey	*string*	*(optional)* SSH コマンドの公開鍵。
privatekey	*string*	*(optional)* SSH コマンドの秘密鍵。
parent_taskid	*number*	親タスク ID。
hostid	*number*	対象ホスト ID。

Example:

proxy→server:


```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
```

```
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
"ip": "127.0.0.1",
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "8.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
```

```
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
}]
}
```

設定パラメーター: **Zabbix** プロキシ

概要

設定パラメータを使用すると、Zabbix プロキシの動作をカスタマイズできます。

パラメータ値は、Zabbix のプロキシ設定ファイル（zabbix_proxy.conf）で指定します。パラメータが必須であることが明記されていない限り、すべてのパラメータは任意です。

このページは次の構成になっています。

- パラメータの概要（パラメータの詳細を表示するには、その名前をクリックします）
- パラメータの詳細

パラメータ	説明
AllowRoot	プロキシを root として実行できるようにします。
AllowUnsupportedDBVersions	サポート対象外のデータベースバージョンでプロキシを動作させます。
APMListenIP	APM コレクターが受信テレメトリデータを待ち受ける IP アドレスを指定します。
APMListenPort	APM コレクターが受信テレメトリデータを待ち受けるポートを指定します。
APMTLSCAFile	APM コレクターにデータを送信するアプリケーションまたは OpenTelemetry コレクターの証明書を検証するための、最上位 CA 証明書を含むファイルの完全なパス名を指定します。
APMTLSCertFile	APM コレクターの証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
APMTLSKeyFile	APM コレクターの秘密鍵を含むファイルの完全なパス名を指定します。
CacheSize	設定キャッシュのサイズを設定します。
ConfigFrequency	このパラメータは非推奨です（代わりに ProxyConfigFrequency を使用してください）。 Zabbix サーバーから設定データを取得する頻度を設定します。
DataSenderFrequency	収集したデータを Zabbix サーバーに送信する頻度を設定します。
DBHost	データベースのホスト（またはソケットディレクトリ）を指定します。
DBName	SQLite3 のデータベース名またはデータベースファイルへのパスを指定します。
DBPassword	データベースのパスワードを指定します。
DBPort	Zabbix データベースホストのポートを指定します。
DBSchema	データベーススキーマ名を指定します。PostgreSQL で使用されます。
DBSocket	MySQL ソケットファイルへのパスを指定します。
DBUser	データベースユーザーを指定します。
DBTLSConnect	データベースへの TLS 接続を強制する値を指定します。
DBTLSCAFile	データベース証明書の検証に使用する最上位 CA 証明書を含むファイルの完全なパス名を指定します。
DBTLSCertFile	データベースへの認証に使用する Zabbix プロキシの証明書を含むファイルの完全なパス名を指定します。
DBTLSKeyFile	データベースへの認証に使用する秘密鍵を含むファイルの完全なパス名を指定します。
DBTLSCipher	TLS v1.2 までの TLS プロトコルで Zabbix プロキシが許可する暗号化暗号のリストを指定します。 MySQL でのみサポートされます。
DBTLSCipher13	TLS v1.3 プロトコルで Zabbix プロキシが許可する暗号スイートのリストを指定します。MySQL でのみ、バージョン 8.0.16 以降でサポートされます。
DebugLevel	デバッグレベルを設定します。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可します。
ExternalScripts	外部スクリプトの場所を指定します。
Fping6Location	fping6 の場所を指定します。
FpingLocation	fping の場所を指定します。
HistoryCacheSize	履歴キャッシュのサイズを設定します。
HistoryIndexCacheSize	履歴インデックスキャッシュのサイズを設定します。
Hostname	プロキシ名を指定します。

パラメータ	説明
HostnameItem	Hostname の設定に使用するアイテムを指定します（未定義の場合）。
HousekeepingFrequency	ハウスキーピング処理の頻度を設定します（時間単位）。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
JavaGateway	Zabbix Java ゲートウェイの IP アドレス（またはホスト名）を指定します。
JavaGatewayPort	Zabbix Java ゲートウェイのポートを指定します。
ListenBacklog	TCP キューで保留できる接続数の最大値を設定します。
ListenIP	トラッパーが待ち受ける IP アドレスを指定します。
ListenPort	トラッパーが待ち受けるポートを指定します。
LoadModule	プロキシの起動時にロードするモジュールを指定します。
LoadModulePath	プロキシモジュールの場所への完全なパスを指定します。
LogFile	プロキシのログファイルを指定します。
LogFileSize	ログファイルサイズの最大値を設定します。
LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録します。
LogSlowQueries	ログに記録するまでのデータベースクエリの最大実行時間を設定します（ミリ秒単位）。
LogType	ログの出力タイプを指定します。
MaxConcurrentChecksPerPoller	各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが一度に実行できる非同期チェック数の最大値を設定します。
PidFile	PID ファイルを指定します。
ProxyBufferMode	履歴、ディスカバリ、自動登録データの保存方式（disk/memory/hybrid）を指定します。
ProxyConfigFrequency	Zabbix サーバーから設定データを取得する頻度を秒単位で設定します。
ProxyLocalBuffer	データがすでにサーバーと同期されている場合でも、データをローカルに保持する期間を設定します（時間単位）。
ProxyMemoryBufferAge	プロキシのメモリバッファー内のデータの最大経過時間を秒単位で設定します。
ProxyMemoryBufferSize	収集した履歴、ディスカバリ、自動登録データ用の共有メモリキャッシュサイズを設定します。
ProxyMode	プロキシの動作モード（active/passive）を設定します。
ProxyOfflineBuffer	Zabbix サーバーに接続できない場合にデータを保持する期間を設定します（時間単位）。
Server	ProxyMode が `active` に設定されている場合：設定データの取得元およびデータの送信先となる Zabbix サーバーの IP アドレスまたは DNS 名（address:port）、またはクラスター（address:port;address2:port）を指定します。 ProxyMode が `passive` に設定されている場合：Zabbix サーバーの IP アドレスまたは DNS 名を指定します。
SNMPTrapperFile	SNMP トラップデーモンからプロキシにデータを渡すために使用する一時ファイルを指定します。
SocketDir	内部 Zabbix サービスが使用する IPC ソケットを保存するディレクトリを指定します。
SourceIP	送信元 IP アドレスを指定します。
SSHKeyLocation	SSH チェックおよびアクション用の公開鍵と秘密鍵の場所を指定します。
SSLCertLocation	クライアント認証用の SSL クライアント証明書ファイルの場所を指定します。
SSLKeyLocation	クライアント認証用の SSL 秘密鍵ファイルの場所を指定します。
SSLCALocation	SSL サーバー証明書の検証に使用する認証局（CA）ファイルの場所を指定します。
StartAgentPollers	フォーク前に生成される非同期 Zabbix エージェントポーラーのインスタンス数を設定します。
StartAPMCollectors	APM コレクターを起動するには 1 に設定します。
StartBrowserPollers	フォーク前に生成されるブラウザーアイテムポーラーのインスタンス数を設定します。
StartDBSyncers	フォーク前に生成される履歴同期処理のインスタンス数を設定します。
StartDiscoverers	フォーク前に生成されるディスカバリワーカーのインスタンス数を設定します。
StartHTTPAgentPollers	フォーク前に生成される非同期 HTTP エージェントポーラーのインスタンス数を設定します。
StartHTTPPollers	フォーク前に生成される HTTP ポーラーのインスタンス数を設定します。
StartIPMIPollers	フォーク前に生成される IPMI ポーラーのインスタンス数を設定します。
StartJavaPollers	フォーク前に生成される Java ポーラーのインスタンス数を設定します。
StartODBCPollers	フォーク前に生成される ODBC ポーラーのインスタンス数を設定します。
StartPingers	フォーク前に生成される ICMP pinger のインスタンス数を設定します。
StartPollersUnreachable	到達不能なホスト（IPMI および Java を含む）用のポーラーの、フォーク前に生成されるインスタンス数を設定します。
StartPollers	フォーク前に生成されるポーラーのインスタンス数を設定します。
StartPreprocessors	事前起動される前処理ワーカーのインスタンス数を設定します。
StartSNMPPollers	フォーク前に生成される非同期 SNMP ポーラーのインスタンス数を設定します。
StartSNMPTrapper	SNMP トラッパープロセスを起動するには”1” に設定します。
StartTelemetryQueryPollers	フォーク前に生成される非同期テレメトリクエリポーラーのインスタンス数を設定します。
StartTrappers	フォーク前に生成されるトラッパーのインスタンス数を設定します。
StartVMwareCollectors	フォーク前に生成される VMware コレクターのインスタンス数を設定します。
StatsAllowedIP	統計情報のリクエストを許可する外部 Zabbix インスタンスの IP アドレスまたは DNS 名を指定します。
TelemetryProvider	テレメトリデータのソースを設定します。

パラメータ	説明
Timeout	Zabbix プロキシ、エージェント、Web サービスとの接続確立およびデータ交換、ならびに SNMP チェック（SNMP `walk[OID]` および `get[OID]` アイテムを除く）のために待機する時間を秒単位で指定します。
TLSAccept	Zabbix サーバーからの受信接続に必要な暗号化レベルを指定します。
TLSCAFile	Zabbix コンポーネント間の暗号化通信で使用する、ピア証明書の検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
TLSCertFile	Zabbix コンポーネント間の暗号化通信で使用する、サーバー証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
TLSCipherAll	GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS の優先度文字列または OpenSSL（TLS 1.2）の暗号文字列を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	Zabbix サーバーへの接続に必要な暗号化レベルを指定します。
TLSCRLFile	失効した証明書を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	Zabbix コンポーネント間の暗号化通信で使用する、プロキシの秘密鍵を含むファイルの完全なパス名を指定します。
TLSListen	トラッパーソケットで TLS のみの接続を要求します。
TLSPSKFile	Zabbix サーバーとの暗号化通信で使用する、プロキシの事前共有鍵を含むファイルの完全なパス名を指定します。
TLSPSKIdentity	Zabbix サーバーとの暗号化通信で使用する、事前共有鍵の識別文字列を指定します。
TLSServerCertIssuer	許可するサーバー証明書の発行者を指定します。
TLSServerCertSubject	許可するサーバー証明書のサブジェクトを指定します。
TmpDir	一時ディレクトリを指定します。
TrapperTimeout	次の処理のタイムアウトを秒単位で指定します。 - Zabbix サーバーからの設定データの取得; - グローバルスクリプトの実行またはリモートコマンドの実行。
UnavailableDelay	利用不可期間中にホストの可用性を確認する頻度を設定します。
UnreachableDelay	到達不能期間中にホストの可用性を確認する頻度を設定します。
UnreachablePeriod	到達不能になってから何秒後にホストを利用不可として扱うかを設定します。
User	権限を、システム上に存在する特定のユーザーに降格します。
Vault	Vault プロバイダーを指定します。
VaultAppRoleID	HashiCorp Vault の AppRole ID を指定します。
VaultAppSecretID	HashiCorp Vault の AppRole シークレット ID を指定します。
VaultDBPath	キーによってデータベース認証情報を取得する場所を指定します。
VaultPrefix	Vault パスまたはクエリのカスタムプレフィックスを指定します。
VaultTLSCertFile	クライアント認証に使用する SSL 証明書ファイルの名前を指定します。
VaultTLSKeyFile	クライアント認証に使用する SSL 秘密鍵ファイルの名前を指定します。
VaultToken	HashiCorp Vault の認証トークンを指定します。
VaultURL	Vault サーバーの URL を指定します。
VMwareCacheSize	VMware データの保存に使用する共有メモリサイズを設定します。
VMwareFrequency	単一の VMware サービスからデータを収集する頻度を秒単位で設定します。
VMwarePerfFrequency	単一の VMware サービスからパフォーマンスカウンター統計を取得する頻度を秒単位で設定します。
VMwareTimeout	`vmware collector` が VMware サービスからの応答を待機する最大秒数を設定します。
WebDriverURL	WebDriver インターフェースの URL を指定します。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、デーモンのデフォルト値を示します。
- 値では環境変数を使用できます。
- Zabbix は、BOMなしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみサポートされます。

パラメーターの詳細

AllowRoot

プロキシが root として実行されることを許可します。無効にされていて、プロキシが root によって起動された場合、プロキシは代わりに zabbix ユーザーへ切り替えようとします。通常ユーザーで起動された場合は、影響はありません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

AllowUnsupportedDBVersions

プロキシがサポート対象外のデータベースバージョンで動作することを許可します。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

APMListenIP

APM コレクターが受信テレメトリデータをリッスンする IP アドレスを指定します。
 指定しない場合、APM コレクターはすべてのネットワークインターフェースをリッスンします。

デフォルト: 0.0.0.0

APMListenPort

APM コレクターが受信テレメトリデータをリッスンするポートを指定します。
 デフォルトのポートは、OpenTelemetry SDK およびコレクターがデフォルトで使用する標準の OpenTelemetry（OTLP/gRPC）ポートです。

デフォルト: 4317
 範囲: 1024-32767

APMTLSCAFile

APM collector にデータを送信するアプリケーションまたは OpenTelemetry コレクターの証明書を検証するための、最上位 CA 証明書を含むファイルの完全なパス名を指定します。
 このパラメータを設定すると、APM コレクターは暗号化された接続のみを受け入れます。設定しない場合は、暗号化されていない接続のみを受け入れます。

APMTLSCertFile

APM コレクターの証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
 このパラメータは、APMTLSKeyFileとともに設定する必要があり、APMTLSCAFileが設定されている場合にのみ設定できます。

APMTLSKeyFile

APM collector の秘密鍵を含むファイルの完全なパス名を指定します。
 このパラメータは、APMTLSCertFile とともに設定する必要があり、APMTLSCAFile が設定されている場合にのみ設定できます。

CacheSize

設定キャッシュサイズをバイト単位で指定します。ホストとアイテムのデータを保存するための共有メモリサイズです。

デフォルト: 32M
 範囲: 128K-64G

ConfigFrequency

このパラメーターは非推奨です（代わりにProxyConfigFrequencyを使用してください）。
Zabbix サーバーから設定データを取得する頻度を秒単位で設定します。
 アクティブプロキシのパラメーターです。パッシブプロキシでは無視されます(ProxyModeパラメーターを参照)。

デフォルト: 3600
 範囲: 1-604800

DataSenderFrequency

プロキシから Zabbix サーバーへ収集したデータを送信する頻度を設定します。アクティブプロキシは、リモートコマンドタスクのために引き続き 1 秒ごとに Zabbix サーバーをポーリングすることに注意してください。
 アクティブプロキシのパラメーターです。パッシブプロキシでは無視されます（ProxyMode パラメーターを参照）。

デフォルト: 1
 範囲: 1-3600

DBHost

データベースホスト（またはソケットディレクトリ）を指定します。

MySQL の場合:

- localhost または空文字列は、デフォルトの UNIX ドメインソケットを使用します。

PostgreSQL の場合:

- localhost は DNS 経由で解決されます（通常は 127.0.0.1）。
- 空文字列は、デフォルトの UNIX ドメインソケットを使用します。

- パス（たとえば /var/run/pgbouncer）を指定すると、そのパス上の UNIX ソケットを使用します。
- カンマ区切りのリストには、複数の host:port 値を含めることができます。各ホストは、読み書き可能な接続が確立されるまで順番に試行されます。
  例: DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]

デフォルト: localhost

DBName

SQLite3 のデータベース名、またはデータベースファイルへのパスを指定します（Zabbix のマルチプロセスアーキテクチャでは、インメモリデータベース（例: :memory:、file::memory:?cache=shared、file:memdb1?mode=memory&cache=shared）は使用できません）。
 警告: Zabbix サーバーが使用しているデータベースと同じデータベースを使用しないでください。

必須: はい

DBPassword

データベースのパスワードを指定します。パスワードを使用しない場合は、この行をコメントアウトしてください。SQLite では無視されます。

DBPort

Zabbix データベースホストのポートを指定します（DBHostを参照）。SQLite では無視されます。
 プロキシ設定では、DBSocket と DBPort は排他的です。どちらか一方のみを指定するか、両方とも未定義のままにしてください。

MySQL のデフォルト: 3306

PostgreSQL のデフォルト: 5432

範囲: 1024-65535

DBSchema

データベーススキーマ名を指定します。PostgreSQL で使用されます。

DBSocket

MySQL ソケットファイルへのパスを指定します。
DBSocket と DBPort は、プロキシ設定において排他的です。どちらか一方のみを指定するか、両方とも未定義のままにしてください。

DBUser

データベースユーザーを指定します。SQLite では無視されます。

DBTLSConnect

データベースへの TLS 接続を強制する値を指定します: - *required* - TLS を使用して接続します; - *verify_ca* - TLS を使用して接続し、証明書を検証します; - *verify_full* - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース識別子がその証明書と一致することを確認します
MySQL 5.7.11 以降および PostgreSQL では、次の値がサポートされています: ”required”, ”verify”, ”verify_full”.
MariaDB 10.2.6 以降では”required” と”verify_full” の値がサポートされています.
 デフォルトではいずれのオプションにも設定されず、動作はデータベースの設定に依存します。

DBTLSCAFile

データベース証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。

必須: いいえ（DBTLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須）

DBTLSCertFile

データベースへの認証に使用する Zabbix プロキシ証明書を含むファイルのフルパスを指定します。

DBTLSKeyFile

データベースへの認証に使用する秘密鍵を含むファイルの完全なパス名を指定します。

DBTLSCipher

Zabbix プロキシが TLS プロトコル（TLS v1.2 まで）で許可する暗号スイートの一覧を指定します。MySQL でのみサポートされます。

DBTLSCipher13

Zabbix プロキシが TLS v1.3 プロトコルで許可する暗号スイートの一覧を指定します。MySQL でのみサポートされ、バージョン 8.0.16 から利用できます。

DebugLevel

デバッグレベルを設定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。

デフォルト: 3
 範囲: 0-5

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可します。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

ExternalScripts

外部スクリプトの配置場所を指定します（datadir のコンパイル時インストール変数に依存します）。

デフォルト: /usr/local/share/zabbix/externalscripts

Fping6Location

fping6 の場所を指定します。fping6 バイナリの所有者が root であり、SUID フラグが設定されていることを確認してください。fping ユーティリティで IPv6 アドレスを処理できる場合は、空 (Fping6Location=) にしてください。

デフォルト: /usr/sbin/fping6

FpingLocation

fping の場所を指定します。fping バイナリの所有者が root であり、SUID フラグが設定されていることを確認してください。

デフォルト: /usr/sbin/fping

HistoryCacheSize

履歴キャッシュサイズをバイト単位で設定します。履歴データを保存するための共有メモリサイズです。

デフォルト: 16M
 範囲: 128K-16G

HistoryIndexCacheSize

履歴インデックスキャッシュのサイズをバイト単位で設定します。履歴キャッシュに保存された履歴データをインデックス化するための共有メモリサイズです。インデックスキャッシュのサイズは、1 アイテムをキャッシュするのにおおよそ 100 バイト必要です。

デフォルト: 4M
 範囲: 128K-16G

ホスト名

一意で、大文字小文字を区別するプロキシ名を指定します。プロキシ名がサーバーに認識されていることを確認してください。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大長: 128

デフォルト: HostnameItem によって設定されます

HostnameItem

Hostname を設定するためのアイテムを指定します（未定義の場合）。
 このアイテムは、エージェントの場合と同様にプロキシ上で実行されます。
 Hostname が設定されている場合は無視されます。
 ユーザーパラメータ、パフォーマンスカウンター、エイリアスはサポートしませんが、system.run[] はサポートします。

デフォルト: system.hostname

HousekeepingFrequency

ハウスキーピング処理の頻度を設定します（時間単位）。ハウスキーピングとは、データベースから古くなった情報を削除する処理です。
 注: プロキシ起動時の負荷を下げるため、ハウスキーピングはプロキシ起動後 30 分間延期されます。そのため、HousekeepingFrequency が 1 の場合、プロキシ起動後の最初のハウスキーピング処理は 30 分後に実行され、その後は 1 時間ごとに繰り返されます。
 自動ハウスキーピングは、HousekeepingFrequency を 0 に設定することで無効化できます。この場合、ハウスキーピング処理は housekeeper_execute ランタイム制御オプションからのみ開始できます。
ランタイム制御 オプションと、ハウスキーピング処理 の詳細も参照してください。

デフォルト: 1
 範囲: 0-24

Include

設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
 Zabbix のインストール時、このパラメーターは /usr/local/etc に設定されますが、コンパイル時に変更した場合はその設定に従います。

Include パラメーターに関する一般的なルール:

- パスは zabbix_proxy.conf ファイルの場所からの相対パスにできます。
- 1 つの設定ファイル内で複数の Include パラメーターを指定できます。
- Zabbix は Include パラメーターを読み取るとすぐに処理するため、設定ファイルの末尾に配置することを検討してください。

含まれるファイルに関するルール:

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルは正しい構文でなければなりません。そうでない場合、そのコンポーネントは起動しません。

ディレクトリまたはファイルパターンを含める際のルール:

- 含める順序を前提にしないでください（たとえば、ファイルはアルファベット順には含まれません）。
- 1 つの設定を上書きする目的であっても、同じパラメーターを複数の含まれるファイルで定義しないでください。どの値が使用されるか予測できないためです。
- テキストエディターが自動的に作成するバックアップコピーに注意してください（たとえば、`include/my_specific.conf` を編集すると `include/my_specific.conf.BAK` が作成される場合があります）。両方のファイルが含まれてしまいます。

ディレクトリ内の関連ファイルのみを含めるには、ワイルドカード (*) を使ってパターンマッチングを行います。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java ゲートウェイの IP アドレスまたはホスト名を指定します。Java poller が起動している場合にのみ必要です。

JavaGatewayPort

Zabbix Java ゲートウェイのポートを指定します。

デフォルト: 10052
 範囲: 1024-32767

ListenBacklog

TCP キュー内で保留中の接続の最大数を設定します。
 デフォルト値はハードコードされた定数で、システムによって異なります。
 サポートされる最大値もシステムに依存し、値が高すぎる場合は'implementation-specified maximum' に静かに切り詰められることがあります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

トラッパーの待ち受け IP アドレスを指定します。カンマ区切りのリストです。
 このパラメーターが指定されていない場合、トラッパーはすべてのネットワークインターフェースで待ち受けます。

デフォルト: 0.0.0.0

ListenPort

トラッパーの待受ポートを指定します。

デフォルト: 10051
 範囲: 1024-32767

LoadModule

プロキシの起動時にロードするモジュールを指定します。モジュールは、プロキシの機能を拡張するために使用されます。モジュールは LoadModulePath で指定したディレクトリに配置するか、モジュール名の前にパスを指定する必要があります。前に指定したパスが絶対パス（/ で始まるパス）の場合、LoadModulePath は無視されます。
 形式: - LoadModule=<module.so> - LoadModule=<path/module.so> - LoadModule=</abs_path/module.so>

複数の LoadModule パラメータを指定できます。

LoadModulePath

プロキシモジュールの配置場所への完全なパスです。デフォルト値はコンパイルオプションによって異なります。

LogFile

プロキシのログファイルを指定します。

必須: はい。LogType が file に設定されている場合。それ以外の場合は不要です。

LogFileSize

最大ログファイルサイズを設定します（MB 単位）。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新しいファイルとして再作成されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogSlowQueries

データベースクエリがログに記録されるまでの最大実行時間を設定します（ミリ秒）。
0 - 遅いクエリをログに記録しません。
 このオプションは DebugLevel=3 から有効になります。

Default: 0
 Range: 0-3600000

LogType

ログ出力タイプを指定します: - *file* - LogFile パラメータで指定されたファイルにログを書き込みます; - *system* - syslog にログを書き込みます; - *console* - 標準出力にログを書き込みます。

デフォルト: file

MaxConcurrentChecksPerPoller

各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが一度に実行できる非同期チェックの最大数を設定します。StartHTTPAgentPollers、StartAgentPollers、およびStartSNMPPollers を参照してください。

デフォルト: 1000
 範囲: 1-1000

PidFile

PID ファイルを指定します。

デフォルト: /tmp/zabbix_proxy.pid

ProxyBufferMode

履歴、ディスカバリ、および自動登録データの保存方式を指定します: - *disk* - データはデータベースに保存され、データベースからアップロードされます; - *memory* - データはメモリに保存され、メモリからアップロードされます。バッファーのメモリが不足すると、古いデータは破棄されます。シャットダウン時にはバッファーは破棄されます。- *hybrid* - プロキシバッファーは通常、メモリモードと同様に動作しますが、メモリが不足するか、最も古いレコードが設定された経過時間を超えるまでその状態が続きます。その場合、バッファーはデータベースにフラッシュされ、すべてのデータがアップロードされるまでディスクモードのように動作し、その後再びメモリで動作を開始します。シャットダウン時には、メモリバッファーはデータベースにフラッシュされます。

こちらも参照: Proxy memory buffer。

デフォルト: disk
 値: disk; memory; hybrid

ProxyConfigFrequency

Zabbix サーバーから設定データを取得する頻度を秒単位で設定します。
 アクティブプロキシのパラメーターです。パッシブプロキシでは無視されます（ProxyModeパラメーターを参照）。

デフォルト: 10
 範囲: 1-604800

ProxyLocalBuffer

データがすでにサーバーと同期されている場合でも、データをローカルに保持する期間を設定します（時間単位）。
 このパラメーターは、ローカルデータをサードパーティ製アプリケーションで使用する場合に利用できます。

デフォルト: 0
 範囲: 0-720

ProxyMemoryBufferAge

プロキシのメモリバッファ内のデータの最大保持時間を秒単位で設定します。
 有効にした場合（0 以外）、プロキシのメモリバッファ内のレコードがこれより古くなると、すべてのレコードがサーバーにアップロードされるまで、プロキシバッファをデータベースモードに切り替えます。このパラメーターは ProxyOfflineBuffer パラメーター以下である必要があります。

デフォルト: 0
 範囲: 0;600-864000

ProxyMemoryBufferSize

収集された履歴、ディスカバリ、および自動登録データ用の共有メモリキャッシュサイズを設定します（バイト単位）。有効にすると (0 以外)、プロキシはキャッシュがいっぱいになるか、保存されたレコードが定義された ProxyMemoryBufferAge より古くなるまで、履歴、ディスカバリ、および自動登録データをメモリ内に保持します。このパラメーターは ProxyLocalBuffer パラメーターと同時に使用できません。

Default: 0
 Range: 0;128K-2G

ProxyMode

プロキシの動作モードを設定します。
0 - アクティブモードのプロキシ
1 - パッシブモードのプロキシ
 注意アクティブプロキシを使用している場合、(機密性のある) プロキシ設定データが Zabbix サーバーの trapper ポートにアクセスできる第三者に公開される可能性があります。これは、誰でもアクティブプロキシを装って設定データを要求でき、認証が行われないためです。

デフォルト: 0
 範囲: 0-1

ProxyOfflineBuffer

Zabbix サーバーとの接続がない場合に、データを保持する期間を時間単位で設定します。
 古いデータは失われます。

デフォルト: 1
 範囲: 1-720

サーバー

ProxyMode が `active` に設定されている場合: 設定データの取得元およびデータの送信先として、Zabbix サーバーの IP アドレスまたは DNS 名 (address:port)、またはcluster (address:port;address2:port) を指定します。
 ポートを指定しない場合は、デフォルトのポートが使用されます。
 クラスターノードはセミコロンで区切る必要があります。

ProxyMode が `passive` に設定されている場合: Zabbix サーバーの IP アドレス (必要に応じて CIDR 表記) または DNS 名を指定します。カンマ区切りのリストです。
 受信接続は、ここに列挙されたアドレスからのみ受け付けられます。IPv6 サポートが有効な場合、`127.0.0.1`、`::127.0.0.1`、`::ffff:127.0.0.1` は同等に扱われます。
`::/0` は任意の IPv4 または IPv6 アドレスを許可します。`0.0.0.0/0` は任意の IPv4 アドレスを許可するために使用できます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: はい

SNMPTrapperFile

SNMP トラップデーモンからプロキシへデータを渡すために使用する一時ファイルを指定します。
`zabbix_trap_receiver.pl` または SNMPTT の設定ファイルと同じである必要があります。

デフォルト: `/tmp/zabbix_traps.tmp`

SocketDir

内部 Zabbix サービスで使用される IPC ソケットを保存するディレクトリを指定します。

デフォルト: `/tmp`

SourceIP

次の接続に使用する送信元 IP アドレスを指定します。

- Zabbix server への送信接続
- エージェントレス接続（VMware、SSH、JMX、SNMP、Telnet、および単純チェック）
- HTTP エージェント接続
- スクリプトアイテムの JavaScript HTTP リクエスト
- 前処理の JavaScript HTTP リクエスト
- Vault への接続

SSHKeyLocation

SSH チェックおよびアクションで使用する公開鍵と秘密鍵の場所を指定します。

SSLCertLocation

クライアント認証に使用する SSL クライアント証明書ファイルの場所を指定します。
 このパラメータは、Web 監視、Vault との通信、およびテレメトリデータソースの ClickHouse で使用されます。

SSLKeyLocation

クライアント認証に使用する SSL 秘密鍵ファイルの場所を指定します。
 このパラメータは、Web 監視、Vault との通信、およびテレメトリデータソースの ClickHouse で使用されます。

SSLCALocation

SSL サーバー証明書の検証に使用する認証局（CA）ファイルの場所を指定します。
 このパラメータの値は、`CURLOPT_CAPATH` libcurl オプションとして設定されることに注意してください。7.42.0 より前の libcurl バージョンでは、libcurl が OpenSSL を使用するようにコンパイルされている場合にのみ、この設定が有効になります。詳細については、cURL web pageを参照してください。
 このパラメータは、Web 監視、SMTP 認証、HTTP エージェントアイテム、Vault との通信、およびテレメトリデータソースの ClickHouse で使用されます。

StartAgentPollers

Zabbix エージェントの事前 fork 済みインスタンス数`pollers` を設定します。`MaxConcurrentChecksPerPoller` を参照してください。

デフォルト: 1
 範囲: 0-1000

StartAPMCollectors

APM コレクターの事前フォークされたインスタンス（apm managerおよびapm workerプロセス）を起動するには、1 に設定します。

デフォルト: 0
 範囲: 0-1

StartBrowserPollers

browser アイテムの事前 fork 済みインスタンス数pollersを設定します。

デフォルト: 1
 範囲: 0-1000

StartDBSyncers

history syncers の事前フォーク済みインスタンス数を設定します。
 注: この値を変更する際は注意してください。増やしても、かえって悪影響を及ぼす場合があります。

デフォルト: 4
 範囲: 1-100

StartDiscoverers

discovery workers の事前フォークされたインスタンス数を設定します。

デフォルト: 5
 範囲: 0-1000

StartHTTPAgentPollers

HTTP agent pollers の事前フォークされたインスタンス数を設定します。MaxConcurrentChecksPerPoller を参照してください。

デフォルト: 1
 範囲: 0-1000

StartHTTPPollers

HTTP pollers の事前フォーク済みインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartIPMIPollers

IPMI pollers の事前フォーク済みインスタンス数を設定します。

デフォルト: 0
 範囲: 0-1000

StartJavaPollers

Java pollers の事前フォーク済みインスタンス数を設定します。

デフォルト: 0
 範囲: 0-1000

StartODBCPollers

ODBC pollers の事前フォークされたインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartPingers

ICMP pingers の事前フォークされたインスタンス数を設定します。

デフォルト: 1
 範囲: 0-1000

StartPollersUnreachable

到達不能なホスト用のポーラーの事前フォークされたインスタンス数を設定します（IPMI と Java を含む）。通常のポーラー、IPMI ポーラー、または Java ポーラーを起動する場合は、少なくとも 1 つの到達不能なホスト用ポーラーが稼働している必要があります。

デフォルト: 1
 範囲: 0-1000

StartPollers

pollers の事前フォークされたインスタンス数を設定します。

デフォルト: 5
 範囲: 0-1000

StartPreprocessors

前もって起動する前処理worker スレッドの数は、利用可能な CPU コア数以上に設定する必要があります。前処理が CPU に依存しておらず、ネットワークリクエストが多い場合は、より多くの worker を設定してください。

デフォルト: 16
 範囲: 1-1000

StartSNMPPollers

SNMP pollers の事前フォークされたインスタンス数を設定します。MaxConcurrentChecksPerPoller を参照してください。

デフォルト: 1
 範囲: 0-1000

StartTelemetryQueryPollers

非同期 Telemetry クエリpollerの、フォーク前に起動するインスタンス数を設定します。
MaxConcurrentChecksPerPollerを参照してください。

デフォルト: 1
 範囲: 0-1000

StartSNMPTrapper

SNMP trapper プロセスを開始するには"1" に設定します。

デフォルト: 0
 範囲: 0-1

StartTrappers

trappers の事前フォークされたインスタンス数を設定します。
 トラッパーは、Zabbix sender およびアクティブエージェントからの受信接続を受け付けます。

デフォルト: 5
 範囲: 0-1000

StartVMwareCollectors

事前に fork されたvmware collector インスタンスの数を設定します。

デフォルト: 0
 範囲: 0-250

StatsAllowedIP

統計情報の要求を許可する外部 Zabbix インスタンスの IP アドレス（必要に応じて CIDR 表記）または DNS 名を指定します。カンマ区切りのリストです。
 このパラメータが設定されていない場合、統計情報の要求は受け付けられません。
IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 は任意の IPv4 または IPv6 アドレスを許可します。0.0.0.0/0 は任意の IPv4 アドレスを許可するために使用できます。

例:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

テレメトリデータソースを、<provider>[;<options>] 形式で設定します。

- <provider> - サポートされているテレメトリデータソースのいずれか: clickhouse
- <options> - コンマ区切りの key=value ペア。値は引用符で囲むことができます。

ClickHouse のオプション:

- url - テレメトリデータソースへの接続に使用する HTTP または HTTPS URL を設定します。
- db - データベース名を設定します。
- username - データベースへのログインに使用するユーザー名を設定します。
- password - データベースへのログインに使用するパスワードを設定します。
- ssl_cert_file、ssl_key_file、ssl_key_password、ssl_verify_peer、ssl_verify_host、ssl_ca_location、ssl_cert_location、ssl_key_location - データベースへのセキュアな（TLS/SSL）接続を設定するためのオプションです。
- source_ip - テレメトリデータソースへの送信接続に使用する送信元 IP アドレスを設定します。
- vault_path - username と password の値を保持する Vault シークレットへのパスを設定します。username および password オプションと同時に使用しないでください。

ClickHouse の設定例:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Vault を使用する ClickHouse の設定例:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

タイムアウト

Zabbix サーバー、エージェント、Web サービス、およびレガシー SNMP チェック（単一の OID 番号または文字列）との接続確立とデータ交換を待機する時間を、秒単位で指定します。

このパラメーターは、さまざまな通信操作の継続時間を定義します。

- Zabbix エージェント上でのリモートコマンド実行
- SSH / Telnet コマンド実行
- IPMI インターフェースが利用できなくなった場合のアイテムの再スケジュール
- 権限または暗号化の問題によりデータ交換に失敗した場合の Zabbix サーバーへの応答送信
- IPC 非同期ソケットおよびランタイム制御オプションの期限
- 非同期ポーラーの DNS リクエスト
- アクティブチェックのハートビートへの応答
- アクティブエージェントからの Zabbix エージェントデータ（値）の取得
- Zabbix sender からのデータ取得
- Zabbix エージェントへのアクティブチェック一覧の送信

このタイムアウトは、Web インターフェースで柔軟なタイムアウト 設定が構成されているチェック（グローバル、プロキシ、またはアイテムごとのレベル）には使用されません。
たとえば、SNMP `walk[OID]` および `get[OID]` アイテムは Web インターフェースで設定されたタイムアウトを使用しますが、レガシー SNMP チェックは引き続きサーバーのタイムアウト値を使用します。

既定値: 3
 範囲: 1-30

TLSAccept

Zabbix サーバーからの受信接続に必要な暗号化レベルを指定します。パッシブプロキシで使用され、アクティブプロキシでは無視されます。複数の値をカンマ区切りで指定できます: - *unencrypted* - 暗号化なしの接続を受け入れます（デフォルト）; - *psk* - TLS と事前共有鍵（PSK）を使用した接続を受け入れます; - *cert* - TLS と証明書を使用した接続を受け入れます。

必須: パッシブプロキシでは、TLS 証明書または PSK パラメータが定義されている場合は必須（*unencrypted* 接続でも）; それ以外の場合は不要

TLSCAFile

ピア証明書の検証に使用する、Zabbix コンポーネント間の暗号化通信向けにトップレベルの CA 証明書を含むファイルのフルパスを指定します。

TLSCertFile

Zabbix コンポーネント間の暗号化通信に使用される、プロキシ証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

TLSCipherAll

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 で OpenSSL 1.1.1 以降に使用する暗号スイート文字列を指定します。証明書ベースおよび PSK ベースの暗号化に対する既定の ciphersuite 選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。
証明書ベースの暗号化に対する既定の暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 で OpenSSL 1.1.1 以降に対する暗号スイート文字列を指定します。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。
PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号スイート文字列を指定します。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Zabbix サーバーへのプロキシ接続に必要な暗号化レベルを指定します。アクティブプロキシで使用され、パッシブプロキシでは無視されます。指定できる値は 1 つだけです。- *unencrypted* - 暗号化なしで接続します（デフォルト）。- *psk* - TLS と事前共有鍵（PSK）を使用して接続します。- *cert* - TLS と証明書を使用して接続します。

必須: アクティブプロキシでは必須。TLS 証明書または PSK パラメータが定義されている場合は、*unencrypted* 接続でも必須。それ以外の場合は不要

TLSCRLFile

失効した証明書を含むファイルのフルパスを指定します。このパラメーターは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

暗号化された Zabbix コンポーネント間の通信に使用される、プロキシの秘密鍵を含むファイルの完全なパス名を指定します。

TLSListen

トラッパーソケットで TLS のみの接続を要求します。

サポートされる値:

- required - TLS 接続のみを受け入れます

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用する、プロキシの事前共有鍵を含むファイルのフルパスを指定します。

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用する、事前共有鍵の識別文字列を指定します。

TLSServerCertIssuer

許可するサーバー証明書の発行者を指定します。

TLSServerCertSubject

許可するサーバー証明書のサブジェクトを指定します。

TmpDir

一時ディレクトリを指定します。

デフォルト: `/tmp`

TrapperTimeout

次の処理のタイムアウトを秒単位で指定します:
 - Zabbix サーバーからの設定データの取得;
 - グローバルスクリプトの実行またはリモートコマンドの実行。

デフォルト: 300
 範囲: 1-300

UnavailableDelay

unavailability 期間中にホストの到達可能性を確認する頻度を秒単位で設定します。

デフォルト: 60
 範囲: 1-3600

UnreachableDelay

到達不能期間中にホストの到達可能性を確認する頻度を秒単位で設定します。

デフォルト: 15
 範囲: 1-3600

UnreachablePeriod

到達不能になってから何秒後にホストを利用不可として扱うかを設定します。

デフォルト: 45
 範囲: 1-3600

ユーザー

システム上に存在する特定のユーザーに権限を下げます。
root として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: zabbix

Vault

vault プロバイダーを指定します: - *HashiCorp* - HashiCorp KV Secrets Engine version 2; - *CyberArk* - CyberArk Central Credential Provider
Web インターフェースで設定された vault プロバイダーと一致している必要があります。

デフォルト: HashiCorp

VaultAppRoleID

HashiCorp Vault の AppRole ID を指定します。
 設定すると、AppRole 認証が有効になります。VaultToken と併用することはできません。
 HashiCorp Vault でのみサポートされています。

VaultAppSecretID

HashiCorp Vault の AppRole secret ID を指定します。
 HashiCorp Vault でのみサポートされています。

VaultDBPath

データベース認証情報をキーで取得するための場所を指定します。Vault に応じて、これは Vault のパスまたはクエリです。

HashiCorp で使用されるキーは password と username です。

VaultPrefix=/v1/secret/data/zabbix/ を指定した場合のパスの例:

```
database
```

VaultPrefix を指定しない場合のパスの例:

```
secret/zabbix/database
```

CyberArk で使用されるキーは Content と UserName です。

例:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

このオプションは、DBUser と DBPassword が指定されていない場合にのみ使用できます。

VaultPrefix

Vault の種類に応じて、Vault のパスまたはクエリに使用するカスタムプレフィックスを指定します。指定しない場合は、最適なデフォルト値が使用されます。
HashiCorp では、VaultPrefix が指定されていない場合、マウントポイントの後に自動的に data が追加されることに注意してください。
HashiCorp では、VaultPrefix が指定されていない場合、マウントポイントの後に自動的に data が追加されることに注意してください。

HashiCorp のプレフィックス例:

```
/v1/secret/data/zabbix/
```

Cyberark のプレフィックス例:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

クライアント認証に使用する SSL 証明書ファイル名を指定します。証明書ファイルは PEM1 形式である必要があります。
 証明書ファイルに秘密鍵も含まれている場合は、SSL キーファイル欄を空欄のままにしてください。
 このファイルを含むディレクトリは、SSLCertLocation 設定パラメータで指定します。
 このオプションは省略できますが、CyberArkCCP vault では推奨されます。

VaultTLSKeyFile

クライアント認証に使用する SSL 秘密鍵ファイルの名前を指定します。秘密鍵ファイルは PEM1 形式である必要があります。
 このファイルを含むディレクトリは、SSLKeyLocation 設定パラメータで指定します。
 このオプションは省略できますが、CyberArkCCP vault では指定することを推奨します。

VaultToken

オプションの VaultDBPath 設定パラメータで指定したパスに対する読み取り専用権限を持ち、Zabbix プロキシ専用に生成された HashiCorp Vault の認証トークンを指定します。
`VaultToken` と `VAULT_TOKEN` 環境変数が同時に定義されている場合はエラーになります。
 VaultAppRoleID と併用できません。
 HashiCorp Vault でのみサポートされます。

VaultURL

vault サーバーの URL を指定します。SSLCALocation が指定されていない場合は、システム全体の CA 証明書ディレクトリが使用されます。

デフォルト: `https://127.0.0.1:8200`

VMwareCacheSize

VMware データを保存するための共有メモリサイズを設定します。
VMware の内部チェック `zabbix[vmware,buffer,...]` を使用して、VMware キャッシュの使用状況を監視できます（内部チェックを参照）。
vmware コレクターのインスタンスが起動するように設定されていない場合、共有メモリは割り当てられないことに注意してください。

デフォルト: `8M`
 範囲: 256K-2G

VMwareFrequency

単一の VMware サービスからデータを収集する頻度を設定します（秒）。
 この頻度は、任意の VMware 監視アイテムの最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwarePerfFrequency

単一の VMware サービスからパフォーマンスカウンター統計を取得する頻度を設定します（秒）。
 この頻度は、VMware パフォーマンスカウンターを使用する VMware 監視アイテム の最小更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwareTimeout

`vmware collector` が VMware サービス（vCenter または ESX ハイパーバイザー）からの応答を待機する最大秒数を設定します。

デフォルト: 10
 範囲: 1-300

WebDriverURL

WebDriver インターフェースの URL を指定します。

例（Selenium WebDriver standalone server と併用する場合）:

```
WebDriverURL=http://localhost:4444
```

エージェント

概要　Zabbix エージェントは監視対象に配置され、ローカルのリソースやアプリケーション（ハードドライブ、メモリ、プロセッサの統計情報など）をアクティブに監視します。

エージェントはローカルで稼働情報を収集し、さらなる処理のためにデータを Zabbix サーバーへ報告します。障害（ハードディスクの容量がいっぱいになった場合や、サービスプロセスがクラッシュした場合など）が発生すると、Zabbix サーバーは障害を報告した特定のマシンの管理者にアクティブに通知できます。

Zabbix エージェントは、統計情報の収集にネイティブのシステムコールを使用するため、非常に効率的です。

1 つのホスト上でエージェントの複数のインスタンスを実行できます。

関連項目:

- パッシブチェックとアクティブチェックのプロトコル
- root としてエージェントを実行する
- 設定パラメータ: Zabbix エージェント

- 設定パラメータ: Zabbix エージェント（Windows）
- Microsoft Windows 上の Zabbix エージェント
- IBM AIX 上の Zabbix エージェント

パッシブチェックとアクティブチェック

Zabbix エージェントは、パッシブチェックとアクティブチェックを実行できます。

- パッシブチェック - Zabbix エージェントは、Zabbix サーバー（またはプロキシ）からの要求に応答します。
  たとえば、サーバーがデータ（CPU 負荷など）を要求し、エージェントが結果を返します。
- アクティブチェック - Zabbix エージェントは、Zabbix サーバー（またはプロキシ）からの要求を待たずにデータを収集して送信します。
  まず、サーバーから監視対象のアイテム一覧（CPU 負荷、使用可能メモリなど）を取得し、その後、必要なデータを収集して、新しい値を定期的に返送します。

エージェントのチェックタイプは、該当する監視アイテムタイプを選択して設定します。
Zabbix エージェントは、*Zabbix agent* または *Zabbix agent (active)* タイプのアイテムを処理します。

サポート対象プラットフォーム　サポート対象プラットフォームについては、要件ページを参照してください。

Unix 系システム上のエージェント　Unix 系システム上の Zabbix エージェントは、監視対象のホスト上で実行されます。

インストール

Zabbix エージェントは、Linux ベースのシステムに次のいずれかの方法でインストールできます。

- Zabbix パッケージ - Agent コンポーネントを選択し（Zabbix のバージョン、OS ディストリビューション、OS のバージョンを選択した後）、表示される手順に従います。
- Zabbix ソース - ソースファイルをダウンロードし、`--enable-agent` オプションを指定してソースを設定し、Zabbix エージェントをコンパイルします。

Attention:
一般的に、32 ビットの Zabbix エージェントは 64 ビットのシステムで動作しますが、場合によっては失敗することがあります。

Note:
macOS、IBM AIX、FreeBSD、OpenBSD、Solaris 用のコンパイル済み Zabbix エージェントバイナリは、ダウンロードできます。現在の Zabbix サーバー/プロキシのバージョンと互換性のあるレガシーバイナリは、NetBSDおよびHP-UXで利用できます。IBM AIX にエージェントをインストールして実行する方法については、IBM AIX 上の Zabbix エージェントを参照してください。

パッケージとしてインストールした場合

Zabbix エージェントはデーモンプロセスとして動作します。エージェントは以下のコマンドで起動できます。

```
systemctl start zabbix-agent
```

これはほとんどの GNU/Linux システムで動作します。他のシステムでは、以下のコマンドを実行する必要があるかもしれません。

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

Zabbix エージェントを停止、再起動、またはステータスを確認するには、以下のコマンドを使用します。

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

バイナリアーカイブからインストールした場合

アーカイブを展開した後、`zabbix_agentd` バイナリと `zabbix_agentd.conf` 設定ファイルを探します。アーカイブ内の場所はプラットフォームによって異なります（例: `sbin/` と `conf/`、または `usr/sbin/` と `etc/zabbix/`）。展開したディレクトリからバイナリを直接実行し、`-c` オプションで設定ファイルを指定します。例:

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

UNIX 系システムでは、Zabbix エージェントは root 以外のユーザーで実行されるように設計されています。root として起動した場合、設定ファイルで [AllowRoot=1](<PARAMS>#allowroot) が設定されていない限り、`zabbix` ユーザー（または [User](<PARAMS>#user) パラメータで指定されたユーザー）に切り替わります。このユーザーがシステムに存在している必要があります。存在しない場合、エージェントは起動しません。

IBM AIX では、エージェントを System Resource Controller（SRC）で実行し、システム起動時に開始することもできます。詳細については、IBM AIX 上の Zabbix エージェントを参照してください。

手動で起動する

zabbix_agentd バイナリを直接実行して、Zabbix エージェントを起動できます。例:

```
zabbix_agentd
```

zabbix_agentd を含むディレクトリが PATH に含まれている場合に機能します。

Windows システム上のエージェント　Windows 上の Zabbix エージェントは、Windows サービスとして動作します。

インストール

Zabbix エージェントは、Windows では次のいずれかの方法でインストールできます。

- 事前コンパイル済みの Zabbix エージェントバイナリ - Zabbix エージェントの MSI インストーラーパッケージをダウンロードし、MSI からの Windows エージェントのインストール ページの手順に従います。
- Zabbix ソース - ソースファイルをダウンロードし、Windows での Zabbix エージェントのビルド ページの手順に従います。

Zabbix エージェントを Windows サービスとしてインストールする方法（ZIP アーカイブから）についての追加情報は、Microsoft Windows 上の Zabbix エージェント ページを参照してください。

オプション　Zabbix エージェントでは、次のコマンドラインパラメータを使用できます。

パラメータ	説明
UNIX および **Windows** エージェント	
-c --config <config-file>	設定ファイルへのパス。 デフォルト以外の設定ファイルを指定する場合に使用できます。
-f --foreground	Zabbix エージェントをフォアグラウンドで実行します（デフォルト: true）。
-p --print	認識されているアイテムを出力して終了します。 ユーザーパラメータの結果も返すには、設定ファイルがデフォルトの場所にない場合、その設定ファイルを指定する必要があります。
-t --test <item key>	指定したアイテムをテストして終了します。 ユーザーパラメータの結果も返すには、設定ファイルがデフォルトの場所にない場合、その設定ファイルを指定する必要があります。
-T --test-config	設定ファイルを検証して終了します。
-h --help	ヘルプ情報を表示します。
-V --version	バージョン番号を表示します。
UNIX エージェントのみ	
-R --runtime-control <option>	管理機能を実行します。ランタイムコントロールを参照してください。
Windows エージェントのみ	
-m --multiple-agents	複数のエージェントインスタンスを使用します（-i、-d、-s、-x オプションと併用）。 インスタンスのサービス名を区別するため、各サービス名には指定した設定ファイルの Hostname の値が含まれます。
-S --startup-type <value>	Zabbix Windows エージェントサービスのスタートアップの種類を設定します。使用できる値: automatic - *(デフォルト)* Windows の起動時にサービスを自動的に開始します。 delayed - 自動的に開始されるサービスの起動が完了した後に、サービスの開始を遅延させます (Windows Server 2008/Vista 以降で使用可能)。 manual - サービスを手動で開始します（ユーザーまたはアプリケーションによる）。 disabled - サービスを無効にし、ユーザーまたはアプリケーションが開始できないようにします。 このオプションは -i オプションと併用することも、インストール済みサービスのスタートアップの種類を変更するために単独で使用することもできます。
-i --install	Zabbix Windows エージェントをサービスとしてインストールします。
-d --uninstall	Zabbix Windows エージェントサービスをアンインストールします。
-s --start	Zabbix Windows エージェントサービスを開始します。
-x --stop	Zabbix Windows エージェントサービスを停止します。

コマンドラインパラメータの使用例:

- 値を含む、組み込みエージェントアイテムをすべて出力する。

- 指定した設定ファイルで mysql.ping キーが定義されているユーザーパラメータをテストする。
- 設定ファイルのデフォルトパスを使用して、Windows 用の Zabbix Agent サービスをインストールする。
- エージェント実行ファイルと同じフォルダーにある設定ファイルを使用して、Windows 用の Zabbix Agent [Hostname] サービスをインストールし、設定ファイルのHostname の値をサービス名に追加して一意にする。
- エージェント実行ファイルと同じフォルダーにある設定ファイルを使用して、インストール済みの Windows 用 Zabbix Agent サービスのスタートアップの種類を変更する。

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

ランタイム制御

ランタイム制御オプションを使用すると、エージェントプロセスのログレベルを変更できます。

オプション	説明	対象
log_level_increase[=target]	ログレベルを上げます。 target を指定しない場合、すべてのプロセスが影響を受けます。	対象は次のように指定できます。 プロセスタイプ - 指定したタイプのすべてのプロセス（例: listener） すべてのエージェントプロセスタイプを参照してください。 プロセスタイプ,**N** - プロセスタイプと番号（例: listener,3） **pid** - プロセス識別子（1～65535）。これより大きい値を指定する場合は、対象を「プロセスタイプ,N」の形式で指定してください。
log_level_decrease[=target]	ログレベルを下げます。 target を指定しない場合、すべてのプロセスが影響を受けます。	
userparameter_reload	現在の設定ファイルから、UserParameterおよびIncludeオプションの値を再読み込みします。	

例:

- すべてのプロセスのログレベルを上げる
- 3 番目のリスナープロセスのログレベルを上げる
- PID 1234 のプロセスのログレベルを上げる
- すべてのアクティブチェックプロセスのログレベルを下げる

```
zabbix_agentd -R log_level_increase
zabbix_agentd -R log_level_increase=listener,3
zabbix_agentd -R log_level_increase=1234
zabbix_agentd -R log_level_decrease="active checks"
```

Note:
OpenBSD、NetBSD、Windows ではランタイム制御はサポートされていません。

エージェントプロセスの種類　エージェントのプロセスの種類は次のとおりです。

- active checks - アクティブチェックを実行するプロセス。
- collector - データ収集を行うプロセス。
- listener - パッシブチェックを待ち受けるプロセス。

これらのプロセスの種類は、エージェントのログファイルで確認できます。

エージェントのログファイルは、ファイル所有者のみが読み書きできる権限で作成されます。さらに、このファイルは所有者グループによる読み取りが可能です。それ以外の権限はすべて拒否されます。

ユーザーとして実行　UNIX 上の Zabbix エージェントは、root 以外のユーザーとして実行するように設計されています。起動した root 以外のユーザーとして実行されます。そのため、問題なく任意の root 以外のユーザーとしてエージェントを実行できます。

「root」として実行しようとすると、ハードコードされた「zabbix」ユーザーに切り替わります。このユーザーはシステム上に存在している必要があります。エージェント設定ファイルの`AllowRoot`パラメータを適切に変更した場合にのみ、「root」としてエージェントを実行できます。

設定ファイル　設定ファイルのデフォルトパスは、オペレーティングシステムと Zabbix のインストール方法によって異なります。

- UNIX の場合、デフォルトは次のとおりです。
  - ソースからインストールした場合: `/usr/local/etc/zabbix_agentd.conf`、またはコンパイル時の変数 `--sysconfdir` または `--prefix` で設定したパス。
  - パッケージまたはアプライアンスからインストールした場合: `/etc/zabbix/zabbix_agentd.conf`。
- Windows の場合、デフォルトは `C:\Program Files\Zabbix Agent\zabbix_agentd.conf` です。

ホスト上でエージェントのインスタンスを 1 つだけ実行している場合は、デフォルトの設定ファイル、またはコマンドラインで指定した設定ファイルを使用できます。複数のインスタンスを実行する場合は、各エージェントインスタンスに専用の設定ファイルが必要です（いずれかのインスタンスではデフォルトの設定ファイルを使用できます）。

Zabbix エージェントの設定の詳細については、UNIX 版 Zabbix エージェントまたはWindows 版 Zabbix エージェントの設定ファイルオプションを参照してください。

ロケール　エージェントは UTF-8 ロケールを必要とするため、一部のテキスト系エージェントアイテムが期待どおりの内容を返せるようにしてください。ほとんどの最新の Unix 系システムでは UTF-8 ロケールがデフォルトですが、明示的に設定する必要があるシステムもあります。

終了コード　Zabbix エージェントは、正常終了の場合に 0 を返し、失敗の場合に 1 を返します。

パッシブおよびアクティブチェックのプロトコル

このセクションでは、Zabbix agent とZabbix agent 2 によって実行されるパッシブチェックおよびアクティブチェックのプロトコル詳細を説明します。

Zabbix は、エージェントとの通信に JSON ベースの通信プロトコルを使用します。

Zabbix agent と Zabbix agent 2 のプロトコルは、Zabbix 7.0 以降で統一されています。Zabbix agent と Zabbix agent 2 のリクエスト/レスポンスの違いは、「variant」タグの値で表されます。

パッシブチェック

パッシブチェックは、単純なデータ要求です。Zabbix サーバーまたはプロキシが何らかのデータ（たとえば CPU 使用率）を要求し、Zabbix エージェントが結果をサーバーに返します。

パッシブチェックは非同期で実行されます。つまり、他のチェックが開始される前に 1 つの要求への応答を受信する必要はありません。DNS 解決も同様に非同期です。

エージェントポーラーは、DNS ルックアップで返されたすべてのアドレスへの接続を試行します。これにより、1 つの IP アドレスに到達できない場合でも、ポーラーは次に利用可能なアドレスを試し、接続成功の可能性を高めます。この拡張は Zabbix サーバーとプロキシの両方に適用されます。

非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPoller により定義）。

非同期エージェントポーラーの数はStartAgentPollers パラメーターで定義されます。

サーバー要求

ヘッダーとデータ長の定義については、protocol details を参照してください。

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field	Type	Mandatory	Value
request	*string*	yes	"passive checks"
data	*array of object*	yes	パッシブチェックのアイテム。

Field		Type	Mandatory	Value
	key	*string*	yes	展開済みマクロを含むアイテムキー。
	timeout	*number*	yes	通信タイムアウト。

エージェント応答

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Field		Type	Mandatory	Value
version		*string*	yes	エージェントのバージョン番号。
variant		*number*	yes	エージェントの種類（*1* - Zabbix agent、*2* - Zabbix agent 2）。
data		*array of object*	yes	チェック結果を含みます。
	value	*string*	no	チェックが成功した場合のアイテム値。
	error	*string*	no	チェックが成功しなかった場合のエラーメッセージ。

たとえば、サポートされているアイテムの場合:

1. サーバーが TCP 接続を開く
2. サーバーが **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”agent.ping”,”timeout”:3}]}** を送信する
3. エージェントが要求を読み取り、次のように応答する **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”value”:1**
4. サーバーがデータを処理して、ここでは値'1' を取得する
5. TCP 接続が閉じられる

サポートされていないアイテムの場合:

1. サーバーが TCP 接続を開く
2. サーバーが **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”vfs.fs.size[/nono]”,”timeout”:3}]}** を送信する
3. エージェントが要求を読み取り、次のように応答する **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”error”:” item key.”}]}**
4. サーバーがデータを処理し、指定されたエラーメッセージとともにアイテムの状態をサポート対象外に変更する
5. TCP 接続が閉じられる

古いプロトコルへのフェイルオーバー

Zabbix サーバーまたはプロキシが、プレーンテキストプロトコルを使用する 7.2 より前のバージョンのエージェントと連携できるように、古いプロトコルへのフェイルオーバーが実装されています。

パッシブチェックは、再起動後またはインターフェース設定が変更されたときに、JSON プロトコル（7.0 以降）を使用して実行されます。
応答として有効な JSON が受信されない場合（エージェントが”ZBX_NOTSUPPORTED” を送信した場合）、Zabbix はそのインターフェースを古いプロトコルとしてキャッシュし、アイテムキーのみを送信してチェックを 再試行します。

なお、Zabbix サーバー/プロキシは 1 時間ごとに、すべてのインターフェースで再び新しいプロトコルでの動作を試み、必要に応じて古いプロトコルにフォールバックします。

アクティブチェック

アクティブチェックでは、より複雑な処理が必要です。エージェントは、独立して処理するために、まずサーバー/プロキシからアイテムおよび/またはリモートコマンドのリストを取得する必要があります。

アクティブチェックを取得するサーバー/プロキシは、エージェントの設定ファイルのServerActiveパラメータに一覧表示します。これらのチェックを要求する頻度は、同じ設定ファイルのRefreshActiveChecksパラメータで設定します。ただし、アクティブチェックの更新に失敗した場合は、ハードコードされた 60 秒後に再試行されます。

Note:
Zabbix 6.4 以降、エージェント（アクティブモード）は、サーバー/プロキシから 2 分ごと（デフォルト）に設定の完全なコピーを受信しなくなりました。その代わりに、ネットワークトラフィックとリソース使用量を削減するため、5 秒ごと（デフォルト）に増分設定同期が実行されます。この同期では、エージェントがまだ設定を受信していない場合、またはホスト設定、グローバルマクロ、グローバル正規表現に変更があった場合にのみ、サーバー/プロキシが設定の完全なコピーを提供します。

エージェントはその後、定期的に新しい値をサーバーに送信します。エージェントが実行するようにremote commands を受信していた場合は、実行結果も送信されます。アクティブエージェントでのリモートコマンド実行は、Zabbix agent 7.0 以降でサポートされていることに注意してください。

Note:
エージェントがファイアウォールの内側にある場合は、Active checks のみを使用することを検討してください。この場合、最初の受信接続を許可するためにファイアウォールを変更する必要がありません。

アイテム一覧の取得

エージェント要求

active checks 要求は、エージェントによって処理されるアクティブチェックを取得するために使用されます。
この要求は、エージェント起動時に送信され、その後はRefreshActiveChecks の間隔で送信されます。

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	ホスト名。
host_metadata	*string*	no	設定パラメータ HostMetadata または HostMetadataItem のメトリック値。
interface	*string*	no	設定パラメータ HostInterface または HostInterfaceItem のメトリック値。
ip	*string*	no	設定パラメータ ListenIP の先頭の IP。設定されている場合。
port	*number*	no	設定パラメータ ListenPort の値。設定されており、かつデフォルトのエージェント待受ポートではな
version	*string*	yes	エージェントのバージョン番号。
variant	*number*	yes	エージェントの種類 (*1* - Zabbix agent、*2* - Zabbix agent 2)。
config_revision	*number*	no	増分設定同期 用の設定識別子。
session	*string*	no	増分設定同期 用のセッション識別子。

サーバー応答

active checks 応答は、active checks 要求の処理後にサーバーからエージェントへ送信されます。

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
```

```
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	失敗時のエラー情報。
data	*array of objects*	no	アクティブチェックのアイテム。ホスト設定が変更されていない場合は省略されます。
key	*string*	no	展開済みマクロを含むアイテムキー。
itemid	*number*	no	アイテム識別子。
delay	*string*	no	アイテム更新間隔。 柔軟/スケジューリング間隔は、Zabbix 7.0 以降、Zabbix agent と Zabbix agent 2 の両方でサポートされます。
lastlogsize	*number*	no	アイテムの lastlogsize。
mtime	*number*	no	アイテムの mtime。
timeout	*string*	no	アイテムのタイムアウト。
refresh_unsupported	*number*	no	サポートされていないアイテムの更新間隔。
regexp	*array of objects*	no	グローバル正規表現。
name	*string*	no	グローバル正規表現名。
expression	*string*	no	グローバル正規表現。
expression_type	*number*	no	グローバル正規表現の種類。
exp_delimiter	*string*	no	グローバル正規表現の区切り文字。
case_sensitive	*number*	no	グローバル正規表現の大文字小文字の区別設定。
commands	*array of objects*	no	実行するリモートコマンド。アクションのoperation または手動script 実行によってリモートコマンドの実行がトリガーされた場合に含まれます。アクティブなエージェント上でのリモートコマンド実行は、Zabbix agent 7.0 以降でサポートされている点に注意してください。古いアクティブエージェントは、active checks のサーバー応答に含まれるリモートコマンドを無視します。
command	*string*	no	リモートコマンド。
id	*number*	no	リモートコマンド識別子。
wait	*number*	no	リモートコマンドの実行モード（アクションoperations からのコマンドでは”0”（nowait）、手動script 実行からのコマンドでは”1”（wait））。
timeout	*number*	no	サーバー/プロキシ 設定におけるリモートコマンド実行タイムアウト。
config_revision	*number*	no	増分設定同期 用の設定識別子。ホスト設定が変更されていない場合は省略されます。ホスト設定が変更された場合は増分されます。

サーバーは success を返さなければなりません。

例:

1. エージェントが TCP 接続を開く
2. エージェントがチェック一覧の取得を要求する
3. サーバーがアイテム一覧と実行するリモートコマンドを返す
4. エージェントが応答を解析する
5. TCP 接続が閉じられる
6. エージェントが定期的なデータ収集を開始し、リモートコマンドを実行する（Zabbix agent 7.0 以降でサポート）

Attention:
active check を使用すると、Zabbix サーバーの trapper ポートにアクセスできる第三者が、機密性のある設定データを取得できる可能性があることに注意してください。これは、誰でもアクティブエージェントを装ってアイテム設定データを要求できるためです。
暗号化 オプションを使用しない限り、認証は行われません。

収集済みデータの送信

エージェント送信

エージェントのデータ要求には、収集されたアイテム値と、実行されたリモートコマンドの値（ある場合）が含まれます。

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
  "commands": [
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "8495cd52070e6ca52b371f29c8574165",
  "host": "Zabbix server",
  "version": "8.0.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	アイテム値。
id	*number*	yes	値の識別子（ネットワーク問題が発生した場合に重複値を確認するために使用される増分カウンター）。
itemid	*number*	yes	アイテムの識別子。
value	*string*	no	アイテム値。
lastlogsize	*number*	no	アイテムの lastlogsize。
mtime	*number*	no	アイテムの mtime。
state	*number*	no	アイテムの状態。
source	*string*	no	値のイベントログソース。
eventid	*number*	no	値のイベントログ eventid。
severity	*number*	no	値のイベントログ severity。
timestamp	*number*	no	値のイベントログ timestamp。
clock	*number*	yes	値のタイムスタンプ（Epoch からの秒数）。
ns	*number*	yes	値のタイムスタンプのナノ秒。

Field	Type	Mandatory	Value
commands	*array of objects*	no	リモートコマンドの実行結果。アクティブなエージェントでのリモートコマンド実行は Zabbix エージェント 7.0 以降でサポートされていることに注意してください。古いアクティブエージェントは、アクティブチェックのサーバー応答に含まれるリモートコマンドを無視します。
id	*number*	no	リモートコマンドの識別子。
value	*string*	no	実行が成功した場合のリモートコマンドの実行結果。
error	*string*	no	実行が失敗した場合のリモートコマンド実行エラーメッセージ。
session	*string*	yes	エージェント起動時に毎回生成される一意のセッション識別子（32 桁の 16 進数）。
host	*string*	yes	ホスト名。
version	*string*	yes	エージェントのバージョン番号。
variant	*number*	yes	エージェントの種類（*1* - Zabbix エージェント、*2* - Zabbix エージェント 2）。

各値には仮想 ID が割り当てられます。値 ID は単純な昇順のカウンターで、1 つのデータセッション内で一意です（セッショントークンで識別されます）。この ID は、接続品質が悪い環境で送信される可能性のある重複値を破棄するために使用されます。

サーバー応答

エージェントデータ応答は、エージェントデータ要求を処理した後にサーバーからエージェントへ返送されます。

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	アイテム処理結果。

Attention:
一部の値の送信がサーバー側で失敗した場合（たとえば、ホストまたはアイテムが無効化または削除されている場合）、エージェントはそれらの値の再送信を行いません。

例:

1. エージェントが TCP 接続を開く
2. エージェントが値の一覧を送信する
3. サーバーがデータを処理し、ステータスを返す
4. TCP 接続が閉じられる

Attention:
エラーメッセージはサーバー側で 2048 文字に切り詰められます。

ハートビートメッセージ

エージェントが送信

ハートビートメッセージは、アクティブエージェントから Zabbix サーバー/プロキシに、`HeartbeatFrequency` 秒ごとに送信されます（Zabbix エージェントまたは Zabbix エージェント 2の設定ファイルで設定）。

アクティブチェックの可用性を監視するために使用されます。

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

フィールド	型	必須	値
request	*string*	yes	`active check heartbeat`
host	*string*	yes	ホスト名。
heartbeat_freq	*number*	yes	エージェントのハートビート間隔（HeartbeatFrequency 設定パラメータ）。
version	*string*	yes	エージェントのバージョン番号。
variant	*number*	yes	エージェントの種類（*1* - Zabbix エージェント、*2* - Zabbix エージェント 2）。

リダイレクト応答

ホストが再割り当てされた場合、サーバーはエージェントに対して、ハートビート（およびそれ以降のアクティブチェック）を別のプロキシまたはサーバーインスタンスにリダイレクトするよう指示できます。

```
{
  "response": "failed",
  "redirect": {
    "revision": 2,
    "address": "192.0.2.1:10055"
  }
}
```

フィールド		型	必須	値
response		*string*	yes	`success` | `failed`
redirect		*object*	yes	リダイレクトの指示。
	revision	*number*	yes	設定リビジョン識別子。
	address	*string*	yes	対象サーバー/プロキシのアドレス。

古い XML プロトコル

Note:
Zabbix は最大 16 MB の XML Base64 エンコードデータを受け取りますが、デコード後の個々の値は 64 KB 以下にする必要があります。これを超える場合、デコード時に 64 KB に切り詰められます。

エージェントを **root** として実行する

Zabbix **5.0.0** 以降、公式パッケージに含まれる Zabbix エージェント用の systemd サービスファイルには、`User` と `Group` のディレクティブが明示的に含まれています。
どちらも `zabbix` に設定されています。

`zabbix_agentd.conf` ファイルを使用して、Zabbix エージェントをどのユーザーで実行するかを設定することは、もはやできません。
エージェントはこの設定を無視し、systemd サービスファイルで指定されたユーザーとして実行されるためです。
Zabbix エージェントを root として実行するには、以下に説明する変更を行う必要があります。

Zabbix エージェント

Zabbix エージェントのデフォルトのユーザーとグループを上書きするには、次のコマンドを実行します。

```
systemctl edit zabbix-agent
```

次に、以下の内容を追加します。

```
[Service]
User=root
Group=root
```

デーモンを再読み込みし、zabbix-agent サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Zabbix エージェントでは、エージェント設定ファイルでユーザーを設定する機能が再び有効になります。次に、設定ファイルで `User=root` および `AllowRoot=1` 設定パラメータを設定する必要があります。

Zabbix エージェント 2

Zabbix エージェント 2 のデフォルトのユーザーとグループを上書きするには、次を実行します。

```
systemctl edit zabbix-agent2
```

次に、以下の内容を追加します。

```
[Service]
User=root
Group=root
```

デーモンを再読み込みし、zabbix-agent2 サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Zabbix エージェント **2** では、これにより実行ユーザーが完全に決まります。追加の変更は不要です。

設定パラメーター: Zabbix エージェント

概要

設定パラメータを使用すると、Zabbix エージェントの動作をカスタマイズできます。

パラメータ値は、Zabbix のエージェント設定ファイル（zabbix_agentd.conf）で指定します。パラメータが必須であることが明示されていない限り、すべてのパラメータは任意です。

このページは次の構成になっています。

- パラメータの概要（パラメータの詳細を表示するには、その名前をクリックしてください）
- パラメータの詳細

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
AllowKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を許可します。
AllowRoot	エージェントを root として実行できるようにします。
BufferSend	バッファーから Zabbix サーバー/プロキシへ値を送信する頻度を設定します。
BufferSize	メモリバッファーに保存できる値の最大数を設定します。
DebugLevel	デバッグレベルを設定します。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
DenyKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を拒否します。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可します。このパラメータは非推奨です。代わりに AllowKey、AllowKeyRegexp、DenyKey、または DenyKeyRegexp を使用してください。
HeartbeatFrequency	ハートビートメッセージの送信間隔を秒単位で設定します。
HostInterface	ホストインターフェースを指定します。
HostInterfaceItem	ホストインターフェースを取得するためのアイテムを指定します。
HostMetadata	ホストメタデータを指定します。
HostMetadataItem	ホストメタデータを取得するための Zabbix エージェントアイテムを指定します。
Hostname	ホスト名を指定します。
HostnameItem	ホスト名を取得するための Zabbix エージェントアイテムを指定します。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
ListenBacklog	TCP キューで保留できる接続数の最大値を設定します。
ListenIP	サーバー/プロキシからの接続を受け付けるエージェントの待ち受け IP アドレスを指定します。
ListenPort	エージェントの待ち受けポートを指定します。
LoadModule	エージェントの起動時にロードするモジュールを指定します。
LoadModulePath	エージェントモジュールの場所への完全なパスを指定します。
LogFile	エージェントのログファイルを指定します。
LogFileSize	ログファイルサイズの最大値を設定します。
LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録できるようにします。
LogType	ログ出力の種類を指定します。
MaxLinesPerSecond	log または logrt のアクティブチェックの処理時に、Zabbix サーバー/プロキシへ送信する 1 秒あたりの新しい行数の最大値を設定します。
PidFile	PID ファイルを指定します。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度を指定します。
Server	Zabbix サーバー/プロキシの IP アドレスまたは DNS 名を指定します。
ServerActive	アクティブチェックを取得する Zabbix サーバー/プロキシのアドレスまたはクラスター設定を指定します。
SourceIP	送信元 IP アドレスを指定します。

パラメータ	説明
StartAgents	パッシブチェックを処理するために事前にフォークされる zabbix_agentd のインスタンス数を設定します。
Timeout	Zabbix プロキシまたはサーバーとの接続の確立およびデータ交換を待機する時間を秒単位で指定します。
TLSAccept	受信接続に必要な暗号化レベルを指定します。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルへの完全なパス名を指定します。これは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルへの完全なパス名を指定します。これは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	GnuTLS の優先順位文字列、または OpenSSL（TLS 1.2）の暗号文字列を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS の優先順位文字列、または OpenSSL（TLS 1.2）の暗号文字列を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS の優先順位文字列、または OpenSSL（TLS 1.2）の暗号文字列を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
TLSCRLFile	失効した証明書を含むファイルへの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルへの完全なパス名を指定します。これは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	エージェントの事前共有鍵を含むファイルへの完全なパス名を指定します。これは、Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列を指定します。これは、Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。
TLSServerCertIssuer	許可するサーバー（プロキシ）証明書の発行者を指定します。
TLSServerCertSubject	許可するサーバー（プロキシ）証明書のサブジェクトを指定します。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡せるようにします。
User	システム上の特定の既存ユーザーに権限を降格します。
UserParameter	監視するユーザー定義パラメータを指定します。
UserParameterDir	UserParameter コマンドのデフォルト検索パスを指定します。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、デーモンのデフォルト値を示します。
- 値では環境変数を使用できます。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみサポートされます。

パラメーターの詳細

エイリアス

アイテムキーのエイリアスを設定します。長く複雑なアイテムキーを、より短く簡単なキーに置き換えるために使用できます。
 複数の Alias パラメータを指定できます。同じ Alias キーを持つパラメータを複数指定することはできません。
 異なる Alias キーから同じアイテムキーを参照できます。
 エイリアスは HostMetadataItem では使用できますが、HostnameItem パラメータでは使用できません。

例 1: ユーザー zabbix の ID を取得する。

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

これで、zabbix.userid という短縮キーを使用してデータを取得できるようになります。

例 2: デフォルトパラメータおよびカスタムパラメータを使用して CPU 使用率を取得する。

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

これにより、cpu.util キーを使用してデフォルトパラメータで CPU 使用率をパーセント単位で取得できるほか、cpu.util[all, idle, avg15] を使用して CPU 使用率に関する特定のデータを取得できます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、各ルールに異なるパラメータを指定して、vfs.fs.discovery を使用する複数のディスカバリルールを設定できます。たとえば、vfs.fs.discovery[foo]、vfs.fs.discovery[bar] などです。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式で、任意の数の任意の文字に一致する * 文字をサポートします。
 複数のキー一致ルールは、DenyKey、AllowKeyRegexp、および DenyKeyRegexp と組み合わせて定義できます。パラメーターは、記述された順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールが優先されるかが決まります。関連項目: エージェントチェックの制限。

AllowKeyRegexp

パターンに一致するアイテムキーの実行を許可します。キーのパターンには正規表現を使用します。
 DenyKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。詳細は、エージェントチェックの制限を参照してください。

例:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

エージェントが root として実行されることを許可します。無効にされていて、エージェントが root によって起動された場合、エージェントは代わりにユーザー zabbix へ切り替えようとします。通常ユーザーで起動された場合は影響しません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

BufferSend

バッファーから Zabbix サーバー/プロキシへ値を送信する頻度を設定します（秒単位）。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファー内の値の最大数を設定します。
 バッファーがいっぱいになると、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。

デフォルト: 100
 範囲: 2-65535

DebugLevel

デバッグレベルを設定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーのパターンはワイルドカード式で、任意の数の任意の文字に一致する * 文字をサポートします。
 AllowKey、AllowKeyRegexp、DenyKeyRegexp と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。詳細については、エージェントチェックの制限を参照してください。

デフォルト: DenyKey=system.run[*]

DenyKeyRegexp

パターンに一致するアイテムキーの実行を拒否します。キーのパターンには正規表現を使用します。
 AllowKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。
 別の system.run[*] ルールが指定されていない限り、すべての system.run[] アイテムはデフォルトで拒否されます（DenyKey=system.run[*]）。

関連項目：エージェントチェックの制限。

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可します。
 このパラメータは非推奨です。代わりに、AllowKey=system.run[*]、AllowKeyRegexp=^system\.run\[.*\]$、DenyKey=system.run[*]、または DenyKeyRegexp=^system\.run\[.*\]$

を使用してください。
 値に応じて、AllowKey、AllowKeyRegexp、DenyKey、および DenyKeyRegexp パラメータの内部エイリアスとして機能します。- 0 - DenyKey=system.run[*]（DenyKeyRegexp=^system\.run\[.*\]$）- 1 - AllowKey=system.run[*]（AllowKeyRegexp=^system\.run\[.*\]$）

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

HeartbeatFrequency

ハートビートメッセージの送信間隔を秒単位で設定します。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストのインターフェース（IP アドレスまたは DNS 名）を指定します。これはホストの自動登録時に使用されます。
この値は、新しく作成されるホストのインターフェースを設定するために使用され、IP アドレスまたは DNS アドレスのいずれかを明示的に設定できます。
詳細は、DNS をデフォルトのインターフェースとして使用するを参照してください。

未定義の場合、この値は HostInterfaceItem から取得されます。

この値が 255 文字を超えると、エージェントはエラーを出力して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストの自動登録時に、ホストインターフェース（IP アドレスまたは DNS 名）を決定するためのアイテムを指定します。
この値は、HostInterface が定義されていない場合にのみ使用されます。
詳細については、デフォルトインターフェースとして DNS を使用するを参照してください。

自動登録リクエスト中に、指定したアイテムから返された値が 255 文字を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[] アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録（アクティブエージェント）中に、ホストを識別または区別するためのメタデータを指定します。
HostMetadata を使用すると、ホスト名以外の情報でホストを区別できます。

定義されていない場合、値は HostMetadataItem から取得されます。

指定した値が 2034 バイトを超えている場合、または UTF-8 文字列でない場合、エージェントはエラーを出力して起動しません。
パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効でも、有効な IP アドレスまたは DNS 名でない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされていません。最初の改行位置で出力が切り詰められます。

範囲: 0～2034 バイト

HostMetadataItem

ホストメタデータを取得する Zabbix エージェントのアイテムを指定します。
このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録の試行時に取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。
HostMetadataItem を使用すると、ホスト名以外の情報に基づいてホストを識別できます。

ユーザーパラメータとエイリアスがサポートされています。
system.run[]アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

自動登録リクエスト中に、指定したアイテムが返す値が 65535 UTF-8 コードポイントの上限を超えている場合、エージェントは警告メッセージをログに記録します。
アイテムが返す値は UTF-8 文字列である必要があります。そうでない場合、その値は無視されます。
パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効でも、有効な IP アドレスまたは DNS 名ではない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。最初の改行で出力が切り捨てられます。

ホスト名

一意で、大文字小文字を区別するホスト名を指定します。カンマ区切りのリストです。
 アクティブチェックに必須であり、サーバーで設定されたホスト名と一致している必要があります。
 値が未定義の場合は、HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大長: ホスト名ごとに 128 文字、行全体で 2048 文字です。

デフォルト: HostnameItem により設定されます

HostnameItem

ホスト名を取得するための Zabbix エージェントのアイテムを指定します。
 このオプションは、Hostname が定義されていない場合にのみ使用されます。
 ユーザーパラメータまたはエイリアスはサポートされませんが、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の値に関係なく、system.run[]アイテムはサポートされます。

デフォルト: system.hostname

Include

設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
 Zabbix のインストール時、このパラメータは /usr/local/etc に設定されます。ただし、コンパイル時にこれを変更した場合は除きます。

Include パラメータに関する一般的なルール:

- パスは zabbix_agentd.conf ファイルの場所からの相対パスにできます。
- 1 つの設定ファイル内で複数の Include パラメータを指定できます。
- Zabbix は Include パラメータを読み取るとすぐに処理するため、設定ファイルの末尾に配置することを検討してください。

含まれるファイルに関するルール:

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルは正しい構文でなければなりません。そうでない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンを含める際のルール:

- 読み込み順序を前提にしないでください（たとえば、ファイルはアルファベット順には読み込まれません）。
- 同じパラメータを複数の含まれるファイルで定義しないでください（たとえば、一般的な設定を特定の設定で上書きする場合など）。どの値が使用されるか予測できないためです。
- 一部のテキストエディタが自動的に作成するバックアップコピーに注意してください(たとえば､include/my_specific.conf を編集すると include/my_specific.conf.BAK が作成される場合があります)。両方のファイルが含まれるためです。

ディレクトリ内の関連するファイルのみを含めるには、パターンマッチングにワイルドカード (*) を使用します。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

TCP キュー内で保留中の接続の最大数を設定します。
 デフォルト値はハードコードされた定数で、システムに依存します。
 サポートされる最大値もシステムに依存し、値が高すぎる場合は「implementation-specified maximum」に静かに切り詰められることがあります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

サーバー/プロキシからの接続を受け付けるエージェントの待ち受け IP アドレスを指定します。カンマ区切りのリストです。

デフォルト: 0.0.0.0

ListenPort

サーバー/プロキシからの接続を受け付けるエージェントの待受ポートを指定します。

デフォルト: 10050
 範囲: 1024-32767

LoadModule

エージェントの起動時に読み込むモジュールを指定します。
 モジュールは、エージェントの機能を拡張するために使用されます。
 モジュールは LoadModulePath で指定されたディレクトリに配置されている必要があります。配置されていない場合は、モジュール名の前にパスを指定する必要があります。先頭のパスが絶対パス（/ で始まる）である場合、LoadModulePath は無視されます。
 形式: - LoadModule=<module.so> - LoadModule=<path/module.so> - LoadModule=</abs_path/module.so>

複数の LoadModule パラメータを含めることができます。

LoadModulePath

エージェントモジュールの配置先へのフルパスを指定します。
 デフォルトはコンパイルオプションによって異なります。

LogFile

エージェントのログファイルを指定します。

必須: はい。LogType が file に設定されている場合。そうでない場合は不要

LogFileSize

最大ログファイルサイズを設定します（MB 単位）。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新しく開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。
 コマンドは、リモートで実行された場合にのみログに記録されます。
 system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力タイプを指定します: - *file* - LogFile パラメータで指定したファイルにログを書き込みます; - *system* - syslog にログを書き込みます; - *console* - 標準出力にログを書き込みます。

デフォルト: file

MaxLinesPerSecond

log または logrt のアクティブチェックを処理する際に、Zabbix サーバー/プロキシへ送信する 1 秒あたりの新しい行の最大数を設定します。
 指定した値は、log または logrt のアイテムキーで指定された maxlines パラメータによって上書きされます。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、MaxLinesPerSecond で設定された値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

PidFile

PID ファイルを指定します。

デフォルト: /tmp/zabbix_agentd.pid

RefreshActiveChecks

アクティブチェックの一覧を更新する頻度を秒単位で指定します。
 アクティブチェックの更新に失敗した場合、次回の更新は 60 秒後に試行されることに注意してください。

デフォルト: 5
 範囲: 1-86400

サーバー

Zabbix サーバー/プロキシの IP アドレス、必要に応じて CIDR 表記、または DNS 名を指定します。カンマ区切りのリストです。
 受信接続は、ここに一覧表示されたホストからのみ受け入れられます。IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 により任意の IPv4 または IPv6 アドレスが許可されます。0.0.0.0/0 を使用すると、任意の IPv4 アドレスを許可できます。”IPv4-compatible IPv6 addresses”（0000::/96 プレフィックス）はサポートされていますが、RFC4291 により非推奨です。スペースを使用できます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: はい、StartAgents が明示的に 0 に設定されていない場合

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレス、またはクラスタ構成を指定します。
 サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名で、必要に応じてコロンで区切ったポート番号を指定します。
 クラスタ構成は、セミコロンで区切られた 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマ区切りで指定できます。プロキシグループを使用しない場合、各 Zabbix サーバー/クラスタから 1 つを超える Zabbix プロキシを指定しないでください。Zabbix プロキシが指定されている場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りアドレスを指定して、複数の独立した Zabbix サーバーを並列に使用できます。スペースは使用できます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
 ホストのポートを指定する場合、IPv6 アドレスは角括弧で囲む必要があります。ポートが指定されていない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用する高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下の接続に使用する送信元 IP アドレスを指定します。

- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 一部のアイテム（`web.page.get`、`net.tcp.port` など）を実行する際の接続

StartAgents

受動チェックを処理するために、事前に fork された `zabbix_agentd` のインスタンス数を設定します。
 0 に設定すると、受動チェックは無効になり、エージェントはどの TCP ポートでも待ち受けしません。

デフォルト: 10
 範囲: 0-100

タイムアウト

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換に、どれだけの時間（秒単位）待機するかを指定します。

このパラメーターは、次のような各種通信処理の継続時間を定義します。

- Zabbix サーバーからの応答待ち;
- アイテム設定要求やactive checks におけるアイテムデータを含む、Zabbix サーバーへの要求送信;
- logfile を通じたログデータの取得;
- ハートビートメッセージの送信;
- `vfs.*` チェックの最大実行時間;
- Zabbix エージェントモジュールでの使用;
- バージョン 7.0 より古いサーバーまたはプロキシがタイムアウトなしでチェックを送信する場合のフォールバックとしての使用。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（全体、プロキシ、またはアイテムごとのレベル）には 使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受信接続に必要な暗号化レベルを指定します。
 パッシブチェックで使用されます。
 複数の値をカンマ区切りで指定できます: - *unencrypted* - 暗号化なしの接続を受け入れます（デフォルト）; - *psk* - TLS と事前共有鍵（PSK）を使用した接続を受け入れます; - *cert* - TLS と証明書を使用した接続を受け入れます。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合（*unencrypted* 接続であっても）は必須です。それ以外の場合は不要です

TLSCAFile

ピア証明書の検証に使用する、Zabbix コンポーネント間の暗号化通信向けにトップレベルの CA 証明書を含むファイルの完全なパス名を指定します。

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

TLSCipherAll

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。証明書ベースおよび PSK ベースの暗号化に対する、デフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 で OpenSSL 1.1.1 以降に使用する暗号スイート文字列を指定します。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先度文字列、または OpenSSL (TLS 1.2) の暗号スイート文字列を指定します。
証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 で OpenSSL 1.1.1 以降の暗号文字列を指定します。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の priority string または OpenSSL (TLS 1.2) の cipher string を指定します。PSK ベースの暗号化に対するデフォルトの ciphersuite 選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 で OpenSSL 1.1.1 以降に使用する暗号スイート文字列を指定します。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

エージェントから Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
 アクティブチェックで使用されます。
 指定できる値は 1 つだけです: - *unencrypted* - 暗号化なしで接続します（デフォルト）。- *psk* - TLS と事前共有キー (PSK) を使用して接続します。- *cert* - TLS と証明書を使用して接続します。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合（*unencrypted* 接続でも）; それ以外の場合はいいえ

TLSCRLFile

失効した証明書を含むファイルのフルパスを指定します。このパラメーターは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用される、エージェントの秘密鍵を含むファイルの完全なパス名を指定します。

TLSPSKFile

エージェントの事前共有鍵を含むファイルの完全なパス名を指定します。この鍵は、Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。

TLSPSKIdentity

Zabbix サーバーまたはプロキシとの暗号化通信に使用する、事前共有鍵の ID 文字列を指定します。

TLSServerCertIssuer

許可するサーバー（プロキシ）証明書の発行者を指定します。

TLSServerCertSubject

許可するサーバー（プロキシ）証明書のサブジェクトを指定します。

UnsafeUserParameters

ユーザー定義パラメータの引数に、すべての文字を渡せるようにします。次の文字は使用できません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
 さらに、改行文字も使用できません。

デフォルト: 0
 値: 0 - 許可しない, 1 - 許可する

ユーザー

システム上に存在する特定のユーザーに権限を下げます。
root として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: zabbix

UserParameter

監視するユーザー定義パラメーターを指定します。ユーザー定義パラメーターは複数指定できます。
 形式: UserParameter=<key>,<shell command>
 なお、シェルコマンドは空文字列または EOL のみを返してはなりません。UserParameterDir パラメーターが指定されている場合、シェルコマンドには相対パスを使用できます。

例:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

UserParameter コマンドのデフォルトの検索パスを指定します。使用すると、エージェントはコマンドを実行する前に、作業ディレクトリをここで指定したディレクトリに変更します。これにより、UserParameter コマンドでは完全なパスの代わりに、相対的な ./ プレフィックスを使用できます。
 指定できるエントリは 1 つだけです。

例:

```
UserParameterDir=/opt/myscripts
```

関連項目

1. バージョン 2.0.0 以降のアクティブチェックとパッシブチェックにおける Zabbix エージェント設定の違い

設定パラメーター: Zabbix エージェント (Windows)

概要

設定パラメータを使用すると、Windows 上の Zabbix エージェントの動作をカスタマイズできます。

パラメータ値は、Zabbix のエージェント設定ファイル（zabbix_agentd.conf）で指定します。
パラメータが必須であることが明示されていない限り、すべてのパラメータは任意です。

このページは次の構成になっています。

- パラメータの概要（パラメータの詳細を表示するには、その名前をクリックします）。
- パラメータの詳細。

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
AllowKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を許可します。
BufferSend	バッファーから Zabbix サーバー/プロキシへ値を送信する頻度を設定します。
BufferSize	メモリバッファーに保存できる値の最大数を設定します。
DebugLevel	デバッグレベルを設定します。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
DenyKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を拒否します。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可します。このパラメータは非推奨です。代わりに AllowKey、AllowKeyRegexp、DenyKey、または DenyKeyRegexp を使用してください。
HeartbeatFrequency	ハートビートメッセージの送信頻度を秒単位で設定します。

パラメータ	説明
HostInterface	ホストインターフェースを指定します。
HostInterfaceItem	ホストインターフェースを取得するためのアイテムを指定します。
HostMetadata	ホストメタデータを指定します。
HostMetadataItem	ホストメタデータを取得するための Zabbix エージェントアイテムを指定します。
Hostname	ホスト名を指定します。
HostnameItem	ホスト名を取得するための Zabbix エージェントアイテムを指定します。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
ListenBacklog	TCP キューで保留できる接続数の最大値を設定します。
ListenIP	サーバー/プロキシからの接続を受け付けるエージェントの待ち受け IP アドレスを指定します。
ListenPort	エージェントの待ち受けポートを指定します。
LogFile	エージェントのログファイルを指定します。
LogFileSize	ログファイルサイズの最大値を設定します。
LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録できるようにします。
LogType	ログの出力タイプを指定します。
MaxLinesPerSecond	`log`、`logrt`、または `eventlog` のアクティブチェックの処理時に、Zabbix サーバー/プロキシへ送信する 1 秒あたりの新しい行数の最大値を設定します。
PerfCounter	指定した期間 <period>（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値を返す新しいパラメータ <parameter_name> を定義します。
PerfCounterEn	指定した期間 <period>（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値を返す新しいパラメータ <parameter_name> を定義します。 PerfCounter とは異なり、パフォーマンスカウンターのパスは英語で指定する必要があります。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度を指定します。
Server	Zabbix サーバー/プロキシの IP アドレスまたは DNS 名を指定します。
ServerActive	アクティブチェックを取得する Zabbix サーバー/プロキシのアドレスまたはクラスター設定を指定します。
SourceIP	送信元 IP アドレスを指定します。
StartAgents	パッシブチェックを処理する `zabbix_agentd` のフォーク済みインスタンス数を設定します。
Timeout	Zabbix サーバーまたはプロキシとの接続の確立およびデータ交換を待機する時間（秒単位）を指定します。
TLSAccept	受信接続に必要な暗号化レベルを指定します。
TLSCAFile	ピア証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェントの証明書または証明書チェーンを含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSConnect	Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
TLSCRLFile	失効した証明書を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	エージェントの事前共有鍵を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列を指定します。このパラメータは、Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。
TLSServerCertIssuer	許可するサーバー（プロキシ）証明書の発行者を指定します。
TLSServerCertSubject	許可するサーバー（プロキシ）証明書のサブジェクトを指定します。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡せるようにします。
UserParameter	監視するユーザー定義パラメータを指定します。
UserParameterDir	UserParameter コマンドのデフォルト検索パスを指定します。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、デーモンのデフォルト値を示します。
- 値では環境変数を使用できます。
- Zabbix は、BOMなしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみ使用できます。

パラメーターの詳細

エイリアス

アイテムキーのエイリアスを設定します。長く複雑なアイテムキーを、より短く簡単なキーに置き換えるために使用できます。
 複数の `Alias` パラメータを指定できます。同じ `Alias` キーを持つパラメータを複数指定することはできません。
 異なる

Alias キーから同じアイテムキーを参照できます。
 エイリアスは HostMetadataItem では使用できますが、HostnameItem または PerfCounter パラメータでは使用できません。

例 1: サーバーからページファイルの使用率をパーセントで取得する。

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

これで、短縮キー pg_usage を使用してデータを取得できます。

例 2: デフォルトパラメータおよびカスタムパラメータで CPU 負荷を取得する。

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

これにより、cpu.load キーを使用してデフォルトパラメータで CPU 負荷を取得できるほか、cpu.load[percpu,avg15] を使用して CPU 負荷に関する特定のデータを取得できます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、各ルールに異なるパラメータを指定して、vfs.fs.discovery を使用する複数のディスカバリルールを設定できます。たとえば、vfs.fs.discovery[foo]、vfs.fs.discovery[bar] などです。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式であり、任意の数の任意の文字に一致する * 文字をサポートします。
 複数のキー一致ルールは、DenyKey、AllowKeyRegexp、DenyKeyRegexp と組み合わせて定義できます。パラメータは、記述された順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールが優先されるかが決まります。関連項目: エージェントチェックの制限。

AllowKeyRegexp

パターンに一致するアイテムキーの実行を許可します。キーのパターンには正規表現を使用します。
 DenyKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。関連項目: エージェントチェックの制限。

例:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

バッファーから Zabbix サーバー/プロキシへ値を送信する頻度を設定します（秒）。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリーバッファー内の値の最大数を設定します。
 バッファーがいっぱいになると、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。

デフォルト: 100
 範囲: 2-65535

DebugLevel

デバッグレベルを設定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーのパターンにはワイルドカード式を使用します。任意の数の任意の文字に一致する * 文字をサポートしています。
 AllowKey、AllowKeyRegexp、DenyKeyRegexp と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。順序によって、どの一致ルールを優先するかが決まります。関連項目: エージェントチェックの制限。

デフォルト: DenyKey=system.run[*]

DenyKeyRegexp

パターンに一致するアイテムキーの実行を拒否します。キーのパターンには正規表現を使用します。
 AllowKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述された順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。
 別の system.run[*] ルールが指定されていない限り、すべての system.run[] アイテムはデフォルトで拒否されます（DenyKey=system.run[*]）。

関連項目：エージェントチェックの制限。

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可します。
 このパラメータは非推奨です。代わりに、AllowKey=system.run[*]、AllowKeyRegexp=^system\.run\[.*\]$、DenyKey=system.run[*]、または DenyKeyRegexp=^system\.run\[.*\]$ を使用してください。
 値に応じて、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp パラメータの内部エイリアスとして機能します。- 0 - DenyKey=system.run[*] (DenyKeyRegexp=^system\.run\[.*\]$) - 1 - AllowKey=system.run[*] (AllowKeyRegexp=^system\.run\[.*\]$)

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

HeartbeatFrequency

ハートビートメッセージの送信間隔を秒単位で設定します。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホスト自動登録 時に使用するホストインターフェース（IP アドレスまたは DNS 名）を指定します。この値は、新しく作成されるホストのインターフェースを設定するために使用され、IP アドレスまたは DNS アドレスのいずれかを明示的に設定できます。詳細については、DNS をデフォルトのインターフェースとして使用する を参照してください。

定義されていない場合、この値は HostInterfaceItem から取得されます。

この値が 255 文字を超えると、エージェントはエラーを出力して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストの自動登録時に、ホストインターフェース（IP アドレスまたは DNS 名）を決定するためのアイテムを指定します。
この値は、HostInterface が定義されていない場合にのみ使用されます。
詳細については、デフォルトインターフェースとして DNS を使用するを参照してください。

自動登録リクエスト中に、指定したアイテムから返された値が 255 文字を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[] アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録（アクティブエージェント）中に、ホストを識別または区別するために使用するメタデータを指定します。
HostMetadata を使用すると、ホスト名以外の情報でホストを区別できます。

定義されていない場合、値は HostMetadataItem から取得されます。

指定した値が 2034 バイトを超える場合、または UTF-8 以外の文字列の場合、エージェントはエラーを出力して起動しません。
パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効でも、IP アドレスまたは DNS 名として有効でない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。最初の改行位置で出力が切り捨てられます。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータを取得する Zabbix エージェントのアイテムを指定します。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録の試行時に取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。HostMetadataItem を使用すると、ホスト名以外の情報に基づいてホストを区別できます。

ユーザーパラメータ、パフォーマンスカウンター、エイリアスがサポートされています。system.run[]アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

自動登録リクエスト中に、指定されたアイテムが返す値が 65535 UTF-8 コードポイントの上限を超えている場合、エージェントは警告メッセージをログに記録します。アイテムが返す値は UTF-8 文字列である必要があります。それ以外の場合、その値は無視されます。パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効であっても、正しい IP アドレスまたは DNS 名ではない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。出力は最初の改行で切り捨てられます。

ホスト名

一意で、大文字小文字を区別するホスト名を指定します。カンマ区切りのリストです。
 アクティブチェックに必要であり、サーバーで設定されたホスト名と一致している必要があります。未定義の場合、値は HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大長: ホスト名ごとに 128 文字、行全体で 2048 文字です。

デフォルト: HostnameItem により設定されます

HostnameItem

ホスト名を取得するための Zabbix エージェントのアイテムを指定します。
 このオプションは、Hostname が定義されていない場合にのみ使用されます。
 ユーザーパラメータ、パフォーマンスカウンター、エイリアスはサポートされませんが、system.run[] アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の値にかかわらずサポートされます。
 詳しくは、詳細な説明も参照してください。

デフォルト: system.hostname

Include

構成ファイルに含めるディレクトリまたは個別のファイルを指定します（Zabbix エージェントを Windows MSI インストーラーパッケージを使用してインストールした場合、デフォルトでは C:\Program Files\Zabbix Agent にあります。Zabbix エージェントを zip アーカイブとしてインストールした場合は、インストール時に指定したフォルダーにあります）。

Include パラメータに関する一般的なルール:

- パスは、zabbix_agentd.conf ファイルの場所を基準にした相対パスにできます。
- 1 つの構成ファイルに複数の Include パラメータを指定できます。
- Zabbix は Include パラメータを読み取るとすぐに処理するため、構成ファイルの末尾に配置することを検討してください。

インクルードするファイルのルール:

- 各ファイルは読み取り可能である必要があります。
- 各ファイルの構文は正しくなければなりません。正しくない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンをインクルードする場合のルール:

- インクルードの順序を想定しないでください（ファイルはアルファベット順にインクルードされるとは限りません）。
- 複数のインクルードファイルで同じパラメータを定義しないでください（たとえば、一般的な設定を特定の設定で上書きする場合）。どの値が使用されるかを予測できないためです。
- 一部のテキストエディターが自動的に作成するバックアップコピーに注意してください(たとえば､include/my_specific.conf を編集すると、include/my_specific.conf.BAK が作成される場合があります)。この場合、両方のファイルがインクルードされます。

ディレクトリから関連するファイルのみをインクルードするには、パターンマッチングにワイルドカード（*）を使用します。

例:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

TCP キュー内で保留中の接続の最大数を設定します。
 デフォルト値はハードコードされた定数で、システムに依存します。
 サポートされる最大値もシステムに依存し、値が高すぎる場合は「実装で指定された最大値」に黙って切り捨てられることがあります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

サーバー/プロキシからの接続に対するエージェントの待ち受け IP アドレスを指定します。カンマ区切りのリストです。

デフォルト: 0.0.0.0

ListenPort

サーバー/プロキシからの接続を受け付けるエージェントの待受ポートを指定します。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイルを指定します。

デフォルト: c:\zabbix_agentd.log
 必須: LogType が file に設定されている場合は必須。それ以外は不要

LogFileSize

最大ログファイルサイズを MB 単位で設定します。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新規に作成されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。
 コマンドは、リモートで実行された場合にのみログに記録されます。
 `system.run[]` が `HostMetadataItem`、`HostInterfaceItem`、または `HostnameItem` パラメータによってローカルで起動された場合、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力の種類を指定します: - *file* - `LogFile` パラメータで指定したファイルにログを書き込みます; - *system* - Windows イベントログにログを書き込みます; - *console* - 標準出力にログを書き込みます。

デフォルト: `file`

MaxLinesPerSecond

`log`、`logrt`、または `eventlog` のアクティブチェックを処理する際に、Zabbix サーバー/プロキシへ送信する 1 秒あたりの新しい行数の最大値を設定します。
 指定した値は、`log`、`logrt`、または `eventlog` のアイテムキーに指定された `maxlines` パラメータによって上書きされます。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、`MaxLinesPerSecond` で設定された値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

PerfCounter

指定した時間期間 <period>（秒）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値となる新しいパラメーター <parameter_name> を定義します。
 構文: <parameter_name>,"<perf_counter_path>",<period>

たとえば、直近 1 分間の 1 秒あたりのプロセッサ割り込みの平均回数を取得したい場合は、次のように新しいパラメーター"interrupts"を定義できます。

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンターのパスは二重引用符で囲む必要がある点に注意してください。パラメーター名（interrupts）は、アイテムを作成する際のアイテムキーとして使用します。平均値の計算用サンプルは 1 秒ごとに取得されます。
Windows で利用可能なすべてのパフォーマンスカウンターの一覧を取得するには、`typeperf -qx` を実行してください。

PerfCounterEn

新しいパラメーター <parameter_name> を定義します。これは、指定した時間期間 <period>（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値です。
PerfCounter と比較して、perfcounter のパスは英語で指定する必要があります。
Windows Server 2008/Vista 以降でのみサポートされます。
 構文: <parameter_name>,"<perf_counter_path>",<period>

たとえば、直近 1 分間の 1 秒あたりのプロセッサー割り込みの平均回数を取得したい場合は、次のように新しいパラメーター"interrupts" を定義できます。

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンターのパスは二重引用符で囲む必要があることに注意してください。
パラメーター名（interrupts）は、アイテムを作成する際のアイテムキーとして使用します。
平均値の計算に使用するサンプルは 1 秒ごとに取得されます。
 英語の文字列の一覧は、次のレジストリキーを参照してください: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`。

RefreshActiveChecks

アクティブチェックの一覧を更新する頻度を秒単位で指定します。
 アクティブチェックの更新に失敗した場合、次回の更新は 60 秒後に試行されることに注意してください。

デフォルト: 5
 範囲: 1-86400

サーバー

Zabbix サーバー/プロキシの IP アドレス、必要に応じて CIDR 表記、または DNS 名を指定します。カンマ区切りのリストです。
 受信接続は、ここに一覧表示されたホストからのみ受け入れられます。IPv6 サポートが有効な場合、`127.0.0.1`、`::127.0.0.1`、`::ffff:127.0.0.1` は同等に扱われ、`::/0` により任意の IPv4 または IPv6 アドレスが許可されます。`0.0.0.0/0` を使用すると、任意の IPv4 アドレスを許可できます。"IPv4 互換 IPv6 アドレス"（`0000::/96` プレフィックス）は RFC4291 でサポートされていますが、非推奨です。スペースを使用できます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須: はい、StartAgents が明示的に 0 に設定されていない場合

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレス、またはクラスタ構成を指定します。
 サーバー/プロキシのアドレスは IP アドレスまたは DNS 名で、必要に応じてコロンで区切ったポートを指定できます。
 クラスタ構成は、セミコロンで区切った 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマ区切りで指定できます。プロキシグループを使用しない場合、各 Zabbix サーバー/クラスタから 1 つを超える Zabbix プロキシを指定しないでください。Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りアドレスを指定して、複数の独立した Zabbix サーバーを並列に使用できます。スペースは使用できます。
 ポートを指定しない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでは、そのホストのポートを指定する場合、角括弧で囲む必要があります。ポートを指定しない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータを指定しない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用する高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

範囲: ServerActive に一覧表示されるアクティブサーバーの数と、StartAgents で指定されるパッシブチェック用の事前 fork されたインスタンス数の合計は 64 未満でなければなりません。

SourceIP

以下の接続に使用する送信元 IP アドレスを指定します。

- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 一部のアイテム（web.page.get、net.tcp.port など）を実行する際の接続確立

StartAgents

受動チェックを処理するために、事前に fork される zabbix_agentd のインスタンス数を設定します。
 0 に設定すると、受動チェックは無効になり、エージェントは TCP ポートで待ち受けしなくなります。

デフォルト: 10
 範囲: 0-100 (*)

タイムアウト

Zabbix プロキシまたはサーバーとの接続を確立し、データを交換するまでに待機する時間を秒単位で指定します。

このパラメーターは、次のような各種通信処理の継続時間を定義します。

- Zabbix サーバーからの応答を待機する;
- アイテム設定要求やactive checks におけるアイテムデータを含む、Zabbix サーバーへの要求送信;
- logfile または Windows イベントログ監視を通じたログデータの取得;
- ハートビートメッセージの送信;
- vfs.* チェックの最大実行時間;
- Zabbix エージェントモジュールで使用される;
- バージョン 7.0 より古いサーバーまたはプロキシがタイムアウトなしでチェックを送信するシナリオにおけるフォールバックとして機能する。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテムごとのレベル）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受信接続に必要な暗号化レベルを指定します。
 パッシブチェックで使用されます。
 複数の値をカンマ区切りで指定できます: - *unencrypted* - 暗号化なしの接続を受け入れます（デフォルト）; - *psk* - TLS と事前共有鍵（PSK）を使用した接続を受け入れます; - *cert* - TLS と証明書を使用した接続を受け入れます。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合は必須です（*unencrypted* 接続の場合でも）。それ以外の場合はいいえ

TLSCAFile

Zabbix コンポーネント間の暗号化通信で使用される、ピア証明書の検証用に最上位 CA 証明書を含むファイルのフルパスを指定します。

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用する、エージェントの証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

TLSConnect

エージェントから Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
 アクティブチェックで使用されます。
 指定できる値は 1 つだけです: - *unencrypted* - 暗号化なしで接続します（デフォルト）。- *psk* - TLS と事前共有キー（PSK）を使用して接続します。- *cert* - TLS と証明書を使用して接続します。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合（*unencrypted* 接続でも）; それ以外の場合はいいえ

TLSCRLFile

失効した証明書を含むファイルのフルパスを指定します。このパラメーターは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

エージェントの秘密鍵を含むファイルの完全なパス名を指定します。これは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSPSKFile

Zabbix サーバーまたはプロキシとの暗号化通信に使用する、エージェントの事前共有鍵を含むファイルのフルパスを指定します。

TLSPSKIdentity

Zabbix サーバーまたはプロキシとの暗号化通信に使用する、事前共有鍵の ID 文字列を指定します。

TLSServerCertIssuer

許可するサーバー（プロキシ）証明書の発行者を指定します。

TLSServerCertSubject

許可するサーバー（プロキシ）証明書のサブジェクトを指定します。

UnsafeUserParameters

ユーザー定義パラメータの引数に、すべての文字を渡せるようにします。次の文字は使用できません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
 さらに、改行文字も使用できません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するためのユーザー定義パラメータを指定します。ユーザー定義パラメータは複数指定できます。
 形式: `UserParameter=<key>,<shell command>`
 なお、シェルコマンドは空文字列または EOL のみを返してはなりません。`UserParameterDir` パラメータが指定されている場合、シェルコマンドには相対パスを使用できます。

例:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

`UserParameter` コマンドのデフォルト検索パスを指定します。使用すると、エージェントはコマンドを実行する前に作業ディレクトリをここで指定したディレクトリに変更します。これにより、`UserParameter` コマンドでは完全なパスの代わりに相対的な `./` プレフィックスを使用できます。指定できるエントリは 1 つだけです。

例:

```
UserParameterDir=C:\opt\myscripts
```

エージェントの設定

Zabbix エージェントは、どちらの世代も Windows サービスとして実行されます。Zabbix エージェント 2 の場合は、以下の手順で `agentd` を `agent2` に置き換えてください。

Microsoft Windows ホスト上では、Zabbix エージェントの単一インスタンスまたは複数のインスタンスを実行できます。単一インスタンスでは、次のいずれかを使用できます。

- エージェントバイナリと同じディレクトリにあるデフォルトの設定ファイル
- コマンドラインで指定した設定ファイル

複数のインスタンスを使用する場合、各エージェントインスタンスには専用の設定ファイルが必要です（いずれかのインスタンスではデフォルトの設定ファイルを使用できます）。

設定ファイルの例は、Zabbix のソースアーカイブに次のファイルとして含まれています。

- Zabbix エージェントの場合は `conf/zabbix_agentd.conf`
- Zabbix agent2の場合は `conf/zabbix_agent2.conf`

設定ファイルを明示的に指定せず、アーカイブから Zabbix エージェント/エージェント 2 を Windows サービスとしてインストールする場合は、エージェントをインストールする前に、次の操作を行ってください。

- `conf/zabbix_agentd.conf` を、`zabbix_agentd.exe` をインストールするディレクトリに手動でコピーします。
- `conf/zabbix_agent2.conf` と `conf/zabbix_agent2.d` ディレクトリを、`zabbix_agent2.exe` をインストールするディレクトリに手動でコピーします。

Zabbix Windows エージェントの設定方法については、Zabbix エージェント設定ファイルまたは Zabbix エージェント 2 設定ファイルのオプションを参照してください。

Hostname パラメーター

ホストでアクティブチェックを実行するには、Zabbix エージェントでホスト名を定義する必要があります。また、エージェント側で設定したホスト名の値は、Web インターフェースでホストに設定されているホスト名と完全に一致している必要があります。

エージェント側のホスト名の値は、エージェント設定ファイルの `Hostname` または `HostnameItem` パラメーターで定義できます。これらのパラメーターが指定されていない場合は、デフォルト値が使用されます。

`HostnameItem` パラメーターのデフォルト値は、`system.hostname` エージェントキーによって返される値です。Windows では、ローカルホスト名を特定するために名前空間プロバイダーへ問い合わせる `gethostname()` 関数の結果を返します。どの名前空間プロバイダーからも応答がない場合は、NetBIOS 名が返されます。

`Hostname` のデフォルト値は、`HostnameItem` パラメーターによって返される値です。したがって、実質的に両方のパラメーターが指定されていない場合、実際のホスト名はホストの NetBIOS 名になります。Zabbix エージェントは、NetBIOS ホスト名を使用して Zabbix サーバーからアクティブチェックのリストを取得し、その結果をサーバーに送信します。

`system.hostname` キーは、*type* と *transform* という 2 つのオプションパラメーターをサポートしています。

Type は、アイテムが返す名前の種類を指定します。

- *netbios*（デフォルト）- 15 文字に制限された、大文字のみの NetBIOS ホスト名を返します。
- *host* - 大文字と小文字を区別し、完全な実際の Windows ホスト名（ドメインなし）を返します。
- *shorthost* - 最初のドットより前のホスト名の一部を返します。名前にドットが含まれていない場合は、文字列全体を返します。
- *fqdn* - 完全修飾ドメイン名（末尾のドットなし）を返します。

Transform を使用すると、ホスト名に適用する追加の変換ルールを指定できます。

- *none*（デフォルト）- 元の大文字と小文字を使用します。
- *lower* - テキストを小文字に変換します。

したがって、`zabbix_agentd.conf` ファイルの設定を簡略化し、統一するために、次のような方法を使用できます。

1. `Hostname` または `HostnameItem` パラメーターを未定義のままにすると、Zabbix エージェントはホスト名として NetBIOS ホスト名を使用します。
2. `Hostname` パラメーターを未定義のままにし、`HostnameItem` を次のように定義します。
   - `HostnameItem=system.hostname[host]` - Zabbix エージェントが完全な実際の Windows ホスト名（大文字と小文字を区別）をホスト名として使用します。
   - `HostnameItem=system.hostname[shorthost,lower]` - Zabbix エージェントが最初のドットより前のホスト名の一部だけを使用し、小文字に変換します。
   - `HostnameItem=system.hostname[fqdn]` - Zabbix エージェントが完全修飾ドメイン名をホスト名として使用します。

ホスト名は、Windows サービスのインストール、開始、停止、アンインストールに使用される Windows サービス名の一部としても使用されます。たとえば、Zabbix エージェント設定ファイルで Hostname=Windows_db_server を指定すると、エージェントは Windows サービス Zabbix Agent [Windows_db_server] としてインストールされます。したがって、Zabbix エージェントの各インスタンスで異なる Windows サービス名を使用するには、各インスタンスで異なるホスト名を使用する必要があります。

Windows サービスとしてエージェントをインストールする

エージェントをインストールする前に、conf/zabbix_agentd.conf を手動で、zabbix_agentd.exe がインストールされるディレクトリにコピーしてください。

デフォルトの設定ファイルを使用して Zabbix エージェントの単一インスタンスをインストールするには、次のコマンドを実行します。

```
zabbix_agentd.exe --install
```

Attention:
64 ビットシステムでは、64 ビットプロセスの実行に関連するすべてのチェックを正しく動作させるために、64 ビット版の Zabbix エージェントが必要です。

デフォルト以外の設定ファイルを使用したい場合は、サービスのインストールに次のコマンドを使用してください。

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

設定ファイルのフルパスを指定する必要があります。

Zabbix エージェントの複数インスタンスは、次のようにサービスとしてインストールできます。

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

インストールされたサービスは、コントロールパネルに表示されるはずです。

エージェントの起動

エージェントサービスを起動するには、コントロールパネルを使用するか、コマンドラインから実行できます。

デフォルトの設定ファイルを使用して Zabbix エージェントの単一インスタンスを起動するには、次のようにします。

```
zabbix_agentd.exe --start
```

別の設定ファイルを使用して Zabbix エージェントの単一インスタンスを起動するには、次のようにします。

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

複数の Zabbix エージェントインスタンスのうち 1 つを起動するには、次のようにします。

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

エージェントの停止

エージェントサービスを停止するには、コントロールパネルを使用するか、コマンドラインから実行できます。

デフォルトの設定ファイルで起動した Zabbix エージェントの単一インスタンスを停止するには、次のコマンドを使用します。

```
zabbix_agentd.exe --stop
```

別の設定ファイルで起動した Zabbix エージェントの単一インスタンスを停止するには、次のコマンドを使用します。

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

複数の Zabbix エージェントインスタンスのうち 1 つを停止するには、次のコマンドを使用します。

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

エージェント Windows サービスのアンインストール

デフォルトの設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには:

```
zabbix_agentd.exe --uninstall
```

デフォルト以外の設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Windows サービスから Zabbix エージェントの複数インスタンスをアンインストールするには:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

制限事項

Windows 用の Zabbix エージェントは、CPU が NUMA ノード間で不均一に分散される非標準の Windows 構成をサポートしていません。論理 CPU が不均一に分散されている場合、一部の CPU では CPU パフォーマンスメトリクスを利用できないことがあります。たとえば、72 個の論理 CPU があり、NUMA ノードが 2 つある場合、両方のノードにそれぞれ 36 個の CPU が必要です。

IBM AIX 上の **Zabbix** エージェント

概要

IBM AIX 用にコンパイル済みの Zabbix エージェントバイナリは、.tar.gz アーカイブとしてダウンロードできます。このページでは、アーカイブからエージェントをインストールする方法、手動または System Resource Controller（SRC）で実行する方法、システム起動時に開始する方法、および SRC から削除する方法について説明します。

エージェントのインストール

AIX の tar コマンドは -z オプションをサポートしていないため、gunzip を使用してアーカイブを展開します。展開されたファイルが、アーカイブに保存されている数値のユーザー ID およびグループ ID ではなく、展開を実行したユーザーの所有になるように、tar の -o オプションを使用します。たとえば、エージェントを /opt/zabbix に配置する場合（root として実行し、/opt/zabbix はまだ存在していない必要があります）：

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

AIX アーカイブでは、エージェントのバイナリは usr/sbin/ に、設定ファイルは etc/zabbix/ にあります。エージェントを起動する前に、必要に応じて /opt/zabbix/etc/zabbix/zabbix_agentd.conf を調整します（少なくともServer、ServerActive、Hostname パラメータ）。

root として起動した場合、設定ファイルでAllowRoot=1 が設定されていない限り、エージェントは zabbix ユーザー（またはUser パラメータで指定されたユーザー）に切り替わります。このユーザーはシステム上に存在している必要があります。存在しない場合、エージェントは起動しません。zabbix ユーザーがまだ存在しない場合は、同じ名前のグループとともに作成します。

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

login=false および rlogin=false 属性により、このアカウントをログインに使用できないようにします。ただし、エージェントはこのユーザーとして実行できます。

エージェントを手動で起動する

エージェントは手動で起動できます。

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

SRC を使用したエージェントの実行

エージェントの登録

エージェントは、System Resource Controller（SRC）にも登録できます。SRC を介して起動する場合、エージェントはフォアグラウンドモード（-f オプション）で実行する必要があります。それ以外の場合、SRC はエージェントプロセスを追跡できなくなり、停止できません。次に例を示します（root として実行）。

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

この例では、次のように指定しています。

- -s は、SRC コマンドで使用するサブシステム名を設定します。
- -p は、エージェントバイナリへの完全なパスを設定します。

- -a は、エージェントのコマンドライン引数を 1 つの引用符付き文字列として設定します。SRC はエージェントの起動時に、この文字列を個別の引数に分割します（-a オプションは 1 つだけ指定できます）。
- -u は、エージェントの実行ユーザーを設定します。サブシステムの作成時にユーザー名が数値のユーザー ID に変換されるため、ユーザーはあらかじめ存在している必要があります。
- -G は、エージェントを tcpip サブシステムグループに追加します。これにより、グループコマンドもエージェントに適用されます（たとえば、stopsrc -g tcpip を実行するとエージェントも停止します）。
- -S -n 15 -f 9 は、SRC がシグナルを使用してエージェントを制御するようにします。通常の停止では stopsrc が SIGTERM（15）を送信し、強制停止（stopsrc -f）では SIGKILL（9）を送信します。
- -R は、エージェントが異常終了した場合（たとえば、プロセスの 1 つがクラッシュした場合）に、SRC がエージェントを再起動するようにします。stopsrc の実行後はエージェントは再起動されません。

ユーザー ID はサブシステム定義に保存されるため、zabbix ユーザーを別のユーザー ID で再作成した場合は、定義を更新してください。

```
chssys -s zabbix_agentd -u zabbix
```

エージェントの起動と停止

エージェントは、startsrc、lssrc、stopsrc を使用して起動、照会、停止できます。

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

強制停止（stopsrc -f）は避けてください。メインのエージェントプロセスのみが強制終了され、他のエージェントプロセスは実行を継続してリスニングポートを開いたままにするため、手動で終了するまで（たとえば、kill -9 を使用して）エージェントを再度起動できなくなります。

システム起動時のエージェントの開始

エージェントを SRC に登録しても、システム起動時に開始されるわけではありません。起動時にエージェントを自動的に開始するには、/etc/inittab にエントリを追加します。

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

SRC からエージェントを削除する

システム起動時および SRC からエージェントを削除するには、次のコマンドを実行します。

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

エージェント **2**

概要　Zabbix エージェント 2 は、Go で書かれたZabbix エージェント の新世代版です（Zabbix エージェントの一部の C コードを再利用しています）。次の目的で設計されています。

- TCP 接続数を削減する;
- チェックの並列性 を改善する;
- 最小限のコードでシンプルなチェックを提供するプラグインや、長時間実行されるスクリプトと定期的なレポートによる独立したデータ収集を含む複雑なチェックをサポートするプラグインによって、容易に拡張できるようにする;
- Zabbix エージェントの代替として機能し、これまでのすべての機能をサポートする。

Zabbix エージェント 2 のすべてのメトリックは、プラグインによって収集されます。

これらの機能は、プラグインに固有です。

- パッシブチェックとアクティブチェックの両方に対する、スケジュール済みおよび柔軟な間隔
- スケジュールとタスクの並列実行数を考慮したタスクキュー管理
- プラグインレベルのタイムアウト
- 起動時の Zabbix エージェント 2 とそのプラグインの互換性チェック

プラグインには 2 種類あります。

- 組み込みプラグイン
- ロード可能プラグイン

独自のプラグイン を作成することもできます。

カスタム作成したプラグインを agent 2 に組み込む必要はありません。ロード可能プラグインとして追加できるため、新しいメトリック向けのプラグイン作成が簡単になります。

ロード可能プラグインは、Zabbix エージェント 2 を再コンパイルせずに実行時に追加できます。

関連項目:

- エージェントと agent 2 の比較
- 設定パラメータ: Zabbix エージェント 2
- 設定パラメータ: Zabbix エージェント 2 (Windows)
- 設定パラメータ: プラグイン
- Zabbix エージェント 2 プラグインプロトコル

パッシブチェックとアクティブチェック

Zabbix エージェント 2 は、Zabbix エージェントと同様に、パッシブチェックとアクティブチェックをサポートしています。さらに、Zabbix エージェント 2 のアクティブチェックでは、1 つのアクティブサーバー内で柔軟な間隔/スケジューリング間隔およびチェックの同時実行がサポートされます。

> **Note:**
> デフォルトでは、再起動後、Zabbix エージェント 2 はリソース使用量の急増を防ぐため、アクティブチェックの最初のデータ収集を、アイテムの更新間隔内の条件付きランダムな時刻にスケジュールします。エージェントの再起動直後に *Scheduling* 更新間隔が設定されていないアクティブチェックを実行するには、設定ファイルで`ForceActiveChecksOnStart`パラメータ（グローバルレベル）または `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` プラグインパラメータ（特定のプラグインチェックにのみ影響）を設定します。プラグインレベルのパラメータが設定されている場合は、グローバルパラメータより優先されます。

同時実行数の確認

異なるプラグインからのチェックは同時に実行できます。
1 つのプラグイン内で同時に実行できるチェック数は、プラグインのキャパシティ設定によって制限されます。
各プラグインには、ハードコードされたキャパシティ設定（既定値は 1000）があり、Plugins 設定のパラメータ である `Plugins.<PluginName>.System.Capacity=N` を使用して下げることができます。

サポート対象プラットフォーム　サポート対象プラットフォームについては、要件ページを参照してください。

Unix 系システム上のエージェント 2　Unix 系システム上の Zabbix エージェント 2 は、監視対象のホスト上で実行されます。

インストール

Zabbix エージェント 2 は、Linux ベースのシステムに以下のいずれかの方法でインストールできます。

- Zabbix パッケージ - Zabbix のバージョン、OS ディストリビューション、OS バージョンを選択した後、Agent 2 コンポーネントを選択し、手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、`--enable-agent2` オプションを指定して設定 し、エージェントをコンパイルします。

> **Note:**
> Zabbix エージェント 2 の監視機能は、個別に提供されるロード可能なプラグインによって拡張できます。詳細はロード可能なプラグイン を参照してください。

パッケージとしてインストールされている場合

Zabbix エージェント 2 はフォアグラウンドプロセスとして動作し、バックグラウンド実行には外部のサービスマネージャー（例: systemd）に依存します。
Zabbix エージェント 2 には、Linux 上でのデーモン化を行うための組み込みサポートはありません。

エージェントは次のコマンドで起動できます:

```
systemctl start zabbix-agent2
```

Zabbix エージェント 2 を停止、再起動、または状態確認するには、次のコマンドを使用します:

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

手動で起動する

zabbix_agent2 バイナリを見つけて直接実行することで、Zabbix エージェントを起動できます。たとえば、次のようにします。

```
zabbix_agent2
```

Windows システム上のエージェント **2**　Zabbix エージェント 2 はスタンドアロンのプロセスとして実行されますが、Windows サービスとしても実行できます。

インストール

Zabbix エージェント 2 は、次のいずれかの方法で Windows にインストールできます。

- 事前コンパイル済みの Zabbix エージェントバイナリ - エージェントの MSI インストーラーパッケージをダウンロードし、MSI からの Windows エージェントのインストール ページの手順に従います。
- Zabbix ソース - ソースファイルをダウンロードし、Windows での Zabbix エージェント 2 のビルド ページの手順に従います。

> **Note:**
> Zabbix エージェント 2 の監視機能は、個別に提供されるロード可能なプラグインによって拡張できます。詳細は、ロード可能なプラグイン を参照してください。

Zabbix エージェント 2 を Windows サービスとしてインストールする方法（ZIP アーカイブから）については、Microsoft Windows 上の Zabbix エージェント ページを参照してください。

オプション　Zabbix エージェント 2 では、次のコマンドラインパラメータを使用できます。

パラメータ	説明
UNIX および **Windows** エージェント	
-c --config <config-file>	設定ファイルへのパス。 デフォルト以外の設定ファイルを指定する場合に使用できます。
-f --foreground	Zabbix エージェントをフォアグラウンドで実行します（デフォルト: true）。
-p --print	認識されているアイテムを出力して終了します。 ユーザーパラメータの結果も返すには、設定ファイルがデフォルトの場所にない場合、その設定ファイルを指定する必要があります。
-t --test <item key>	指定したアイテムをテストして終了します。 ユーザーパラメータの結果も返すには、設定ファイルがデフォルトの場所にない場合、その設定ファイルを指定する必要があります。
-T --test-config	設定ファイルを検証して終了します。
-h --help	ヘルプ情報を出力して終了します。
-v --verbose	デバッグ情報を出力します。このオプションは、-p および -t オプションとともに使用します。
-V --version	エージェントのバージョンおよびライセンス情報を出力します。
-R --runtime-control <option>	管理機能を実行します。ランタイムコントロールを参照してください。
Windows エージェントのみ	
-m --multiple-agents	複数のエージェントインスタンスを使用します（-i、-d、-s、-x オプションを使用）。 インスタンスのサービス名を区別するため、各サービス名には指定した設定ファイルの Hostname の値が含まれます。
-S --startup-type <value>	Zabbix Windows エージェントサービスのスタートアップの種類を設定します。使用できる値は次のとおりです。 automatic - *(デフォルト)* Windows の起動時にサービスを自動的に開始します。 delayed - 自動的に開始されるサービスの起動が完了した後に、サービスの開始を遅延させます。 manual - サービスを手動で開始します（ユーザーまたはアプリケーションによる開始）。 disabled - サービスを無効にし、ユーザーまたはアプリケーションが開始できないようにします。 このオプションは -i オプションとともに使用することも、単独で使用してインストール済みサービスのスタートアップの種類を変更することもできます。
-i --install	Zabbix Windows エージェントをサービスとしてインストールします。
-d --uninstall	Zabbix Windows エージェントサービスをアンインストールします。
-s --start	Zabbix Windows エージェントサービスを開始します。
-x --stop	Zabbix Windows エージェントサービスを停止します。

コマンドラインパラメータの使用例を次に示します。

- 値を含む、組み込みエージェントアイテムをすべて出力します。
- 指定した設定ファイルで mysql.ping キーが定義されているユーザーパラメータをテストします。
- デフォルトの設定ファイルのパスを使用して、Windows 用の Zabbix エージェントサービスをインストールします。
- エージェント実行ファイルと同じフォルダーにある設定ファイルを使用して、インストール済みの Windows 用 Zabbix エージェントサービスのスタートアップの種類を変更します。

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

ランタイム制御

ランタイム制御では、リモート制御のためのいくつかのオプションを利用できます。

Option	Description
log_level_increase	ログレベルを上げます。
log_level_decrease	ログレベルを下げます。
metrics	利用可能なメトリクスを一覧表示します。
periodic_prof_enable	定期プロファイリングを有効にします。 CPU データを除き、すべてのメトリクスはその時点でデータを収集してダンプします。CPU データは、非表示の一時ファイルに時間をかけて収集されます。間隔が終了すると、そのファイルは CPU プロファイルとして公開され、設定された数の完了済みファイルが保持されます。 一時ファイルは ProfilerMaxFilesPerProfile の数には含まれません。
periodic_prof_disable	定期プロファイリングを無効にします。
periodic_prof_execute	5 秒間のサンプル取得後にプロファイルを書き込みます。 定期プロファイリングが有効な場合、このコマンドは現在の CPU 間隔を終了し、5 秒間の CPU プロファイルを別途書き込んだ後、新しい定期 CPU ファイルを開始します。 明示的に要求されたプロファイルは、コマンドの完了時に ProfilerMaxFilesPerProfile に従ってローテーションされます。 このコマンドがプロファイルを収集中は、すべてのプロファイラーランタイムコマンドが「profiler is busy」というメッセージで拒否されます。
periodic_prof_set_interval	プロファイリング間隔を設定します。
version	エージェントのバージョンを表示します。
userparameter_reload	現在の設定ファイルからUserParameterおよびIncludeオプションの値を再読み込みします。
help	ランタイム制御に関するヘルプ情報を表示します。

例:

- エージェント 2 のログレベルを上げる。
- ランタイム制御のオプションを表示する。

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

設定ファイル　設定ファイルのデフォルトパスは、オペレーティングシステムと Zabbix のインストール方法によって異なります。

- UNIX の場合、デフォルトは次のとおりです。
  - ソースからインストールした場合: /usr/local/etc/zabbix_agent2.conf、またはコンパイル時変数 --sysconfdir または --prefix で設定したパス。
  - パッケージからインストールした場合: /etc/zabbix/zabbix_agent2.conf。
- Windows の場合、デフォルトは C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf です。

ホスト上でエージェントの単一インスタンスが実行されている場合、デフォルトの設定ファイル、またはコマンドラインで指定した設定ファイルを使用できます。複数のインスタンスを実行する場合、各エージェントインスタンスには独自の設定ファイルが必要です（いずれかのインスタンスではデフォルトの設定ファイルを使用できます）。

Zabbix エージェント 2 の設定パラメータは、いくつかの例外を除き、Zabbix エージェントの設定パラメータとほぼ同じです。

パラメータ		説明
Zabbix エージェント 2 固有の設定パラメータ		
	ControlSocket	ランタイム制御ソケットのパス。Zabbix エージェント 2 は、ランタイムコマンドに制御ソケットを使用します。
	EnablePersistentBuffer、PersistentBufferFile、PersistentBufferPeriod	これらのパラメータは、アクティブアイテム用の Zabbix エージェント 2 の永続ストレージを設定するために使用します。
	ForceActiveChecksOnStart	エージェントが再起動直後にアクティブチェックを実行するか、時間をかけて均等に実行するかを決定します。
	Plugins	プラグインには、Plugins.<Plugin name>.<Parameter>=<value> 形式の独自のパラメータを設定できます。一般的なプラグインパラメータとして System.Capacity があり、同時に実行できるチェック数の上限を設定します。
	StatusPort	Zabbix エージェント 2 が HTTP ステータス要求を受け付け、設定済みプラグインの一覧と一部の内部パラメータを表示するためにリッスンするポートです。
Zabbix エージェント固有で、Zabbix エージェント 2 ではサポートされない設定パラメータ		
	AllowRoot、User	Zabbix エージェント 2 ではデーモン化をサポートしていないため、サポートされません。
	EnableRemoteCommands	非推奨です。両方のエージェントでは、代わりに AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp パラメータを使用します。
	ListenBacklog	Zabbix エージェントでは、TCP キューで保留できる接続の最大数を定義するために使用します。Zabbix エージェント 2 ではサポートされません。
	LoadModule、LoadModulePath	ロード可能なモジュールは Zabbix エージェント 2 ではサポートされません。
	MaxLinesPerSecond、LogRemoteCommands	Zabbix エージェント 2 では、代わりに Plugins.Log.MaxLinesPerSecond と Plugins.SystemRun.LogRemoteCommands を使用します。

パラメータ	説明
StartAgents	このパラメータは、Zabbix エージェントでパッシブチェックの同時実行数を増やすか、パッシブチェックを無効にするために使用します。Zabbix エージェント 2 では、同時実行数はプラグインレベルで設定し、capacity 設定で制限できます。Server パラメータが指定されていない場合、Zabbix エージェント 2 ではパッシブチェックが無効になります。

詳細については、Zabbix エージェント 2の設定ファイルオプションを参照してください。

終了コード　Zabbix エージェント 2 は、古い OpenSSL バージョン (1.0.1、1.0.2) でもコンパイルできます。

この場合、Zabbix は OpenSSL のロック用にミューテックスを提供します。ミューテックスのロックまたはアンロックに失敗すると、エラーメッセージが標準エラーストリーム (STDERR) に出力され、エージェント 2 はそれぞれ戻りコード 2 または 3 で終了します。

エージェントと **agent 2** の比較

このセクションでは、Zabbix エージェントと Zabbix agent 2 の違いについて説明します。

Parameter	Zabbix agent	Zabbix agent 2
Programming language	C	Go with some parts in C
Daemonization	yes	by systemd only (yes on Windows)
Supported extensions	Custom loadable modules in C.	Custom plugins in Go.
Requirements		
Supported platforms	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: all desktop and server versions since XP	Linux, Windows: all desktop and server versions, on which a supported Go version can be installed.
Supported crypto libraries	GnuTLS 3.1.18 and newer. OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. LibreSSL - tested with versions 2.7.4, 2.8.2 (certain limitations apply, see the Encryption page for details).	Linux: OpenSSL 1.0.1 and later. MS Windows: OpenSSL 1.1.1 or later. The OpenSSL library must have PSK support enabled. LibreSSL is not supported.
Monitoring processes		
Processes	A separate active check process for each server/proxy record.	Single process with automatically created threads. The maximum number of threads is determined by the GOMAXPROCS environment variable.
Metrics	**UNIX**: see a list of supported items. **Windows**: see a list of additional Windows-specific items.	**UNIX**: All metrics supported by Zabbix agent. Additionally, the agent 2 provides Zabbix-native monitoring solution for: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd, and other monitoring targets - see a full list of agent 2 specific items. **Windows**: All metrics supported by Zabbix agent, and also net.tcp.service* checks of HTTPS, LDAP. Additionally, the agent 2 provides Zabbix-native monitoring solution for: PostgreSQL, Redis.
Concurrency	Active checks for single server are executed sequentially.	Checks from different plugins or multiple checks within one plugin can be executed concurrently.
Third-party traps	no	yes
Additional features		
Persistent storage	no	yes

Parameter	Zabbix agent	Zabbix agent 2
Persistent files for `log*[]` metrics	yes (only on Unix)	no
Log data upload	Can be performed during log gathering to free the buffer.	Log gathering is stopped when the buffer is full, therefore the `BufferSize` parameter must be at least `MaxLinesPerSecond` x 2.
Changes user at runtime	yes (Unix-like systems only)	no (controlled by systemd)
User-configurable ciphersuites	yes	no

関連情報:

- *Zabbix* プロセスの説明: Zabbix エージェント, Zabbix agent 2
- 設定パラメーター: Zabbix エージェント UNIX / Windows, Zabbix agent 2 UNIX / Windows

組み込みプラグイン

これらのプラグインは Zabbix エージェント 2 に組み込まれており、追加設定なしですぐに利用できます。
プラグイン名をクリックすると、追加情報が記載されたプラグインリポジトリに移動します。

プラグイン名	説明	サポートされるアイテムキー	コメント
Agent	使用中の Zabbix エージェントのメトリクス。	agent.hostname, agent.ping, agent.version	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
CPU	システム CPU の監視 (CPU 数/CPU コア数、検出された CPU、使用率)。	system.cpu.discovery, system.cpu.num, system.cpu.util	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Docker	Docker コンテナの監視。	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	参照: 設定パラメータ
File	ファイルメトリクスの収集。	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Kernel	カーネルの監視。	kernel.maxfiles, kernel.maxproc	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Log	ログファイルの監視。	log, log.count, logrt, logrt.count	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。 参照: プラグイン設定パラメータ (Unix/Windows)
Memcached	Memcached サーバーの監視。	memcached.ping, memcached.stats	
Modbus	Modbus データを読み取ります。	modbus.get	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。

プラグイン名	説明	サポートされるアイテムキー	コメント
MQTT	公開された MQTT トピックの値を受信します。	mqtt.get	MQTT ブローカーへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッション またはデフォルト パラメータとして指定します。現在、TLS パラメータをアイテムキーのパラメータとして渡すことはできません。
MySQL	MySQL およびその派生製品の監視。	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッション またはデフォルト パラメータとして指定します。現在、TLS パラメータをアイテムキーのパラメータとして渡すことはできません。
NetIf	ネットワークインターフェースの監視。	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Oracle	Oracle Database の監視。	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	使用前に Oracle Instant Client をインストールしてください。
Proc	プロセスの CPU 使用率。	proc.cpu.util	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Redis	Redis サーバーの監視。	redis.config, redis.info, redis.ping, redis.slowlog.count	Redis への暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッション またはデフォルト パラメータとして指定します。TLS パラメータをアイテムキーのパラメータとして渡すことはできません。誤った、または無効な TLS 設定により Zabbix エージェント 2 が起動できなくなる場合があるため、TLS を有効にする前に証明書ファイル、権限、パスを確認してください。

プラグイン名	説明	サポートされるアイテムキー	コメント

Smart	S.M.A.R.T. の監視。	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	必要な smartctl の最小バージョンは 7.1 です。
			Zabbix エージェント 2 を実行するユーザーには smartctl への sudo/root 権限が必要です。
			このプラグインは次のコマンドのみを使用します。
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			サポートされるキー は、Linux/Windows 上の Zabbix エージェント 2 でのみ、パッシブチェックとアクティブチェックの両方で使用できます。
			参照:
			設定パラメータ

SW	インストール済みパッケージの一覧。	system.sw.packages, system.sw.packages.get	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Swap	スワップ領域のサイズ(バイト/割合)。	system.swap.size	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
SystemRun	指定したコマンドを実行します。	system.run	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。 参照: プラグイン設定パラメータ (Unix/Windows)
Systemd	systemd サービスの監視。	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP 接続の可用性チェック。	net.tcp.port	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
UDP	UDP サービスの可用性とパフォーマンスの監視。	net.udp.service, net.udp.service.perf	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Uname	システム情報の取得。	system.hostname, system.sw.arch, system.uname	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
Uptime	システム稼働時間メトリクスの収集。	system.uptime	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
VFSDev	VFS メトリクスの収集。	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
WebCertificate	TLS/SSL Web サイト証明書の監視。	web.certificate.get	
WebPage	Web ページの監視。	web.page.get, web.page.perf, web.page.regexp	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
ZabbixAsync	非同期メトリクス収集。	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。

プラグイン名	説明	サポートされるアイテムキー	コメント
ZabbixStats	Zabbix サーバー/プロキシの内部メトリクス、またはキュー内の遅延アイテム数。	zabbix.stats	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。
ZabbixSync	同期メトリクス収集。	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	サポートされるキーは、Zabbix エージェントのキー と同じパラメータを持ちます。

ロード可能なプラグイン

ロード可能なプラグインは、Zabbix エージェント 2 では標準で利用できないため、個別にインストールする必要があります。

- Linux では、パッケージを使用するか(例: Ubuntu に Ember+ をインストールする場合は `apt install zabbix-agent2-plugin-embe` プラグインをビルドできます。
- Windows では、MSI からインストールするか、プラグインをビルドできます。

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

プラグイン名をクリックすると、追加情報が記載されたプラグインの README ファイルを含むプラグインリポジトリに移動します。

プラグイン名	説明	サポートされるアイテムキー	コメント
Ceph	Ceph の監視。	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Ceph プラグインの設定パラメータ も参照してください。
Ember+	Ember+ の監視。	ember.get	Ember+ プラグインの設定パラメータ も参照してください。

プラグイン名	説明	サポートされるアイテムキー	コメント
MongoDB	MongoDB サーバーおよびクラスター（ドキュメント指向の分散データベース）の監視。	mongodb.cfg.discovery, mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションのパラメータとして指定します。 現在、TLS パラメータをアイテムキーのパラメータとして渡すことはできません。 MongoDB プラグインの設定パラメータも参照してください。
MSSQL	MSSQL データベースの監視。	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトのパラメータとして指定します。現在、TLS パラメータをアイテムキーのパラメータとして渡すことはできません。 MSSQL プラグインの設定パラメータも参照してください。
NVIDIA GPU	NVIDIA GPU の監視。	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	NVIDIA GPU プラグインの設定パラメータも参照してください。

プラグイン名	説明	サポートされるアイテムキー	コメント
PostgreSQL	PostgreSQL およびそのフォークの監視。	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトのパラメータとして指定します。 現在、TLS パラメータをアイテムキーのパラメータとして渡すことはできません。 PostgreSQL プラグインの設定パラメータも参照してください。

Note:
次のオプションを指定して起動したロード可能なプラグイン:
 - `-V --version` - プラグインのバージョンおよびライセンス情報を表示します。
 - `-h --help` - ヘルプ情報を表示します。
 - `-t, --test <item key>` - テスト用にプラグインを起動します（プラグイン設定は無視されます）。

設定パラメーター: **Zabbix** エージェント **2**

概要

設定パラメータを使用すると、Zabbix エージェント 2 の動作をカスタマイズできます。

Zabbix エージェント 2 は新世代の Zabbix エージェントであり、Zabbix エージェントの代わりに使用できます。

パラメータ値は、Zabbix のエージェント 2 設定ファイル（`zabbix_agent2.conf`）で指定します。パラメータが必須であることが明示されていない限り、すべてのパラメータは任意です。

このページは次の構成になっています。

- パラメータの概要（パラメータの詳細を確認するには、その名前をクリックしてください）
- パラメータの詳細

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
AllowKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を許可します。
BufferSend	バッファーから Zabbix サーバー/プロキシへ値を送信する頻度を設定します。
BufferSize	メモリバッファー内の値の最大数を設定します。
ControlSocket	「-R」オプションでランタイムコマンドを送信するためのコントロールソケットを指定します。
DebugLevel	デバッグレベルを設定します。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
DenyKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を拒否します。
EnablePersistentBuffer	アクティブアイテム用のローカル永続ストレージの使用を有効にします。
EnableProfiler	エージェントの起動時に Go プロファイリングを定期的に実行します。

パラメータ	説明
ForceActiveChecksOnStart	最初に受信した設定に対して、再起動直後にアクティブチェックを実行します。
HeartbeatFrequency	ハートビートメッセージの間隔を秒単位で設定します。
HostInterface	ホストインターフェースを指定します。
HostInterfaceItem	ホストインターフェースを取得するためのアイテムを指定します。
HostMetadata	ホストメタデータを指定します。
HostMetadataItem	ホストメタデータを取得するための Zabbix エージェントアイテムを指定します。
Hostname	ホスト名を指定します。
HostnameItem	ホスト名を取得するための Zabbix エージェントアイテムを指定します。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
ListenIP	サーバー/プロキシからの接続を受け付けるエージェントの待ち受け IP アドレスを指定します。
ListenPort	エージェントの待ち受けポートを指定します。
LogFile	エージェントのログファイルを指定します。
LogFileSize	ログファイルの最大サイズを設定します。
LogType	ログの出力タイプを指定します。
PersistentBufferFile	SQLite データベースを保持するファイルを指定します。
PersistentBufferPeriod	Zabbix サーバーまたはプロキシへの接続がない場合にデータを保存する期間を設定します。
PidFile	PID ファイルを指定します。
Plugins.<PluginName>.System.Capacity	同時に実行できるプラグインごとのチェック数の上限を設定します。
Plugins.Log.MaxLinesPerSecond	`log` または `logrt` のアクティブチェックを処理する際に、Zabbix サーバー/プロキシへ送信する 1 秒あたりの新しい行数の最大値を設定します。
Plugins.SystemRun.LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録します。
PluginSocket	ロード可能なプラグインとの通信に使用する UNIX ソケットのパスを指定します。
PluginTimeout	ロード可能なプラグインへの接続のタイムアウトを設定します。
ProfilerDir	プロファイリングデータを書き込むディレクトリです。
ProfilerMaxFilesPerProfile	プロファイルタイプごとに保持するファイルの最大数です。
ProfilerInterval	プロファイリングデータを書き込む間隔です。
RefreshActiveChecks	アクティブチェックの一覧を更新する頻度を指定します。
Server	Zabbix サーバー/プロキシの IP アドレスまたは DNS 名を指定します。
ServerActive	アクティブチェックを取得する Zabbix サーバー/プロキシのアドレスまたはクラスター設定を指定します。
SourceIP	送信元 IP アドレスを指定します。
StatusPort	HTTP ステータスリクエスト（`http://localhost:<port>/status`）を待ち受けるポートを指定します。
Timeout	Zabbix サーバーまたはプロキシとの接続の確立およびデータ交換を待機する時間（秒）を指定します。
TLSAccept	受信接続に必要な暗号化レベルを指定します。
TLSCAFile	ピア証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	OpenSSL の暗号スイート文字列（TLS 1.2）を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	OpenSSL 1.1.1 以降向けの OpenSSL の暗号スイート文字列（TLS 1.3）を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	OpenSSL の暗号スイート文字列（TLS 1.2）を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	OpenSSL 1.1.1 以降向けの OpenSSL の暗号スイート文字列（TLS 1.3）を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	OpenSSL の暗号スイート文字列（TLS 1.2）を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	OpenSSL 1.1.1 以降向けの OpenSSL の暗号スイート文字列（TLS 1.3）を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
TLSCRLFile	失効した証明書を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェントの秘密鍵を含むファイルの完全なパス名を指定します。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	エージェントの事前共有鍵を含むファイルの完全なパス名を指定します。Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列を指定します。Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。

パラメータ	説明
TLSServerCertIssuer	許可するサーバー（プロキシ）証明書の発行者を指定します。
TLSServerCertSubject	許可するサーバー（プロキシ）証明書のサブジェクトを指定します。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡せるようにします。
UserParameter	監視するユーザー定義パラメータを指定します。
UserParameterDir	UserParameter コマンドのデフォルト検索パスを指定します。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数を使用できます。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみサポートされます。

パラメーターの詳細

エイリアス

アイテムキーのエイリアスを設定します。長く複雑なアイテムキーを、より短く簡単なキーに置き換えるために使用できます。
 複数の Alias パラメータを指定できます。同じ Alias キーを持つパラメータを複数指定することはできません。
 異なる Alias キーから同じアイテムキーを参照できます。
 エイリアスは HostMetadataItem では使用できますが、HostnameItem パラメータでは使用できません。

例 1: ユーザー zabbix の ID を取得する。

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

これで、短縮キー zabbix.userid を使用してデータを取得できます。

例 2: デフォルトパラメータおよびカスタムパラメータを使用して CPU 使用率を取得する。

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

これにより、cpu.util キーを使用してデフォルトパラメータで CPU 使用率（パーセント）を取得できるほか、cpu.util[all, idle, avg15] を使用して CPU 使用率に関する特定のデータを取得できます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、各ルールに異なるパラメータを指定して、vfs.fs.discovery を使用する複数のディスカバリルールを設定できます。たとえば、vfs.fs.discovery[foo]、vfs.fs.discovery[bar] などです。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式であり、任意の数の任意の文字に一致する * 文字をサポートします。
DenyKey と組み合わせて複数のキー一致ルールを定義できます。DenyKey、AllowKeyRegexp、および DenyKeyRegexp と組み合わせて複数のキー一致ルールを定義できます。パラメータは、記載順に従って 1 つずつ処理されます。この順序によって、どの一致ルールが優先されるかが決まります。関連項目: エージェントチェックの制限。

AllowKeyRegexp

パターンに一致するアイテムキーの実行を許可します。キーのパターンには正規表現を使用します。
 DenyKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。詳細は、エージェントチェックの制限を参照してください。

例:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

バッファから Zabbix サーバー/プロキシへ値を送信する頻度を設定します（秒単位）。
 バッファがいっぱいの場合は、データはより早く送信されることに注意してください。

Default: 5
 Range: 1-3600

BufferSize

メモリバッファー内の値の最大数を設定します。
 バッファーがいっぱいになると、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。このパラメーターは、永続バッファーが無効 (EnablePersistentBuffer=0) の場合にのみ使用してください。

デフォルト: 1000
 範囲: 2-65535

ControlSocket

-R オプションを使用して実行時コマンドを送信するための制御ソケットを指定します。

デフォルト: /tmp/agent.sock

DebugLevel

デバッグレベルを設定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。
 キーのパターンは、任意の数の任意の文字に一致する * 文字をサポートするワイルドカード式です。
 複数のキー一致ルールを AllowKey と組み合わせて定義できます。複数のキー一致ルールを AllowKey、AllowKeyRegexp、DenyKeyRegexp と組み合わせて定義できます。パラメータは、記述された順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールが優先されるかが決まります。関連項目: エージェントチェックの制限。

デフォルト: DenyKey=system.run[*]

DenyKeyRegexp

パターンに一致するアイテムキーの実行を拒否します。キーのパターンには正規表現を使用します。
 AllowKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。
 別の system.run[*] ルールが指定されていない限り、すべての system.run[] アイテムはデフォルトで拒否されます（DenyKey=system.run[*]）。

関連項目：エージェントチェックの制限。

例：

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

EnablePersistentBuffer

アクティブアイテムに対してローカルの永続ストレージの使用を有効にします。
 永続ストレージが無効な場合は、メモリバッファーが使用されます。

デフォルト: 0
 値: 0 - 無効、1 - 有効

EnableProfiler

エージェントの起動時に、定期的な Go プロファイリングを有効にします。プロファイリングは、-R オプションを使用して実行時に有効または無効にすることもできます。periodic_prof_execute がプロファイルを収集している間は、すべてのプロファイラー実行時コマンドがビジー状態として拒否されます。

デフォルト: 0
 値: 0 - 無効、1 - 有効

ForceActiveChecksOnStart

再起動後、最初に受信した設定に対してアクティブチェックを直ちに実行します。
 プラグインごとの設定パラメーターとしても利用できます。例: Plugins.Uptime.System.ForceActiveChecksOnStart=1

デフォルト: 0
 値: 0 - 無効、1 - 有効

HeartbeatFrequency

ハートビートメッセージの送信間隔を秒単位で設定します。
 アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージは無効です。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホスト自動登録 時に使用するホストインターフェース（IP アドレスまたは DNS 名）を指定します。
 この値は、新しく作成されるホストのインターフェースに設定され、IP アドレスまたは DNS アドレスのいずれかを明示的に構成できます。
 詳細については、DNS をデフォルトのインターフェースとして使用する を参照してください。

定義されていない場合、この値は HostInterfaceItem から取得されます。

この値が 255 文字の上限を超えると、エージェントはエラーを出力して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストの自動登録時にホストインターフェース（IP アドレスまたは DNS 名）を決定するためのアイテムを指定します。
 この値は、HostInterface が定義されていない場合にのみ使用されます。
 詳細については、デフォルトインターフェースとして DNS を使用するを参照してください。

自動登録リクエスト中に、指定したアイテムから返された値が 255 文字を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[] アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録中に、ホストを識別または区別するために使用するメタデータを指定します。
 HostMetadata を使用すると、ホスト名以外の情報でホストを区別できます。

定義されていない場合、値は HostMetadataItem から取得されます。

指定した値が 2034 バイトの上限を超えている場合、または UTF-8 以外の文字列の場合、エージェントはエラーを発生させて起動しません。パラメーターが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効でも、IP アドレスまたは DNS 名として無効な値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。最初の改行で出力が切り詰められます。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータを取得するアイテムを指定します。
 このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録の試行時に取得され、ホストの自動登録処理でのみ使用されます。HostMetadataItem を使用すると、ホスト名以外の情報に基づいてホストを識別できます。

ユーザーパラメータとエイリアスがサポートされています。system.run[] アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

自動登録リクエスト中に、指定されたアイテムが返す値が 65535 UTF-8 コードポイントの上限を超えている場合、エージェントは警告メッセージをログに記録します。アイテムが返す値は UTF-8 文字列である必要があります。それ以外の場合、その値は無視されます。パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効であっても、有効な IP アドレスまたは DNS 名ではない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。出力は最初の改行で切り捨てられます。

ホスト名

一意で大文字小文字を区別するホスト名を指定します。カンマ区切りのリストです。
 アクティブチェックに必須であり、サーバーで設定されているホスト名と一致している必要があります。
 未定義の場合、値は HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大長: ホスト名ごとに 128 文字、行全体で 2048 文字です。

デフォルト: HostnameItem により設定されます

HostnameItem

ホスト名を取得するためのアイテムを指定します。
 このオプションは、Hostname が定義されていない場合にのみ使用されます。
 ユーザーパラメータまたはエイリアスはサポートされませんが、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の値に関係なく、system.run[] アイテムはサポートされます。

デフォルト: system.hostname

Include

設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
 Zabbix のインストール時、このパラメーターは /usr/local/etc に設定されますが、コンパイル時にこれを変更した場合はその設定が使用されます。

Include パラメーターに関する一般的なルール:

- パスは zabbix_agent2.conf ファイルの場所からの相対パスとして指定できます。
- 1 つの設定ファイル内で複数の Include パラメーターを指定できます。
- Zabbix は Include パラメーターを読み取るとすぐに処理するため、設定ファイルの末尾に配置することを検討してください。

含まれるファイルに関するルール:

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルは正しい構文でなければなりません。そうでない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンを含める際のルール:

- 含める順序を前提にしないでください（たとえば、ファイルはアルファベット順には含まれません）。
- 同じパラメーターを複数の含まれるファイルで定義しないでください（たとえば、一般的な設定を個別の設定で上書きする場合など）。どの値が使用されるか予測できないためです。
- 一部のテキストエディターが自動的に作成するバックアップコピーに注意してください(たとえば､include/my_specific.conf を編集すると include/my_specific.conf.BAK が作成される場合があります)。両方のファイルが含まれるためです。

ディレクトリ内の関連するファイルのみを含めるには、ワイルドカード (*) を使用してパターンマッチングを行います。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

サーバー/プロキシからの接続を受け付けるエージェントの待受 IP アドレスを指定します。カンマ区切りのリストです。
 最初の IP アドレスは、Zabbix サーバーに接続する場合に、アクティブチェックの一覧を取得するために送信されます。

デフォルト: 0.0.0.0

ListenPort

サーバー/プロキシからの接続を受け付けるエージェントの待受ポートを指定します。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイルを指定します。

デフォルト: /tmp/zabbix_agent2.log
 必須: はい。LogType が file に設定されている場合のみ。そうでない場合は不要です

LogFileSize

最大ログファイルサイズを MB 単位で設定します。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新規に開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力タイプを指定します: - *file* - LogFile パラメータで指定したファイルにログを書き込みます; - *system* - syslog にログを書き込みます; - *console* - 標準出力にログを書き込みます。

デフォルト: file

PersistentBufferFile

SQLite データベースを保持するためのファイルを指定します。完全なファイル名である必要があります。
 このパラメーターは、persistent buffer が有効な場合 (EnablePersistentBuffer=1) にのみ使用されます。

PersistentBufferPeriod

Zabbix サーバーまたはプロキシへの接続がない場合にデータを保存する期間を設定します。古いデータは失われます。
 ログデータは保持されます。このパラメーターは、persistent buffer が有効な場合のみ使用されます (EnablePersistentBuffer=1)。

デフォルト: 1h
 範囲: 1m-365d

PidFile

PID ファイルを指定します。

デフォルト: /tmp/zabbix_agent2.pid

プラグイン.<PluginName>.System.Capacity

同時に実行できる <PluginName> プラグインのチェック数の上限を設定します。

デフォルト: 1000 範囲: 1-1000

Plugins.Log.MaxLinesPerSecond

log または logrt のアクティブチェックを処理する際に、Zabbix サーバー/プロキシへ送信する 1 秒あたりの新しい行数の最大値を設定します。
 指定した値は、log および logrt のアイテムキーで指定される maxlines パラメータによって上書きされま

す。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、MaxLinesPerSecond で設定した値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

Plugins.SystemRun.LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。
 コマンドは、リモートで実行された場合にのみログに記録されます。
 system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

PluginSocket

ロード可能なプラグイン通信に使用する UNIX ソケットのパスを指定します。

デフォルト: /tmp/agent.plugin.sock

PluginTimeout

ロード可能なプラグインへの接続のタイムアウトを秒単位で指定します。

デフォルト: Timeout
 範囲: 1-30

ProfilerDir

プロファイリングデータが書き込まれるディレクトリ。

デフォルト: /tmp/zabbix_agent2_profiles

ProfilerMaxFilesPerProfile

各プロファイルタイプで保持するファイルの最大数です。この上限は、各プロファイルダンプ後に完了したファイルに適用されます。アクティブな CPU プロファイルは非表示の一時ファイルに収集されるため、上限にはカウントされません。periodic_prof_execute によって作成されたファイルも上限の対象となります。

デフォルト: 10
 範囲: 1-100

ProfilerInterval

プロファイリングデータの書き込み間隔（秒）。

デフォルト: 3600
 範囲: 1～86400

RefreshActiveChecks

アクティブチェックの一覧を更新する頻度を秒単位で指定します。
 アクティブチェックの更新に失敗した場合、次回の更新は 60 秒後に試行されることに注意してください。

デフォルト: 5
 範囲: 1-86400

サーバー

Zabbix サーバー/プロキシの IP アドレス、必要に応じて CIDR 表記、または DNS 名を指定します。カンマ区切りのリストです。
 受信接続は、ここに一覧表示されたホストからのみ受け入れられます。IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 により任意の IPv4 または IPv6 アドレスが許可されます。0.0.0.0/0 を使用すると、任意の IPv4 アドレスを許可できます。スペースも使用できます。このパラメータが指定されていない場合、パッシブチェックは無効になり、エージェントはどの TCP ポートでも待ち受けません。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレス、またはクラスタ構成を指定します。
 サーバー/プロキシのアドレスは IP アドレスまたは DNS 名で、必要に応じてコロンで区切ったポートを指定できます。
 クラスタ構成は、セミコロンで区切った 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマ区切りで指定できます。プロキシグループを使用しない場合、各 Zabbix サーバー/クラスタから 1 つを超える Zabbix プロキシを指定しないでください。Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定しないでください。
 複数のカンマ区切りアドレスを指定して、複数の独立した Zabbix サーバーを並列に使用できます。スペースは使用できます。
 ポートを指定しない場合は、デフォルトのポートが使用されます。
 IPv6 アドレスでは、そのホストのポートを指定する場合、角括弧で囲む必要があります。ポートを指定しない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータを指定しない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用する高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

次の接続に使用する送信元 IP アドレスを指定します。

- Zabbix サーバーまたは Zabbix プロキシへの送信接続。
- 一部のアイテム（`web.page.get`、`net.tcp.port` など）を実行する際の接続。

StatusPort

HTTP ステータスリクエスト（`http://localhost:<port>/status`）を受け付けるポートを指定します。

範囲: 1024-32767

タイムアウト

Zabbix プロキシまたはサーバーとの接続を確立し、データを交換するまでの待機時間を秒単位で指定します。

このパラメーターは、以下を含むさまざまな通信操作の継続時間を定義します。

- Zabbix サーバーからの応答を待機する;
- active checks におけるアイテム設定要求やアイテムデータを含む、Zabbix サーバーへの要求送信;
- logfile を通じたログデータの取得;
- ハートビートメッセージの送信;
- `vfs.*` チェックの最大継続時間;
- バージョン 7.0 より古いサーバーまたはプロキシがタイムアウトなしでチェックを送信するシナリオにおけるフォールバックとして機能する。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテムごとのレベル）には 使用されません。

既定値: 3
 範囲: 1-30

TLSAccept

受信接続に必要な暗号化レベルを指定します。
 パッシブチェックで使用されます。
 複数の値をカンマ区切りで指定できます: - *unencrypted* - 暗号化なしの接続を受け入れます（デフォルト）; - *psk* - TLS と事前共有鍵（PSK）を使用した接続を受け入れます; - *cert* - TLS と証明書を使用した接続を受け入れます。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合は必須です（*unencrypted* 接続の場合でも）。それ以外の場合はいいえ

TLSCAFile

ピア証明書の検証に使用する、最上位の CA 証明書を含むファイルのフルパスを指定します。これは、Zabbix コンポーネント間の暗号化通信で使用されます。

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

TLSCipherAll

OpenSSL の暗号スイート文字列（TLS 1.2）を指定します。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

OpenSSL 1.1.1 以降で使用する OpenSSL の暗号スイート文字列 (TLS 1.3) を指定します。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

OpenSSL の暗号スイート文字列（TLS 1.2）を指定します。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

このパラメーターは TLSAccept=cert,psk と同時には使用できないことに注意してください。証明書接続（TLSConnect=cert）では、代わりに TLSCipherAll を使用してください。

TLSCipherCert13

OpenSSL 1.1.1 以降で使用する OpenSSL の暗号スイート文字列 (TLS 1.3) を指定します。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

このパラメーターは TLSAccept=cert,psk と同時には使用できないことに注意してください。証明書接続 (TLSConnect=cert) には、代わりに TLSCipherAll13 を使用してください。

TLSCipherPSK

OpenSSL の暗号スイート文字列（TLS 1.2）を指定します。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

OpenSSL 1.1.1 以降で使用する OpenSSL の暗号文字列 (TLS 1.3) を指定します。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

エージェントから Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
 アクティブチェックで使用されます。
 指定できる値は 1 つだけです: - *unencrypted* - 暗号化なしで接続します（デフォルト）。- *psk* - TLS と事前共有キー (PSK) を使用して接続します。- *cert* - TLS と証明書を使用して接続します。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合（*unencrypted* 接続でも）; それ以外はいいえ

TLSCRLFile

失効した証明書を含むファイルのフルパスを指定します。このパラメーターは、Zabbix コンポーネント間の暗号化通信で使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用される、エージェントの秘密鍵を含むファイルの完全なパス名を指定します。

TLSPSKFile

Zabbix サーバーまたはプロキシとの暗号化通信に使用する、エージェントの事前共有鍵を含むファイルのフルパスを指定します。

TLSPSKIdentity

Zabbix サーバーまたはプロキシとの暗号化通信に使用する、事前共有鍵の ID 文字列を指定します。

TLSServerCertIssuer

許可するサーバー（プロキシ）証明書の発行者を指定します。

TLSServerCertSubject

許可されるサーバー（プロキシ）証明書のサブジェクトを指定します。

UnsafeUserParameters

ユーザー定義パラメータの引数に、すべての文字を渡せるようにします。次の文字は使用できません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
 さらに、改行文字も使用できません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータを指定します。ユーザー定義パラメータは複数指定できます。
 形式: UserParameter=<key>,<shell command>
 なお、シェルコマンドは空文字列または EOL のみを返してはなりません。UserParameterDir パラメータが指定されている場合、シェルコマンドには相対パスを使用できます。

例:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

UserParameter コマンドのデフォルト検索パスを指定します。使用すると、エージェントはコマンドを実行する前に作業ディレクトリをここで指定したディレクトリに変更します。これにより、UserParameter コマンドでは完全なパスの代わりに相対的な ./ プレフィックスを使用できます。
 指定できるエントリは 1 つだけです。

例:

```
UserParameterDir=/opt/myscripts
```

設定パラメーター: Zabbix エージェント 2 (Windows)

概要

設定パラメータを使用すると、Windows 上の Zabbix エージェント 2 の動作をカスタマイズできます。

パラメータ値は、Zabbix のエージェント 2 設定ファイル（zabbix_agent2.conf）で指定します。パラメータが必須であると明記されていない限り、すべてのパラメータは任意です。

このページは次の構成になっています。

- パラメータの概要（パラメータの詳細を表示するには、その名前をクリックしてください）
- パラメータの詳細

パラメータ	説明
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
AllowKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を許可します。
BufferSend	バッファーから Zabbix サーバーへ値を送信する頻度を設定します。
BufferSize	メモリバッファー内の値の最大数を設定します。
ControlSocket	-R オプションで実行時コマンドを送信するために使用するコントロールソケットを指定します。
DebugLevel	デバッグレベルを設定します。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
DenyKeyRegexp	正規表現ベースのパターンに一致するアイテムキーの実行を拒否します。
EnablePersistentBuffer	アクティブアイテム用のローカル永続ストレージの使用を有効にします。
EnableProfiler	エージェントの起動時に定期的な Go プロファイリングを有効にします。
ForceActiveChecksOnStart	最初に受信した設定に対して、再起動直後にアクティブチェックを実行します。
HeartbeatFrequency	ハートビートメッセージの間隔を秒単位で設定します。
HostInterface	ホストインターフェースを指定します。
HostInterfaceItem	ホストインターフェースを取得するためのアイテムを指定します。
HostMetadata	ホストメタデータを指定します。
HostMetadataItem	ホストメタデータを取得するために使用する Zabbix エージェントアイテムを指定します。
Hostname	ホスト名を指定します。
HostnameItem	ホスト名を取得するための Zabbix エージェントアイテムを指定します。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
ListenIP	サーバー/プロキシからの接続を受け付けるエージェントの待ち受け IP アドレスを指定します。
ListenPort	エージェントの待ち受けポートを指定します。
LogFile	エージェントのログファイルを指定します。
LogFileSize	ログファイルの最大サイズを設定します。

パラメータ	説明
LogType	ログ出力タイプを指定します。
PerfCounter	指定した期間 <period>（秒）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値を返す新しいパラメータ <parameter_name> を指定します。
PerfCounterEn	指定した期間 <period>（秒）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値を返す新しいパラメータ <parameter_name> を指定します。PerfCounter とは異なり、パフォーマンスカウンターのパスは英語で指定する必要があります。
PersistentBufferFile	Zabbix エージェント 2 が SQLite データベースを保持するファイルを指定します。
PersistentBufferPeriod	サーバーまたはプロキシに接続できない場合にデータを保存する期間を設定します。
Plugins.<PluginName>.System.Capacity	同時に実行できるプラグインごとのチェック数の上限を設定します。
Plugins.Log.MaxLinesPerSecond	`log`、`logrt`、`eventlog` のアクティブチェックの処理時に、エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行数の最大値を設定します。
Plugins.SystemRun.LogRemoteCommands	実行されたシェルコマンドを警告としてログに記録する機能を有効にします。
PluginSocket	ロード可能なプラグインとの通信に使用する Windows 名前付きパイプのパスを指定します。
PluginTimeout	ロード可能なプラグインとの接続のタイムアウトを秒単位で設定します。
ProfilerDir	プロファイリングデータを書き込むディレクトリです。
ProfilerMaxFilesPerProfile	プロファイルタイプごとに保持するファイルの最大数です。
ProfilerInterval	プロファイリングデータを書き込む間隔です。
RefreshActiveChecks	アクティブチェックの一覧を更新する頻度を指定します。
Server	Zabbix サーバー/プロキシの IP アドレスまたは DNS 名を指定します。
ServerActive	アクティブチェックを取得する Zabbix サーバー/プロキシのアドレスまたはクラスタ設定を指定します。
SourceIP	送信元 IP アドレスを指定します。
StatusPort	エージェントが HTTP ステータス要求（`http://localhost:<port>/status`）を待ち受けるポートを設定します。
Timeout	Zabbix サーバーまたはプロキシとの接続の確立およびデータ交換を待機する時間を秒単位で指定します。
TLSAccept	受信接続に必要な暗号化レベルを指定します。
TLSCAFile	ピア証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。このファイルは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	Zabbix コンポーネント間の暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
TLSCipherAll13	OpenSSL の暗号文字列（TLS 1.3）を指定します。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	OpenSSL の暗号文字列（TLS 1.3）を指定します。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	OpenSSL の暗号文字列（TLS 1.3）を指定します。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
TLSCRLFile	失効した証明書を含むファイルの完全なパス名を指定します。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	Zabbix コンポーネント間の暗号化通信に使用する、エージェントの秘密鍵を含むファイルの完全なパス名を指定します。
TLSPSKFile	Zabbix サーバーまたはプロキシとの暗号化通信に使用する、エージェントの事前共有鍵を含むファイルの完全なパス名を指定します。
TLSPSKIdentity	Zabbix サーバーまたはプロキシとの暗号化通信に使用する、事前共有鍵の識別文字列を指定します。
TLSServerCertIssuer	許可するサーバー（プロキシ）証明書の発行者を指定します。
TLSServerCertSubject	許可するサーバー（プロキシ）証明書のサブジェクトを指定します。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡せるようにします。
UserParameter	監視するユーザー定義パラメータを指定します。
UserParameterDir	`UserParameter` コマンドのデフォルト検索パスを指定します。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、プロセスのデフォルト値を示します。
- 値では環境変数を使用できます。
- Zabbix は、BOMなしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみ使用できます。

パラメーターの詳細

Alias

アイテムキーのエイリアスを設定します。長く複雑なアイテムキーを、より短く簡単なキーに置き換えるために使用できます。
 複数の Alias パラメータを指定できます。同じ Alias キーを持つパラメータを複数指定することはできません。
 異なる Alias キーから同じアイテムキーを参照できます。
 エイリアスは HostMetadataItem で使用できますが、HostnameItem パラメータでは使用できません。

例 1: サーバーからページファイルの使用率をパーセントで取得します。

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

これで、短縮キー pg_usage を使用してデータを取得できます。

例 2: デフォルトパラメータおよびカスタムパラメータを使用して CPU 負荷を取得します。

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

これにより、cpu.load キーを使用してデフォルトパラメータの CPU 負荷を取得できるほか、cpu.load[percpu,avg15] を使用して CPU 負荷に関する特定のデータを取得できます。

例 3: 同じディスカバリアイテムを処理する複数のローレベルディスカバリルールを実行します。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、各ルールに異なるパラメータを指定して、vfs.fs.discovery を使用する複数のディスカバリルールを設定できます。たとえば、vfs.fs.discovery[foo]、vfs.fs.discovery[bar] などです。

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーのパターンはワイルドカード式で、任意の数の任意の文字に一致する * 文字をサポートします。
 複数のキー一致ルールは、DenyKey、AllowKeyRegexp、DenyKeyRegexp と組み合わせて定義できます。パラメータは、記載順に 1 つずつ処理されます。この順序によって、どの一致ルールが優先されるかが決まります。関連項目: エージェントチェックの制限。

AllowKeyRegexp

パターンに一致するアイテムキーの実行を許可します。キーのパターンには正規表現を使用します。
 DenyKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。関連項目: エージェントチェックの制限。

例:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

バッファーから Zabbix サーバー/プロキシへ値を送信する頻度を設定します（秒単位）。
 バッファーがいっぱいの場合は、データはより早く送信されることに注意してください。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファー内の値の最大数を設定します。
 バッファーがいっぱいになった場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。
 このパラメーターは、永続バッファーが無効 (EnablePersistentBuffer=0) の場合にのみ使用してください。

デフォルト: 1000
 範囲: 2-65535

ControlSocket

-R オプションを使用して実行時コマンドを送信するために使用される制御ソケットを指定します。

デフォルト: \\.\pipe\agent.sock

DebugLevel

デバッグレベルを設定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。アイテムキーパターンは、任意の数の任意の文字に一致する * 文字をサポートするワイルドカード式です。
 AllowKey、AllowKeyRegexp、DenyKeyRegexp と組み合わせて、複数のアイテムキー一致

ルールを定義できます。パラメータは、記述された順序に従って 1 つずつ処理されます。どの一致ルールを優先するかは、その順序によって決まります。

関連項目: エージェントチェックの制限。

デフォルト: DenyKey=system.run[*]

DenyKeyRegexp

パターンに一致するアイテムキーの実行を拒否します。キーのパターンには正規表現を使用します。
 AllowKeyRegexp、AllowKey、DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは、記述されている順序に従って 1 つずつ処理されます。この順序によって、どの一致ルールを優先するかが決まります。
 別の system.run[*] ルールが指定されていない限り、すべての system.run[] アイテムはデフォルトで拒否されます（DenyKey=system.run[*]）。

関連項目: エージェントチェックの制限。

EnablePersistentBuffer

アクティブアイテムに対してローカルの永続ストレージの使用を有効にします。永続ストレージが無効な場合は、メモリバッファーが使用されます。

デフォルト: 0
 値: 0 - 無効、1 - 有効

EnableProfiler

エージェントの起動時に定期的な Go プロファイリングを有効にします。プロファイリングは、実行時に -R オプションを使用して有効または無効にすることもできます。periodic_prof_execute がプロファイルを収集している間、すべてのプロファイラー実行時コマンドはビジー状態として拒否されます。

デフォルト: 0
 値: 0 - 無効、1 - 有効

ForceActiveChecksOnStart

再起動後、最初に受信した設定に対してアクティブチェックを直ちに実行します。これはプラグインごとの設定パラメーターとしても利用できます。例: Plugins.Uptime.System.ForceActiveChecksOnStart=1

デフォルト: 0
 値: 0 - 無効、1 - 有効

HeartbeatFrequency

ハートビートメッセージの送信間隔を秒単位で設定します。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージを無効にします。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホスト自動登録 時に使用するホストインターフェース（IP アドレスまたは DNS 名）を指定します。この値は、新しく作成されるホストのインターフェースに設定され、IP アドレスまたは DNS アドレスのいずれかを明示的に構成できるようにします。詳細は、DNS をデフォルトのインターフェースとして使用する を参照してください。

定義されていない場合、値は HostInterfaceItem から取得されます。

値が 255 文字の上限を超えると、エージェントはエラーを出力して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストの自動登録時に、ホストインターフェース（IP アドレスまたは DNS 名）を決定するためのアイテムを指定します。
この値は、HostInterface が定義されていない場合にのみ使用されます。
詳細については、デフォルトインターフェースとして DNS を使用するを参照してください。

自動登録リクエスト中に、指定したアイテムから返された値が 255 文字を超えている場合、エージェントは警告メッセージをログに記録します。

system.run[] アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

HostMetadata

ホストの自動登録（アクティブエージェント）中に、ホストを識別または区別するために使用するメタデータを指定します。
HostMetadata を使用すると、ホスト名以外の情報でホストを区別できます。

定義されていない場合、値は HostMetadataItem から取得されます。

指定した値が 2034 バイトを超えている場合、または UTF-8 以外の文字列の場合、エージェントはエラーを出力して起動しません。
パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効でも、有効な IP アドレスまたは DNS 名ではない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。最初の改行で出力が切り捨てられます。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータを取得する Zabbix エージェントのアイテムを指定します。
このオプションは、HostMetadata が定義されていない場合にのみ使用されます。

HostMetadataItem の値は、各自動登録の試行時に取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。
HostMetadataItem を使用すると、ホスト名以外の情報に基づいてホストを識別できます。

ユーザーパラメータとエイリアスがサポートされています。
system.run[]アイテムは、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の設定に関係なくサポートされます。

自動登録リクエスト中に、指定したアイテムが返す値が 65535 UTF-8 コードポイントの上限を超えている場合、エージェントは警告メッセージをログに記録します。
アイテムが返す値は UTF-8 文字列である必要があります。それ以外の場合、その値は無視されます。
パラメータが IP アドレスまたは DNS 名を想定している場合、UTF-8 として有効であっても、有効な IP アドレスまたは DNS 名ではない値は拒否され、無効として報告されます。

複数行のメタデータはサポートされません。出力は最初の改行で切り捨てられます。

ホスト名

一意で、大文字小文字を区別するホスト名を指定します。カンマ区切りのリストです。
 アクティブチェックに必要であり、サーバーで設定されたホスト名と一致している必要があります。値が未定義の場合は HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大長: ホスト名ごとに 128 文字、行全体で 2048 文字。

デフォルト: HostnameItem により設定されます

HostnameItem

ホスト名を取得するためのアイテムを指定します。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータまたはエイリアスはサポートされませんが、AllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexp の値に関係なく、system.run[] アイテムはサポートされます。

デフォルト: system.hostname

Include

構成ファイルに含めるディレクトリまたは個別のファイルを指定します（Zabbix エージェントを Windows MSI インストーラーパッケージを使用してインストールした場合、デフォルトでは C:\Program Files\Zabbix Agent 2 にあります。Zabbix エージェントを zip アーカイブとしてインストールした場合は、インストール時に指定したフォルダーにあります）。Zabbix のインストール時、このパラメータは、コンパイル時に変更しない限り、/usr/local/etc に設定されます。

Include パラメータの一般的なルール：

- パスは zabbix_agent2.conf ファイルの場所を基準にした相対パスにできます(例：Include=.\zabbix_agent2.d\plugins.d*
- 1 つの構成ファイルに複数の Include パラメータを指定できます。
- Zabbix は Include パラメータを読み取るとすぐに処理するため、構成ファイルの末尾に配置することを検討してください。

インクルードするファイルのルール：

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルの構文は正しくなければなりません。そうでない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンをインクルードする場合のルール：

- インクルードの順序を想定しないでください（ファイルはアルファベット順にインクルードされるとは限りません）。
- 同じパラメータを複数のインクルードファイルで定義しないでください（例：一般的な設定を特定の設定で上書きする場合）。どの値が使用されるかを予測できないためです。
- 一部のテキストエディターが自動的に作成するバックアップコピーに注意してください（例：include/my_specific.conf を編集すると、include/my_specific.conf.BAK が作成される場合があります）。この場合、両方のファイルがインクルードされます。

ディレクトリから関連するファイルのみをインクルードするには、パターンマッチングにワイルドカード（*）を使用します。

例：

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

サーバー/プロキシからの接続を受け付けるエージェントの待受 IP アドレスを指定します。カンマ区切りのリストです。
 最初の IP アドレスは、アクティブチェックの一覧を取得するためにサーバーへ接続する場合、Zabbix サーバーに送信されます。

デフォルト: `0.0.0.0`

ListenPort

サーバー/プロキシからの接続を受け付けるエージェントの待受ポートを指定します。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイルを指定します。

デフォルト: `c:\zabbix_agent2.log`
 必須: LogType が `file` に設定されている場合は Yes、それ以外は no

LogFileSize

最大ログファイルサイズを MB 単位で設定します。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新規に開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力タイプを指定します。- *file* - `LogFile` パラメーターで指定したファイルにログを書き込みます。- *console* - 標準出力にログを書き込みます。

デフォルト: `file`

PerfCounter

指定した時間期間 <period>（秒）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値となる新しいパラメーター <parameter_name> を定義します。
 構文: <parameter_name>,”<perf_counter_path>”,<period>

たとえば、直近 1 分間の 1 秒あたりのプロセッサ割り込みの平均回数を取得したい場合は、次のように新しいパラメーター”interrupts” を定義できます。

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンターのパスは二重引用符で囲む必要があることに注意してください。パラメーター名（interrupts）は、アイテムを作成する際のアイテムキーとして使用します。平均値を計算するためのサンプルは 1 秒ごとに取得されます。
Windows で利用可能なすべてのパフォーマンスカウンターの一覧を取得するには、`typeperf -qx` を実行してください。

PerfCounterEn

新しいパラメーター <parameter_name> を定義します。これは、指定した期間 <period>（秒単位）におけるシステムパフォーマンスカウンター <perf_counter_path> の平均値です。
PerfCounter と比較して、perfcounter のパスは英語で指定する必要があります。
Windows Server 2008/Vista 以降でのみサポートされます。
 構文: <parameter_name>,”<perf_counter_path>”,<period>

たとえば、直近 1 分間の 1 秒あたりのプロセッサー割り込みの平均回数を取得したい場合は、次のように新しいパラメーター”interrupts” を定義できます。

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

パフォーマンスカウンターのパスを二重引用符で囲むことに注意してください。
パラメーター名（interrupts）は、アイテムを作成する際のアイテムキーとして使用します。
平均値の計算用サンプルは 1 秒ごとに取得されます。
 英語の文字列の一覧は、次のレジストリキーを参照してください: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`。

PersistentBufferFile

Zabbix agent 2 が SQLite データベースを保持するファイルを指定します。完全なファイル名である必要があります。
 このパラメータは、persistent buffer が有効な場合 (`EnablePersistentBuffer=1`) にのみ使用されます。

PersistentBufferPeriod

サーバーまたはプロキシへの接続がない場合にデータを保存する期間を設定します。古いデータは失われます。ログデータは保持されます。
 このパラメーターは、persistent buffer が有効な場合 (`EnablePersistentBuffer=1`) にのみ使用されます。

デフォルト: 1h
 範囲: 1m-365d

プラグイン.<PluginName>.System.Capacity

同時に実行できる `<PluginName>` プラグインのチェック数の上限を設定します。

デフォルト: 1000 範囲: 1-1000

Plugins.Log.MaxLinesPerSecond

log、logrt、eventlog のアクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシへ 1 秒あたりに送信する新しい行の最大数を設定します。指定した値は、log、logrt または eventlog アイテムキーで指定された maxlines パラメーターによって上書きされます。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、MaxLinesPerSecond で設定された値の 10 倍の新しい行を処理します。

デフォルト: 20
 範囲: 1-1000

Plugins.SystemRun.LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。コマンドは、リモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

PluginSocket

ロード可能なプラグイン通信に使用する Windows の名前付きパイプのパスを指定します。

デフォルト: \\.\pipe\agent.plugin.sock

PluginTimeout

ロード可能なプラグインとの接続のタイムアウトを秒単位で設定します。

デフォルト: Timeout
 範囲: 1-30

ProfilerDir

プロファイリングデータが書き込まれるディレクトリ。

デフォルト: c:\zabbix_pprof

ProfilerMaxFilesPerProfile

各プロファイルタイプで保持するファイルの最大数です。この制限は、各プロファイルダンプ後に完了したファイルに適用されます。アクティブな CPU プロファイルは非表示の一時ファイルに収集されるため、制限の対象にはなりません。periodic_prof_execute によって作成されたファイルも制限の対象です。

デフォルト: 10
 範囲: 1-100

ProfilerInterval

プロファイリングデータの書き込み間隔（秒）。

デフォルト: 3600
 範囲: 1～86400

RefreshActiveChecks

アクティブチェックの一覧を更新する頻度を秒単位で指定します。
 アクティブチェックの更新に失敗した場合、次回の更新は 60 秒後に試行されることに注意してください。

デフォルト: 5
 範囲: 1-86400

サーバー

Zabbix サーバー/プロキシの IP アドレス、必要に応じて CIDR 表記、または DNS 名を指定します。カンマ区切りのリストです。
 このパラメーターが指定されていない場合、パッシブチェックは無効になり、エージェントは TCP ポートで待ち受けしません。
 受信接続は、ここに列挙されたホストからのみ受け入れられます。IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 は任意の IPv4 または IPv6 アドレスを許可します。0.0.0.0/0 を使用すると、任意の IPv4 アドレスを許可できます。スペースも使用できます。

例:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレス、またはクラスタ構成を指定します。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名で、必要に応じてコロンで区切ったポートを指定します。ポートが指定されていない場合は、デフォルトのポートが使用されます。
 IPv6 アドレスでは、そのホストのポートを指定する場合、角括弧で囲む必要があります。ポートが指定されていない場合、IPv6 アドレスの角括弧は省略可能です。

クラスタ構成は、セミコロンで区切られた 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。

複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマ区切りで指定できます。

プロキシグループを使用しない場合、各 Zabbix サーバー/クラスタから 2 つ以上の Zabbix プロキシを指定しないでください。Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定しないでください。

カンマ区切りの複数アドレスを指定して、複数の独立した Zabbix サーバーを並列に使用できます。スペースは使用可能です。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用した高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

次の接続に使用する送信元 IP アドレスを指定します。

- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 一部のアイテム（`web.page.get`、`net.tcp.port` など）を実行する際の接続

StatusPort

エージェントが HTTP ステータス要求 (`http://localhost:<port>/status`) を待ち受けるポートを設定します。

範囲: 1024-32767

タイムアウト

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間を、秒単位で指定します。

このパラメーターは、次のような各種通信操作の継続時間を定義します。

- Zabbix サーバーからの応答を待機する;
- アイテム設定要求やactive checks におけるアイテムデータを含む、Zabbix サーバーへの要求送信;
- logfile または Windows イベントログ監視を通じたログデータの取得;
- ハートビートメッセージの送信;
- `vfs.*` チェックの最大実行時間;
- バージョン 7.0 より古いサーバーまたはプロキシがタイムアウトなしでチェックを送信するシナリオにおけるフォールバックとしての使用。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテムごとのレベル）には 使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受信接続に必要な暗号化レベルを指定します。
 パッシブチェックで使用されます。

複数の値を指定でき、カンマで区切ります。

- *unencrypted* - 暗号化なしの接続を受け入れます（デフォルト）。
- *psk* - TLS と事前共有鍵（PSK）を使用した接続を受け入れます。
- *cert* - TLS と証明書を使用した接続を受け入れます。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合は必須です（*unencrypted* 接続の場合でも）。それ以外の場合はいいえ

TLSCAFile

ピア証明書の検証に使用する、最上位の CA 証明書を含むファイルのフルパスを指定します。これは、Zabbix コンポーネント間の暗号化通信で使用されます。

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用する、エージェントの証明書または証明書チェーンを含むファイルの完全なパス名を指定します。

TLSCipherAll13

OpenSSL の暗号スイート文字列（TLS 1.3）を指定します。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

OpenSSL の暗号スイート文字列（TLS 1.3）を指定します。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

このパラメータは `TLSAccept=cert,psk` と同時には使用できません。証明書接続（`TLSConnect=cert`）では、代わりに `TLSCipherAll13` を使用してください。

TLSCipherPSK13

OpenSSL の暗号スイート文字列 (TLS 1.3) を指定します。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

エージェントから Zabbix サーバーまたはプロキシへの接続に必要な暗号化レベルを指定します。
 アクティブチェックで使用されます。
 指定できる値は 1 つだけです: - *unencrypted* - 暗号化なしで接続します（デフォルト）。- *psk* - TLS と事前共有キー（PSK）を使用して接続します。- *cert* - TLS と証明書を使用して接続します。

必須: はい。TLS 証明書または PSK パラメータが定義されている場合（*unencrypted* 接続でも）は必須です。それ以外の場合は不要です。

TLSCRLFile

失効した証明書を含むファイルのフルパスを指定します。このパラメーターは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用される、エージェントの秘密鍵を含むファイルの完全なパス名を指定します。

TLSPSKFile

エージェントの事前共有鍵を含むファイルの完全なパス名を指定します。この鍵は、Zabbix サーバーまたはプロキシとの暗号化通信に使用されます。

TLSPSKIdentity

Zabbix サーバーまたはプロキシとの暗号化通信に使用する、事前共有鍵の ID 文字列を指定します。

TLSServerCertIssuer

許可するサーバー（プロキシ）証明書の発行者を指定します。

TLSServerCertSubject

許可されるサーバー（プロキシ）証明書のサブジェクトを指定します。

UnsafeUserParameters

ユーザー定義パラメータの引数に、すべての文字を渡せるようにします。次の文字は使用できません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
 さらに、改行文字も使用できません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータを指定します。ユーザー定義パラメータは複数指定できます。
 形式: `UserParameter=<key>,<shell command>`
 なお、シェルコマンドは空文字列または EOL のみを返してはなりません。`UserParameterDir` パラメータが指定されている場合、シェルコマンドには相対パスを使用できます。

例:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

UserParameter コマンドのデフォルト検索パスを指定します。使用すると、エージェントはコマンドを実行する前に作業ディレクトリをここで指定したディレクトリに変更します。これにより、UserParameter コマンドでは完全なパスの代わりに相対的な .\ プレフィックスを使用できます。
 指定できるエントリは 1 つだけです。

例:

```
UserParameterDir=C:\opt\myscripts
```

設定パラメータ: プラグイン

Zabbix エージェント 2 プラグインの設定パラメータについては、次を参照してください:

- Ceph プラグイン
- Docker プラグイン
- Ember+ プラグイン
- Memcached プラグイン
- Modbus プラグイン
- MongoDB プラグイン
- MQTT プラグイン
- MSSQL プラグイン
- MySQL プラグイン
- NVIDIA GPU プラグイン
- Oracle プラグイン
- PostgreSQL プラグイン
- Redis プラグイン
- S.M.A.R.T. プラグイン

プラグイン設定の原則

このページでは、プラグインの設定原則とベストプラクティスについて説明します。

すべてのプラグインは Plugins.* パラメータを使用して設定します。このパラメータは、Zabbix エージェント 2 の設定ファイルまたはプラグインの設定ファイルの一部として指定できます。プラグインが個別の設定ファイルを使用する場合は、Zabbix エージェント 2 の設定ファイルの Include パラメータにこのファイルへのパスを指定する必要があります。

一般的なプラグインパラメータは、次の構造になります。

Plugins.<PluginName>.<Parameter>=<Value>

また、次の 2 つの特殊なパラメータグループがあります。

- Plugins.<PluginName>.Default.<Parameter>=<Value> は、デフォルトのパラメータ値を定義するために使用します。
- Plugins.<PluginName>.<SessionName>.<Parameter>=<Value> は、名前付きセッションを使用して、異なる監視対象に対する個別のパラメータセットを定義するために使用します。

すべてのパラメータ名は、次の要件に従う必要があります。

- プラグイン名は大文字で始めることを推奨します。
- パラメータは大文字で始める必要があります。
- 特殊文字は使用できません。
- ネストの最大レベルに制限はありません。
- パラメータ数に制限はありません。

たとえば、Uptime プラグインに対してのみ、エージェントの再起動直後に *Scheduling* 更新間隔を持たないアクティブチェックを実行するには、設定ファイルに Plugins.Uptime.System.ForceActiveChecksOnStart=1 を設定します。同様に、CPU プラグインの同時チェックにカスタムの上限を設定するには、設定ファイルに Plugins.CPU.System.Capacity=N を設定します。

デフォルト値

接続関連のパラメータ（URI、ユーザー名、パスワードなど）のデフォルト値は、設定ファイルで次の形式で設定できます。

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

たとえば、Plugins.Mysql.Default.Username=zabbix、Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017 などです。

このようなパラメータの値がアイテムキーまたは名前付きセッションのパラメータで指定されていない場合、プラグインはデフォルト値を使用します。デフォルトパラメータも定義されていない場合は、ハードコードされたデフォルト値が使用されます。

> **Note:**
> アイテムキーにパラメータがない場合、Zabbix エージェント 2 は、デフォルトパラメータセクションで定義された値を使用してメトリクスの収集を試みます。

名前付きセッション

名前付きセッションは、プラグインパラメータの追加レベルを表し、監視対象の各インスタンスごとに個別の認証パラメータセットを指定するために使用できます。
各名前付きセッションパラメータは、次の構造にする必要があります。

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

セッション名は、URI、ユーザー名、パスワードを個別に指定する代わりに、connString のアイテムキーパラメータとして使用できます。

アイテムキーでは、最初のパラメータは connString または URI のいずれかにできます。
最初のキーパラメータがいずれのセッション名にも一致しない場合は、URI として扱われます。
アイテムキーで埋め込み URI 認証情報を渡すことはサポートされていないため、代わりに名前付きセッションパラメータを使用してください。

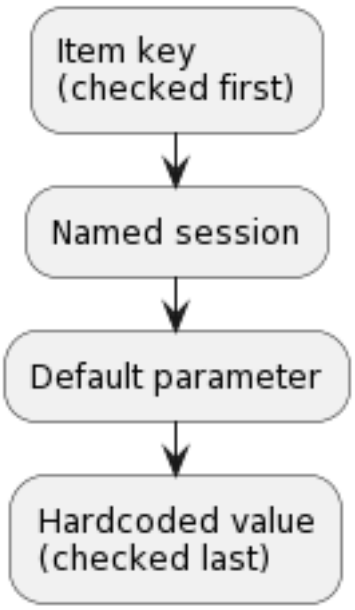
利用可能な名前付きセッションパラメータ の一覧は、プラグインによって異なります。

アイテムキーパラメータで新しい値を指定することで、セッションパラメータを上書きできます（例 を参照）。

パラメータが名前付きセッションに定義されていない場合、Zabbix エージェント 2 はデフォルトのプラグインパラメータ で定義された値を使用します。

パラメーターの優先順位

Zabbix エージェント 2 のプラグインは、接続関連のパラメーター値を次の順序で検索します。



1. 最初のアイテムキーのパラメーターは、セッション名と比較されます。
   一致しない場合は、実際の値として扱われます。この場合、手順 3 はスキップされます。
   一致した場合、パラメーター値（通常は URI）は、その名前付きセッションで定義されている必要があります。
2. その他のパラメーターは、定義されていればアイテムキーから取得されます。
3. アイテムキーのパラメーター（たとえば password）が空の場合、プラグインは対応する名前付きセッションのパラメーターを探します。
4. セッションパラメーターも指定されていない場合は、対応するデフォルトパラメーターで定義された値が使用されます。
5. それでも該当しない場合、プラグインはハードコードされたデフォルト値を使用します。

例 1

”MySQL1” と”MySQL2” の 2 つのインスタンスの監視。

設定パラメータ:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.1:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

この設定の結果、各セッション名はアイテムキー の connString として使用できます。たとえば、`mysql.ping[MySQL1]` または `mysql.ping[MySQL2]` です。

例 2

アイテムキーで一部のパラメータを指定する場合。

設定パラメータ:

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

アイテムキー: `pgsql.ping[session1,new_username,,postgres]`

この設定の結果、エージェントは次のパラメータを使用して PostgreSQL に接続します。

- セッションパラメータからの URI: 192.0.2.234:5432
- アイテムキーからのユーザー名: new_username
- セッションパラメータからのパスワード（アイテムキーでは省略されているため）: session_password
- アイテムキーからのデータベース名: postgres

例 3

デフォルトの設定パラメータを使用してメトリックを収集します。

設定パラメータ:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

アイテムキー: `pgsql.ping[,,,postgres]`

この設定の結果、エージェントは次のパラメータを使用して PostgreSQL に接続します。

- デフォルト URI: 192.0.2.234:5432
- デフォルトのユーザー名: zabbix
- デフォルトのパスワード: password
- アイテムキーからのデータベース名: postgres

接続

一部のプラグインは、複数のインスタンスから同時にメトリクスを収集することをサポートしています。ローカルおよびリモートのインスタンスの両方を監視できます。TCP および Unix ソケット接続がサポートされています。

プラグインは、インスタンスへの接続をオープン状態に保つように設定することを推奨します。これにより、接続数が少なくなるため、ネットワークの混雑、遅延、CPU およびメモリ使用量を削減できます。この処理はクライアントライブラリが担当します。

> **Note:**
> 未使用の接続をオープン状態のまま維持する期間は、`Plugins.<PluginName>.KeepAlive` パラメータで指定できます。
> 例: `Plugins.Memcached.KeepAlive`

Ceph プラグイン

概要

設定パラメータを使用すると、Ceph プラグインの動作をカスタマイズできます。

これらは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: `ceph.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

すべてのパラメータは、パラメータが必須であると明示されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を確認するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.Ceph.Default.ApiKey	Ceph への接続に使用するデフォルトの API キーを指定します。
Plugins.Ceph.Default.Mode	Ceph への接続に使用するデフォルトのモードを指定します。
Plugins.Ceph.Default.URI	Ceph への接続に使用するデフォルトの URI を指定します。
Plugins.Ceph.Default.User	Ceph への接続に使用するデフォルトのユーザー名を指定します。
Plugins.Ceph.InsecureSkipVerify	HTTP クライアントがサーバーの証明書チェーンとホスト名を検証するかどうかを指定します。
Plugins.Ceph.KeepAlive	未使用のプラグイン接続が閉じられるまでの最大待機時間を設定します。
Plugins.Ceph.Sessions.<SessionName>.ApiKey	Ceph への接続に使用する、名前付きセッションの API キーを指定します。
Plugins.Ceph.Sessions.<SessionName>.Mode	Ceph への接続に使用する、名前付きセッションのモードを指定します。
Plugins.Ceph.Sessions.<SessionName>.URI	Ceph への接続に使用する、名前付きセッションの URI を指定します。
Plugins.Ceph.Sessions.<SessionName>.User	Ceph への接続に使用する、名前付きセッションのユーザー名を指定します。
Plugins.Ceph.Timeout	リクエスト実行のタイムアウトを設定します。

注意事項:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値には環境変数を使用できます。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Ceph.Default.ApiKey

Ceph に接続するためのデフォルトの API キーを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Ceph.Default.Mode

Ceph への接続に使用するデフォルトモードを指定します。
 ネイティブモードは Linux でのみサポートされ、Ceph 16 以降で利用できます。restful モードは Ceph バージョン 20 以降では動作しません。
 各モードで必要なユーザー認証情報のセットは異なり、互換性がないことに注意してください。

デフォルト: `restful` (deprecated)
 範囲: native/restful

Plugins.Ceph.Default.URI

Ceph に接続するためのデフォルト URI を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

 埋め込み認証情報を含めないでください（無視されます）。
URI 形式に一致している必要があります。
https スキームのみサポートされており、スキームは省略できます。
 ポートは省略可能です（default=8003）。

デフォルト: `https://localhost:8003`

例:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Ceph に接続するためのデフォルトのユーザー名を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Ceph.InsecureSkipVerify

HTTP クライアントがサーバーの証明書チェーンとホスト名を検証するかどうかを指定します。
 true の場合、TLS はサーバーが提示した任意の証明書と、その証明書内の任意のホスト名を受け入れます。このモードでは、TLS は中間者攻撃を受けやすくなります（テスト用途にのみ使用してください）。

デフォルト: `false`
 範囲: false/true

Plugins.Ceph.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）を設定します。

デフォルト: 300
 範囲: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Ceph に接続するための、指定したセッションの API キーを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Ceph.Sessions.<SessionName>.Mode

Ceph に接続するための、名前付きセッションモードを指定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。
 native モードは Linux でのみサポートされ、Ceph 16 以降で利用できます。restful モードは Ceph バージョン 20 以降では動作しません。
 ユーザー認証情報のセットはモードごとに異なり、互換性がないことに注意してください。

デフォルト: `restful` (deprecated)
 範囲: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Ceph に接続するための、名前付きセッション URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込み資格情報を含めないでください（無視されます）。
URI 形式に一致している必要があります。
 `https` スキームのみサポートされます。スキームは省略できます。
 ポートは省略できます（デフォルト =8003）。

デフォルト: `https://localhost:8003`

例:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Ceph に接続するための、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Ceph.Timeout

リクエスト実行のタイムアウト（リクエストが完了するまで待機してから終了するまでの時間）を秒単位で設定します。

デフォルト: `global timeout`
 範囲: 1-30

関連情報

- プラグインの設定手順
- Zabbix エージェント 2 の一般的な設定パラメータの説明: Zabbix エージェント 2 (UNIX) / Zabbix エージェント 2 (Windows)

Docker プラグイン

概要

設定パラメータを使用すると、Docker プラグインの動作をカスタマイズできます。

これらは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: `docker.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

すべてのパラメータは、明示的に必須と記載されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、その名前をクリックします）
- パラメータ詳細

Parameter	Description
Plugins.Docker.Endpoint	Docker デーモンの UNIX ソケットの場所を指定します。
Plugins.Docker.Timeout	リクエスト実行のタイムアウトを指定します。

次の点に注意してください。

- デフォルト値は、同梱されている設定ファイルの値ではなく、プロセスのデフォルト値を反映します。
- 値は環境変数 をサポートします。
- Zabbix は、BOM を含まない UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Docker.Endpoint

Docker デーモンの UNIX ソケットの場所を指定します。
 スキームを含める必要があります（unix:// のみサポートされています）。

デフォルト: unix:///var/run/docker.sock

Plugins.Docker.Timeout

リクエスト実行のタイムアウトを指定します（リクエストが完了するまで待機し、その後停止するまでの秒数）。

デフォルト: global timeout
 範囲: 1–30

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

Ember+ プラグイン

概要

設定パラメータを使用すると、Ember+ プラグインの動作をカスタマイズできます。

これらは agent 2 の設定ファイルではなく、専用の設定ファイル（例: emberplus.conf）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix agent 2 の設定に含めます。

Ember+ プラグインはロード可能なプラグインであり、Ember+ plugin repository で利用可能で、詳細もそこに記載されています。

このプラグインは現在、ソースからのみビルド可能です（Unix と Windows の両方）。

特に明記されていない限り、すべてのパラメータは任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.EmberPlus.Default.Uri	Ember+ への接続に使用するデフォルトの URI を指定します。
Plugins.EmberPlus.KeepAlive	未使用のプラグイン接続を切断するまでの最大待機時間を設定します。
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Ember+ への接続に使用する、名前付きセッションの URI を指定します。
Plugins.EmberPlus.System.Path	Ember+ プラグイン実行ファイルへのパスを指定します。
Plugins.EmberPlus.Timeout	リクエスト実行のタイムアウトを設定します。

注意事項:

- デフォルト値は、配布される設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.EmberPlus.Default.Uri

Ember+ への接続に使用するデフォルトの URI を指定します。
 サポートされているスキームは tcp:// のみです。スキームは省略できます。埋め込み資格情報は無視されます。

デフォルト: tcp://localhost:9998

Plugins.EmberPlus.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間を秒単位で設定します。

デフォルト: 300
 範囲: 60–900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Ember+ に接続するための、名前付きセッションの URI を指定します。
 サポートされているスキーマは tcp:// のみです。スキーマは省略できます。埋め込み資格情報は無視されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

デフォルト: tcp://localhost:9998

Plugins.EmberPlus.System.Path

Ember+ プラグインの実行ファイルへのパスを指定します。

例: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

最初の接続時およびセッション内の後続の操作で、サーバーからの応答を待機する時間を秒単位で設定します。

デフォルト: `global timeout`
 範囲: 1–30

オプション

Parameter	Description
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テスト用にプラグインを起動します（プラグイン設定は無視されます）。

参照:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

Memcached プラグイン

概要

設定パラメータを使用すると、Memcached プラグインの動作をカスタマイズできます。

これらは agent 2 の設定ファイルではなく、専用の設定ファイル（例: `memcached.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix agent 2 の設定に含めます。

すべてのパラメータは、パラメータが必須であると明記されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を確認するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.Memcached.Default.Password	Memcached への接続に使用するデフォルトのパスワードを指定します。
Plugins.Memcached.Default.Uri	Memcached への接続に使用するデフォルトの URI を指定します。
Plugins.Memcached.Default.User	Memcached への接続に使用するデフォルトのユーザー名を指定します。
Plugins.Memcached.KeepAlive	未使用のプラグイン接続が閉じられるまでの最大待機時間を設定します。
Plugins.Memcached.Sessions.*SessionName*.Password	Memcached への接続に使用する、名前付きセッションのパスワードを指定します。
Plugins.Memcached.Sessions.*SessionName*.Uri	Memcached への接続に使用する、名前付きセッションの URI を指定します。
Plugins.Memcached.Sessions.*SessionName*.User	Memcached への接続に使用する、名前付きセッションのユーザー名を指定します。
Plugins.Memcached.Timeout	リクエスト実行のタイムアウトを設定します。

次の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値には環境変数を使用できます。
- Zabbix がサポートする設定ファイルの文字エンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーター詳細

Plugins.Memcached.Default.Password

Memcached に接続するためのデフォルトパスワードを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Memcached.Default.Uri

Memcached に接続するためのデフォルト URI を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
 埋め込み認証情報は無視されます。
 URI 形式に一致している必要があります。サポートされるスキーム: `tcp`, `unix`。
 スキームは省略できます。
 ポートは省略できます（デフォルト =11211）。

デフォルト: tcp://localhost:11211

例:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Memcached に接続するためのデフォルトのユーザー名を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Memcached.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間を秒単位で設定します。

デフォルト: 300
 範囲: 60–900

Plugins.Memcached.Sessions.<SessionName>.Password

Memcached に接続するための、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Memcached.Sessions.<SessionName>.Uri

Memcached に接続するための、名前付きセッション URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込み認証情報は無視されます。
 URI 形式に一致している必要があります。サポートされるスキーム: tcp, unix。
 スキームは省略できます。
 ポートは省略できます（default=11211）。

例:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Sessions.<SessionName>.User

Memcached に接続するための、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Memcached.Timeout

リクエストの実行タイムアウトを設定します（リクエストが完了するまで待機し、その後に終了するまでの時間を秒単位で指定します）。

デフォルト: global timeout
 範囲: 1–30

関連項目:

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

Modbus プラグイン

概要

設定パラメータを使用すると、Modbus プラグインの動作をカスタマイズできます。

これらは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: modbus.conf）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

特に明記されていない限り、すべてのパラメータは任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を確認するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Modbus に接続するための、名前付きセッションのエンドポイントを指定します。
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Modbus に接続するための、名前付きセッションのスレーブ ID を指定します。
Plugins.Modbus.Sessions.<SessionName>.Timeout	名前付きセッションのタイムアウトを秒単位で設定します。

次の点に注意してください。

- デフォルト値は、同梱されている設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。
- Zabbix がサポートする設定ファイルの文字エンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Modbus に接続するための、名前付きセッションのエンドポイントを指定します。
 接続文字列は、プロトコルスキーム、ホストアドレス、およびポートまたはシリアルポート名と属性で構成されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Modbus に接続するための、名前付きセッションのスレーブ ID を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

例: `Plugins.Modbus.Sessions.MB1.SlaveID=20`

注この名前付きセッションパラメータは、アイテムキー の slave ID パラメータに指定された値が空の場合にのみ確認されます。

Plugins.Modbus.Sessions.<SessionName>.Timeout

指定したセッションのタイムアウトを秒単位で設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

例: `Plugins.Modbus.Sessions.MB1.Timeout=2`

リクエスト実行タイムアウト（リクエストが完了するまで待機し、その後停止するまでの時間）を設定する必要がある場合は、アイテム設定 フォームを使用してください。

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

MongoDB プラグイン

概要

設定パラメータを使用すると、MongoDB プラグインの動作をカスタマイズできます。

これらのパラメータは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: `mongodb.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

MongoDB プラグインはロード可能なプラグインであり、MongoDB plugin repository にて利用可能で、詳細が完全に記載されています。

特に明記されていない限り、すべてのパラメータは任意です。

MongoDB に接続する際、プラグインは設定を特定の順序で確認し、どの値を使用するかを決定します。優先順位の高い順から低い順まで、次のとおりです。

1. 接続 URI で指定されたパラメータ（例: `Plugins.MongoDB.Sessions.<session_name>.Uri` の ? の後に定義されたクエリパラメータが最優先）。
2. Zabbix エージェント 2 のアイテムキー内のパラメータ（例: mongodb.collection.stats）。
3. 名前付きセッションで定義されたパラメータ(例: `Plugins.MongoDB.Sessions.<SessionName>.User`、`Plugins.MongoDB.Sessi`
4. デフォルト値（例: `Plugins.MongoDB.Default.User`、`Plugins.MongoDB.Default.Password`）。

例:

- MongoDB サーバーへの接続を認証するためのデータベースを `*.<SessionName>.Uri` パラメータで指定した場合（例: `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=admin`）、`*.<SessionName>.AuthSource` パラメータに別の値が設定されていても、プラグインは `*.<SessionName>.AuthSource` パラメータが `admin` に設定されているものとして扱います。
- `*.<SessionName>.Uri` パラメータで URI を指定し(例: `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://127.0`)、ユーザー名とパスワードを Zabbix エージェント 2 のアイテムキーで指定した場合、`Plugins.MongoDB.Sessions.<SessionName>.Us` および `Plugins.MongoDB.Sessions.<SessionName>.Password` が設定されていても、プラグインはアイテムキーのユーザー名とパスワードを使用します。

Attention:
mongodb+srv:// スキームが使用されている場合、URI は DNS TXT レコードからパラメータを取得することがあります。この TXT レコードから取得したパラメータは、URI パラメータ、Zabbix Agent 2 のアイテムキーパラメータ、セッションパラメータを含む他のすべての設定よりも優先されます。
 接続文字列で使用されるホスト名に対しては、TXT レコードは 1 つ □□□□□□□ 存在できます。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.MongoDB.Default.AuthMechanism	MongoDB サーバーへの接続に使用するデフォルトの認証メカニズムを指定します。
Plugins.MongoDB.Default.AuthSource	MongoDB サーバーへの接続を認証するためのデフォルトのデータベースを指定します。
Plugins.MongoDB.Default.Direct	MongoDB レプリカセットノードの自動検出を無効にします。
Plugins.MongoDB.Default.Password	デフォルトのパスワードを指定します。
Plugins.MongoDB.Default.ReplicaSet	MongoDB レプリカセットノードの自動検出に使用するデフォルトのレプリカセット名を指定します。
Plugins.MongoDB.Default.ReadPreference	MongoDB レプリカセットノードへの接続、またはフォールバック時のデフォルトの順序を指定します。
Plugins.MongoDB.Default.TLSConnect	Zabbix エージェント 2 と監視対象データベース間の通信に必要な暗号化レベルを指定します。
Plugins.MongoDB.Default.TLSCAFile	TLS 証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.MongoDB.Default.TLSCertFile	エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.MongoDB.Default.TLSKeyFile	データベースの秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.MongoDB.Default.Uri	デフォルトの URI を指定します。
Plugins.MongoDB.Default.User	デフォルトのユーザー名を指定します。
Plugins.MongoDB.KeepAlive	未使用のプラグイン接続を閉じるまでの最大待機時間を設定します。
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	MongoDB サーバーへの接続に使用する名前付きセッションの認証メカニズムを指定します。
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	MongoDB サーバーへの接続を認証するための名前付きセッションのデータベースを指定します。
Plugins.MongoDB.Sessions.<SessionName>.Direct	名前付きセッションに対する MongoDB レプリカセットノードの自動検出を無効にします。
Plugins.MongoDB.Sessions.<SessionName>.Password	名前付きセッションのパスワードを指定します。
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	MongoDB レプリカセットノードの自動検出に使用する名前付きセッションのレプリカセット名を指定します。
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	MongoDB レプリカセットノードへの接続、またはフォールバック時の名前付きセッションの順序を指定します。
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	名前付きセッションに対して、Zabbix エージェント 2 と監視対象データベース間の通信に必要な暗号化レベルを指定します。
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	名前付きセッションに対して、TLS 証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	名前付きセッションに対して、エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	名前付きセッションに対して、データベース秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.MongoDB.Sessions.<SessionName>.Uri	名前付きセッションに対して、デフォルトの URI を指定します。
Plugins.MongoDB.Sessions.<SessionName>.User	名前付きセッションに対して、デフォルトのユーザー名を指定します。
Plugins.MongoDB.System.Path	プラグイン実行ファイルへのパスを指定します。
Plugins.MongoDB.Timeout	リクエスト実行のタイムアウトを設定します。

注意:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映します。
- 値には環境変数を使用できます。
- Zabbix は、BOM を含まない UTF-8 エンコーディングの設定ファイルのみをサポートします。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.MongoDB.Default.AuthMechanism

MongoDB サーバーへの接続に使用するデフォルトの認証メカニズムを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲: SCRAM-SHA-1, SCRAM-SHA-256, または MONGODB-X509

Plugins.MongoDB.Default.AuthSource

MongoDB サーバーへの接続を認証するためのデフォルトデータベースを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

デフォルト: admin

MONGODB-X509 認証メカニズムが使用されている場合、このパラメーターは自動的に $external に設定されます。これを他の値に変更すると、起動時にエラーが発生します。

Plugins.MongoDB.Default.Direct

MongoDB レプリカセットノードの自動検出を無効にします。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

デフォルト: false
 範囲: true - 指定したサーバーのみに接続し、レプリカセットの他のメンバーは無視します。false - レプリカセット内のすべてのノードを自動的に検出します。

tcp:// (非推奨) スキームを使用する場合、このパラメーターはデフォルトで true に設定されます。

Plugins.MongoDB.Default.Password

デフォルトのパスワードを指定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。

Plugins.MongoDB.Default.ReplicaSet

MongoDB レプリカセットノードの自動検出に使用される、デフォルトのレプリカセット名を指定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。
 このパラメーターを mongodb+srv:// スキームとともに使用すると、複数の URL が指定されている場合と同様に動作します。DNS サーバーに、このパラメーター（または他の任意のパラメーター）を上書きできる TXT レコードが存在する場合があります。
 Plugins.MongoDB.Sessions.<SessionName>.Direct パラメーターが true に設定されている場合は使用できません。エージェントの起動に失敗します。

Plugins.MongoDB.Default.ReadPreference

MongoDB レプリカセットノードへの接続またはフォールバック時のデフォルトの順序を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 Plugins.MongoDB.Sessions.<SessionName>.Direct パラメータが true に設定されている場合は使用できません。エージェントは起動に失敗します。

範囲: primary、primaryPreferred、secondary、secondaryPreferred、nearest、PrimaryMode、PrimaryPreferredMode、SecondaryMode、SecondaryPreferredMode、または NearestMode

primary、primaryPreferred、secondary、secondaryPreferred、nearest の値を推奨します。Mode で終わる値は、従来の互換性エイリアスとしてサポートされています。

Plugins.MongoDB.Default.TLSConnect

Zabbix エージェント 2 と監視対象データベース間の通信に必要な暗号化レベルを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 プラグインバージョン 1.2.1 以降でサポートされています。

範囲: required - TLS 接続を必須にする、verify_ca - 証明書を検証する、verify_full - 証明書と IP アドレスを検証する

Plugins.MongoDB.Default.TLSCAFile

ピア証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。これは、Zabbix エージェント 2 と監視対象データベース間の暗号化通信で使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.MongoDB.Sessions.<SessionName>.TLSConnect が verify_ca または verify_full に設定されている場合

Plugins.MongoDB.Default.TLSCertFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用される、エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile が指定されている場合

Plugins.MongoDB.Default.TLSKeyFile

Zabbix エージェント 2 と監視対象のデータベース間の暗号化通信に使用される、データベースの秘密鍵を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile が指定されている場合

Plugins.MongoDB.Default.Uri

デフォルトの URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 URI 形式に一致している必要があります。
 サポートされるスキーム: mongodb:// および mongodb+srv://; tcp:// スキームは非推奨であり、既存の設定との後方互換性のために残されています。
 ポートは省略できます（default=27017）。
 例: “

Default: tcp://127.0.0.1:27017

Examples:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

デフォルトのユーザー名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MongoDB.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間を秒単位で設定します。

デフォルト: 300
 範囲: 60-900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

MongoDB サーバーへの接続に使用する、名前付きセッションの認証メカニズムを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲: SCRAM-SHA-1、SCRAM-SHA-256、または MONGODB-X509

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

MongoDB サーバーへの接続を認証するための、指定したセッションのデータベースを指定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。

デフォルト: admin

MONGODB-X509 認証メカニズムが使用されている場合、このパラメータは自動的に $external に設定されます。これを他の値に変更すると、起動時にエラーが発生します。

Plugins.MongoDB.Sessions.<SessionName>.Direct

名前付きセッションに対して、MongoDB レプリカセットノードの自動検出を無効にします。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

デフォルト: false
 範囲: true - 指定したサーバーのみに接続し、レプリカセットの他のメンバーは無視します。false - レプリカセット内のすべてのノードを自動的に検出します。

tcp:// (非推奨) スキームが使用されている場合、このパラメーターはデフォルトで true に設定されます。

Plugins.MongoDB.Sessions.<SessionName>.Password

指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

MongoDB レプリカセットノードの自動検出に使用される、名前付きセッションのレプリカセット名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 このパラメーターを mongodb+srv:// スキームと併用すると、複数の URL が指定された場合と同様に動作します。DNS サーバーに TXT レコードがあり、このパラメーター（または他の任意のパラメーター）を上書きできる場合があります。
 Plugins.MongoDB.Sessions.<SessionName>.Direct パラメーターが true に設定されている場合は使用できず、エージェントの起動に失敗します。

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

MongoDB レプリカセットノードまたはフォールバックへの接続に使用する、名前付きセッションの順序を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 Plugins.MongoDB.Sessions.<SessionName>.Direct パラメーターが true に設定されている場合は使用できません。エージェントは起動に失敗します。

範囲: primary、primaryPreferred、secondary、secondaryPreferred、nearest、PrimaryMode、PrimaryPreferredMode、SecondaryMode、SecondaryPreferredMode、または NearestMode

primary、primaryPreferred、secondary、secondaryPreferred、nearest の値を推奨します。Mode で終わる値は、従来の互換性エイリアスとしてサポートされています。

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Zabbix エージェント 2 と監視対象データベース間の通信に必要な暗号化レベルを、指定したセッションに対して設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 プラグインバージョン 1.2.1 以降でサポートされています。

範囲: required - TLS 接続を必須にする、verify_ca - 証明書を検証する、verify_full - 証明書と IP アドレスを検証する

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信で使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパスを指定します。これは、指定したセッションに対して使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.MongoDB.Sessions.<SessionName>.TLSConnect が verify_ca または verify_full に設定されている場合

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する、指定したセッション用のエージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile が指定されている場合

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用される、データベースの秘密鍵を含むファイルの完全なパス名を指定します。これは、指定したセッションに対して使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile が指定されている場合

Plugins.MongoDB.Sessions.<SessionName>.Uri

指定したセッションの URI を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 URI 形式に一致している必要があります。
 サポートされるスキーム: mongodb:// と mongodb+srv://; tcp:// スキームは非推奨で、既存の設定との後方互換性のために残されています。
 ポートは省略できます（default=27017）。
Examples: “

Default: tcp://127.0.0.1:27017

Examples:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MongoDB.System.Path

MongoDB プラグインの実行ファイルへのパスを指定します。

使用例: Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb

Plugins.MongoDB.Timeout

リクエスト実行のタイムアウトを設定します（リクエストが完了するまで待機し、完了しない場合は終了するまでの秒数）。

デフォルト: global timeout
 範囲: 1-30

オプション

Parameter	Description
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テスト用にプラグインを起動します（プラグイン設定は無視されます）。

参照

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

MQTT プラグイン

概要

設定パラメータを使用すると、MQTT プラグインの動作をカスタマイズできます。

これらは agent 2 の設定ファイルではなく、専用の設定ファイル（例: `mqtt.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix agent 2 の設定に含めます。

すべてのパラメータは、パラメータが必須であると明示されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を確認するには、その名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.MQTT.Default.Password	MQTT への接続に使用するデフォルトのパスワードを指定します。
Plugins.MQTT.Default.TLSCAFile	ピア証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.MQTT.Default.TLSCertFile	エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.MQTT.Default.TLSKeyFile	MQTT の秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.MQTT.Default.Topic	MQTT サブスクリプション用のデフォルトトピックを指定します。
Plugins.MQTT.Default.Url	デフォルトの MQTT ブローカー接続文字列を指定します。
Plugins.MQTT.Default.User	MQTT への接続に使用するデフォルトのユーザー名を指定します。
Plugins.MQTT.Sessions.<SessionName>.Password	MQTT への接続に使用する、名前付きセッションのパスワードを指定します。
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	名前付きセッションのピア証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	名前付きセッションのエージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	名前付きセッションの MQTT 秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.MQTT.Sessions.<SessionName>.Topic	MQTT サブスクリプション用の名前付きセッションのトピックを指定します。
Plugins.MQTT.Sessions.<SessionName>.Url	名前付きセッションの MQTT ブローカー接続文字列を指定します。
Plugins.MQTT.Sessions.<SessionName>.User	MQTT への接続に使用する、名前付きセッションのユーザー名を指定します。

注意:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値はenvironment variables をサポートします。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM は含まれていない必要があります。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーター詳細

Plugins.MQTT.Default.Password

MQTT に接続するためのデフォルトパスワードを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.Default.TLSCAFile

Zabbix agent 2 と MQTT ブローカー間の暗号化通信におけるピア証明書検証のため、最上位 CA 証明書を含むファイルの完全なパス名を指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.Default.TLSCertFile

Zabbix agent 2 と MQTT ブローカー間の暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.Default.TLSKeyFile

Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用する MQTT の秘密鍵を含むファイルの完全なパス名を指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.Default.Topic

MQTT サブスクリプションのデフォルトトピックを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
 トピックにはワイルドカード（”+”、”#”）を含めることができます。

例:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

デフォルトの MQTT ブローカー接続文字列を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
 クエリパラメータを含めないでください。
 URL 形式に一致している必要があります。
 サポートされるスキーム: tcp (デフォルト), ws, tls
 スキームは省略できます。
 ポートは省略できます (default=1883)

デフォルト: tcp://localhost:1883

例:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

MQTT に接続するためのデフォルトのユーザー名を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.Sessions.<SessionName>.Password

MQTT に接続するための、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

指定したセッションのピア証明書検証に使用する、最上位 CA 証明書を含むファイルのフルパス名を指定します。これは、Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信で使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

指定したセッションで使用する、エージェント証明書または証明書チェーンを含むファイルのフルパスを指定します。これは、Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用する、指定したセッション用の MQTT 秘密鍵を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MQTT.Sessions.<SessionName>.Topic

MQTT サブスクリプション用の名前付きセッショントピックを指定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。
 トピックにはワイルドカード（”+”、”#”）を含めることができます。

例:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Sessions.<SessionName>.Url

指定されたセッションの MQTT ブローカー接続文字列を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 クエリパラメータを含めないでください。
 URL 形式に一致している必要があります。
 サポートされるスキーム: tcp (デフォルト), ws, tls
 スキームは省略できます。
 ポートは省略できます (デフォルト =1883)

デフォルト: tcp://localhost:1883

例:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

MQTT に接続するための、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

リクエスト実行タイムアウト（リクエストを終了する前に完了を待機する時間）を設定する必要がある場合は、アイテム設定フォームを使用します。

参照

- Zabbix エージェント 2 の一般的な設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

MSSQL プラグイン

概要

設定パラメータを使用すると、MSSQL プラグインの動作をカスタマイズできます。

agent 2 の設定ファイルではなく、専用の設定ファイル（例: `mssql.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix agent 2 の設定に含めます。

MSSQL プラグインはロード可能なプラグインであり、MSSQL plugin repository で利用可能で、詳細も記載されています。

すべてのパラメータは、パラメータが必須であると明記されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、その名前をクリックします）
- パラメータ詳細

Parameter	Description
Plugins.MSSQL.CustomQueriesDir	プラグインが実行できる、ユーザー定義のカスタムクエリを含む `.sql` ファイルのディレクトリへのパスを指定します。
Plugins.MSSQL.CustomQueriesEnabled	`mssql.custom.query` アイテムキーの実行を有効にします。
Plugins.MSSQL.Default.CACertPath	MSSQL サーバーの証明書を発行した認証局（CA）の公開鍵証明書へのデフォルトのファイルパスを指定します。
Plugins.MSSQL.Default.Database	デフォルトのデータベース名を指定します。
Plugins.MSSQL.Default.Encrypt	デフォルトの暗号化レベルを指定します。
Plugins.MSSQL.Default.HostnameInCertificate	デフォルトで MSSQL サーバー証明書のコモンネーム（CN）を指定します。
Plugins.MSSQL.Default.Password	デフォルトのパスワードを指定します。
Plugins.MSSQL.Default.TLSMinVersion	デフォルトの最小 TLS バージョンを指定します。
Plugins.MSSQL.Default.TrustServerCertificate	検証せずにサーバー証明書をデフォルトで信頼します。
Plugins.MSSQL.Default.Uri	デフォルトの URI を指定します。
Plugins.MSSQL.Default.User	デフォルトのユーザー名を指定します。
Plugins.MSSQL.KeepAlive	未使用のプラグイン接続を閉じるまでの最大待機時間（秒）を指定します。
Plugins.MSSQL.Sessions.<Session Name>.CACertPath	指定したセッションで、MSSQL サーバーの証明書を発行した認証局（CA）の公開鍵証明書へのファイルパスを指定します。
Plugins.MSSQL.Sessions.<Session Name>.Database	指定したセッションのデータベース名を指定します。
Plugins.MSSQL.Sessions.<Session Name>.Encrypt	指定したセッションの暗号化レベルを指定します。
Plugins.MSSQL.Sessions.<Session Name>.HostnameInCertificate	指定したセッションで、MSSQL サーバー証明書のコモンネーム（CN）を指定します。
Plugins.MSSQL.Sessions.<Session Name>.Password	指定したセッションのパスワードを指定します。
Plugins.MSSQL.Sessions.<Session Name>.TLSMinVersion	指定したセッションの最小 TLS バージョンを指定します。
Plugins.MSSQL.Sessions.<Session Name>.TrustServerCertificate	検証せずに指定したセッションでサーバー証明書を信頼します。
Plugins.MSSQL.Sessions.<Session Name>.Uri	指定したセッションの URI を指定します。
Plugins.MSSQL.Sessions.<Session Name>.User	指定したセッションのユーザー名を指定します。
Plugins.MSSQL.System.Path	MSSQL プラグインの実行ファイルへのパスを指定します。
Plugins.MSSQL.Timeout	初回接続時およびセッション内の後続操作時にサーバーの応答を待つ時間を、秒単位で設定します。

注意事項:

- デフォルト値は、配布された設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM なしです。
- "\#" で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.MSSQL.CustomQueriesDir

プラグインが実行できるカスタムクエリを含むユーザー定義の `.sql` ファイルが格納されたディレクトリへのパスを指定します。

 プラグインは、起動時に設定されたディレクトリ内の利用可能なすべての.sql ファイルを読み込みます。つまり、カスタムクエ

リファイルへの変更は、プラグインを再起動するまで反映されません。プラグインは Zabbix エージェント 2 とともに起動および停止します。

デフォルト:
/usr/local/share/zabbix/custom-queries/mssql - Unix システムの場合,
:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL - Windows システムの場合 (は ProgramFiles 環境変数のドライブ名)

Plugins.MSSQL.CustomQueriesEnabled

mssql.custom.query アイテムキーの実行を有効にします。
 無効の場合、カスタムクエリディレクトリのパスからクエリは読み込まれません。

デフォルト: false
 範囲: true/false

Plugins.MSSQL.Default.CACertPath

MSSQL サーバーの証明書を発行した認証局（CA）の公開鍵証明書へのデフォルトのファイルパスを指定します。
 証明書は PEM 形式である必要があります。

Plugins.MSSQL.Default.Database

既定のデータベース名を指定します。

Plugins.MSSQL.Default.Encrypt

デフォルトの暗号化レベルを指定します。

範囲:
true - プラグインとサーバー間で送信されるデータは暗号化されます;
false - プラグインとサーバー間で送信されるデータは、ログインパケット以外では暗号化されません;
strict - プラグインとサーバー間で送信されるデータは、TDS8 を使用して E2E 暗号化されます;
disable - プラグインとサーバー間で送信されるデータは暗号化されません。

Plugins.MSSQL.Default.HostNameInCertificate

既定で MSSQL サーバー証明書のコモンネーム (CN) を指定します。

Plugins.MSSQL.Default.Password

保護された MSSQL サーバーのデフォルトパスワードを指定します。

Plugins.MSSQL.Default.TLSMinVersion

デフォルトで最小の TLS バージョンを指定します。

範囲: 1.0, 1.1, 1.2, 1.3

Plugins.MSSQL.Default.TrustServerCertificate

既定では、サーバー証明書を検証せずに信頼します。

範囲: true, false

Plugins.MSSQL.Default.Uri

MSSQL サーバーに接続するためのデフォルト URI を指定します。
 埋め込み資格情報は無視されます。
 サポートされているスキーマは sqlserver:// のみです。
 スキーマは省略できます。

デフォルト: sqlserver://localhost:1433

Plugins.MSSQL.Default.User

保護された MSSQL サーバーのデフォルトユーザー名を指定します。

Plugins.MSSQL.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）を指定します。

デフォルト: 300
 範囲: 60–900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

MSSQL サーバーの証明書を発行した認証局（CA）の公開鍵証明書への、指定したセッションのファイルパスを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 証明書は PEM 形式である必要があります。

Plugins.MSSQL.Sessions.<SessionName>.Database

指定したセッションのデータベース名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

名前付きセッションの暗号化レベルを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲:
true - プラグインとサーバー間のデータ送信は暗号化されます;
false - プラグインとサーバー間のデータ送信は、ログインパケット以外は暗号化されません;
strict - プラグインとサーバー間のデータ送信は、TDS8 を使用して E2E 暗号化されます;
disable - プラグインとサーバー間のデータ送信は暗号化されません。

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

名前付きセッションに対する MSSQL サーバー証明書の共通名 (CN) を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MSSQL.Sessions.<SessionName>.Password

保護された MSSQL サーバーの、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

指定したセッションの最小 TLS バージョンを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

指定したセッションのサーバー証明書を検証せずに信頼します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲: `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

MSSQL サーバーに接続するための、名前付きセッションの URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込み資格情報は無視されます。
 サポートされているスキーマは `sqlserver://` のみです。
 スキーマは省略できます。

デフォルト: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

保護された MSSQL サーバーの、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.MSSQL.System.Path

MSSQL プラグインの実行ファイルへのパスを指定します。
 MSSQL プラグインのグローバル設定です。すべての接続に適用されます。

例: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

最初の接続時およびセッション内の後続の操作で、サーバーからの応答を待機する時間を秒単位で設定します。

デフォルト: `global timeout`
 範囲: 1–30

オプション

Parameter	Description
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テスト用にプラグインを起動します（プラグイン設定は無視されます）。

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

MySQL プラグイン

概要

設定パラメータを使用すると、MySQL プラグインの動作をカスタマイズできます。

これらは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: `mysql.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

明示的に必須と記載されていない限り、すべてのパラメータは任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、その名前をクリックします）
- パラメータ詳細

Parameter	Description
Plugins.Mysql.CallTimeout	要求の完了を待つ最大時間を秒単位で指定します。
Plugins.Mysql.CustomQueriesPath	カスタムクエリを含む .sql ファイルがあるディレクトリの完全なパス名を指定します。
Plugins.Mysql.CustomQueriesEnabled	mysql.custom.query アイテムキーの実行を有効にします。
Plugins.Mysql.Default.Password	デフォルトのパスワードを指定します。
Plugins.Mysql.Default.TLSConnect	Zabbix エージェント 2 と監視対象データベース間の通信に使用するデフォルトの暗号化レベルを指定します。
Plugins.Mysql.Default.TLSCAFile	ピア証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.Mysql.Default.TLSCertFile	エージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.Mysql.Default.TLSKeyFile	データベースの秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.Mysql.Default.Uri	デフォルトの URI を指定します。
Plugins.Mysql.Default.User	デフォルトのユーザー名を指定します。
Plugins.Mysql.KeepAlive	未使用のプラグイン接続を閉じるまでの最大待機時間を秒単位で指定します。
Plugins.Mysql.Sessions.<SessionName>.Password	指定した名前のパスワードを指定します。
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	指定した名前のセッションにおける、Zabbix エージェント 2 と監視対象データベース間の通信の暗号化レベルを指定します。
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	指定した名前のピア証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	指定した名前のエージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	指定した名前のデータベース秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.Mysql.Sessions.<SessionName>.Uri	指定した名前のセッションの URI を指定します。
Plugins.Mysql.Sessions.<SessionName>.User	指定した名前のセッションのユーザー名を指定します。
Plugins.Mysql.Timeout	接続を確立する必要がある場合の待機最大時間を秒単位で設定します。

次の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値には環境変数を使用できます。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Mysql.CallTimeout

リクエストの完了を待機する最大時間を秒単位で指定します。

デフォルト: `global timeout`
 範囲: 1-30

Plugins.Mysql.CustomQueriesPath

カスタムクエリを含む `.sql` ファイルが格納されたディレクトリの完全なパス名を指定します。

デフォルト:
`/usr/local/share/zabbix/custom-queries/mysql` - Unix システムの場合。
`*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` - Windows システムの場合（ここで `*` は ProgramFiles 環境変数のドライブ名です）

Plugins.Mysql.CustomQueriesEnabled

`mysql.custom.query` アイテムキーの実行を有効にします。
 無効の場合、カスタムクエリディレクトリのパスからクエリは読み込まれません。

デフォルト: `false`
 範囲: `true/false`

Plugins.Mysql.Default.Password

保護された MySQL サーバーのデフォルトパスワードを指定します。

Plugins.Mysql.Default.TLSConnect

Zabbix エージェント 2 と監視対象データベース間の通信に対するデフォルトの暗号化レベルを指定します。

範囲:
required - TLS 接続を必須にする;
verify_ca - 証明書を検証する;
verify_full - 証明書と IP アドレスを検証する

Plugins.Mysql.Default.TLSCAFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信で使用される、ピア証明書の検証用に最上位 CA 証明書を含むファイルのフルパスを指定します。

必須: yes。Plugins.Mysql.Default.TLSConnect が verify_ca または verify_full に設定されている場合。

Plugins.Mysql.Default.TLSCertFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用される、エージェント証明書または証明書チェーンを含むファイルのフルパス名を指定します。

必須: yes、Plugins.Mysql.Default.TLSKeyFile が指定されている場合

Plugins.Mysql.Default.TLSKeyFile

Zabbix エージェント 2 と監視対象のデータベース間での暗号化通信に使用される、データベースの秘密鍵を含むファイルの完全なパス名を指定します。

必須: yes、Plugins.Mysql.Default.TLSCertFile が指定されている場合

Plugins.Mysql.Default.Uri

MySQL サーバーに接続するためのデフォルト URI を指定します。
 埋め込み資格情報は無視されます。
 サポートされるスキーム: tcp, unix
 スキーマは省略できます。
 ポートも省略できます（デフォルト =3306）。

例:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Default.User

保護された MySQL サーバーのデフォルトのユーザー名を指定します。

Plugins.Mysql.KeepAlive

未使用のプラグイン接続を切断するまでの最大待機時間を秒単位で指定します。

デフォルト: 300
 範囲: 60-900

Plugins.Mysql.Sessions.<SessionName>.Password

保護された MySQL サーバーに対する、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Zabbix エージェント 2 と監視対象データベース間の通信に対する、名前付きセッションの暗号化レベルを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲:
required - TLS 接続を必須にする;
verify_ca - 証明書を検証する;
verify_full - 証明書と IP アドレスを検証する

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

名前付きセッションのピア証明書検証に使用する、最上位 CA 証明書を含むファイルのフルパスを指定します。これは、Zabbix エージェント 2 と監視対象データベース間の暗号化通信で使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.Mysql.Sessions.<SessionName>.TLSConnect が verify_ca または verify_full に設定されている場合

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

名前付きセッションのエージェント証明書または証明書チェーンを含むファイルの完全なパス名を指定します。これは、Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile が指定されている場合

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

名前付きセッションのデータベース秘密鍵を含むファイルの完全なパス名を指定します。これは、Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用されます。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.Mysql.Sessions.<SessionName>.TLSCertFile が指定されている場合

Plugins.Mysql.Sessions.<SessionName>.Uri

MySQL サーバーに接続するための、名前付きセッションの URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込みの認証情報は無視されます。
 サポートされるスキーム: `tcp`, `unix`
 スキームは省略できます。
 ポートも省略できます（default=3306）。

例:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Sessions.<SessionName>.User

保護された MySQL サーバーに対して、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Mysql.Timeout

接続を確立する必要がある場合に、待機する最大時間を秒単位で設定します。

デフォルト: `global timeout`
 範囲: 1–30

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

NVIDIA GPU プラグイン

概要

設定パラメータを使用すると、NVIDIA GPU プラグインの動作をカスタマイズできます。

これらは agent 2 の設定ファイルではなく、専用の設定ファイル（例: `nvidiagpu.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix agent 2 の設定に含めます。

NVIDIA GPU プラグインはロード可能なプラグインであり、NVIDIA GPU plugin repository にて利用可能で、詳細も記載されています。

すべてのパラメータは、パラメータが必須であると明示されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を確認するには、その名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.NVIDIA.System.Path	NVIDIA GPU プラグインの実行ファイルへのパスを指定します。
Plugins.NVIDIA.Timeout	接続試行およびセッション内の後続操作中に、サーバー応答を待機する最大時間（秒）を指定します。

注意事項:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。
- Zabbix がサポートする設定ファイルの文字エンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.NVIDIA.System.Path

NVIDIA GPU プラグインの実行ファイルへのパスを指定します。

例:

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

接続試行およびセッション内の後続の操作中に、サーバー応答を待機する最大時間（秒）を指定します。
 グローバルなアイテムタイプのタイムアウトまたは個別のアイテムのタイムアウトが、この値より大きい場合はそれが優先されます。

デフォルト: `global timeout`
 範囲: 1–30

参照

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

Oracle プラグイン

概要

設定パラメータを使用すると、Oracle プラグインの動作をカスタマイズできます。

これらのパラメータは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: `oracle.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

すべてのパラメータは、パラメータが必須であると明示されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.Oracle.CallTimeout	要求の完了を待機する最大時間（秒）です。このパラメータは現在効果がなく、代わりにitem timeout 設定が使用されます。
Plugins.Oracle.ConnectTimeout	接続の確立を待機する最大時間（秒）を指定します。
Plugins.Oracle.CustomQueriesPath	カスタムクエリを含む `.sql` ファイルが置かれているディレクトリの完全なパス名を指定します。
Plugins.Oracle.CustomQueriesEnabled	`custom.query` アイテムキーの実行を有効にします。
Plugins.Oracle.Default.Password	デフォルトのパスワードを指定します。
Plugins.Oracle.Default.Service	デフォルトのサービス名を指定します。
Plugins.Oracle.Default.URI	デフォルトの URI を指定します。
Plugins.Oracle.Default.User	デフォルトのユーザー名を指定します。
Plugins.Oracle.KeepAlive	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）を設定します。
Plugins.Oracle.ResolveTNS	Oracle サーバーへの接続に使用する接続文字列（ConnString）の解釈方法を指定します。
Plugins.Oracle.Sessions.<SessionName>.Password	指定したセッションのパスワードを指定します。
Plugins.Oracle.Sessions.<SessionName>.Service	指定したセッションのサービス名を指定します。
Plugins.Oracle.Sessions.<SessionName>.URI	指定したセッションの URI を指定します。
Plugins.Oracle.Sessions.<SessionName>.User	指定したセッションのユーザー名を指定します。

注意事項:

- デフォルト値は、配布される設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値には環境変数 を使用できます。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM なしである必要があります。
- `\#` で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Oracle.CallTimeout

リクエストが完了するまでの最大待機時間（秒）です。
 このパラメータは現在、効果がありません。最大待機時間は代わりにitem timeoutによって決定されます。

デフォルト: `Timeout`[1]
 範囲: 1-30

Plugins.Oracle.ConnectTimeout

接続が確立されるまでの最大待機時間を秒単位で指定します。

デフォルト: `Timeout`[1]
 範囲: 1-30

Plugins.Oracle.CustomQueriesPath

カスタムクエリを含む `.sql` ファイルが格納されたディレクトリの完全なパス名を指定します。

デフォルト:
/usr/local/share/zabbix/custom-queries/oracle - Unix システムの場合、
:\Program Files\Zabbix Agent 2\Custom Queries\Oracle - Windows システムの場合 (は ProgramFiles 環境変数のドライブ名)

例:

```
/etc/zabbix/oracle/sql
```

Plugins.Oracle.CustomQueriesEnabled

`oracle.custom.query` アイテムキーの実行を有効にします。
 無効の場合、カスタムクエリディレクトリのパスからクエリは読み込まれません。

デフォルト: `false`
 範囲: `true/false`

Plugins.Oracle.Default.Password

Oracle に接続するためのデフォルトパスワードを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Oracle.Default.Service

Oracle に接続するためのデフォルトのサービス名を指定します（SID はサポートされていません）。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Oracle.Default.Uri

Oracle に接続するためのデフォルト URI を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
 埋め込み資格情報は無視されます。
 URI 形式に一致する必要があります。
 サポートされるスキーム: `tcp`, `tcps`
 スキーマは省略できます。
 ポートも省略できます（デフォルト =1521）。
 接続文字列として TNS キーまたは TNS 値を指定することもできます。TNS 値は空白を含めずに構成する必要があります。

Examples:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Default.User

Oracle に接続するためのデフォルトのユーザー名を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Oracle.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間を秒単位で設定します。

デフォルト: 300
 範囲: 60–900

Plugins.Oracle.ResolveTNS

Oracle サーバーへの接続に使用する接続文字列（ConnString）の解釈方法を指定します。
 `false` に設定すると、既定のスキームとポートが ConnString に追加されます（存在しない場合）。
 このオプションを `true` に設定すると、既定のスキームとポートは省略されます（ConnString に明示的に指定されている場合を除く）。また、ConnString はそのまま Oracle クライアントに渡されます。Oracle クライアントが `tnsnames.ora` ファイル内で ConnString を見つけた場合は、見つかった接続記述を使用して Oracle サーバーに接続します。

既定: `true`
 範囲: `true/false`

Plugins.Oracle.Sessions.<SessionName>.Password

Oracle に接続するための、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Oracle.Sessions.<SessionName>.Service

Oracle に接続するための、名前付きセッションのサービス名を指定します (SID はサポートされていません)。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Oracle.Sessions.<SessionName>.Uri

Oracle に接続するための、名前付きセッション URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込み資格情報は無視されます。
 URI 形式に一致している必要があります。
 サポートされるスキーム: `tcp`, `tcps`
 スキームは省略できます。
 ポートは省略できます（default=1521）。
 接続文字列として TNS キーまたは TNS 値を指定することもできます。TNS 値は空白を含めずに記述する必要があります。

Examples:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Oracle に接続するための、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

脚注

[1] タイムアウトは、Zabbix エージェント 2 (UNIX) または Zabbix エージェント 2 (Windows) の設定ファイル (`zabbix_agent2.conf`) にある `Timeout` オプションで定義されます。

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

PostgreSQL プラグイン

概要

設定パラメータを使用すると、PostgreSQL プラグインの動作をカスタマイズできます。

agent 2 の設定ファイルではなく、専用の設定ファイル（例: `postgresql.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix agent 2 の設定に含めます。

PostgreSQL プラグインはロード可能なプラグインであり、PostgreSQL plugin repository にて利用可能で、詳細が完全に記載されています。

すべてのパラメータは、パラメータが必須であると明示されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、その名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.PostgreSQL.CallTimeout	要求の完了を待機する最大時間を秒単位で設定します。
Plugins.PostgreSQL.CustomQueriesPath	カスタムクエリを含む `.sql` ファイルがあるディレクトリの完全なパス名を指定します。
Plugins.PostgreSQL.CustomQueriesEnabled	`Queries` と `custom.query` アイテムキーの実行を有効にします。
Plugins.PostgreSQL.Default.KeepAlive	PostgreSQL 接続のデフォルトのキャッシュモードを指定します。
Plugins.PostgreSQL.Default.Database	デフォルトのデータベース名を指定します。
Plugins.PostgreSQL.Default.Password	デフォルトのパスワードを指定します。
Plugins.PostgreSQL.Default.TLSCAFile	ピア証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.PostgreSQL.Default.TLSCertFile	PostgreSQL 証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.PostgreSQL.Default.TLSConnect	Zabbix agent 2 と監視対象データベース間の通信に使用するデフォルトの暗号化レベルを指定します。
Plugins.PostgreSQL.Default.TLSKeyFile	PostgreSQL 秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.PostgreSQL.Default.Uri	デフォルトの URI を指定します。
Plugins.PostgreSQL.Default.User	デフォルトのユーザー名を指定します。
Plugins.PostgreSQL.KeepAlive	未使用のプラグイン接続が閉じられるまでの最大待機時間を秒単位で設定します。
Plugins.PostgreSQL.Sessions.<SessionName>.KeepAlive	PostgreSQL 接続の名前付きセッションのキャッシュモードを指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.Database	名前付きセッションのデータベース名を指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.Password	名前付きセッションのパスワードを指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	ピア証明書検証用の最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	PostgreSQL 証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Zabbix agent 2 と監視対象データベース間の通信に使用する、名前付きセッションの暗号化レベルを指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	PostgreSQL 秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.Uri	名前付きセッションの URI を指定します。
Plugins.PostgreSQL.Sessions.<SessionName>.User	名前付きセッションのユーザー名を指定します。

|Plugins.PostgreSQL.System.Path|PostgreSQL プラグインの実行ファイルへのパスを指定します。| |Plugins.PostgreSQL.Timeout| 要求の実行タイムアウトを設定します。|

注意事項:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。

- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- "\#" で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.PostgreSQL.CallTimeout

リクエストが完了するまでの最大待機時間を秒単位で設定します。

デフォルト: `global timeout`
 範囲: 1-30

Plugins.PostgreSQL.CustomQueriesPath

カスタムクエリを含む `.sql` ファイルが格納されたディレクトリの完全なパス名を指定します。

デフォルト:
`/usr/local/share/zabbix/custom-queries/postgreSQL` - Unix システムの場合、
`*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` - Windows システムの場合（`*` は `ProgramFiles` 環境変数のドライブ名）

例:

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

`postgresql.custom.query` アイテムキーの実行を有効にします。
 無効にすると、カスタムクエリディレクトリのパスからクエリは読み込まれません。

デフォルト: `false`
 範囲: `true/false`

Plugins.PostgreSQL.Default.CacheMode

PostgreSQL 接続のデフォルトのキャッシュモードを指定します。

デフォルト: `prepare`
 範囲:
`prepare` - PostgreSQL サーバー上にプリペアドステートメントを作成します;
`describe` - サーバー上にステートメントを作成せずに、匿名のプリペアドステートメントを使用してステートメントを記述します。

`describe` は、PgBouncer のような接続プーラーを使用している場合など、環境がプリペアドステートメントを許可しないときに主に有用です。

Plugins.PostgreSQL.Default.Database

PostgreSQL に接続するためのデフォルトデータベースを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.PostgreSQL.Default.Password

PostgreSQL に接続するためのデフォルトパスワードを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.PostgreSQL.Default.TLSCAFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信におけるピア証明書検証のため、最上位 CA 証明書を含むファイルのフルパスを指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

必須: `yes`、`Plugins.PostgreSQL.Default.TLSConnect` が `verify_ca` または `verify_full` に設定されている場合

Plugins.PostgreSQL.Default.TLSCertFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する、PostgreSQL の証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

必須: `yes`、`Plugins.PostgreSQL.Default.TLSKeyFile` が指定されている場合

Plugins.PostgreSQL.Default.TLSConnect

Zabbix エージェント 2 と監視対象データベース間の通信に使用するデフォルトの暗号化レベルを指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

範囲:
`required` - TLS を転送モードとして使用し、ID チェックを行わずに接続します。
`verify_ca` - TLS を使用して接続し、証明書を検証します。
`verify_full` - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID (CN) がその証明書と一致することを検証します。

暗号化タイプが未定義の場合は、暗号化されていない接続を意味します。

Plugins.PostgreSQL.Default.TLSKeyFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する PostgreSQL の秘密鍵を含むファイルの完全なパス名を指定します。
 名前付きセッションで値が指定されていない場合に使用されます。

必須: yes、Plugins.PostgreSQL.Default.TLSCertFile が指定されている場合

Plugins.PostgreSQL.Default.Uri

PostgreSQL に接続するためのデフォルト URI を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
 埋め込み認証情報は無視されます。
 URI 形式に一致している必要があります。
 サポートされるスキーム: tcp, unix

例:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Default.User

PostgreSQL に接続するためのデフォルトのユーザー名を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.PostgreSQL.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）を設定します。

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

PostgreSQL 接続の名前付きセッションキャッシュモードを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

デフォルト: prepare
 範囲:
prepare - PostgreSQL サーバー上でプリペアドステートメントを作成します;
describe - サーバー上にステートメントを作成せずに、匿名のプリペアドステートメントを使用してステートメントを記述します。

describe は、PgBouncer のような接続プーラーを使用している場合など、環境がプリペアドステートメントを許可しないときに主に有用であることに注意してください。

Plugins.PostgreSQL.Sessions.<SessionName>.Database

PostgreSQL に接続するための、名前付きセッションのデータベースを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.PostgreSQL.Sessions.<SessionName>.Password

PostgreSQL に接続するための、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッションの名前を定義します。
 パスワードの形式と一致している必要があります。

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信におけるピア証明書検証のため、最上位 CA 証明書を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect が verify_ca または verify_full に設定されている場合

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する、PostgreSQL の証明書または証明書チェーンを含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile が指定されている場合

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Zabbix エージェント 2 と監視対象のデータベース間の通信に対して、指定したセッションレベルの暗号化方式を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

範囲:
 required - TLS を転送モードとして使用して接続しますが、ID の確認は行いません;
 verify_ca - TLS を使用して接続し、証明書を検証します;
 verify_full - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベースの識別情報 (CN) がその証明書と一致することを検証します。

暗号化タイプが未定義の場合は、暗号化されていない接続を意味します。

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する PostgreSQL の秘密鍵を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: yes、Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile が指定されている場合。

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

PostgreSQL に接続するための、名前付きセッション URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込み資格情報は無視されます。
 URI 形式に一致している必要があります。
 サポートされるスキーム: tcp, unix

例:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Sessions.<SessionName>.User

PostgreSQL に接続するための、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.PostgreSQL.System.Path

PostgreSQL プラグインの実行ファイルへのパスを指定します。

必須: yes

例:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

リクエストの実行タイムアウトを設定します（リクエストが完了するまで待機してから終了するまでの秒数）。

デフォルト: global timeout
 範囲: 1-30

オプション

Parameter	Description
-V --version	プラグインのバージョンとライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（省略形）。
-t, --test <item key>	テスト用にプラグインを起動します（プラグイン設定は無視されます）。

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

Redis プラグイン

概要

設定パラメータを使用すると、Redis プラグインの動作をカスタマイズできます。

これらのパラメータは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: redis.conf）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

すべてのパラメータは、パラメータが必須であると明示されていない限り、任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、名前をクリックしてください）
- パラメータ詳細

Parameter	Description
Plugins.Redis.Default.Password	Redis への接続に使用するデフォルトパスワードを指定します。
Plugins.Redis.Default.TLSConnect	Zabbix エージェント 2 の通信に必要な暗号化レベルを指定します。
Plugins.Redis.Default.TLSCAFile	最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.Redis.Default.TLSCertFile	エージェント証明書を含むファイルの完全なパス名を指定します。

Parameter	Description
Plugins.Redis.Default.TLSKeyFile	エージェント秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.Redis.Default.Uri	Redis への接続に使用するデフォルト URI を指定します。
Plugins.Redis.Default.User	Redis への接続に使用するデフォルトユーザー名を指定します。
Plugins.Redis.KeepAlive	未使用のプラグイン接続を切断するまでの最大待機時間（秒）を設定します。
Plugins.Redis.Sessions.<SessionName>.Password	Redis への接続に使用する、名前付きセッションのパスワードを指定します。
Plugins.Redis.Sessions.<SessionName>.TLSConnect	名前付きセッションで Zabbix エージェント 2 通信に必要な暗号化レベルを指定します。
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	名前付きセッションの最上位 CA 証明書を含むファイルの完全なパス名を指定します。
Plugins.Redis.Sessions.<SessionName>.TLSCertFile	名前付きセッションのエージェント証明書を含むファイルの完全なパス名を指定します。
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile	名前付きセッションのエージェント秘密鍵を含むファイルの完全なパス名を指定します。
Plugins.Redis.Sessions.<SessionName>.Uri	Redis への接続に使用する、名前付きセッションの URI を指定します。
Plugins.Redis.Sessions.<SessionName>.User	Redis への接続に使用する、名前付きセッションのユーザー名を指定します。
Plugins.Redis.Timeout	リクエスト実行のタイムアウトを設定します。

注意事項:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。
- Zabbix がサポートする設定ファイルのエンコーディングは UTF-8 のみで、BOM は含められません。
- \# で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Redis.Default.Password

Redis に接続するためのデフォルトパスワードを指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Redis.Default.TLSConnect

Zabbix エージェント 2 の通信に必要な暗号化レベルを指定します。
 値が空、または未設定の場合は、TLS なしとして扱われます。

範囲: `required` - 暗号化、検証なし（テスト用のみ）、`verify_ca` - 暗号化、サーバー証明書、`verify_full` - 暗号化、サーバー証明書および SAN で検証されたサーバー名

Plugins.Redis.Default.TLSCAFile

トップレベルの CA 証明書を含むファイルのフルパスを指定します。

Plugins.Redis.Default.TLSCertFile

エージェント証明書を含むファイルの完全なパス名を指定します。

Plugins.Redis.Default.TLSKeyFile

エージェントの秘密鍵を含むファイルの完全なパス名を指定します。

Plugins.Redis.Default.Uri

Redis に接続するためのデフォルト URI を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
 埋め込み認証情報は無視されます。URI 形式に一致している必要があります。
 サポートされるスキーム: `tcp`, `unix`
 スキームは省略できます。
 ポートも省略できます（デフォルト =6379）。

デフォルト: `tcp://localhost:6379`

例:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Redis に接続するためのデフォルトのユーザー名を指定します。
 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

デフォルト: `default`

Plugins.Redis.KeepAlive

未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）を設定します。

デフォルト: 300
 範囲: 60–900

Plugins.Redis.Sessions.<SessionName>.Password

Redis に接続するための、指定したセッションのパスワードを設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Redis.Sessions.<SessionName>.TLSConnect

指定したセッションに対する Zabbix エージェント 2 の通信で必要な暗号化レベルを指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 空または未設定の値は、TLS なしとして扱われます。

範囲: `required` - 暗号化あり、検証なし（テスト用のみ）、`verify_ca` - 暗号化あり、サーバー証明書あり、`verify_full` - 暗号化あり、サーバー証明書と SAN によるサーバー名検証あり

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

指定されたセッションのトップレベル CA 証明書を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

必須: `yes`、`Plugins.Redis.Sessions.<SessionName>.TLSConnect` が `verify_ca` または `verify_full` に設定されている場合

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

指定したセッションのエージェント証明書を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

指定されたセッション用のエージェント秘密鍵を含むファイルの完全なパス名を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Redis.Sessions.<SessionName>.Uri

Redis に接続するための、名前付きセッション URI を指定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。
 埋め込み認証情報は無視されます。URI 形式に一致している必要があります。
 サポートされるスキーム: `tcp`, `unix`
 スキームは省略できます。
 ポートは省略できます（default=6379）。

デフォルト: `tcp://localhost:6379`

例:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Redis に接続するための、指定したセッションのユーザー名を設定します。
 <SessionName> - アイテムキーで使用するセッション名を定義します。

Plugins.Redis.Timeout

リクエストの実行タイムアウトを設定します（リクエストが完了するまで待機し、その後に終了するまでの秒数）。

デフォルト: `global timeout`
 範囲: 1–30

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

SMART プラグイン

概要

設定パラメータを使用すると、SMART プラグインの動作をカスタマイズできます。

これらは、エージェント 2 の設定ファイルではなく、専用の設定ファイル（例: `smart.conf`）に指定することを推奨します。その後、Include ディレクティブを使用して、このファイルを Zabbix エージェント 2 の設定に含めます。

特に明記されていない限り、すべてのパラメータは任意です。

このページの構成は次のとおりです。

- パラメータ概要（パラメータの詳細を表示するには、その名前をクリックします）

- パラメータ詳細

Parameter	Description
Plugins.Smart.Path	smartctl 実行ファイルへのパスを指定します。
Plugins.Smart.Timeout	リクエストの実行タイムアウトを設定します（リクエストが完了するまで待機してから停止するまでの時間を秒単位で指定します）。

次の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映します。
- 値では環境変数を使用できます。
- `smartctl` 実行ファイルへのパスは、システムの `PATH` 環境変数にそのディレクトリを追加するか、`Plugins.Smart.Path` を設定することで指定する必要があります。これは Linux と Windows の両方に適用されます。
- Zabbix がサポートする設定ファイルの文字エンコーディングは UTF-8 のみで、BOM は含められません。
- `"\#"` で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

Plugins.Smart.Path

smartctl 実行ファイルへのパスを指定します。

デフォルト: `smartctl`

Plugins.Smart.Timeout

リクエスト実行のタイムアウトを設定します（リクエストが完了するまで待機し、その後停止するまでの秒数）。

デフォルト: `global timeout<br>` 範囲: 1–30

関連項目

- 一般的な Zabbix エージェント 2 の設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- プラグイン の設定手順

Zabbix agent 2 プラグインプロトコル

Zabbix agent 2 のプロトコルは、コード、サイズ、およびデータモデルに基づいています。

コード

Type	Size	Comments
Byte	4	ペイロードの種類。現在は JSON のみがサポートされています。

サイズ

Type	Size	Comments
Byte	4	現在のペイロードのバイト数。

ペイロードデータ

Type	Size	Comments
Byte	*Size* フィールドで定義される	JSON 形式のデータ。

ペイロードデータ定義

共通データ

これらのパラメータは、すべてのリクエスト/レスポンスに含まれます:

Name	Type	Comments
id	uint32	リクエストの場合 - リクエストとレスポンスを関連付けるために使用される増分 ID。リクエスト方向ごとに一意です（つまり、エージェントからプラグイン、またはプラグインからエージェント）。 レスポンスの場合 - 対応するリクエストの ID。
type	uint32	リクエストの種類。

ログリクエスト

プラグインがエージェントのログファイルにログメッセージを書き込むために送信するリクエスト。

direction	plugin → エージェント
response	no

ログリクエスト固有のパラメータ:

Name	Type	Comments
severity	uint32	メッセージの重大度（ログレベル）。
message	string	ログに記録するメッセージ。

例:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

登録リクエスト

エージェントの起動フェーズ中にエージェントから送信される、提供されるメトリクスを取得してプラグインを登録するためのリクエストです。

direction	エージェント → プラグイン
response	yes

登録リクエストに固有のパラメータ:

Name	Type	Comments
version	string	プロトコルバージョン <major>.<minor>

例:

```
{"id":1,"type":2,"version":"1.0"}
```

登録レスポンス

登録リクエストに対するプラグインの応答。

direction	plugin → エージェント
response	n/a

登録レスポンスに固有のパラメータ:

Name	Type	Comments
name	string	プラグイン名。
metrics	array of strings (optional)	プラグイン内で使用される、説明付きのメトリクス。RegisterMetrics() を返します。エラーが返された場合は存在しません。

Name	Type	Comments
interfaces	uint32 (optional)	プラグインがサポートするインターフェースのビットマスク。エラーが返された場合は存在しません。
error	string (optional)	プラグインを起動できない場合に返されるエラーメッセージ。メトリクスが返される場合は存在しません。

例:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

または

```
{"id":2,"type":3,"error":"error message"}
```

開始リクエスト

Runner インターフェースの Start 関数を実行するためのリクエストです。

direction	エージェント → plugin
response	no

このリクエストには特定のパラメータはなく、共通データパラメータのみを含みます。

例:

```
{"id":3,"type":4}
```

リクエストの終了

エージェントからプラグインをシャットダウンするために送信されるリクエストです。

direction	エージェント → プラグイン
response	no

このリクエストには特定のパラメーターはなく、共通データパラメーターのみが含まれます。

例:

```
{"id":3,"type":5}
```

エクスポートリクエスト

Exporter インターフェースの Export 関数を実行するためのリクエストです。

direction	エージェント → プラグイン
response	no

エクスポートリクエストに固有のパラメーター:

Name	Type	Comments
key	string	プラグインキー。
parameters	array of strings (optional)	Export 関数のパラメーター。

例:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

エクスポート応答

Exporter インターフェースの Export 関数からの応答。

direction	plugin → エージェント

response	n/a

エクスポート応答に固有のパラメータ:

Name	Type	Comments
value	string (optional)	Export 関数からの応答値。エラーが返された場合は存在しません。
error	string (optional)	Export 関数が正常に実行されなかった場合のエラーメッセージ。値が返された場合は存在しません。

例:

```
{"id":5,"type":7,"value":"response"}
```

または

```
{"id":5,"type":7,"error":"error message"}
```

Configure リクエスト

Configurator インターフェースの *Configure* 関数を実行するためのリクエストです。

direction	エージェント → プラグイン
response	n/a

Configure リクエストに固有のパラメータ:

Name	Type	Comments
global_options	JSON object	グローバルなエージェント設定オプションを含む JSON オブジェクト。
private_options	JSON object (optional)	指定されている場合、プラグイン固有の設定オプションを含む JSON オブジェクト。

例:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

リクエストの検証

Configurator インターフェースの *Validate* 関数を実行するためのリクエストです。

direction	エージェント → plugin
response	yes

Validate リクエストに固有のパラメーター:

Name	Type	Comments
private_options	JSON object (optional)	private plugin configuration options を含む JSON object。指定されている場合に使用されます。

例:

```
{"id":7,"type":9,"private_options":{...}}
```

応答の検証

Configurator インターフェースの *Validate* 関数からの応答。

direction	plugin → エージェント
response	n/a

Validate 応答に固有のパラメーター:

Name	Type	Comments
error	string (optional)	Validate 関数が正常に実行されなかった場合に返されるエラーメッセージ。正常に実行された場合は存在しません。

例:

```
{"id":8,"type":10}
```

または

```
{"id":8,"type":10,"error":"error message"}
```

Java ゲートウェイ

概要　Zabbix Java ゲートウェイは、ソースコードまたはパッケージからインストールできます。

JMX アプリケーションの監視は、「Zabbix Java ゲートウェイ」と呼ばれる Zabbix デーモンによってネイティブにサポートされています。Zabbix Java ゲートウェイは、Java で記述されたデーモンです。ホスト上の特定の JMX カウンターの値を取得するために、Zabbix サーバーは Zabbix Java ゲートウェイに問い合わせます。Zabbix Java ゲートウェイは、JMX 管理 APIを使用して、対象のアプリケーションにリモートで問い合わせます。アプリケーションに追加のソフトウェアをインストールする必要はありません。コマンドラインで`-Dcom.sun.management.jmxremote` オプションを指定して起動するだけです。

Java ゲートウェイは、Zabbix サーバーまたはプロキシからの受信接続を受け付け、「パッシブプロキシ」としてのみ使用できます。Zabbix プロキシとは異なり、Zabbix プロキシから使用することもできます（Zabbix プロキシをチェーン接続することはできません）。各 Java ゲートウェイへのアクセスは、Zabbix サーバーまたは Zabbix プロキシの設定ファイルで直接設定します。そのため、1 台の Zabbix サーバーまたは Zabbix プロキシにつき、設定できる Java ゲートウェイは 1 つだけです。ホストに *JMX* エージェントタイプのアイテムと、その他のタイプのアイテムがある場合、取得のために Java ゲートウェイへ渡されるのは *JMX* エージェントアイテムのみです。

アイテムを Java ゲートウェイ経由で更新する必要がある場合、Zabbix サーバーまたはプロキシは Java ゲートウェイに接続して値を要求します。Java ゲートウェイは JMX から値を取得し、サーバーまたはプロキシに返します。そのため、Java ゲートウェイは値をキャッシュしません。

Zabbix サーバーまたはプロキシには、Java ゲートウェイに接続する専用のプロセスがあります。このプロセス数は `StartJavaPollers` オプションで制御します。Java ゲートウェイは内部で複数のスレッドを起動します。スレッド数は `START_POLLERS`オプションで制御します。サーバー側では、接続に「Timeout」秒を超える時間がかかると接続が終了しますが、Java ゲートウェイは JMX カウンターからの値の取得処理を続けている場合があります。これを解決するため、Java ゲートウェイには `TIMEOUT` オプションが用意されており、JMX ネットワーク操作のタイムアウトを設定できます。

関連項目:

- 設定パラメータ: Zabbix Java ゲートウェイ
- Debian/Ubuntu パッケージからのセットアップ
- RHEL パッケージからのセットアップ
- ソースからのセットアップ

Zabbix サーバーまたはプロキシは、可能な限り複数の要求を単一の JMX ターゲットにまとめて処理しようとします（アイテムの間隔の影響を受けます）。これらは、より高いパフォーマンスのために単一の接続で Java ゲートウェイに送信されます。

`StartJavaPollers` は `START_POLLERS` 以下に設定することを推奨します。そうしないと、Java ゲートウェイで受信要求を処理するためのスレッドが利用できない状況が発生する可能性があります。
そのような場合、Java ゲートウェイは `ThreadPoolExecutor.CallerRunsPolicy` を使用します。つまり、メインスレッドが受信要求を処理し、一時的に新しい要求を受け付けなくなります。

Zabbix Java ゲートウェイで Wildfly ベースの Java アプリケーションを監視する場合は、Wildfly download page で入手できる最新の `jboss-client.jar` をインストールしてください。

設定パラメーター**: Zabbix Java** ゲートウェイ

Zabbix Java ゲートウェイ の起動に `startup.sh` および `shutdown.sh` スクリプトを使用する場合は、必要な設定パラメーターを `settings.sh` ファイルで指定できます。startup および shutdown スクリプトは settings ファイルを読み込み、シェル変数（1 列目に記載）を Java プロパティー（2 列目に記載）へ変換します。

Zabbix Java ゲートウェイを java を直接実行して手動で起動する場合は、対応する Java プロパティーをコマンドラインで指定します。

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	待ち受ける IP アドレス。
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	待ち受けるポート。
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	PID ファイル名。省略した場合、Zabbix Java Gateway はコンソールアプリケーションとして起動します。
PROPERTIES_FILE	zabbix.propertiesFile	no			プロパティーファイル名。コマンドラインから見えないようにキーと値の形式で追加のプロパティーを設定したり、既存のプロパティーを上書きしたりするために使用できます。 例: "javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	起動するワーカースレッド数。
TIMEOUT	zabbix.timeout	no	1-30	3	ネットワーク操作の待機時間（秒）。

Warning:
ポート 10052 は IANA 登録済み ではありません。

Debian/Ubuntu パッケージからのセットアップ

概要

パッケージからインストールした場合、以下の情報は Zabbix Java ゲートウェイのセットアップに役立ちます。

Java ゲートウェイの設定と起動

Java ゲートウェイの設定は、次のファイルで調整できます。

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

詳細については、Zabbix Java ゲートウェイの設定パラメータを参照してください。

Zabbix Java ゲートウェイを起動するには、次を実行します。

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

起動時に Zabbix Java ゲートウェイを自動的に起動するには、次を実行します。

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

Java ゲートウェイを使用するためのサーバーの設定

Java ゲートウェイが起動して稼働している状態になったら、Zabbix サーバーに Zabbix Java ゲートウェイの場所を指定する必要があります。
これは、サーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定して行います。
JMX アプリケーションが稼働しているホストが Zabbix プロキシで監視されている場合は、代わりにプロキシ設定ファイルで接続パラメーターを指定します。

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

デフォルトでは、サーバーは JMX 監視に関連するプロセスを起動しません。
ただし、これを使用したい場合は、Java ポーラーの事前フォーク済みインスタンス数を指定する必要があります。
これは、通常のポーラーやトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動することを忘れないでください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです。

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

ログ出力を増やしたい場合は、次のファイルを編集します。

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

そして、level="info" の info を debug、またはさらに詳細なトラブルシューティング用に trace に変更します。

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

</configuration>
```

JMX 監視

詳細はJMX monitoring ページを参照してください。

RHEL パッケージからのセットアップ

概要

パッケージからインストールした場合、以下の情報は Zabbix Java ゲートウェイのセットアップに役立ちます。

Java ゲートウェイの設定と起動

Zabbix Java ゲートウェイの設定パラメータは、次のファイルで調整できます。

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

詳細については、Zabbix Java ゲートウェイの設定パラメータを参照してください。

Zabbix Java ゲートウェイを起動するには、次を実行します。

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

起動時に Zabbix Java ゲートウェイを自動的に開始するには、次のようにします。

RHEL 7 以降:

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

RHEL 7 より前:

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

Java ゲートウェイを使用するためのサーバーの設定

Java ゲートウェイが起動して稼働している状態になったら、Zabbix サーバーに Zabbix Java ゲートウェイの場所を指定する必要があります。
これは、サーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定して行います。
JMX アプリケーションが稼働しているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメーターを指定します。

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

デフォルトでは、サーバーは JMX 監視に関連するプロセスを起動しません。
ただし、これを使用したい場合は、Java poller の事前フォーク済みインスタンス数を指定する必要があります。
これは、通常の poller や trapper を指定する場合と同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動することを忘れないでください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです:

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

ログ出力を増やしたい場合は、次のファイルを編集します:

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

そして、level="info" の info を debug、またはさらに trace（詳細なトラブルシューティング用）に変更します:

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

</configuration>
```

JMX 監視

詳細については、JMX monitoringページを参照してください。

ソースからのセットアップ

概要

ソースからインストールした場合、以下の情報は Zabbix Java ゲートウェイ の設定に役立ちます。

ファイルの概要

ソースから Java ゲートウェイを取得した場合は、$PREFIX/sbin/zabbix_java の下にシェルスクリプト、JAR ファイル、設定ファイルが一式作成されます。これらのファイルの役割を以下に示します。

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

Java ゲートウェイ本体の JAR ファイルです。

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Java ゲートウェイの依存ライブラリであるLogback、SLF4J、およびAndroid JSONライブラリです。

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

Logback の設定ファイルです。

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

Java ゲートウェイを起動および停止するための便利なスクリプトです。

```
settings.sh
```

上記の起動および停止スクリプトから読み込まれる設定ファイルです。

Java ゲートウェイの設定と起動

デフォルトでは、Java ゲートウェイはポート 10052 で待ち受けます。Java ゲートウェイを別のポートで起動する予定がある場合は、settings.sh スクリプトで指定できます。
これとその他のオプションの指定方法については、Java ゲートウェイ設定ファイルの説明を参照してください。

> **Warning:**
> ポート 10052 は IANA に登録されていません。

設定に問題がなければ、startup.sh スクリプトを実行して Java ゲートウェイを起動できます。

```
./startup.sh
```

同様に、Java ゲートウェイが不要になったら、shutdown.sh スクリプトを実行して停止します。

```
./shutdown.sh
```

なお、サーバーやプロキシとは異なり、Java ゲートウェイは軽量で、データベースを必要としません。

Java ゲートウェイを使用するためのサーバーの設定

Java ゲートウェイが起動して動作している状態になったら、Zabbix サーバーに Zabbix Java ゲートウェイの場所を指定する必要があります。
これは、サーバー設定ファイルで JavaGateway および JavaGatewayPort パラメータを指定することで行います。
JMX アプリケーションが動作しているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメータを指定します。

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

デフォルトでは、サーバーは JMX 監視に関連するプロセスを起動しません。
ただし、これを使用したい場合は、Java ポーラーの事前フォーク済みインスタンス数を指定する必要があります。
これは、通常のポーラーやトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動することを忘れないでください。

Java ゲートウェイのデバッグ

Java ゲートウェイに問題がある場合や、Web インターフェースで表示されるアイテムに関するエラーメッセージが十分に分かりやすくない場合は、Java ゲートウェイのログファイルを確認するとよいでしょう。

デフォルトでは、Java ゲートウェイはログレベル”info” で活動内容を /tmp/zabbix_java.log ファイルに記録します。
この情報だけでは不十分な場合があり、ログレベル”debug” の情報が必要になることがあります。
ログレベルを上げるには、lib/logback.xml ファイルを変更し、<root> タグの level 属性を debug に変更します。

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

Zabbix サーバーや Zabbix プロキシとは異なり、logback.xml ファイルを変更した後に Zabbix Java ゲートウェイを再起動する必要はありません。logback.xml の変更は自動的に反映されます。
デバッグが終わったら、ログレベルを”info” に戻せます。

別のファイルや、データベースのようなまったく別の媒体にログを出力したい場合は、必要に応じて logback.xml ファイルを調整してください。
詳細は Logback Manual を参照してください。

デバッグ目的では、Java ゲートウェイをデーモンではなくコンソールアプリケーションとして起動すると便利な場合があります。
その場合は、settings.sh 内の PID_FILE 変数をコメントアウトします。
PID_FILE が省略されると、startup.sh スクリプトは Java ゲートウェイをコンソールアプリケーションとして起動し、Logback は lib/logback-console.xml ファイルを使用します。この設定ではコンソールへのログ出力だけでなく、ログレベル”debug” も有効になります。

最後に、Java ゲートウェイはログ記録に SLF4J を使用しているため、適切な JAR ファイルを lib ディレクトリに配置することで、Logback を任意のフレームワークに置き換えることができます。
詳細は SLF4J Manual を参照してください。

JMX 監視

詳細はJMX monitoring のページを参照してください。

送信者

概要　Zabbix sender は、処理のためにパフォーマンスデータを Zabbix サーバーまたはプロキシに送信するために使用できるコマンドラインユーティリティです。

このユーティリティは通常、可用性データとパフォーマンスデータを定期的に送信する、長時間実行されるユーザースクリプトで使用されます。

結果を Zabbix サーバーまたはプロキシに直接送信するには、*Zabbix trapper*アイテムタイプを設定する必要があります。

関連項目:

- Zabbix sender プロトコル
- Windows 用 Zabbix sender 動的リンクライブラリ

- Zabbix sender として動作する組み込み機能を備えたZabbix 用 Python ライブラリ

Zabbix sender の実行　Zabbix UNIX sender を実行する例:

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

ここで:

- z - Zabbix サーバーのホスト（IP アドレスも使用可能）;
- s - 監視対象ホストのテクニカル名（Zabbix の Web インターフェースに登録されている名前）;
- k - アイテムキー;
- o - 送信する値。

Attention:
空白を含むオプションは、二重引用符で囲む必要があります。

Zabbix sender は、入力ファイルから複数の値を送信するために使用できます。詳細については、Zabbix sender の man ページを参照してください。

設定ファイルを指定すると、Zabbix sender はエージェントのServerActive設定パラメータで定義されているすべてのアドレスを使用してデータを送信します。1 つのアドレスへの送信に失敗した場合、sender は別のアドレスへの送信を試みます。バッチデータの送信が 1 つのアドレスで失敗した場合、それ以降のバッチはそのアドレスには送信されません。

Zabbix sender は、ファイルの先頭にバイトオーダーマーク（BOM）がない UTF-8 エンコーディングの文字列を受け入れます（Unix 系システムと Windows の両方）。

Windows 上の Zabbix sender も、同様に実行できます。

```
zabbix_sender.exe [options]
```

zabbix_sender のリアルタイム送信シナリオでは、短い間隔で渡された複数の値を収集し、1 つの接続でサーバーに送信します。直前の値との差が 0.2 秒以内の値は同じスタックに入れることができますが、最大ポーリング時間は 1 秒です。

Note:
指定した設定ファイルに無効なパラメータエントリ（parameter=value 表記に従っていないエントリ）が存在する場合、Zabbix sender は終了します。

Zabbix sender を低レベルディスカバリで実行する　低レベルディスカバリ用の JSON 形式の値を送信するために Zabbix sender を実行する例:

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

これを機能させるには、低レベルディスカバリルールで *Zabbix trapper* のアイテムタイプを使用する必要があります（この例では、trapper.discovery.item キーを使用しています）。

Zabbix sender プロトコル

Zabbix サーバーと Zabbix プロキシは、Zabbix sender からデータを受信するために JSON ベースの通信プロトコルを使用します。データは、trapper アイテムまたはトラップ機能を有効にしたHTTP エージェントアイテムで受信できます。

リクエストメッセージとレスポンスメッセージは、ヘッダーとデータ長で始まる必要があります。

Zabbix sender リクエスト

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Zabbix サーバーの応答

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

タイムスタンプ付きの Zabbix sender リクエスト

また、Zabbix sender はタイムスタンプとナノ秒を含むリクエストを送信できます。

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710794,
            "ns": 592397170
        },
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710795,
            "ns": 192399456
        }
    ],
    "clock": 1516712029,
    "ns": 873386094
}
```

Zabbix サーバーの応答

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Windows 向け **Zabbix sender** 動的リンクライブラリ

Windows 環境では、アプリケーションは外部プロセス（zabbix_sender.exe）を起動する代わりに、Zabbix sender 動的リンクライブラリ（zabbix_sender.dll）を使用して Zabbix サーバー/プロキシにデータを送信できます。

ユーザーアプリケーションを zabbix_sender.dll でコンパイルするには、zabbix_sender.h と zabbix_sender.lib が必要です。

入手方法

zabbix_sender.dll を入手する方法は 2 つあります。

1. ダウンロード して、zabbix_sender.h、zabbix_sender.lib、zabbix_sender.dll ファイルを ZIP アーカイブとして取得します。

ダウンロードオプションを選択する際は、*Encryption* で"No encryption"、*Packaging* で"Archive" を選択してください。その後、Zabbix agent（Zabbix agent 2 ではありません）をダウンロードします。

zabbix_sender.h、zabbix_sender.lib、zabbix_sender.dll ファイルは、ダウンロードした ZIP アーカイブ内の bin\dev ディレクトリにあります。必要な場所に展開してください。

2. ソースから zabbix_sender.dll をビルドします（手順 を参照）。

開発用ファイルを含む動的リンクライブラリは bin\winXX\dev ディレクトリに配置されます。使用するには、zabbix_sender.h ヘッダーファイルをインクルードし、zabbix_sender.lib ライブラリをリンクしてください。

関連情報

- Zabbix sender 動的リンクライブラリを使用して実装された、シンプルな Zabbix sender ユーティリティの例。ライブラリの使用方法を示しています。

- Zabbix sender 動的リンクライブラリのインターフェース関数については、zabbix_sender.h ファイルを参照してください。このファイルには、各インターフェース関数の目的、引数、戻り値を説明するドキュメントが含まれています。

取得

概要　Zabbix get は、Zabbix エージェントと通信して、エージェントから必要な情報を取得するために使用できるコマンドラインユーティリティです。

このユーティリティは、通常、Zabbix エージェントのトラブルシューティングに使用されます。

Zabbix get のように動作する組み込み機能を備えたZabbix 用 Python ライブラリ も参照してください。

Zabbix get の実行　UNIX で Zabbix get を実行して、エージェントからプロセッサ負荷の値を取得する例です:

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

Web サイトから文字列を取得するために Zabbix get を実行する別の例です:

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

ここでのアイテムキーにはスペースが含まれているため、シェルに対してアイテムキーを示すために引用符を使用しています。引用符はアイテムキーの一部ではありません。シェルによって取り除かれ、Zabbix エージェントには渡されません。

アイテムキーがサポートされていない場合、Zabbix get は終了コード 1 を返します。

Zabbix get は次のコマンドラインパラメーターを受け付けます:

```
-s --host <host name or IP>             ホストのホスト名または IP アドレスを指定
-p --port <port number>                 ホスト上で動作しているエージェントのポート番号を指定 (デフォルト: 10050)
-I --source-address <IP address>        ソース IP アドレスを指定
-t --timeout <seconds>                  タイムアウトを指定。有効範囲: 1-600 秒 (デフォルト: 30 秒)
-k --key <item key>                     値を取得するアイテムのキーを指定
-P --protocol <value>                   エージェントとの通信に使用するプロトコル。値:
                                            auto - JSON プロトコルを使用して接続し、プレーンテキストプロトコルでフォ
                                            json - JSON プロトコルを使用して接続する
                                            plaintext - アイテムキーのみを送信するプレーンテキストプロトコルを使用し
-h --help                               このヘルプメッセージを表示
-V --version                            バージョン番号を表示

--tls-connect <value>                   エージェントへの接続方法。値:
                                            unencrypted - 暗号化なしで接続する (デフォルト)
                                            psk - 事前共有鍵を使用して TLS で接続する
                                            cert - 証明書を使用して TLS で接続する
--tls-ca-file <CA file>                 ピア証明書検証用の最上位 CA 証明書を含むファイルの完全パス名
--tls-crl-file <CRL file>               失効した証明書を含むファイルの完全パス名
--tls-agent-cert-issuer <cert issuer>   許可されるエージェント証明書の発行者
--tls-agent-cert-subject <cert subject> 許可されるエージェント証明書のサブジェクト
--tls-cert-file <cert file>             証明書または証明書チェーンを含むファイルの完全パス名
--tls-key-file <key file>               秘密鍵を含むファイルの完全パス名
--tls-psk-identity <PSK-identity>       事前共有鍵を識別するために使用される、一意で大文字小文字を区別する文字列
--tls-psk-file <PSK-file>               事前共有鍵を含むファイルの完全パス名
--tls-cipher13 <cipher-string>          OpenSSL 1.1.1 以降向けの TLS 1.3 用暗号スイート文字列。既定の暗号スイート選
--tls-cipher <cipher-string>            TLS 1.2 以降向けの GnuTLS 優先度文字列、または TLS 1.2 のみ向けの OpenSSL
```

詳細については、Zabbix get の man ページも参照してください。

Windows での Zabbix get も同様に実行できます:

```
zabbix_get.exe [options]
```

JS

概要　zabbix_js は、組み込みスクリプトのテストに使用できるコマンドラインユーティリティです。

このユーティリティは、文字列パラメータを指定してユーザースクリプトを実行し、その結果を出力します。スクリプトは、組み込みの Zabbix スクリプトエンジンを使用して実行されます。

コンパイルまたは実行時にエラーが発生した場合、zabbix_js は stderr にエラーを出力し、終了コード 1 で終了します。

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

使用方法　zabbix_js は、次のコマンドラインパラメータを受け付けます。

```
-s, --script script-file          実行するスクリプトのファイル名を指定します
                                  ファイル名に "-" を指定した場合、スクリプトは stdin から読み込まれます
-i, --input input-file            入力情報のファイル名を指定します
                                  内容は値としてスクリプトに渡されます
                                  -p とは排他的です
                                  ファイル名に "-" を指定した場合、スクリプトは stdin から読み込まれます
-p, --param input-param           入力パラメータを指定します。値としてスクリプトに渡される変数です
                                  -i とは排他的です
-l, --loglevel log-level          ログレベルを指定します
                                  範囲: 0-5
-t, --timeout timeout             タイムアウトを秒単位で指定します
                                  有効範囲: 1-600 秒（デフォルト: 10 秒）
-h, --help                        ヘルプ情報を表示します
-V, --version                     バージョン番号を表示します
-w <webdriver url>                ブラウザ監視を有効にします
```

例: 例 1: 値を含む変数を使ってスクリプトを実行する

スクリプト (script-file.js):

```
return value;
```

使用方法:

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

出力: example_value

例 2: 入力情報を含むファイルを使ってスクリプトを実行する

スクリプトのファイル (script-file.js):

```
return value;
```

入力情報のファイル (example.txt):

```
ファイルからの入力情報の例
```

使用方法:

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

出力: ファイルからの入力情報の例

例 3: stdin から読み込んだ入力を使ってスクリプトを実行する

スクリプトのファイル (script-file.js):

```
return value;
```

使用方法:

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

入力 (stdin):

```
stdin からの入力の例
```

出力: stdin からの入力の例

Web サービス

概要　Zabbix web service は、外部の web サービスとの通信に使用されるプロセスです。現在、Zabbix web service はスケジュールレポート の生成と送信に使用されており、将来的には追加機能の実装も予定されています。

Zabbix サーバーは HTTP(S) 経由で web service に接続します。Zabbix web service には、同じホストにGoogle Chrome がインストールされている必要があります。一部のディストリビューションでは、Chromium でも動作する場合があります（既知の問題 を参照）。

インストール　公式の zabbix-web-service パッケージは、Zabbix リポジトリで入手できます。

Zabbix Web サービスをソースからコンパイルするには、`--enable-webservice` configure オプションを指定します。

Zabbix Web サービスを設定するには、Web サービス設定ファイルのパラメータを更新します。

Attention:
Zabbix サーバーと Zabbix Web サービス間の暗号化を、証明書を使用して設定することを強く推奨します。デフォルトでは、Zabbix サーバーと Zabbix Web サービス間で送信されるデータは暗号化されないため、不正アクセスにつながる可能性があります。

設定パラメーター**: Zabbix web service**

概要

Zabbix web service は、外部の web サービスとの通信に使用されるプロセスです。

Zabbix web service の設定ファイル (`zabbix_web_service.conf`) でサポートされるパラメータを、このセクションに示します。

パラメータは追加情報なしで一覧表示されています。パラメータをクリックすると、詳細を表示できます。

Parameter	Description
AllowedIP	カンマ区切りの IP アドレスの一覧。必要に応じて CIDR 表記、または Zabbix サーバーおよび Zabbix プロキシの DNS 名を指定できます。
DebugLevel	デバッグレベル。
IgnoreURLCertErrors	Web インターフェースの URL にアクセスする際の TLS 証明書検証エラーの処理方法を指定します。
Include	設定ファイル内で個別のファイル、またはディレクトリ内のすべてのファイルを含めることができます。
ListenPort	このサービスがサーバーからの接続を待ち受けるポートです。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力の種類。
Timeout	ダッシュボードの PDF report の書式設定に費やす最大時間（秒）。
TLSAccept	受け入れる受信接続の種類。
TLSCAFile	Zabbix コンポーネント間の暗号化通信で使用される、相手側証明書の検証用トップレベル CA 証明書を含むファイルの完全パス名。
TLSCertFile	Zabbix コンポーネント間の暗号化通信で使用される、サービス証明書または証明書チェーンを含むファイルの完全パス名。
TLSKeyFile	Zabbix コンポーネント間の暗号化通信で使用される、サービス秘密鍵を含むファイルの完全パス名。

すべてのパラメータは、明示的に必須と記載されていない限り、必須ではありません。

注意事項:

- デフォルト値は、配布される設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値はenvironment variables をサポートします。
- Zabbix がサポートする設定ファイルの文字エンコーディングは UTF-8 のみで、BOM は含められません。
- # で始まるコメントは、行頭でのみサポートされます。

パラメーターの詳細

AllowedIP

Zabbix サーバーおよび Zabbix プロキシの、カンマ区切りの IP アドレス一覧です。必要に応じて CIDR 表記、または DNS 名を指定できます。受信接続は、ここに一覧表示されたホストからのみ受け入れられます。
IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同等に扱われ、::/0 は任意の IPv4 または IPv6 アドレスを許可します。0.0.0.0/0 は任意の IPv4 アドレスを許可するために使用できます。

例:

127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

必須: はい

DebugLevel

デバッグレベルを指定します: - *0* - Zabbix プロセスの開始と停止に関する基本情報; - *1* - 重大な情報; - *2* - エラー情報; - *3* - 警告; - *4* - デバッグ用（大量の情報を出力します）; - *5* - 拡張デバッグ用（さらに多くの情報を出力します）。

デフォルト: 3
 範囲: 0-5

IgnoreURLCertErrors

Web インターフェースの URL にアクセスする際の TLS 証明書検証エラーの処理を指定します:
0 - 証明書エラーを無視しない;
1 - 証明書エラーを無視する.

デフォルト: 0
 範囲: 0-1

Include

設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
 Zabbix のインストール時、このパラメーターは /usr/local/etc に設定されますが、コンパイル時に変更した場合はその設定が反映されます。

Include パラメーターに関する一般的なルール:

- パスは zabbix_web_service.conf ファイルの場所からの相対パスにできます。
- 1 つの設定ファイル内で複数の Include パラメーターを指定できます。
- Zabbix は Include パラメーターを読み取るとすぐに処理するため、設定ファイルの末尾に配置することを検討してください。

含まれるファイルに関するルール:

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルは正しい構文でなければなりません。そうでない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンを含める際のルール:

- 読み込み順序を想定しないでください（たとえば、ファイルはアルファベット順には読み込まれません）。
- 同じパラメーターを複数の含まれるファイルで定義しないでください（たとえば、一般的な設定を個別の設定で上書きする場合など）。どの値が使用されるか予測できないためです。
- テキストエディターが自動的に作成するバックアップコピーに注意してください（たとえば、include/my_specific.conf を編集すると include/my_specific.conf.BAK が作成されることがあります）。両方のファイルが含まれるためです。

ディレクトリから関連するファイルのみを含めるには、ワイルドカード (*) をパターンマッチングに使用します。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

このサービスは、サーバーからの接続をこのポートで待ち受けます。

デフォルト: 10053
 範囲: 1024-32767

LogFile

ログファイルの名前。

例:

/tmp/zabbix_web_service.log

必須: はい。LogType が file に設定されている場合。そうでない場合は不要

LogFileSize

ログファイルの最大サイズ（MB）。
0 - 自動ログローテーションを無効にします。
 注: ログファイルサイズの上限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り詰められ、新規に開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力の種類: - *file* - LogFile パラメータで指定されたファイルにログを書き込みます。- *system* - syslog にログを書き込みます。- *console* - 標準出力にログを書き込みます。

デフォルト: `file`

タイムアウト

ダッシュボードの PDF レポート の書式設定に費やす最大時間（秒）。

デフォルト: 10
 範囲: 1-30

TLSAccept

受け入れる受信接続の種類: - *unencrypted* - 暗号化なしの接続を受け入れる（デフォルト）; - *cert* - TLS と証明書を使用した接続を受け入れる。

デフォルト: `unencrypted`

TLSCAFile

ピア証明書の検証に使用される、Zabbix コンポーネント間の暗号化通信向けに、最上位の CA 証明書を含むファイルの完全なパス名。

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用される、サービス証明書または証明書チェーンを含むファイルの完全なパス名。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用される、サービスの秘密鍵を含むファイルの完全なパス名。

MCP サーバー

概要　Zabbix MCP サーバーは、AI エージェントがModel Context Protocol（MCP）を使用して Zabbix と連携できるようにするプロセスです。MCP サーバーは、MCP クライアントおよびZabbix APIと通信します。

MCP サーバーを使用すると、AI エージェントは次の操作を実行できます。

- ホストグループ、テンプレートグループ、ホスト、テンプレート、アイテム、LLD ルール、インターフェース、マクロ、トリガー、障害、メンテナンス期間、およびアイテムの履歴データ（バイナリアイテムを除く）を取得する
- 障害を更新する: コメントの追加、認識または認識解除、深刻度の変更、抑制または抑制解除、障害のクローズ
- メンテナンス期間を作成、更新、削除する

認証には Zabbix API トークンが使用され、トークンはすべてのリクエストに付加されます。アクセスはトークン所有者の権限によって制限されます。つまり、そのユーザーが利用できるホストグループのデータのみが表示され、ユーザーに実行権限のない操作は拒否されます。

MCP サーバーはステートレスです。リクエスト間でセッションは保持されません。

> **Note:**
> Zabbix MCP サーバーは、古いバージョンまたは新しいバージョンの Zabbix でも動作する場合がありますが、互換性が公式にサポートされるのは、MCP サーバー自体と同じバージョンの Zabbix のみです。

関連項目:

- 設定パラメータ: MCP サーバー
- ツールとリソース

インストール　公式の zabbix-mcp-server パッケージは、Zabbix リポジトリで提供されています。

Zabbix MCP サーバーをソースからインストールするには、`make` でビルドします。

Zabbix MCP サーバーを設定するには、MCP サーバー設定ファイルのパラメータを更新します。少なくとも `FrontendURL` および `AllowedIP` パラメータを設定する必要があります。

デフォルトの設定ファイル（`zabbix_mcp_server.conf`）は`/etc/zabbix/`にあります。このファイルでは、機密データを公開する可能性が最も高いツールが拒否されます。ツールを有効にするには、対応する `DenyTool` 行を削除します。

関連項目：セキュリティのベストプラクティス

実行　MCP サーバーを起動するには、次のコマンドを実行します。

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

サーバーを起動せずに設定ファイルを確認して終了するには、-T オプションを使用します。

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

次のコマンドラインオプションがサポートされています。

オプション	説明
-c <file>	設定ファイルへのパス。 デフォルトは /etc/zabbix/zabbix_mcp_server.conf です。
-T	設定ファイルを検証して終了します。
-h	ヘルプ情報を表示します。
-V	バージョン番号を表示します。

MCP サーバーをソースからインストールした場合は、systemd ユニットを手動で作成する必要があります。例:

```
[Unit]
Description=Zabbix MCP server
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

MCP クライアントの接続　Streamable HTTP トランスポートをサポートする MCP クライアントは、MCP サーバーに接続できます。以下の設定が必要です。

設定	値
トランスポート	Streamable HTTP
URL	https://<host>:<ListenPort>/mcp
ヘッダー	Authorization: Bearer <Zabbix API token>（すべてのリクエストで送信）

各ユーザーは、自分の API トークンを設定する必要があります。

設定を確認するには、MCP サーバーに 1 回リクエストを送信します。

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

HTTP ステータスコード 200 とレスポンス内の利用可能なツールの一覧によって、MCP サーバーに到達でき、トークンが受け入れられたことを確認できます。この一覧には、公開されているツールが表示されます。その他のステータスコードは、以下の問題を示します。

- 400 - Accept ヘッダーに、トランスポートで必要な両方のメディアタイプが含まれていない。
- 401 - トークンを含む Authorization ヘッダーがない。
- 403 - クライアントアドレスが AllowedIP パラメータに記載されていない。

承認とツールアクセス　MCP クライアントがフォームによる情報要求をサポートしている場合、すべての変更は Zabbix に送信する前にユーザーの承認を受ける必要があります。MCP クライアントは変更内容をユーザーに表示し、ユーザーが確認した場合にのみ変更が適用されます。ユーザーが要求を拒否または閉じた場合、Zabbix には何も変更されません。承認の有効期間は 5 分間で、表示された変更内容と完全に一致する変更に対してのみ有効です。この期間内であれば、承認を再利用できます。

MCP クライアントがフォームによる情報要求をサポートしていない場合（URL による情報要求のみをサポートするクライアントを含む）、変更は承認なしで直ちに適用されます。この場合は、書き込みツールと Zabbix API の権限を制限してください。

監視対象システムからの信頼できないテキストがツールの出力に含まれ、エージェントに影響を与える可能性があるため、承認は特に重要です。

AllowTool、DenyTool、AllowToolRegexp、DenyToolRegexp を使用して、MCP サーバーが公開するツールを制御します。拒否されたツールは呼び出すことも一覧表示することもできず、関連付けられたリソースにもアクセスできません。

Warning:
AllowTool、DenyTool、AllowToolRegexp、DenyToolRegexp のルールは MCP へのアクセスを制限しますが、Zabbix トークンの権限は変更しません。Web インターフェースのアドレスが AI エージェントに公開されている場合、エージェントは直接 API リクエストを送信しようとする可能性があります。リクエストが MCP サーバーとそのセキュリティ制御を経由するように、Zabbix Web インターフェースの場所と MCP サーバーの設定にエージェントがアクセスできないようにしてください。

セキュリティのベストプラクティス　API トークン

認証と認可はすべて Zabbix によって処理されます。MCP クライアントは各リクエストに Zabbix API トークンを付加して送信します。MCP サーバーはトークンを保存またはログに記録せずに転送し、Zabbix がトークンを検証し、権限を適用して、操作を監査ログに記録します。

エージェントとユーザーごとに、権限を最小限に設定し、有効期限を設けた個別のトークンを使用してください。トークンは安全な場所に保管し、API トークンを定期的にローテーションしてください。

TLS 暗号化

トークンが平文で送信されないようにするには、`TLSAccept=cert` を設定し、証明書とキー（`TLSCertFile` および `TLSKeyFile`）を指定するか、信頼できる TLS 終端リバースプロキシの背後で MCP サーバーを実行し、`TLSAccept=unencrypted` を設定します。暗号化されていない HTTP は、信頼できるネットワーク内でのみ使用できます。

アクセス制限

`AllowedIP` パラメータを信頼できる MCP クライアントのアドレスに制限し、Zabbix の権限を使用して API トークンで実行できる操作を制限してください。

MCP サーバーはリクエストのレート制限を提供しません。通常、各 MCP リクエストは 1 回の Zabbix API 呼び出しになりますが、ループで実行されるエージェントによって大きな負荷が発生する可能性があります。

機密データの保護

アイテムキーには認証情報（SSH、JMX、ODBC、HTTP エージェントのアイテムなど）が含まれる場合があり、マクロの値にはパスワードやコミュニティ文字列が含まれる場合があります。また、インターフェースからは内部アドレスが明らかになり、ログまたはテキストアイテムの値にはあらゆるデータが含まれる可能性があります。このため、`item_get`、`lld_get`、`interface_get`、`macro_get`、`history_get` ツールは、提供される設定ファイルではデフォルトで拒否されています。

接続されたエージェントの LLM プロバイダーに機密性の高いマクロ値が公開されるのを防ぐには、シークレットマクロを使用してください。シークレットマクロの値が API から返されることはありません。

ツールとリソース

概要

Zabbix MCP サーバーは、Zabbix の機能を MCP クライアントにツールおよびリソースとして公開します。

ツール名はアクティビティを表します。ToolPrefixパラメータが設定されている場合、すべてのツール名にプレフィックスが追加されます。ツールアクセスルールでは、以下に示すプレーンな名前が照合されます。

読み取りツール

ツール	Zabbix API メソッド	引数	並べ替え基準	リソース
hostgroup_get	hostgroup.get	groupids, names, search, limit	name	zabbix://hostgroup/<i
templategroup_get	templategroup.get	groupids, names, search, limit	name	zabbix://templategrou
host_get	host.get	hostids, groupids, search, limit	name	zabbix://host/<id>
template_get	template.get	templateids, groupids, search, limit	name	zabbix://template/<id
item_get	item.get	hostids, itemids, search, limit	name	zabbix://item/<id>
lld_get	discoveryrule.get	hostids, ruleids, limit	name	-
interface_get	hostinterface.get	hostids, interfaceids, limit	interface ID	-
macro_get	usermacro.get	hostids または global, macro_names, limit	macro	-
problem_get	problem.get	eventids, groupids, hostids, severities, acknowledged, suppressed, recent, limit	event ID、降順	-

ツール	Zabbix API メソッド	引数	並べ替え基準	リソース
maintenance_get	maintenance.get	maintenanceids, groupids, hostids, limit	name	-
history_get	history.get	itemids, time_from, time_till, type, limit	clock、降順	-
trigger_get	trigger.get	triggerids, hostids, search, limit	description	zabbix://trigger/<id>

次の点に注意してください。

- host_get と template_get は、各オブジェクトのトリガー（限定されたフィールドのセット）、タグ、リンクされたテンプレートも返します。アイテム、ディスカバリールール、インターフェース、マクロは、対応するツールによって返されます。
- history_get では、7 日間を超えない時間範囲を指定する必要があり、開始時刻は 7 日前より前にすることはできません。type 引数には float、unsigned、character、text、log、json のいずれかを指定し、アイテムの値の形式と一致させる必要があります。形式が正しくない場合は空の結果が返されますが、これはデータのないアイテムの場合と区別できません。バイナリアイテム（値の形式 5）の履歴は、MCP サーバーを介して取得できません。
- 読み取りツールは固定されたフィールドのリストを要求します。すべてのフィールド（output: extend）が要求されることはありません。
- リソースを持つツールの結果には uri フィールドが含まれるため、エージェントは一覧から単一のエンティティへ進むことができます。

レスポンス形式

すべての読み取りツールは、同じ形式でレスポンスを返します。

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

レスポンスは、limit 引数（デフォルト: 100、最大: 1000）と、レスポンスのシリアライズ後のサイズ（MaxResponseSize、デフォルト: 64K）によって制限されます。いずれかの制限に達すると、レスポンスは停止します。レスポンスは結果の間でのみ切り詰められ、結果の途中で切り詰められることはありません。また、切り詰められたレスポンスには、どの制限に達したかが示されます。すべてのツールには固定されたソート順があるため、切り詰められたレスポンスは常に完全な結果セットの先頭部分になります。ページングはサポートされていません。別の結果を取得するには、クエリを絞り込んでください。

障害ツール

すべての障害ツールは、対応するアクションフラグを指定してevent.acknowledge を呼び出します。
eventids 引数は必須です。イベント ID は problem_get で取得できます。
任意の障害ツールでメッセージ（コメント）を追加できます。
message 引数は、problem_annotate を除いて任意です。指定した場合、メッセージは同じ Zabbix API 呼び出しで記録されます。

Tool	Effect	Additional arguments
problem_acknowledge	障害を承認します。	-
problem_unacknowledge	承認を解除します。	-
problem_close	障害を手動でクローズします（手動クローズを許可するトリガーの場合のみ）。	-
problem_annotate	メッセージを記録します。	message（必須）
problem_change_severity	障害の深刻度を変更します。	severity（0-5）
problem_suppress	指定した時刻まで、または無期限に障害を抑制します。	suppress_until
problem_unsuppress	抑制を解除します。	-

障害を作成または削除することはできません。すべての障害ツールは既存の障害を更新します。

メンテナンスツール

ツール	Zabbix API メソッド	引数
maintenance_create	maintenance.create	name、groupids および/または hostids、active_since、active_till、timeperiods、description、tags、tags_evaltype
maintenance_update	maintenance.update	maintenanceid および上記の引数のいずれか。リスト引数は値のセット全体を置き換えます。空のリストを指定すると、セットがクリアされます。

ツール	Zabbix API メソッド	引数
maintenance_delete	maintenance.delete	maintenanceids

メンテナンス期間を更新すると、リスト引数（hostids、groupids、timeperiods など）は既存の値に追加されるのではなく、既存の値を置き換えます。たとえば、メンテナンス期間がホスト A とホスト B を対象としていて、hostids にホスト C のみを含めて maintenance_update を呼び出した場合、メンテナンス期間の対象はホスト C のみになります。

maintenance_create は冪等ではない唯一の操作であり、maintenance_delete は元に戻せない唯一の操作です。これらの使用を防止するには、DenyTool パラメータでこれらのツールを拒否します。

書き込みツールの動作

障害およびメンテナンスツールは書き込みツールです。次のように動作します。

- Zabbix API を呼び出す前に引数が検証されます。検証に失敗した場合、満たされていない制約がエラーに示されます。
- クライアントが対応している場合、変更はユーザーによる承認後にのみ適用されます。
- 変更は、呼び出し元のトークンを使用して Zabbix によって実行されます。Zabbix は権限を確認し、トークン所有者に対する監査ログに変更を記録します。
- 権限エラーを含む Zabbix のエラーは、code、message、data フィールドを持つエラーオブジェクトとして変更せずに返されます。
- 書き込みツールは Zabbix の結果を直接返します。読み取りツールの応答制限は適用されません。
- 書き込み呼び出しが MCP サーバーによって再試行されることはありません。

リソース

リソースでは、zabbix://<kind>/<id> URI スキームを使用します。resources/list のレスポンスに含まれるのは zabbix://about のみです。エンティティ ID は列挙できず、ツールの結果から取得します。resources/templates/list のレスポンスには、エンティティの種類が一覧表示されます。

URI	内容
zabbix://about	API URL、Web インターフェースの URL、Zabbix のバージョン、呼び出し元のトークンが解決される ID、および公開されているツールの一覧。
zabbix://host/<hostid>	アイテム、トリガー、インターフェース、マクロを含まないホストレコード。
zabbix://hostgroup/<groupid>	ホストの一覧を含まないホストグループレコード。
zabbix://template/<templateid>	トリガー、ディスカバリルールを含まないテンプレートレコード。
zabbix://templategroup/<groupid>	テンプレートの一覧を含まないテンプレートグループレコード。
zabbix://item/<itemid>	現在の値を含まないアイテム定義。
zabbix://trigger/<triggerid>	依存関係、関数、タグを含まないトリガー定義。

対応するツールへのアクセスが拒否されているリソースは存在しません。存在しないエンティティと、トークン所有者がアクセス権を持たないエンティティは、どちらも *resource not found* として報告されます。これらのケースは意図的に区別できないようになっています。zabbix://about リソースは、ツールのアクセスルールの影響を受けません。

設定パラメータ**: Zabbix MCP** サーバー

概要

Zabbix MCP サーバーは、AI エージェントが Model Context Protocol（MCP）を使用して Zabbix を操作できるようにするプロセスです。

このセクションでは、Zabbix MCP サーバーの設定ファイル（zabbix_mcp_server.conf）でサポートされるパラメータを一覧表示します。

パラメータは追加情報なしで一覧表示されています。パラメータをクリックすると、詳細を確認できます。

パラメータ	説明
AllowedIP	MCP クライアントの、カンマ区切りの IP アドレス（CIDR 表記も使用可能）または DNS 名のリスト。
AllowTool	ツールを許可するワイルドカードルール。
AllowToolRegexp	ツールを許可する正規表現ルール。
DebugLevel	デバッグレベル。
DenyTool	ツールを拒否するワイルドカードルール。
DenyToolRegexp	ツールを拒否する正規表現ルール。

パラメータ	説明
FrontendURL	Zabbix API エンドポイント。
IgnoreURLCertErrors	Zabbix API へのアクセス時における TLS 証明書の検証エラーの処理方法を指定します。
ListenIP	待ち受ける IP アドレス。
ListenPort	待ち受けるポート。
LogFile	ログファイルの名前。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力のタイプ。
MaxResponseSize	読み取りツールが返すレスポンスの最大サイズ。
PidFile	PID ファイルの名前。
Timeout	Zabbix API からのレスポンスを待機する最大時間（秒）。
TLSAccept	受け入れる受信接続。
TLSCertFile	サーバー証明書または証明書チェーンを含むファイルの完全なパス名。
TLSKeyFile	サーバーの秘密鍵を含むファイルの完全なパス名。
ToolPrefix	すべてのツール名に追加されるプレフィックス。

パラメータが必須であることが明示されていない限り、すべてのパラメータは任意です。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、プロセスのデフォルト値を示します。
- 値では、`${NAME}` 形式の環境変数を使用できます。
- Zabbix は、BOMなしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- # で始まるコメントは、行の先頭でのみサポートされます。

パラメータの詳細

AllowedIP

MCP クライアントの IP アドレス（CIDR 表記も使用可能）または DNS 名をカンマ区切りで指定したリストです。受信接続は、ここに記載されたホストからのみ受け入れられます。

例:

```
AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com
```

必須: はい

AllowTool

ツールを許可するワイルドカードルールです。このパラメータは複数回指定できます。
 ワイルドカードパターンにはアイテムキーの構文を使用し、* は任意の文字列に一致します。このルールはプレーンなツール名に一致し、ToolPrefix は無視します。
 すべての AllowTool、DenyTool、AllowToolRegexp、DenyToolRegexp ルールは、指定された順序で単一のリストを形成します。最初に一致したルールによって判定されます。どのルールにも一致しないツールは許可されます。

例:

```
AllowTool=host_get
```

AllowToolRegexp

ツールを許可する正規表現ルールです。このパラメータは複数回指定できます。
 正規表現には RE2 構文を使用し、アンカーがない場合はプレーンなツール名の任意の部分に一致します。

例:

```
AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$
```

DebugLevel

デバッグレベルを指定します。- *0* - Zabbix プロセスの開始および停止に関する基本情報 - *1* - 重大な情報 - *2* - エラー情報 - *3* - 警告 - *4* - デバッグ用（大量の情報を出力）- *5* - 拡張デバッグ用（さらに大量の情報を出力）

デフォルト: 3
 範囲: 0-5

DenyTool

ツールを拒否するワイルドカードルールです。このパラメータは複数回指定できます。
 拒否されたツールはツールの一覧に含まれず、呼び出すこともできません。詳細については、AllowToolを参照してください。

例:

```
DenyTool=maintenance_delete
```

DenyToolRegexp

ツールを拒否する正規表現ルールです。このパラメータは複数回指定できます。
 詳細については、AllowToolRegexpを参照してください。

例:

```
DenyToolRegexp=^problem_(close|suppress)$
```

FrontendURL

Zabbix API エンドポイント: `api_jsonrpc.php` で終わる Zabbix Web インターフェースの URL。

例:

```
FrontendURL=https://zabbix.example.com/api_jsonrpc.php
```

必須: はい

IgnoreURLCertErrors

Zabbix API へのアクセス時における TLS 証明書の検証エラー処理を指定します。
 false - 証明書エラーを無視しない;
 true - 証明書エラーを無視する（開発環境でのみ使用してください）。

デフォルト: `false`

ListenIP

リッスンする IP アドレス。

デフォルト: `0.0.0.0`

ListenPort

リッスンするポート。MCP エンドポイントのパスは `/mcp` です。

デフォルト: 8443
 範囲: 1024-32767

LogFile

ログファイルの名前。

デフォルト: `/tmp/zabbix_mcp_server.log`

必須: いいえ

LogFileSize

ログファイルの最大サイズ（MB）。
0 - 自動ログローテーションを無効にします。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力のタイプ: - *file* - LogFile パラメータで指定したファイルにログを書き込む; - *system* - syslog にログを書き込む; - *console* - 標準出力にログを書き込む。

デフォルト: `file`

MaxResponseSize

read ツールによって返されるレスポンスの最大サイズ。この制限とツールの `limit` 引数のいずれか早く到達した方によって、レスポンスが停止します。
 単位のサフィックスをサポートします。

デフォルト: 64K
 範囲: 16K-256K

PidFile

PID ファイルの名前。

デフォルト: `/tmp/zabbix_mcp_server.pid`

タイムアウト

Zabbix API からの応答を待機する最大時間（秒）。

デフォルト: 10
 範囲: 3-30

TLSAccept

受信接続で許可する接続方式: - 暗号化なし - 暗号化されていない接続を許可します（デフォルト）。- 証明書 - TLS および証明書を使用した接続を許可します。

デフォルト: unencrypted

TLSCertFile

サーバー証明書または証明書チェーンを含むファイルの完全なパス名。

必須: TLSAccept が cert に設定されている場合は「はい」、それ以外の場合は「いいえ」

TLSKeyFile

サーバーの秘密鍵を含むファイルの完全なパス名。

必須: TLSAccept が cert に設定されている場合は「はい」、それ以外の場合は「いいえ」

ToolPrefix

すべてのツール名に追加されるプレフィックスです。たとえば、他の MCP サーバーのツールとの名前の競合を避けるために使用できます。ToolPrefix=zbx_ の場合、host_get ツールは zbx_host_get として公開されます。
 使用できる文字: 英字、数字、アンダースコア（_）、ハイフン（-）、ドット（.）。
 ツールアクセスルールでは、プレフィックスなしのツール名が使用されます。

例:

```
ToolPrefix=zbx_
```

ブリッジアダプター

Zabbix ブリッジアダプターを使用すると、Zabbix サーバーとZabbix モバイルアプリ間で通信できます。

Zabbix モバイルアプリは、Zabbix サーバーに直接接続しません。代わりに、2 つのコンポーネントが両者の間でデータを受け渡します。

- Zabbix Bridge - モバイルデバイス上の Zabbix モバイルアプリとデータを交換する、中央ホスト型のサービス。
- Zabbix ブリッジアダプター - インストールして実行するコンポーネント。Zabbix サーバーを Zabbix Bridge に接続します。

ブリッジアダプターは、パッケージを使用して、Zabbix サーバーと同じシステム、または別のシステムにインストールできます。

Zabbix ブリッジアダプターを設定するには、その設定ファイル(デフォルトのパス: /etc/zabbix/zabbix_bridge_adapter.conf)を更新します。

Zabbix ブリッジアダプターのセキュリティを確保する方法については、モバイルアクセスのセキュリティ保護を参照してください。

データフロー　Zabbix モバイルアプリを使用すると、データは次のように送信されます。

1. Zabbix モバイルアプリで操作を行うと、Zabbix Bridge にクエリが送信されます。
2. Zabbix Bridge は、JSON-RPC リクエストとしてクエリを Zabbix bridge adapter に渡します。
3. Zabbix bridge adapter は、要求された Zabbix API メソッドが許可されているかどうかを確認します。
   メソッドが許可されていない場合、bridge adapter はリクエストを拒否します。
   メソッドが許可されている場合、bridge adapter はリクエストを Zabbix API に送信します。
4. Zabbix API はリクエストを処理し、レスポンスを Zabbix bridge adapter に返します。
5. Zabbix bridge adapter はレスポンスを Zabbix Bridge に返します。
6. Zabbix Bridge はレスポンスを Zabbix モバイルアプリに渡し、Zabbix モバイルアプリはモバイルデバイスにデータを表示します。

設定パラメータ**: Zabbix** ブリッジアダプター

概要

設定パラメータを使用すると、Zabbix ブリッジアダプターの動作をカスタマイズできます。

パラメータ値は設定ファイル（zabbix_bridge_adapter.conf）で指定します。パラメータが必須であることが明示されていない限り、すべてのパラメータは任意です。

このページは次のように構成されています。

- パラメータの概要（パラメータの詳細を確認するには、その名前をクリックしてください）
- パラメータの詳細

パラメータ	説明
AllowedAPIMethods	Zabbix ブリッジアダプター経由で呼び出すことができる Zabbix API メソッドを指定します。
DataDir	ブリッジアダプターの状態ファイルと暗号化キーを作成および保存するディレクトリを指定します。
DebugLevel	デバッグレベルを設定します。
Include	設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
ListenIP	Zabbix サーバーからの HTTP(S) 接続を受け付けるブリッジアダプターの待ち受け IP アドレスを指定します。
ListenHealthPort	HTTP ヘルスチェック接続を受け付けるブリッジアダプターの待ち受けポートを指定します。
ListenPort	Zabbix サーバーからの HTTP(S) 接続を受け付けるブリッジアダプターの待ち受けポートを指定します。
LogFile	ブリッジアダプターのログファイルへのフルパスを指定します。
LogFileSize	ログファイルの最大サイズ（MB）を設定します。
LogType	ログの出力タイプを指定します。
PidFile	PID ファイルを指定します。
TLSConfig.AllowedCN	許可するクライアントの Common Name（mTLS のみ）のカンマ区切りリストを指定します。
TLSConfig.CAFile	mTLS モードでクライアント証明書の検証に使用する最上位 CA 証明書を含むファイルへのフルパスを指定します。
TLSConfig.CertFile	証明書または証明書チェーンを含むファイルへのフルパスを指定します。
TLSConfig.KeyFile	秘密鍵を含むファイルへのフルパスを指定します。
TLSConfig.Mode	ListenPort で受信する接続の TLS モードを設定します。
ServerId	Zabbix サーバーの一意の識別子を指定します。必須。
ZabbixAPITLSCAFile	ブリッジアダプターが Zabbix API 証明書の検証に使用する最上位 CA 証明書を含むファイルへのフルパスを指定します。
ZabbixAPITLSConnect	Zabbix API への送信接続の TLS モードを設定します。
ZabbixAPIURL	Zabbix API エンドポイントの URL を指定します。必須。
ZabbixBridgeURL	Zabbix Bridge 設定の取得に使用する Zabbix Bridge URL を指定します（ブリッジアダプターの起動時に取得されます）。

次の点に注意してください。

- デフォルト値は、提供されている設定ファイルの値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数を使用できます。
- Zabbix がサポートする設定ファイルは、BOMなしの UTF-8 エンコーディングのみです。
- # で始まるコメントは、行の先頭でのみサポートされます。

パラメータの詳細

AllowedAPIMethods

Zabbix bridge adapter を介して呼び出すことができる Zabbix API メソッドを指定します。
 複数のメソッドを指定する場合は、パラメータをメソッドごとに 1 行ずつ繰り返します。

各パラメータ値には、完全一致するメソッド名（例: `problem.get`）、API 名全体のみを置き換えるワイルドカードパターン（例: `*.get`）、またはメソッド名全体を置き換えるワイルドカードパターン（例: `problem.*`）のいずれかを指定する必要があります。

その他の値は無効であり、Zabbix bridge adapter の起動を妨げます。これには、単独のワイルドカード（`*`）、部分的なワイルドカードパターン（`host*` または `host*.*`）、空の値（`AllowedAPIMethods=`）が含まれます。

例:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

Zabbix モバイルアプリに必要なメソッドの最小セット:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
```

```
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

デフォルト: 未設定（すべての Zabbix API メソッドが拒否されます）

DataDir

ブリッジアダプターの状態ファイルと暗号化キーを作成および保存するディレクトリを指定します。
 ディレクトリには 0700 の権限が必要で、ディレクトリ内に保存されるすべてのファイルには 0600 の権限が必要です。

デフォルト: `/var/lib/zabbix-bridge-adapter`

DebugLevel

デバッグレベルを設定します。

- 0 - Zabbix プロセスの開始および停止に関する基本情報
- 1 - 重大な情報
- 2 - エラー情報
- 3 - 警告
- 4 - デバッグ用（大量の情報を出力）
- 5 - 拡張デバッグ（さらに多くの情報を出力）

デフォルト: 3
 範囲: 0-5

Include

設定ファイルに含めるディレクトリまたは個別のファイルを指定します。
 Zabbix のインストール時、このパラメータはコンパイル時に変更しない限り、`/usr/local/etc` に設定されます。

`Include` パラメータに関する一般的なルール:

- パスは、`zabbix_bridge_adapter.conf` ファイルの場所を基準とした相対パスにできます。
- 1 つの設定ファイルに複数の `Include` パラメータを指定できます。
- Zabbix は `Include` パラメータを読み取るとすぐに処理するため、設定ファイルの末尾に配置することを検討してください。

インクルードされるファイルに関するルール:

- 各ファイルは読み取り可能でなければなりません。
- 各ファイルの構文は正しくなければなりません。正しくない場合、コンポーネントは起動しません。

ディレクトリまたはファイルパターンをインクルードする場合のルール:

- インクルードの順序を想定しないでください（ファイルはアルファベット順にインクルードされるとは限りません）。
- 複数のインクルードファイルで同じパラメータを定義しないでください（例えば、一般的な設定を特定の設定で上書きする場合）。どの値が使用されるかを予測できないためです。
- 一部のテキストエディタが自動的に作成するバックアップコピーに注意してください(例えば､`include/my_specific.conf` を編集すると、`include/my_specific.conf.BAK` が作成される場合があります)。この場合、両方のファイルがインクルードされます。

ディレクトリから関連するファイルのみをインクルードするには、パターンマッチングにワイルドカード（`*`）を使用します。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Zabbix サーバーからの HTTP(S) 接続を受信するブリッジアダプターのリッスン IP アドレスを指定します。
 指定しない場合、ブリッジアダプターはすべての IP アドレスでリッスンします。

ListenHealthPort

HTTP ヘルスチェック接続を受信するブリッジアダプターのリッスンポートを指定します。
 `/health` および `/ready` エンドポイントを提供します。
 指定しない場合、ヘルスサービスは起動しません。

デフォルト: 8081
 範囲: 1024-32767

ListenPort

Zabbix サーバーからの HTTP(S) 接続を受信するブリッジアダプターのリッスンポートを指定します。

デフォルト: 10054
 範囲: 1024-32767

LogFile

ブリッジアダプターのログファイルへのフルパスを指定します。

必須: LogType が file に設定されている場合は はい

デフォルト: /tmp/zabbix_bridge_adapter.log

LogFileSize

ログファイルの最大サイズ（MB）を設定します。
 0 - 自動ログローテーションを無効にします。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力タイプを指定します。

- file - LogFile パラメータで指定したファイルにログを書き込みます。
- system - syslog にログを書き込みます。
- console - 標準出力にログを書き込みます。

デフォルト: file

PidFile

PID ファイルを指定します。

デフォルト: /tmp/zabbix_bridge_adapter.pid

ServerId

Zabbix サーバーの一意の識別子を指定します。
 Zabbix ユーザーインターフェースのレポート > システム情報で確認できます。

必須: はい

TLSConfig.AllowedCN

許可するクライアントの Common Name のカンマ区切りリストを指定します（mTLS のみ）。
 指定しない場合、ブリッジアダプターはクライアントの Common Name をチェックしません。

TLSConfig.CAFile

mTLS モードでクライアント証明書の検証に使用する、最上位 CA 証明書を含むファイルの完全なパスを指定します。

TLSConfig.CertFile

証明書または証明書チェーンを含むファイルの完全なパスを指定します。

TLSConfig.KeyFile

秘密鍵を含むファイルのフルパスを指定します。

TLSConfig.Mode

ListenPort で受信接続の TLS モードを設定します。

- none - 暗号化なしで接続を受け入れます。
- tls - 最小 TLS v1.2 の標準 TLS を使用します。
- mtls - 相互認証によって TLS を拡張します。

デフォルト: none

ZabbixAPITLSCAFile

ブリッジアダプターが Zabbix API の証明書を検証するために使用する、最上位 CA 証明書を含むファイルの完全なパス名を指定します。

必須: ZabbixAPITLSConnect が verify_ca または verify_full に設定されている場合は はい。

ZabbixAPITLSConnect

Zabbix API への送信接続の TLS モードを設定します。

- disabled（または空）- プレーン HTTP を使用します。ZabbixAPIURL には http:// を使用する必要があります。
- required - 証明書の検証なしで HTTPS を使用します（テスト専用）。
- verify_ca - HTTPS を使用し、Zabbix API の証明書を検証します。
- verify_full - HTTPS を使用し、Zabbix API の証明書とホスト名を検証します。

ZabbixAPIURL

Zabbix API エンドポイントの URL を指定します。

必須: はい

例:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
```

ZabbixBridgeURL

Zabbix Bridge の設定取得用 URL を指定します（ブリッジアダプターの起動時に取得されます）。

デフォルト: `ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration`

環境変数

概要　環境変数を使用すると、設定ファイルに値をハードコードすることなく Zabbix コンポーネントを設定できます。これにより、Docker のような動的な環境でも、実行時に変数を渡してさまざまな構成に対応できるため、設定を簡単に管理できます。

最も簡単な例として、Zabbix サーバーのDebugLevel 設定パラメータの値を環境変数に設定すると、起動時にそれを使ってサーバーを設定できます。

```
### Zabbix server configuration file:
DebugLevel=${NEW_DEBUG_LEVEL}

### Starting Zabbix server:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

環境変数は、以下の Zabbix コンポーネントでサポートされています。

- サーバー
- プロキシ
- エージェント (UNIX またはWindows)
- エージェント 2 (UNIX またはWindows)、プラグイン を含む
- Web サービス
- Zabbix sender（-c, --config option を使用する場合）

重要な注意事項

- 設定パラメータが環境変数に設定されている場合でも、コンポーネントの起動時にその環境変数が指定されていないと、そのパラメータのデフォルト値が使用されます。
- runtime commands を使用する場合（例: エージェントのログレベルを上げる場合）、以前に使用していた環境変数をすべて指定する必要があります。これは、Zabbix コンポーネントが runtime commands の実行に設定ファイルを使用するためです。環境変数が省略されると、設定パラメータのデフォルト値が使用されます。Examples を参照してください。
- `userparameter_reload` runtime command は、環境変数の再読み込みをサポートしていません。再読み込み時には変数は無視され、通常の値を持つパラメータのみが再読み込みされます。
- 設定ファイルで使用されていた現在の環境変数は、Zabbix コンポーネントの起動後にクリアされます。これにより、子プロセス（例: Zabbix によって実行されるリモートスクリプト）がこれらの変数にアクセスできなくなります。ただし、プロセスの初期変数は引き続き取得できることに注意してください（例: `/proc/<PID>/environ` ファイル経由）。

構文　環境変数は次の構文を使用する必要があります: `${英数字/アンダースコア}`。

変数名に含めることができるのは、英字 (a-z, A-Z)、アンダースコア (_)、および数字 (0-9) のみで、数字で始めてはいけません。

必要な構文に一致しない変数、または通常の値と組み合わされた変数は通常の値として扱われ、エラーが発生する場合があります。

正しい変数の構文:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

誤った変数の構文:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

Note:
Windows では、環境変数名は大文字と小文字を区別しません。

例　以下の例では、Zabbix コンポーネントで環境変数を設定および使用する方法を示します。

例 1: Zabbix エージェントの設定とテスト

1. エージェント設定ファイルで環境変数を設定します。

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. 設定ファイルをテストします。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. 環境変数を指定してエージェントを起動します。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

ランタイムコマンドを使用する場合（エージェントのログレベルを上げる場合など）、以前に使用した環境変数を指定する必要があります。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

これは、エージェントが設定ファイルを使用してランタイムコマンドを実行するためです。環境変数を省略すると、設定パラメータのデフォルト値が使用されます。

また、エージェント設定ファイルで環境変数を設定した後、プロセスから利用できるようにすることもできます（export コマンドを使用するなど）。これにより、変数が不足している、または正しく設定されていないことによる予期しない動作のリスクを軽減できます。

```
export ZBX_HOSTNAME="New Zabbix agent"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

例 2: コンテナ用の Zabbix エージェントの設定

Zabbix コンポーネント（Zabbix エージェントなど）用の独自のカスタムイメージを作成して設定する場合は、環境変数を使用して設定パラメータを定義し、それらの変数を指定してコンテナを起動できます。

1. イメージを準備する際に、エージェント設定ファイルで環境変数を設定します。

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. コンテナイメージをビルドした後、環境変数を指定してエージェントコンテナ（Docker など）を起動します。

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. ランタイムコマンドを使用する場合（エージェントのログレベルを上げる場合など）は、コンテナのシェルにアクセスしてランタイムコマンドを実行します。

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
userparameter_reload ランタイムコマンドは、環境変数の再読み込みをサポートしていません。再読み込み中は変数が無視され、通常の値を持つパラメータのみが再読み込みされます。

設定

概要　Zabbix の設定では、どのホストとシステムを監視するか、どのデータを収集するか、そして障害が発生したときに通知をどのように配信するかを指定します。

サイドバーを使用して、関連する設定項目に移動してください。

主要な設定タスク

- 新しいホストまたはホストグループを設定する → ホストとホストグループ
- ホストから収集するデータを定義する → アイテム
- 障害を検出するためのルールを作成する → トリガーによる障害検出
- 通知の配信方法を設定する → イベント発生時の通知
- ダッシュボード、グラフ、マップを通じて監視データを表示する → 可視化
- 複数のホスト間で設定を再利用する → テンプレートとテンプレートグループ, 標準搭載のテンプレート
- ユーザーアクセスと権限を管理する → ユーザーとユーザーグループ

設定のベストプラクティス　設定の選択は、パフォーマンスと保守性に影響します。Zabbix の構成はそれぞれ異なりますが、以下のベストプラクティスを基盤として、必要に応じて調整できます。

各監視インスタンスごとに個別のホストを作成する

Zabbix のホストは、物理的なマシンやデバイスではなく、論理的なエンティティです。
仮想マシン、データベース、コンテナ、ネットワークスイッチなど、独立した各インスタンスごとに個別のホストを作成してください。

この方法には、次の利点があります。

1. 各ホストごとにアイテム、トリガー、アラート通知を分けられるため、監視データが整理されます。
2. ユーザーロール と最小権限の原則 に沿った、きめ細かなユーザーアクセス制御が可能になります。

テンプレートと一括更新を類似ホストに使用する

Network switch 1 や *Network switch 2* のような類似したホストでは、必要なメトリクスを含むテンプレート (*Network switch template*) を作成し、ホストをホストグループ (*Network switches*) にまとめることができます。

新しいスイッチを追加する際は、*Data Collection* > *Hosts* で一括更新を使用して、ホストを手動で複製する代わりにテンプレートをリンクできます。

マスターアイテムと依存アイテムを設定してリクエストを最小化する

対象エンティティへのリクエストを最小限に抑えるには、マスターアイテムを設定して 1 回のリクエストでデータを収集し、その後、前処理を行う依存アイテムを使用して特定の値を抽出します。

たとえば、マスターアイテムで複数のメトリクスを含む JSON または XML の応答を収集したり、複数の列を返すデータベースクエリを実行したりできます（例: オープン接続数、切断された接続数、同時接続の最大許容数、起動以降の累積接続総数）。依存アイテムは、それぞれの値を個別に解析して保存します。

収集後はマスターアイテムの履歴を破棄し、依存アイテムのデータのみを保持します。

必要に応じて Zabbix プロキシをデプロイする

すべてのホストが Zabbix サーバーと同じローカルネットワーク上にあり、スケーラビリティやパフォーマンスに関する懸念がない場合は、プロキシは必要ない可能性があります。より大規模または複雑な環境では、プロキシによって負荷をより均等に分散できます。

次の場合は、Zabbix プロキシを追加します。

1. ファイアウォールの背後にある複数のホストを、さまざまなメトリクス収集方法で監視する場合。プロキシはホストからデータを収集し、Zabbix サーバーに転送するため、ファイアウォールで開放するポート数を減らせます。
2. リモート拠点、支店、またはネットワークを監視する場合。リモート拠点の Zabbix プロキシは、ネットワークが中断している間もデータ収集を継続し、接続が復旧するとデータを送信します。
3. 大規模なデプロイメントを管理する場合。多数のホストがある場合や、1 秒あたりに大量の値を収集する場合は、プロキシをデプロイしてサーバーの負荷を軽減し、パフォーマンスを向上させます。

独立したシークレットマクロ取得の設定

secret ユーザーマクロは、シークレットテキストまたはシークレットボールトマクロのいずれかとして使用できます。

デフォルトでは、シークレットマクロの値は Zabbix サーバーによって取得され、プロキシに伝播されます。シークレットボールトマクロのセキュリティをさらに強化するには、マクロ値を設定 して、Zabbix サーバーとプロキシがそれぞれ独立して取得するようにします。

テンプレートの設定

概要

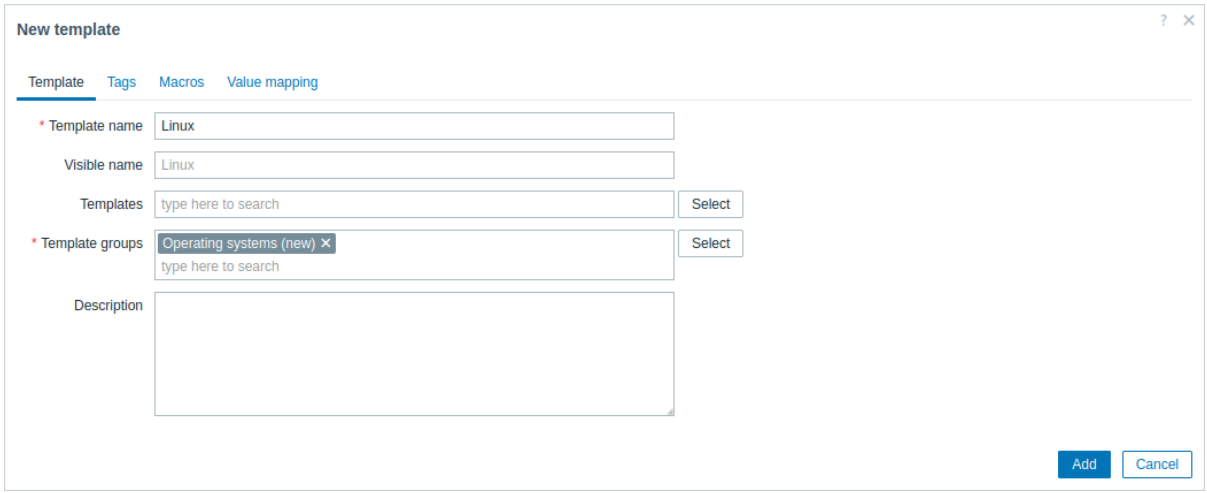
テンプレートの設定は、まずテンプレートを作成し、その一般的なパラメータを定義し、次にエンティティ（アイテム、トリガー、グラフなど）を追加する必要があります。

テンプレートの作成

テンプレートを作成するには、次の手順を実行します。

1. データ収集 > テンプレートに移動します。
2. テンプレートの作成をクリックします。
3. テンプレートの属性を編集します。

テンプレートタブには、テンプレートの一般的な属性が含まれています。



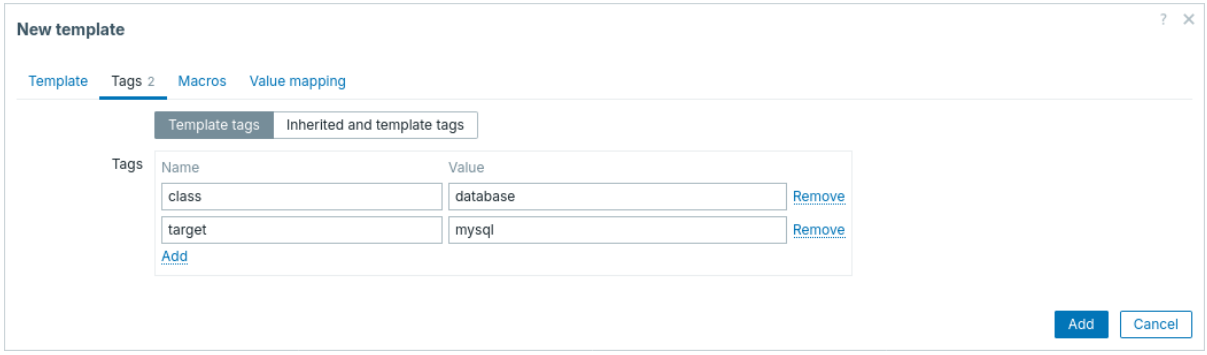
必須入力フィールドには、赤いアスタリスクが付いています。

テンプレートの属性:

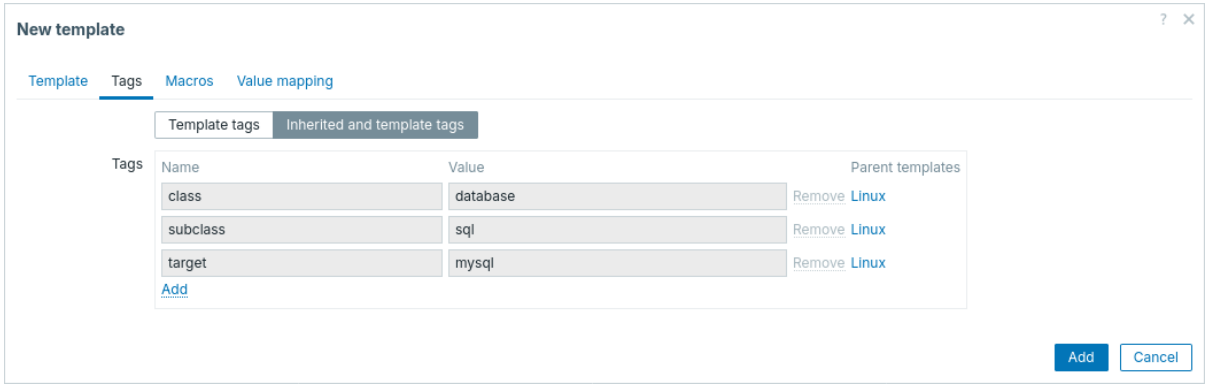
パラメータ	説明
テンプレート名	一意のテンプレート名。 英数字、スペース、ドット、ハイフン、アンダースコアを使用できます。 先頭および末尾のスペースは使用できません。
表示名	この名前を設定すると、リストやマップなどで表示される名前になります。
テンプレート	1 つ以上のテンプレートをこのテンプレートにリンクします。リンクされたテンプレートから、すべてのエンティティ（アイテム、トリガーなど）が継承されます。 新しいテンプレートをリンクするには、テンプレートフィールドに名前を入力します。一致するテンプレートのドロップダウンリストが表示されます。または、選択をクリックして テンプレートポップアップを開くこともできます。このポップアップには、最初はテンプレートが表示されません。リストにテンプレートを表示するには、テンプレートグループフィールドにテンプレートグループ名を入力して候補リストから選択するか、テンプレートグループフィールドの横にある 選択ボタンをクリックして テンプレートグループポップアップを開きます。テンプレートグループを選択すると、そのグループに属するテンプレートが テンプレートポップアップに表示されます。選択したテンプレートは、設定を保存または更新するとリンクされます。 テンプレートのリンクを解除するには、テンプレートブロックの 2 つのオプションのいずれかを使用します。 リンク解除 - テンプレートのリンクを解除しますが、そのエンティティ（アイテム、トリガーなど）は保持します。 リンク解除してクリア - テンプレートのリンクを解除し、そのすべてのエンティティ（アイテム、トリガーなど）を削除します。

パラメータ	説明
テンプレートグループ	テンプレートが属するテンプレートグループ。
説明	テンプレートの説明。
ベンダーとバージョン	テンプレートのベンダーとバージョン。既存のテンプレート（Zabbix が提供する標準テンプレート、インポートされたテンプレート、またはTemplate API を使用して変更されたテンプレート）の設定にこの情報が含まれている場合にのみ表示されます。 Zabbix の Web インターフェースでは変更できません。 標準テンプレートの場合、バージョンは次の形式で表示されます。Zabbix のメジャーバージョン、区切り文字（「-」）、リビジョン番号（テンプレートの新しいバージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされます）。例: 7.0-0、7.0-5、8.0-0、8.0-3。

タグタブでは、テンプレートレベルのタグ を定義できます。このテンプレートにリンクされたホストのすべての障害に、ここで入力した値のタグが付けられます。



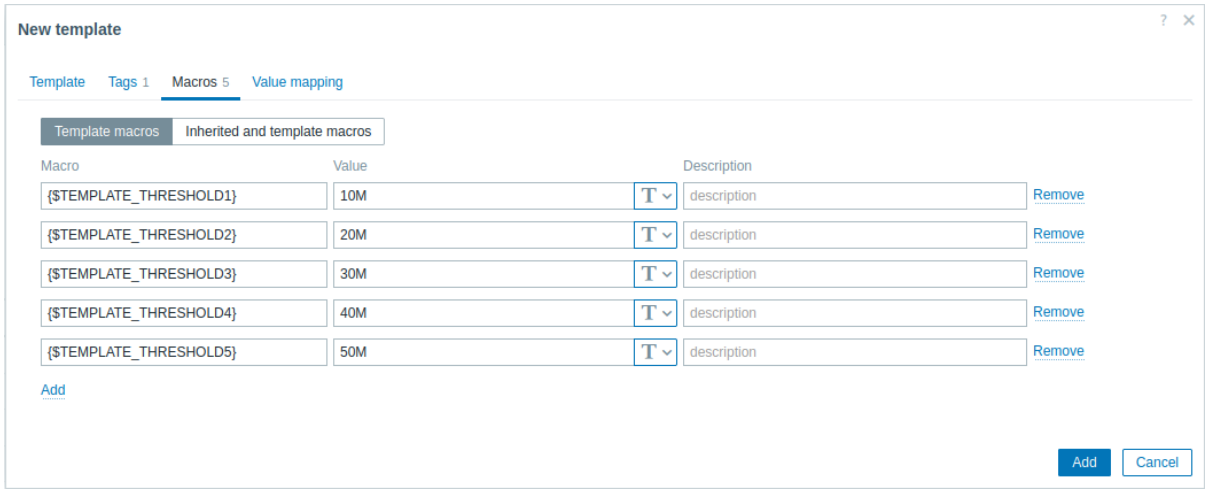
継承されたタグとテンプレートタグオプションを選択すると、テンプレートが継承するリンクされたテンプレートのタグも表示できます。



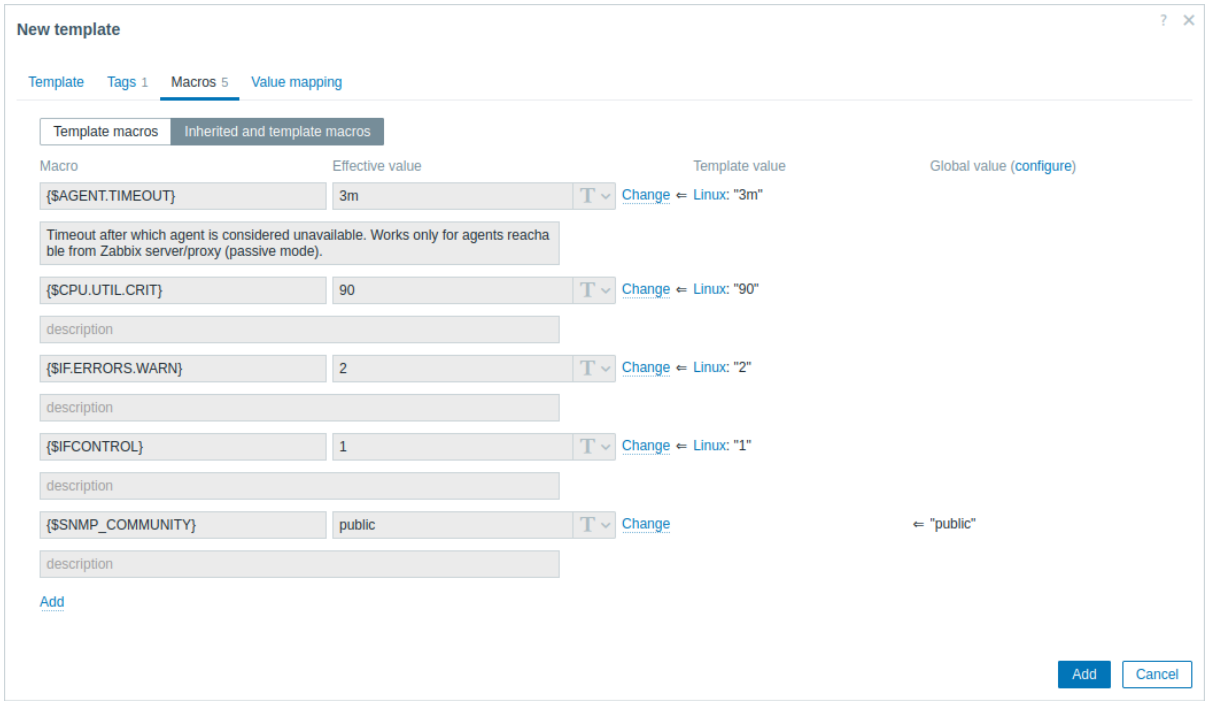
利便性のため、対応するテンプレートへのリンクが提供されます。

ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロをタグで使用できます。

マクロタブでは、テンプレートレベルのユーザーマクロ を名前と値のペアとして定義できます。マクロの値は、プレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できます。説明を追加することもできます。



継承されたマクロとテンプレートマクロオプションを選択すると、テンプレートが継承するリンクされたテンプレートのマクロおよびグローバルマクロと、マクロが解決される値も表示できます。



利便性のため、対応するテンプレートへのリンクと、グローバルマクロ設定へのリンクが提供されます。リンクされたテンプレートのマクロまたはグローバルマクロをテンプレートレベルで編集し、テンプレート上にマクロのコピーを作成することもできます。

値のマッピングタブでは、値のマッピング でアイテムデータを人間が理解しやすい形式で表示するよう設定できます。

ボタン:

Add	テンプレートを追加します。追加したテンプレートがリストに表示されます。
Update	既存のテンプレートのプロパティを更新します。
Clone	現在のテンプレートのプロパティに基づいて、別のテンプレートを作成します。リンクされたテンプレートから継承されたエンティティ（アイテム、トリガーなど）と、現在のテンプレートに直接関連付けられたエンティティの両方が 含まれます。ただし、複製したテンプレートを元のテンプレートと区別できるように、現在のテンプレートのベンダーとバージョンは含まれません。
Delete	テンプレートを削除します。テンプレートのエンティティ（アイテム、トリガーなど）は、リンクされたホストに残ります。
Delete and clear	テンプレートと、そのすべてのエンティティをリンクされたホストから削除します。
Cancel	テンプレートのプロパティの編集をキャンセルします。

アイテム、トリガー、グラフの追加

> **Attention:**
> アイテムは、まずテンプレートに追加する必要があります。トリガーとグラフは、対応するアイテムがなければ追加できません。

既存のアイテムをテンプレートに追加することも、新しいアイテムを作成することもできます。

テンプレートにアイテムを追加するには、次の手順を実行します。

1. データ収集 > ホスト（または テンプレート）に移動します。
2. 対象のホスト/テンプレートの行で アイテムをクリックします。
3. テンプレートに追加したいアイテムのチェックボックスをオンにします。
4. アイテム一覧の下にある コピーをクリックします。
5. アイテムをコピーするテンプレート（またはテンプレートグループ）を選択し、コピーをクリックします。
 選択したすべてのアイテムがテンプレートにコピーされます。

トリガーとグラフの追加も同様の手順で行います（それぞれトリガー一覧、グラフ一覧から操作します）。ただし、必要なアイテムが先に追加されている場合にのみ追加できる点に注意してください。

ダッシュボードの追加

Data collection > *Templates* でテンプレートにダッシュボードを追加するには、次の手順を実行します。

1. テンプレートの行にある *Dashboards* をクリックします。
2. ダッシュボードの設定 のガイドラインに従ってダッシュボードを設定します。

Attention:
テンプレートのダッシュボード（グローバルダッシュボードではなく）でウィジェットを設定する場合、ホスト関連のパラメータは使用できず、一部のパラメータはラベルが異なります。これは、テンプレートのダッシュボードが、そのテンプレートがリンクされているホストのデータのみを表示するためです。たとえば、*Problems* ウィジェットのパラメータ *Host groups*、*Exclude host groups*、*Hosts* は使用できず、*Host availability* ウィジェットのパラメータ *Host groups* は使用できません。また、パラメータ *Show hosts in maintenance* は *Show data in maintenance* に名称が変更されます。などです。テンプレートのダッシュボードウィジェットで使用できるパラメータの詳細については、各ダッシュボードウィジェット の個別のパラメータを参照してください。

Note:
テンプレートのダッシュボードから作成されたホストのダッシュボードへのアクセス方法の詳細については、ホストのダッシュボード セクションを参照してください。

ローレベルディスカバリルールの設定

マニュアルのローレベルディスカバリのセクションを参照してください。

Web シナリオの追加

データ収集 > テンプレートでテンプレートに web シナリオを追加するには、次の手順を実行します。

1. テンプレートの行で *Web* をクリックします。
2. 通常のweb シナリオの設定 方法に従って web シナリオを設定します。

テンプレートグループの設定

概要

テンプレートグループは、テンプレートを論理的にグループ化し、それらに対するユーザー権限を割り当てるために使用されます。

各テンプレートには、少なくとも 1 つのテンプレートグループを割り当てる必要があります。テンプレートは複数のテンプレートグループに属することができ、各テンプレートグループには複数のテンプレートを含めることができます。

Zabbix では、すべての権限はグループに基づいていることに注意してください: user groups、host groups、およびテンプレートグループです。そのため、単一のユーザーが単一のテンプレートにアクセスする必要がある場合でも、そのテンプレートを含むテンプレートグループに対する権限をそのユーザーグループに割り当てるために、ユーザーをユーザーグループに追加することで付与されます。

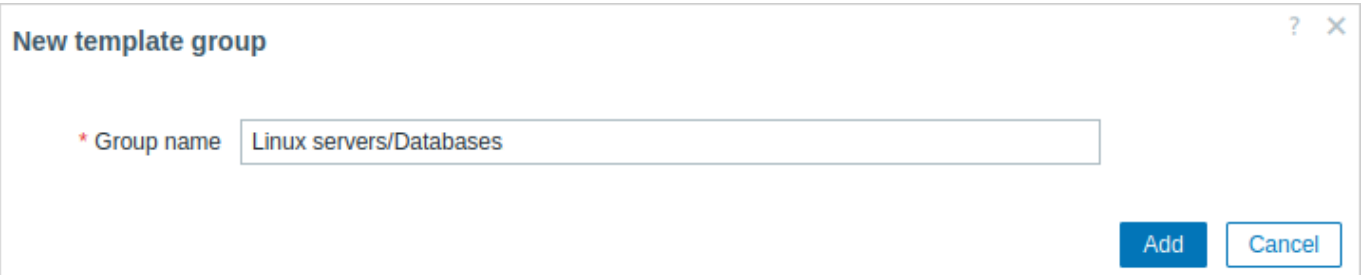
設定

Attention:
テンプレートグループを作成できるのは Super admin ユーザーのみです。

Zabbix Web インターフェースでテンプレートグループを作成する方法は 2 つあります。

方法 **1**:

1. データ収集 > テンプレートグループに移動します。
2. 画面右上の テンプレートグループの作成をクリックします。
3. フォームにグループ名を入力します。



方法 **2**: テンプレートを設定する ときに、テンプレートグループ入力欄に存在しないグループ名を入力します。

テンプレートグループを作成すると、データ収集 > テンプレートグループの一覧でテンプレート名をクリックして、グループ名の編集、グループの複製、またはグループの削除を行えます。

テンプレートグループを削除しても、削除されるのは論理グループのみで、グループ内のテンプレートは削除されません。既存のテンプレートに対して唯一のグループであるテンプレートグループは削除できません。

テンプレートサブグループの作成

テンプレートサブグループ（またはネストされたテンプレートグループ）は、それを含む親テンプレートグループの子です。

サブグループは、グループ名入力フィールドでスラッシュ'/' を使用して親グループとの関係を示すことで作成されます。例:

- `Linux servers/Databases` と入力すると、親グループ `Linux servers` のサブグループ `Linux servers/Databases` が作成されます。
- `Linux servers/Databases/MySQL/Tokyo` と入力すると、ネストされた親グループ `Linux servers`、`Linux servers/Databases`、`Linux servers/Databases/MySQL` 内にそれぞれのサブグループが作成されます。

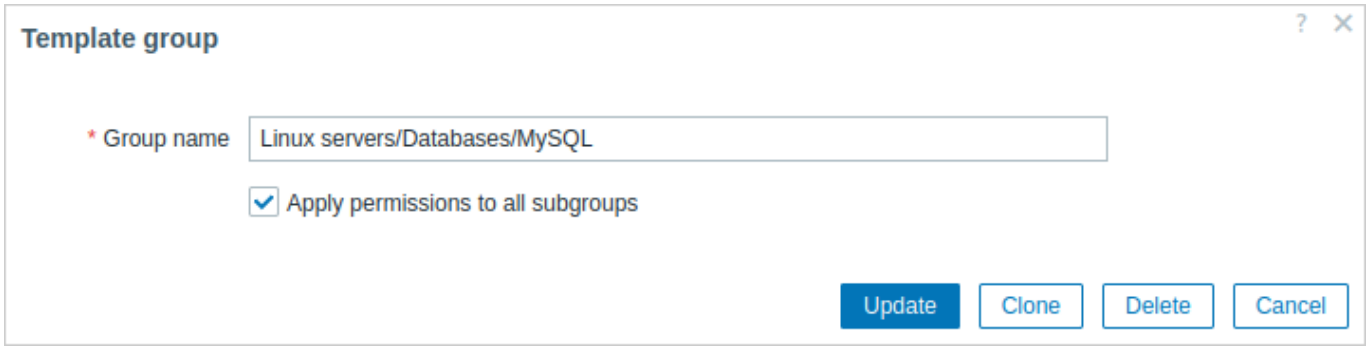
サブグループを作成する際、先頭または末尾にスラッシュを付けたり、連続して複数のスラッシュを使用することはできません。'/' のエスケープはサポートされていません。

サブグループを作成する前に、親テンプレートグループを作成する必要はありません。サブグループ（例: `Linux servers/Databases`）から作成を開始するか、親テンプレートグループ（この例では `Linux servers`）から作成を開始するかを選択できます。サブグループから作成を開始した場合、親テンプレートグループは自動的には作成されません。

テンプレートグループへの権限

- 既存の親テンプレートグループにサブグループを作成する場合（たとえば、`Linux servers` がすでに存在する状態で `Linux servers/Databases` を作成する場合）、サブグループに対するユーザーグループ の権限は親から継承されます。
- 既存のサブグループに親テンプレートグループを作成する場合（たとえば、`Linux servers/Databases` がすでに存在する状態で `Linux servers` を作成する場合）、親に対する権限は設定されません。

任意のテンプレートグループを編集する際には、追加オプション すべてのサブグループに権限を適用も設定できます。



このチェックボックスをオンにして 更新をクリックすると、編集中のテンプレートグループの現在および将来のすべてのサブグループに、同じ権限レベルが適用されます。

そのため、編集中のテンプレートグループのサブグループに対して、ユーザーグループごとに異なる権限 が設定されている場合でも、このチェックボックスをオンにすると、現在および将来のすべてのサブグループに、編集中のグループと同じユーザー権限が付与されます。

このオプションはデータベースには保存されず、既存の権限を上書きする点に注意してください。このオプションによって行われた変更は、手動でのみ元に戻せます。

リンク/リンク解除

概要

リンクはテンプレートがホストに適用されるプロセスであり、リンク解除はホストからテンプレートとの関連付けを削除します。

テンプレートのリンク

テンプレートをホストにリンクするには、次の手順を実行します。

1. データ収集 > ホストに移動します。
2. 対象のホストをクリックします。
3. テンプレートフィールドにテンプレート名の入力を開始します。一致するテンプレートの一覧が表示されるので、スクロールして選択します。
   または、フィールドの横にある 選択をクリックし、ポップアップウィンドウの一覧から 1 つ以上のテンプレートを選択します。
4. ホスト属性フォームで 追加/更新をクリックします。

これで、ホストにテンプレートのすべてのエンティティが追加されます。
これには、アイテム、トリガー、グラフ、ローレベルディスカバリルール、Web シナリオ、およびダッシュボードが含まれます。

Attention:
同じホストに複数のテンプレートをリンクする場合、それらのテンプレートに同じアイテムキーを持つアイテムが含まれていると、リンクに失敗します。
また、トリガーとグラフはアイテムを使用するため、同一のアイテムキーを使用している場合、複数のテンプレートから単一のホストにリンクすることはできません。

テンプレートからエンティティ（アイテム、トリガーなど）が追加されると、次のようになります。

- ホストに以前から存在していた同一のエンティティは、テンプレートのエンティティとして更新され、エンティティに対してホストレベルで行われていたカスタマイズはすべて失われます。
- テンプレートのエンティティが追加されます。
- テンプレートのリンク前からホストにのみ存在していた、直接リンクされたエンティティは変更されません。

一覧では、テンプレートのすべてのエンティティの先頭にテンプレート名が付くようになります。これは、それらが特定のテンプレートに属していることを示します。
テンプレート名自体（グレーで表示）はリンクになっており、テンプレートレベルでそれらのエンティティの一覧にアクセスできます。

Note:
外部チェック、HTTP エージェントチェック、シンプルチェック、SSH チェック、Telnet チェック など、一部のアイテムではホストインターフェースは任意です。
テンプレートのリンク時にホストにインターフェースが定義されていない場合、これらのアイテムはインターフェースなしで追加されます。
後からホストインターフェースを追加しても、既存のアイテムに自動的に割り当てられることはありません。
新しく追加したホストインターフェースをすべてのテンプレートアイテムに一度に割り当てるには、テンプレートをホストからリンク解除してから、再度リンクします。
アイテムの履歴を保持するには、リンク解除オプションを使用し、リンク解除して消去は使用しないでください。

テンプレート名が先頭に付いていないエンティティは、テンプレートによって追加されたものではなく、ホストに以前から存在していたことを意味します。

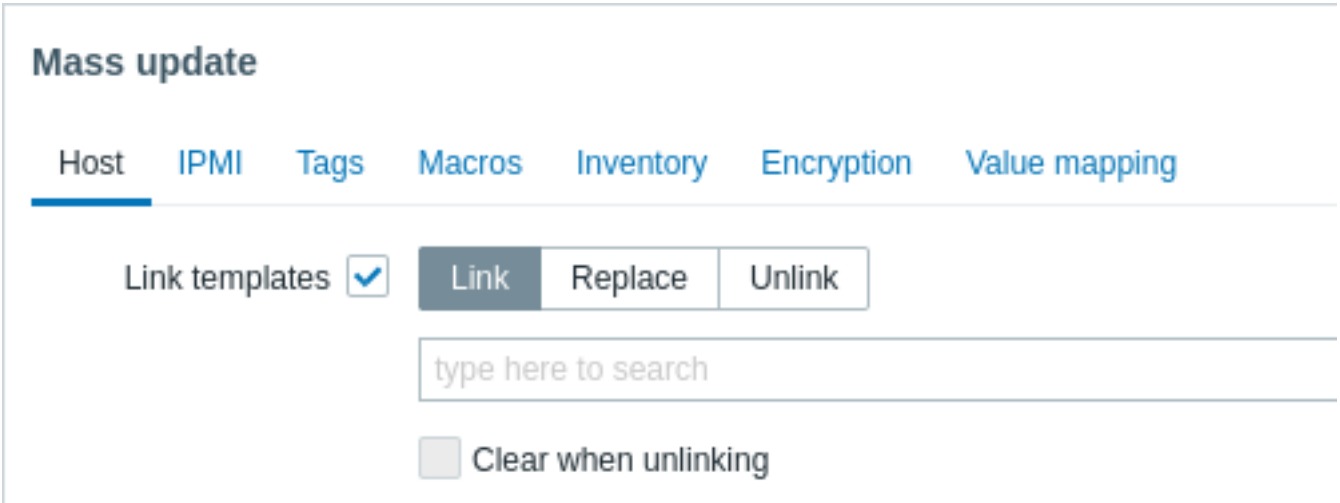
エンティティの一意性の基準

テンプレートからエンティティ（アイテム、トリガーなど）を追加する場合、ホスト上にすでに存在し更新が必要なエンティティと、異なるエンティティを知ることが重要です。同一性/差異を判断するための一意性の基準は以下の通りです。

- アイテムの場合 - アイテムキー
- トリガーの場合 - トリガー名と式
- カスタムグラフの場合 - グラフ名とそのアイテム

複数のホストへのテンプレートのリンク

多数のホストのテンプレートのリンクを更新するには、データ収集 > ホストで、チェックボックスをオンにしていくつかのホストを選択し、一覧の下にある 一括更新 をクリックしてから、テンプレートのリンクを選択します。



追加のテンプレートをリンクするには、候補のテンプレートを含むドロップダウンが表示されるまで、オートコンプリートフィールドにテンプレート名を入力します。下にスクロールして、リンクするテンプレートを選択します。

置換オプションを使用すると、新しいテンプレートをリンクすると同時に、以前にホストへリンクされていたテンプレートのリンクを解除できます。リンク解除オプションを使用すると、リンクを解除するテンプレートを指定できます。リンク解除時にクリアオプションを使用すると、以前にリンクされていたテンプレートのリンクを解除するだけでなく、それらのテンプレートから継承されたすべてのエンティティ（アイテム、トリガーなど）も削除できます。

Note:
Zabbix には、あらかじめ定義されたテンプレートが多数用意されています。これらを参考として使用できますが、アイテムが多すぎたり、データを頻繁にポーリングしたりする場合があるため、本番環境で変更せずに使用する際は注意してください。使用する場合は、実際の要件に合うように調整してください。

リンクされたエンティティの編集

テンプレートからリンクされたアイテムやトリガーを編集しようとすると、多くの主要なオプションが編集できないことに気付くかもしれません。これは、テンプレートの考え方が、テンプレートレベルで一括して編集することにあるため、理にかなっています。ただし、個々のホストでアイテムを有効/無効にしたり、更新間隔、履歴の長さ、その他のパラメータを設定したりすることは可能です。

エンティティを完全に編集したい場合は、テンプレートレベルで編集する必要があります（フォーム名にテンプレートレベルのショートカットが表示されます）。これらの変更は、このテンプレートがリンクされているすべてのホストに影響することに注意してください。

Attention:
テンプレートレベルで実装されたエンティティのカスタマイズは、ホストレベルでのエンティティの以前のカスタマイズを上書きします。

テンプレートのリンク解除

ホストからテンプレートのリンクを解除するには、次の手順を実行します。

1. データ収集 > ホストに移動します。
2. 対象のホストをクリックし、テンプレートフィールドを見つけます。
3. テンプレートの横にある リンク解除または リンク解除して削除をクリックして、テンプレートのリンクを解除します。
4. ホスト属性フォームで 更新をクリックします。

リンク解除オプションを選択すると、テンプレートとの関連付けのみが削除され、そのテンプレートのすべてのエンティティはホストに残ります。
これにはアイテム、トリガー、グラフ、低レベルディスカバリルール、Web シナリオが含まれますが、ダッシュボードは含まれません。
リンクされたテンプレートから継承された値マップとタグも削除されることに注意してください。

リンク解除して削除オプションを選択すると、テンプレートとの関連付けとそのすべてのエンティティ（アイテム、トリガーなど）の両方が削除されます。

ネスト化

概要

ネストは、1 つのテンプレートが 1 つ以上の他のテンプレートを包含する方法です。

さまざまなサービス、アプリケーションなどの個別のテンプレートでエンティティを分離することは理にかなっていますが、最終的に大量のテンプレートが作成され、またそのすべてを大量の数のホストにリンクする必要になる可能性があります。単純化するために、複数のテンプレートを 1 つのテンプレートにリンクすることができます。

ネストの利点は、ホストにリンクするテンプレートが 1 つだけで済み、ホストはその 1 つのテンプレートにリンクされているテンプレートからすべてのエンティティを自動的に継承することです。たとえば、テンプレート *T1* と *T2* を *T3* にリンクすると、*T1* と *T2* のエンティティが *T3* に追加されますが、その逆はありません。*T1* を *T2* と *T3* にリンクすると、*T1* のエンティティが *T2* と *T3* に追加されます。

ネストされたテンプレートの設定

テンプレートをリンクするには、既存のテンプレートを使用するか、新しいテンプレートを作成してから、次の手順を実行します。

1. テンプレート設定フォームを開きます。
2. *Templates* フィールドを見つけます。
3. *Select* をクリックして、*Templates* ポップアップウィンドウを開きます。
4. ポップアップウィンドウで必要なテンプレートを選択し、*Select* をクリックしてテンプレートを一覧に追加します。
5. テンプレート設定フォームで *Add* または *Update* をクリックします。

これにより、設定したテンプレートのすべてのエンティティと、リンクされたテンプレートのすべてのエンティティが、テンプレート設定に表示されるようになります。
これにはアイテム、トリガー、グラフ、低レベルディスカバリルール、Web シナリオが含まれますが、ダッシュボードは含まれません。
ただし、リンクされたテンプレートのダッシュボードは、ホストに継承されます。

リンクされたテンプレートのリンクを解除するには、テンプレート設定フォームで *Unlink* または *Unlink and clear* をクリックし、その後 *Update* をクリックします。

Unlink オプションは、リンクされたテンプレートとの関連付けのみを削除し、そのエンティティ（アイテム、トリガーなど）は削除しません。

Unlink and clear オプションは、リンクされたテンプレートとの関連付けに加えて、そのすべてのエンティティ（アイテム、トリガーなど）も削除します。

一括更新

概要

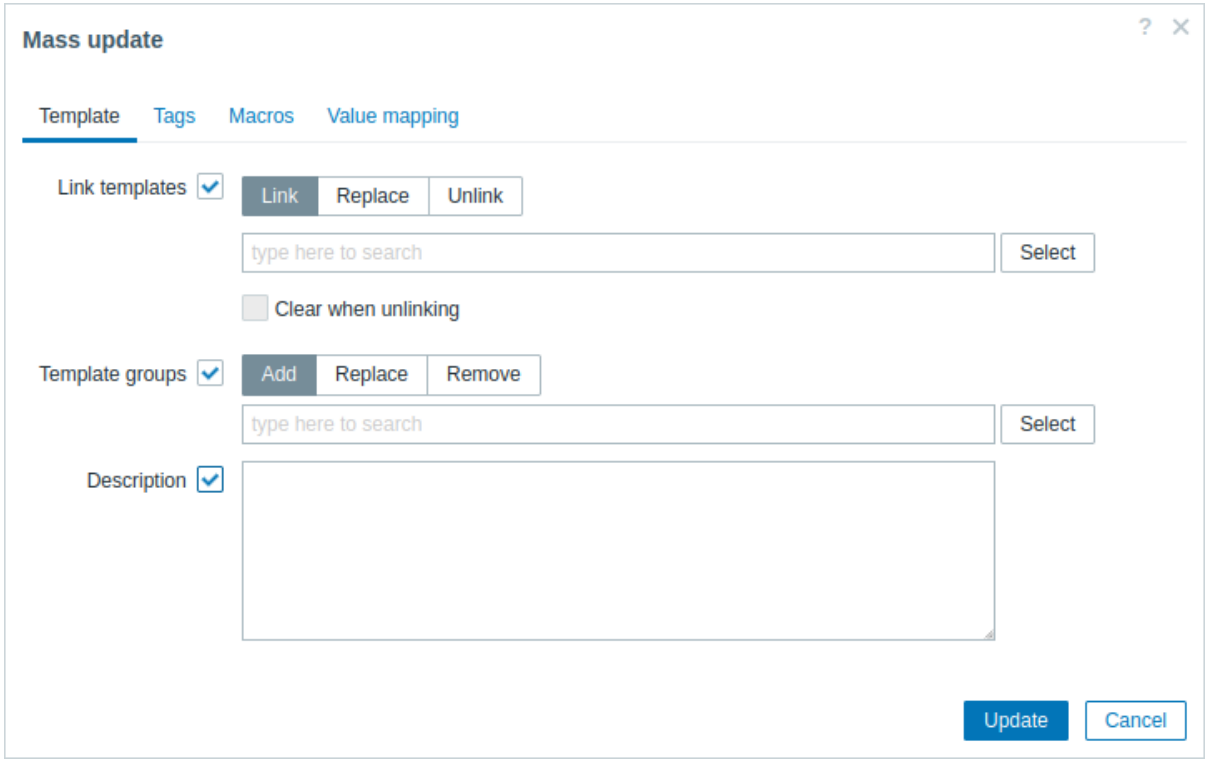
いくつかのテンプレートの属性を一度に変更したい場合があります。個々のテンプレートを開いて編集する代わりに、一括更新機能を使用できます。

一括更新の使用

一部のテンプレートを一括更新するには、次の手順を実行します。

1. 更新したいテンプレートの前にあるチェックボックスを、テンプレート一覧で選択します。
2. 一覧の下にある 一括更新をクリックします。
3. 必要な属性があるタブ（*Template*、*Tags*、*Macros*、または *Value mapping*）に移動します。
4. 更新する属性のチェックボックスを選択し、新しい値を入力します。

テンプレートタブには、テンプレートの一括更新に関する一般的なオプションが含まれています。



テンプレートのリンクの更新で該当するボタンを選択すると、次のオプションを使用できます。

- リンク - 追加でリンクするテンプレートを指定します。
- 置換 - リンクするテンプレートを指定します。同時に、以前にリンクされていたテンプレートのリンクを解除します。
- リンク解除 - リンクを解除するテンプレートを指定します。

リンクまたはリンク解除するテンプレートを指定するには、オートコンプリートフィールドにテンプレート名を入力し、該当するテンプレートがドロップダウンに表示されるまで続けます。下にスクロールして、必要なテンプレートを選択します。

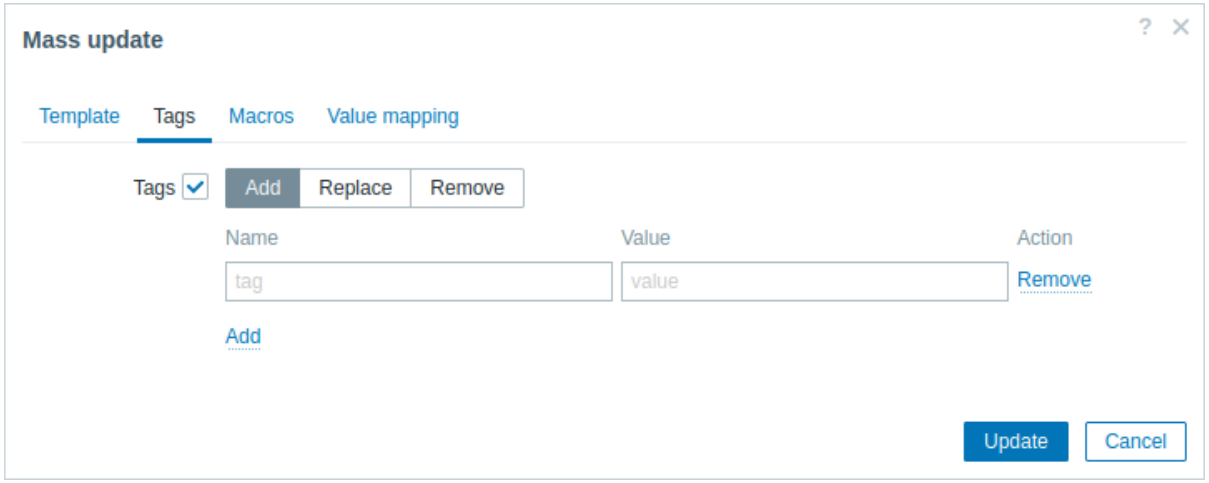
リンク解除時にクリアオプションを使用すると、以前にリンクされたテンプレートのリンクを解除するだけでなく、それらのテンプレートから継承されたすべてのエンティティ（アイテム、トリガー、グラフなど）も削除できます。

テンプレートグループの更新で該当するボタンを選択すると、次のオプションを使用できます。

- 追加 - 既存のテンプレートグループから追加するテンプレートグループを指定するか、テンプレート用にまったく新しいテンプレートグループを入力します。
- 置換 - 既存のテンプレートグループからテンプレートを削除し、このフィールドで指定したテンプレートグループ（既存または新規）に置き換えます。
- 削除 - テンプレートから特定のテンプレートグループを削除します。

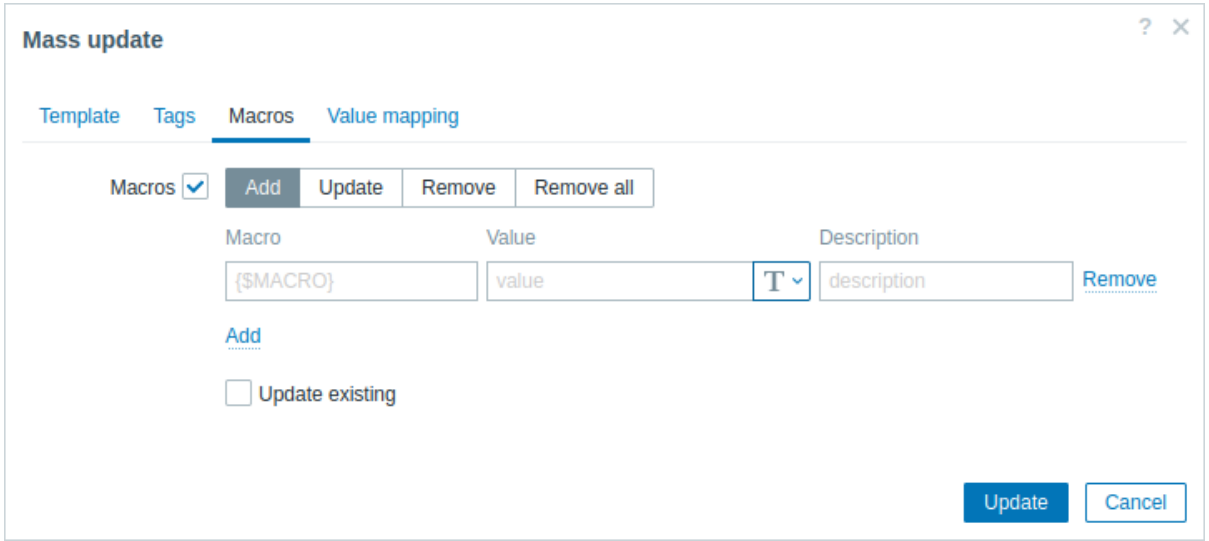
これらのフィールドはオートコンプリートに対応しています。フィールドへの入力を開始すると、一致するテンプレートグループのドロップダウンが表示されます。新しいテンプレートグループの場合もドロップダウンに表示され、文字列の後に *(新規)* と示されます。下にスクロールして選択します。

タグタブでは、テンプレートレベルのタグを一括更新できます。



タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。同じ名前でも異なる値のタグは「重複」とみなされず、同じテンプレートに追加できます。

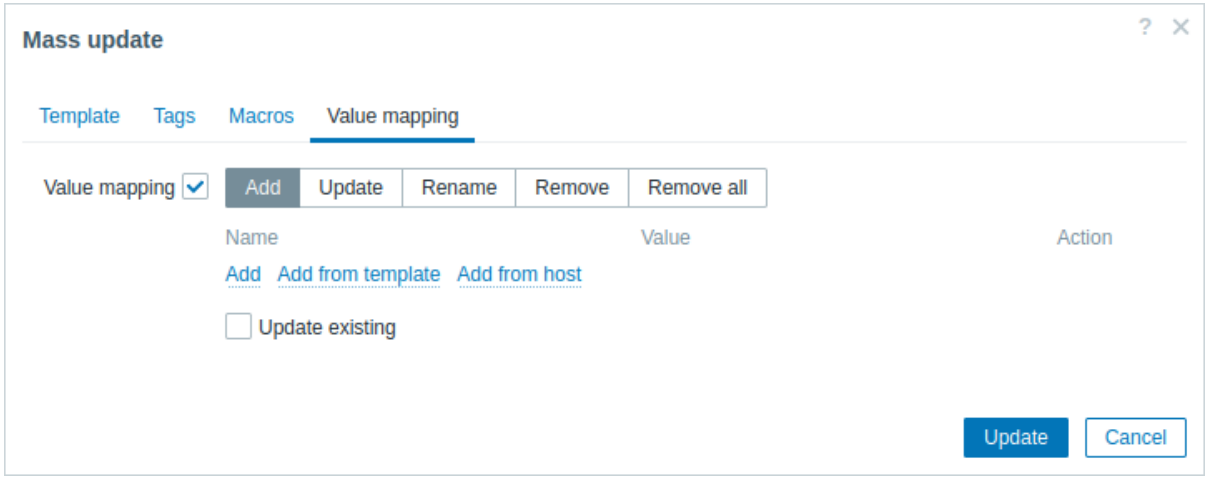
マクロタブでは、テンプレートレベルのマクロを一括更新できます。



マクロの更新で該当するボタンを選択すると、次のオプションを使用できます。

- 追加 - テンプレートに追加するユーザーマクロを指定します。既存のマクロを更新チェックボックスをオンにすると、指定したマクロ名の値、タイプ、説明が更新されます。オフの場合、同じ名前のマクロがテンプレートにすでに存在していても、更新されません。
- 更新 - このリストで指定したマクロの値、タイプ、説明を置き換えます。不足しているマクロを追加チェックボックスをオンにすると、テンプレートにまだ存在しないマクロが新しいマクロとして追加されます。オフの場合、テンプレートにすでに存在するマクロのみが更新されます。
- 削除 - 指定したマクロをテンプレートから削除します。選択したマクロを除外チェックボックスをオンにすると、リストで指定したマクロ以外のすべてのマクロが削除されます。オフの場合、リストで指定したマクロのみが削除されます。
- すべて削除 - テンプレートからすべてのユーザーマクロを削除します。すべてのマクロを削除することを確認しましたチェックボックスをオンにしていない場合、すべてのマクロを削除するか確認する新しいポップアップウィンドウが開きます。

値のマッピングタブでは、値のマッピングを一括更新できます。



値のマッピングを更新するために、次のオプションを持つボタンを使用できます。

- 追加 - 値のマッピングをテンプレートに追加します。既存の値のマッピングを更新を選択すると、この名前の値のマッピングのすべてのプロパティが更新されます。選択しない場合、この名前の値のマッピングがすでに存在していても更新されません。
- 更新 - 既存の値のマッピングを更新します。不足している値のマッピングを追加を選択すると、テンプレートにまだ存在しない値のマッピングが新しい値のマッピングとして追加されます。選択しない場合、テンプレートにすでに存在する値のマッピングのみが更新されます。
- 名前の変更 - 既存の値のマッピングに新しい名前を付けます。
- 削除 - 指定した値のマッピングをテンプレートから削除します。選択したもの以外を選択すると、指定した値のマッピングを除くすべての値のマッピングが削除されます。
- すべて削除 - すべての値のマッピングをテンプレートから削除します。すべての値のマッピングを削除することを確認しますチェックボックスを選択していない場合、削除の確認を求める新しいポップアップウィンドウが開きます。

値のマッピングの追加または更新操作では、テンプレートから追加およびホストから追加オプションを使用できます。これらのオプションを使用すると、それぞれテンプレートまたはホストから値のマッピングを選択できます。

必要な変更をすべて終えたら、*Update* をクリックします。属性は、選択したすべてのテンプレートに対してそれぞれ更新されます。

ホストとホストグループ

「ホスト」とは？

Zabbix において、「ホスト」とは、物理または仮想デバイス、アプリケーション、サービス、または監視対象パラメータの論理的に関連するコレクションを指します。

ホストの作成は、Zabbix での最初の監視タスクの 1 つです。たとえば、サーバー「x」のいくつかのパラメータを監視したい場合、まず「Server X」と呼ばれるホストを作成し、その後で監視アイテムを追加することができます。

ホストはホストグループに編成されます。

次の手順に進みます:

- ホストウィザード - 新規または既存のホストを作成・設定するためのガイド付きステップバイステップインターフェース
- ホストの設定 - ホストの作成と設定のための従来のアプローチ

ホストウィザード

概要

Host Wizard は、Zabbix で監視対象（デバイス、アプリケーション、サービスなど）を設定するための、案内付きのステップ形式のインターフェースです。次の手順に沿って進みます。

- テンプレートの選択
- ホストの作成または選択
- Zabbix エージェントまたは agent 2 のインストール
- ホストインターフェースの追加
- 監視対象または Zabbix ホストへの追加設定の適用（テンプレートで必要な場合）

Zabbix Web インターフェースで Host Wizard にアクセスするには、次の手順を実行します。

1. 次へ移動します: *Data collection* > *Hosts*。

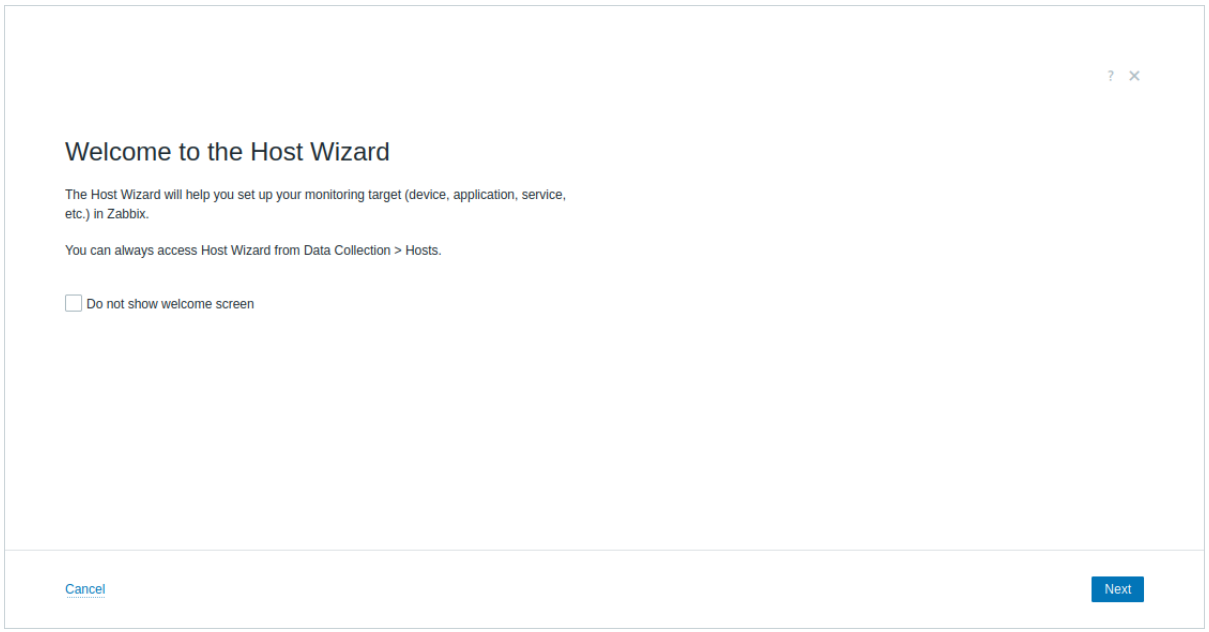
2. 画面右上の *Host Wizard* をクリックします。
3. ウィザードの指示に従います。

また、既存のホストを編集する場合は、各種 Web インターフェースセクションでその名前（または横にある三点アイコン）をクリックして、ホストメニュー から Host Wizard を開くこともできます。既存のホストを編集する場合、選択したテンプレートに必要な設定がすでに含まれていると、一部の手順はスキップされる点に注意してください。

別の方法として、ホストの作成と設定 の従来の手順を使用することもできます。

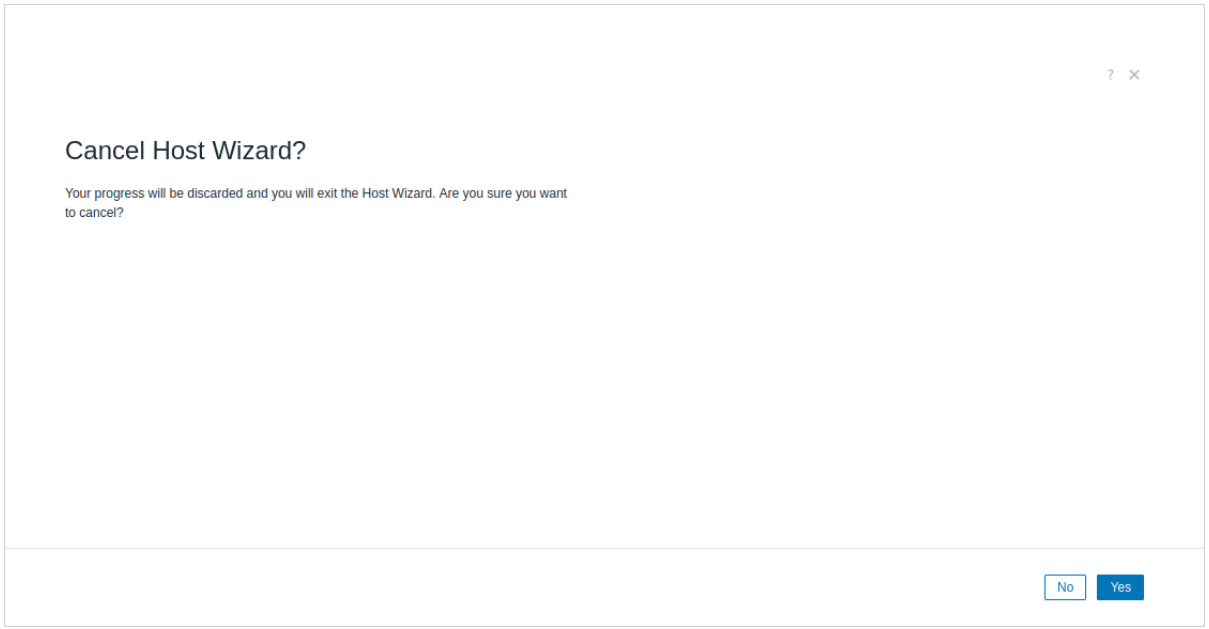
ホストウィザードへようこそ

ホストウィザードを開始すると、ようこそ画面が表示されます。



今後この画面を表示しないようにするには、ようこそ画面を表示しないチェックボックスをオンにして、次へをクリックします。

次のホストウィザードの手順で設定を開始したあと、閉じるアイコンをクリックするか Esc キーを押してウィザードを閉じようとすると、確認画面が表示されます。



はいをクリックすると、進行中の内容を保存せずにホストウィザードを終了します。いいえ、Esc、または閉じるアイコンをクリックすると、最後の手順に戻ります。

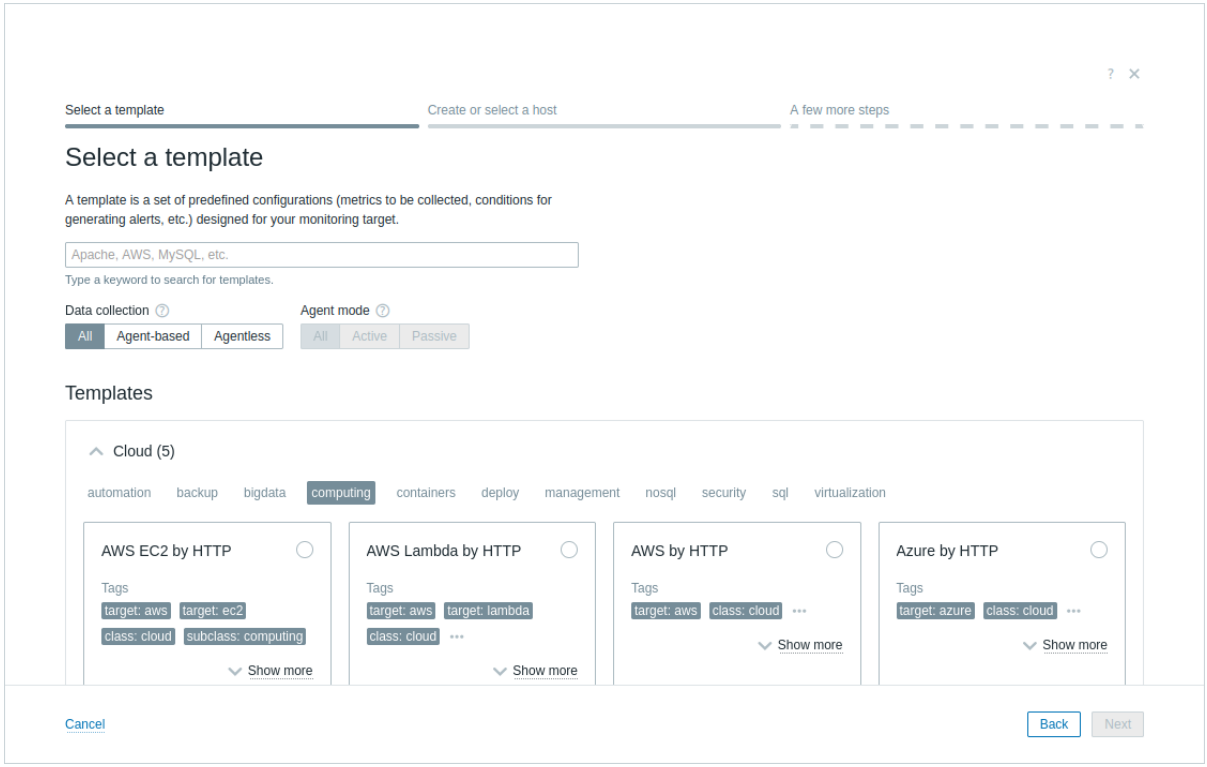
テンプレートを選択する

監視対象を設定する最初の手順は、監視対象向けに設計された事前定義済みの設定（収集するメトリクス、アラートを生成する条件など）のセットであるテンプレートを選択することです。

Zabbix のすぐに使用できるテンプレートには、オペレーティングシステム、アプリケーション、データベース、ネットワークデバイス、サービスなど、さまざまな事前定義済みの監視設定が用意されています。

この手順では、次の操作を実行できます。

- クラス（Cloud、Database、Network など）およびサブクラス（automation、backup など）別にテンプレートを参照できます。これらはテンプレートのタグに基づいています。
- テンプレート名、またはテンプレートのタグ名と値に含まれるキーワードでテンプレートを検索できます。
- データ収集方法（エージェントベースまたはエージェントレス）でテンプレートをフィルターできます。エージェントベースのテンプレートには、少なくとも 1 つのZabbix エージェントアイテムが含まれます。
- エージェントモード（アクティブまたはパッシブ）でテンプレートをフィルターできます。アクティブテンプレートには少なくとも 1 つの Zabbix エージェント（アクティブ）アイテムが含まれ、パッシブテンプレートには少なくとも 1 つのパッシブアイテムが含まれます。詳細については、パッシブチェックとアクティブチェックを参照してください。
- ホストウィザードを使用して既存のホストを設定する場合は、そのホストにすでにリンクされているテンプレートでフィルターすることもできます。



Attention:
テンプレートを選択すると、次のメッセージが表示される場合があります。一部のテンプレート（*n*）はホストウィザードと互換性がありません。更新方法を参照してください。カスタムテンプレートはサポートされていません。これは、一部のすぐに使用できるテンプレートがまだホストウィザードと互換性がなく、アップグレードが必要であることを示しています。テンプレートのアップグレードの手順を参照してください。

選択したテンプレートによって、ホストウィザードの次の手順が決まります。たとえば、データ収集に Zabbix エージェントを使用する *MySQL by Zabbix agent* テンプレートを選択した場合、ウィザードではエージェントのインストール手順が案内されます。

ホストを作成または選択する

テンプレートは、監視対象を表す Zabbix 内のエンティティであるホストにリンクする必要があります。リンクすると、ホストは監視に必要なすべてのテンプレートエンティティ（収集するメトリクスであるアイテムや、アラートを生成する条件であるトリガーなど）を受け取ります。

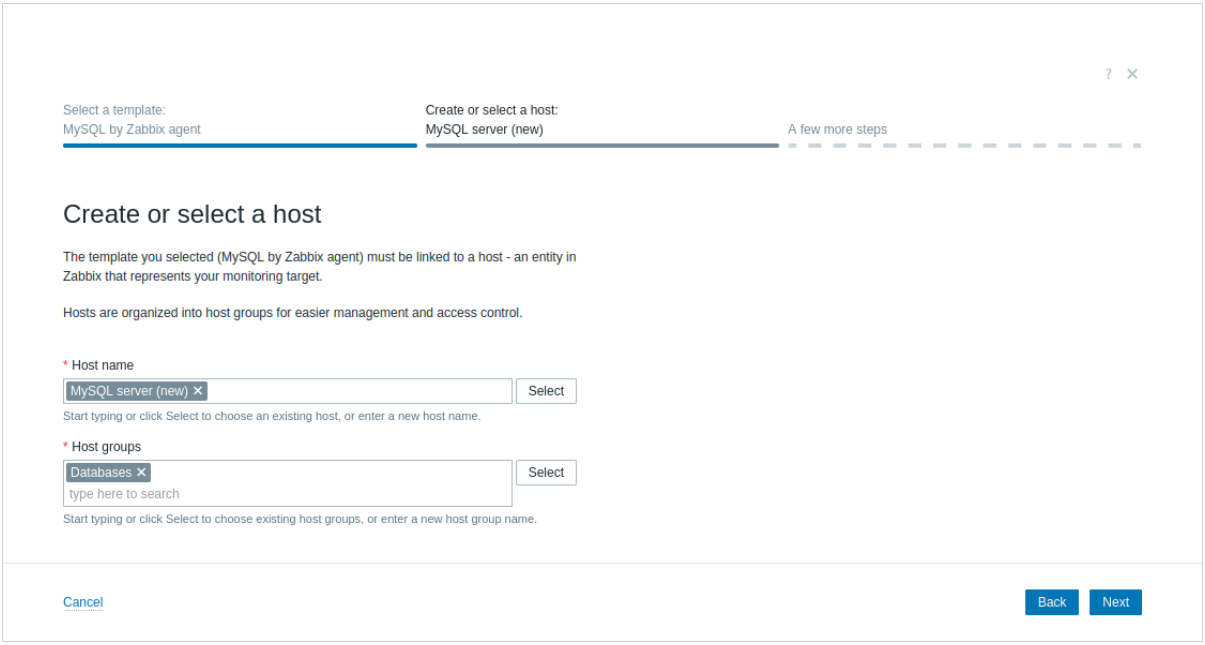
各ホストは、少なくとも 1 つのホストグループにも所属する必要があります。ホストグループは、ホストを整理したり、ホストに対するユーザー権限を割り当てたりするために使用されます。

このステップでは、次の操作を実行できます。

- 両方の名前を入力して、新しいホストとホストグループを作成する。
- 新しいホストの名前を入力し、既存のホストグループを選択して、新しいホストをそのグループに割り当てる。
- 既存のホストを選択する。現在所属しているグループから削除せずに、このホストを追加のホストグループに割り当てることもできます。

Note:
既存のホストを選択しても、後のステップで明示的に記載されていない限り、現在の設定が削除または上書きされることはありません。検出されたホストは選択できません。

たとえば、監視対象とするローカルにインストールされた MySQL サーバーを表すために、*MySQL server* という名前の新しいホストを作成できます。以前に選択した *MySQL by Zabbix agent* テンプレートが、このホストにリンクされます。さらに、既存の *Databases* ホストグループ（または *MySQL servers* などの新しいホストグループ）を選択して、このホストを他のデータベース関連のホストとまとめ、後で権限管理を簡略化することもできます。



Zabbix エージェントのインストール

エージェントベースのテンプレート（*MySQL by Zabbix agent* など）では、監視対象にデプロイしてローカルのリソースやアプリケーションをアクティブに監視するプロセスである、Zabbix エージェント またはエージェント 2 のインストールが必要です（テンプレートで必要な場合）。

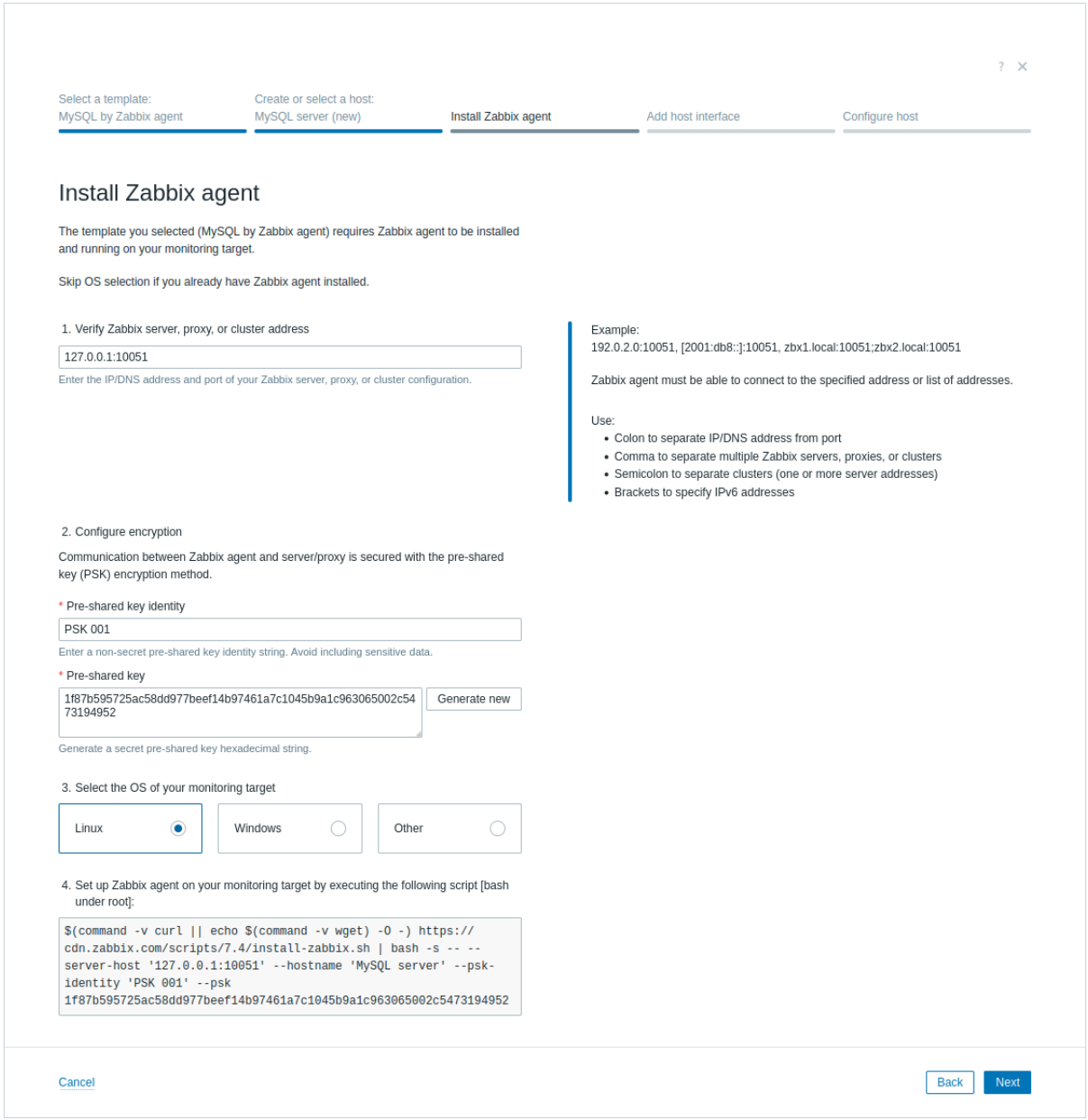
このステップでは、次の操作を行います。

- Zabbix サーバー、プロキシ、またはクラスターのアドレス（例: 192.0.2.1:10051）を確認します。この値は、Zabbix エージェントのServer およびServerActive パラメータの設定に使用されます。
- 新しく、一意で、機密情報を含まない事前共有キーの ID（例: *PSK 001* または *mysql-agent-psk1*）を入力し、事前共有キーを生成します。これらの値は、Zabbix ホスト（例: *MySQL server*）と Zabbix エージェント間の事前共有キー（PSK）暗号化の設定に使用されます。

Attention:
既存のホストを編集する場合、PSK の設定によって、ホスト上の既存の暗号化設定がすべて上書きされます。

- 監視対象をホストするマシンのオペレーティングシステムを選択します。
- 提供されたスクリプトをそのシステム上で実行するか、インストール手順に従って、そのシステムに Zabbix エージェントをインストールします。

たとえば、*Pre-shared key identity* に *PSK 001* を設定し、新しい *Pre-shared key* を生成できます。次に、オペレーティングシステムとして *Linux* を選択し、提供されたスクリプトをそのシステム上で実行します。Zabbix エージェントのインストール後、ホストウィザードに戻ります。

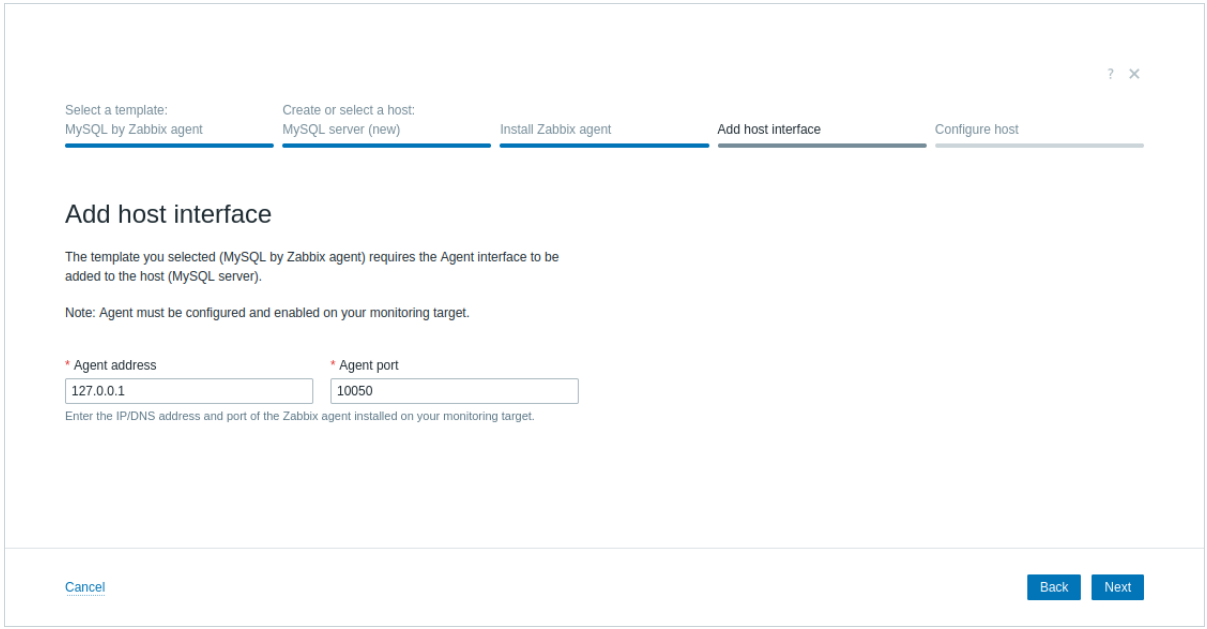


ホストインターフェースの追加

ホストインターフェースは、Zabbix サーバーがネットワーク経由で監視対象に接続する方法を定義します。これは、選択したテンプレートで必要なデータ収集方法に応じて、IP アドレス、DNS 名、ポート、インターフェース種別（Agent、SNMP、JMX、または IPMI）などの接続パラメータを定義します。

この手順では、選択したテンプレートに必要なホストインターフェースの詳細を入力します。

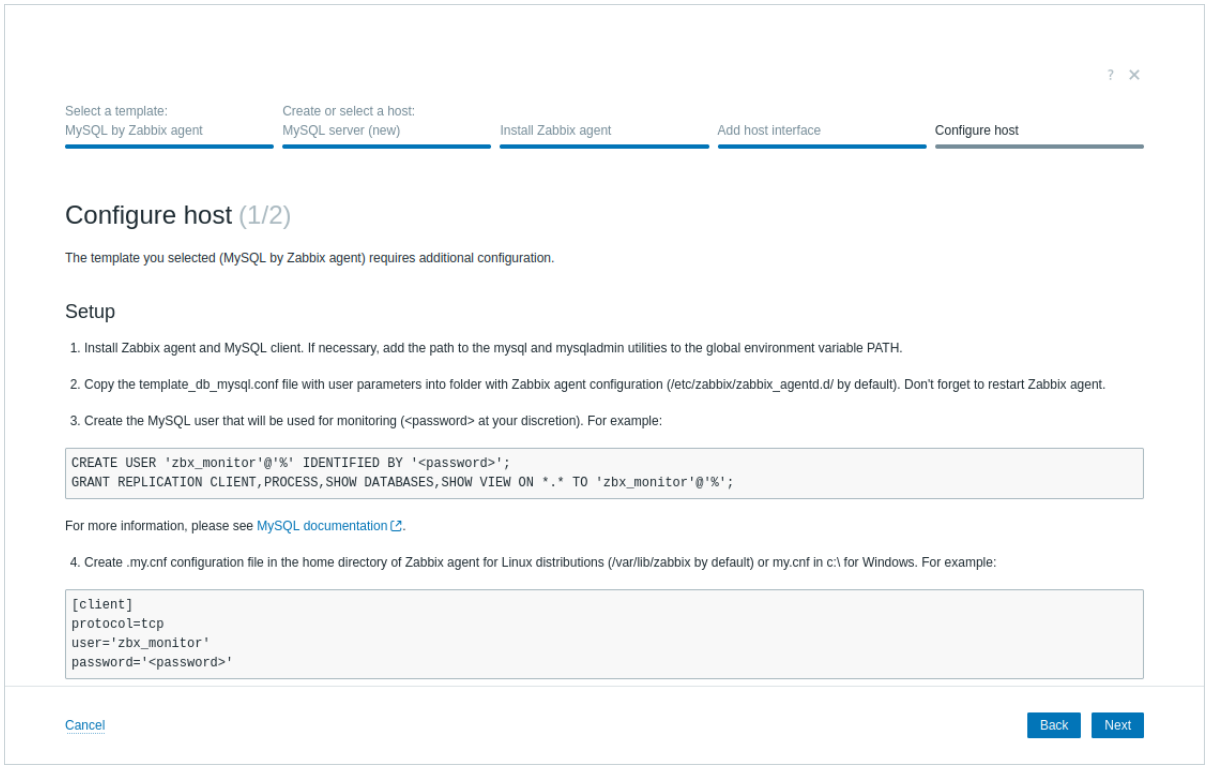
たとえば、Zabbix サーバー、Zabbix エージェント、および MySQL サーバーがすべて同じマシン上でローカルに実行されている場合は、デフォルトのエージェントアドレス（127.0.0.1）とポート（10050）を使用できます。



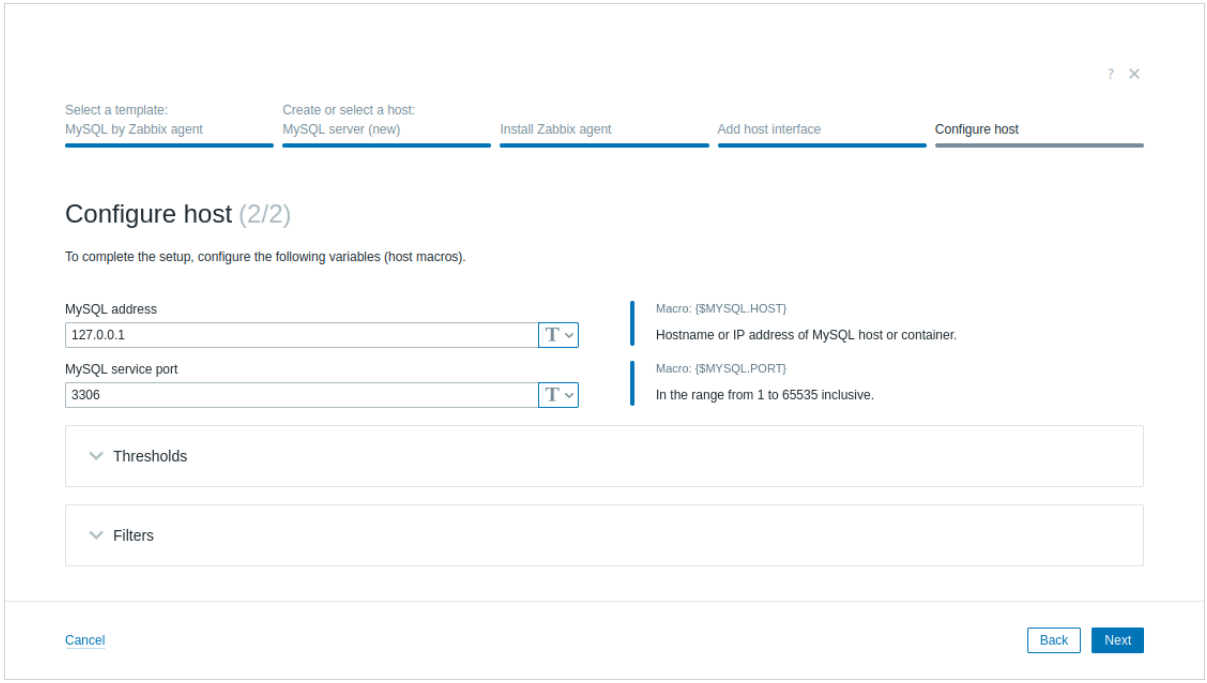
ホストの設定

一部のテンプレートでは、ホストを作成する前に追加の設定が必要です。これには次のような設定が含まれます。

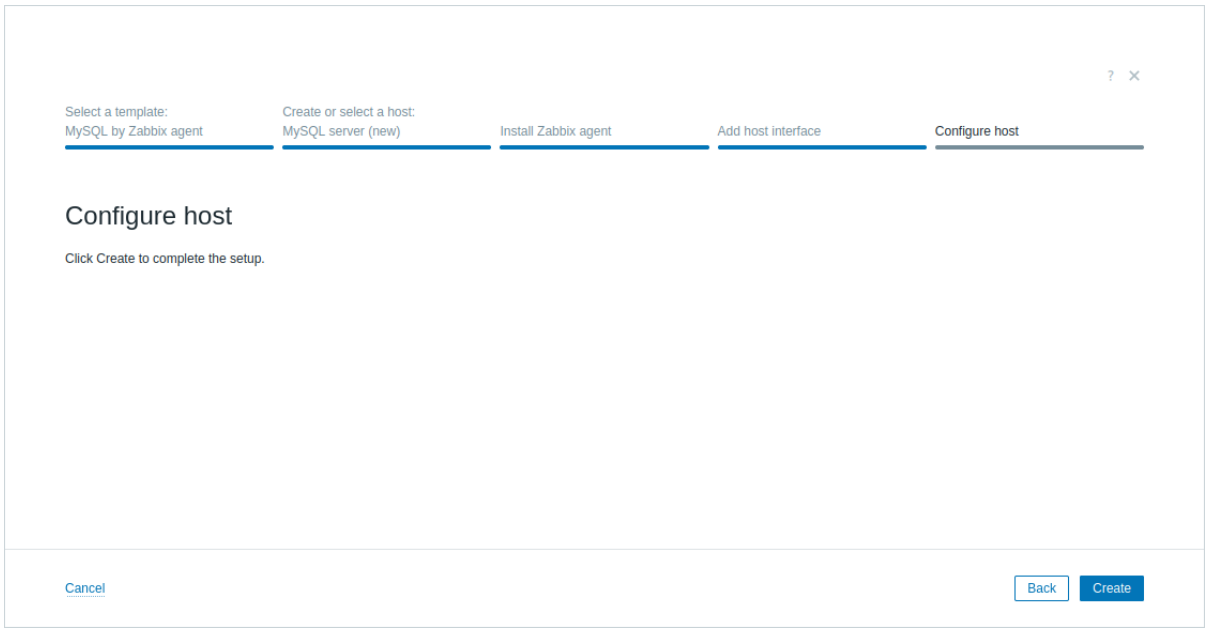
- 監視対象の手動設定（特定のサービスの有効化、サービスユーザーの作成、権限の付与など）：



- ホストマクロの定義 - アイテムの動作、接続設定、認証情報、その他のパラメータを制御する変数:



必要な設定手順をすべて完了したら、作成をクリックしてホストを Zabbix に追加します（既存のホストの設定を更新する場合は更新をクリックします）：



設定完了

この時点で、Zabbix はすでにターゲットの監視を開始しています。

新しいホストの場合は、完了をクリックして最新データ セクションに移動し、ホストの最新データを表示します。既存のホストの場合は、完了をクリックしてホストウィザードを閉じ、ウィザードを開いた画面に戻ります。

ホストの設定

概要

Zabbix Web インターフェースでホストを設定するには、次の手順を実行します。

1. データ収集 > ホストまたは 監視 > ホストに移動します。
2. 画面右上の ホストを作成をクリックします（既存のホストを編集する場合は、ホスト名をクリックします）。
3. フォームにホストのパラメータを入力します。

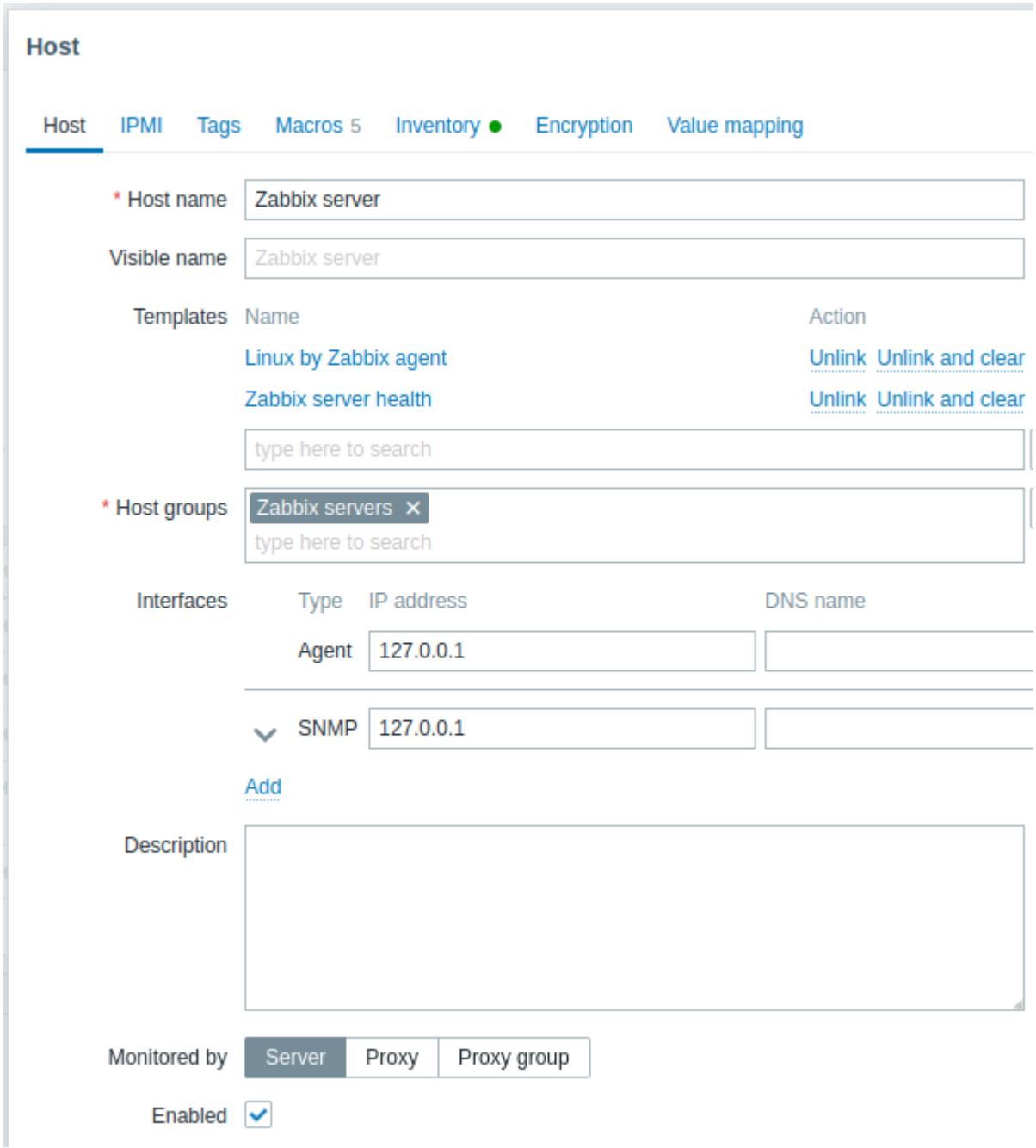
既存のホストの設定フォームにある 複製ボタンを使用して、新しいホストを作成することもできます。
このホストには、リンクされたテンプレート、それらのテンプレートから継承されたエンティティ（アイテム、トリガーなど）、および既存のホストに直接追加されたエンティティを含め、既存のホストのすべてのプロパティが引き継がれます。

ホストを複製すると、テンプレート上に元々あるすべてのテンプレートエンティティがそのまま保持される点に注意してください。既存のホストレベルでそれらのエンティティに加えられた変更（たとえば、アイテムの間隔の変更、正規表現の修正、低レベルディスカバリルールへのプロトタイプの追加など）は新しいホストには複製されません。代わりに、テンプレート上の状態が適用されます。

また、ホストウィザード を使用して、ガイド付きのステップ形式のインターフェースでホストを設定することもできます。

設定

ホストタブには、ホストの一般的な属性が含まれます。



必須入力フィールドには、赤いアスタリスクが付いています。

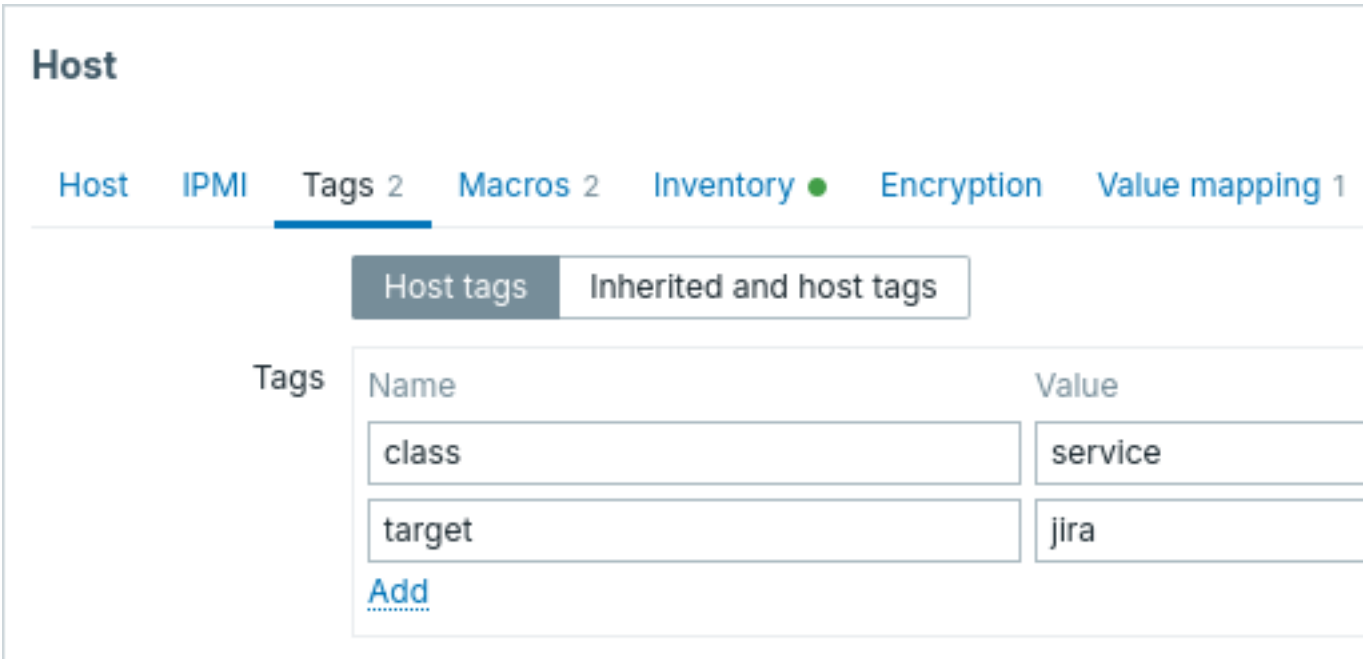
パラメーター		説明
ホスト名		一意のホスト名を入力します。英数字、スペース、ドット、ハイフン、アンダースコアを使用できます。ただし、先頭および末尾のスペースは使用できません。 注: 設定対象のホストで Zabbix エージェントが実行されている場合、エージェント設定ファイルのパラメーター *Hostname* には、ここで入力したホスト名と同じ値を設定する必要があります。このパラメーターの名前は、アクティブチェックの処理に必要です。
表示名		ホストの一意の表示名を入力します。この名前を設定すると、技術的なホスト名の代わりに、リストやマップなどでこの名前が表示されます。この属性は UTF-8 をサポートしています。
テンプレート		ホストにテンプレートをリンクします。すべてのエンティティ（アイテム、トリガーなど）がテンプレートから継承されます。 新しいテンプレートをリンクするには、テキスト入力フィールドにテンプレート名の入力を開始します。一致するテンプレートの一覧が表示されるので、スクロールして選択します。または、フィールドの横にある選択をクリックし、ポップアップウィンドウの一覧からテンプレートを選択することもできます。選択したすべてのテンプレートは、ホスト設定フォームを保存または更新するとホストにリンクされます。 テンプレートのリンクを解除するには、リンクされたテンプレートブロックにある次の 2 つのオプションのいずれかを使用します。 リンク解除 - テンプレートのリンクを解除しますが、そのエンティティ（アイテム、トリガーなど）は保持します。 リンク解除してクリア - テンプレートのリンクを解除し、そのすべてのエンティティ（アイテム、トリガーなど）を削除します。 一覧に表示されるテンプレート名はクリック可能なリンクで、テンプレート設定フォームに移動できます。
ホストグループ		ホストが所属するホストグループを選択します。ホストは少なくとも 1 つのホストグループに所属する必要があります。存在しないグループ名を追加すると、新しいグループを作成してホストにリンクできます。
インターフェース		ホストでは、複数のホストインターフェースタイプがサポートされています。エージェント、*SNMP*、*JMX*、*IPMI* です。 デフォルトではインターフェースは定義されていません。新しいインターフェースを追加するには、インターフェースブロックで追加をクリックし、インターフェースタイプを選択して、*IP/DNS*、接続先、ポートの情報を入力します。 注: いずれかのアイテムで使用されているインターフェースは削除できず、その削除リンクはグレー表示されます。 SNMP インターフェースの「IP」または「DNS」は、SNMP トラップにも使用されます。 照合時には、ホストインターフェースで選択されている「IP」または「DNS」のみが使用されます。 SNMP インターフェース（v1、v2、v3）の設定については、SNMP 監視の設定を参照してください。
	IP アドレス	ホストの IP アドレス（任意）。
	DNS 名	ホストの DNS 名（任意）。
	接続先	該当するボタンをクリックすると、エージェントからデータを取得するために使用する接続先を Zabbix サーバーに指定できます。 **IP** - ホストの IP アドレスに接続します（推奨） **DNS** - ホストの DNS 名に接続します
	ポート	TCP/UDP ポート番号。デフォルト値は、Zabbix エージェントが 10050、SNMP エージェントが 161、JMX が 12345、IPMI が 623 です。
	デフォルト	ラジオボタンを選択して、デフォルトインターフェースに設定します。
説明		ホストの説明を入力します。
監視者		ホストの監視者を選択します。 サーバー - ホストは Zabbix サーバーによって監視されます。 プロキシ - ホストはスタンドアロンプロキシによって監視されます。 プロキシグループ - ホストはプロキシグループによって監視されます。

パラメーター	説明
プロキシ有効	割り当てられたプロキシ名が表示されます（Zabbix サーバーが選択したプロキシグループからプロキシを割り当てた場合のみ）。 このフィールドは、ホストがプロキシグループによって監視されている場合のみ表示されます。 チェックボックスを選択すると、ホストが有効になり、監視できる状態になります。 チェックボックスの選択を解除すると、ホストが無効になり、監視されなくなります。 Zabbix サーバー/プロキシによって開始されるパッシブデータ要求（例: Zabbix エージェント、SNMP エージェント、シンプルチェック）では、設定の同期後に監視が無効になります。ホストにリンクされているトリガーとアクションも、設定が再読み込みされた後にのみ無効になります。 Zabbix エージェントのアクティブチェックでは、ホストが無効になったという情報を Zabbix エージェントが受信してから、その時間枠（約 5 秒）以内に監視が停止します。この短い間、ホストはアクティブチェック用のデータをローカルで収集し続け、サーバー/プロキシへの送信を試みます。ただし、ホストは無効としてマークされているため、サーバー/プロキシはデータを拒否します。 ホストを無効にすると、そのアイテムは履歴キャッシュから直ちに削除されます（ログ用に保持される最新の値を除く）。

IPMI タブには、IPMI 管理属性が含まれます。

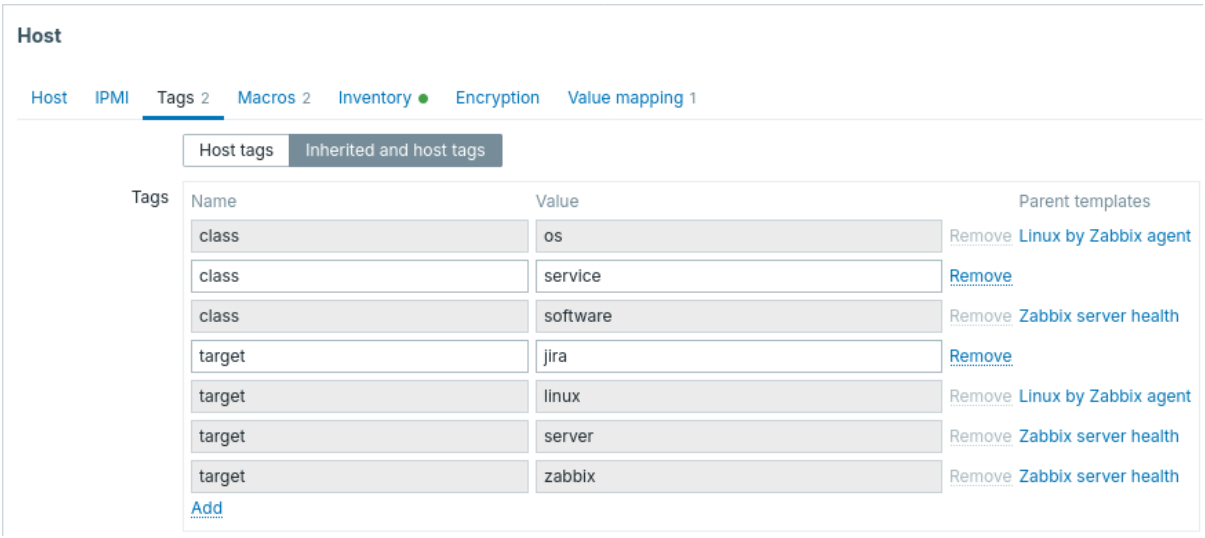
パラメーター	説明
認証アルゴリズム	認証アルゴリズムを選択します。
権限レベル	権限レベルを選択します。
ユーザー名	認証に使用するユーザー名。ユーザーマクロを使用できます。
パスワード	認証に使用するパスワード。ユーザーマクロを使用できます。

タグタブでは、ホストレベルのタグを定義できます。
このホストのすべての障害には、ここで入力した値のタグが付けられます。



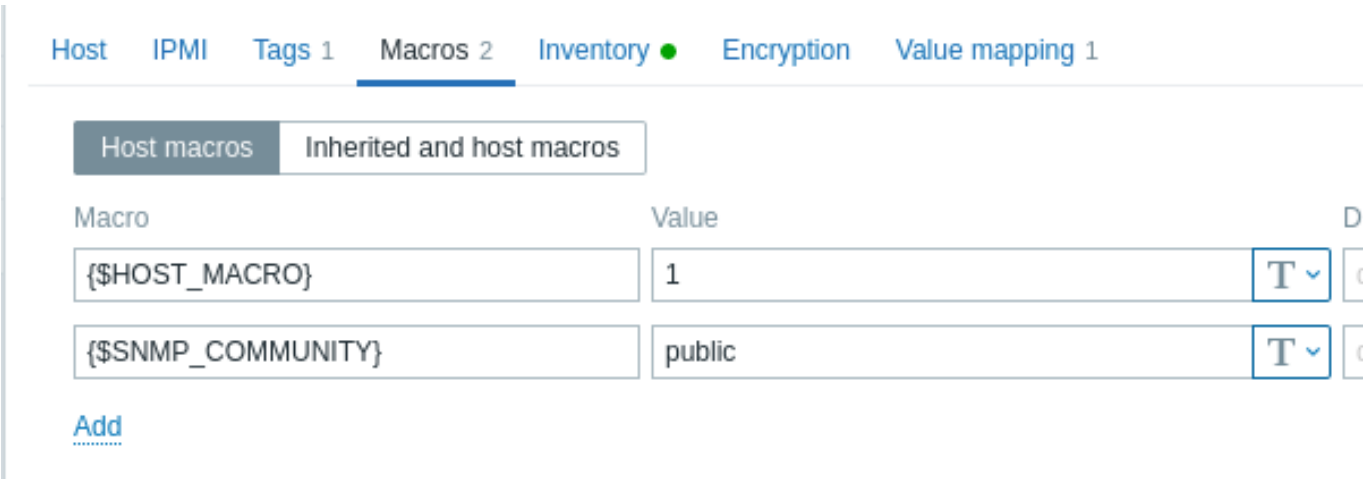
タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。

継承されたタグとホストタグオプションを選択すると、ここでテンプレートレベルのタグも表示できます。
ここには、ホストに定義されているすべてのユーザータグと、その由来が表示されます。

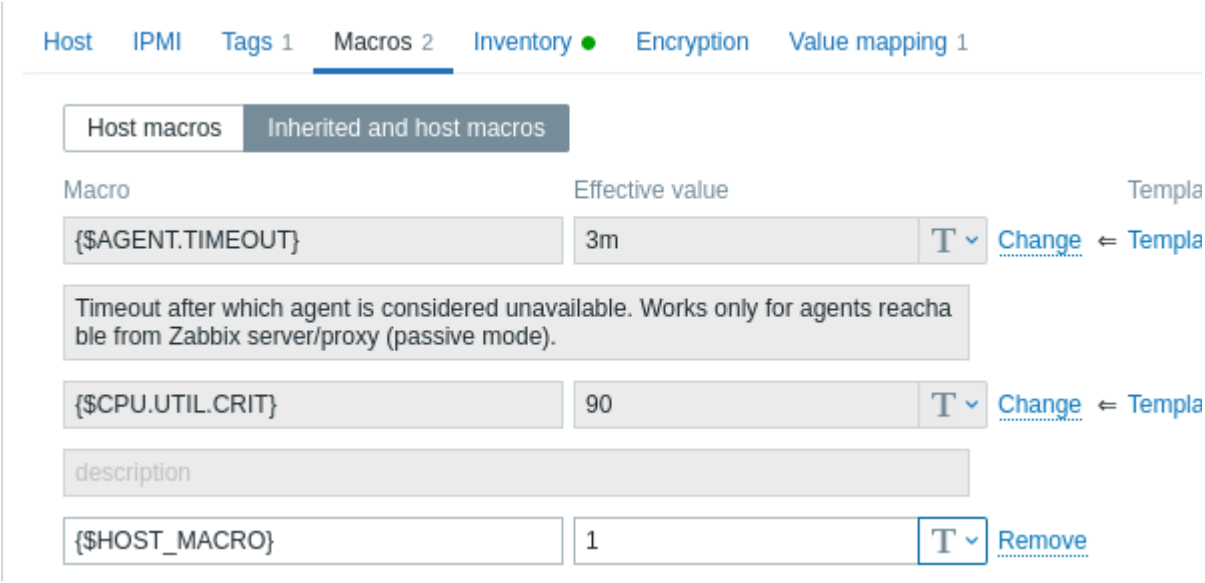


利便性のため、対応するテンプレートへのリンクが提供されています。

マクロタブでは、ホストレベルのユーザーマクロを名前と値のペアとして定義できます。
マクロの値は、プレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できます。
説明を追加することもできます。

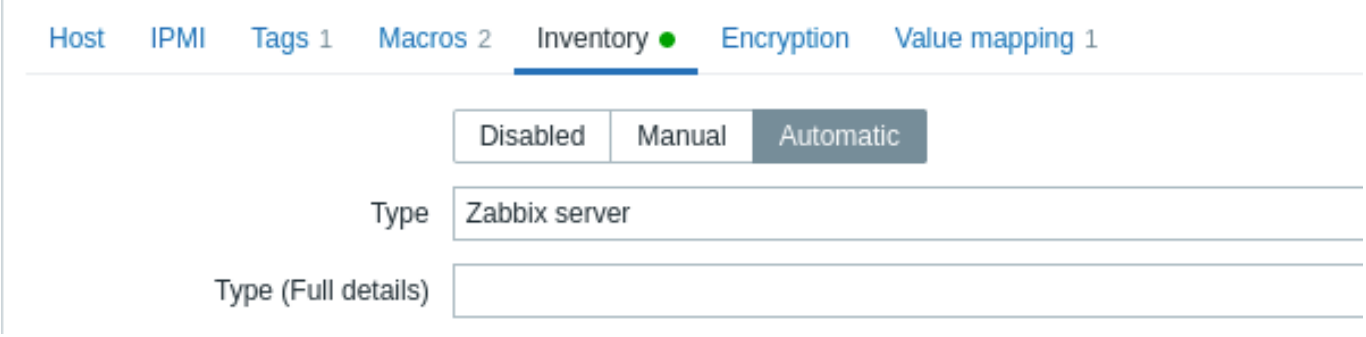


継承されたマクロとホストマクロオプションを選択すると、ここでテンプレートレベルおよびグローバルのユーザーマクロも表示できます。
ここには、ホストに定義されているすべてのユーザーマクロと、解決後の値、およびその由来が表示されます。



利便性のため、対応するテンプレートおよびグローバルマクロ設定へのリンクが提供されています。
ホストレベルでテンプレート/グローバルマクロを編集することもできます。この場合、ホスト上にマクロのコピーが作成されます。

インベントリタブでは、ホストのインベントリ情報を手動で入力できます。
また、自動インベントリ登録を有効にするか、このホストのインベントリ登録を無効にするかを選択できます。



インベントリが有効（手動または自動）になっている場合、タブ名の横に緑色のドットが表示されます。

暗号化

暗号化タブでは、ホストとの暗号化済み接続を必須にできます。

Parameter	Description
Connections to host	Zabbix サーバーまたはプロキシがホスト上の Zabbix エージェントに接続する方法: 暗号化なし（デフォルト）、PSK（事前共有鍵）を使用、または証明書を使用。
Connections from host	ホストから許可する接続の種類を選択します（つまり、Zabbix エージェントおよび Zabbix sender からの接続）。複数の接続タイプを同時に選択できます（別の接続タイプへの切り替えやテストに便利です）。デフォルトは「暗号化なし」です。
Issuer	証明書の許可された発行者。証明書はまず CA（認証局）で検証されます。有効で CA によって署名されている場合、*Issuer* フィールドを使用して許可する CA をさらに制限できます。このフィールドは、Zabbix のインストールで複数の CA の証明書を使用している場合に利用することを想定しています。このフィールドが空の場合、任意の CA が受け入れられます。
Subject	証明書の許可されたサブジェクト。証明書はまず CA で検証されます。有効で CA によって署名されている場合、*Subject* フィールドを使用して *Subject* 文字列の値を 1 つだけ許可できます。このフィールドが空の場合、設定された CA によって署名された有効な証明書はすべて受け入れられます。
PSK identity	事前共有鍵の識別文字列。 PSK identity に機密情報を含めないでください。どの PSK を使用するかを受信側に通知するため、ネットワーク上に暗号化されずに送信されます。
PSK	事前共有鍵（16 進文字列）。最大長: Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 16 進桁（256 バイトの PSK）、Zabbix が mbed TLS（PolarSSL）ライブラリを使用する場合は 64 16 進桁（32 バイトの PSK）。例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

値のマッピング

値のマッピングタブでは、値のマッピングでアイテムデータを分かりやすく表示するための設定を行えます。

ホストグループの設定

概要

ホストグループは、ホストを論理的にグループ化し、それらに対してユーザー権限を割り当てるために使用されます。

各ホストには、少なくとも 1 つのホストグループを割り当てる必要があります。1 つのホストは複数のホストグループに属することができ、各ホストグループには複数のホストを含めることができます。

Zabbix では、すべての権限はグループに基づいていることに注意してください: user groups、ホストグループ、template groupsです。そのため、たとえ 1 人のユーザーが 1 つのホストへのアクセスだけを必要とする場合でも、そのホストを含むホストグループに対する権限をそのユーザーグループに割り当てるために、ユーザーをユーザーグループに追加することでアクセスが付与されます。

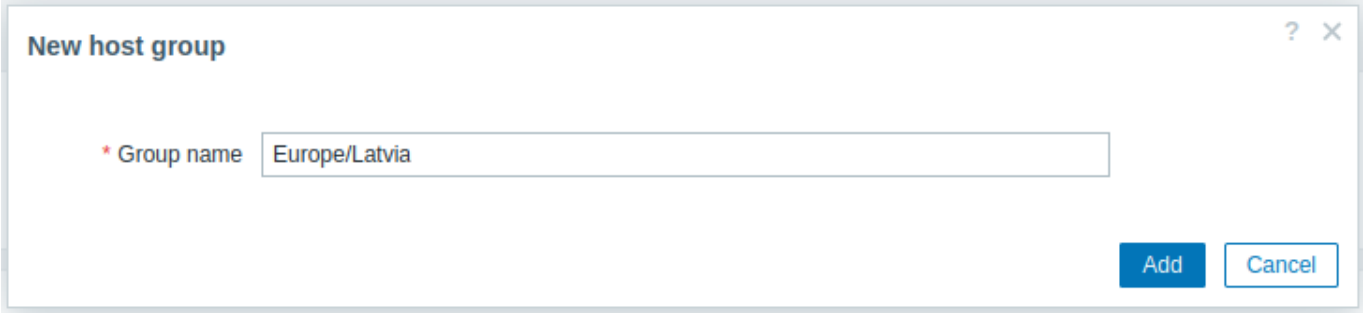
設定

Attention:
ホストグループを作成できるのは Super admin ユーザーのみです。

Zabbix Web インターフェースでホストグループを作成する方法は 2 つあります。

方法 **1**:

1. データ収集 > ホストグループに移動します。
2. 画面右上の ホストグループの作成をクリックします。
3. フォームにグループ名を入力します。



方法 **2**: ホストを設定する際に、ホストグループ入力欄に存在しないグループ名を入力します。

ホストグループが作成されると、データ収集 > ホストグループの一覧でグループ名をクリックして、グループ名の編集、グループの複製、または削除を行えます。

ホストグループを削除しても、削除されるのは論理グループのみで、グループ内のホストは削除されません。既存のホストに対して唯一のグループであるホストグループは削除できません。

ホストサブグループの作成

ホストサブグループ（またはネストされたホストグループ）は、それを含む親ホストグループの子です。

サブグループは、グループ名入力フィールドでスラッシュ「/」を使用して親グループとの関係を示すことで作成されます。例えば:

- Europe/Latvia と入力すると、親グループ Europe のサブグループ Europe/Latvia が作成されます。
- Europe/Latvia/Riga/Zabbix servers と入力すると、ネストされた親グループ Europe、Europe/Latvia、Europe/Latvia/Riga 内にそれぞれのサブグループが作成されます。

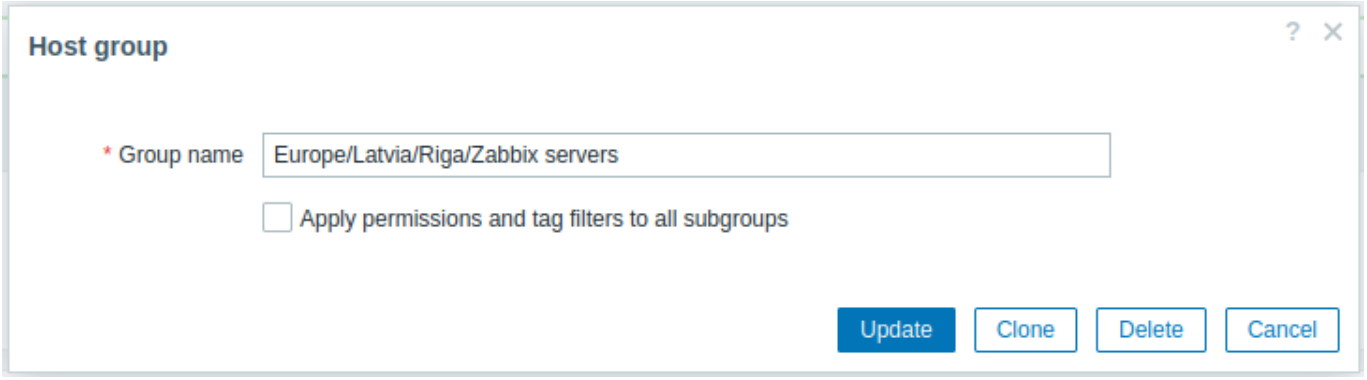
サブグループを作成する際、先頭または末尾のスラッシュや、連続した複数のスラッシュの使用はできません。/のエスケープはサポートされていません。

サブグループを作成する前に親ホストグループを作成する必要はありません。サブグループ（例：Europe/Latvia）から作成を開始するか、親ホストグループ（例：Europe）から作成を開始するかを選択できます。サブグループから作成を開始した場合、親ホストグループは自動的には作成されません。

ホストグループへの権限

- 既存の親ホストグループにサブグループを作成する場合（たとえば、Europe がすでに存在する状態で Europe/Latvia を作成する場合）、サブグループに対するユーザーグループ の権限は親から継承されます。
- 既存のサブグループに対して親ホストグループを作成する場合（たとえば、Europe/Latvia がすでに存在する状態で Europe を作成する場合）、親に対する権限は設定されません。

任意のホストグループを編集する際には、追加オプションとして 権限とタグフィルターをすべてのサブグループに適用するを設定できます。



このチェックボックスをオンにして 更新をクリックすると、編集中のホストグループの現在および将来のすべてのサブグループに、同じレベルの権限とタグフィルターが適用されます。

そのため、編集中のホストグループのサブグループに対して、いずれかのユーザーグループに異なる権限 が付与されている場合、このチェックボックスをオンにすると、現在および将来のすべてのサブグループに、編集中のグループと同じユーザー権限およびタグベースの権限が付与されます。

このオプションはデータベースには保存されず、既存の権限を上書きする点に注意してください。このオプションによって行われた変更は、手動でのみ元に戻せます。

インベントリ

概要

Zabbix でネットワークデバイスのインベントリを管理することができます。

Zabbix フロントエンドには特別なインベントリメニューがあります。しかし、初期状態ではそこにデータは表示されず、データを入力する場所でもありません。インベントリデータの構築は、ホストを設定するときに手動で行うか、自動配置オプションを使用して自動的に行われます。

インベントリの構築

手動モード

ホストを設定する際、インベントリタブで、デバイスの種類、シリアル番号、場所、担当者、URL などの詳細情報を入力できます。これらはインベントリ情報として登録されます。

ホストのインベントリ情報に URL が含まれており、その URL が「http」または「https」で始まる場合、インベントリセクションでクリック可能なリンクとして表示されます。

自動モード

ホストインベントリは自動的に登録することもできます。これを機能させるには、ホストの設定時に インベントリタブのインベントリモードを 自動に設定する必要があります。

その後、ホストのアイテムを設定して、アイテムの値を任意のホストインベントリフィールドに登録できます。アイテムの設定で、それぞれの属性（ホストインベントリフィールドに登録するアイテムと呼ばれます）を使用して登録先のフィールドを指定します。

自動インベントリデータ収集に特に役立つアイテム:

- system.hw.chassis[full|type|vendor|model|serial] - デフォルトは [full]、root 権限が必要
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - デフォルトは [all,full]
- system.hw.devices[pci|usb] - デフォルトは [pci]
- system.hw.macaddr[interface,short|full] - デフォルトは [all,full]、interface は正規表現
- system.sw.arch
- system.sw.os[name|short|full] - デフォルトは [name]
- system.sw.packages[regexp,manager,short|full] - デフォルトは [all,all,full]。

インベントリモードの選択

インベントリモードは、ホスト設定フォームで選択できます。

新しいホストのデフォルトのインベントリモードは、管理 > 一般 > その他にあるデフォルトのホストインベントリモード設定に基づいて選択されます。

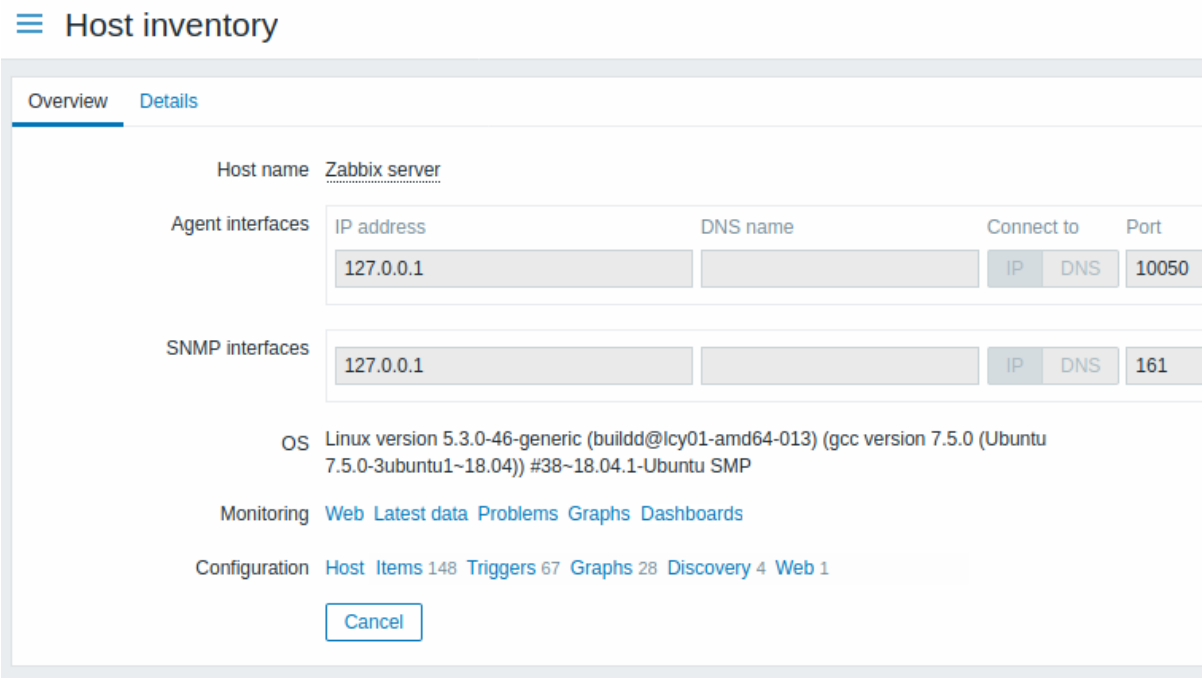
ネットワーク検出または自動登録アクションによって追加されたホストでは、手動モードまたは自動モードを選択するホストインベントリモードを設定操作を定義できます。この操作は、デフォルトのホストインベントリモード設定を上書きします。

インベントリの概要

既存のすべてのインベントリデータの詳細は、*Inventory* メニューで確認できます。

Inventory > *Overview* では、インベントリの各種フィールドごとにホスト数を確認できます。

Inventory > *Hosts* では、インベントリ情報を持つすべてのホストを確認できます。ホスト名をクリックすると、フォームでインベントリの詳細が表示されます。



Overview タブには次の項目が表示されます。

Parameter	Description
Host name	ホスト名。 名前をクリックすると、そのホストに定義されているスクリプトのメニューが開きます。 ホストがメンテナンス中の場合、ホスト名はオレンジ色のアイコンで表示されます。
Visible name	ホストの表示名（定義されている場合）。
Host (Agent, SNMP, JMX, IPMI) interfaces	このブロックには、ホストに設定されているインターフェースの詳細が表示されます。
OS	ホストのオペレーティングシステムのインベントリフィールド（定義されている場合）。
Hardware	ホストのハードウェアのインベントリフィールド（定義されている場合）。
Software	ホストのソフトウェアのインベントリフィールド（定義されている場合）。
Description	ホストの説明。
Monitoring	このホストのデータを含む監視セクションへのリンク: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*。
Configuration	このホストの設定セクションへのリンク: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*。 各リンクの後ろに、設定済みのエンティティ数が表示されます。

Details タブには、入力済みのすべてのインベントリフィールド（空でない項目）が表示されます。

インベントリマクロ

通知で使用できるホストインベントリマクロ `{INVENTORY.*}` があります。たとえば、次のように使用します。

”サーバー {INVENTORY.LOCATION1} で障害が発生しました。担当者は {INVENTORY.CONTACT1}、電話番号は {INVENTORY.POC.PRIMARY.PHONE.A1} です。”

詳細については、サポートされるマクロ のページを参照してください。

一括更新

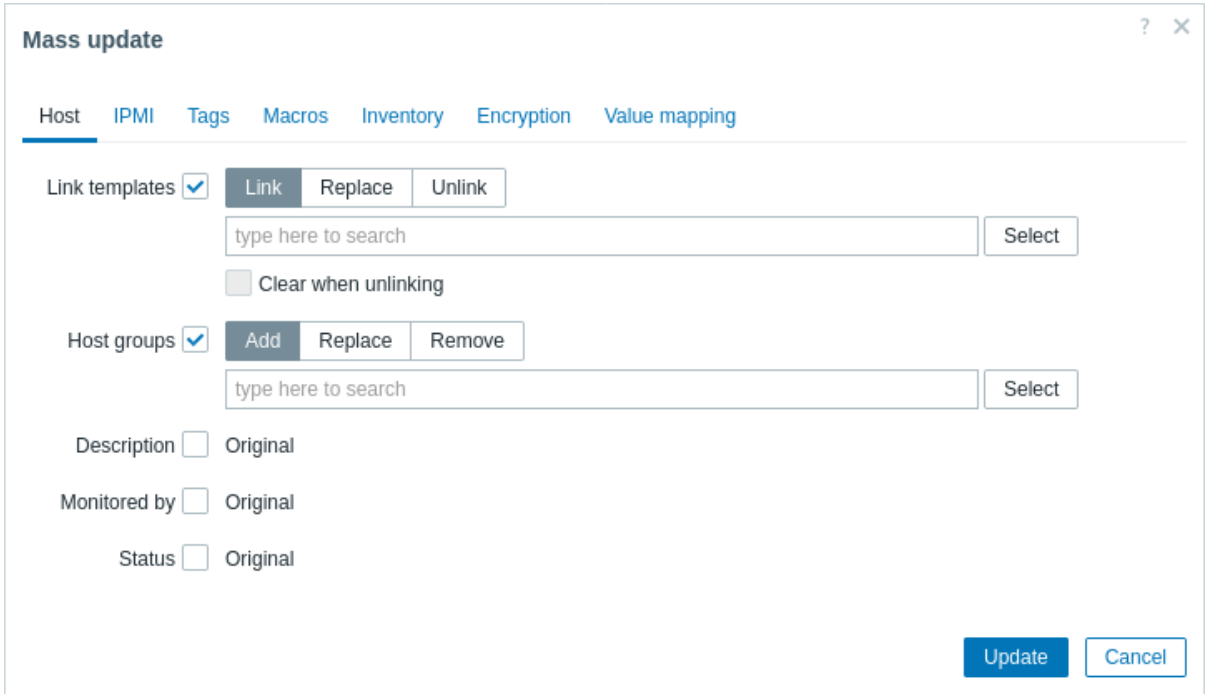
概要

一度に複数のホストの属性を変更したい場合があります。個々のホストを開いて編集する代わりに一括更新機能を使います。

一括更新の使用

複数のホストを一括更新するには、次の手順を実行します。

- 更新するホストの前にあるチェックボックスを、ホスト一覧で選択します
- 一覧の下にある 一括更新をクリックします
- 必要な属性（ホスト、*IPMI*、タグ、マクロ、インベントリ、暗号化または 値のマッピング）のタブに移動します
- 更新する属性のチェックボックスを選択し、新しい値を入力します



テンプレートのリンクを更新するために該当するボタンを選択すると、次のオプションを使用できます。

- リンク - 追加でリンクするテンプレートを指定します
- 置換 - リンクするテンプレートを指定し、それ以前にホストにリンクされていたテンプレートのリンクを解除します
- リンク解除 - リンクを解除するテンプレートを指定します

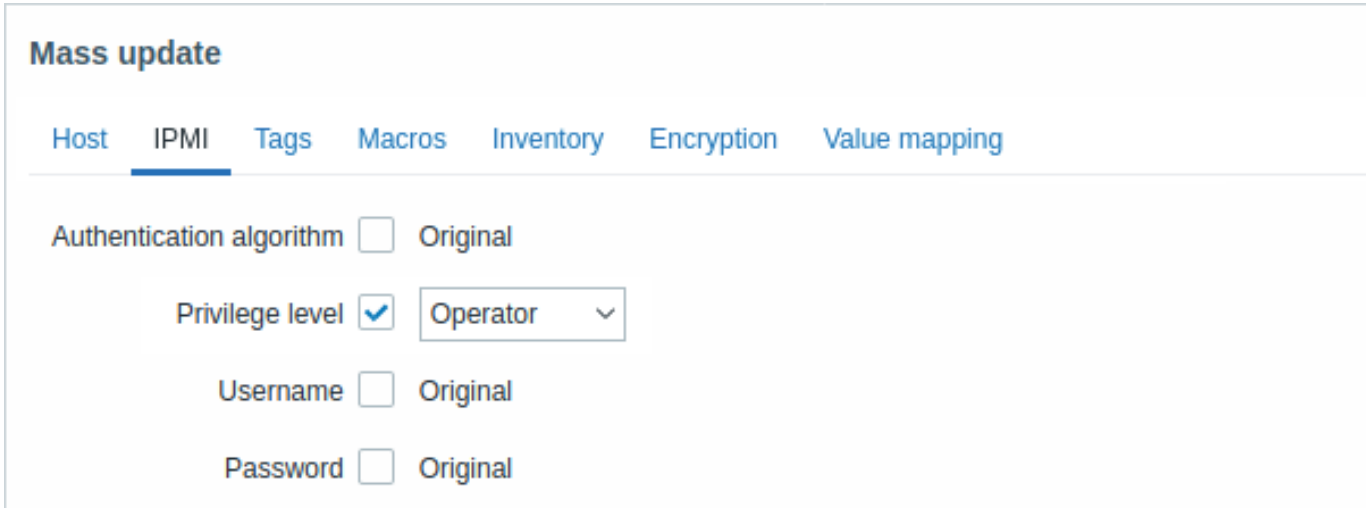
リンクまたはリンク解除するテンプレートを指定するには、オートコンプリートフィールドにテンプレート名を入力し、該当するテンプレートを示すドロップダウンが表示されるまで入力を続けます。下にスクロールして必要なテンプレートを選択します。

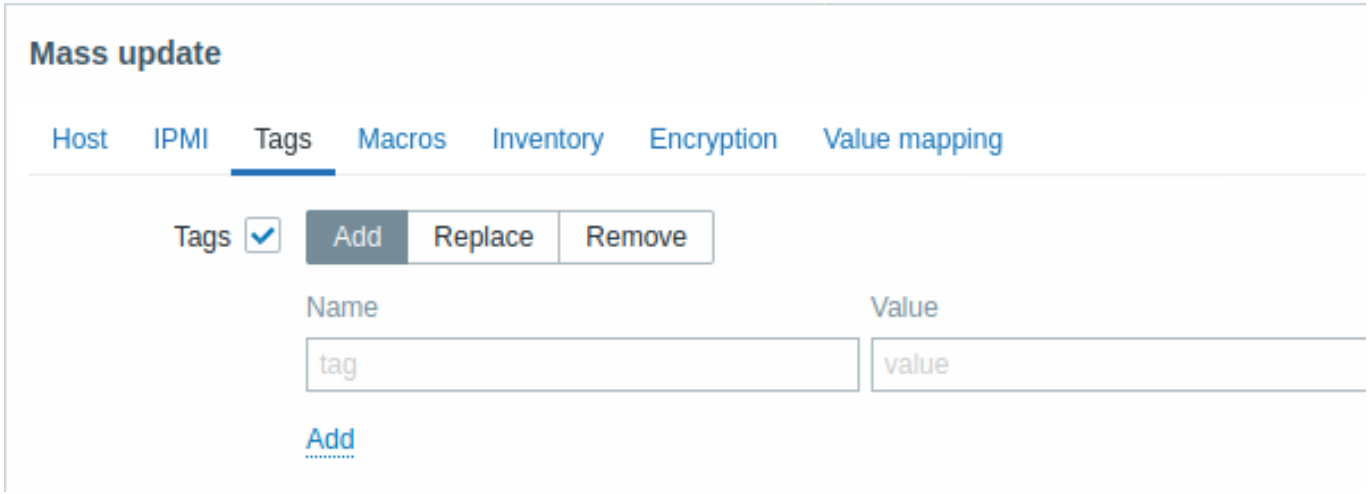
リンク解除時にクリアオプションを使用すると、以前にリンクされていたテンプレートのリンクを解除するだけでなく、そこから継承されたすべてのエンティティ（アイテム、トリガー、グラフなど）も削除できます。

ホストグループを更新するために該当するボタンを選択すると、次のオプションを使用できます。

- 追加 - 既存のホストグループから追加するホストグループを指定するか、ホスト用にまったく新しいホストグループを入力します。
- 置換 - ホストを既存のすべてのホストグループから削除し、このフィールドで指定したホストグループ（既存または新規）に置き換えます。
- 削除 - ホストから特定のホストグループを削除します。

これらのフィールドはオートコンプリートに対応しています。入力を開始すると、一致するホストグループのドロップダウンが表示されます。新しいホストグループの場合もドロップダウンに表示され、文字列の後に *(新規)* と表示されます。下にスクロールして選択します。

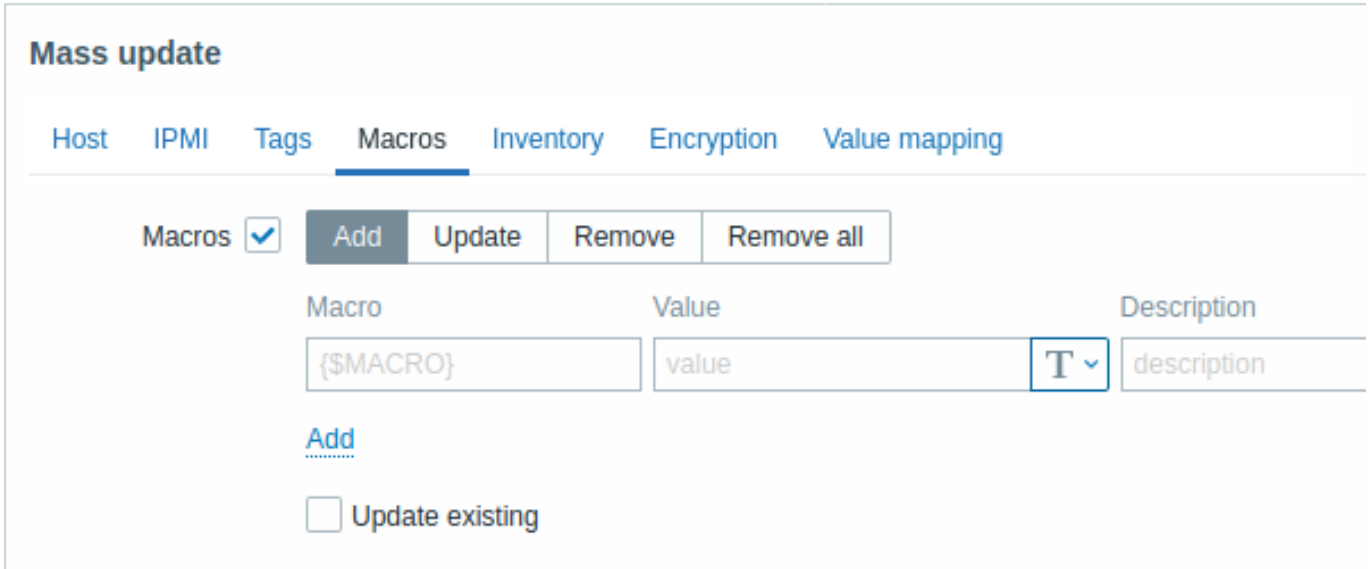




タグを更新するために該当するボタンを選択すると、次のオプションを使用できます。

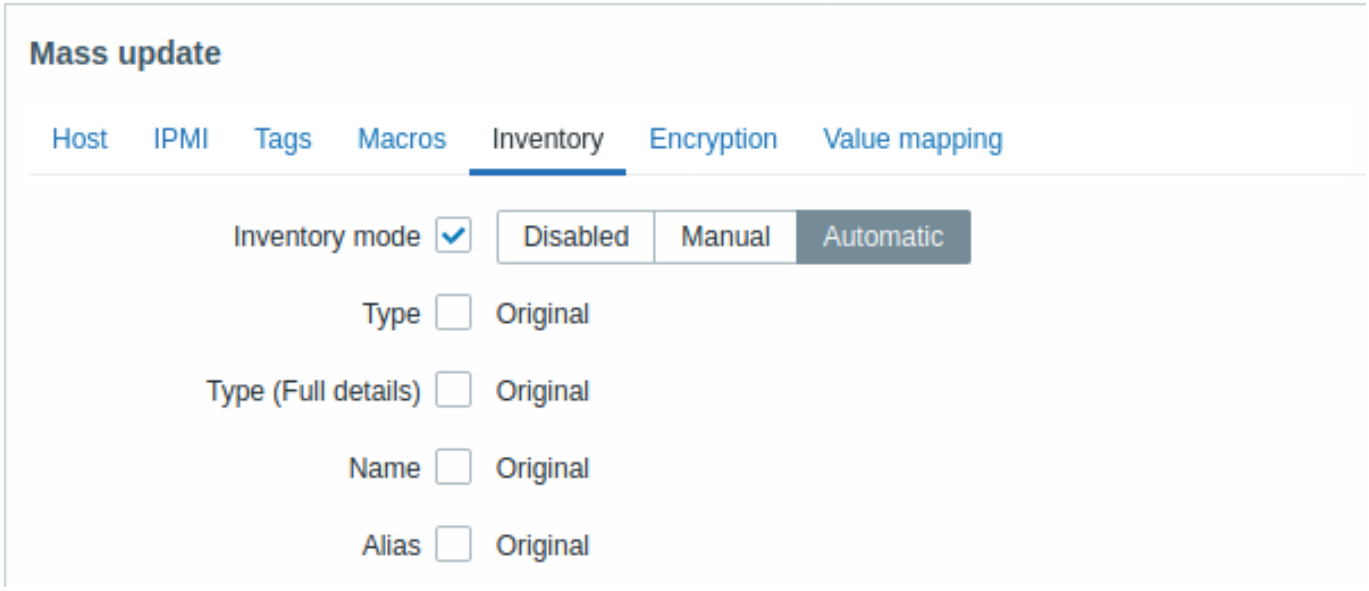
- 追加 - ホストに新しいタグを追加します。
- 置換 - ホストから既存のタグを削除し、下で指定したタグに置き換えます。
- 削除 - 指定したタグをホストから削除します。

ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT} および {HOST.ID} マクロは、タグでサポートされています。同じ名前でも値が異なるタグは「重複」とみなされず、同じホストに追加できることに注意してください。

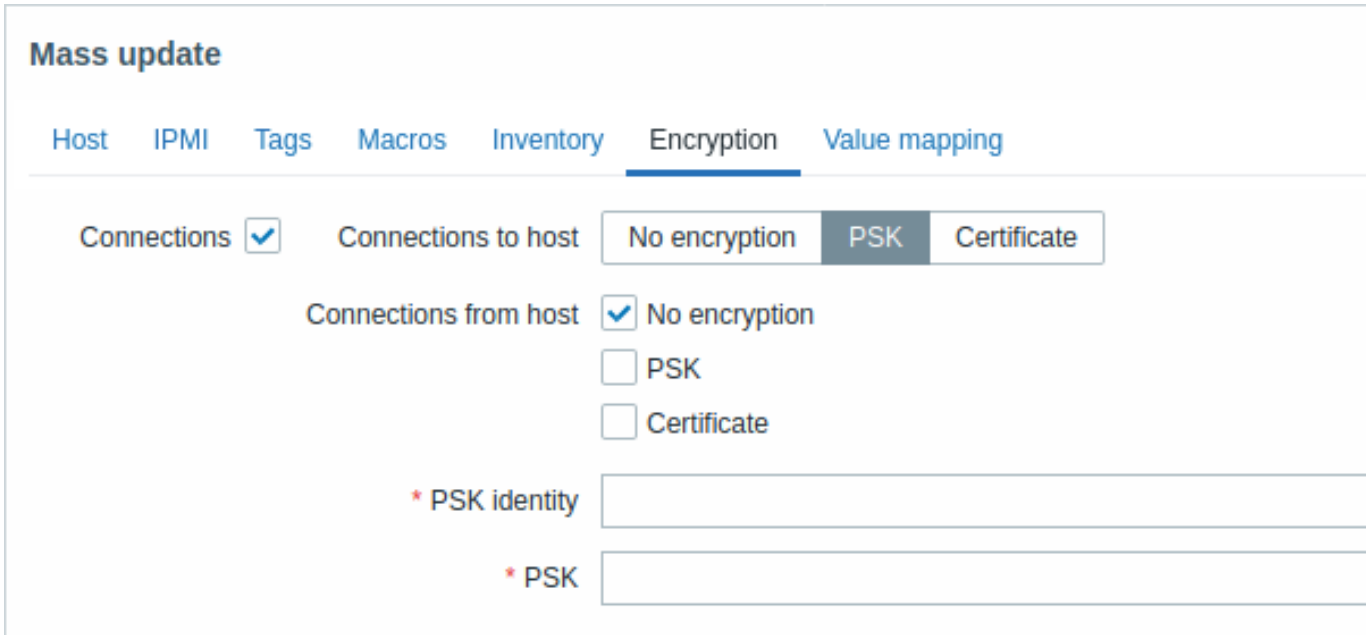


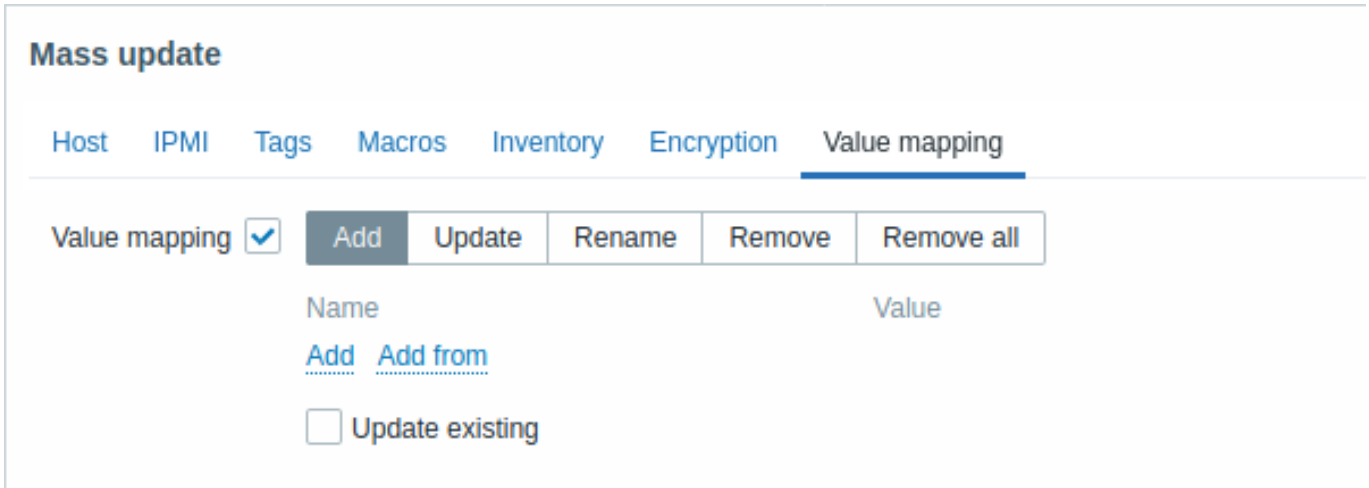
マクロを更新するために該当するボタンを選択すると、次のオプションを使用できます。

- 追加 - ホストに追加するユーザーマクロを指定します。既存の項目を更新チェックボックスを選択すると、指定したマクロ名の値、タイプ、説明が更新されます。選択しない場合、その名前のマクロがホストにすでに存在していても更新されません。
- 更新 - この一覧で指定したマクロの値、タイプ、説明を置き換えます。不足している項目を追加チェックボックスを選択すると、ホストに以前存在しなかったマクロが新しいマクロとして追加されます。選択しない場合、ホストにすでに存在するマクロのみが更新されます。
- 削除 - 指定したマクロをホストから削除します。選択した項目以外チェックボックスを選択すると、一覧で指定したマクロ以外のすべてのマクロが削除されます。選択しない場合、一覧で指定したマクロのみが削除されます。
- すべて削除 - ホストからすべてのユーザーマクロを削除します。すべてのマクロを削除することを確認しますチェックボックスを選択していない場合、すべてのマクロを削除するか確認する新しいポップアップウィンドウが開きます。



インベントリフィールドを一括更新するには、インベントリモードを「手動」または「自動」に設定する必要があります。





値のマッピングを更新するために、次のオプションを持つボタンを使用できます。

- 追加 - 値のマッピングをホストに追加します。既存の項目を更新を選択すると、この名前の値のマッピングのすべてのプロパティが更新されます。選択しない場合、その名前の値のマッピングがすでに存在していても更新されません。
- 更新 - 既存の値のマッピングを更新します。不足している項目を追加を選択すると、ホストに以前存在しなかった値のマッピングが新しい値のマッピングとして追加されます。選択しない場合、ホストにすでに存在する値のマッピングのみが更新されます。

- 名前変更 - 既存の値のマッピングに新しい名前を付けます
- 削除 - 指定した値のマッピングをホストから削除します。選択した項目以外を選択すると、指定した値のマッピングを除くすべての値のマッピングが削除されます。
- すべて削除 - ホストからすべての値のマッピングを削除します。すべての値のマッピングを削除することを確認しますチェックボックスを選択していない場合、削除を確認する新しいポップアップウィンドウが開きます。

必要な変更をすべて完了したら、更新をクリックします。選択したすべてのホストの属性が、それに応じて更新されます。

アイテム

概要

アイテムは個別の指標です。

アイテムはデータ収集に使用されます。ホストを設定したら、実際のデータを取得するためのアイテムを追加する必要があります。多くのアイテムをすばやく追加する方法は、事前定義されたテンプレートの 1 つをホストにアタッチすることです。ただし、システムのパフォーマンスを最適化するには、必要なだけのアイテムと頻度の監視を行うようにテンプレートを微調整する必要があります。

ホストからどのようなデータを収集するかを指定するには、アイテムキーを使用します。たとえば、キー名 **system.cpu.load** のアイテムはプロセッサ負荷データを収集し、キー名 **net.if.in** のアイテムは受信トラフィック情報を収集します。

キー名の後の角括弧内に追加のパラメーターを指定できます。たとえば、system.cpu.load**[avg5]** は過去 5 分間のプロセッサ負荷平均を返し、net.if.in**[eth0]** はインターフェース”eth0” の受信トラフィックを表示します。

> **Note:**
> サポートされているすべてのアイテムタイプとアイテムキーについては、アイテムタイプの個々のセクションを参照してください。

アイテムの作成と設定に進んでください。

アイテムの作成

概要

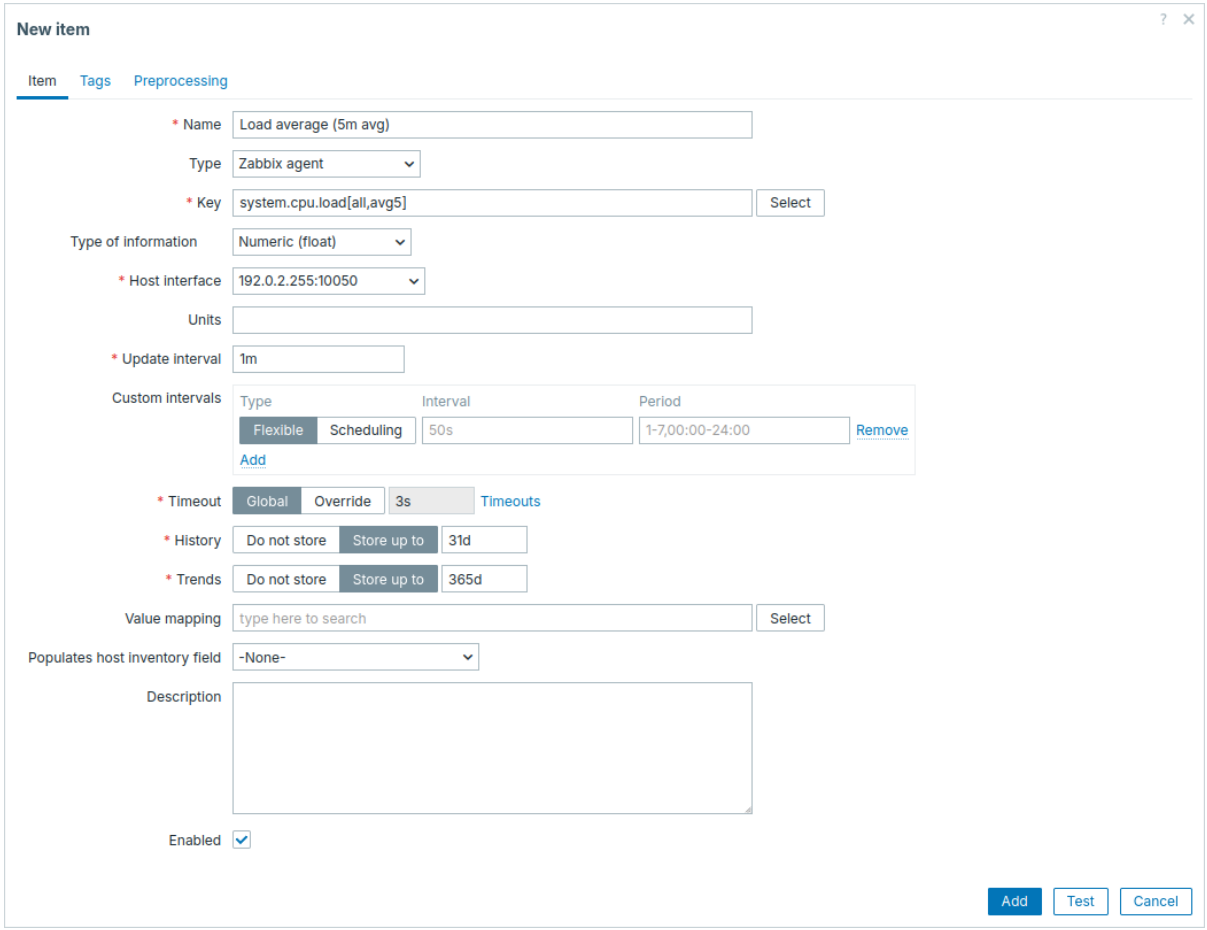
Zabbix の Web インターフェースでアイテムを作成するには、次の手順に従います。

1. データ収集 > ホストに移動します。
2. ホストの行にあるアイテムをクリックします。
3. 画面右上のアイテムの作成をクリックします。
4. フォームにアイテムのパラメータを入力します。

または、既存のアイテムを開いて複製ボタンをクリックし、別の名前で保存してアイテムを作成することもできます。

設定

アイテムタブには、アイテムの一般的な属性が含まれています。



必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
名前	アイテム名。 ユーザーマクロがサポートされています。
タイプ	アイテムのタイプ。各アイテムタイプのセクションを参照してください。
キー	アイテムキー（最大 2048 文字）。 サポートされているアイテムキーは、各アイテムタイプのセクションに記載されています。 キーは、1 つのホスト内で一意である必要があります。 キータイプが「Zabbix エージェント」、「Zabbix エージェント（アクティブ）」または「シンプルチェック」の場合、キーの値は Zabbix エージェントまたは Zabbix サーバーでサポートされている必要があります。 次も参照してください：正しいキー形式。
情報のタイプ	変換後を含め、アイテムの値を検証し、データベースに保存するために使用されるデータ型： 数値（符号なし） - 64 ビット符号なし整数。 数値（浮動小数） - 64 ビット浮動小数点数。 文字 - 短いテキスト。 ログ - ログプロパティ（タイムスタンプ、ソース、重要度、ログイベント ID）の有無にかかわらず、長いテキスト。 テキスト - 長いテキスト。 バイナリ - バイナリ値（従属アイテムでのみサポート）。 **JSON** - データベースにネイティブ形式で保存される構造化 JSON データ（計算アイテムではサポートされません）。 バイナリまたは JSON データ型の値を返すアイテムは、計算アイテムの数式またはトリガーの式ではサポートされません。 データ型およびデータベースバックエンドごとの保存上限については、アイテムデータの制限を参照してください。 特定の単一形式でデータを返すアイテムでは、一致するデータ型が自動的に選択されます。たとえば、`system.cpu.load`と`system.cpu.util`はいずれも 64 ビット浮動小数点数を返すため、数値（浮動小数）が自動的に選択されます。
ホストインターフェース	ホストインターフェースを選択します。このフィールドは、ホストレベルでアイテムを編集するときに使用できます。

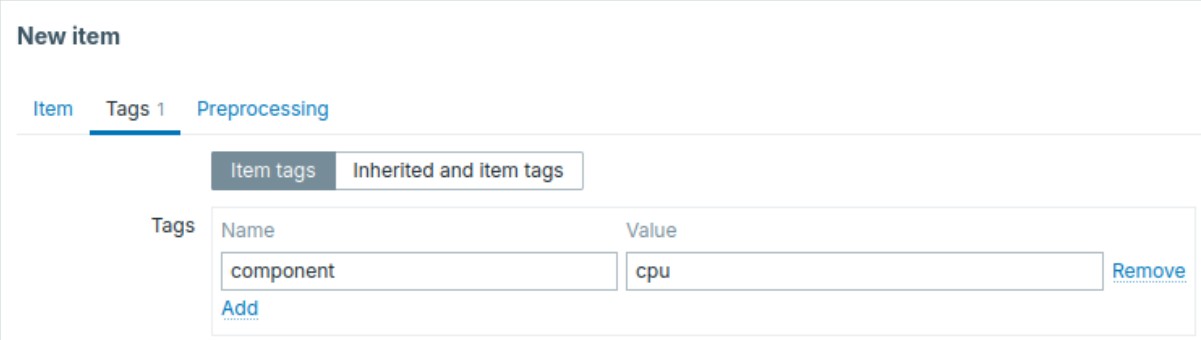
パラメータ	説明
単位	単位記号を設定すると、Zabbix は受信したアイテムの値に後処理を適用し、指定した単位の接尾辞を付けて表示します。 特殊な書式が適用される単位記号（受信したアイテムの値 → 表示値の例）： **B** - バイト（1024 → 1 KB） **Bps** - 1 秒あたりのバイト数（1024 → 1 KBps） **s** - 秒。ゼロではない最大 3 つの時間単位を使用して表示（881764 → 10d 4h 56m） **uptime** - hh:mm:ss または N 日、hh:mm:ss 形式の経過時間（881764 → 10 days, 04:56:04） **unixtime** - yyyy.mm.dd hh:mm:ss 形式の Unix タイムスタンプ（881764 → 1970-01-11 04:56:04 AM）。正しく書式設定するには、受信したアイテムの値が数値（符号なし）である必要があります。 その他の単位（Hz、W など）では、受信した値が 1000 を超える場合、1000 で割って対応する接頭辞を付けて表示します（5000 → 5 KHz、881764 → 881.76 KW）。 単位記号の先頭に**!** を付けると、単位変換と接頭辞の付加が無効になります（1024 !B → 1024 B、61 !s → 61 s）。 単位記号と単位変換の詳細およびその他の例については、アイテム値の接尾辞を参照してください。
更新間隔	このアイテムの新しい値を N 秒ごとに取得します。許可される更新間隔の最大値は 86400 秒（1 日）です。 時間の接尾辞がサポートされています（例：30s、1m、2h、1d）。 ユーザーマクロがサポートされています。 1 つのマクロでフィールド全体を埋める必要があります。1 つのフィールド内で複数のマクロを使用したり、マクロとテキストを混在させたりすることはできません。 注: カスタム間隔に 0 以外の値が存在する場合にのみ、更新間隔を「0」に設定できます。「0」に設定し、0 以外の値を持つカスタム間隔（フレキシブルまたはスケジュール）が存在する場合、アイテムはカスタム間隔の期間中にポーリングされます。 注：アイテムが有効になった直後、または更新間隔を変更した直後の最初のアイテムポーリングは、設定値より早く実行される場合があります。 新しいアイテムは、スケジューリングまたはフレキシブル更新間隔が設定され、更新間隔が 0 の場合を除き、作成から 60 秒以内にチェックされます。 既存のパッシブアイテムは、今すぐ実行ボタンを押すことで、直ちに値をポーリングできます。
カスタム間隔	アイテムをチェックするためのカスタムルールを作成できます。 フレキシブル - 更新間隔の例外（異なる頻度の間隔）を作成します。 スケジューリング - カスタムポーリングスケジュールを作成します。 詳細については、カスタム間隔を参照してください。 間隔フィールドでは時間の接尾辞がサポートされています（例：30s、1m、2h、1d）。 ユーザーマクロがサポートされています。 1 つのマクロでフィールド全体を埋める必要があります。1 つのフィールド内で複数のマクロを使用したり、マクロとテキストを混在させたりすることはできません。
タイムアウト	アイテムチェックのタイムアウトを設定します（サポートされているアイテムタイプで使用可能）。 タイムアウトオプションを選択します。 グローバル - プロキシまたはグローバルのタイムアウトを使用します（グレー表示されたタイムアウトフィールドに表示されます）。 上書き - カスタムタイムアウトを使用します（タイムアウトフィールドで設定。許可される範囲：1～600 秒）。時間の接尾辞（例：30s、1m）およびユーザーマクロがサポートされています。 タイムアウトリンクをクリックすると、プロキシのタイムアウト、またはプロキシを使用していない場合はグローバルのタイムアウトを設定できます。タイムアウトリンクは、管理 > 一般または管理 > プロキシの Web インターフェースセクションへの権限を持つ特権管理者タイプのユーザーにのみ表示されます。
ヒストリ	次のいずれかを選択します。 保存しない - アイテムのヒストリを保存しません。従属アイテムのみがヒストリを保持する必要があるマスターアイテムに便利です。 この設定は、グローバルなハウスキーパーの設定で上書きできません。 保存期間 - 詳細なヒストリをデータベースに保持する期間を指定します（1 時間～25 年）。古いデータはハウスキーパーによって削除されます。秒単位で保存されます。 時間の接尾辞（例：2h、1d）がサポートされています。ユーザーマクロがサポートされています。 保存期間の値は、管理 > ハウスキーピングでグローバルに上書きできます。  情報アイコンが表示されます。マウスをその上に置くと、警告メッセージ（例：グローバルなハウスキーパー設定（*1d*）によって上書きされています）が表示されます。 データベース内の値のヒストリのサイズを削減するため、記録された値は可能な限り短い期間だけ保持することを推奨します。値のヒストリを長期間保存する代わりに、トレンドデータをより長期間保存できます。 ヒストリとトレンドも参照してください。

パラメータ	説明
トレンド	次のいずれかを選択します。 保存しない - トレンドを保存しません。 この設定は、グローバルなハウスキーパーの設定で上書きできません。 保存期間 - 集計されたヒストリ（1 時間ごとの最小値、最大値、平均値、件数）をデータベースに保持する期間を指定します（1 日～25 年）。古いデータはハウスキーパーによって削除されます。秒単位で保存されます。 時間の接尾辞（例：24h、1d）がサポートされています。ユーザーマクロがサポートされています。 保存期間の値は、管理 > ハウスキーピングでグローバルに上書きできます。 グローバルな上書き設定が存在する場合、オレンジ色の 情報アイコンが表示されます。マウスをその上に置くと、警告メッセージ（例：グローバルなハウスキーパー設定（*7d*）によって上書きされています）が表示されます。 注: 文字、ログ、テキストなどの数値以外のデータでは、トレンドを保持できません。 ヒストリとトレンドも参照してください。
値のマッピング	このアイテムに値のマッピングを適用します。値のマッピングは受信した値を変更せず、データの表示にのみ使用されます。 数値（符号なし）、数値（浮動小数）、文字のアイテムで機能します。 例：「Windows サービスの状態」。
ログ時間の形式	ログタイプのアイテムでのみ使用できます。サポートされているプレースホルダー： **y**: 年（*1970～2038*）。 **M**: 月（*01～12*）。 **d**: 日（*01～31*）。 **h**: 時（*00～23*）。 **m**: 分（*00～59*）。 **s**: 秒（*00～59*）。 空欄の場合、タイムスタンプは Unix 時間の 0（1970 年 1 月 1 日）に設定されます。 たとえば、Zabbix エージェントのログファイルに次の行があるとします。 ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” この行は、PID 用の 6 文字分の位置で始まり、その後に日付、時刻、残りのメッセージが続きます。 この行のログ時間の形式は「ppppp:yyyyMMdd:hhmmss」です。 「p」と「:」はプレースホルダーであり、「yMdhms」以外の任意の文字を使用できます。
ホストインベントリフィールドに格納	アイテムの値を格納するホストインベントリフィールドを選択できます。ホストで自動インベントリ登録が有効になっており、アイテムが数値（符号なし）、数値（浮動小数）、文字、またはテキストの場合に機能します。
説明	アイテムの説明を入力します。ユーザーマクロがサポートされています。
有効	チェックボックスをオンにしてアイテムを有効にし、処理されるようにします。 アイテムを無効にすると、ヒストリキャッシュから直ちに削除されます（ログについては、最後の値のみ保持されます）。
最新データ	リンクをクリックすると、アイテムの最新データを表示できます。 このリンクは、既存のアイテムを編集するときにのみ使用できます。

Note:
アイテムタイプ固有のフィールドについては、対応するページで説明しています。

Note:
既存のテンプレートレベルのアイテムをホストレベルで編集する場合、多くのフィールドは読み取り専用になります。フォームヘッダーのリンクを使用してテンプレートレベルに移動し、そこで編集できます。ただし、テンプレートレベルでの変更は、そのテンプレートがリンクされているすべてのホストのアイテムに反映されます。

タグタブでは、アイテムレベルのタグを定義できます。



アイテム値の前処理

前処理タブでは、受信した値に対する変換ルールを定義できます。

アイテムのテスト

Attention:
アイテムのテストを実行するには、サーバーとプロキシのシステム時刻がsynchronizedされていることを確認してください。サーバー時刻が遅れている場合、アイテムのテストで"The task has been expired." というエラーメッセージが返されることがあります。ただし、サーバーとプロキシで異なるタイムゾーンを設定していても、テスト結果には影響しません。

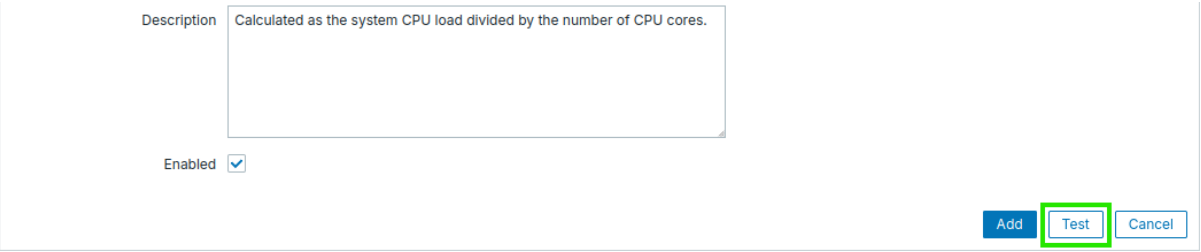
アイテムをテストして、設定が正しければ実際の値を取得できます。アイテムを保存する前でもテストできます。

テストは、ホストおよびテンプレートのアイテム、アイテムプロトタイプ、低レベルディスカバリルールで利用できます。アクティブアイテムではテストできません。

アイテムのテストは、次のパッシブアイテムタイプで利用できます。

- Zabbix エージェント
- SNMP エージェント (v1, v2, v3)
- IPMI エージェント
- SSH チェック
- Telnet チェック
- JMX エージェント
- 単純チェック (`icmpping*`、`vmware.*` アイテムを除く)
- Zabbix internal
- 計算アイテム
- 外部チェック
- データベースモニター
- HTTP エージェント
- スクリプト
- ブラウザ

アイテムをテストするには、アイテム設定フォーム下部の *Test* ボタンをクリックします。*Test* ボタンは、テストできないアイテム(アクティブチェック、除外された単純チェックなど)では無効になることに注意してください。



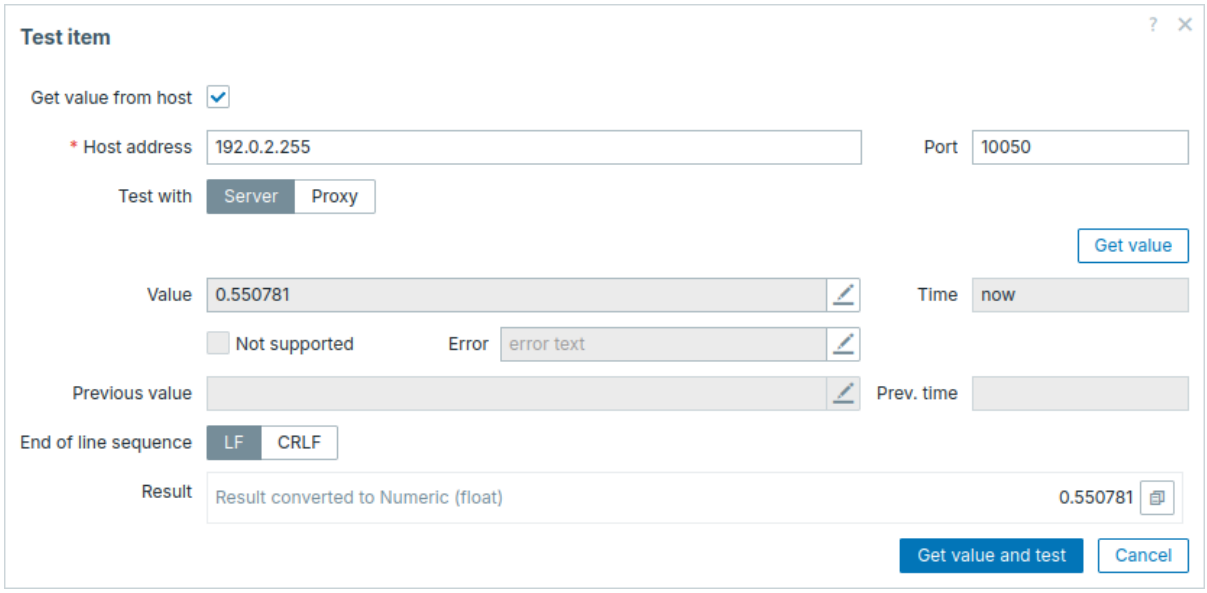
アイテムのテストフォームには、必要なホストパラメータ(ホストアドレス、ポート、サーバー/プロキシでテストする(プロキシ名))と、アイテム固有の詳細(SNMPv2 のコミュニティや SNMPv3 のセキュリティ認証情報など)を入力するフィールドがあります。これらのフィールドはコンテキストに応じて動作します。

- 値は可能な場合に事前入力されます。つまり、エージェントを必要とするアイテムでは、ホストの選択されたエージェントインターフェースから情報を取得します。
- テンプレートのアイテムでは、値を手動で入力する必要があります。
- プレーンテキストのマクロ値は解決されます。
- 値(または値の一部)が secret または Vault マクロであるフィールドは空になり、手動で入力する必要があります。アイテムパラメータに secret マクロ値が含まれている場合、次の警告メッセージが表示されます: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."

- アイテムタイプのコンテキストで不要なフィールドは無効になります（例: 計算アイテムではホストアドレスフィールドとプロキシフィールドが無効になります）

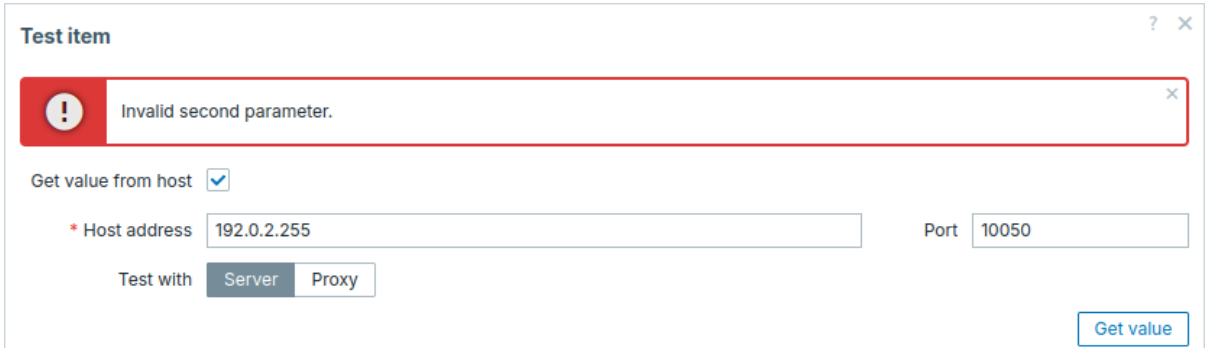
アイテムをテストするには、*Get value* をクリックします。値の取得に成功すると、*Value* フィールドに値が入力され、現在の値（存在する場合）は *Previous value* フィールドに移動します。同時に *Prev. time* フィールド、つまり 2 つの値（クリック）の時間差が計算され、取得した値に"\n\r" が含まれている場合は EOL シーケンスを検出して CRLF に切り替えようとします。

前処理をテストするには、*Get value and test* をクリックします。



ホストから取得した値とテスト結果は、Web インターフェースに送信される際に最大 512KB まで切り詰められます。テスト結果が切り詰められた場合は警告アイコンが表示され、詳細はその上にマウスを置くと確認できます。値が切り詰められると、大きな値（たとえば JSON）を比較する連続したテストでデータ型の検証に失敗することがあります。その場合はエラーメッセージが表示されます。512KB を超えるデータも、Zabbix サーバー側では引き続き完全に処理されることに注意してください。

設定が正しくない場合は、考えられる原因を説明するエラーメッセージが表示されます。



ホストから正常に取得した値は、preprocessing steps のテストにも使用できます。

フォームのボタン

フォーム下部のボタンを使用すると、いくつかの操作を実行できます。

Add	アイテムを追加します。このボタンは新しいアイテムでのみ使用できます。
Update	アイテムのプロパティを更新します。
Clone	現在のアイテムのプロパティに基づいて、別のアイテムを作成します。
Execute now	新しいアイテムの値のチェックを直ちに実行します。パッシブチェックのみサポートされます（詳細を参照）。 注値を直ちにチェックする場合、設定キャッシュは更新されないため、アイテム設定に対するごく最近の変更は値に反映されません。
Test	値を取得して、アイテム設定が正しいかどうかをテストします。
Clear history and trends	アイテムの履歴とトレンドを削除します。
Delete	アイテムを削除します。
Cancel	アイテムのプロパティの編集をキャンセルします。

アイテムデータの制限

アイテムデータの制限は、アイテムのデータ型とデータベースバックエンドによって異なります。

数値（符号なし）の値は、そのまま保存されます。浮動小数点値を受信した場合は、整数部分のみが保存されます（例: 1.23 → 1）。

数値（浮動小数点）の値はそのまま保存され、約 15〜17 桁の精度と、約 -1.79E+308 から 1.79E+308 の範囲をサポートします。科学表記の値もサポートされます（例: 1.23E+7、1e308、1.1E-4）。

テキスト値は、データベースの値型の制限に合わせるため、保存前に切り詰められます。

Database	Type of information		
	Character	**Log**	**Text**
MySQL	255 characters	65536 bytes	65536 bytes
PostgreSQL	255 characters	65536 characters	65536 characters
SQLite (only Zabbix proxy)	255 characters	65536 characters	65536 characters

バイナリ値は 16MiB（16777216 bytes）に制限されます。この制限を超える値は破棄され、対応するエラーが Web インターフェースに表示されます。

JSON データ型の値は 128MiB（134217728 bytes）に制限されます。この制限を超える値は破棄され、対応するエラーが Web インターフェースに表示されます。MySQL/MariaDB では、JSON 値を保存する `history_json` テーブルに対して日次パーティションを設定することを検討してください。これは、テーブルが急速に大きくなり、housekeeper を長時間ロックする可能性があるためです。ユースケースで大きな JSON 値（1MiB 以上）を扱う場合は、large JSON value support のためのシステム設定方法を参照してください。

Zabbix がアイテムデータをデータベースに書き込む際は、INSERT クエリを使用し、`DebugLevel` が 4 または 5 に設定されている場合は、それらをログに記録します。大きなログエントリは 64KB に切り詰められます。

カスタムスクリプトの制限

使用可能なカスタムスクリプトの長さは、使用するデータベースによって異なります。

データベース	文字数の上限	バイト数の上限
MySQL	65535	65535
PostgreSQL	65535	制限なし
SQLite (Zabbix プロキシのみ)	65535	制限なし

アイテムのタイムアウト

アイテムのタイムアウトは、Zabbix がチェックを失敗として中止するまでにどれだけ待機するかを指定します。

タイムアウトに達すると、データ取得が完了していなくてもチェックは中止されます。データが部分的に受信された場合、アイテムはサポート対象外となり、エラーメッセージがログに記録されます（たとえば、SNMP チェックで複数の OID のうち 1 つに対してのみデータの収集に成功した場合などです）。

多くのアイテムタイプでは、柔軟なアイテムのタイムアウトを設定できます。

- 単一アイテムごと
- アイテムタイプごと（プロキシレベル）
- アイテムタイプごと（グローバルレベル）

単一アイテムごとのカスタムタイムアウトは、特定のアイテムに対してより長いタイムアウトを指定しつつ、他のアイテムのタイムアウトを低く保ちたい場合に便利です。

タイムアウトの優先順位

1. 単一アイテムのタイムアウトが、他のすべてのタイムアウトより優先されます。
2. プロキシレベルのタイムアウトが、グローバルタイムアウトより優先されます。

Zabbix サーバー またはプロキシ の設定にあるタイムアウトは、柔軟なタイムアウトが設定されているチェックでは影響しません。

柔軟なタイムアウトのサポート

柔軟なアイテムタイムアウトは、次のアイテムタイプでサポートされています。

- Zabbix エージェント（パッシブチェックとアクティブチェックの両方）
- SNMP エージェント（従来の SNMP チェック[1] を除く）
- 単純チェック（`icmpping*` と VMware アイテム[2] を除く）
- SSH エージェント
- Telnet エージェント
- 外部チェック

- データベース監視
- スクリプト
- HTTP エージェント
- ブラウザ

[1] 従来の SNMP チェック（単一 OID のポーリング）では、サーバー またはプロキシ のタイムアウト設定が適用されます。
 [2] `icmpping*` アイテムでは、タイムアウト値はアイテムキー内で直接指定します。また、ハードコードされた最大タイムアウト 600 秒も適用されます。VMware 監視アイテムでは、サーバー またはプロキシ の設定にある独自の VMwareTimeout パラメータが使用されます。

サポートされていないアイテム

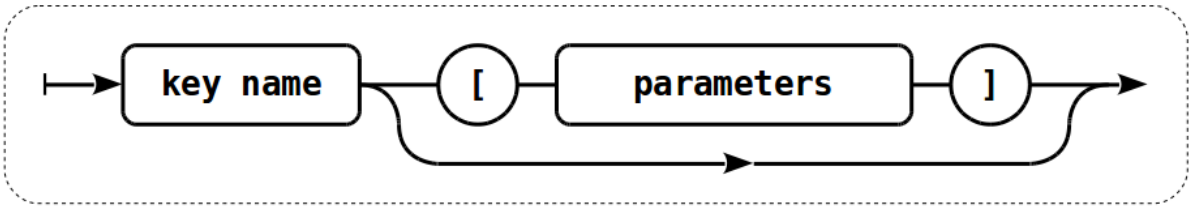
サポートされていないアイテムは、サポート外のステータスとして報告され、標準の更新間隔で再チェックされます。

アイテムの値が何らかの理由で取得できない場合（例：接続エラー、アイテムを処理するポーラーが設定されていない場合）、アイテムはサポート外になります。さらに、履歴データを受信しないアイテムはサポート外のステータスのままです。これは想定された動作であり、アイテムは新しい有効な履歴データを受信したときにのみ通常のステータスに移行します。

問題と同様に、サポート外のアイテムも新しいデータが受信されたときにのみ再評価されます。たとえそのアイテムに履歴データがもう存在しなくてもです。言い換えれば、アイテムとトリガーは新しいデータを受信したときにのみ状態が変化します。ただし、トリガー式に日付と時刻やnodata()関数が含まれている場合、トリガーは 30 秒ごとに再計算されます（詳細はトリガーの計算時間を参照してください）。

アイテムキーの形式

アイテムキーの形式は、キーのパラメータを含めて、構文ルールに従う必要があります。以下の図は、サポートされている構文を示しています。各位置で許可される要素と文字は、矢印に従って判断できます。あるブロックに線をたどって到達できる場合は許可され、到達できない場合は許可されません。



有効なアイテムキーを作成するには、まずキー名を指定し、その後にパラメータを含めるかどうかを選択します。これは、たどることができる 2 本の線で示されています。

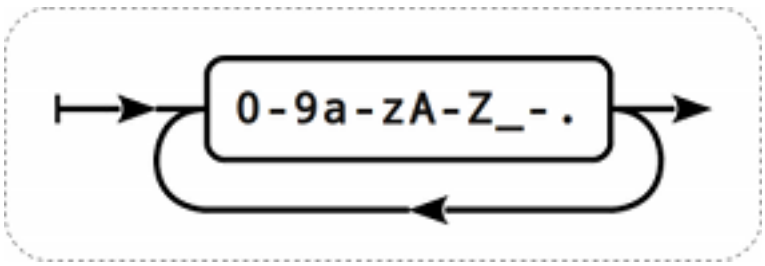
キー名

キー名自体には、使用できる文字の範囲が限られており、その文字が相互に続くだけです。許容される文字は、

`0-9a-zA-Z_-.`

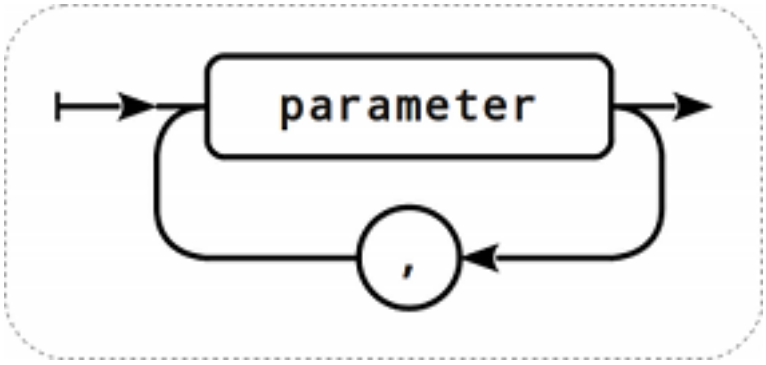
つまり、以下の文字種を指します。

- すべての数字
- すべての小文字
- すべての大文字
- アンダースコア
- ダッシュ
- ドット

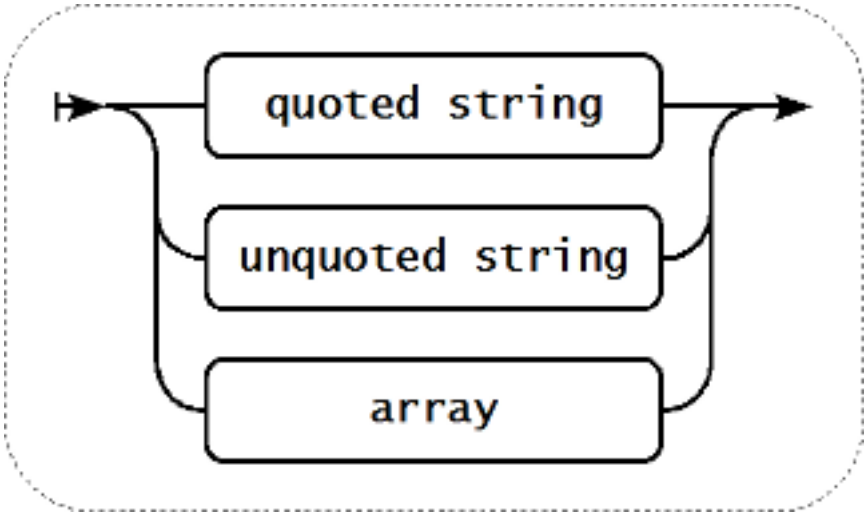


キーパラメーター

アイテムキーは、カンマ区切りで複数のパラメーターを持つことができます。



各キーパラメーターは、引用符で囲まれた文字列、引用符で囲まれていない文字列、または配列のいずれかになります。

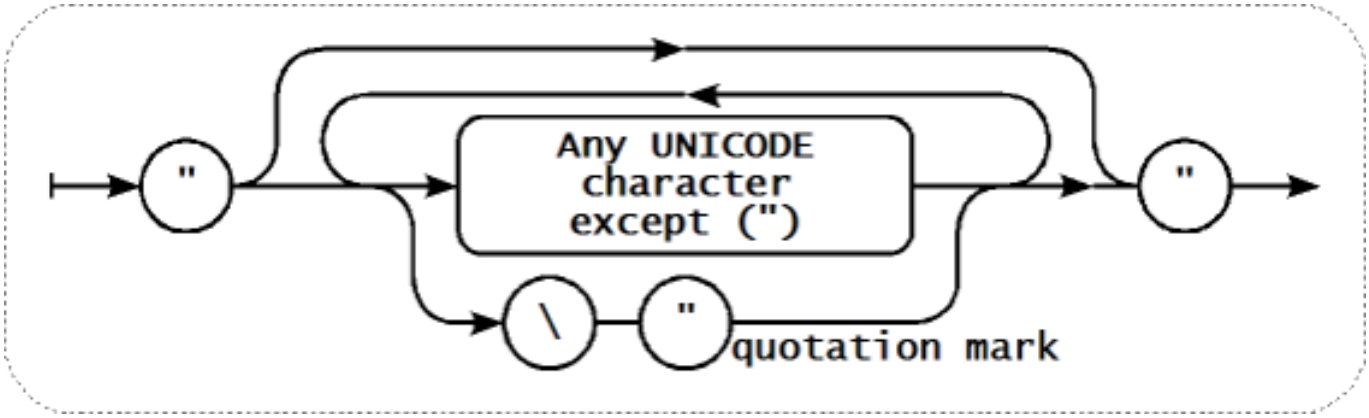


また、パラメーターを空にして、デフォルト値を使用することも可能です。その場合、さらにパラメーターを指定する場合は、適切な数のカンマを追加する必要があります。例えば、アイテムキー **icmpping[„200„500]** は、個々の ping の間隔が 200 ミリ秒、ping のタイムアウトが 500 ミリ秒、その他のパラメーターはデフォルトのままです。

パラメーターにマクロを含めることができます。マクロはユーザーマクロまたは組み込みマクロの一部です。アイテムキーパラメーターでサポートされている特定の組み込みマクロを確認するには、サポートされているマクロページで” アイテムキーパラメーター” を検索してください。

パラメーター - 二重引用符 (”) で囲まれた文字列

キーパラメーターが二重引用符で囲まれた文字列の場合、任意の Unicode 文字が使用可能です。キーパラメーターが二重引用符を含む場合、このパラメーターを二重引用符で囲み、パラメーター文字列の一部である各二重引用符はバックスラッシュ (\) でエスケープする必要があります。キーパラメーターがカンマを含む場合、このパラメーターは二重引用符で囲む必要があります。

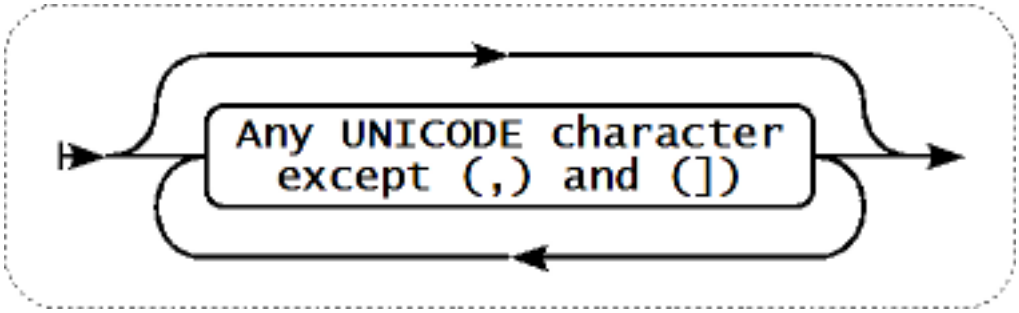


Warning:
アイテムのキーパラメーターを引用符で囲むには、二重引用符 (”) のみを使用してください。一重引用符 (’) はサポートされていません。

Attention:
`[a,[b,[c,d]],e]` のようなマルチレベルパラメーター配列は許可されていません。

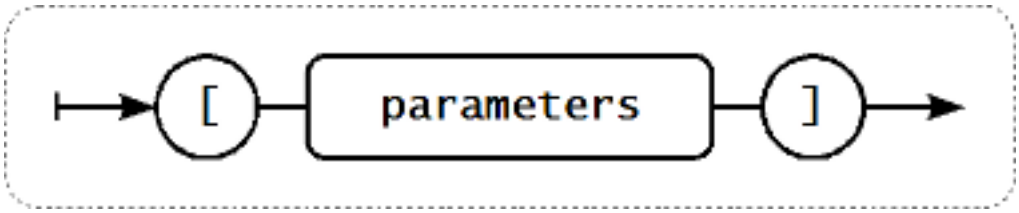
パラメーター - 二重引用符 (") がない文字列

キーパラメーターに二重引用符がない文字列の場合、カンマと右大括弧 (]) を除く全ての Unicode 文字が使用可能です。二重引用符で囲まれていないパラメータで、左大括弧 ([) で始まる文字列は使用できません。



パラメーター - 配列

キーパラメーターが配列の場合、これも角括弧で囲まれ、個々のパラメーターは複数のパラメーターを指定する規則と構文に従います。



カスタム間隔

概要

アイテムをチェックする時間に関するカスタムルールを作成できます。その方法には、デフォルトの更新間隔を再定義できる 柔軟な間隔と、特定の時刻または一連の時刻にアイテムチェックを実行できる スケジューリングの 2 つがあります。

フレキシブル間隔

フレキシブル間隔を使用すると、特定の期間に対してデフォルトの更新間隔を再定義できます。フレキシブル間隔は、*Interval* と *Period* で定義します。

- *Interval* - 指定した期間の更新間隔。時間のサフィックスがサポートされています（例: 30s、1m、2h、1d）。
- *Period* - フレキシブル間隔が有効になる期間（*Period* の形式については、時間帯を参照してください）

複数のフレキシブル間隔が重複する場合、重複する期間には最小の *Interval* 値が使用されます。重複するフレキシブル間隔の最小値が「0」の場合、ポーリングは実行されないことに注意してください。フレキシブル間隔の範囲外では、デフォルトの更新間隔が使用されます。

フレキシブル間隔が期間の長さと等しい場合、アイテムは正確に 1 回チェックされます。フレキシブル間隔が期間より長い場合、アイテムは 1 回チェックされることも、まったくチェックされないこともあります（そのため、このような設定は推奨されません）。フレキシブル間隔が期間より短い場合、アイテムは少なくとも 1 回チェックされます。

フレキシブル間隔を「0」に設定すると、フレキシブル間隔の期間中はアイテムのポーリングが行われず、期間終了後にデフォルトの *Update interval* に従ってポーリングが再開されます。例:

Interval	Period	Description
10	1-5,09:00-18:00	勤務時間中、アイテムは 10 秒ごとにチェックされます。
0	1-7,00:00-7:00	夜間はアイテムがチェックされません。
0	7-7,00:00-24:00	日曜日はアイテムがチェックされません。
60	1-7,12:00-12:01	毎日 12:00 にアイテムがチェックされます。これはスケジュールチェックの回避策として使用されていたものです。このようなチェックにはスケジューリング間隔を使用することを推奨します。

スケジューリング間隔

スケジューリング間隔は、特定の時刻にアイテムをチェックするために使用されます。柔軟な間隔はデフォルトのアイテム更新間隔を再定義するために設計されていますが、スケジューリング間隔は、並行して実行される独立したチェックのスケジュールを指定するために使用されます。

スケジューリング間隔は次のように定義されます: md<filter>wd<filter>h<filter>m<filter>s<filter> ここで:

- **md** - 月の日
- **wd** - 曜日
- **h** - 時
- **m** - 分
- **s** - 秒

<filter> は、その接頭辞に対応する値（日、時、分、秒）を指定するために使用され、次のように定義されます: [<from>[-<to>]][/<step>][,<filter>] ここで:

- <from> と <to> は一致する値の範囲 (両端を含む) を定義します。<to> が省略された場合、フィルターは <from> - <from> の範囲に一致します。<from> も省略された場合、フィルターは取り得るすべての値に一致します。
- <step> は、範囲内で値をいくつ飛ばしで扱うかを定義します。デフォルトでは <step> の値は 1 であり、定義された範囲内のすべての値に一致することを意味します。

フィルター定義は省略可能ですが、少なくとも 1 つのフィルターを使用する必要があります。フィルターには、範囲または *<step>* の値のいずれかが定義されていなければなりません。

空のフィルターは、下位レベルのフィルターが定義されていない場合は'0' に一致し、それ以外の場合は取り得るすべての値に一致します。たとえば、時フィルターが省略されている場合、分および秒フィルターも省略されていれば'0' 時のみに一致します。そうでない場合、空の時フィルターはすべての時の値に一致します。

それぞれのフィルター接頭辞に対して有効な <from> および <to> の値は次のとおりです:

Prefix	Description	*<from>*	*<to>*
md	月の日	1-31	1-31
wd	曜日	1-7	1-7
h	時	0-23	0-23
m	分	0-59	0-59
s	秒	0-59	0-59

<from> の値は <to> の値以下でなければなりません。<step> の値は 1 以上で、<to> - <from> 以下でなければなりません。

1 桁の月の日、時、分、秒の値には、先頭に 0 を付けることができます。たとえば、md01-31 と h/02 は有効な間隔ですが、md01-031 と wd01-07 は無効です。

Zabbix Web インターフェースでは、複数のスケジューリング間隔は別々の行に入力します。Zabbix API では、それらはセミコロン ; を区切り文字として 1 つの文字列に連結されます。

ある時刻が複数の間隔に一致する場合でも、実行は 1 回だけです。たとえば、wd1h9;h9 は月曜日の午前 9 時に 1 回だけ実行されます。

例:

Interval	Will be executed
m0-59	毎分
h9-17/2	9:00 から開始して 2 時間ごと (9:00, 11:00 ...)
m0,30 or m/30	毎時 hh:00 と hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	5 分ごと
wd1-5h9	毎週月曜日から金曜日の 9:00
wd1-5h9-18	毎週月曜日から金曜日の 9:00,10:00,...,18:00
h9,10,11 or h9-11	毎日 9:00、10:00、11:00
md1h9m30	毎月 1 日の 9:30
md1wd1h9m30	毎月 1 日が月曜日の場合、その日の 9:30
h9m/30	毎日 9:00、9:30
h9m0-59/30	毎日 9:00、9:30
h9,10m/30	毎日 9:00、9:30、10:00、10:30
h9-10m30	毎日 9:30、10:30
h9m10-40/30	毎日 9:10、9:40
h9,10m10-40/30	毎日 9:10、9:40、10:10、10:40
h9-10m10-40/30	毎日 9:10、9:40、10:10、10:40
h9m10-40	毎日 9:10、9:11、9:12、... 9:40
h9m10-40/1	毎日 9:10、9:11、9:12、... 9:40
h9-12,15	毎日 9:00、10:00、11:00、12:00、15:00
h9-12,15m0	毎日 9:00、10:00、11:00、12:00、15:00

Interval	Will be executed
h9-12,15m0s30	毎日 9:00:30、10:00:30、11:00:30、12:00:30、15:00:30
h9-12s30	毎日 9:00:30、9:01:30、9:02:30 ... 12:58:30、12:59:30
h9m/30;h10 (*API-specific syntax*)	毎日 9:00、9:30、10:00
h9m/30 h10 (*add this as another row in frontend*)	毎日 9:00、9:30、10:00

プロキシとエージェントのタイムゾーンの調整

Zabbix プロキシとエージェントは、スケジューリング間隔を処理する際にローカルのタイムゾーンを使用することに注意してください。

このため、Zabbix プロキシまたはアクティブエージェントによって監視されるアイテムにスケジューリング間隔を適用する場合は、キューがアイテムの遅延を正しく報告しない可能性があるため、該当するプロキシまたはエージェントのタイムゾーンを Zabbix サーバーと同じに設定することを推奨します。

Zabbix プロキシまたはエージェントのタイムゾーンは、systemd のユニットファイルで環境変数 TZ を使用して設定できます:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

アイテム値の前処理

概要

保存する前に、前処理を使用して受信したアイテムの値に変換を適用できます。これらの変換/前処理ステップは、Zabbix サーバーまたはプロキシ（アイテムがプロキシによって監視されている場合）によって実行されます。

この機能は、次のようなさまざまな用途に対応しています。

- バイトをビットに変換する（例: ネットワークトラフィック値に「8」を掛ける）;
- 増加し続ける値について、1 秒あたりの統計を計算する;
- 正規表現を適用して値を抽出または変更する;
- 値に対してカスタムスクリプトを実行する;
- データベースストレージを最適化するために、変更されていない値を破棄する。

1 つのアイテムに対して、1 つ以上の前処理ステップを設定できます。これらのステップは、設定された順序で実行されます。

Attention:
前処理ステップが失敗すると、アイテムはサポートされていない状態になります。これは、*Custom on fail* エラーハンドリング（ほとんどの変換で利用可能）を使用することで回避でき、値を破棄したり、カスタム値を設定したりできます。

ログアイテムの場合、ログメタデータ（値を含まない）によって、アイテムのサポートされていない状態は常にリセットされ、再びサポートされる状態になります。これは、最初のエラーがエージェントからログ値を受信した後に発生した場合でも同様です。

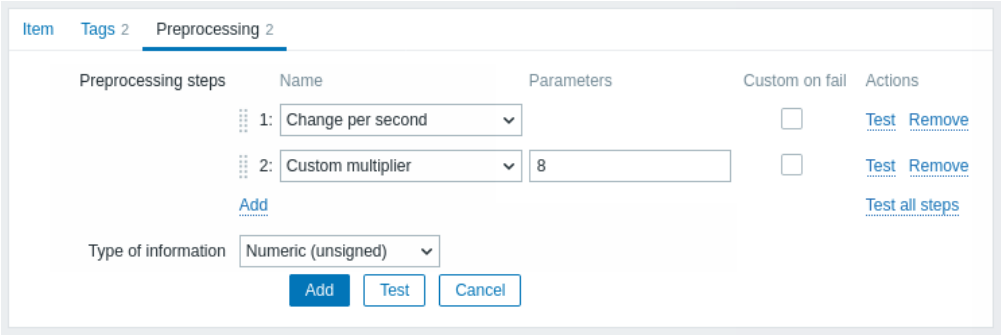
前処理に渡されるすべての値は、最初は文字列として扱われます。目的の値の型（アイテム設定で定義）への変換は、前処理パイプラインの最後に実行されます。ただし、特定の前処理ステップでは、必要に応じてそれより前に変換が行われる場合があります。技術的な詳細については、前処理の詳細を参照してください。

前処理の設定が期待どおりに動作することを確認するには、テストできます。

関連項目: 前処理の例

設定

前処理ステップは、アイテム設定フォームの前処理タブで定義します。



追加をクリックして、サポートされている変換を選択します。

少なくとも 1 つの前処理ステップが定義されている場合、タブの下部に情報の種類フィールドが表示されます。必要に応じて、前処理タブを離れずに情報の種類を変更できます。パラメータの詳細な説明については、アイテムの作成を参照してください。

サポートされている変換

サポートされているすべての変換を以下に示します。
変換名をクリックすると、その詳細を確認できます。

名前	説明	タイプ
正規表現	値を正規表現と照合し、必要な出力に置き換えます。	テキスト
置換	検索文字列を検索し、別の文字列（または何も指定しない）に置き換えます。	
トリム	値の先頭と末尾から指定した文字を削除します。	
右トリム	値の末尾から指定した文字を削除します。	
左トリム	値の先頭から指定した文字を削除します。	
XML XPath	XPath 機能を使用して、XML データから値またはフラグメントを抽出します。	構造化デ
JSON Path	JSONPath 機能を使用して、JSON データから値またはフラグメントを抽出します。	
CSV から JSON	CSV ファイルのデータを JSON 形式に変換します。	
XML から JSON	XML 形式のデータを JSON に変換します。	
SNMP walk 値	指定した OID/MIB 名で値を抽出し、書式設定オプションを適用します。	SNMP
SNMP walk から JSON	SNMP 値を JSON に変換します。	
SNMP get 値	SNMP get 値に書式設定オプションを適用します。	
カスタム乗数	指定した整数または浮動小数点値を値に乗算します。	算術
単純変化	現在の値と前回の値の差を計算します。	変化
1 秒あたりの変化	値の変化（現在の値と前回の値の差）の 1 秒あたりの速度を計算します。	
ブール値から 10 進数	値をブール形式から 10 進数に変換します。	数体系
8 進数から 10 進数	値を 8 進数形式から 10 進数に変換します。	
16 進数から 10 進数	値を 16 進数形式から 10 進数に変換します。	
JavaScript	JavaScript コードを入力します。	カスタム
範囲内	値が含まれる範囲を定義します。	検証
正規表現に一致	値が一致する必要がある正規表現を指定します。	
正規表現に一致しない	値が一致してはならない正規表現を指定します。	
JSON のエラーを確認	JSONPath にあるアプリケーションレベルのエラーメッセージを確認します。	
XML のエラーを確認	XPath にあるアプリケーションレベルのエラーメッセージを確認します。	
正規表現を使用してエラーを確認	正規表現を使用してアプリケーションレベルのエラーメッセージを確認します。	
サポートされていない値を確認	アイテムの値を取得できなかったかどうかを確認します。	
変更されていない値を破棄	値が変更されていない場合、その値を破棄します。	スロット
ハートビート付きで変更されていない値を破棄	定義した期間内に値が変更されていない場合、その値を破棄します。	
Prometheus パターン	次のクエリを使用して、Prometheus メトリクスから必要なデータを抽出します。	Prometh
Prometheus から JSON	必要な Prometheus メトリクスを JSON に変換します。	

選択した変換で失敗時のカスタム処理チェックボックスが有効になっている場合は、その項目を選択して、カスタムのエラー処理オプションを指定できます。値を破棄（値を破棄）、値を設定（指定した値を設定）、またはエラーを設定（指定したエラーメッセージを設定）から選択できます。
前処理ステップが失敗した場合、値を破棄または値を設定オプションのいずれかが選択されていれば、アイテムはサポートされていない状態にはなりません。

変化およびスロットリングの前処理ステップでは、必要に応じて新しい値を計算または比較するために、Zabbix が最後の値を記憶する必要があります。
これらの以前の値は、前処理マネージャーによって処理されます。
Zabbix サーバーまたはプロキシが再起動された場合、または前処理ステップに変更が加えられた場合、該当するアイテムの最後の値はリセットされます。その結果、次のようになります。

- 単純変化、*1* 秒あたりの変化のステップでは、変化を計算するための前回の値がないため、次の値は無視されます。
- 変更されていない値を破棄、ハートビート付きで変更されていない値を破棄のステップでは、破棄ルールにより破棄されるべき場合でも、次の値は破棄されません。

正規表現

値を正規表現と照合し、必要な出力に置き換えます。

パラメータ:

- **pattern** - 正規表現;
- **output** - 出力形式のテンプレート。\N（N=1...9）エスケープシーケンスは、N 番目に一致したグループに置き換えられます。\0 エスケープシーケンスは、一致したテキストに置き換えられます。

コメント:

- 入力値が一致しない場合、アイテムはサポート対象外になります;
- 正規表現では、\N シーケンスを使用して最大 10 個のキャプチャグループを抽出できます;
- 既存の例については、正規表現のセクションを参照してください。

{#replace} を置換

検索文字列を検索し、別の文字列（または何も指定しない）に置き換えます。

パラメータ:

- 検索文字列 - 検索して置き換える文字列。大文字と小文字が区別されます（必須）。
- 置換文字列 - 検索文字列を置き換える文字列。置換文字列を空にすることもできます。これにより、検索文字列が見つかった場合に削除できます。

コメント:

- 検索文字列のすべての出現箇所が置き換えられます。
- エスケープシーケンスを使用して、改行、キャリッジリターン、タブ、スペース”\n \r \t \s” を検索または置換できます。バックスラッシュは”\\” としてエスケープでき、エスケープシーケンスは”\\n” のようにエスケープできます。
- 改行、キャリッジリターン、タブのエスケープは、ローレベルディスカバリ中に自動的に行われます。

Trim

値の先頭と末尾から指定した文字を削除します。

右側トリム

値の末尾から指定した文字を削除します。

左トリム

値の先頭から指定した文字を削除します。

XML XPath

XPath 機能を使用して、XML データから値またはフラグメントを抽出します。

コメント:

- このオプションを使用するには、Zabbix サーバー（または Zabbix プロキシ）が libxml サポート付きでコンパイルされている必要があります。
- 名前空間はサポートされていません。

例:

```
number(/document/item/value) #<document><item><value>10</value></item></document> から「10」を抽出します
number(/document/item/@attribute) #<document><item attribute="10"></item></document> から「10」を抽出します
/document/item #<document><item><value>10</value></item></document> から「<item><value>10</value></item>」を
```

JSON Path

JSONPath 機能を使用して、JSON データから値またはフラグメントを抽出します。

CSV から JSON へ

CSV ファイルのデータを JSON 形式に変換します。

詳細については、CSV から JSON への前処理を参照してください。

XML から JSON へ

XML 形式のデータを JSON に変換します。

詳細については、シリアライズのルールを参照してください。

SNMP walk value

指定した OID/MIB 名で値を抽出し、書式設定オプションを適用します。

- 変更なし - 16 進文字列をエスケープされていない 16 進文字列として返します（注：表示ヒントは引き続き適用されます）。
- **16** 進 **STRING** から **UTF-8** - 16 進文字列を UTF-8 文字列に変換します。
- **16** 進 **STRING** から **MAC** - 16 進文字列を MAC アドレスとして検証し、適切な MAC アドレス文字列を返します（' ' は ':' に置き換えられます）。
- **BITS** から整数 - 16 進文字のシーケンス（例：「1A 2B 3C 4D」）として表現されたビット文字列の最初の 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、後続のバイトは無視されます。

SNMP walk から JSON へ

SNMP の値を JSON に変換します。

JSON 内のフィールド名と、対応する SNMP OID パスを指定します。フィールドの値には、指定した SNMP OID パスの値が設定されます。

コメント:

- *SNMP walk value* ステップと同様の値のフォーマットオプションを使用できます。
- この前処理ステップは、SNMP OID ディスカバリに使用できます。

SNMP get value

SNMP get value にフォーマットオプションを適用します。

- **Hex-STRING** から **UTF-8** - 16 進文字列を UTF-8 文字列に変換します。
- **Hex-STRING** から **MAC** - 16 進文字列を MAC アドレスとして検証し、適切な MAC アドレス文字列を返します（' ' は ':' に置き換えられます）。
- **BITS** から整数 - 16 進文字のシーケンス（例: ”1A 2B 3C 4D”）として表現されたビット文字列の最初の 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、後続のバイトは無視されます。

カスタム乗数

指定した整数または浮動小数点値を値に乗算します。

コメント:

- このオプションを使用して、KB、MBps などで受信した値を B、Bps に変換します。これにより、Zabbix は接頭辞（K、M、G など）を正しく設定できます。
- アイテムの情報の種類が *Numeric (unsigned)* の場合、カスタム乗数が適用される前に、小数部分を含む受信値は切り捨てられる（つまり、「0.9」は「0」になる）ことに注意してください。
- 情報の種類が *Numeric (unsigned)* に設定されたアイテムで、カスタム乗数を使用するか、値を *Change per second* として保存し、計算結果が実際には浮動小数点数になる場合でも、小数部分を切り捨てて値を整数として保存することで、計算値は正しい値として受け入れられます。
- 科学表記法（例: 1e+70）、ユーザーマクロおよび LLD マクロ、マクロを含む文字列（例: `{#MACRO}e+10`、`{$MACRO1}e+{$MACRO2}`）がサポートされています。マクロは整数または浮動小数点数に解決される必要があります。

単純変化

現在の値と前回の値の差を計算します。

コメント:

- 常に増加する値の測定に役立ちます。
- **value**-**prev_value** として評価されます。*value* は現在の値、*prev_value* は以前に受信した値を示します。
- 1 つのアイテムにつき、変更操作（「単純変化」または「1 秒あたりの変化」）は 1 つだけ設定できます。
- 現在の値が前回の値より小さい場合、Zabbix はその差を破棄し（何も保存せず）、次の値を待機します。

1 秒あたりの変化量

値の変化（現在の値と前回の値の差）の 1 秒あたりの速度を計算します。

コメント:

- 常に増加する値の 1 秒あたりの速度を計算する場合に便利なステップです。
- この計算では浮動小数点数が生成される可能性があるため、受信する生の値が整数であっても、「情報の種類」を数値（浮動小数）に設定することを推奨します。これは、小さい数値で小数部分が重要になる場合に特に当てはまります。浮動小数点値が大

きく、「float」フィールドの長さを超える可能性があり、その場合に値全体が失われる可能性があるときは、数値（整数）を使用して小数部分のみを切り捨てることを推奨します。
- (**value-prev_value**)/(**time-prev_time**) として評価されます。*value* - 現在の値、*prev_value* - 直前に受信した値、*time* - 現在のタイムスタンプ、*prev_time* - 直前の値のタイムスタンプ。
- 1 つのアイテムにつき、変更操作（「単純な変化」または「1 秒あたりの変化量」）は 1 つだけ許可されます。
- 現在の値が前回の値より小さい場合、Zabbix はその差を破棄し（何も保存せず）、次の値を待機します。これは、32 ビット SNMP カウンターのラップ（オーバーフロー）などを正しく処理するのに役立ちます。

ブール値から 10 進数へ

値をブール形式から 10 進数に変換します。

コメント:

- テキスト表現は 0 または 1 に変換されます。したがって、'TRUE' は 1 として保存され、'FALSE' は 0 として保存されます。すべての値は大文字と小文字を区別せずに照合されます。現在認識される値は、*TRUE* の場合は true、t、yes、y、on、up、running、enabled、available、ok、master、*FALSE* の場合は false、f、no、n、off、down、unused、disabled、unavailable、err、slave です。さらに、0 以外の数値は TRUE、0 は FALSE とみなされます。

8 進数から 10 進数へ

値を 8 進数形式から 10 進数に変換します。

16 進数から 10 進数へ

値を 16 進数形式から 10 進数に変換します。

JavaScript

パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。

> **Attention:**
> 前処理の JavaScript で未宣言の代入を使用しないでください。ローカル変数を宣言するには `var` を使用してください。

コメント:

- 利用可能な JavaScript の長さは使用するデータベースによって異なります。
- 詳細については、Javascript 前処理を参照してください。

範囲内

値が含まれる範囲を、最小値と最大値（両端を含む）を指定して定義します。

コメント:

- 数値を指定できます（桁数に制限はなく、小数部および指数部は任意で、負の値も使用できます）。
- 最小値は最大値より小さくする必要があります。
- 少なくとも 1 つの値を指定する必要があります。
- ユーザーマクロおよびローレベルディスカバリマクロを使用できます。

正規表現 {#matchesregexp} に一致

値が一致する必要がある正規表現を指定します。

正規表現 {#doesnotmatchregexp} に一致しない

値が一致してはならない正規表現を指定します。

JSON のエラーを確認

JSONPath にあるアプリケーションレベルのエラーメッセージを確認します。成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前にあった値で処理を続行します。

コメント:

- これらの外部サービスエラーは、前処理ステップの情報を追加せず、そのままユーザーに報告されます。
- 無効な JSON の解析に失敗した場合、エラーは報告されません。

XML のエラーを確認

XPath で指定されたアプリケーションレベルのエラーメッセージを確認します。
確認に成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前にあった値で処理を続行します。

コメント:

- これらの外部サービスエラーは、前処理ステップの情報を追加せず、そのままユーザーに報告されます。
- 無効な XML の解析に失敗した場合、エラーは報告されません。

正規表現を使用したエラーのチェック

正規表現を使用して、アプリケーションレベルのエラーメッセージをチェックします。
成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。

パラメータ:

- **pattern** - 正規表現;
- **output** - 出力フォーマット用テンプレート。
  \N（N=1...9）エスケープシーケンスは、N 番目に一致したグループに置き換えられます。
  \0 エスケープシーケンスは、一致したテキストに置き換えられます。

コメント:

- これらの外部サービスエラーは、前処理ステップの情報を追加せず、そのままユーザーに報告されます。

サポートされていない値を確認

アイテムの値を取得できなかったかどうかを確認します。返されたエラーメッセージを調べ、その内容に基づいて障害を処理する方法を指定します。

パラメータ:

- **scope** - エラー処理の範囲を選択します:
any error - すべてのエラー;
error matches - *pattern* に指定した正規表現に一致するエラーのみ;
error does not match - *pattern* に指定した正規表現に一致しないエラーのみ
- **pattern** - エラーに一致させる正規表現。scope パラメータで *any error* を選択した場合、このフィールドは表示されません。表示される場合、このフィールドは必須です。

コメント:

- 通常、値を取得できない場合、アイテムはサポートされていない状態になります。この前処理ステップを使用すると、この動作を変更できます。
- この前処理ステップでは、アイテムの値を取得できなかったかどうかのみを確認します。取得した値の型（文字列など）がアイテムの情報の型（数値など）と一致するかどうかは確認しません。詳細については、前処理の例を参照してください。型が一致しない場合、すべての前処理ステップの実行後も、アイテムがサポートされていない状態になることがあります。型の不一致を確認するには、たとえば カスタム乗数前処理ステップを使用できます。詳細については、前処理の例を参照してください。
- 正規表現のキャプチャグループは、値を次に設定または エラーを次に設定フィールドで使用できます。一致した N 番目のグループを取得するには \N（N=1...9）を使用し、一致したテキストを取得するには \0 を使用します。
- これらのステップは常に最初の前処理ステップとして実行され、アイテムへの変更を保存した後、他のすべてのステップより上に配置されます。
- 複数の サポートされていない値を確認ステップを、指定した順序で使用できます。*any error* のステップは、このグループの最後のステップとして自動的に配置されます。

変更されていない値を破棄する

値が変更されていない場合は破棄します。

コメント:

- 値が破棄された場合、データベースには保存されず、Zabbix サーバーはこの値が受信されたことを認識しません。トリガー式は評価されず、その結果、関連するトリガーの問題は作成/解決されません。関数は、実際にデータベースに保存されたデータに基づいてのみ動作します。トレンドはデータベース内のデータに基づいて構築されるため、1 時間に値が保存されていない場合、その時間のトレンドデータもありません。
- アイテムごとに指定できるスロットリングオプションは 1 つだけです。

ハートビートで変更されていない値を破棄

定義した期間（秒単位）内に変更されていない値を破棄します。

コメント:

- 秒数を指定するには、正の整数値を使用できます（最小値は 1 秒です）。
- 時間のサフィックスを使用できます（例: 30s、1m、2h、1d）。
- ユーザーマクロおよびローレベルディスカバリマクロを使用できます。
- 値が破棄された場合、その値はデータベースに保存されず、Zabbix サーバーはこの値が受信されたことを認識しません。その結果、トリガーの関連する障害が作成または解決されることはなく、トリガーの式は評価されません。関数は、実際にデータベースに保存されたデータに基づいてのみ動作します。トレンドはデータベース内のデータに基づいて作成されるため、1 時間値が保存されなかった場合、その 1 時間分のトレンドデータも存在しません。
- 1 つのアイテムにつき、指定できるスロットリングオプションは 1 つだけです。

Prometheus パターン

Prometheus メトリクスから必要なデータを抽出するには、次のクエリを使用します。

詳細はPrometheus チェックを参照してください。

Prometheus から JSON へ

必要な Prometheus メトリクスを JSON に変換します。

詳細はPrometheus チェックを参照してください。

マクロのサポート

ユーザーマクロおよびコンテキスト付きユーザーマクロは、以下でサポートされています。

- 前処理ステップのパラメータ（JavaScript コードを含む）
- カスタムエラーハンドリングパラメータ（値を設定およびエラーを設定フィールド）

Note:
マクロが値に置き換えられるとき、マクロのコンテキストは無視されます。マクロ値はそのままコードに挿入され、JavaScript コードに値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があることにご注意ください。

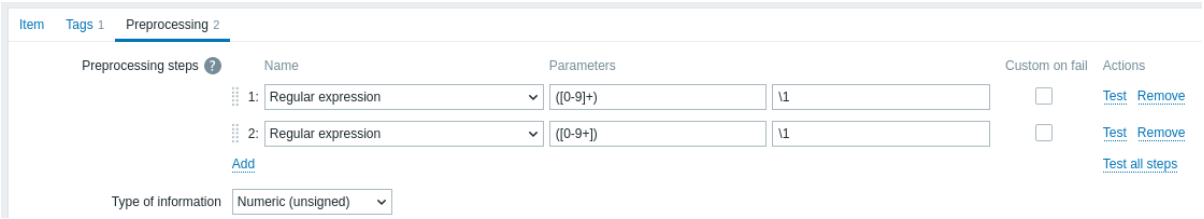
テスト

事前処理のテストを参照してください。

前処理のテスト

テスト

前処理ステップのテストは、複雑な前処理パイプラインが期待どおりの結果を返すことを確認するのに役立ちます。これにより、アイテムの値が受信されて前処理されるのを待たずに済みます。

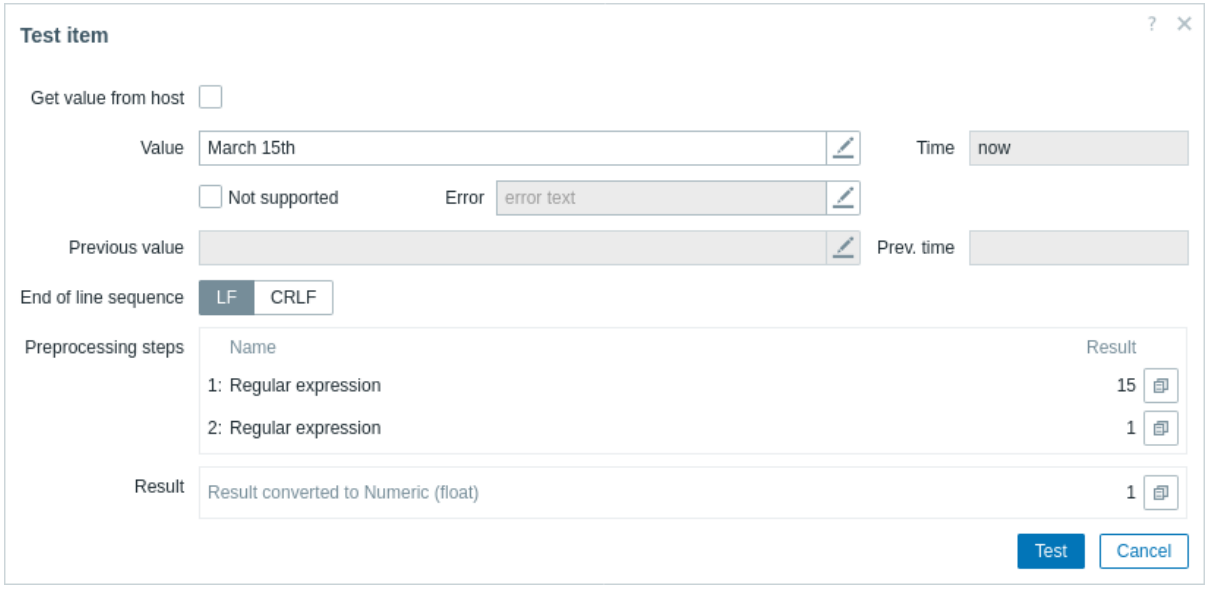


テストできる内容は次のとおりです。

- 仮想的な値に対して
- ホストからの実際の値に対して

各前処理ステップは個別にテストでき、すべてのステップをまとめてテストすることもできます。Actions ブロックでそれぞれ *Test* または *Test all steps* ボタンをクリックすると、テストウィンドウが開きます。

仮想値のテスト



Parameter	Description
Get value from host	仮想値をテストしたい場合は、このチェックボックスをオフのままにします。 参照: 実値のテスト。
Value	テストする入力値を入力します。 パラメータ欄または表示/編集ボタン をクリックすると、値またはコードブロックを入力するためのテキストエリアウィンドウが開きます。
Not supported	サポートされていない値をテストする場合は、このチェックボックスをオンにします。 このオプションは、サポートされていない値のチェックの前処理ステップをテストするのに便利です。
Error	エラーテキストを入力します。 このフィールドは、*Get value from host* がオフで、かつ *Not supported* がオンのときに有効になります。 *Get value from host* がオンの場合、このフィールドにはホストからの実際のエラーメッセージが入力されます（読み取り専用）。
Time	入力値の時刻が表示されます: `now`（読み取り専用）。
Previous value	比較対象となる前回の入力値を入力します。 *Change* および *Throttling* の前処理ステップでのみ使用します。
Previous time	比較対象となる前回の入力値の時刻を入力します。 *Change* および *Throttling* の前処理ステップでのみ使用します。 デフォルト値は、アイテムの'Update interval' フィールドの値に基づきます（'1m' の場合、このフィールドには `now-1m` が入力されます）。何も指定されていない場合、またはユーザーがホストにアクセスできない場合、デフォルトは `now-30s` です。
Macros	マクロが使用されている場合は、その値とともに一覧表示されます。値はテスト目的で編集できますが、変更はテストコンテキスト内でのみ保存されます。
End of line sequence	複数行の入力値に対する行末シーケンスを選択します: **LF** - LF（改行）シーケンス **CRLF** - CRLF（キャリッジリターン + 改行）シーケンス。
Preprocessing steps	前処理ステップが一覧表示されます。*Test* ボタンをクリックすると、各ステップのテスト結果が表示されます。 テスト結果は、Web インターフェースに送信される際に最大 512KB まで切り詰められます。テスト結果は、切り詰められた 512KB を超えない範囲でコピーできます。結果が切り詰められた場合は警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも、Zabbix サーバーによって完全に処理される点に注意してください。 ステップのテストに失敗した場合は、エラーアイコンが表示されます。エラーの説明はマウスオーバーで表示されます。 ステップに"Custom on fail" が指定され、そのアクションが実行された場合は、前処理テストステップ行の直後に新しい行が表示され、実行されたアクションとその結果（エラーまたは値）が示されます。

Parameter	Description
Result	すべてのステップをまとめてテストした場合（*Test all steps* ボタンをクリックした場合）、前処理ステップの最終結果がすべてのケースで表示されます。 アイテムの値タイプへの変換タイプも表示されます。たとえば `Result converted to Numeric (unsigned)` のように表示されます。 テスト結果は、Web インターフェースに送信される際に最大 512KB まで切り詰められます。最終結果は、切り詰められた 512KB を超えない範囲でコピーできます。結果が切り詰められた場合は警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも、Zabbix サーバーによって完全に処理される点に注意してください。

各前処理ステップの後の結果を確認するには、*Test* をクリックします。

テスト値は、個別のステップまたはすべてのステップについて、テストセッション間で保存されます。これにより、ユーザーは前処理ステップやアイテム設定を変更した後でも、情報を再入力せずにテストウィンドウへ戻ることができます。ただし、ページを更新すると値は失われます。

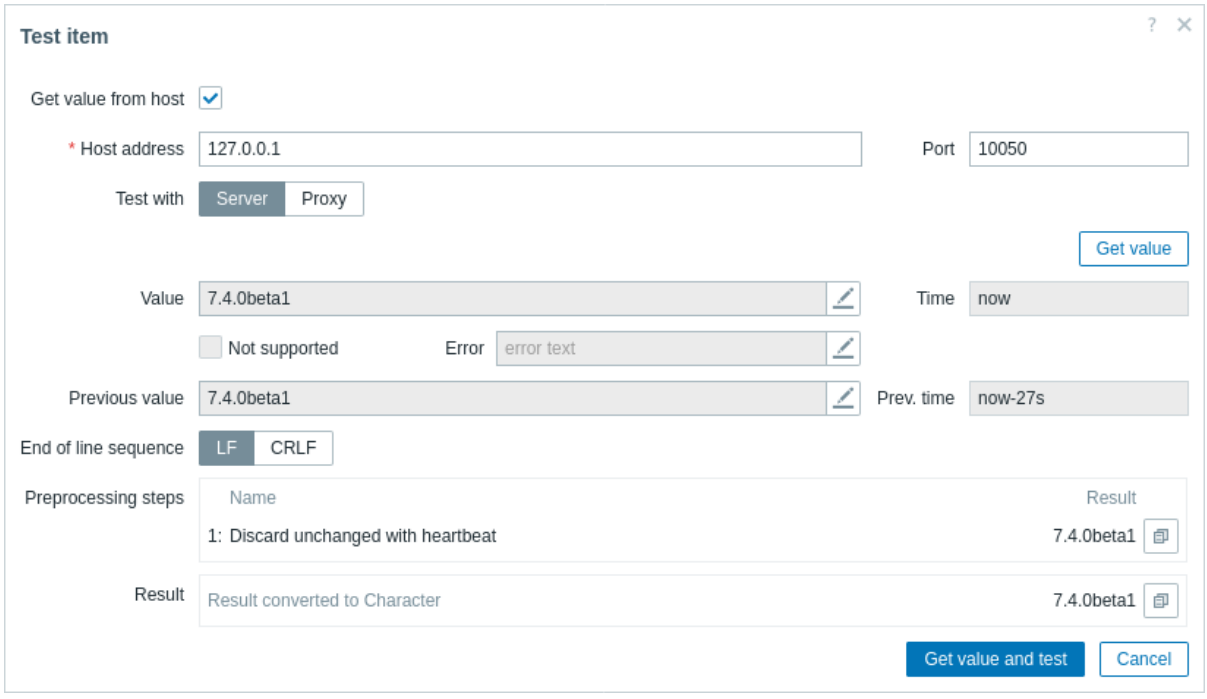
テストは Zabbix サーバーによって実行されます。Web インターフェースは対応するリクエストをサーバーに送信し、結果を待機します。リクエストには入力値と前処理ステップ（展開済みのユーザーマクロを含む）が含まれます。*Change* および *Throttling* ステップでは、任意で前回の値と時刻を指定できます。サーバーは各前処理ステップの結果を返します。

すべての技術的エラーまたは入力検証エラーは、テストウィンドウ上部のエラーボックスに表示されます。

実際の値をテストする

実際の値に対して前処理をテストするには、次の手順を実行します。

- ホストから値を取得チェックボックスを選択します
- ホストパラメータ（ホストアドレス、ポート、プロキシ名またはプロキシなし）と、アイテム固有の詳細（SNMPv2 コミュニティや SNMPv3 セキュリティ認証情報など）を入力または確認します。これらのフィールドはコンテキストに応じて処理されます。
  - 可能な場合、値はあらかじめ入力されます。たとえばエージェントを必要とするアイテムの場合、ホストで選択されているエージェントインターフェースの情報が使用されます
  - テンプレートのアイテムでは、値を手動で入力する必要があります
  - プレーンテキストのマクロ値は解決されます
  - フィールドの値（または値の一部）がシークレットまたは Vault マクロの場合、フィールドは空になり、手動で入力する必要があります。いずれかのアイテムパラメータにシークレットマクロの値が含まれている場合、次の警告メッセージが表示されます。「アイテムには、シークレット値を持つユーザー定義マクロが含まれています。これらのマクロの値は手動で入力する必要があります。」
  - アイテムの種類のコンテキストで不要なフィールドは無効になります（たとえば、計算アイテムではホストアドレスとプロキシのフィールドが無効になります）
- 値を取得してテストをクリックして前処理をテストします



アイテム設定フォームで値のマッピング（値を表示フィールド）を指定した場合、アイテムのテストダイアログには、最終結果の後に 値のマッピング適用後の結果という別の行が表示されます。

ホストから実際の値を取得する際に固有のパラメータ：

パラメータ	説明
ホストから値を取得	ホストから実際の値を取得するには、このチェックボックスを選択します。
ホストアドレス	ホストアドレスを入力します。 このフィールドには、アイテムのホストインターフェースのアドレスが自動的に入力されます。
ポート	ホストのポートを入力します。 このフィールドには、アイテムのホストインターフェースのポートが自動的に入力されます。
SNMP インターフェースの追加フィールド * * (*SNMP* バージョン、*SNMP* コミュニティ、コンテキスト名など)	SNMP インターフェース（v1、v2、v3）の設定に関する詳細は、SNMP 監視の設定を参照してください。 これらのフィールドには、アイテムのホストインターフェースから値が自動的に入力されます。
プロキシ	ホストがプロキシによって監視されている場合は、プロキシを指定します。 このフィールドには、ホストのプロキシ（存在する場合）が自動的に入力されます。
値	ホストから取得した値。 パラメータフィールド内、または表示/編集ボタン  をクリックすると、値またはコードブロックのテキストエリアウィンドウが開きます。 値は最大 512KB までに切り詰められます。これは Web インターフェースでのみ行われます。結果が切り詰められた場合は、警告アイコンが表示されます。警告の説明は、マウスオーバー時に表示されます。512KB を超えるデータも、Zabbix サーバーによって完全に処理されることに注意してください。

その他のパラメータについては、上記の仮の値をテストするを参照してください。

前処理の詳細

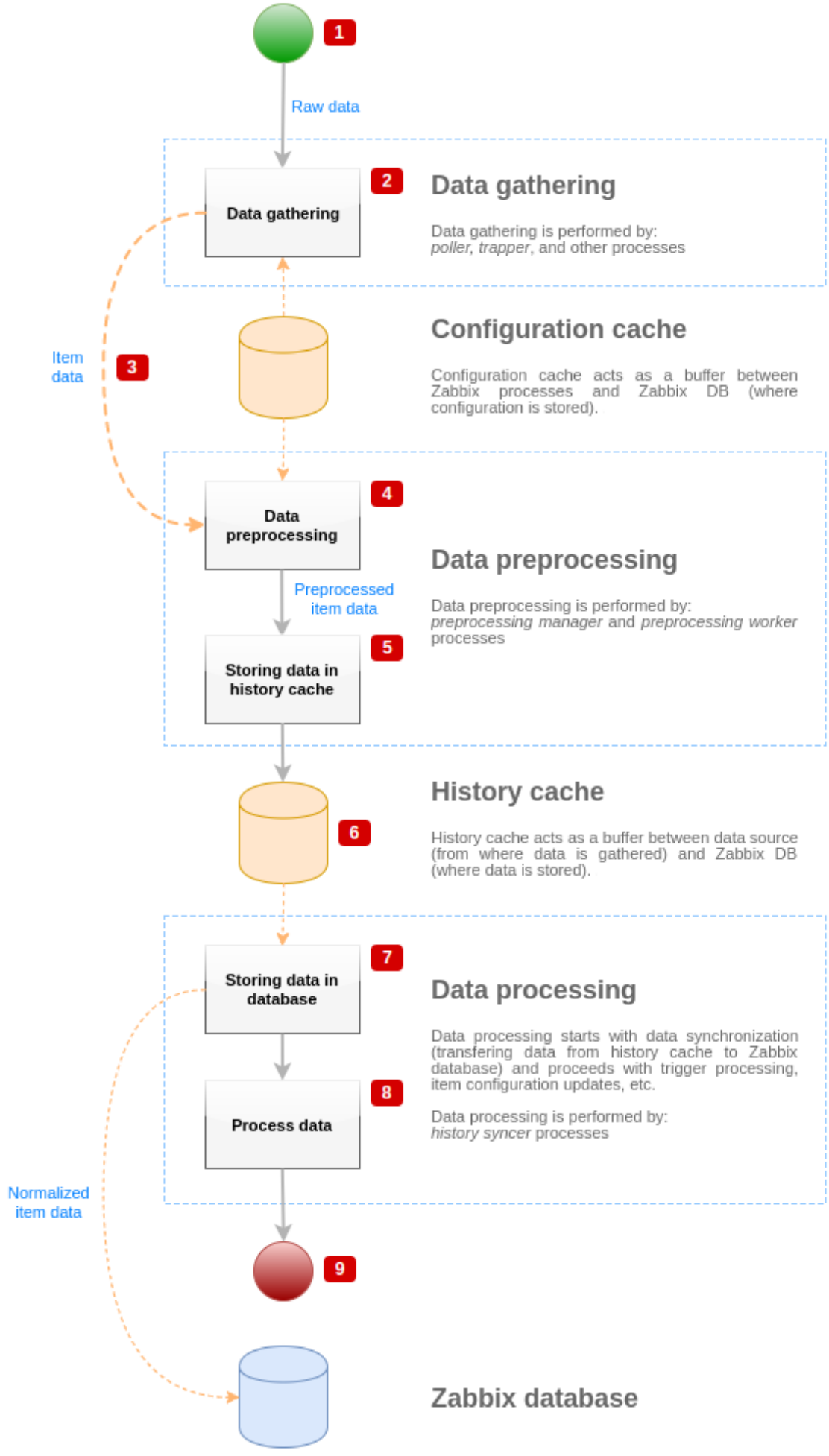
概要

このセクションでは、アイテムの値の前処理について詳しく説明します。アイテムの値の前処理を使用すると、受信したアイテムの値に対する変換ルールを定義して実行できます。

前処理は、前処理の手順を実行する前処理ワーカープロセスとともに、前処理マネージャープロセスによって管理されます。異なるデータ収集プロセスから受信した前処理対象のすべての値は、ヒストリキャッシュに追加される前に前処理マネージャーを通過します。データ収集プロセス（ポーラー、トラッパーなど）と前処理プロセスの間では、ソケットベースの IPC 通信が使用されます。前処理の手順は、Zabbix サーバーまたは Zabbix プロキシ（プロキシによって監視されるアイテムの場合）が実行します。

アイテム値の処理

データソースから Zabbix データベースへのデータフローを視覚化するために、以下の簡略化した図を使用できます。


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

上記の図は、簡略化された形でアイテム値の処理に関連するプロセス、オブジェクト、およびアクションのみを示しています。この図には、条件による方向変更、エラー処理、ループは表示されていません。また、前処理マネージャのローカルデータキャッシュも、データフローに直接影響しないため表示されていません。この図の目的は、アイテム値の処理に関与するプロセスと、それらがどのように相互作用するかを示すことです。

- データ収集は、データソースからの生データから始まります。この時点では、データには ID、タイムスタンプ、値（複数の値も可）のみが含まれています。
- どのタイプのデータ収集プロセスが使用されていても、アクティブチェックやパッシブチェック、トラッパーアイテムなど、考え方は同じです。違いはデータフォーマットと通信の開始者（データ収集プロセスが接続とデータを待つか、通信を開始してデータを要求するか）だけです。生データは検証され、アイテムの設定が設定キャッシュから取得されます（データは設定データで拡張されます）。
- データ収集プロセスから前処理マネージャへデータを渡すために、ソケットベースの IPC メカニズムが使用されます。この時点で、データ収集プロセスは前処理マネージャからの応答を待たずにデータ収集を続行します。
- データの前処理が実行されます。これには、前処理ステップの実行や従属アイテムの処理が含まれます。

Note:
前処理の実行中に、いずれかの前処理ステップが失敗した場合、アイテムは NOT SUPPORTED 状態に変更されることがあります。

- 前処理マネージャのローカルデータキャッシュからヒストリキャッシュへヒストリデータがフラッシュされます。
- この時点で、次のヒストリキャッシュの同期（ヒストリシンカープロセスがデータ同期を実行するタイミング）までデータフローは停止します。
- 同期プロセスは、Zabbix データベースにデータを保存する前にデータの正規化から始まります。データの正規化では、アイテム設定で定義されたタイプへの変換や、定義済みサイズ（文字列の場合は HISTORY_STR_VALUE_LEN、テキストの場合は HISTORY_TEXT_VALUE_LEN、ログ値の場合は HISTORY_LOG_VALUE_LEN）に基づくテキストデータの切り捨てなどが行われます。正規化が完了した後、データは Zabbix データベースに送信されます。

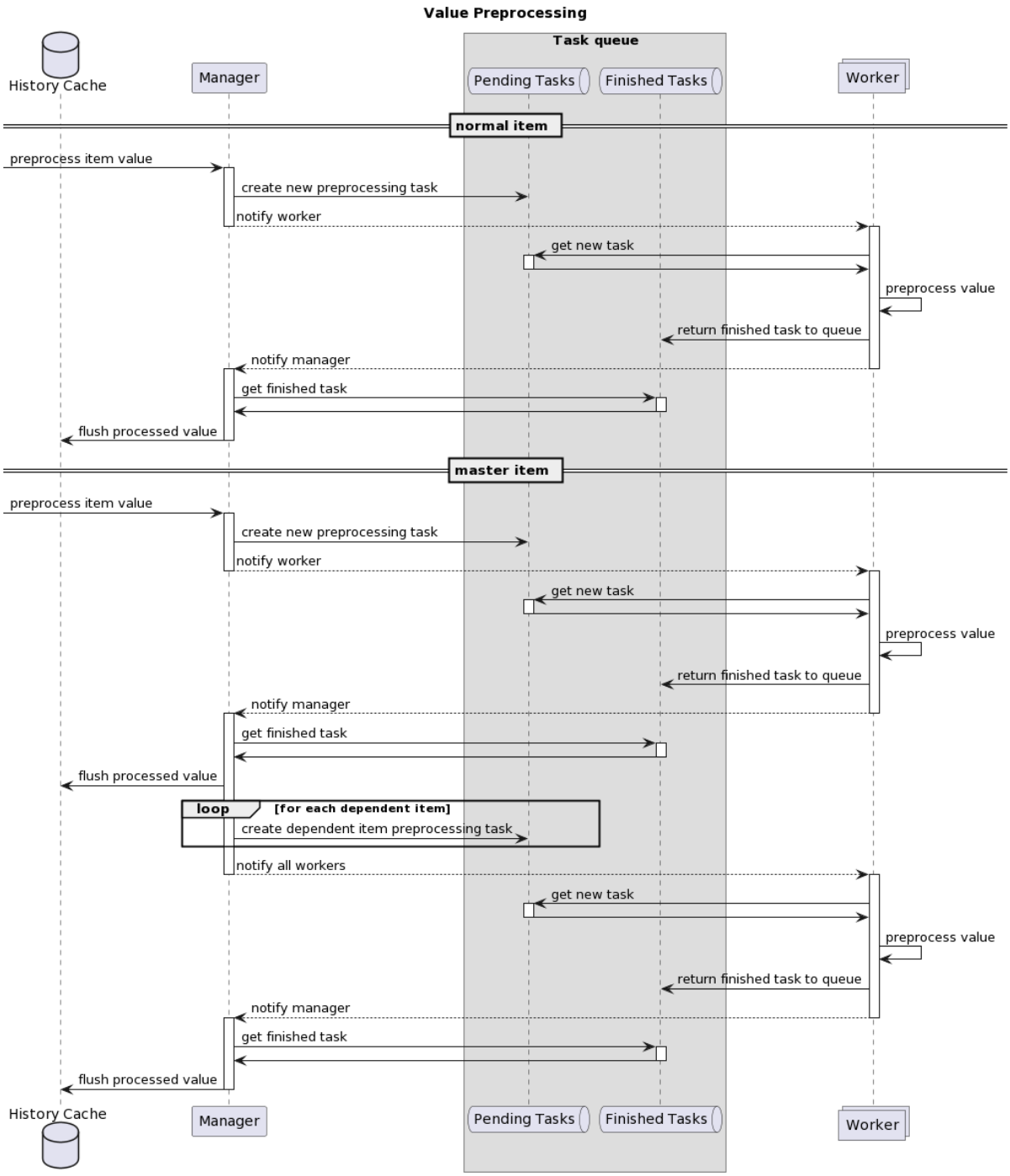
Note:
データの正規化に失敗した場合（たとえば、テキスト値を数値に変換できない場合など）、アイテムは NOT SUPPORTED 状態に変更されることがあります。

- 収集されたデータが処理されます。トリガーのチェック、アイテムが NOT SUPPORTED になった場合のアイテム設定の更新などが行われます。
- これは、アイテム値の処理の観点から見たデータフローの終了と見なされます。

アイテム値の事前処理

データの事前処理は以下の手順で実行されます。

- アイテムに事前処理も従属アイテムもない場合、その値はヒストリキャッシュに追加されるか、LLD マネージャに送信されます。そうでない場合、アイテム値は UNIX ソケットベースの IPC メカニズムを使用して事前処理マネージャに渡されます。
- 事前処理タスクが作成され、キューに追加され、事前処理ワーカーに新しいタスクが通知されます。
- この時点で、少なくとも 1 つの空いている（すなわち、タスクを実行していない）事前処理ワーカーが現れるまでデータフローは停止します。
- 事前処理ワーカーが利用可能になると、キューから次のタスクを取得します。
- 事前処理が完了すると（事前処理ステップの失敗・成功の両方の場合）、事前処理済みの値が完了タスクキューに追加され、マネージャに新しい完了タスクが通知されます。
- 事前処理マネージャは、結果を希望するフォーマット（アイテム値タイプで定義）に変換し、ヒストリキャッシュに追加するか、LLD マネージャに送信します。
- 処理されたアイテムに従属アイテムがある場合、従属アイテムは事前処理済みのマスターアイテム値で事前処理キューに追加されます。従属アイテムは通常の値の事前処理リクエストをバイパスしてキューに追加されますが、値が設定されていて NOT SUPPORTED 状態でないマスターアイテムに対してのみ行われます。



図では、事前処理のキャッシュを省略することで、マスターアイテムの事前処理がやや簡略化されていることに注意してください。

前処理キュー

前処理キューは次のように構成されています:

- 保留中タスクのリスト:
  - 受信順に値の前処理リクエストから直接作成されたタスク
- 即時タスクのリスト (保留中タスクより先に処理される):
  - テストタスク (フロントエンドからのアイテム/前処理テストリクエストに応じて作成)
  - 従属アイテムのタスク
  - シーケンスタスク (厳密な順序で実行する必要があるタスク):
    * 最終値を使用する前処理ステップを持つもの:
      · 差分
      · スロットリング
      · JavaScript (バイトコードキャッシュ)
    * 従属アイテムの前処理キャッシュ
- 完了したタスクのリスト

前処理キャッシュ

前処理キャッシュは、同様の前処理ステップを持つ複数の依存アイテム (これは一般的な LLD の結果です) の前処理パフォーマンスを向上させるために導入されました。

キャッシュは、1 つの依存アイテムを前処理し、内部の前処理データの一部を残りの依存アイテムで再利用することで行われます。前処理キャッシュは、以下のタイプの最初の前処理ステップのみサポートされています:

- Prometheus パターン (メトリクスで入力をインデックス化)
- JSONPath(データをオブジェクトツリーに解析し、最初の式 `[?(@.path == "value")]` をインデックス化)

プリプロセッシングワーカー

Zabbix サーバーの設定ファイルでは、プリプロセッシングワーカースレッドの数を設定できます。StartPreprocessors 設定パラメータを使用して、事前起動するプリプロセッシングワーカーのインスタンス数を設定してください。この数は、少なくとも利用可能な CPU コア数と同じである必要があります。

プリプロセッシングタスクが CPU に依存せず、頻繁なネットワークリクエストを伴う場合は、追加のワーカーを設定することを推奨します。最適なプリプロセッシングワーカー数は、”preprocessable” なアイテム数（何らかのプリプロセッシング手順の実行が必要なアイテム)、データ収集プロセス数、アイテムのプリプロセッシングにおける平均ステップ数など、さまざまな要因によって決まります。ワーカー数が不足すると、メモリ使用量が増加する可能性があります。Zabbix 環境でメモリ使用量が過剰になっている場合のトラブルシューティングについては、tcmalloc を使用した過剰なメモリ使用量のプロファイリング を参照してください。

ただし、大きな XML/JSON チャンクの解析のような負荷の高いプリプロセッシング処理がない場合は、プリプロセッシングワーカーの数をデータ収集プロセスの合計数に合わせることができます。こうすることで、ほとんどの場合（データ収集側からのデータが一括で到着する場合を除く)、収集済みデータを処理するための未使用のプリプロセッシングワーカーが少なくとも 1 つ存在する状態になります。

Warning:
poller、unreachable poller、ODBC poller、HTTP poller、Java poller、pinger、trapper、proxypoller といったデータ収集プロセスが多すぎると、IPMI manager、SNMP trapper、およびプリプロセッシングワーカーと合わせて、プリプロセッシングマネージャーのプロセスごとのファイルディスクリプタ上限を使い切る可能性があります。

 プロセスごとのファイルディスクリプタ上限を使い切ると、Zabbix サーバーは停止します。通常は起動直後に発生しますが、場合によってはより時間がかかることもあります。このような問題を回避するには、Zabbix サーバーの設定ファイル を確認し、同時チェック数とプロセス数を最適化してください。さらに、必要に応じてシステムの制限を確認・調整し、ファイルディスクリプタ上限が十分に高く設定されていることを確認してください。

値処理パイプライン

アイテムの値の処理は、複数のプロセスによって複数のステップ（またはフェーズ）で実行されます。これにより、以下のようなことが発生する可能性があります。

- 従属アイテムは値を受け取ることができますが、マスター値は受け取れません。これは、次のユースケースを使用することで実現できます。
  - マスターアイテムの値の型は `UINT`（トラッパーアイテムを使用可能)、従属アイテムの値の型は `TEXT`。
  - マスターアイテムと従属アイテムの両方に前処理ステップは不要です。
  - テキスト値（例: ”abc”）をマスターアイテムに渡す必要があります。
  - 実行する前処理ステップがないため、前処理マネージャはマスターアイテムが「サポートされていない」状態でないことと値が設定されていることを確認し（両方とも真)、従属アイテムをマスターアイテムと同じ値でキューに入れます（前処理ステップがないため)。
  - マスターアイテムと従属アイテムの両方が履歴同期フェーズに到達すると、マスターアイテムは値変換エラー（テキストデータを符号なし整数に変換できないため）により「サポートされていない」状態になります。

その結果、従属アイテムは値を受け取りますが、マスターアイテムは「サポートされていない」状態に変わります。

- 従属アイテムが、マスターアイテムの履歴に存在しない値を受け取る場合があります。ユースケースは前述のものと非常に似ていますが、マスターアイテムの型が異なります。たとえば、マスターアイテムに `CHAR` 型を使用した場合、マスターアイテムの値は履歴同期フェーズで切り捨てられますが、従属アイテムはマスターアイテムの初期値（切り捨てられていない値）から値を受け取ります。

前処理の例

概要

このセクションでは、保存前処理のステップを使用していくつかの実用的なタスクを実行する例を紹介します。

VMware イベントログレコードのフィルタリング

この例では、正規表現に一致の前処理ステップを使用して、VMware イベントログから不要なイベントをフィルタリングします。

1. 稼働中の VMware Hypervisor ホストで、vmware.eventlogアイテムが存在し、正常に動作していることを確認します。
 イベントログアイテムは、ホスト作成時にVMwareテンプレートがリンクされている場合、すでにハイパーバイザー上に存在している可能性があります。

2. VMware Hypervisor ホストで、*Log* タイプの従属アイテムを作成し、イベントログアイテムをマスターアイテムとして設定します。

3. 従属アイテムの前処理タブで、追加をクリックして前処理ステップを作成し、ドロップダウンから正規表現に一致を選択します。その後、以下のいずれかのパターンを指定します。

- すべてのログイベントをフィルタリングする場合:

```
.* logged in .*
```

- "User" の後にユーザー名を含む行をフィルタリングする場合:

```
\bUser\s+\K\S+
```

Attention:
正規表現に一致しない場合、従属アイテムは該当するエラーメッセージとともにサポート対象外になります。
 これを回避するには、失敗時にカスタムチェックボックスをマークし、値を破棄するかカスタム値を設定するなどのオプションを選択してください。
 破棄された値はデータベースに保存されないため、トリガーの評価やトレンドデータの生成は行われませんのでご注意ください。

また、正規表現の前処理ステップを使用して、一致するグループを抽出し、出力を制御することもできます。

- "logged in" を含むログイベント全体を抽出して出力する場合、以下のパラメータを指定します:

```
パターン: .*logged in.*
出力: \0
```

- "User" の後に続くユーザー名を抽出して出力する場合:

```
パターン: User (.*?)(?=\ )
出力: \1
```

取得した値の型のチェック

この例では、カスタム乗数の前処理ステップを使用して、取得したアイテム値の型が数値であるかどうかをチェックします。

アイテムの前処理タブで、カスタム乗数の前処理ステップを選択し、次のパラメータを指定します（取得した値に 1 を掛けます）:

```
1
```

Attention:
前処理が失敗した場合（例：入力が数値でない場合）、アイテムは対応するエラーメッセージとともにサポート対象外になります。これを回避するには、失敗時にカスタムのチェックボックスをマークし、値を破棄する、またはカスタム値を設定するなどのオプションを選択してください。破棄された値はデータベースに保存されないため、トリガーの評価やトレンドデータの生成は行われませんのでご注意ください。

サポートされていない値のチェック

この例では、サポートされていない値のチェックの前処理ステップを使用して、アイテム値が取得できなかったかどうかを確認します。

Zabbix サーバー/プロキシのポーラープロセスがアイテム値の収集を試みると、次のいずれかが発生します。

- 有効な結果を返す。
- 一見有効な結果を返すが、後でサポートされなくなる場合がある（例：前処理後に値の型が一致しない場合）。
- 値の収集エラーを返し、アイテムがサポートされなくなる。一般的な原因は次のとおりです。
  - 不明なアイテムキー（Zabbix エージェント、シンプルチェック、Zabbix 内部アイテムの場合）
  - 不明な OID（SNMP エージェント）、不明なセンサー（IPMI エージェント）、JMX メトリックなし（JMX エージェント）
  - トラップファイルが読み取れない（SNMP トラップ）
  - スクリプトが見つからない（外部チェック）
  - そのような URL が存在しない（HTTP エージェント、ブラウザー）
  - ログイン失敗（SSH エージェント、TELNET エージェント）
  - 無効な式の構文（計算アイテム）、JavaScript 構文エラー（スクリプト）、無効な SQL（データベース監視）

アイテム値の収集エラーを検出して処理するには、サポートされていない値のチェックの前処理ステップを使用できます。このステップは常に最初に実行され、前処理が開始される前に発生したエラーのみを検出することに注意してください。

アイテムの前処理タブで、サポートされていない値のチェックの前処理ステップを選択し、次のいずれかのパラメーターを指定します。

- すべてのエラーの場合:

```
パラメーター: any error
```

- ”cannot connect” を含むエラーの場合:

```
パラメーター: error matches
パターン: (?i)cannot connect
```

次に、失敗時のカスタムオプションを使用して、値（この場合はエラー）を破棄したり、カスタム値を設定したり、カスタムエラーメッセージを返したりできます。破棄された値はデータベースに保存されないため、トリガーは評価されず、トレンドデータも生成されないことに注意してください。

JSONPath の機能

概要

このセクションでは、アイテム値の事前処理ステップでサポートされている JSONPath の機能について説明します。

JSONPath は、ドットで区切られたセグメントで構成されます。セグメントは、JSON 値名を表す単純な単語、ワイルドカード文字 (*)、または角括弧で囲まれたより複雑な構造のいずれかの形式を取ることができます。角括弧で囲まれたセグメントの前のドットは省略可能で、省略することができます。

JSONPath の例	説明
`$.object.name`	`object.name` の内容を返します。
`$.object['name']`	`object.name` の内容を返します。
`$.object.['name']`	`object.name` の内容を返します。
`$["object"]['name']`	`object.name` の内容を返します。
`$.['object'].["name"]`	`object.name` の内容を返します。
`$.object.history.length()`	`object.history` 配列の要素数を返します。
`$[?(@.name == 'Object')].price.first()`	名前が”Object” の最初のオブジェクトの `price` プロパティの値を返します。
`$[?(@.name == 'Object')].history.first().length()`	名前が”Object” の最初のオブジェクトの history 配列の要素数を返します。
`$[?(@.price > 10)].length()`	価格が 10 より大きいオブジェクトの数を返します。

参考: JSONPath での LLD マクロ値の特殊文字のエスケープ

サポートされているセグメント

セグメント	説明
`<name>`	名前でオブジェクトプロパティに一致します。
`*`	すべてのオブジェクトプロパティに一致します。
`['<name>']`	名前でオブジェクトプロパティに一致します。
`['<name>', '<name>', ...]`	リストされた名前のいずれかに一致するオブジェクトプロパティに一致します。
`[<index>]`	インデックスで配列要素に一致します。
`[<number>, <number>, ...]`	リストされたインデックスのいずれかに一致する配列要素に一致します。
`[*]`	すべてのオブジェクトプロパティまたは配列要素に一致します。
`[<start>:<end>]`	定義された範囲で配列要素に一致します: **<start>** - 一致する最初のインデックス (含む); 指定しない場合は配列の先頭からすべての配列要素に一致; 負の場合は配列の末尾からの開始オフセットを指定; **<end>** - 一致する最後のインデックス (含まない); 指定しない場合は配列の末尾までのすべての配列要素に一致; 負の場合は配列の末尾からの開始オフセットを指定。
`[?(<expression>)]`	フィルター式を適用してオブジェクト/配列要素に一致します。

祖先を無視して一致するセグメント (切り離されたセグメント) を見つけるには、2 つのドット (`..`) をプレフィックスとして付ける必要があります。たとえば、`$..name` や `$..['name']` は、すべての `name` プロパティの値を返します。

一致した要素名は、JSONPath にチルダ (~) サフィックスを追加することで抽出できます。これにより、一致したオブジェクトの名前または一致した配列アイテムのインデックス (文字列形式) が返されます。出力形式は他の JSONPath クエリと同じルールに従います - 明確なパスの結果は「そのまま」返され、不明確なパスの結果は配列で返されます。ただし、明確なパスに一致する要素の名前を抽出することには最小限の価値しかありません。なぜなら、それはすでに分かっているからです。

フィルター式

フィルター式は、中置記法による算術式です。

サポートされるオペランド:

Operand	Description

`"<text>"`
`'<text>'`
	テキスト定数。
	
	例:
	`'value: \'1\''`
	`"value: '1'"`

`<number>`	科学表記法をサポートする数値定数。 例: 123
`<jsonpath starting with $>`	入力ドキュメントのルートノードから JSONPath により参照される値。確定パスのみサポートされます。 例: `$.object.name`
`<jsonpath starting with @>`	現在のオブジェクト/要素から JSONPath により参照される値。確定パスのみサポートされます。 例: `@.name`

サポートされる演算子: |Operator|Type|Description|Result| |-|---|---|---| |−|Binary| 減算 |Number| |+|Binary| 加算 |Number| |/|Binary| 除算 |Number| |*|Binary| 乗算 |Number| |==|Binary| 等価 |Boolean (1/0)| |!=|Binary| 不等価 |Boolean (1/0)| |<|Binary| より小さい |Boolean (1/0)| |<=|Binary| 以下 |Boolean (1/0)| |>|Binary| より大きい |Boolean (1/0)| |>=|Binary| 以上 |Boolean (1/0)| |=~|Binary| 正規表現に一致 |Boolean (1/0)| |!|Unary| ブール NOT|Boolean (1/0)| |||Binary| ブール OR|Boolean (1/0)| |&&|Binary| ブール AND|Boolean (1/0)|

関数

関数は JSONPath の末尾で使用できます。複数の関数を連結することができ、前の関数が後続の関数で受け入れられる値を返す場合に有効です。

サポートされている関数:

関数	説明	入力	出力
`avg`	入力配列内の数値の平均値	数値の配列	数値
`min`	入力配列内の数値の最小値	数値の配列	数値
`max`	入力配列内の数値の最大値	数値の配列	数値
`sum`	入力配列内の数値の合計	数値の配列	数値
`length`	入力配列内の要素数	配列	数値
`first`	配列の最初の要素	配列	入力配列の内容に応じた JSON 構造（オブジェクト、配列、値）

JSONPath の集約関数は、引用符付きの数値も受け入れます。これらの値は、集約が必要な場合に自動的に文字列から数値型に変換されます。互換性のない入力は、関数がエラーを生成する原因となります。

出力値

JSONPath は、確定パスと不確定パスに分けることができます。確定パスは、null または単一の一致のみを返すことができます。不確定パスは、複数の一致を返すことができます。これは、分離された複数の名前/インデックスリスト、配列スライス、または式セグメントを持つ JSONPath です。ただし、関数が使用される場合、関数は常に単一の値を出力するため、JSONPath は確定パスになります。

確定パスは、参照しているオブジェクト/配列/値を返します。対照的に、不確定パスは一致したオブジェクト/配列/値の配列を返します。

Attention:
内部の最適化手法により、JSONPath クエリ結果のプロパティ順序が元の JSON のプロパティ順序と一致しない場合があります。たとえば、JSONPath `$.books[1]["author", "title"]` は `["title", "author"]` を返す場合があります。元のプロパティ順序を保持することが重要な場合は、クエリ後の処理方法を検討してください。

パスの書式設定ルール

ブラケット表記のセグメントや式には、空白（スペース、タブ文字）を使用できます。例：`$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`。

文字列はシングルクォート（'）またはダブルクォート（"）で囲む必要があります。文字列内では、シングルクォートまたはダブルクォート（どちらで囲むかによる）およびバックスラッシュ（\）は、バックスラッシュ（\）文字でエスケープされます。

例


```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": " 地元の町への翌日配送",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "A5 形式での印刷と製本",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": " 各種修復方法",
```

```
      "active": false,
      "methods": [
        {
          "description": " 化学洗浄",
          "price": 46
        },
        {
          "description": " 湿気で損傷したページのプレス",
          "price": 24.5
        },
        {
          "description": " 破れた本の再製本",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": " 店舗は閉店しています",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	no ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }
`$.books[1].title`	definite	Sword of Honour
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sayings of the Century”]
`$.books[1]['author', "title"]`	indefinite	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] 注: このクエリは、算術演算をクエリ内で使用できることを示しています。`$.books[?(@.id == 2)].title` に簡略化できます。
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]

JSONPath	Type	Result
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”A5 形式での印刷と製本”, ” 破れた本の再製本”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour
`$..tags.first().length()`	definite	5 注: `$..tags` は不定パスであるため、一致した要素の配列、つまり `[["a", "b", "c", "d", "e" ]]` を返します。`first()` は最初の要素、つまり `["a", "b", "c", "d", "e"]` を返します。`length()` は要素の長さ、つまり 5 を計算します。
`$.books[*].price.min()`	definite	8.95

JSONPath	Type	Result
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	注: 一致しないクエリは、definite パスと indefinite パスの両方で NULL を返します。
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000]
`$.services[?(@.active=="false")].servicegroup`	indefinite	注: ブール値の比較では、テキスト定数を使用する必要があります。 [1002]
`$.services[?(@.servicegroup=="1002")].first()`	definite	注: ブール値の比較では、テキスト定数を使用する必要があります。 {"servicegroup":1002,"description":"Description 2"}

JSONPath での LLD マクロ値における特殊文字のエスケープ

低レベルディスカバリのマクロが JSONPath の前処理で使用され、その値が解決されると、次の特殊文字のエスケープ規則が適用されます。

- エスケープ対象として考慮されるのは、バックスラッシュ (\) と二重引用符 (") のみです。
- 解決されたマクロ値にこれらの文字が含まれている場合、それぞれにバックスラッシュを付けてエスケープします。
- すでにバックスラッシュでエスケープされている場合でも、それはエスケープ済みとは見なされず、バックスラッシュとその後ろの特殊文字の両方が再度エスケープされます。

例:

JSONPath	LLD macro value	After substitution
`$.[?(@.value == "{#MACRO}")]`	special "value"	`$.[?(@.value == "special \"value\"")]`
	c:\temp	`$.[?(@.value == "c:\\temp")]`
	a\\b	`$.[?(@.value == "a\\\\b")]`

式で使用する場合、特殊文字を含む可能性のあるマクロは二重引用符で囲む必要があります。

JSONPath	LLD macro value	After substitution	Result
`$.[?(@.value == "{#MACRO}")]`	special "value"	`$.[?(@.value == "special \"value\"")]`	OK
`$.[?(@.value == {#MACRO})]`		`$.[?(@.value == special \"value\")]`	**Bad JSONPath expression**

パスで使用する場合、特殊文字を含む可能性のあるマクロは、角括弧 および二重引用符で囲む必要があります。

JSONPath	LLD macro value	After substitution	Result
`$.["{#MACRO}"].value`	c:\temp	`$.["c:\\temp"].value`	OK
`$.{#MACRO}.value`		`$.c:\\temp.value`	**Bad JSONPath expression**

JavaScript の前処理

概要

このセクションでは、JavaScript による前処理の詳細について説明します。

Attention:
前処理の JavaScript で宣言されていない代入を使用しないでください。ローカル変数を宣言するには `var` を使用してください。

JavaScript 前処理

JavaScript 前処理は、単一のパラメーター value とユーザーが指定した関数本体を持つ JavaScript 関数を呼び出すことで実行されます。
前処理ステップの結果は、この関数が返す値です。たとえば、華氏から摂氏への変換を行うには、次を入力します。

```
return (value - 32) * 5 / 9
```

これを JavaScript 前処理パラメーターに指定すると、サーバーによって次のような JavaScript 関数にラップされます。

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

入力パラメーター value は常に文字列として渡されます。
戻り値は toString() メソッドによって自動的に文字列へ変換されます（失敗した場合は、エラーが文字列値として返されます）。ただし、いくつか例外があります。

- undefined を返すとエラーになります。
- null を返すと、入力値は破棄されます。これは Custom on fail アクションにおける Discard value 前処理と同様です。

エラーは、値やオブジェクトを throw することで返せます（通常は文字列または Error オブジェクトです）。

例:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

各スクリプトの実行タイムアウトは 10 秒です（スクリプトによっては、タイムアウトが発生するまでにそれ以上かかる場合があります）。これを超えるとエラーが返されます。
512 メガバイトのヒープ制限が適用されます。

JavaScript 前処理ステップのバイトコードはキャッシュされ、次回そのステップが適用される際に再利用されます。
アイテムの前処理ステップに変更を加えると、キャッシュされたスクリプトはリセットされ、後で再コンパイルされます。

連続した実行失敗が 3 回続くと、1 つのスクリプトが次のスクリプトの実行環境を壊してしまう可能性を軽減するため、エンジンが再初期化されます（この動作は DebugLevel 4 以上でログに記録されます）。

JavaScript 前処理は Duktape JavaScript エンジンで実装されています。

> **Note:**
> 組み込みの JavaScript オブジェクトのメソッド定義は、アイテム前処理スクリプト内で再定義できません。

参照: 追加の JavaScript オブジェクトとグローバル関数

スクリプトでのマクロの使用

JavaScript コードでユーザーマクロ（および低レベルディスカバリのコンテキストでLLD マクロ）を使用することができます。スクリプトにユーザーマクロが含まれている場合、これらのマクロは特定の前処理ステップを実行する前にサーバー/プロキシによって解決されます。Web インターフェースで前処理ステップをテストする場合、マクロ値は取得されず、手動で入力する必要があることに注意してください。

> **Note:**
> マクロが値に置き換えられる際、コンテキストは無視されます。マクロ値はそのままコードに挿入され、JavaScript コードに値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があることにご注意ください。

以下の例では、受信値が {$THRESHOLD} マクロ値を超えた場合、しきい値（存在する場合）が返されます:

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

例

次の例は、JavaScript の保存前処理の使用方法を示しています。各例には、簡単な説明、JavaScript 保存前処理パラメーターの関数本体、および保存前処理のステップの結果 (関数によって返される値) が含まれています。

例 1: 数値の変換 (科学的記数法から整数へ)

数値”2.62128e+07” を科学的記数法から整数に変換します。

```
return (Number(value))
```

関数によって返される値: 26212800

例 2: 数値の変換 (2 進数から 10 進数へ)

2 進数"11010010" を 10 進数に変換します。

```
return(parseInt(value,2))
```

関数によって返される値: 210

例 3: 数値を丸める

数値"18.2345" を 2 桁に丸めます。

```
return(Math.round(value* 100) / 100)
```

関数によって返される値: 18.23

例 4: 文字列内の文字数を数える

文字列"Zabbix" 内の文字数を数えます。

```
return (value.length)
```

関数によって返される値: 6

例 5: 残り時間を取得する

証明書の有効期限 (Feb 12 12:33:56 2022 GMT) までの残り時間 (秒単位) を取得します。

```
var split = value.split(' '),
    MONTHS_LIST = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
    ISOdate = split[3] + '-' + month_index + '-' + split[1] + 'T' + split[2],
    now = Date.now();

return parseInt((Date.parse(ISOdate) - now) / 1000);
```

関数によって返される値: 44380233

例 6: JSON プロパティの削除

キー"data_size" または"index_size" を持つプロパティを削除して、JSON データ構造を変更します。

```
var obj=JSON.parse(value);

for (i = 0; i < Object.keys(obj).length; i++) {
    delete obj[i]["data_size"];
    delete obj[i]["index_size"];
}

return JSON.stringify(obj)
```

関数によって受け入れられる値:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

関数によって返される値:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

例 7: Apache ステータスを JSON に変換する

web.page.get Zabbix エージェントアイテムから受信した値（例: web.page.get[http://127.0.0.1:80/server-status?auto]）を JSON オブジェクトに変換します。

```
// Apache ステータスを JSON に変換

// 値を部分文字列に分割し、それらの部分文字列を配列に格納
var lines = value.split('\n');

// 空のオブジェクト "output" を作成
var output = {};

// 定義済みプロパティを持つオブジェクト "workers" を作成
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// "lines" 配列の部分文字列をプロパティ（キーと値のペア）として "output" オブジェクトに追加
for (var i = 0; i < lines.length; i++) {
    var line = lines[i].match(/([A-z0-9 ]+): (.*)/);

    if (line !== null) {
        output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);
    }
}

// 複数バージョン対応メトリクス
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// ワーカー数を取得するために "Scoreboard" プロパティを解析
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// ワーカーデータを "output" オブジェクトに追加
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// JSON 文字列を返す
return JSON.stringify(output);
```

関数が受け付ける値:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 11:08:39 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 405
Content-Type: text/plain; charset=ISO-8859-1

127.0.0.1
ServerVersion: Apache/2.4.52 (Ubuntu)
ServerMPM: prefork
Server Built: 2023-03-08T17:32:01
CurrentTime: Monday, 27-Mar-2023 14:08:39 EEST
RestartTime: Monday, 27-Mar-2023 12:19:59 EEST
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 6520
ServerUptime: 1 hour 48 minutes 40 seconds
Load1: 0.56
Load5: 0.33
Load15: 0.28
Total Accesses: 2476
Total kBytes: 8370
Total Duration: 52718
CPUUser: 8.16
CPUSystem: 3.44
CPUChildrenUser: 0
CPUChildrenSystem: 0
CPULoad: .177914
Uptime: 6520
ReqPerSec: .379755
BytesPerSec: 3461.58
BytesPerReq: 3461.58
DurationPerReq: 21.2916
BusyWorkers: 2
IdleWorkers: 6
Scoreboard: ____KW__.....................................................................................
```

関数が返す値:


```
{
    "Date": "Mon, 27 Mar 2023 11:08:39 GMT",
    "Server": "Apache/2.4.52 (Ubuntu)",
    "Vary": "Accept-Encoding",
    "Encoding": "gzip",
    "Length": 405,
    "Type": "text/plain; charset=ISO-8859-1",
    "ServerVersion": "Apache/2.4.52 (Ubuntu)",
    "ServerMPM": "prefork",
    "Server Built": "2023-03-08T17:32:01",
    "CurrentTime": "Monday, 27-Mar-2023 14:08:39 EEST",
    "RestartTime": "Monday, 27-Mar-2023 12:19:59 EEST",
    "ParentServerConfigGeneration": 1,
    "ParentServerMPMGeneration": 0,
    "ServerUptimeSeconds": 6520,
    "ServerUptime": "1 hour 48 minutes 40 seconds",
    "Load1": 0.56,
    "Load5": 0.33,
    "Load15": 0.28,
    "Total Accesses": 2476,
    "Total kBytes": 8370,
    "Total Duration": 52718,
    "CPUUser": 8.16,
```

```
    "CPUSystem": 3.44,
    "CPUChildrenUser": 0,
    "CPUChildrenSystem": 0,
    "CPULoad": 0.177914,
    "Uptime": 6520,
    "ReqPerSec": 0.379755,
    "BytesPerSec": 1314.55,
    "BytesPerReq": 3461.58,
    "DurationPerReq": 21.2916,
    "BusyWorkers": 2,
    "IdleWorkers": 6,
    "Scoreboard": "____KW__.....................................................................
    "Workers": {
        "waiting": 6,
        "starting": 0,
        "reading": 0,
        "sending": 1,
        "keepalive": 1,
        "dnslookup": 0,
        "closing": 0,
        "logging": 0,
        "finishing": 0,
        "cleanup": 0,
        "slot": 142
    }
}
```

追加の JavaScript オブジェクト

概要

このセクションでは、Duktape で実装された JavaScript 言語への Zabbix の追加機能と、サポートされているグローバル JavaScript 関数について説明します。

Attention:
プリプロセスの JavaScript では、宣言されていない代入を使用しないでください。ローカル変数を宣言するには `var` を使用してください。

組込オブジェクト

Zabbix

Zabbix オブジェクトは、Zabbix の内部機能との連携を提供します。

メソッド	説明
`log(loglevel, message)`	<loglevel> のログレベル（設定ファイルの DebugLevel パラメータを参照）で <message> を Zabbix ログに書き込みます。

例:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

以下のエイリアスを使用できます:

エイリアス	エイリアス先
console.log(object)	Zabbix.log(4, JSON.stringify(object))
console.warn(object)	Zabbix.log(3, JSON.stringify(object))
console.error(object)	Zabbix.log(2, JSON.stringify(object))

Attention:
スクリプト実行ごとに記録されるすべてのメッセージの合計サイズは 8MB に制限されています。

メソッド	説明
`sleep(delay)`	JavaScript の実行を `delay` ミリ秒遅延させます。

例（15 秒遅延）：

```
Zabbix.sleep(15000)
```

HttpRequest

このオブジェクトは cURL ハンドルをカプセル化し、簡単な HTTP リクエストを実行できるようにします。エラーは例外としてスローされます。

Attention:
1 回のスクリプト実行で初期化できる `HttpRequest` オブジェクトは最大 10 個に制限されています。

Method	Description
`addHeader(value)`	HTTP ヘッダーフィールドを追加します。このフィールドは、`clearHeader()` メソッドでクリアされるまで、後続のすべてのリクエストで使用されます。 1 つの `HttpRequest` オブジェクトに追加できるヘッダーフィールドの合計長は、128 Kbytes（特殊文字とヘッダー名を含む）に制限されています。
`clearHeader()`	HTTP ヘッダーをクリアします。ヘッダーフィールドが設定されていない場合、投稿するデータが JSON 形式であれば `HttpRequest` は Content-Type に application/json を設定し、それ以外の場合は text/plain を設定します。
`connect(url)`	URL に HTTP CONNECT リクエストを送信し、レスポンスを返します。
`customRequest(method, url, data)`	最初のパラメータで任意の HTTP メソッドを指定できます。オプションの *data* ペイロードとともに指定されたメソッドのリクエストを URL に送信し、レスポンスを返します。
`delete(url, data)`	オプションの *data* ペイロードとともに HTTP DELETE リクエストを URL に送信し、レスポンスを返します。
`getHeaders(<asArray>)`	受信した HTTP ヘッダーフィールドのオブジェクトを返します。 `asArray` パラメータには”true”（例: `getHeaders(true)`）、”false”、または未定義を設定できます。”true” に設定した場合、受信した HTTP ヘッダーフィールドの値は配列として返されます。同じ名前の複数のヘッダーのフィールド値を取得する場合に使用してください。 設定しない場合、または”false” に設定した場合、受信した HTTP ヘッダーフィールドの値は文字列として返されます。
`get(url, data)`	オプションの *data* ペイロードとともに HTTP GET リクエストを URL に送信し、レスポンスを返します。
`head(url)`	URL に HTTP HEAD リクエストを送信し、レスポンスを返します。
`options(url)`	URL に HTTP OPTIONS リクエストを送信し、レスポンスを返します。
`patch(url, data)`	オプションの *data* ペイロードとともに HTTP PATCH リクエストを URL に送信し、レスポンスを返します。
`put(url, data)`	オプションの *data* ペイロードとともに HTTP PUT リクエストを URL に送信し、レスポンスを返します。
`post(url, data)`	オプションの *data* ペイロードとともに HTTP POST リクエストを URL に送信し、レスポンスを返します。
`getStatus()`	最後の HTTP リクエストのステータスコードを返します。
`setProxy(proxy)`	HTTP プロキシを”proxy” の値に設定します。このパラメータが空の場合、プロキシは使用されません。

`setHttpAuth(bitmask, username, password)`	`bitmask` パラメータで有効にする HTTP 認証方式（HTTPAUTH_BASIC、HTTPAUTH_DIGEST、HTTPAUTH_NEGOTIATE、HTTPAUTH_NTLM、HTTPAUTH_NONE）を設定します。
	HTTPAUTH_NONE フラグを使用すると、HTTP 認証を無効にできます。
	例:
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	オプションの *data* ペイロードとともに HTTP TRACE リクエストを URL に送信し、レスポンスを返します。

例:

```
try {
    Zabbix.log(4, 'jira webhook script value='+value);

    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new HttpRequest(),
    fields = {},
    resp;

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic '+params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
    fields.issuetype = {"id": params.issue_id};
    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: '+req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
} catch (error) {
    Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
    Zabbix.log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);
```

XML

XML オブジェクトは、アイテムおよびローレベルディスカバリの前処理や Webhook で XML データの処理を可能にします。

Attention:
XML オブジェクトを使用するには、サーバー/プロキシを libxml2 サポート付きでコンパイルする必要があります。

メソッド	説明
XML.query(data, expression)	XPath を使用してノードの内容を取得します。ノードが見つからない場合は null を返します。 **expression** - XPath 式; **data** - 文字列としての XML データ。
XML.toJson(data)	XML 形式のデータを JSON に変換します。
XML.fromJson(object)	JSON 形式のデータを XML に変換します。

例:

入力:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
```

```
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

出力:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

シリアライズルール

XML から JSON への変換は、以下のルールに従って処理されます（JSON から XML への変換では逆のルールが適用されます）。

1. XML 属性は、名前の前に「@」が付いたキーに変換されます。

例:

入力:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

出力:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. セルフクローズ要素（<foo/>）は「null」値として変換されます。

例:

入力:

```
<xml>
  <foo/>
</xml>
```

出力:

```
{
  "xml": {
    "foo": null
  }
}
```

3. 空の属性（値が""）は空文字列（''）値として変換されます。

例:

入力:

```xml
<xml>
  <foo bar="" />
</xml>
```

出力:

```json
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. 同じ要素名の複数の子ノードは、値の配列を持つ単一のキーに変換されます。

例:

入力:

```xml
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

出力:

```json
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. テキスト要素に属性や子要素がない場合、文字列として変換されます。

例:

入力:

```xml
<xml>
    <foo>BAZ</foo>
</xml>
```

出力:

```json
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. テキスト要素に子要素はないが属性がある場合、テキスト内容は「#text」キーと値を持つ要素に変換され、属性はシリアライズルール 1 で説明したように変換されます。

例:

入力:

```xml
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

出力:

```json
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
```

```
    }
  }
}
```

グローバル JavaScript 関数

Duktape で追加のグローバル JavaScript 関数が実装されています。

- btoa(data) - データを base64 文字列にエンコードします。
- atob(base64_string) - base64 文字列を Uint8Array バッファとしてデコードします。

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    // デコードロジックはバッファのデータフォーマットに依存することに注意してください。
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- md5(data) - データの MD5 ハッシュを計算します。
- sha256(data) - データの SHA256 ハッシュを計算します。
- hmac('<hash type>',key,data) - HMAC ハッシュを 16 進数形式の文字列として返します。hash type として md5 と sha256 がサポートされています。key と data パラメータはバイナリデータをサポートします。

  例:

  ```
  - hmac('md5',key,data)
  - hmac('sha256',key,data)
  ```

- sign(hash,key,data) - 計算された署名（SHA-256 による RSA 署名）を文字列として返します。ここで:
 hash - sha256 のみ許可され、それ以外の場合はエラーがスローされます。
 key - 秘密鍵。PKCS#1 または PKCS#8 標準に準拠している必要があります。鍵はさまざまな形式で指定できます。

  ```
  - 改行の代わりにスペースを使用
  - 改行の代わりにエスケープされたまたはエスケープされていない\n を使用
  - 改行なしの 1 行の文字列
  - JSON 形式の文字列
  ```

  鍵はユーザーマクロ/シークレットマクロ/ボールトからロードすることもできます。

  data - 署名されるデータ。文字列（バイナリデータもサポート）またはバッファ（Uint8Array/ArrayBuffer）で指定できます。

  例:

  ```
  - sign('sha256',key,data)
  ```

  署名の計算には OpenSSL または GnuTLS が使用されます。これらの暗号化ライブラリのいずれかが組み込まれていない場合、エラーがスローされます（'missing OpenSSL or GnuTLS library'）。

ブラウザーアイテムの JavaScript オブジェクト

概要

このセクションでは、ブラウザーアイテムスクリプトで使用するために Duktape で実装された JavaScript 言語への Zabbix の追加機能について説明します。これらの追加機能は、追加の JavaScript オブジェクトページで説明されている JavaScript オブジェクトを補完するものです。

Browser

Browser オブジェクトは WebDriver セッションを管理し、作成時にセッションを初期化し、破棄時にセッションを終了します。
1 つのスクリプトで最大 4 つの Browser オブジェクトをサポートできます。

Browser オブジェクトを構築するには、new Browser(options) 構文を使用します。
options (*JSON object*) パラメータはブラウザーオプションを指定し、通常は WebDriver options メソッドの結果です（例: Browser.chromeOptions()）。

Browser オブジェクトでは、以下のメソッドがサポートされています。

Method	Description
navigate(url)	指定した URL に移動します。 パラメータ: url - (string) 移動先の URL。
getUrl()	開いているページの URL 文字列を返します。
getPageSource()	開いているページのソース文字列を返します。
findElement(strategy, selector)	開いているページ内の 1 つの要素を含むElement オブジェクトを返します (strategy と selector に一致する要素がない場合は null を返します)。 パラメータ: strategy - (string, css selector/link text/partial link text/tag name/xpath) Location strategy; selector - (string) 指定したロケーション戦略を使用する要素セレクター。
findElements(strategy, target)	開いているページ内の複数の要素を含むElement オブジェクトの配列を返します（ロケーション戦略と target に一致する要素がない場合は空の配列を返します)。 パラメータ: strategy - (string, css selector/link text/partial link text/tag name/xpath) Location strategy; target - (string) 指定したロケーション戦略を使用する要素セレクター。
getCookies()	Cookie オブジェクトの配列を返します。
addCookie(cookie)	Cookie を設定します。 パラメータ: cookie - (Cookie object) 設定する Cookie。
getScreenshot()	ブラウザーのビューポートの画像を表す base64 エンコード文字列を返します。
setScreenSize(x,y)	ブラウザーのビューポートサイズを設定します。 パラメータ: x - (string) ピクセル単位の幅; y - (string) ピクセル単位の高さ。
setScriptTimeout(timeout)	スクリプト読み込みタイムアウトを設定します。 パラメータ: timeout - (integer) ミリ秒単位のタイムアウト値。
setSessionTimeout(timeout)	セッション（ページ読み込み）タイムアウトを設定します。 パラメータ: timeout - (integer) ミリ秒単位のタイムアウト値。
setElementWaitTimeout(timeout)	要素ロケーション戦略の（暗黙的）タイムアウトを設定します。 パラメータ: timeout - (integer) ミリ秒単位のタイムアウト値。
collectPerfEntries(mark)	getResult() メソッドで取得するパフォーマンスエントリを収集します。 パラメータ: mark - (string, optional) パフォーマンススナップショットのマーク。
getRawPerfEntries()	パフォーマンスエントリオブジェクトの配列を返します。
getResult()	ブラウザーセッションの統計情報（エラー情報、パフォーマンススナップショットなど）を含むResult オブジェクトを返します。
getError()	ブラウザーエラーを含む BrowserError オブジェクトを返します（ブラウザーエラーがない場合は null を返します)。
setError(message)	Result オブジェクトに含めるカスタムエラーメッセージを設定します。 パラメータ: message - (string) エラーメッセージ。
discardError()	Result オブジェクトで返されるエラーを破棄します。
getAlert()	ブラウザーアラートを含むAlert オブジェクトを返します（ブラウザーアラートがない場合は null を返します)。

Method	Description
`chromeOptions()`	定義済みの Chrome ブラウザーオプションを含む `chromeOptions` オブジェクトを返します。
`firefoxOptions()`	定義済みの Firefox ブラウザーオプションを含む `firefoxOptions` オブジェクトを返します。
`safariOptions()`	定義済みの Safari ブラウザーオプションを含む `safariOptions` オブジェクトを返します。
`edgeOptions()`	定義済みの Edge ブラウザーオプションを含む `edgeOptions` オブジェクトを返します。
`switchFrame(target)`	指定したフレームに切り替えます。 パラメータ: `target` - (browser element or integer, optional) 対象フレーム。要素でフレームを選択するには、その要素を渡します。インデックスでフレームを選択するには、その番号を渡します。空のままにすると、最上位のブラウジングコンテキストに切り替わります。

すべての Browser メソッドは、以下のエラーをスローする可能性があります。

- `BrowserError` - Error オブジェクトから派生し、Browser コンストラクタが失敗した場合にスローされます。追加の `browser` プロパティを含み、この `BrowserError` をスローした Browser オブジェクトを保持します。
- `WebdriverError` - `BrowserError` から派生します。`BrowserError` オブジェクトと同じプロパティを含み、このエラーが WebDriver レスポンス内のエラーへの応答として生成されたかどうかを示します。

Element

`Element` オブジェクトは、`Browser` オブジェクトの `findElement()`/`findElements()` メソッドによって返され、直接構築することはできません。

`Element` オブジェクトは Web ページ内の要素を表し、それとやり取りするためのメソッドを提供します。

`Element` オブジェクトでサポートされるメソッドは次のとおりです。

Method	Description
`getAttribute(name)`	要素の属性値を文字列として返します（指定した属性が見つからない場合は `null` を返します）。 Parameters: `name` - (string) 属性名。
`getProperty(name)`	要素のプロパティ値を文字列として返します（指定したプロパティが見つからない場合は `null` を返します）。 Parameters: `name` - (string) プロパティ名。
`getText()`	要素のテキスト値を文字列として返します。
`click()`	要素をクリックします。
`clear()`	編集可能な要素の内容をクリアします。
`sendKeys(keys)`	キーを送信します。 Parameters: `keys` - (string) 送信するキー。

Cookie

`Cookie` オブジェクトは、`Browser` オブジェクトの `getCookies()` メソッドによって返され、`addCookie()` メソッドに渡されます。

`Cookie` オブジェクトにはメソッドはありませんが、以下のプロパティを含めることができます。

Property	Type	Description
name	string	Cookie の名前。
value	string	Cookie の値。
path	string	Cookie が有効なパス。 Cookie の追加時に省略した場合、デフォルトは `"/"` です。

Property	Type	Description
domain	string	Cookie を参照できるドメイン。 Cookie の追加時に省略した場合、デフォルトはセッションの現在のブラウジングコンテキストのアクティブなドキュメントの URL ドメインです。
secure	boolean	Cookie がセキュアであるかどうかを示します。 Cookie の追加時に省略した場合、デフォルトは `false` です。
httpOnly	boolean	Cookie が HTTP 専用であるかどうかを示します。 Cookie の追加時に省略した場合、デフォルトは `false` です。
expiry	integer	Cookie の有効期限（Unix エポックからの秒数）。Cookie の追加時に省略する場合は、設定してはいけません。
sameSite	string	Cookie の `sameSite` 属性で、Cookie をファーストパーティまたは same-site コンテキストに制限するかどうかを制御します。 `"Lax"` または `"Strict"` のいずれかを設定できます。 Cookie の追加時に省略した場合、デフォルトは `"None"` です。

アラート

`Alert` オブジェクトはウェブページのアラートを表し、`Browser` オブジェクトの `getAlert()` メソッドによって返され、直接構築することはできません。

`Alert` オブジェクトには、アラートのテキスト（またはアラートがない場合は `null`）を持つ `text` プロパティが含まれています。

`Alert` オブジェクトでは、以下のメソッドがサポートされています。

メソッド	説明
`accept()`	アラートを受け入れます。
`dismiss()`	アラートを閉じます。

Result

`Result` オブジェクトにはセッション統計が含まれており、`Browser` オブジェクトの `getResult()` メソッドによって返されます。

通常、`Result` オブジェクトは文字列化されてスクリプトから返され、前処理を通じて依存アイテム値にパースされます。

`Result` オブジェクトにはメソッドはありませんが、以下のプロパティを含めることができます。

プロパティ			型	説明
duration			string	セッション作成から結果取得までのセッション継続時間。
error			object	エラー情報。
	http_status		integer	WebDriver から返された HTTP ステータス（WebDriver エラーがない場合
	error_code		string	WebDriver から返されたエラー（WebDriver エラーがない場合は空文字列
	message		string	WebDriver エラーメッセージ（WebDriver エラーがない場合は空文字列）。
performance_data			object	パフォーマンス統計。
	summary		object	パフォーマンスサマリー。
		navigation	object	ナビゲーションサマリー。
		resource	object	リソースサマリー。
	details		array of objects	ナビゲーションにつながる可能性のある各操作後のパフォーマンス統計。
		mark	string	*(オプション)* `collectPerfEntries()` メソッドで指定されたパフォーマ
		navigation	object	ナビゲーション統計。
		resource	object	このステップのリソースサマリー。
		user	array of objects	mark/measure タイプの統計の配列。
	marks		array of objects	マークされたパフォーマンススナップショットインデックス。
		name	string	パフォーマンススナップショットマーク名。
		index	integer	details 配列内のパフォーマンススナップショットインデックス。

CSV から **JSON** への前処理

概要

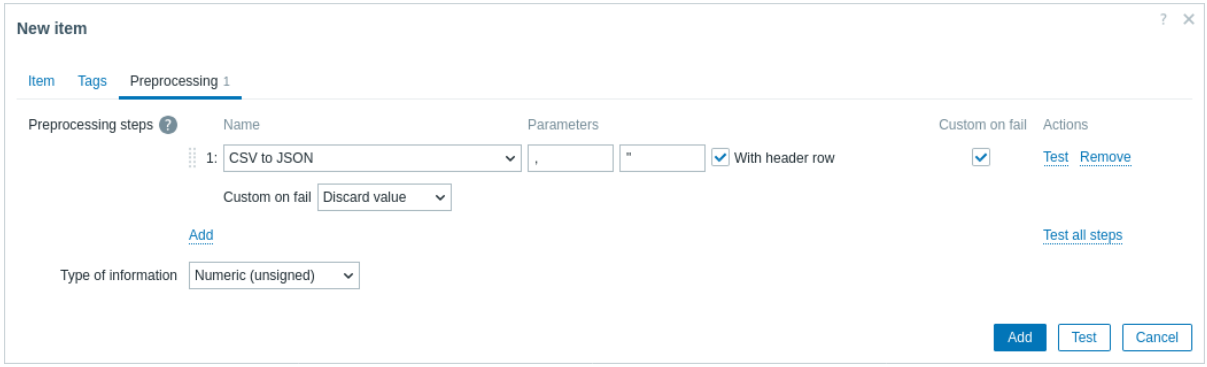
この前処理ステップでは、CSV ファイルのデータを JSON 形式に変換できます。以下でサポートされています。

- アイテム（アイテムのプロトタイプ）
- ローレベルディスカバリルール

設定

CSV から JSON への前処理ステップを設定するには、次の手順に従います。

1. アイテム/ディスカバリルールの設定で、前処理タブに移動します。
2. 追加をクリックします。
3. *CSV から JSON* オプションを選択します。



最初のパラメータでは、カスタム区切り文字を設定できます。CSV 入力の最初の行が「Sep=」で始まり、その後に単一の UTF-8 文字が続く場合、最初のパラメータが設定されていなければ、その文字が区切り文字として使用されます。最初のパラメータが設定されておらず、「Sep=」行から区切り文字を取得できない場合は、カンマが区切り文字として使用されます。

2 番目のオプションパラメータでは、引用符を設定できます。

ヘッダー行ありチェックボックスをオンにすると、ヘッダー行の値が列名として解釈されます（詳細については、ヘッダーの処理を参照してください）。

失敗時のカスタム処理チェックボックスをオンにすると、前処理ステップに失敗してもアイテムはサポートされていない状態になりません。さらに、カスタムエラー処理オプションとして、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかを選択できます。

ヘッダー処理

CSV ファイルのヘッダー行は、2 種類の方法で処理することができます。

- ヘッダ行を含むチェックボックスがチェックされている場合 - ヘッダー行の値はカラム名として解釈されます。この場合、カラム名は一意でなければならず、データ行はヘッダー行より多くのカラムを含んではなりません。
- ヘッダ行を含むチェックボックスがチェックされていない場合 - ヘッダー行はデータとして解釈されます。カラム名は自動的に生成されます (1,2,3,4...)。

CSV ファイルの例:

```
Nr,Item name,Key,Qty
1,active agent item,agent.hostname,33
"2","passive agent item","agent.version","44"
3,"active,passive agent items",agent.ping,55
```

Note:
入力の引用フィールド内の引用文字は、その前に別の引用文字を置くことでエスケープしなければいけません。

ヘッダー行の処理

ヘッダー行が検出される場合の JSON 出力:

```
[
   {
      "Nr":"1",
      "Item name":"active agent item",
      "Key":"agent.hostname",
      "Qty":"33"
   },
   {
      "Nr":"2",
      "Item name":"passive agent item",
      "Key":"agent.version",
```

```
      "Qty":"44"
   },
   {
      "Nr":"3",
      "Item name":"active,passive agent items",
      "Key":"agent.ping",
      "Qty":"55"
   }
]
```


ヘッダー行の処理なし

ヘッダー行が検出できない場合の JSON 出力:


```
[
   {
      "1":"Nr",
      "2":"Item name",
      "3":"Key"
      "4":"Qty"
   },
   {
      "1":"1",
      "2":"active agent item",
      "3":"agent.hostname"
      "4":"33"
   },
   {
      "1":"2",
      "2":"passive agent item",
      "3":"agent.version"
      "4":"44"
   },
   {
      "1":"3",
      "2":"active,passive agent items",
      "3":"agent.ping"
      "4":"55"
   }
]
```


アイテムタイプ

概要

アイテムのタイプは、システムからデータを取得するさまざまな方法を対象としています。各アイテムタイプには、サポートされるアイテムキーと必要なパラメータのセットがあります。

現在、Zabbix では次のアイテムタイプが提供されています。

- Zabbix エージェント
  - ログファイル監視
- シンプルチェック
  - VMware 監視
- SNMP エージェント
- SNMP トラップ
- Zabbix 内部
- Zabbix トラッパー
- 外部チェック
- データベースモニター
- HTTP エージェント
  - Prometheus チェック
- IPMI エージェント
- SSH エージェント
- Telnet エージェント

- JMX エージェント
- 計算アイテム
- 依存アイテム
- スクリプトアイテム
- ブラウザーアイテム
- テレメトリクエリ

すべてのアイテムタイプの詳細は、このセクションのサブページに記載されています。アイテムタイプにはデータ収集のための多くのオプションがありますが、user parametersやロード可能モジュールを使用して、さらに多くのオプションを利用できます。

一部のチェックは Zabbix サーバーだけで実行されます（エージェントレス監視）。一方、その他のチェックでは Zabbix エージェント、または Zabbix Java ゲートウェイ（JMX 監視の場合）が必要です。

Attention:
特定のアイテムタイプで特定のインターフェースが必要な場合（たとえば、IPMI チェックではホストに IPMI インターフェースが必要）、そのインターフェースがホスト定義に存在している必要があります。

ホスト定義には複数のインターフェースを設定できます。Zabbix エージェント、SNMP エージェント、JMX、IPMI が対象です。アイテムが複数のインターフェースを使用できる場合、利用可能なホストインターフェースを（次の順序で: Agent→SNMP→JMX→IPMI）検索し、リンクする最初の適切なインターフェースを使用します。

テキストを返すすべてのアイテム（文字、ログ、テキストの情報タイプ）は、該当する場合、空白のみを返すこともできます。その場合、戻り値は空の文字列に設定されます（2.0 以降でサポート）。

Zabbix エージェント

概要

このセクションでは、データ収集に Zabbix エージェントとの通信を使用するアイテムキーの詳細を説明します。

パッシブおよびアクティブ エージェントチェックがあります。
アイテムを設定する際に、必要なタイプを選択できます。

- *Zabbix agent* - パッシブチェック用
- *Zabbix agent (active)* - アクティブチェック用

Zabbix エージェントでサポートされるすべてのアイテムキーは、新世代の Zabbix agent 2 でもサポートされることに注意してください。
agent 2 でのみ使用できる追加のアイテムキー を参照してください。

サポートされているアイテムキー

Zabbix エージェントで使用できるアイテムキーを以下に示します。

アイテムキーは、パラメータおよび追加情報を省略して記載しています。詳細を確認するには、アイテムキーをクリックしてください。

アイテムキー	説明	アイテムグループ
kernel.maxfiles	OS でサポートされるオープンファイル数の最大値。	カーネル
kernel.maxproc	OS でサポートされるプロセス数の最大値。	
kernel.openfiles	現在オープンされているファイルディスクリプター数。	
log	ログファイルの監視。	ログ監視
log.count	監視対象のログファイルで一致した行数。	
logrt	ローテーションされるログファイルの監視。	
logrt.count	ローテーションされる監視対象のログファイルで一致した行数。	
modbus.get	Modbus データの読み取り。	Modbus
net.dns	DNS サービスの状態を確認します。	ネットワーク
net.dns.perf	DNS サービスのパフォーマンスを確認します。	
net.dns.record	DNS クエリを実行します。	
net.if.collisions	ウィンドウ外の衝突数。	
net.if.discovery	ネットワークインターフェースの一覧。	
net.if.get	詳細情報を含むネットワークインターフェースの一覧。	
net.if.in	ネットワークインターフェースの受信トラフィック統計。	
net.if.out	ネットワークインターフェースの送信トラフィック統計。	
net.if.total	ネットワークインターフェースの受信および送信トラフィック統計の合計。	

アイテムキー	説明	アイテムグループ
net.tcp.listen	この TCP ポートが LISTEN 状態かどうかを確認します。	
net.tcp.port	指定したポートへの TCP 接続が可能かどうかを確認します。	
net.tcp.service	サービスが実行され、TCP 接続を受け付けているかどうかを確認します。	
net.tcp.service.perf	TCP サービスのパフォーマンスを確認します。	
net.tcp.socket.count	パラメータに一致する TCP ソケット数を返します。	
net.udp.listen	この UDP ポートが LISTEN 状態かどうかを確認します。	
net.udp.service	サービスが実行され、UDP リクエストに応答しているかどうかを確認します。	
net.udp.service.perf	UDP サービスのパフォーマンスを確認します。	
net.udp.socket.count	パラメータに一致する UDP ソケット数を返します。	
proc.cpu.util	プロセスの CPU 使用率。	プロセス
proc.get	OS プロセスとそのパラメータの一覧。	
proc.mem	プロセスが使用しているメモリ量（バイト単位）。	
proc.num	プロセス数。	
sensor	ハードウェアセンサーの読み取り値。	センサー
system.boottime	システムの起動時刻。	システム
system.cpu.discovery	検出された CPU/CPU コアの一覧。	
system.cpu.intr	デバイス割り込み。	
system.cpu.load	CPU 負荷。	
system.cpu.num	CPU 数。	
system.cpu.switches	コンテキストスイッチ数。	
system.cpu.util	CPU 使用率。	
system.hostname	システムのホスト名。	
system.hw.chassis	シャーシ情報。	
system.hw.cpu	CPU 情報。	
system.hw.devices	PCI または USB デバイスの一覧。	
system.hw.macaddr	MAC アドレスの一覧。	
system.localtime	システム時刻。	
system.run	ホスト上で指定したコマンドを実行します。	
system.stat	システム統計。	
system.sw.arch	ソフトウェアアーキテクチャ情報。	
system.sw.os	オペレーティングシステム情報。	
system.sw.os.get	オペレーティングシステムに関する詳細情報（バージョン、種類、ディストリビューション名、マイナーバージョンおよびメジャーバージョンなど）。	
system.sw.packages	インストール済みパッケージの一覧。	
system.sw.packages.get	インストール済みパッケージの詳細な一覧。	
system.swap.in	スワップイン（デバイスからメモリへの読み込み）統計。	
system.swap.out	スワップアウト（メモリからデバイスへの書き出し）統計。	
system.swap.size	スワップ領域のサイズ（バイト単位または合計に対する割合）。	
system.uname	システムの識別情報。	
system.uptime	システムの稼働時間（秒単位）。	
system.users.num	ログイン中のユーザー数。	
vfs.dev.discovery	ブロックデバイスとその種類の一覧。	仮想ファイルシステム
vfs.dev.get	詳細情報を含むブロックデバイスの一覧。	
vfs.dev.read	ディスク読み取り統計。	
vfs.dev.write	ディスク書き込み統計。	
vfs.dir.count	ディレクトリエントリ数。	
vfs.dir.get	ディレクトリエントリの一覧。	
vfs.dir.size	ディレクトリサイズ。	
vfs.file.cksum	UNIX cksum アルゴリズムで計算されたファイルチェックサム。	
vfs.file.contents	ファイルの内容を取得します。	
vfs.file.exists	ファイルが存在するかどうかを確認します。	
vfs.file.get	ファイルに関する情報を返します。	
vfs.file.md5sum	ファイルの MD5 チェックサム。	
vfs.file.owner	ファイルの所有者を取得します。	
vfs.file.permissions	UNIX パーミッションの 8 進数を含む 4 桁の文字列を返します。	
vfs.file.regexp	ファイル内の文字列を取得します。	
vfs.file.regmatch	ファイル内の文字列を検索します。	
vfs.file.size	ファイルサイズ。	
vfs.file.time	ファイル時刻情報。	

アイテムキー	説明	アイテムグループ
vfs.fs.discovery	マウントされているファイルシステムと、その種類およびマウントオプションの一覧。	
vfs.fs.get	マウントされているファイルシステムと、その種類およびマウントオプションの一覧。*full* モードでは、使用可能なディスク領域と inode 統計も含まれます。	
vfs.fs.inode	inode 数または割合。	
vfs.fs.size	ディスク領域（バイト単位または合計に対する割合）。	
vm.memory.size	メモリサイズ（バイト単位または合計に対する割合）。	仮想メモリ
web.page.get	Web ページの内容を取得します。	Web 監視
web.page.perf	Web ページ全体の読み込み時間。	
web.page.regexp	Web ページ上の文字列を検索します。	
agent.hostmetadata	エージェントのホストメタデータ。	Zabbix
agent.hostname	エージェントのホスト名。	
agent.ping	エージェントの可用性チェック。	
agent.variant	Zabbix エージェントの種類（Zabbix エージェントまたは Zabbix エージェント 2）。	
agent.version	Zabbix エージェントのバージョン。	
zabbix.stats	Zabbix サーバーまたはプロキシの内部メトリクスのセットを返します。`ip` と `port` が指定されている場合はリモートインスタンスから、それ以外の場合はローカルインスタンスからメトリクスを取得します。	
zabbix.stats (queue)	Zabbix サーバーまたはプロキシで遅延している、キュー内の監視対象アイテム数を返します。`ip` と `port` が指定されている場合はリモートインスタンスから、それ以外の場合はローカルインスタンスからメトリクスを取得します。	

サポートされているプラットフォーム

アイテムの詳細で特に指定がない限り、エージェントアイテム（およびすべてのパラメータ）は以下でサポートされています:

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

多くのエージェントアイテムは **Windows** でもサポートされています。詳細はWindows エージェントアイテムのページを参照してください。

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

kernel.maxfiles

 OS がサポートする最大オープンファイル数。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD。

kernel.maxproc

 OS がサポートする最大プロセス数。
 戻り値: 整数。
 対応プラットフォーム: Linux 2.6 以降、FreeBSD、Solaris、MacOS X、OpenBSD、NetBSD。

kernel.openfiles

 現在開いているファイルディスクリプタの数。
 戻り値：整数。
 対応プラットフォーム：Linux（他の UNIX 系プラットフォームでも動作する場合があります）。

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 ログファイルの監視。
 戻り値: *Log*。
 supported platforms を参照してください。

パラメーター:

- **file** - ログファイルのフルパスとファイル名。
- **regexp** - 必要なパターンを記述する正規expression。
- **encoding** - コードページidentifier。

- **maxlines** - エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新しい行の最大数。このパラメーターはzabbix_agentd.conf の MaxLinesPerSecond の値を上書きします。
- **mode** - 使用可能な値:
  - *all* (default);
  - *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。
- **output** - 任意の出力フォーマットテンプレート。**\0** エスケープシーケンスは一致したテキストの一部（一致が始まる最初の文字から一致が終わる文字まで）に置き換えられ、**\N**（N=1...9）は N 番目の一致グループ（または N が取得されたグループ数を超える場合は空文字列）に置き換えられます。
- **maxdelay** - 最大遅延（秒）。型: float。値:
  - 0 - (default) ログファイルの行を決して無視しない。
  - > 0.0 - ”maxdelay” 秒以内に解析された最新の行を取得するため、古い行を無視する。使用する前にmaxdelay の注意事項をお読みください。
- **options** - 追加オプション:
mtime-noreread - 一意でないレコード。ファイルサイズが変化した場合のみ再読み込みします（更新時刻の変更は無視します）。(このパラメーターは現在 mtime が無視されるため、5.0.2 以降は非推奨です。）
- **persistent dir**（Unix システム上の zabbix_agentd でのみ使用可能。Zabbix agent 2 ではサポートされません）- 永続ファイルを保存するディレクトリの絶対パス名。ログアイテムのpersistent files に関する追加の注意事項も参照してください。

コメント:

- アイテムはactive check として設定する必要があります。
- ファイルが存在しない場合、または権限によりアクセスできない場合、アイテムはサポート対象外になります。
- output が空の場合、マッチしたテキストを含む行全体が返されます。*Matches regular expression* を除くすべてのグローバル正規表現タイプは常に一致した行全体を返し、output パラメーターは無視されることに注意してください。
- output パラメーターを使用した内容の抽出はエージェント上で行われます。

例:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

ログレコードから数値を抽出するために output パラメーターを使用する例:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] # このアイテムはロ
```

サーバーに送信する前にログレコードを書き換えるために output パラメーターを使用する例:

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

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 監視対象のログファイル内で一致した行数を数えます。
 戻り値: *Integer*。
 supported platforms を参照してください。

パラメーター:

- **file** - ログファイルの完全なパスと名前;
- **regexp** - 必要なパターンを記述する正規expression;
- **encoding** - コードページのidentifier;
- **maxproclines** - エージェントが 1 秒あたりに解析する新しい行の最大数（10000 を超えることはできません）。デフォルト値はzabbix_agentd.conf の MaxLinesPerSecond の 10 倍です。
- **mode** - 指定可能な値:
  - *all*（デフォルト）;
  - *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。
- **maxdelay** - 最大遅延秒数。型: float。値:
  - 0 -（デフォルト）ログファイルの行を決して無視しません;
  - > 0.0 - ”maxdelay” 秒以内に最新の行を解析できるようにするため、古い行を無視します。使用する前にmaxdelay の注意事項をお読みください。
- **options** - 追加オプション:
mtime-noreread - 一意でないレコード。ファイルサイズが変化した場合のみ再読み込みします（更新時刻の変更は無視します）。(このパラメーターは、現在 mtime が無視されるため、5.0.2 以降は非推奨です。）
- **persistent dir**（Unix システム上の zabbix_agentd でのみ使用可能。Zabbix agent 2 ではサポートされません）- 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルに関する追加の注意事項については、persistent files も参照してください。

コメント:

- アイテムはactive check として設定する必要があります。
- 一致した行は、エージェントによる前回のログチェック以降の新しい行としてカウントされるため、アイテムの更新間隔に依存します。

- ファイルが存在しない場合、または権限によりアクセスできない場合、アイテムはサポート対象外になります。

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 ローテーションされるログファイルの監視。
 戻り値: *Log*。
 supported platforms を参照してください。

パラメーター:

- **file regexp** - ファイルへの絶対パス。ファイル名は正規表現を使用して指定します。正規表現はファイル名にのみ適用され、名前全体に一致する必要はありません (例: `/path/to/agent` は `zabbix_agentd.log` に一致します)。
- **regexp** - 必要な内容パターンを記述する正規表現。
- **encoding** - コードページ識別子。
- **maxlines** - エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行の最大数。このパラメーターは`zabbix_agentd.conf` の `MaxLinesPerSecond` の値を上書きします。
- **mode** - 指定可能な値:
  - *all* (デフォルト);
  - *skip* - 古いデータの処理をスキップします (新規作成されたアイテムにのみ影響します)。
- **output** - 任意の出力フォーマット用テンプレート。**\0** エスケープシーケンスは一致したテキストの一部 (一致が始まる最初の文字から一致が終わる文字まで) に置き換えられ、**\N** (N=1...9) エスケープシーケンスは N 番目の一致グループ (または N が取得されたグループ数を超える場合は空文字列) に置き換えられます。
- **maxdelay** - 最大遅延秒数。型: float。値:
  - 0 - (デフォルト) ログファイルの行を決して無視しない;
  - > 0.0 - "maxdelay" 秒以内に解析される最新の行を取得するため、古い行を無視する。使用する前にmaxdelay の注意事項をお読みください!
- **options** - ログファイルローテーションの種類およびその他のオプション。指定可能な値:
  - *rotate* (デフォルト)。
  - *copytruncate* - *copytruncate* は *maxdelay* と併用できないことに注意してください。この場合、*maxdelay* は 0 であるか、指定されていない必要があります。詳細はcopytruncate の注意事項を参照してください。
  - *mtime-reread* - 一意でないレコード。更新時刻またはサイズが変化した場合に再読み込みします (明示的に `mtime-*` オプションが設定されていない場合はデフォルトで使用されます)。
  - *mtime-noreread* - 一意でないレコード。サイズが変化した場合のみ再読み込みします (更新時刻の変化は無視します)。
- **persistent dir** (`zabbix_agentd` の Unix システムでのみ使用可能。Zabbix agent 2 ではサポートされません) - 永続ファイルを保存するディレクトリの絶対パス名。persistent files に関する追加の注意事項も参照してください: persistent files。

コメント:

- このアイテムはactive check として設定する必要があります。
- ログローテーションはファイルの最終更新時刻に基づいています。
- `logrt` は、現在アクティブな 1 つのログファイルと、ローテーションされた複数の一致する非アクティブファイルで動作するように設計されています。たとえば、ディレクトリに多数のアクティブなログファイルがある場合は、それぞれに対して個別の `logrt` アイテムを作成してください。そうしないと、1 つの `logrt` アイテムが多すぎるファイルを拾うことで、メモリ不足や監視のクラッシュにつながる可能性があります。
- `output` が空の場合は、一致したテキストを含む行全体が返されます。なお、*Matches regular expression* 以外のすべてのグローバル正規表現タイプは常に一致した行全体を返し、`output` パラメーターは無視されます。
- `output` パラメーターを使用した内容の抽出はエージェント側で行われます。
- `file regexp` パラメーターでは、ログディレクトリのパスとログファイルの正規表現は正しいディレクトリ区切り文字で区切る必要があります:
  - Windows では、区切り文字はバックスラッシュ (\) でなければなりません。スラッシュ (/) は、ログディレクトリのパスとログファイルの正規表現を区切る部分以外では許容される場合があります (以下の例を参照)。
  - その他のシステムでは、スラッシュ (/) でなければなりません。

Windows の例:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、ファイル名が "filename" で始まり
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、ファイル名が "
```

その他のシステムの例:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] # このアイテムは "logfile1" のようなファイルに一致します (".
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] # このアイテムは "logfile_abc_1" 
```

ログレコードから数値を抽出するために `output` パラメーターを使用する例:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] # このアイ
```

サーバーへ送信する前にログレコードを書き換えるために `output` パラメーターを使用する例:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 ローテーションされる監視対象のログファイル内で、条件に一致した行の数を返します。
 戻り値: *Integer*。
 supported platforms を参照してください。

パラメーター:

- **file regexp** - ファイルへの絶対パス。ファイル名は正規expressionで指定します。正規表現はファイル名にのみ適用され、名前全体に一致する必要はありません（例: `/path/to/agent` は `zabbix_agentd.log` に一致します）。
- **regexp** - 必要なパターンを表す正規expression。
- **encoding** - コードページidentifier。
- **maxproclines** - エージェントが 1 秒あたりに解析する新しい行の最大数（10000 を超えることはできません）。デフォルト値は`zabbix_agentd.conf` の MaxLinesPerSecond の 10 倍です。
- **mode** - 指定可能な値:
  - *all*（デフォルト）;
  - *skip* - 古いデータの処理をスキップします（新しく作成されたアイテムにのみ影響します）。
- **maxdelay** - 最大遅延秒数。型: float。値:
  - 0 -（デフォルト）ログファイルの行を決して無視しません;
  - > 0.0 - ”maxdelay” 秒以内に最新の行を解析できるように、古い行を無視します。使用前に`maxdelay` の注意事項をお読みください!
- **options** - ログファイルローテーションの種類およびその他のオプション。指定可能な値:
  - *rotate*（デフォルト）。
  - *copytruncate* - copytruncate は maxdelay と併用できないことに注意してください。この場合、maxdelay は 0 であるか、指定しない必要があります。`copytruncate` の注意事項を参照してください。
  - *mtime-reread* - 一意でないレコード。更新時刻またはサイズが変化した場合に再読み込みします（デフォルト）。
  - *mtime-noreread* - 一意でないレコード。サイズが変化した場合のみ再読み込みします（更新時刻の変化は無視します）。
- **persistent dir**（Unix システム上の zabbix_agentd でのみ使用可能。Zabbix agent 2 ではサポートされません）- 永続ファイルを保存するディレクトリの絶対パス名。永続ファイルの追加の注意事項については、persistent files も参照してください。

コメント:

- このアイテムはactive check として設定する必要があります。
- 一致した行は、エージェントによる前回のログチェック以降の新しい行としてカウントされるため、アイテムの更新間隔に依存します。
- ログローテーションは、ファイルの最終更新時刻に基づいています。
- `file regexp` パラメーターでは、ログディレクトリのパスとログファイルの正規表現を正しいディレクトリ区切り文字で区切る必要があります:
  - Windows では、区切り文字はバックスラッシュ (\) でなければなりません。スラッシュは他の位置では許容される場合がありますが、ログディレクトリのパスとログファイルの正規表現を区切る位置では使用できません（以下の例を参照）。
  - その他のシステムでは、フォワードスラッシュ (/) でなければなりません。

Windows の例:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、"c:/dir1/dir2/dir3" にあ
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、ネット"
```

その他のシステムの例:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] # このアイテムは、"/home/zabbix/logs" ディレクト
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] # このアイテムは、"/home/use
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Modbus データを読み取ります。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux, Windows。

パラメーター:

- **endpoint** - `protocol://connection_string` として定義されたエンドポイント;
- **slave id** - スレーブ ID;
- **function** - Modbus 関数;
- **address** - 最初のレジスタ、コイル、または入力のアドレス;
- **count** - 読み取るレコード数;
- **type** - データの種類;
- **endianness** - エンディアン設定;
- **offset** - address から開始するレジスタ数で、その結果は破棄されます。

Windows では、このアイテムは endpoint パラメーターで UNC パスに従います。これはセキュリティリスクとなる可能性があります。

パラメーターの詳細な説明を参照してください。

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS サービスの状態を確認します。
 戻り値: - 0 - DNS 解決に失敗しました（DNS サーバーが応答しなかったか、エラーを返しました）; - 1 - DNS 解決に成功しました。
 対応プラットフォームを参照してください。

パラメーター:

- **ip**（Windows では Zabbix agent 2 を使用しない限り無視されます）- DNS サーバーの IP アドレス（空欄の場合は既定の DNS サーバーを使用します）;
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ（既定は SOA）;
- **timeout**（Windows では Zabbix agent 2 を使用しない限り無視されます）- リクエストのタイムアウト（秒単位、既定は 1 秒）;
- **count**（Windows では Zabbix agent 2 を使用しない限り無視されます）- リクエストの試行回数（既定は 2）;
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp*（既定）または *tcp*。

コメント:

- type に指定できる値: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS*（Windows 上の Zabbix agent、およびすべての OS 上の Zabbix agent 2 ではサポートされません）, *HINFO*, *MINFO*, *TXT*, *SRV*。
- 逆引き DNS 検索（type が *PTR* に設定されている場合）では、DNS 名を逆順形式と非逆順形式の両方で指定できます（以下の例を参照）。PTR レコードが要求される場合、DNS 名は実際には IP アドレスです。
- 国際化ドメイン名はサポートされていません。代わりに IDNA でエンコードされた名前を使用してください。

例:

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS サービスのパフォーマンスを確認します。
 戻り値: *Float* (0 - サービスが停止している; 秒 - サービスからの応答を待機するのに費やした秒数)。
 supported platforms を参照してください。

パラメーター:

- **ip** (Zabbix agent 2 を使用しない限り、Windows では無視されます) - DNS サーバーの IP アドレス (空欄にすると既定の DNS サーバーを使用します);
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ (既定値は SOA);
- **timeout** (Zabbix agent 2 を使用しない限り、Windows では無視されます) - リクエストのタイムアウトを秒単位で指定します (既定値は 1 秒);
- **count** (Zabbix agent 2 を使用しない限り、Windows では無視されます) - リクエストの試行回数 (既定値は 2);
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp* (既定) または *tcp*。

コメント:

- type に指定できる値は次のとおりです:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (Windows 上の Zabbix agent、すべての OS 上の Zabbix agent 2 ではサポートされません), *HINFO*, *MINFO*, *TXT*, *SRV*。
- 逆引き DNS 検索 (type を *PTR* に設定した場合) では、DNS 名を逆順形式と非逆順形式の両方で指定できます (以下の例を参照してください)。PTR レコードが要求される場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていません。代わりに IDNA でエンコードされた名前を使用してください。
- DNS サーバーがエラーコード (たとえば NXDOMAIN や SERVFAIL) を返した場合、アイテムは 0 ではなく応答時間を返します。

例:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]
```

```
net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS クエリを実行します。

Zabbix エージェント 2 ではnet.dns.get も提供されており、より多くのレコードタイプや DNS 監視のより細かな制御など、追加機能を利用できます。戻り値: 必要な種類の情報を含む文字列。
 supported platforms を参照してください。

パラメーター:

- **ip** (Zabbix エージェント 2 を使用しない限り、Windows では無視されます) - DNS サーバーの IP アドレス（空欄の場合はデフォルトの DNS サーバーを使用）;
- **name** - クエリ対象の DNS 名;
- **type** - クエリするレコードタイプ（デフォルトは SOA）;
- **timeout** (Zabbix エージェント 2 を使用しない限り、Windows では無視されます) - リクエストのタイムアウト（秒単位、デフォルトは 1 秒）;
- **count** (Zabbix エージェント 2 を使用しない限り、Windows では無視されます) - リクエストの試行回数（デフォルトは 2）;
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp*（デフォルト）または *tcp*。

コメント:

- type に指定可能な値は次のとおりです:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (Windows 上の Zabbix エージェント、およびすべての OS 上の Zabbix エージェント 2 ではサポートされません), *HINFO*, *MINFO*, *TXT*, *SRV*。
- 逆引き DNS ルックアップ（type が *PTR* に設定されている場合）では、DNS 名を逆順または通常順のどちらでも指定できます (以下の例を参照)。PTR レコードが要求される場合、DNS 名は実際には IP アドレスです。
- 国際化ドメイン名はサポートされていません。代わりに IDNA でエンコードされた名前を使用してください。

例:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 ウィンドウ外の衝突回数。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名

net.if.discovery

 ネットワークインターフェースの一覧です。
ローレベルディスカバリに使用されます。
 戻り値: *JSON string*。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD、Windows。

net.if.get[<regexp>]

 詳細情報を含むネットワークインターフェースの一覧です。
 低レベルディスカバリに使用されます。
 このアイテムは、`net.if.discovery` Zabbix エージェントアイテムの拡張版です。
 戻り値: *JSON object*。
 対応プラットフォーム: Linux。

パラメーター:

- **regexp** - ネットワークインターフェース名に一致する正規expression。

コメント:

- 名前が指定された正規表現に一致するネットワークインターフェースを含む JSON object を返します（詳細を参照）。

net.if.in[if,<mode>]

 ネットワークインターフェース上の受信トラフィック統計です。
 戻り値: *Integer*。
 対応プラットフォーム: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows。NetBSD では root 権限が必要です。

パラメーター:

- **if** - ネットワークインターフェース名 (Unix); ネットワークインターフェースの完全な説明または IPv4 アドレス; または、波括弧内の場合はネットワークインターフェース GUID (Windows);
- **mode** - 使用可能な値:
  - *bytes* - バイト数 (デフォルト);
  - *packets* - パケット数;
  - *errors* - エラー数;
  - *dropped* - ドロップされたパケット数;
  - *overruns (fifo)* - FIFO バッファエラー数;
  - *frame* - パケットフレーミングエラー数;
  - *compressed* - デバイスドライバーが受信した圧縮パケット数;
  - *multicast* - デバイスドライバーが受信したマルチキャストフレーム数。

コメント:

- 1 秒あたりの変化量の前処理ステップとこのキーを組み合わせることで、1 秒あたりのバイト数の統計を取得できます。
- *dropped* モードは Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD でのみサポートされています。
- *overruns*, *frame*, *compressed*, *multicast* モードは Linux でのみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 ネットワークインターフェース上の送信トラフィック統計です。
 戻り値: *Integer*。
 サポートされるプラットフォーム: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows。NetBSD では root 権限が必要です。

パラメーター:

- **if** - ネットワークインターフェース名 (Unix); ネットワークインターフェースの完全な説明または IPv4 アドレス; または、波括弧で囲む場合はネットワークインターフェース GUID (Windows);
- **mode** - 指定可能な値:
  - *bytes* - バイト数 (デフォルト);
  - *packets* - パケット数;
  - *errors* - エラー数;
  - *dropped* - ドロップされたパケット数;
  - *overruns (fifo)* - FIFO バッファエラー数;
  - *collisions (colls)* - インターフェースで検出された衝突数;
  - *carrier* - デバイスドライバーによって検出されたキャリア損失数;
  - *compressed* - デバイスドライバーによって送信された圧縮パケット数。

コメント:

- バイト毎秒の統計を取得するには、このキーを *Change per second* の前処理ステップと組み合わせて使用できます。
- *dropped* モードは Linux, HP-UX でのみサポートされます。
- *overruns*, *collision*, *carrier*, *compressed* モードは Linux でのみサポートされます。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 ネットワークインターフェース上の受信トラフィック統計と送信トラフィック統計の合計。
 戻り値: *Integer*。
 サポートされるプラットフォーム: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows。NetBSD では root 権限が必要です。

パラメーター:

- **if** - ネットワークインターフェース名 (Unix); ネットワークインターフェースの完全な説明または IPv4 アドレス; または、波括弧で囲む場合はネットワークインターフェース GUID (Windows);
- **mode** - 使用可能な値:
  - *bytes* - バイト数 (デフォルト);
  - *packets* - パケット数;
  - *errors* - エラー数;

- *dropped* - ドロップされたパケット数;
- *overruns (fifo)* - FIFO バッファエラー数;
- *collisions (colls)* - インターフェースで検出された衝突数;
- *compressed* - デバイスドライバーによって送受信された圧縮パケット数。

コメント:

- bytes/秒の統計を取得するために、このキーを *Change per second* 前処理ステップと組み合わせて使用できます。
- *dropped* モードは Linux, HP-UX でのみサポートされます。ドロップされたパケットは、お使いのプラットフォームで `net.if.in` と `net.if.out` の両方がドロップされたパケットに対して動作する場合にのみサポートされます。
- *overruns*, *collision*, *compressed* モードは Linux でのみサポートされます。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 この TCP ポートが LISTEN 状態かどうかを確認します。
 戻り値: 0 - LISTEN 状態ではない; 1 - LISTEN 状態である。
 対応プラットフォーム: Linux, FreeBSD, Solaris, MacOS X, Windows.

パラメーター:

- **port** - TCP ポート番号

Linux カーネル 2.6.14 以降では、可能であれば LISTEN 中の TCP ソケットに関する情報はカーネルの NETLINK インターフェースから取得されます。それ以外の場合、情報は `/proc/net/tcp` および `/roc/net/tcp6` ファイルから取得されます。

例:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 指定したポートに TCP 接続できるかどうかを確認します。
 戻り値: 0 - 接続できない; 1 - 接続できる。
 supported platforms を参照してください。

パラメーター:

- **ip** - IP アドレスまたは DNS 名（デフォルトは `127.0.0.1`）;
- **port** - ポート番号。

コメント:

- 単純な TCP パフォーマンステストには `net.tcp.service.perf[tcp,<ip>,<port>]` を使用してください。
- これらのチェックにより、システムデーモンのログファイルに追加メッセージが記録される場合があります（通常、SMTP および SSH セッションのログが記録されます）。

例:

```
net.tcp.port[,80] # このアイテムは、ポート 80 で動作している Web サーバーの可用性をテストするために使用できます
```

net.tcp.service[service,<ip>,<port>]

 サービスが稼働しており、TCP 接続を受け付けているかどうかを確認します。
 戻り値: 0 - サービス停止中; 1 - サービス稼働中。
 supported platforms を参照してください。

パラメーター:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, または *telnet* (details を参照);
- **ip** - IP アドレスまたは DNS 名（デフォルトは `127.0.0.1`）;
- **port** - ポート番号（デフォルトでは標準のサービスのポート番号が使用されます）。

コメント:

- これらのチェックにより、システムのデーモンログファイルに追加メッセージが記録される場合があります（通常、SMTP および SSH セッションがログに記録されます）。
- 暗号化されたプロトコル（ポート 993 の IMAP やポート 995 の POP など）のチェックは、現在サポートされていません。回避策として、このようなチェックには `net.tcp.port[]` を使用してください。
- Windows での LDAP および HTTPS のチェックは、Zabbix エージェント 2 でのみサポートされています。
- telnet チェックは、ログインプロンプト（末尾が':'）を探します。

例:

```
net.tcp.service[ftp,,45] # このアイテムは、TCP ポート 45 上の FTP サーバーの可用性をテストするために使用できます
```

net.tcp.service.perf[service,<ip>,<port>]

 TCP サービスのパフォーマンスを確認します。
 戻り値: *Float* (0 - サービスが停止している; 秒 - サービスからの応答を待機するのに費やした秒数)。
 supported platforms を参照してください。

パラメーター:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, または *telnet* (details を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

コメント:

- 暗号化されたプロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは、現在サポートされていません。回避策として、このようなチェックには `net.tcp.service.perf[tcp,<ip>,<port>]` を使用してください。
- telnet チェックでは、ログインプロンプト (末尾が :) を探します。

例:

```
net.tcp.service.perf[ssh] # このアイテムは SSH サーバーからの初期応答速度をテストするために使用できます
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 パラメータに一致する TCP ソケットの数を返します。
 戻り値：整数。
 対応プラットフォーム：Linux、Windows。

パラメータ:

- **laddr** - ローカル IPv4/6 アドレスまたは CIDR サブネット
- **lport** - ローカルポート番号またはサービス名
- **raddr** - リモート IPv4/6 アドレスまたは CIDR サブネット
- **rport** - リモートポート番号またはサービス名
- **state** - 接続状態（*established*、*syn_sent*、*syn_recv*、*fin_wait1*、*fin_wait2*、*time_wait*、*close*、*close_wait*、*last_ack*、*listen*、*closing*）

例:

```
net.tcp.socket.count[,80,,,established] # 確立状態のローカル TCP ポート 80 への接続数
```

net.udp.listen[port]

 この UDP ポートが LISTEN 状態かどうかをチェックします。
 戻り値: 0 - LISTEN 状態ではありません; 1 - LISTEN 状態です。
 対応プラットフォーム: Linux, FreeBSD, Solaris, MacOS X。

パラメータ:

- **port** - UDP ポート番号

例:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 サービスが稼働しており、UDP リクエストに応答しているかどうかを確認します。
 戻り値: 0 - サービス停止中; 1 - サービス稼働中。
 supported platforms を参照してください。

パラメーター:

- **service** - *ntp* (details を参照);
- **ip** - IP アドレスまたは DNS 名（デフォルトは 127.0.0.1）;
- **port** - ポート番号（デフォルトでは標準のサービスのポート番号が使用されます）。

例:

```
net.udp.service[ntp,,45] # このアイテムは、UDP ポート 45 上の NTP サービスの可用性をテストするために使用できます
```

net.udp.service.perf[service,<ip>,<port>]

 UDP サービスのパフォーマンスを確認します。
 戻り値: *Float* (0 - サービスが停止している; seconds - サービスからの応答を待機するのに費やした秒数)。
 supported platforms を参照してください。

パラメーター:

- **service** - *ntp* (details を参照);

- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1 です);
- **port** - ポート番号 (デフォルトでは標準のサービスのポート番号が使用されます)。

例:

```
net.udp.service.perf[ntp] #this item can be used to test response time from NTP service
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 パラメータに一致する UDP ソケットの数を返します。
 戻り値: 整数。
 対応プラットフォーム: Linux, Windows。

パラメータ:

- **laddr** - ローカル IPv4/6 アドレスまたは CIDR サブネット
- **lport** - ローカルポート番号またはサービス名
- **raddr** - リモート IPv4/6 アドレスまたは CIDR サブネット
- **rport** - リモートポート番号またはサービス名
- **state** - 接続状態 (*established*, *unconn*)

例:

```
net.udp.socket.count[,,,,established] # 接続状態の UDP ソケット数を返す
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 プロセスの CPU 使用率。
 戻り値: *Float*。
 サポートされるプラットフォーム: Linux, Solaris[6]。

パラメータ:

- **name** - プロセス名（デフォルトは *all processes*）;
- **user** - ユーザー名（デフォルトは *all users*）;
- **type** - CPU 使用率の種類: *total*（デフォルト）、*user*、または *system*;
- **cmdline** - コマンドラインでフィルタリングします（正規表現です）;
- **mode** - データ収集モード: *avg1*（デフォルト）、*avg5*、または *avg15*;
- **zone** - 対象ゾーン: *current*（デフォルト）または *all*。このパラメータは Solaris でのみサポートされます。

コメント:

- 返される値は、単一の CPU コア使用率を基準にしています。たとえば、2 コアを 100% 使用しているプロセスの CPU 使用率は 200% です。
- プロセスの CPU 使用率データは、名前、ユーザー名、コマンドラインによって一意となる最大 1024 件のクエリをサポートするコレクターによって収集されます。過去 24 時間にアクセスされなかったクエリは、コレクターから削除されます。
- zone パラメータを *current*（またはデフォルト）に設定した場合、エージェントが zone サポートなしで Solaris 上にコンパイルされているが、zone がサポートされる新しい Solaris 上で実行されていると、エージェントは NOTSUPPORTED を返します (エージェントは結果を現在の zone のみに制限できません)。ただし、この場合でも *all* はサポートされます。

例:

```
proc.cpu.util[,root] # "root" ユーザーで実行されているすべてのプロセスの CPU 使用率
proc.cpu.util[zabbix_server,zabbix] # zabbix ユーザーで実行されているすべての zabbix_server プロセスの CPU 使用率
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 OS プロセスとそのパラメーターの一覧です。低レベルディスカバリに使用できます。
 戻り値: *JSON* 文字列。
 サポートされるプラットフォーム: Linux, FreeBSD, Windows, OpenBSD, NetBSD。

パラメーター:

- **name** - プロセス名（デフォルトは *all processes*）。
- **user** - ユーザー名（デフォルトは *all users*）。
- **cmdline** - コマンドラインでフィルタします（正規表現です）。このパラメーターは Windows ではサポートされません。その他のプラットフォームでは、mode が'summary' に設定されている場合はサポートされません。
- **mode** - 指定可能な値:
  - *process*（デフォルト）;
  - *thread*（NetBSD ではサポートされません）;
  - *summary*。各 mode と OS で返されるプロセスパラメーターの一覧を参照してください。

コメント:

- 値を取得できない場合、たとえばエラー（プロセスがすでに終了している、権限不足、システムコールの失敗など）が原因であれば、-1 が返されます。
- name および cmdline パラメーターを使用してプロセスを選択する際の注意事項を参照してください（Linux 固有）。

例:

```
proc.get[zabbix_server,zabbix,,process] #zabbix ユーザーで実行中のすべての zabbix_server プロセスの一覧。PID ごとに
proc.get[java,,,thread] # すべての Java プロセスの一覧。スレッドごとに 1 エントリを返します
proc.get[,zabbix,,summary] #zabbix ユーザーで実行中の各種類のプロセスの集計データ。プロセス名ごとに 1 エントリを返します
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 プロセスが使用しているメモリ量をバイト単位で返します。
 戻り値: - *Integer* - mode が *max*、*min*、*sum* の場合; - *Float* - mode が *avg* の場合
 サポートされるプラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

パラメータ:

- **name** - プロセス名（デフォルトは *all processes*）;
- **user** - ユーザー名（デフォルトは *all users*）;
- **mode** - 指定可能な値: *avg*、*max*、*min*、または *sum*（デフォルト）;
- **cmdline** - コマンドラインでフィルタします（正規expressionです）;
- **memtype** - プロセスが使用するメモリの種類

コメント:

- memtype パラメータは Linux、FreeBSD、Solaris[6]、AIX でのみサポートされます。
- 複数のプロセスが共有メモリを使用している場合、プロセスが使用しているメモリの合計は大きく、現実的でない値になることがあります。

name および cmdline パラメータを使用したプロセスの選択に関する注記を参照してください（Linux 固有）。

 このアイテムがコマンドラインから呼び出され、コマンドラインパラメータを含む場合（例: エージェントのテストモードを使用する場合: zabbix_agentd -t proc.mem[,,,apache2]）、エージェント自身もカウントされるため、1 つ余分なプロセスが数えられます。

例:

```
proc.mem[,root] # "root" ユーザーで実行されているすべてのプロセスが使用しているメモリ
proc.mem[zabbix_server,zabbix] # zabbix ユーザーで実行されているすべての zabbix_server プロセスが使用しているメモリ
proc.mem[,oracle,max,oracleZABBIX] # コマンドラインに oracleZABBIX を含み、Oracle で実行されている最もメモリを消費す
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 プロセス数。
 戻り値: *Integer*。
 サポートされるプラットフォーム: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows。

パラメータ:

- **name** - プロセス名（デフォルトは *all processes*）。
- **user** - ユーザー名（デフォルトは *all users*）。
- **state** - 指定可能な値:
  - *all*（デフォルト）;
  - *disk* - 割り込み不可のスリープ;
  - *run* - 実行中;
  - *sleep* - 割り込み可能なスリープ;
  - *trace* - 停止中;
  - *zomb* - ゾンビ。
- **cmdline** - コマンドラインでフィルタします（正規expressionです）。
- **zone** - 対象ゾーン: *current*（デフォルト）または *all*。このパラメータは Solaris でのみサポートされます。

コメント:

- *disk* および *trace* の state パラメータは Linux, FreeBSD, OpenBSD, NetBSD でのみサポートされます。
- Windows では、name と user パラメータのみサポートされます。
- このアイテムがコマンドラインから呼び出され、コマンドラインパラメータを含む場合（例: エージェントのテストモードを使用する場合: zabbix_agentd -t proc.num[,,,apache2]）、エージェント自身もカウントされるため、1 つ余分なプロセスがカウントされます。
- zone パラメータを current（またはデフォルト）に設定した場合、エージェントが zone サポートなしで Solaris 上にコンパイルされているが、zone がサポートされる新しい Solaris 上で動作していると、エージェントは NOTSUPPORTED を返します（エージェントは結果を現在の zone のみに制限できません）。ただし、この場合でも *all* はサポートされます。
- name および cmdline パラメータを使用してプロセスを選択する方法については、notes を参照してください（Linux 固有）。

例:

```
proc.num[,mysql] #mysql ユーザーで実行されているプロセス数
proc.num[apache2,www-data] #www-data ユーザーで実行されている apache2 プロセス数
proc.num[,oracle,sleep,oracleZABBIX] #oracle の下で実行され、コマンドラインに oracleZABBIX を含む sleep 状態のプロ
```

sensor[device,sensor,<mode>]

 ハードウェアセンサーの読み取り値。
 戻り値: *Float*。
 サポートされるプラットフォーム: Linux, OpenBSD。

パラメーター:

- **device** - デバイス名;
- **sensor** - センサー名;
- **mode** - 指定可能な値: *avg*, *max*, または *min*。

コメント:

- Linux 2.4 では `/proc/sys/dev/sensors` を読み取ります。
- Linux 2.6+ では `/sys/class/hwmon` を読み取ります。Linux 上のsensor アイテムのより詳細な説明を参照してください。
- OpenBSD では `hw.sensors` MIB を読み取ります。
- Linux 2.4 では、mode を指定すると device と sensor は正規表現として扱われます。
- Linux 2.6+ および OpenBSD では、または mode を省略した場合、device と sensor はそのままの文字列として扱われます。

例:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #1 つの CPU の温度
sensor["cpu[0-2]$",temp,avg] # 最初の 3 つの CPU の平均温度
```

system.boottime

 システムの起動時刻。
 戻り値: 整数 *(Unix* タイムスタンプ*)*。
 対応プラットフォーム: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD。

system.cpu.discovery

 検出された CPU/CPU コアのリスト。低レベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

system.cpu.intr

 デバイス割り込み。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD。

system.cpu.load[<cpu>,<mode>]

 CPU ロード。
 戻り値：*Float*。
 サポートされているプラットフォームを参照。

パラメータ:

- **cpu** - 可能な値：*all*（デフォルト）または *percpu*（合計ロードをオンライン CPU 数で割った値）;
- **mode** - 可能な値：*avg1*（1 分間の平均、デフォルト）、*avg5*、または *avg15*。

例:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 CPU の数。
 戻り値：整数。
 対応プラットフォーム：Linux、FreeBSD、Solaris、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。

パラメータ：

- **type** - 可能な値：*online*（デフォルト）または *max*

max タイプのパラメータは、Linux、FreeBSD、Solaris、MacOS X でのみサポートされています。

例：

```
system.cpu.num
```

system.cpu.switches

 コンテキストスイッチの回数。
 戻り値: 整数。
 対応プラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD、NetBSD。

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 CPU 使用率（%）を返します。
 戻り値：*Float*。
 対応プラットフォーム：Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD、Windows。

パラメータ：

- **cpu** - *<CPU 番号 >* または *all*（デフォルト）；
- **type** - 指定可能な値：*user*（デフォルト）、*idle*、*nice*、*system*、*iowait*、*interrupt*、*softirq*、*steal*、*spin*（OpenBSD のみ）、*guest*（Linux カーネル 2.6.24 以降）、または *guest_nice*（Linux カーネル 2.6.33 以降）；
- **mode** - 指定可能な値：*avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*;
- **logical or physical** - 指定可能な値：*logical*（デフォルト）または *physical*。このパラメータは AIX のみ対応しています。

コメント：

- *nice* タイプパラメータは Linux、FreeBSD、HP-UX、OpenBSD、NetBSD のみ対応しています。
- *iowait* タイプパラメータは Linux 2.6 以降、Solaris、AIX のみ対応しています。
- *interrupt* タイプパラメータは Linux 2.6 以降、FreeBSD、OpenBSD のみ対応しています。
- *softirq*、*steal*、*guest*、*guest_nice* タイプパラメータは Linux 2.6 以降のみ対応しています。
- *avg5* および *avg15* モードパラメータは Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD で対応しています。

例：

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 システムのホスト名。
 戻り値: *String*。
 supported platforms を参照してください。

パラメーター:

- **type** - 可能な値:
  - *netbios* (Windows でのデフォルト);
  - *host* (Linux でのデフォルト);
  - *shorthost* (ホスト名の最初のドットより前の部分を返します。ドットがない名前の場合は完全な文字列を返します);
  - *fqdn* (完全修飾ドメイン名を返します)。
- **transform** - 可能な値:
  - *none* (デフォルト);
  - *lower* (小文字に変換します)。

この値は、uname() システム API の出力から `nodename` を取得することで得られます。

返される値の例:

```
system.hostname → linux-w7x1
system.hostname → example.com
system.hostname[shorthost] → example
system.hostname → WIN-SERV2008-I6
system.hostname[host] → Win-Serv2008-I6LonG
system.hostname[host,lower] → win-serv2008-i6long
system.hostname[fqdn,lower] → blog.zabbix.com
```

system.hw.chassis[<info>]

 シャーシ情報。
 戻り値：文字列。
 対応プラットフォーム：Linux。

パラメータ:

- **info** - 可能な値：*full*（デフォルト）、*model*、*serial*、*type*、*vendor*

コメント:

- このアイテムキーはSMBIOSテーブルの利用可能性に依存します。
- sysfs から DMI テーブルの読み取りを試み、sysfs へのアクセスに失敗した場合はメモリから直接読み取ります。
- 値は sysfs またはメモリから読み取るため、**root** 権限が必要です。

例:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 CPU 情報。
 戻り値: *String* または *Integer*。
 対応プラットフォーム: Linux。

パラメータ:

- **cpu** - *<CPU 番号 >* または *all* (デフォルト);
- **info** - 指定可能な値: *full* (デフォルト), *curfreq*, *maxfreq*, *model* または *vendor*。

コメント:

- /proc/cpuinfo および /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq から情報を取得します;
- CPU 番号と *curfreq* または *maxfreq* が指定された場合、数値 (Hz) が返されます。

例:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 PCI または USB デバイスの一覧。
 戻り値: テキスト。
 対応プラットフォーム: Linux。

パラメータ:

- **type** - *pci* (デフォルト) または *usb*

lspci または lsusb ユーティリティ (パラメータなしで実行) の出力を返します。

例:

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

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

 MAC アドレスの一覧。
 戻り値: *String*。
 対応プラットフォーム: Linux。

パラメーター:

- **interface** - *all*（デフォルト）または正規表現;
- **format** - *full*（デフォルト）または *short*。

コメント:

- 指定した interface の正規表現に名前が一致するインターフェースの MAC アドレスを一覧表示します（*all* はすべてのインターフェースを一覧表示します)。
- format に *short* を指定した場合、インターフェース名と同一の MAC アドレスは表示されません。

例:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 システム時刻。
 戻り値: *Integer* - type が *utc* の場合; *String* - type が *local* の場合。
 supported platforms を参照してください。

パラメーター:

- **type** - 指定可能な値:
  - *utc* - (デフォルト) エポック (1970 年 1 月 1 日 00:00:00 UTC) からの経過秒数;
  - *local* - 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm' 形式の時刻。

Zabbix エージェントではpassive check として使用する必要があります。Zabbix agent 2 では active check として設定できます。

例:

```
system.localtime[local] # このキーを使用してアイテムを作成し、それを使ってホストの時刻を *Clock* ダッシュボードウィジェッ
```

system.run[command,<mode>]

 ホスト上で指定されたコマンドを実行します。
 戻り値: コマンドの *Text* 結果、または mode が *nowait* の場合は 1（コマンドの結果に関係なく)。
 対応プラットフォームを参照してください。

パラメーター:

- **command** - 実行するコマンド;
- **mode** - 指定可能な値:
  - *wait* - 実行終了まで待機する（デフォルト）;
  - *nowait* - 待機しない。

コメント:

- このアイテムはデフォルトでは無効です。有効化する方法はこちらを参照してください。
- アイテムの戻り値は、コマンドによって生成された標準出力と標準エラー出力です。終了コードのチェックは実行されません。
- 正しく処理されるには、コマンドの戻り値は text データ型である必要があります。空の結果も許可されます。
- 戻り値は 16MB に制限されます（末尾の空白は切り捨てられます)。データベースの制限も適用されます。
- 参照: コマンド実行。

例:

```
system.run[ls -l /] # ルートディレクトリの詳細なファイル一覧を返す
```

system.stat[resource,<type>]

 システム統計情報。
 戻り値: *Integer* または *float*。
 対応プラットフォーム: AIX。

パラメーター:

- **ent** - このパーティションが受け取る権利のあるプロセッサーユニット数 (float)。
- **kthr,<type>** - カーネルスレッドの状態に関する情報:
  - *r* - 実行可能なカーネルスレッドの平均数 (float);
  - *b* - Virtual Memory Manager の待機キューに配置されたカーネルスレッドの平均数 (float)。
- **memory,<type>** - 仮想メモリーおよび実メモリーの使用状況に関する情報:
  - *avm* - アクティブな仮想ページ (integer);
  - *fre* - 空きリストのサイズ (integer)。
- **page,<type>** - ページフォールトおよびページング活動に関する情報:
  - *fi* - 1 秒あたりのファイルページイン数 (float);
  - *fo* - 1 秒あたりのファイルページアウト数 (float);
  - *pi* - ページングスペースからページインされたページ数 (float);
  - *po* - ページングスペースへページアウトされたページ数 (float);
  - *fr* - 解放されたページ数 (ページ置換) (float);
  - *sr* - ページ置換アルゴリズムによってスキャンされたページ数 (float)。
- **faults,<type>** - トラップおよび割り込みの発生率:
  - *in* - デバイス割り込み (float);
  - *sy* - システムコール (float);
  - *cs* - カーネルスレッドのコンテキストスイッチ (float)。
- **cpu,<type>** - プロセッサー時間の使用率の内訳:
  - *us* - ユーザー時間 (float);
  - *sy* - システム時間 (float);
  - *id* - アイドル時間 (float);
  - *wa* - システムに未処理のディスク/NFS I/O 要求がある間のアイドル時間 (float);
  - *pc* - 消費された物理プロセッサー数 (float);
  - *ec* - 消費された割り当て容量の割合 (float);
  - *lbusy* - ユーザーおよびシステムレベルで実行中に発生した論理プロセッサー使用率の割合を示します (float);
  - *app* - 共有プールで利用可能な物理プロセッサーを示します (float)。
- **disk,<type>** - ディスク統計情報:
  - *bps* - ドライブに転送されたデータ量（読み取りまたは書き込み）をバイト/秒で示します (integer);
  - *tps* - 物理ディスク/テープに対して発行された 1 秒あたりの転送回数を示します (float)。

コメント:

- 次のアイテムには制限があることに注意してください:
  - `system.stat[cpu,app]` - AIX LPAR の”Shared” タイプでのみサポートされます;
  - `system.stat[cpu,ec]` - AIX LPAR の”Shared” および”Dedicated” タイプでサポートされます (”Dedicated” は常に 100 (percent) を返します);
  - `system.stat[cpu,lbusy]` - AIX LPAR の”Shared” タイプでのみサポートされます;
  - `system.stat[cpu,pc]` - AIX LPAR の”Shared” および”Dedicated” タイプでサポートされます;
  - `system.stat[ent]` - AIX LPAR の”Shared” および”Dedicated” タイプでサポートされます。

system.sw.arch

 ソフトウェアアーキテクチャ情報。
 戻り値: *String*。
 サポートされているプラットフォームを参照。

情報は `uname()` 関数から取得されます。

例:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 オペレーティングシステムの情報。
 戻り値: *String*。
 対応プラットフォーム: Linux、Windows。

パラメーター:

- **info** - 指定可能な値: *full*（デフォルト）、*short*、または *name*

info は次の場所から取得されます（なお、すべてのディストリビューションにすべてのファイルとオプションが存在するわけではありません）:

- Linux では /proc/version（*full*）;
- Linux では /proc/version_signature（*short*）;
- Linux で対応しているシステムでは /etc/os-release の PRETTY_NAME パラメーター、または /etc/issue.net（*name*）;
- Windows では HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion レジストリキー。

例:

```
system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11
system.sw.os[full] → [s|Windows 10 Enterprise 22621.1.asd64fre.ni_release.220506-1250 Build 22621.963]
```

system.sw.os.get

 オペレーティングシステムに関する詳細情報（バージョン、タイプ、ディストリビューション名、マイナーおよびメジャーバージョンなど）。
 戻り値：*JSON* 文字列。
 対応プラットフォーム：Linux、Windows。

system.sw.packages[<regexp>,<manager>,<format>]

 インストール済みパッケージの一覧です。
 戻り値: *Text*。
 対応プラットフォーム: Linux。

パラメータ:

- **regexp** - *all*（デフォルト）または正規表現;
- **manager** - *all*（デフォルト）またはパッケージマネージャー;
- **format** - *full*（デフォルト）または *short*。

コメント:

- 指定した正規表現に名前が一致するインストール済みパッケージを、アルファベット順に一覧表示します（*all* はすべてを一覧表示します）。
- 対応するパッケージマネージャー（実行されるコマンド）:
  - pkg (dpkg --get-selections);
  - pkgtool (ls /var/log/packages);
  - rpm (rpm -qa);
  - pacman (pacman -Q);
  - portage.
- format が *full* に指定されている場合、パッケージはパッケージマネージャーごとにグループ化されます（各マネージャーは、角括弧で囲まれた名前を先頭にした別々の行に表示されます）。
- format が *short* に指定されている場合、パッケージはグループ化されず、1 行で表示されます。

例:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 インストール済みパッケージの詳細一覧を返します。
 戻り値: *JSON string*。
 対応プラットフォーム: Linux。

パラメータ:

- **regexp** - *all* (デフォルト) または正規expression;
- **manager** - *all* (デフォルト) またはパッケージマネージャー（指定可能な値: *rpm*, *dpkg*, *pkgtool*, *pacman*, *portage*）。

コメント:

- 指定した正規表現に名前が一致するインストール済みパッケージの、整形されていない JSON を返します。
- 出力はオブジェクトの配列で、各オブジェクトには次のキーが含まれます: name, manager, version, size, architecture, buildtime, installtime（詳細はこちら）。

system.swap.in[<device>,<type>]

 スワップイン（デバイスからメモリへの読み込み）の統計情報。
 戻り値: *Integer*。
 対応プラットフォーム: Linux, FreeBSD, OpenBSD。

パラメーター:

- **device** - スワップに使用するデバイスを指定します（Linux のみ）または *all*（デフォルト）;
- **type** - 指定可能な値:
  - *count*（スワップイン回数、Linux 以外のプラットフォームではデフォルト）;
  - *sectors*（スワップインされたセクター数）;
  - *pages*（スワップインされたページ数、Linux ではデフォルト）。

コメント:

- この情報の取得元は次のとおりです:
  - /proc/swaps, /proc/partitions, /proc/stat（Linux 2.4）;
  - /proc/swaps, /proc/diskstats, /proc/vmstat（Linux 2.6）。
- *pages* は、device が指定されていない場合にのみ動作することに注意してください。
- *sectors* の type パラメーターは Linux でのみサポートされます。

例:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 スワップアウト（メモリからデバイスへの書き出し）の統計情報。
 戻り値: *Integer*。
 サポートされるプラットフォーム: Linux, FreeBSD, OpenBSD。

パラメーター:

- **device** - スワップに使用するデバイスを指定します（Linux のみ）または *all*（デフォルト）;
- **type** - 指定可能な値:
  - *count*（スワップアウト回数、Linux 以外のプラットフォームでのデフォルト）;
  - *sectors*（スワップアウトされたセクター数）;
  - *pages*（スワップアウトされたページ数、Linux でのデフォルト）。

コメント:

- この情報の取得元は次のとおりです:
  - /proc/swaps, /proc/partitions, /proc/stat（Linux 2.4）;
  - /proc/swaps, /proc/diskstats, /proc/vmstat（Linux 2.6）。
- *pages* は、device が指定されていない場合にのみ動作します。
- *sectors* の type パラメーターは Linux でのみサポートされます。

例:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 スワップ領域のサイズをバイト単位、または合計に対する割合で返します。
 戻り値: *Integer* - バイト単位の場合; *Float* - パーセンテージの場合。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows。

パラメータ:

- **device** - スワップに使用するデバイスを指定します（FreeBSD のみ）または *all*（デフォルト）;
- **type** - 指定可能な値:
  - *free*（空きスワップ領域、デフォルト）;
  - *pfree*（空きスワップ領域、パーセンテージ）;
  - *pused*（使用済みスワップ領域、パーセンテージ）;
  - *total*（スワップ領域の合計）;
  - *used*（使用済みスワップ領域）。

コメント:

- *pfree*、*pused* は、Windows で swap size が 0 の場合はサポートされないことに注意してください。
- device を指定しない場合、Zabbix エージェントはスワップデバイス（ファイル）のみを考慮し、物理メモリは無視します。たとえば Solaris システムでは、`swap -s` コマンドは物理メモリの一部とスワップデバイスを含みます（`swap -l` とは異なります）。

例:

```
system.swap.size[,pfree] → 空きスワップ領域の割合
```

system.uname

 システムの識別。
 戻り値: 文字列。
 サポートされているプラットフォームを参照してください。

コメント:

- UNIX では、このアイテムの値は uname() システムコールで取得されます。
- Windows では OS アーキテクチャを返し、UNIX では CPU アーキテクチャを返します。

例:

```
system.uname → FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386
system.uname → Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86
```

system.uptime

 システムの稼働時間 (秒単位)。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows。

アイテムの設定で、**s** または **uptime** の単位を使用すると、読みやすい値が得られます。

system.users.num

 ログインしているユーザー数。
 戻り値: 整数。
 サポートされているプラットフォームを参照してください。

値を取得するために、エージェント側で **who** コマンドが使用されます。

vfs.dev.discovery

 ブロックデバイスとそのタイプのリスト。低レベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux。

vfs.dev.get[<mode>,<regexp>]

 詳細情報を含むブロックデバイスの一覧です。低レベルディスカバリに使用されます。
 戻り値: *JSON* 文字列。
 サポートされるプラットフォーム: Linux, AIX.

パラメータ:

- **mode** - 指定可能な値:
  - *disks* (default) - ディスクの一覧を返します;
  - *disk_stats* - 各ディスクの統計情報を返します;
  - *devices* (linux only) - ブロックデバイス（ディスクとパーティション）の一覧を返します;
  - *device_stats* (linux only) - 各デバイス（ディスクとパーティション）の統計情報を返します。
- **regexp** - 名前でブロックデバイスをフィルタリングするために使用される正規表現です。

vfs.dev.read[<device>,<type>,<mode>]

 ディスク読み取り統計です。
 戻り値: - *Integer* - `type` が *sectors*、*operations*、*bytes* の場合; - *Float* - `type` が *sps*、*ops*、*bps* の場合。
 サポートされるプラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD.

パラメータ:

- **device** - ディスクデバイス（デフォルトは *all* [3]）;
- **type** - 指定可能な値: *sectors*、*operations*、*bytes*、*sps*、*ops*、または *bps*（*sps*、*ops*、*bps* はそれぞれ sectors、operations、bytes per second を意味します）;
- **mode** - 指定可能な値: *avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*。このパラメータは `type` が sps、ops、bps の場合にのみサポートされます。

コメント:

- 3 時間以上の更新間隔[2] を使用している場合、このアイテムは常に 0 を返します。
- *sectors* および *sps* の type パラメータは Linux でのみサポートされます。
- *ops* の type パラメータは Linux と FreeBSD でのみサポートされます。
- *bps* の type パラメータは FreeBSD でのみサポートされます。
- *bytes* の type パラメータは FreeBSD、Solaris、AIX、OpenBSD でのみサポートされます。
- `mode` パラメータは Linux、FreeBSD でのみサポートされます。
- 相対的なデバイス名（たとえば `sda`）に加えて、任意で `/dev/` プレフィックス（たとえば `/dev/sda`）も使用できます。
- LVM 論理ボリュームがサポートされています。
- OS ごとの'type' パラメータのデフォルト値:
  - *AIX* - operations;
  - *FreeBSD* - bps;
  - *Linux* - sps;
  - *OpenBSD* - operations;
  - *Solaris* - bytes.
- サポートされるプラットフォームにおける *sps*、*ops*、*bps* は 1024 デバイス（個別 1023 台と *all* 用 1 台）に制限されます。

例:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 ディスク書き込み統計です。
 戻り値: *Integer* - *sectors*、*operations*、*bytes* の場合; *Float* - *sps*、*ops*、*bps* の場合。
 サポートされるプラットフォーム: Linux, FreeBSD, Solaris, AIX, OpenBSD。

パラメーター:

- **device** - ディスクデバイス（デフォルトは *all* [3]）；
- **type** - 指定可能な値: *sectors*、*operations*、*bytes*、*sps*、*ops*、または *bps*（*sps*、*ops*、*bps* はそれぞれ sectors、operations、bytes per second を意味します）；
- **mode** - 指定可能な値: *avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*。このパラメーターは、type が sps、ops、bps の場合にのみサポートされます。

コメント:

- 更新間隔が 3 時間以上[2] の場合、このアイテムは常に 0 を返します。
- *sectors* および *sps* の type パラメーターは Linux でのみサポートされます。
- *ops* の type パラメーターは Linux と FreeBSD でのみサポートされます。
- *bps* の type パラメーターは FreeBSD でのみサポートされます。
- *bytes* の type パラメーターは FreeBSD、Solaris、AIX、OpenBSD でのみサポートされます。
- mode パラメーターは Linux、FreeBSD でのみサポートされます。
- 相対的なデバイス名（たとえば sda）に加えて、必要に応じて /dev/ プレフィックス（たとえば /dev/sda）も使用できます。
- LVM 論理ボリュームがサポートされています。
- OS ごとの type パラメーターのデフォルト値:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- サポートされるプラットフォームでの *sps*、*ops*、*bps* は 1024 デバイス（個別に 1023、*all* 用に 1）に制限されます。

例:

```
vfs.dev.write[,operations]
```

vfs.dir.count[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 ディレクトリエントリの数。
 戻り値: *Integer*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **dir** - ディレクトリへの絶対パス。
- **regex incl** - 含めるエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを表す正規表現。空の場合（デフォルト値）はすべてを含める。
- **regex excl** - 除外するエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを表す正規表現。空の場合（デフォルト値）は何も除外しない。
- **types incl** - カウントするディレクトリエントリのタイプ。指定可能な値:
  - *file* - 通常のファイル。
  - *dir* - サブディレクトリ。
  - *sym* - シンボリックリンク。
  - *sock* - ソケット。
  - *bdev* - ブロックデバイス。
  - *cdev* - キャラクターデバイス。
  - *fifo* - FIFO。
  - *dev* - ”bdev,cdev” と同義。
  - *all* - すべてのタイプ（デフォルト）。つまり、”file,dir,sym,sock,bdev,cdev,fifo”。複数のタイプはカンマで区切り、引用符で囲む必要があります。
- **types excl** - カウントしないディレクトリエントリのタイプ（types incl を参照）。types incl と types excl の両方に含まれるエントリタイプはカウントされません。
- **max depth** - 探索するサブディレクトリの最大深度:
  - **-1**（デフォルト）- 無制限。
  - **0** - サブディレクトリに降りない。
- **min size** - カウント対象となるファイルの最小サイズ（バイト）。これより小さいファイルはカウントされません。メモリサフィックスを使用できます。
- **max size** - カウント対象となるファイルの最大サイズ（バイト）。これより大きいファイルはカウントされません。メモリサフィックスを使用できます。
- **min age** - カウント対象となるディレクトリエントリの最小経過時間（秒）。これより新しいエントリはカウントされません。時間サフィックスを使用できます。
- **max age** - カウント対象となるディレクトリエントリの最大経過時間（秒）。この時間以上経過した古いエントリはカウントされません（変更時刻に基づく）。時間サフィックスを使用できます。
- **regex excl dir** - 除外するディレクトリの名前パターンを表す正規表現。そのディレクトリのすべての内容が除外されます（regex_excl とは異なります）。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクになる可能性があります。
- %APP_HOME%、$HOME、%TEMP% などの環境変数はサポートされていません。
- 疑似ディレクトリ”.” と”..” は決してカウントされません。
- ディレクトリ探索では、シンボリックリンクをたどりません。
- エントリ数の計算時には、`regex incl` と `regex excl` の両方がファイルとディレクトリに適用されますが、探索対象のサブディレクトリを選択する際には無視されます（`regex incl` が `(?i)^.+\.zip$` で、`max depth` が設定されていない場合、すべてのサブディレクトリが探索されますが、zip タイプのファイルのみがカウントされます）。
- 実行時間はエージェント設定のタイムアウト値（3 秒）によって制限されます。大規模なディレクトリの探索にはそれ以上の時間がかかる場合があるため、データは返されず、アイテムはサポート対象外になります。部分的なカウントは返されません。
- サイズでフィルタリングする場合、意味のあるサイズを持つのは通常のファイルのみです。Linux および BSD では、ディレクトリにも通常はゼロではないサイズ（一般的に数 KB）があります。デバイスのサイズはゼロです。たとえば、/dev/sda1 のサイズは対応するパーティションのサイズを示しません。そのため、<min_size> および <max_size> を使用する場合は、予期しない結果を避けるため、<types_incl> に*”file”* を指定することを推奨します。

例:

```
vfs.dir.count[/dev] #/dev 内のデバイス数を監視（Linux）
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] # 一時ディレクトリ内のファイル数を監視
```

vfs.dir.get[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 ディレクトリエントリの一覧。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **dir** - ディレクトリへの絶対パス。
- **regex incl** - 含めるエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを表す正規表現。空の場合（デフォルト値）はすべてを含めます。
- **regex excl** - 除外するエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを表す正規表現。空の場合（デフォルト値）は何も除外しません。
- **types incl** - 一覧表示するディレクトリエントリの種類。指定可能な値:
  - *file* - 通常のファイル。
  - *dir* - サブディレクトリ。
  - *sym* - シンボリックリンク。
  - *sock* - ソケット。
  - *bdev* - ブロックデバイス。
  - *cdev* - キャラクターデバイス。
  - *fifo* - FIFO。
  - *dev* - ”bdev,cdev” と同義。
  - *all* - すべての種類（デフォルト）。つまり、”file,dir,sym,sock,bdev,cdev,fifo”。複数の種類はカンマで区切り、引用符で囲む必要があります。
- **types excl** - 一覧表示しないディレクトリエントリの種類（`types incl` を参照）。`types incl` と `types excl` の両方に含まれる種類のディレクトリエントリは一覧表示されません。
- **max depth** - 探索するサブディレクトリの最大深度:
  - **-1**（デフォルト）- 無制限。
  - **0** - サブディレクトリに降りません。
- **min size** - 一覧表示するファイルの最小サイズ（バイト単位）。これより小さいファイルは一覧表示されません。メモリサフィックスを使用できます。
- **max size** - 一覧表示するファイルの最大サイズ（バイト単位）。これより大きいファイルは一覧表示されません。メモリサフィックスを使用できます。
- **min age** - 一覧表示するディレクトリエントリの最小経過時間（秒単位）。これより新しいエントリは一覧表示されません。時間サフィックスを使用できます。
- **max age** - 一覧表示するディレクトリエントリの最大経過時間（秒単位）。この時間以上古いエントリは一覧表示されません（変更時刻に基づきます）。時間サフィックスを使用できます。
- **regex excl dir** - 除外するディレクトリの名前パターンを表す正規表現。そのディレクトリのすべての内容が除外されます（`regex excl` とは異なります）。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクになる可能性があります。
- %APP_HOME%、$HOME、%TEMP% などの環境変数はサポートされていません。
- 疑似ディレクトリ”.” と”..” は一覧表示されません。
- ディレクトリ探索でシンボリックリンクをたどることはありません。

- エントリー一覧の生成時には、`regex incl` と `regex excl` の両方がファイルとディレクトリに適用されますが、探索するサブディレクトリの選択時には無視されます（`regex incl` が `(?i)^.+\.zip$` で、`max depth` が設定されていない場合、すべてのサブディレクトリが探索されますが、zip タイプのファイルのみがカウントされます）。
- 実行時間はエージェント設定のタイムアウト値によって制限されます。大規模なディレクトリの探索にはそれ以上の時間がかかる場合があるため、データは返されず、アイテムはサポート対象外になります。部分的な一覧は返されません。
- サイズでフィルタリングする場合、意味のあるサイズを持つのは通常のファイルだけです。Linux および BSD では、ディレクトリにもゼロではないサイズ（通常は数 KB）があります。デバイスのサイズはゼロです。たとえば、/dev/sda1 のサイズは対応するパーティションのサイズを反映しません。そのため、`min size` および `max size` を使用する場合は、予期しない結果を避けるため、`types incl` に*"file"* を指定することを推奨します。

例:

```
vfs.dir.get[/dev] #/dev 内のデバイス一覧を取得します（Linux）
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] # 一時ディレクトリ内のファイル一覧を取得します
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 ディレクトリのサイズ（バイト単位）。
 戻り値: *Integer*。
 対応プラットフォーム: Linux, Windows。このアイテムは他の UNIX 系プラットフォームでも動作する場合があります。

パラメーター:

- **dir** - ディレクトリへの絶対パス;
- **regex incl** - 含める対象の名前パターン（ファイル、ディレクトリ、シンボリックリンク）を記述する正規表現; 空の場合はすべて含める（デフォルト値）;
- **regex excl** - 除外する対象の名前パターン（ファイル、ディレクトリ、シンボリックリンク）を記述する正規表現; 空の場合は何も除外しない（デフォルト値）;
- **mode** - 使用可能な値:
  - *apparent*（デフォルト）- ディスク使用量ではなく見かけのファイルサイズを取得する（`du -sb dir` として動作する）;
  - *disk* - ディスク使用量を取得する（`du -s -B1 dir` として動作する）。du コマンドとは異なり、vfs.dir.size アイテムはディレクトリサイズを計算する際に隠しファイルも考慮する（dir 内で `du -sb .[^.]* *` として動作する）。
- **max depth** - 走査するサブディレクトリの最大深度:
  - **-1**（デフォルト）- 無制限;
  - **0** - サブディレクトリに降りない。
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。ディレクトリ内のすべてのコンテンツが除外される（`regex excl` とは異なる）

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクになる可能性があります。
- *zabbix* ユーザーに少なくとも読み取り権限があるディレクトリのみが計算されます。読み取り権限のみのディレクトリでは、ディレクトリ自体のサイズが計算されます。読み取りおよび実行権限があるディレクトリでは、内容を含めて計算されます。
- 大きなディレクトリや低速なドライブでは、エージェント およびサーバー/プロキシ の設定ファイルにある Timeout 設定により、このアイテムがタイムアウトする場合があります。必要に応じてタイムアウト値を増やしてください。
- ファイルサイズの上限はlarge file support に依存します。

例:

```
vfs.dir.size[/tmp,log] # /tmp 内で名前に 'log' を含むすべてのファイルのサイズを計算する
vfs.dir.size[/tmp,log,^.+\.old$] # /tmp 内で名前に 'log' を含むすべてのファイルのサイズを計算し、名前が '.old' で終わ
```

vfs.file.cksum[file,<mode>]

 UNIX の cksum アルゴリズムで計算されたファイルのチェックサム。
 戻り値: *Integer* - `mode` が *crc32* の場合、*String* - `mode` が *md5*、*sha256* の場合。
 サポートされるプラットフォーム を参照してください。

パラメーター:

- **file** - ファイルへの完全なパス;
- **mode** - *crc32* (デフォルト)、*md5*、または *sha256*。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスク となる可能性があります。
- ファイルサイズの上限は大きなファイルのサポート に依存します。

例:

```
vfs.file.cksum[/etc/passwd]
```

戻り値の例 (それぞれ crc32/md5/sha256):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 ファイルの内容を取得します[7]。
 戻り値: *Text*。
 supported platforms を参照してください。

パラメータ:

- **file** - ファイルへの完全なパス;
- **encoding** - コードページの識別子。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスク になる可能性があります。
- 戻り値は 16MB に制限されます（切り捨てられる末尾の空白を含む）。データベースの制限 も適用されます。
- ファイルが空、または LF/CR 文字のみを含む場合は、空文字列が返されます。
- バイトオーダーマーク（BOM）は出力から除外されます。

例:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 ファイルが存在するかどうかを確認します。
 戻り値: 0 - 見つからない; 1 - 指定した種類のファイルが存在する。
 supported platforms を参照してください。

パラメーター:

- **file** - ファイルへのフルパス;
- **types incl** - 含めるファイル種類の一覧。指定可能な値:
  - *file* (通常のファイル、デフォルト (`types_excl` が設定されていない場合));
  - *dir* (ディレクトリ);
  - *sym* (シンボリックリンク);
  - *sock* (ソケット);
  - *bdev* (ブロックデバイス);
  - *cdev* (キャラクターデバイス);
  - *fifo* (FIFO);
  - *dev* ("bdev,cdev" の同義語);
  - *all* (記載されているすべての種類、`types_excl` が設定されている場合のデフォルト)。
- **types excl** - 除外するファイル種類の一覧。指定可能な値については `types_incl` を参照してください (デフォルトでは除外される種類はありません)

コメント:

- Windows では、このアイテムは UNC パスに従います。これはsecurity risk になる可能性があります。
- 複数の種類はカンマで区切り、全体を引用符"" で囲む必要があります。
- 同じ種類が <types_incl> と <types_excl> の両方に含まれている場合、その種類のファイルは除外されます。
- ファイルサイズの上限はlarge file support に依存します。

例:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 ファイルに関する情報を返します。
 戻り値: *JSON string*。
 supported platforms を参照してください。

パラメーター:

- **file** - ファイルへの完全なパス

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスク になる可能性があります。
- UNIX 系システムでサポートされるファイルタイプ: 通常ファイル、ディレクトリ、シンボリックリンク、ソケット、ブロックデバイス、キャラクターデバイス、FIFO。
- ファイルサイズの上限はlarge file support に依存します。

例:

```
vfs.file.get[/etc/passwd] #return a JSON with information about the /etc/passwd file (type, user, permissi
```

vfs.file.md5sum[file]

 ファイルの MD5 チェックサム。
 戻り値: 文字列（ファイルの MD5 ハッシュ）。
 supported platforms を参照してください。

パラメーター:

- **file** - ファイルへの完全なパス

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスク になる可能性があります。
- ファイルサイズの上限はlarge file support に依存します。

例:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

返り値の例:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 ファイルの所有者を取得します。
 戻り値: *String*。
 supported platforms を参照してください。

パラメーター:

- **file** - ファイルへの完全パス;
- **ownertype** - *user* (デフォルト) または *group* (Unix のみ);
- **resulttype** - *name* (デフォルト) または *id*; id の場合、Unix では uid/gid を、Windows では SID を返します。

コメント:

- Windows では、このアイテムは UNC パスをたどります。これはsecurity risk になる可能性があります。
- ファイルサイズの上限はlarge file support に依存します。

例:

```
vfs.file.owner[/tmp/zabbix_server.log] # /tmp/zabbix_server.log のファイル所有者を返す
vfs.file.owner[/tmp/zabbix_server.log,,id] # /tmp/zabbix_server.log のファイル所有者 ID を返す
```

vfs.file.permissions[file]

 UNIX パーミッションを 8 進数で表した 4 桁の文字列を返します。
 戻り値: 文字列。
 対応プラットフォーム: Linux。他の UNIX 系プラットフォームでも動作する場合があります。

パラメータ:

- **file** - ファイルのフルパス

ファイルサイズの制限は大きなファイルのサポートに依存します。

例:

```
vfs.file.permissions[/etc/passwd] #/etc/passwd のパーミッションを返します。例えば'0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 ファイル内の文字列を取得します[7]。
 戻り値: 一致した文字列を含む行、またはオプションの output パラメータで指定された内容。
 対応プラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス;
- **regexp** - 必要なパターンを記述する正規expression;
- **encoding** - コードページのidentifier;
- **start line** - 検索を開始する最初の行番号（既定ではファイルの先頭行）;
- **end line** - 検索を終了する最後の行番号（既定ではファイルの最終行）;
- **output** - オプションの出力フォーマットテンプレート。**\0** エスケープシーケンスは一致したテキストの一部（一致が始まる最初の文字から一致が終わる文字まで）に置き換えられ、**\N**（N=1...9）は N 番目の一致グループに置き換えられます（N がキャプチャされたグループ数を超える場合は空文字列）。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの上限は大きなファイルのサポートに依存します。
- 最初に一致した行のみが返されます。
- 式に一致する行がない場合は空文字列が返されます。
- バイトオーダーマーク（BOM）は出力から除外されます。
- output パラメータを使用した内容の抽出はエージェント上で行われます。

例:

```
vfs.file.regexp[/etc/passwd,zabbix]
vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1]
vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] → ユーザー *zabbix* の ID を取得
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 ファイル内で文字列を検索します[7]。
 戻り値: 0 - 一致なし; 1 - 一致あり。
 supported platforms を参照してください。

パラメータ:

- **file** - ファイルへのフルパス;
- **regexp** - 必要なパターンを表す正規expression;
- **encoding** - コードページのidentifier;
- **start line** - 検索を開始する最初の行番号（既定ではファイルの先頭行）;
- **end line** - 検索を終了する最後の行番号（既定ではファイルの最終行）。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはsecurity risk となる可能性があります。
- ファイルサイズの上限はlarge file support に依存します。
- バイトオーダーマーク (BOM) は無視されます。

例:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 ファイルサイズ（バイト単位）。
 戻り値: *Integer*。
 supported platforms を参照してください。

Parameters:

- **file** - ファイルへの完全なパス。
- **mode** - 指定可能な値:
  - *bytes*（デフォルト）;
  - *lines*（空行もカウントされます）。

Comments:

- Windows では、このアイテムは UNC パスに従います。これはsecurity risk になる可能性があります。
- ファイルにはユーザー *zabbix* の読み取り権限が必要です。
- ファイルサイズの上限はlarge file support に依存します。

Example:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 ファイルの時刻情報。
 戻り値: *Integer* (Unix タイムスタンプ)。
 supported platforms を参照してください。

パラメータ:

- **file** - ファイルへの完全なパス;
- **mode** - 指定可能な値:
  - *modify* (default) - ファイル内容が最後に変更された時刻 (default);
  - *access* - ファイルが最後に読み取られた時刻;
  - *change* - ファイル属性が最後に変更された時刻。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはsecurity risk となる可能性があります。
- ファイルサイズの上限はlarge file support に依存します。

例:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 ファイルシステムのタイプとマウントオプションを含む、マウントされたファイルシステムのリスト。ローレベルディスカバリで使用されます。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows。

vfs.fs.get[<mode>,<mountpoint>]

 マウントされているファイルシステムの一覧と、そのタイプおよびマウントオプションを返します。*full* モードでは、使用可能なディスク領域と inode 統計情報も返します。ローレベルディスカバリに使用できます。
 戻り値: *JSON* 文字列。
 サポート対象プラットフォーム: Linux、FreeBSD、Solaris、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。

パラメータ:

- **mode** - 指定可能な値:
  - *full* (デフォルト) - ディスク領域および inode 統計情報を含むファイルシステムのプロパティ。
  - *short* - ディスク領域および inode 統計情報を含まない、ファイルシステムのプロパティのみ。
- **mountpoint** - データを返すマウントポイント (例: `/var` または `C:`)。指定しない場合、マウントされているすべてのファイルシステムが返されます。

コメント:

- *short* モードでは、ファイルシステム名、タイプ、マウントオプションのみが返されます (Windows では、名前、タイプ、ラベル、ドライブタイプ)。ファイルシステムの統計情報は照会されません。Unix 系システムでは、マウントされているファイルシステムの一覧のみが読み取られます (Linux の `/proc/mounts` など)。そのため、このチェックによって自動マウントがトリガーされることはなく、応答しないネットワークファイルシステムの影響も受けません。このため、*short* モードはローレベルディスカバリに適しています。
- **mountpoint** の値は、`fsname` プロパティと完全に一致する必要があります。Windows では大文字と小文字を区別せずに比較されます。末尾のバックスラッシュを付けずにマウントポイントを指定してください (例: `C:`)。
- **mountpoint** を指定した場合、そのマウントポイントのデータのみが返されます。一致するマウント済みファイルシステムがない場合、空の配列 (`[]`) が返されます。
- 同じディレクトリに複数のファイルシステムがマウントされている場合、出力に同じ `fsname` を持つエントリが複数含まれることがあります。
- *full* モードでは、Zabbix エージェントユーザーがマウントポイントのパスにあるすべてのディレクトリを通過する権限を持っている場合にのみ、使用可能なディスク領域と inode 統計情報が返されます。それ以外の場合、そのファイルシステムは出力から除外されます。これは *short* モードには適用されません。そのため、*short* モードでは *full* モードの出力に含まれないファイルシステムが一覧表示される場合があります。
- `fslabel` および `fsdrivetype` プロパティは Windows でのみ返されます。`inodes` および `options` プロパティは Windows では返されません。
- inode 数が 0 のファイルシステムも報告されます。これは、動的 inode を使用するファイルシステム (例: btrfs) で発生する場合があります。
- こちらも参照してください: マウントされているファイルシステムのディスカバリ。

例:

```
vfs.fs.get # 統計情報を含む、マウントされているすべてのファイルシステム。vfs.fs.get[full] と同じ
vfs.fs.get[short] # 統計情報を含まない、マウントされているすべてのファイルシステム
vfs.fs.get[full,/var] #/var マウントポイントのみの統計情報
vfs.fs.get[short,C:] #C: ボリュームのみのプロパティ (Windows)
```

vfs.fs.inode[fs,<mode>]

 inode の数またはパーセンテージ。
 戻り値: *Integer* - 数値の場合; *Float* - パーセンテージの場合。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **fs** - ファイルシステム;
- **mode** - 可能な値: *total* (デフォルト), *free*, *used*, *pfree* (free, パーセンテージ), *pused* (used, パーセンテージ)。

inode 数がゼロの場合（例えば btrfs のような動的 inode を持つファイルシステム）、pfree/pused の値はそれぞれ”100” と”0” として報告されます。

例:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 ディスク容量をバイト単位または全体に対するパーセンテージで取得します。
 戻り値: バイトの場合は *Integer*、パーセンテージの場合は *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **fs** - ファイルシステム;
- **mode** - *total* (デフォルト), *free*, *used*, *pfree* (空き容量のパーセンテージ), *pused* (使用容量のパーセンテージ) のいずれか。

コメント:

- ファイルシステムがマウントされていない場合、マウントポイントが存在するローカルファイルシステムのサイズを返します;
- ファイルシステムの予約領域は考慮され、*free* モードを使用した場合は含まれません。

例:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 メモリサイズをバイト単位、または総メモリに対する割合で返します。
 戻り値: バイトの場合は *Integer*、割合の場合は *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **mode** - 指定可能な値: *total* (デフォルト)、*active*、*anon*、*buffers*、*cached*、*exec*、*file*、*free*、*inactive*、*pinned*、*shared*、*slab*、*wired*、*used*、*pused* (使用中、割合)、*available*、または *pavailable* (利用可能、割合)。

コメント:

- このアイテムは 3 つのカテゴリのパラメータを受け付けます:
  - *total* - メモリの総量;
  - プラットフォーム固有のメモリ種別: *active*、*anon*、*buffers*、*cached*、*exec*、*file*、*free*、*inactive*、*pinned*、*shared*、*slab*、*wired*;
  - 使用中および利用可能なメモリ量に関するユーザーレベルの推定値: *used*、*pused*、*available*、*pavailable*。
- *active* モードパラメータは、FreeBSD、HP-UX、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *anon*、*exec*、*file* モードパラメータは、NetBSD でのみサポートされています。
- *buffers* モードパラメータは、Linux、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- *cached* モードパラメータは、Linux、FreeBSD、AIX、OpenBSD、NetBSD でのみサポートされています。
- *inactive*、*wired* モードパラメータは、FreeBSD、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *pinned* モードパラメータは、AIX でのみサポートされています。
- *shared* モードパラメータは、Linux 2.4、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- このアイテムについては、追加の詳細も参照してください。

例:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>,<redirect_limit>]

 Web ページのコンテンツを取得します。
 戻り値: ヘッダーを含む Web ページのソース（テキスト）。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（`scheme://host:port/path` 形式。*host* のみ必須）。使用できる URL スキーム: *http*、*https*[4]。スキームが指定されていない場合は、*http* として扱われます。URL を指定する場合、`path` と `port` は空にする必要があります。認証が必要なサーバーへの接続時にユーザー名とパスワードを指定する（例: `http://user:password@www.example.com`）には、cURL のサポートが必要です [4]。ホスト名ではPunycodeがサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは /）。
- **port** - ポート番号（HTTP の場合のデフォルトは 80）。
- **redirect_limit** - 追従するリダイレクトの最大数 [4]。0～1000（デフォルトは 0。リダイレクトには追従しません）。

コメント:

- `host` で指定されたリソースが存在しないか利用できない場合、または `redirect_limit` が 0 より大きく、リダイレクト数がその値を超えた場合、このアイテムはサポート対象外になります。
- `host` には、ホスト名、ドメイン名、IPv4 アドレス、IPv6 アドレスを指定できます。ただし、IPv6 アドレスを使用するには、IPv6 サポートを有効にして Zabbix エージェントをコンパイルする必要があります。
- リダイレクトに追従する場合、戻り値には各リダイレクト応答のヘッダーに続いて、最終応答のヘッダーと本文が含まれます。

例:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
web.page.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 完全な Web ページの読み込み時間（秒単位）。
 戻り値: *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（scheme://host:port/path 形式。*host* のみ必須）。使用できる URL スキーム: *http*、*https*[4]。スキームが指定されていない場合は、*http* として扱われます。URL を指定する場合、path と port は空にする必要があります。認証が必要なサーバーへの接続時にユーザー名とパスワードを指定する（例: http://user:password@www.example.com）には、cURL のサポートが必要です [4]。ホスト名では Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは/）。
- **port** - ポート番号（HTTP の場合のデフォルトは 80）。
- **redirect_limit** - 追従するリダイレクトの最大数 [4]。0～1000 の範囲で指定します（デフォルトは 0。リダイレクトには追従しません）。

コメント:

- host で指定されたリソースが存在しないか利用できない場合、または redirect_limit が 0 より大きく、リダイレクト数がその値を超えた場合、このアイテムはサポート対象外になります。
- host には、ホスト名、ドメイン名、IPv4 アドレス、IPv6 アドレスを指定できます。ただし、IPv6 アドレスを使用する場合、Zabbix エージェントは IPv6 サポートを有効にしてコンパイルする必要があります。
- リダイレクトに追従する場合、読み込み時間にはすべてのリダイレクトにかかった時間が含まれます。

例:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Web ページ上の文字列を検索します。
 戻り値: 一致した文字列、またはオプションの output パラメータで指定された文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（scheme://host:port/path 形式。*host* のみ必須）。使用できる URL スキーム: *http*、*https*[4]。スキームが指定されていない場合は、*http* として扱われます。URL を指定する場合、path と port は空にする必要があります。認証が必要なサーバーへの接続時にユーザー名とパスワードを指定する（例: http://user:password@www.example.com）には、cURL のサポートが必要です [4]。ホスト名では Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは /）。
- **port** - ポート番号（HTTP のデフォルトは 80）。
- **regexp** - 必要なパターンを記述する正規表現。
- **length** - 返す文字数の最大値。
- **output** - オプションの出力形式テンプレート。**\0** エスケープシーケンスは、テキストの一致部分（一致が始まる最初の文字から、一致が終わる文字まで）に置き換えられます。一方、**\N**（N=1...9）エスケープシーケンスは、N 番目に一致したグループに置き換えられます（N がキャプチャされたグループ数を超える場合は空の文字列）。
- **redirect_limit** - 追従するリダイレクトの最大数 [4]。0～1000 の範囲で指定します（デフォルトは 0 - リダイレクトには追従しません）。

コメント:

- host で指定されたリソースが存在しない、または利用できない場合、あるいは redirect_limit が 0 より大きく、リダイレクト数がその値を超えた場合、このアイテムはサポート対象外になります。
- host にはホスト名、ドメイン名、IPv4 アドレス、IPv6 アドレスを指定できます。ただし、IPv6 アドレスの場合、Zabbix エージェントは IPv6 サポートを有効にしてコンパイルする必要があります。
- output パラメータを使用したコンテンツの抽出は、エージェント上で実行されます。
- リダイレクトに追従する場合、各リダイレクト応答のヘッダーも検索されます。

例:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

agent.hostmetadata

 エージェントのホストメタデータ。
 戻り値: *String*。
 supported platforms を参照してください。

HostMetadata またはHostMetadataItem パラメータの値を返します。定義されていない場合は空文字列を返します。

エージェント.hostname

 エージェントのホスト名。
 戻り値: *String*。
 supported platforms を参照してください。

戻り値:

- パッシブチェックの場合 - エージェント設定ファイルのHostname パラメータで最初に列挙されているホスト名。
- アクティブチェックの場合 - 現在のホスト名。

agent.ping

 エージェントの可用性チェックです。
 戻り値: なし - 利用不可; 1 - 利用可能。
 supported platforms を参照してください。

ホストの利用不可を確認するには、nodata() トリガー関数を使用します。

agent.variant

 Zabbix エージェントのバリアント（Zabbix agent または Zabbix agent 2）。
 戻り値: 1 - Zabbix agent; 2 - Zabbix agent 2。
 サポートされているプラットフォームを参照。

agent.version

 Zabbix エージェントのバージョン。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

返される値の例:

```
6.0.3
```

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

 Zabbix サーバーまたはプロキシの内部メトリクスのセットを返します。ip と port が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON* 文字列。
 supported platforms を参照してください。

パラメータ:

- **ip** - リモートで問い合わせるサーバー/プロキシの IP/DNS/ネットワークマスクのリスト（デフォルトは 127.0.0.1）;
- **port** - リモートで問い合わせるサーバー/プロキシのポート（デフォルトは 10051）

コメント:

- このアイテムは、内部メトリクスの選択されたセットを返します。詳細は、Remote monitoring of Zabbix stats を参照してください。
- stats リクエストは、対象インスタンスの StatsAllowedIP server/proxy パラメータに □□□□ されているアドレスからのみ受け付けられることに注意してください。

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

 Zabbix サーバーまたはプロキシ上で遅延している、キュー内の監視対象アイテム数を返します。ip と port が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON string*。
 supported platforms を参照してください。

パラメータ:

- **ip** - リモートで問い合わせるサーバー/プロキシの IP/DNS/ネットワークマスクのリスト（デフォルトは 127.0.0.1）
- **port** - リモートで問い合わせるサーバー/プロキシのポート（デフォルトは 10051）
- **queue** - 定数（そのまま使用）
- **from** - 少なくともこれだけ遅延している（デフォルトは 6 秒）
- **to** - 最大でこれだけ遅延している（デフォルトは無限大）

stats リクエストは、対象インスタンスの'StatsAllowedIP' server/proxy パラメータに列挙されたアドレスからのみ受け付けられることに注意してください。

脚注

[1]Linux 固有の注記です。
Zabbix エージェントには、ファイルシステム /proc への読み取り専用アクセス権が必要です。
www.grsecurity.org のカーネルパッチにより、非特権ユーザーのアクセス権が制限されます。

[2] vfs.dev.read[]、vfs.dev.write[]: アイテムの値が 3 時間を超えてアクセスされない場合、Zabbix エージェントは「古い」デバイス接続を終了します。
これは、パスが動的に変化するデバイスがシステムに存在する場合や、デバイスが手動で削除された場合に発生することがあります。
また、これらのアイテムで更新間隔を 3 時間以上に設定した場合、常に「0」が返されることにも注意してください。

[3] vfs.dev.read[]、vfs.dev.write[]: 最初のパラメータにデフォルトの all を使用すると、キーは、sda、sdb などのすべてのブロックデバイスと、それらのパーティション（sda1、sda2、sdb3...）、さらにそれらのブロックデバイスまたはパーティションを基にした複数のデバイス（MD RAID）および論理ボリューム（LVM）を含む、集計統計を返します。
このような場合、返される値は絶対値ではなく、相対値（時間とともに変動する値）としてのみ扱ってください。

[4] HTTPS およびリダイレクト（redirect_limit が 0 より大きい場合）に対応しているのは、Zabbix エージェントが cURL 対応でコンパイルされている場合のみです。それ以外の場合、アイテムはサポート対象外になります。Zabbix エージェント 2 では、いずれの場合も cURL は必要ありません。

[5] Solaris システム（Solaris 10 6/06 以前を含む）では、バイト数、エラー数、使用率の統計がカーネルに保存または報告されないため、ループバックインターフェースでは bytes および errors の値はサポートされません。
ただし、net-snmp 経由で Solaris システムを監視している場合は、値が返されることがあります。net-snmp には、1997 年頃の cmu-snmp 由来のレガシーコードが含まれており、インターフェース統計からバイト値を読み取れない場合、ループバックインターフェースに存在するパケットカウンターに任意の値 308 を乗算して返すためです。
これは、パケットの平均長が 308 オクテットであるという仮定に基づいていますが、Solaris システムのループバックインターフェースにおける MTU の上限は 8892 バイトであるため、非常に大まかな推定です。
これらの値は、正しい値、または正確に近い値であると見なさないでください。
推定値にすぎません。
Zabbix エージェントは推測による処理を行いませんが、net-snmp はこれらのフィールドに値を返します。

[6] Solaris で /proc/pid/psinfo から取得されるコマンドラインは 80 バイトに制限されており、プロセスの起動時点でのコマンドラインが含まれます。

[7] vfs.file.contents[]、vfs.file.regexp[]、vfs.file.regmatch[] アイテムを使用して、ファイルの内容を取得できます。
機密情報を含む特定のファイルへのアクセスを制限する場合は、これらのファイルを閲覧する権限を持たないユーザーで Zabbix エージェントを実行してください。

コマンドラインユーティリティでの使用

zabbix_agentd や zabbix_get でアイテムキーをコマンドラインからテストまたは使用する場合、シェルの構文も考慮する必要があります。

たとえば、キーの特定のパラメータをダブルクォーテーションで囲む必要がある場合、ダブルクォーテーションを明示的にエスケープする必要があります。そうしないと、シェルによって特殊文字としてトリムされ、Zabbix ユーティリティに渡されません。

例:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

エンコーディング設定

取得したデータが破損しないように、encoding パラメータでチェック（例: vfs.file.contents）の処理に使用する正しいエンコーディングを指定できます。サポートされているエンコーディング（コードページ識別子）の一覧は、libiconv（GNU プロジェクト）のドキュメント、または Microsoft Windows SDK の "Code Page Identifiers" に関するドキュメントで確認できます。Microsoft は、一部のコードページを「マネージドアプリケーションでのみ使用可能」としている場合があります。これは、それらのコードページが.NET ランタイム経由でのみ公開され、ネイティブ Win32 API では利用できないことを意味します。Zabbix エージェントは独自のエンコーディング変換ロジックを実装しているため、ネイティブ Windows 関数で提供されていない場合でも、それらのコードページは Zabbix エージェントでサポートされます。

encoding パラメータでエンコーディングを指定しない場合、次の解決方法が適用されます。

- エンコーディングが指定されていない場合（または空の文字列の場合）、UTF-8 とみなされ、データは「そのまま」処理されます。
- BOM の解析 - アイテム vfs.file.contents、vfs.file.regexp、vfs.file.regmatch に適用されます。
  ファイルの先頭にあるバイト順マーク（BOM）を使用して、正しいエンコーディングの判別を試みます。
  BOM が存在しない場合は、代わりに標準の解決方法（上記を参照）が適用されます。

エージェントのアイテムのトラブルシューティング

パッシブチェックの場合、サーバーからエージェントへのリクエストが先にタイムアウトし、アイテムが値を取得できなくなるのを防ぐため、次の点に注意してください。

- エージェントのバージョンがサーバーのバージョンより古い場合、アイテムの設定（またはグローバルタイムアウト）の Timeout 値を、エージェント設定ファイルの Timeout 値より大きくする必要がある場合があります。
- エージェントのバージョンがサーバーのバージョンより新しい場合、サーバー設定ファイルの Timeout 値を、エージェント設定ファイルの Timeout 値より大きくする必要がある場合があります。

Zabbix エージェント 2

Zabbix エージェント 2 は、Unix およびWindows の Zabbix エージェントでサポートされるすべてのアイテムキーをサポートします。このページでは、Zabbix エージェント 2 でのみ使用できる追加のアイテムキーについて、属するプラグインごとに分類して説明します。

アイテムキーは、パラメータや追加情報を省略した形で一覧表示されています。アイテムキーをクリックすると、詳細を確認できます。

Item key	Description	Plugin
ceph.df.details	クラスタのデータ使用量とプール間の分布。	Ceph (loadable)
ceph.osd.stats	集計された OSD ごとの統計。	
ceph.osd.discovery	検出された OSD の一覧。	
ceph.osd.dump	OSD の使用しきい値と状態。	
ceph.ping	Ceph への接続を確立できるかどうかをテストします。	
ceph.pool.discovery	検出されたプールの一覧。	
ceph.status	クラスタ全体の状態。	
docker.container_info	コンテナーに関する低レベル情報。	Docker
docker.container_stats	コンテナーのリソース使用統計。	
docker.containers	コンテナーの一覧を返します。	
docker.containers.discovery	コンテナーの一覧を返します。低レベルディスカバリに使用します。	
docker.data.usage	現在のデータ使用量に関する情報。	
docker.images	イメージの一覧を返します。	
docker.images.discovery	イメージの一覧を返します。低レベルディスカバリに使用します。	
docker.info	システム情報。	
docker.ping	Docker デーモンが稼働しているかどうかをテストします。	
ember.get	必要なデバイスの結果を返します。	Ember+ (loadable)
memcached.ping	接続が有効かどうかをテストします。	Memcached
memcached.stats	STATS コマンドの出力を取得します。	
mongodb.cfg.discovery	検出された設定サーバーの一覧を返します。	MongoDB (loadable)
mongodb.collection.stats	指定したコレクションのさまざまなストレージ統計を返します。	
mongodb.collections.discovery	検出されたコレクションの一覧を返します。	
mongodb.collections.usage	コレクションの使用統計を返します。	
mongodb.connpool.stats	現在のデータベースインスタンスから、シャーディングクラスタまたはレプリカセットの他のメンバーへの、開いている送信接続に関する情報を返します。	
mongodb.db.stats	指定したデータベースのシステム状態を反映する統計を返します。	
mongodb.db.discovery	検出されたデータベースの一覧を返します。	
mongodb.jumbo_chunks.count	jumbo chunk の数を返します。	
mongodb.oplog.stats	oplog から取得したデータを使用して、レプリカセットの状態を返します。	
mongodb.ping	接続が有効かどうかをテストします。	
mongodb.rs.config	レプリカセットの現在の設定を返します。	
mongodb.rs.status	メソッドが実行されたメンバーの視点から見たレプリカセットの状態を返します。	
mongodb.server.status	データベースの状態を返します。	
mongodb.sh.discovery	クラスタ内に存在する検出済みシャードの一覧を返します。	
mongodb.version	データベースサーバーのバージョンを返します。	
mqtt.get	指定されたブローカーの特定のトピック、またはワイルドカード付きの複数トピックを購読し、公開を待機します。	MQTT
mssql.availability.group.get	可用性グループを返します。	MSSQL (loadable)
mssql.custom.query	カスタムクエリの結果を返します。	
mssql.db.get	利用可能なすべての MSSQL データベースを返します。	

Item key	Description	Plugin
mssql.job.status.get	ジョブの状態を返します。	
mssql.last.backup.get	すべてのデータベースの最新バックアップ時刻を返します。	
mssql.local.db.get	Always On 可用性グループとレプリカ（プライマリまたはセカンダリ）に参加しており、接続が確立されたサーバー上にあるデータベースを返します。	
mssql.mirroring.get	ミラーリング情報を返します。	
mssql.nonlocal.db.get	Always On 可用性グループとレプリカ（プライマリまたはセカンダリ）に参加しており、他のサーバー上にあるデータベースを返します（そのデータベースは、接続が確立された SQL Server インスタンス上のローカルではありません）。	
mssql.perfcounter.get	パフォーマンスカウンターを返します。	
mssql.ping	接続が有効かどうかをテストします。	
mssql.quorum.get	クォーラム情報を返します。	
mssql.quorum.member.get	クォーラムメンバーを返します。	
mssql.replica.get	レプリカを返します。	
mssql.version	MSSQL のバージョンを返します。	
mysql.custom.query	カスタムクエリの結果を返します。	MySQL
mysql.db.discovery	MySQL データベースの一覧を返します。	
mysql.db.size	データベースサイズ（バイト）。	
mysql.get_status_variables	グローバルステータス変数の値。	
mysql.ping	接続が有効かどうかをテストします。	
mysql.replication.discovery	MySQL レプリケーションの一覧を返します。	
mysql.replication.get_slave_status	レプリケーションの状態。	
mysql.version	MySQL のバージョン。	
net.dns.get	DNS クエリを実行し、詳細な DNS レコード情報を返します。	Network
nvml.device.count	GPU デバイスの数。	NVIDIA GPU (loadable)
nvml.device.decoder.utilization	GPU デバイスのデコーダー使用率。	
nvml.device.ecc.mode	GPU デバイスの ECC モード。	
nvml.device.encoder.stats.get	GPU デバイスのエンコーダー統計。	
nvml.device.encoder.utilization	GPU デバイスのエンコーダー使用率。	
nvml.device.energy.consumption	GPU デバイスの消費電力。	
nvml.device.errors.memory	GPU デバイスの ECC メモリエラー情報。	
nvml.device.errors.register	GPU デバイスの ECC レジスターエラー情報。	
nvml.device.fan.speed.avg	GPU デバイスのファン速度の平均。	
nvml.device.get	UUID と名前を含む検出済み GPU。	
nvml.device.graphics.frequency	GPU デバイスのグラフィックスクロック速度。	
nvml.device.memory.bar1.get	GPU デバイスの BAR1 メモリ情報。	
nvml.device.memory.fb.get	GPU デバイスのフレームバッファーメモリ情報。	
nvml.device.memory.frequency	GPU デバイスのメモリクロック速度。	
nvml.device.pci.utilization	GPU デバイスの PCI 使用率。	
nvml.device.performance.state	GPU デバイスのパフォーマンス状態。	
nvml.device.power.limit	GPU デバイスの電力制限。	
nvml.device.power.usage	GPU デバイスの消費電力使用量。	
nvml.device.serial	GPU デバイスのシリアル番号。	
nvml.device.sm.frequency	GPU デバイスのストリーミングマルチプロセッサーのクロック速度。	
nvml.device.temperature	GPU デバイスの温度。	
nvml.device.utilization	GPU デバイスの使用率統計。	
nvml.device.video.frequency	GPU デバイスのビデオクロック速度。	
nvml.system.driver.version	NVIDIA ドライバーバージョン。	
nvml.version	NVML ライブラリのバージョン。	
oracle.diskgroups.stats	Automatic Storage Management (ASM) ディスクグループの統計を返します。	Oracle
oracle.diskgroups.discovery	ASM ディスクグループの一覧を返します。	
oracle.archive.info	アーカイブログの統計。	
oracle.cdb.info	Container Database (CDB) の情報。	
oracle.custom.query	カスタムクエリの結果。	
oracle.datafiles.stats	データファイルの統計を返します。	
oracle.db.discovery	データベースの一覧を返します。	
oracle.fra.stats	Fast Recovery Area (FRA) の統計を返します。	
oracle.instance.info	インスタンスの統計。	
oracle.pdb.info	Pluggable Database (PDB) の情報。	
oracle.pdb.discovery	PDB の一覧を返します。	

Item key	Description	Plugin
oracle.pga.stats	Program Global Area (PGA) の統計を返します。	
oracle.ping	Oracle への接続を確立できるかどうかをテストします。	
oracle.proc.stats	プロセスの統計を返します。	
oracle.redolog.info	制御ファイルからのログファイル情報。	
oracle.sga.stats	System Global Area (SGA) の統計を返します。	
oracle.sessions.stats	セッションの統計を返します。	
oracle.sys.metrics	一連のシステムメトリック値を返します。	
oracle.sys.params	一連のシステムパラメーター値を返します。	
oracle.ts.stats	表領域の統計を返します。	
oracle.ts.discovery	表領域の一覧を返します。	
oracle.user.info	Oracle ユーザー情報を返します。	
oracle.version	データベースサーバーのバージョンを返します。	
pgsql.autovacuum.count	autovacuum ワーカーの数。	PostgreSQL (loadable)
pgsql.archive	アーカイブ済みファイルに関する情報。	
pgsql.bgwriter	データベースクラスタのチェックポイント総数を、チェックポイント種別ごとに分類したもの。	
pgsql.cache.hit	PostgreSQL バッファーキャッシュのヒット率。	
pgsql.connections	種類別の接続を返します。	
pgsql.custom.query	カスタムクエリの結果を返します。	
pgsql.db.age	データベース内で最も古い FrozenXID の経過時間。	
pgsql.db.bloating_tables	データベースごとの bloating table の数。	
pgsql.db.discovery	PostgreSQL データベースの一覧。	
pgsql.db.size	データベースサイズ（バイト）。	
pgsql.dbstat	データベースごとの統計を収集します。	
pgsql.dbstat.sum	クラスタ内のすべてのデータベースの集計データ。	
pgsql.locks	データベースごとの付与済みロックに関する情報。	
pgsql.oldest.xid	最も古い XID の経過時間。	
pgsql.ping	接続が有効かどうかをテストします。	
pgsql.queries	実行時間別のクエリメトリック。	
pgsql.replication.count	スタンバイサーバーの数。	
pgsql.replication.process	各送信プロセスごとの flush lag、write lag、replay lag。	
pgsql.replication.process.discovery	レプリケーションプロセス名の検出。	
pgsql.replication.recovery_role	リカバリ状態。	
pgsql.replication.status	レプリケーションの状態。	
pgsql.replication_lag.b	レプリケーション遅延（バイト）。	
pgsql.replication_lag.sec	レプリケーション遅延（秒）。	
pgsql.uptime	PostgreSQL の稼働時間（ミリ秒）。	
pgsql.version	PostgreSQL のバージョンを返します。	
pgsql.wal.stat	WAL 統計。	
redis.config	指定したパターンに一致する Redis インスタンスの設定パラメーターを取得します。	Redis
redis.info	INFO コマンドの出力を取得します。	
redis.ping	接続が有効かどうかをテストします。	
redis.slowlog.count	Redis の起動以降の slow log エントリ数。	
smart.attribute.discovery	S.M.A.R.T. デバイス属性の一覧を返します。	S.M.A.R.T.
smart.disk.discovery	S.M.A.R.T. デバイスの一覧を返します。	
smart.disk.get	S.M.A.R.T. デバイスで利用可能なすべてのプロパティを返します。	
systemd.unit.get	systemd ユニットのすべてのプロパティを返します。	Systemd
systemd.unit.info	systemd ユニット情報。	
systemd.unit.discovery	systemd ユニットとその詳細の一覧。	
web.certificate.get	証明書を検証し、証明書の詳細を返します。	Web certificates

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **<** **>** で囲まれたパラメータはオプションです。

ceph.df.details[connString,<user>,<apikey>]

 クラスタのデータ使用量とプール間の分布。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

ceph.osd.stats[connString,<user>,<apikey>]

 集約および OSD ごとの統計情報。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

ceph.osd.discovery[connString,<user>,<apikey>]

 検出された OSD のリスト。
 ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

ceph.osd.dump[connString,<user>,<apikey>]

 OSD の使用率のしきい値とステータス。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

ceph.ping[connString,<user>,<apikey>]

 Ceph への接続が確立できるかどうかをテストします。
 戻り値: *0* - 接続が切断されています (認証や設定の問題を含むエラーが発生した場合); *1* - 接続が成功しました。

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

ceph.pool.discovery[connString,<user>,<apikey>]

 検出されたプールのリスト。
 ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

ceph.status[connString,<user>,<apikey>]

 クラスタ全体のステータス。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, apikey** - Ceph に接続するためのユーザー名と API キー。

docker.container_info[<ID>,<info>]

 コンテナのローレベル情報。
 戻り値: ContainerInspect API 呼び出しの出力を JSON としてシリアライズしたもの。

パラメータ:

- **ID** - コンテナの ID または名前;
- **info** - 返される情報の量。サポートされる値: *short* (デフォルト) または *full*。

十分な権限を得るために、Agent 2 のユーザー ('zabbix') を'docker' グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.container_stats[<ID>]

 コンテナのリソース使用状況の統計情報です。
 戻り値: ContainerStats API 呼び出しの出力と CPU 使用率のパーセンテージを JSON としてシリアライズしたもの。

パラメータ:

- **ID** - コンテナの ID または名前

Agent 2 のユーザー (’zabbix’) は、十分な権限を得るために’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.containers

 コンテナのリスト。
 戻り値: ContainerList API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー (’zabbix’) を’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.containers.discovery[<options>]

 コンテナのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **options** - すべてのコンテナまたは実行中のコンテナのみをディスカバリするかどうかを指定します。サポートされる値: *true* - すべてのコンテナを返す; *false* - 実行中のコンテナのみを返す (デフォルト)。

十分な権限を得るために、Agent 2 のユーザー (’zabbix’) を’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.data.usage

 現在のデータ使用量に関する情報。
 戻り値: SystemDataUsage API 呼び出しの出力を JSON としてシリアル化したもの。

十分な権限を得るために、Agent 2 のユーザー（’zabbix’）を’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.images

 イメージのリストを返します。
 戻り値: ImageList API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー (’zabbix’) を’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.images.discovery

 イメージのリストを返します。ローレベルディスカバリで使用されます。
 戻り値 : *JSON* オブジェクト

Agent 2 のユーザー（’zabbix’）は、十分な権限を持つために’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.info

 システム情報。
 戻り値: SystemInfo API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー (’zabbix’) を’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

docker.ping

 Docker デーモンが生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている。

十分な権限を得るために、Agent 2 のユーザー (’zabbix’) を’docker’ グループに追加する必要があります。そうしないと、チェックは失敗します。

ember.get[<uri>,<path>]

 要求されたデバイスの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - Ember+ デバイスの URI。デフォルト: 127.0.0.1:9998;
- **path** - デバイスの OID パス。デフォルトは空で、ルートコレクションデータを返します。

memcached.ping[connString,<user>,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切断されている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Memcached のログイン認証情報。

memcached.stats[connString,<user>,<password>,<type>]

 STATS コマンドの出力を取得します。
 戻り値: *JSON* - 出力は JSON としてシリアライズされます。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - Memcached のログイン認証情報;
- **type** - 返される統計情報のタイプ: *items*、*sizes*、*slabs*、*settings*（デフォルトは空、一般的な統計情報を返します）。

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 指定したコレクションのさまざまなストレージ統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報;
- **database** - データベース名（デフォルト: admin）;
- **collection** - コレクション名。

mongodb.cfg.discovery[connString,<user>,<password>]

 検出された構成サーバーのリストを返します。
 戻り値: *JSON* オブジェクト。

パラメーター:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.collections.discovery[connString,<user>,<password>]

 検出されたコレクションのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.collections.usage[connString,<user>,<password>]

 コレクションの使用統計を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ :

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.connpool.stats[connString,<user>,<password>]

 現在のデータベースインスタンスからシャードクラスタまたはレプリカセットの他のメンバーへの開いているアウトゴーイング接続に関する情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.db.stats[connString,<user>,<password>,<database>]

 指定したデータベースのシステム状態を反映する統計情報を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報;
- **database** - データベース名（デフォルト: admin）。

mongodb.db.discovery[connString,<user>,<password>]

 検出されたデータベースのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 ジャンボチャンクの数を返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.oplog.stats[connString,<user>,<password>]

 oplog から取得したデータを使用して、レプリカセットのステータスを返します。
 戻り値 : *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.ping[connString,<user>,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.rs.config[connString,<user>,<password>]

 レプリカセットの現在の構成を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.rs.status[connString,<user>,<password>]

 このメソッドが実行されるメンバーの観点からレプリカセットのステータスを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.server.status[connString,<user>,<password>]

 データベースの状態を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.sh.discovery[connString,<user>,<password>]

 クラスタ内に存在する検出されたシャードのリストを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mongodb.version[connString,<user>,<password>]

 データベースサーバーのバージョンを返します。
 戻り値: *String*。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MongoDB のログイン認証情報。

mqtt.get[<broker url>,topic,<user>,<password>]

 指定したブローカーの特定のトピック、またはワイルドカードを含むトピックにサブスクライブし、公開を待機します。
 戻り値: トピックの内容に応じて異なります。ワイルドカードが使用されている場合は、トピックの内容を JSON として返します。

パラメーター:

- **broker url** - MQTT ブローカーの URL。形式は `protocol://host:port` で、クエリパラメーターは含めません（サポートされるプロトコル: `tcp`, `ssl`, `ws`）。値が指定されていない場合、エージェントは `tcp://localhost:1883` を使用します。プロトコルまたはポートが省略された場合は、既定のプロトコル (`tcp`) またはポート (1883) が使用されます;
- **topic** - MQTT トピック（必須）。ワイルドカード (+,#) がサポートされています;
- **user, password** - 認証情報（必要な場合）。

コメント:

- アイテムはactive check（'Zabbix agent (active)' アイテムタイプ）として設定する必要があります;
- TLS 暗号化証明書は、既定の場所（例: Ubuntu では `/etc/ssl/certs/` ディレクトリ）に保存することで使用できます。TLS では `tls://` スキームを使用してください。

mssql.availability.group.get[URI,<user>,<password>]

 可用性グループを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **URI** - MSSQL サーバーの URI（サポートされているスキーマは `sqlserver://`のみ）。埋め込み認証情報は無視されます。URI の一部としてインスタンス名を指定することができます。例: `sqlserver://localhost/InstanceName`（ポートなし）。ポートが指定されている場合、インスタンス名は無視されます;
- **user, password** - 保護された MSSQL サーバーに送信するユーザー名、パスワード。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **URI** - MSSQL サーバーの URI（サポートされているスキーマは `sqlserver://`のみ）。埋め込まれた認証情報は無視されます。URI の一部としてインスタンス名を指定することができます。例: `sqlserver://localhost/InstanceName`（ポートなし）。ポートが指定されている場合、インスタンス名は無視されます;
- **user, password** - 保護された MSSQL サーバーに送信するユーザー名、パスワード;
- **queryName** - `.sql` 拡張子なしで `Plugins.MSSQL.CustomQueriesDir` に設定されたカスタムクエリの名前;
- **args** - クエリに渡す 1 つまたは複数のカンマ区切りの引数。

詳細はMSSQL プラグインの readme を参照してください。

mssql.db.get

 利用可能なすべての MSSQL データベースを返します。
 戻り値 : *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.job.status.get

 ジョブのステータスを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.last.backup.get

 すべてのデータベースの最終バックアップ時刻を返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.local.db.get

 Always On 可用性グループおよびレプリカ (プライマリまたはセカンダリ) に参加しており、接続が確立されたサーバーに存在するデータベースを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグイン の readme を参照してください。

mssql.mirroring.get

 ミラーリング情報を返します。
 戻り値: *JSON* オブジェクト。

詳細は MSSQL プラグイン の readme を参照してください。

mssql.nonlocal.db.get

 Always On 可用性グループに参加しているデータベースおよび他のサーバーにあるレプリカ (プライマリまたはセカンダリ) を返します (接続が確立された SQL Server インスタンスにローカルではないデータベース)。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグイン の readme を参照してください。

mssql.perfcounter.get

 パフォーマンスカウンターを返します。
 戻り値: *JSON* オブジェクト。

詳細は MSSQL プラグイン の readme を参照してください。

mssql.ping

 データベースに ping を送信します。接続が正しく設定されているかテストします。
 戻り値: *1* - 生存, *0* - 非生存。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.quorum.get

 クォーラム情報を返します。
 戻り値 : *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.quorum.member.get

 クォーラムメンバーを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.replica.get

 レプリカを返します。
 戻り値 : *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.version

 MSSQL のバージョンを返します。
 戻り値: 文字列。

詳細については、MSSQL プラグインの readme を参照してください。

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **queryName** - カスタムクエリの名前。SQL ファイル名 (拡張子なし) と一致する必要があります;
- **args** - クエリに渡す 1 つまたは複数のカンマ区切りの引数。

詳細はMySQL プラグインの readme の *Custom queries* セクションを参照してください。

mysql.db.discovery[connString,<user>,<password>]

 MySQL データベースのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: LLD JSON 形式の”show databases” SQL クエリの結果。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.db.size[connString,<user>,<password>,<database name>]

 バイト単位のデータベースサイズ。
 戻り値: バイト単位で特定のデータベースに対して”select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?” SQL クエリの結果。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **database name** - データベース名。

mysql.get_status_variables[connString,<user>,<password>]

 グローバルステータス変数の値。
 戻り値: ”show global status” SQL クエリの結果を JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.ping[connString,<user>,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切断されている（AUTH や設定の問題を含むエラーが発生した場合）。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.replication.discovery[connString,<user>,<password>]

 MySQL レプリケーションのリストを返します。ローレベルディスカバリで使用されます。
 戻り値: `SHOW SLAVE STATUS` または `SHOW REPLICA STATUS` SQL クエリの結果を LLD JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 レプリケーションのステータス。
 戻り値: `SHOW SLAVE STATUS` または `SHOW REPLICA STATUS` SQL クエリの結果を JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **master host** - レプリケーションマスターホスト名。見つからない場合はエラーを返します。このパラメータが指定されていない場合は、すべてのホストが返されます。

mysql.version[connString,<user>,<password>]

 MySQL のバージョン。
 戻り値: *String* (MySQL インスタンスのバージョン)。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報。

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

DNS クエリを実行し、詳細な DNS レコード情報を返します。
 このアイテムは、`net.dns.record` Zabbix エージェントアイテムの拡張版で、より多くのレコードタイプと、カスタマイズ可能なフラグをサポートします。
 戻り値: *JSON object*

パラメーター:

- **ip** - DNS サーバーの IP アドレス（空欄の場合はデフォルトの DNS サーバーを使用）;
- **name** - クエリ対象の DNS 名;
- **type** - クエリするレコードタイプ（デフォルトは *SOA*）;
- **timeout** - リクエストのタイムアウト（秒単位、デフォルトは 1 秒）;
- **count** - リクエストの試行回数（デフォルトは 2）;
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp*（デフォルト）または *tcp*;
- **flags** - クエリに渡す、カンマ区切りの 1 つ以上の引数。

コメント:

- `type` に指定できる値は次のとおりです: *A*, *NS*, *MD*, *MF*, *CNAME*, *SOA*, *MB*, *MG*, *MR*, *NULL*, *PTR*, *HINFO*, *MINFO*, *MX*, *TXT*, *RP*, *AFSDB*, *X25*, *ISDN*, *RT*, *NSAPPTR*, *SIG*, *KEY*, *PX*, *GPOS*, *AAAA*, *LOC*, *NXT*, *EID*, *NIMLOC*, *SRV*, *ATMA*, *NAPTR*, *KX*, *CERT*, *DNAME*, *OPT*, *APL*, *DS*, *SSHFP*, *IPSECKEY*, *RRSIG*, *NSEC*, *DNSKEY*, *DHCID*, *NSEC3*, *NSEC3PARAM*, *TLSA*, *SMIMEA*, *HIP*, *NINFO*, *RKEY*, *TALINK*, *CDS*, *CDNSKEY*, *OPENPGPKEY*, *CSYNC*, *ZONEMD*, *SVCB*, *HTTPS*, *SPF*, *UINFO*, *UID*, *GID*, *UNSPEC*, *NID*, *L32*, *L64*, *LP*, *EUI48*, *EUI64*, *URI*, *CAA*, *AVC*, *AMTRELAY*。値は大文字のみで指定する必要があります。小文字または大文字小文字混在の値はサポートされません。
- 逆引き DNS ルックアップ（`type` が *PTR* に設定されている場合）では、DNS 名を逆順形式と非逆順形式の両方で指定できます (以下の例を参照)。PTR レコードが要求される場合、DNS 名は実際には IP アドレスです。
- `flags` に指定できる値は次のとおりです:
  - *cdflag* または *nocdflag*（デフォルト）;
  - *rdflag*（デフォルト）または *nordflag*;
  - *dnssec* または *nodnssec*（デフォルト）;
  - *nsid* または *nonsid*（デフォルト）;
  - *edns0*（デフォルト）または *noedns0*;
  - *aaflag* または *noaaflag*（デフォルト）;

    - *adflag* または *noadflag*（デフォルト）。flags の *dnssec* と *nsid* は、どちらも *edns0* を必要とするため、*noedns0* と組み合わせて使用することはできません。値は小文字のみで指定する必要があります。大文字または大文字小文字混在の値はサポートされません。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。
- 出力は、指定されたパラメーターに基づく DNS レコード情報を含むオブジェクトです（詳細はこちらを参照）。

例:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 GPU デバイスの数。
 戻り値: 整数。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのデコーダー使用率 (%)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの ECC モード情報 (現在、保留中)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのエンコーダ統計情報。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスエンコーダの使用率 (%)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 ドライバが最後にリロードされてからの、ミリジュール (mJ) 単位での GPU デバイスの総エネルギー消費量。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの README を参照してください。

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

 GPU デバイスの ECC メモリエラー情報 (修正済み、未修正)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの ECC レジスタエラー情報（修正済み、未修正）。
 戻り値：*JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの最大速度に対する平均ファン速度の割合。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.get

 UUID と名前で検出された GPU。
 戻り値：*JSON*。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのグラフィックスクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのフレームバッファメモリ統計情報 (total, reserved, free, used)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの BAR1 メモリ統計情報 (total, free, used)。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのメモリクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの PCI 使用率（送信/受信スループット、KBps 単位）。
 戻り値：*JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのパフォーマンス状態。
 戻り値：整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 ミリワット単位の GPU デバイスの電力制限。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスの現在の消費電力 (ミリワット単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.serial[<deviceUUID>]

 GPU デバイスのシリアル番号。
 戻り値：文字列。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのストリーミングマルチプロセッサのクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.temperature[<deviceUUID>]

 GPU デバイスの温度 (摂氏)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.device.utilization[<deviceUUID>]

 GPU デバイスの使用率統計（GPU/メモリの使用率、パーセンテージ)。
 戻り値：*JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

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

 GPU デバイスのビデオクロック速度 (MHz 単位)。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細はNVIDIA GPU プラグインの readme を参照してください。

nvml.system.driver.version

 NVIDIA ドライバーのバージョン。
 戻り値: 文字列。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

nvml.version

 NVML ライブラリのバージョン。
 戻り値: 文字列。

詳細については、NVIDIA GPU プラグインの readme を参照してください。

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Automatic Storage Management (ASM) ディスクグループの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
 スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。
 この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名;
- **diskgroup** - 問い合わせ対象の ASM ディスクグループ名。

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 ASM ディスクグループの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - プラグイン設定ファイル内でその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 アーカイブログの統計情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
 スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 先頭が開き括弧"(" の場合（先頭の空白は無視されます）。
 この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **destination** - 問い合わせる宛先の名前。

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

 コンテナデータベース（CDB）の情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:

    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
    - セッション名 - その名前がプラグイン設定ファイル内に存在する場合。
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。
- **database** - 問い合わせ対象のデータベース名。

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 カスタムクエリの結果です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、かつ `Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。
- **queryName** - カスタムクエリの名前。拡張子を除いた SQL ファイル名と一致する必要があります。
- **args** - クエリに渡す、カンマ区切りの 1 つまたは複数の引数。

コメント: - 返されたデータは自動的に JSON に変換されます。- クエリから直接 JSON を返すことは避けてください。プラグインが再度 JSON に変換しようとする際に破損するためです。

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 データファイルの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
 これにスキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。
 この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.db.discovery[connString,<user>,<password>,<service>]

 データベースの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、

いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.fra.stats[connString,<user>,<password>,<service>]

 Fast Recovery Area（FRA）の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。
    この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 インスタンスの統計情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。
    この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。

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

 プラガブルデータベース（PDB）の情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字・小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード。

- **service** - Oracle のサービス名。
- **database** - 問い合わせ対象のデータベース名。

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 PDB のリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 先頭が開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.pga.stats[connString,<user>,<password>,<service>]

 Program Global Area (PGA) の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内に存在する場合。
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
  - TNS 値 - 先頭が開き括弧"(" の場合（先頭の空白は無視されます）。
この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。

oracle.ping[connString,<user>,<password>,<service>]

 Oracle への接続を確立できるかどうかをテストします。
 戻り値: *1* - 接続成功; *0* - 接続切断（AUTH および設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを `user as sysdba` の形式で末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

例:

```
oracle.ping[tcp://127.0.0.1:1521,ZABBIX_MON,zabbix,xe]
oracle.ping[localhost,ZABBIX_MON,zabbix,xe]
oracle.ping[zbx_tns_example,ZABBIX_MON,zabbix,xe]
oracle.ping["(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=x
```

oracle.proc.stats[connString,<user>,<password>,<service>]

 プロセスの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
  - TNS 値 - 開き括弧”(” で始まる場合（先頭の空白は無視されます）。
    この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、`as sysrac` のいずれかを `user as sysdba` の形式で末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。

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

 制御ファイルから取得したログファイル情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、かつ `Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
  - TNS 値 - 開き括弧”(” で始まる場合（先頭の空白は無視されます）。
    この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。

oracle.sga.stats[connString,<user>,<password>,<service>]

 システムグローバル領域（SGA）の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧”(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを付加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 セッションの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:

    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。
      この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **lockMaxTime** - セッションを長時間ロックされているものとしてカウントするための、セッションロックの最大継続時間(秒)。デフォルト: 600 秒。

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 システムメトリクス値のセットを返します。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **connString** - 以下のいずれかを指定できます。
    - URI - 指定した名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、常に URI として扱われます。
    - セッション名 - プラグイン設定ファイルにその名前が見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定されており、上記の条件のいずれにも該当しない場合。
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを `user as sysdba` の形式で追加できます（ログインオプションでは大文字と小文字が区別されず、末尾に空白を含めることはできません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。
- **duration** - システムメトリクス値の取得間隔（秒）。指定可能な値: *60* - 長い間隔（デフォルト）、*15* - 短い間隔。

oracle.sys.params[connString,<user>,<password>,<service>]

 システムパラメータ値のセットを返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 表領域の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
    - セッション名 - その名前がプラグイン設定ファイル内に存在する場合。
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。

    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。
      この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。
- **tablespace** - 問い合わせ対象の表領域名。
  デフォルト（空のままで `type` が設定されている場合）：
    - "TEMP"（`type` が"TEMPORARY" に設定されている場合）
    - "USERS"（`type` が"PERMANENT" に設定されている場合）
- **type** - 問い合わせ対象の表領域のタイプ。
  デフォルト（`tablespace` が設定されている場合）："PERMANENT"。
- **conname** - 情報が必要なコンテナ名。

`tablespace`、`type`、または `conname` が省略された場合、このアイテムは一致するすべてのコンテナ（PDB および CDB を含む）の表領域統計情報を返します。

oracle.ts.discovery[connString,<user>,<password>,<service>]

 表領域のリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
    - TNS 値 - 先頭が開き括弧"(" の場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle パスワード。
- **service** - Oracle サービス名。

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

 Oracle ユーザー情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名;
- **username** - 情報が必要なユーザー名。小文字のユーザー名はサポートされていません。デフォルト: 現在のユーザー。

oracle.version[connString,<user>,<password>,<service>]

 データベースサーバーのバージョンを返します。
 戻り値: *String*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;

  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー - ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値 - 開き括弧"(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾に空白を含めてはいけません）。
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 autovacuum ワーカーの数。
 戻り値: *Integer*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 アーカイブされたファイルに関する情報。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 データベースクラスタのチェックポイント総数を、チェックポイントの種類ごとに分類して示します。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 PostgreSQL のバッファキャッシュヒット率。
 戻り値: *Float*。

パラメーター:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 種類ごとの接続を返します。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）;

- **queryName** - カスタムクエリの名前。拡張子を除いた SQL ファイル名と一致している必要があります;
- **args** - クエリに渡す、カンマ区切りの 1 つ以上の引数。

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

 データベース内で最も古い FrozenXID の経過時間（トランザクション数）。
 戻り値: *Integer*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 各データベースごとの肥大化したテーブル数。
 戻り値: *Integer*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 PostgreSQL データベースの一覧です。low-level discovery に使用されます。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 データベースサイズをバイト単位で返します。
 戻り値: *Integer*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 データベースごとの統計情報を収集します。低レベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 クラスター内のすべてのデータベースの集計データ。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 データベースごとの付与済みロックに関する情報です。低レベルディスカバリに使用されます。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;

- **database name** - データベース名 (省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます)。

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

 最も古い XID の経過時間（トランザクション数）。
 戻り値: *Integer*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます)。

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

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続は生きています; *0* - 接続は切断されています (AUTH や設定の問題を含む、何らかのエラーが発生した場合)。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名 (省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます)。

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

 実行時間でクエリのメトリクスを取得します。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）;
- **time period** - スロークエリ数の実行時間しきい値（正の整数である必要があります)。

pgsql.replication.count[uri,<username>,<password>]

 スタンバイサーバーの数。
 戻り値：整数。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.process[uri,<username>,<password>]

 各送信プロセスごとのフラッシュラグ、書き込みラグ、リプレイラグ。
 戻り値：*JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.process.discovery[uri,<username>,<password>]

 レプリケーションプロセス名のディスカバリ。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.recovery_role[uri,<username>,<password>]

 リカバリーステータス。
 戻り値: *0* - マスターモード; *1* - リカバリーがまだ進行中 (スタンバイモード)。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.status[uri,<username>,<password>]

 レプリケーションのステータス。
 戻り値: *0* - ストリーミングがダウン; *1* - ストリーミングがアップ; *2* - マスターモード。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication_lag.b[uri,<username>,<password>]

 レプリケーションラグ（バイト単位）
 戻り値：整数

パラメータ:

- **uri** - URI またはセッション名
- **username, password** - PostgreSQL の認証情報

pgsql.replication_lag.sec[uri,<username>,<password>]

 レプリケーションラグ（秒単位）
 戻り値：整数

パラメータ:

- **uri** - URI またはセッション名
- **username, password** - PostgreSQL の認証情報

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

 PostgreSQL の稼働時間をミリ秒単位で返します。
 戻り値: *Float*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 PostgreSQL のバージョンを返します。
 戻り値: *String*。

パラメーター:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

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

 WAL 統計情報。
 戻り値: *JSON object*。

パラメーター:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報;
- **database name** - データベース名（省略した場合は、`postgresql.conf` の`Plugins.PostgreSQL.Default.Database` の値が使用されます）。

redis.config[connString,<password>,<pattern>]

 パターンに一致する Redis インスタンスの設定パラメータを取得します。
 戻り値: *JSON* - グロブスタイルのパターンが使用された場合; パターンにワイルドカード文字が含まれていない場合は単一の値。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード;
- **pattern** - グロブスタイルのパターン (デフォルトは **)。

コメント:

- セキュリティ警告: `redis.config` は Redis の `CONFIG GET` コマンドを実行し、機密性の高い設定パラメータ (例えば `requirepass`、`masterauth`、TLS 関連の設定、その他のシークレット) を返す場合があります。デフォルトパターン (*) や広範なパターンを使用すると、Zabbix でアイテム値の表示権限を持つ誰にでもこれらのシークレットが公開される可能性があります。

redis.info[connString,<password>,<section>]

 INFO コマンドの出力を取得します。
 戻り値: *JSON* - 出力は JSON としてシリアライズされます。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード;
- **section** - 情報のセクション（デフォルトは *default*）。

redis.ping[connString,<password>]

 接続が生きているかどうかをテストします。
 戻り値: *1* - 接続が生きている; *0* - 接続が切れている (AUTH や設定の問題を含むエラーが発生した場合)。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード。

redis.slowlog.count[connString,<password>]

 Redis が起動してからのスローログエントリの数。
 戻り値: 整数。

パラメータ:

- **connString** - URI またはセッション名;
- **password** - Redis のパスワード。

smart.attribute.discovery

 S.M.A.R.T. デバイス属性の一覧を返します。
 戻り値: *JSON object*。

コメント:

- 次のマクロとその値が返されます: `{#NAME}`、`{#DISKTYPE}`、`{#ID}`、`{#ATTRNAME}`、`{#THRESH}`;
- HDD、SSD、NVME のドライブタイプがサポートされています。ドライブは単体でも RAID で組み合わせても使用できます。RAID の場合、`{#NAME}` には追加情報が付与されます。例: `{"{#NAME}": "/dev/sda cciss,2"}`。

smart.disk.discovery[<type>]

 S.M.A.R.T. デバイスの一覧を返します。
 戻り値: *JSON object*。

パラメーター:

- **type** - ディスクをスキャンするための値を指定します。使用可能な値: *id*, *name* (デフォルト)。*id* は Windows ではサポートされておらず、選択するとエラーを返します

コメント:

- 次のマクロとその値が返されます: `{#NAME}`, `{#DISKTYPE}`, `{#MODEL}`, `{#SN}`, `{#PATH}`, `{#ATTRIBUTES}`, `{#RAIDTYPE}`;
- HDD、SSD、NVME ドライブタイプがサポートされています。ドライブが RAID に属していない場合、`{#RAIDTYPE}` は空になります。`{#NAME}` には RAID の場合に追加情報が付きます。例: `{"{#NAME}": "/dev/sda cciss,2"}`。

smart.disk.get[<path>,<raid type>]

 S.M.A.R.T. デバイスで利用可能なすべてのプロパティを返します。
 戻り値: *JSON object*。

パラメーター:

- **path** - ディスクのパス。値として `{#PATH}` マクロを使用できます;
- **raid_type** - RAID の種類。値として `{#RAID}` マクロを使用できます

コメント:

- HDD、SSD、NVME ドライブタイプがサポートされています。ドライブは単体でも RAID に組み合わせても使用できます;
- データには smartctl のバージョンと呼び出し引数、および追加フィールドが含まれます:
disk_name - RAID 検出に必要な追加情報を含む名前を保持します。例: `{"disk_name": "/dev/sda cciss,2"}`
disk_type - HDD、SSD、または NVME のディスクタイプを保持します。例: `{"disk_type": "ssd"}`;
- パラメーターが指定されていない場合、このアイテムはすべてのディスクに関する情報を返します。

systemd.unit.get[unit name,<interface>]

 systemd ユニットのすべてのプロパティを返します。
 戻り値: *JSON object*。

パラメーター:

- **unit name** - ユニット名（名前を取得するには、アイテムプロトタイプで `{#UNIT.NAME}` マクロを使用するとよいでしょう）;

- **interface** - ユニットのインターフェース種別。指定可能な値: *Unit* (デフォルト)、*Service*、*Socket*、*Device*、*Mount*、*Automount*、*Swap*、*Target*、*Path*。

コメント:

- このアイテムは Linux プラットフォームでのみサポートされます;
- Unit インターフェースの LoadState、ActiveState、UnitFileState はテキストと整数で返されます: `"ActiveState":{"state":1,"text"`

systemd.unit.info[unit name,<property>,<interface>]

 systemd ユニットの情報。
 戻り値: *String*。

パラメータ:

- **unit name** - ユニット名（名前を検出するには、アイテムのプロトタイプで `{#UNIT.NAME}` マクロを使用するとよいでしょう）;
- **property** - ユニットのプロパティ（例: ActiveState（デフォルト）、LoadState、Description）;
- **interface** - ユニットのインターフェースタイプ（例: Unit（デフォルト）、Socket、Service）。

コメント:

- このアイテムは Linux プラットフォームでのみサポートされています;
- このアイテムを使用すると、dbus APIで説明されている特定のインターフェースタイプから、特定のプロパティを取得できます。

例:

```
systemd.unit.info["{#UNIT.NAME}"] # 検出された systemd ユニットのアクティブ状態（active、reloading、inactive、fai
systemd.unit.info["{#UNIT.NAME}",LoadState] # 検出された systemd ユニットのロード状態情報を収集
systemd.unit.info[mysqld.service,Id] # サービスの技術名（mysqld.service）を取得
systemd.unit.info[mysqld.service,Description] # サービスの説明（MySQL Server）を取得
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] # サービスが最後にアクティブ状態になった時刻（1562565036283
systemd.unit.info[dbus.socket,NConnections,Socket] # このソケットユニットから接続数を収集
```

systemd.unit.discovery[<type>]

 systemd ユニットとその詳細のリスト。ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **type** - 可能な値: *all*, *automount*, *device*, *mount*, *path*, *service* (デフォルト), *socket*, *swap*, *target*。

このアイテムは Linux プラットフォームでのみサポートされています。

web.certificate.get[hostname,<port>,<address>]

 証明書を検証し、証明書の詳細を返します。
 戻り値: *JSON object*。

パラメーター:

- **hostname** - IP または DNS のいずれかを指定できます。
URL スキーム（*https* のみ）、パス（無視されます）、およびポートを含めることができます。
 最初のパラメーターと 2 番目のパラメーターの両方でポートが指定されている場合、それらの値は一致している必要があります。
address（3 番目のパラメーター）が指定されている場合、hostname は SNI と hostname 検証にのみ使用されます。
- **port** - ポート番号（HTTPS のデフォルトは 443）;
- **address** - IP または DNS のいずれかを指定できます。指定されている場合、接続にはこれが使用され、hostname（1 番目のパラメーター）は SNI と host 検証に使用されます。1 番目のパラメーターが IP で 3 番目のパラメーターが DNS の場合、1 番目のパラメーターは接続に使用され、3 番目のパラメーターは SNI と host 検証に使用されます。

コメント:

- このアイテムは、host 設定で指定された宛先が存在しない、利用できない、または TLS ハンドシェイクが無効な証明書以外のエラーで失敗した場合、サポート対象外になります。
- 現在、AIA（Authority Information Access）X.509 拡張、CRL および OCSP（OCSP stapling を含む）、Certificate Transparency はサポートされていません。
- JSON 応答フィールド:
  - *x509*: X.509 証明書の詳細を含みます。
    - *version*: X.509 のバージョン（例: 3）。
    - *serial_number*: 証明書のシリアル番号。
    - *signature_algorithm*: 証明書の署名に使用されたアルゴリズム（例: SHA256-RSA）。
    - *issuer*: 証明書の発行者。
    - *not_before*: 証明書の有効開始日。
    - *not_after*: 証明書の有効期限。

        * *subject*: 証明書のサブジェクト。
        * *public_key_algorithm*: 公開鍵に使用されたアルゴリズム（例: RSA）。
        * *alternative_names*: サブジェクト代替名（存在する場合）。存在しない場合は null。
      - result: 検証結果を含みます。
        * *value*: 検証ステータス（以下の可能な値を参照）。
        * *message*: 詳細な検証メッセージ（例: "certificate verified successfully"）。
      - *sha1_fingerprint*: 証明書の SHA-1 フィンガープリント。
      - *sha256_fingerprint*: 証明書の SHA-256 フィンガープリント。
  - $.result.value フィールドは証明書の検証結果を示します。可能な値は次のとおりです。
      - *valid* - 証明書は有効で、信頼されています。
      - *valid-but-self-signed* - 証明書は有効ですが自己署名です。つまり、サブジェクトが発行者と一致します。
      - *invalid* - 有効期限切れ、ホスト名の不一致、または不明な署名機関などの問題により、証明書は無効です。

例:

```
web.certificate.get[example.com,443]
```

JSON 応答:


```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


Windows Zabbix エージェント

概要

Windows Zabbix エージェントのアイテムは、2 つのリストで表示されます。

- 共通アイテム - UNIX Zabbix エージェントと共通のアイテムキー
- Windows 固有のアイテム - **Windows** のみでサポートされているアイテムキー

Windows 上の Zabbix エージェントでサポートされているすべてのアイテムキーは、新世代の Zabbix エージェント 2 でもサポートされていることに注意してください。エージェント 2 でのみ使用できる追加のアイテムキーも参照してください。

参考: Windows アイテムの最小権限

共有アイテム

以下の表は、Windows でサポートされ、UNIX Zabbix エージェントと共有される Zabbix エージェントのアイテムを示しています。

- アイテムキーは、UNIX Zabbix エージェントのアイテムの詳細へのリンクです

- Windows に関連するアイテムのコメントが含まれています

Item key	Description	Item group
log	ログファイルを監視します。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	Log monitoring
log.count	監視対象のログファイル内で一致した行数を返します。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	
logrt	ローテーションされるログファイルを監視します。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	
logrt.count	ローテーションされる監視対象ログファイル内で一致した行数を返します。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	
modbus.get	Modbus データを読み取ります。	Modbus
net.dns	DNS サービスが稼働しているか確認します。 ip、timeout、count パラメータは、Zabbix agent 2 を使用しない限り Windows では無視されます。	Network
net.dns.perf	DNS サービスのパフォーマンスを確認します。 ip、timeout、count パラメータは、Zabbix agent 2 を使用しない限り Windows では無視されます。	
net.dns.record	DNS クエリを実行します。 ip、timeout、count パラメータは、Zabbix agent 2 を使用しない限り Windows では無視されます。	
net.if.discovery	ネットワークインターフェースの一覧です。 Windows の一部のバージョン（たとえば Server 2008）では、インターフェース名に非 ASCII 文字を使用するために最新の更新プログラムのインストールが必要な場合があります。	
net.if.in	ネットワークインターフェースの受信トラフィック統計です。 Windows では、利用可能な場合、アイテムは 64 ビットカウンターから値を取得します。64 ビットのインターフェース統計カウンターは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが利用できない場合、エージェントは 32 ビットカウンターを使用します。 Windows ではマルチバイトのインターフェース名がサポートされます。 ネットワークインターフェースの説明は、net.if.discovery または net.if.list アイテムで取得できます。	
net.if.out	ネットワークインターフェースの送信トラフィック統計です。 Windows では、利用可能な場合、アイテムは 64 ビットカウンターから値を取得します。64 ビットのインターフェース統計カウンターは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが利用できない場合、エージェントは 32 ビットカウンターを使用します。 Windows ではマルチバイトのインターフェース名がサポートされます。 ネットワークインターフェースの説明は、net.if.discovery または net.if.list アイテムで取得できます。	
net.if.total	ネットワークインターフェースの受信および送信トラフィック統計の合計です。 Windows では、利用可能な場合、アイテムは 64 ビットカウンターから値を取得します。64 ビットのインターフェース統計カウンターは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが利用できない場合、エージェントは 32 ビットカウンターを使用します。 Windows ではマルチバイトのインターフェース名がサポートされます。 ネットワークインターフェースの説明は、net.if.discovery または net.if.list アイテムで取得できます。	
net.tcp.listen	この TCP ポートが LISTEN 状態かどうかを確認します。	
net.tcp.port	指定したポートへの TCP 接続が可能かどうかを確認します。	
net.tcp.service	サービスが稼働しており、TCP 接続を受け付けているかどうかを確認します。 Windows での LDAP と HTTPS の確認は、Zabbix agent 2 のみがサポートします。	
net.tcp.service.perf	TCP サービスのパフォーマンスを確認します。 Windows での LDAP と HTTPS の確認は、Zabbix agent 2 のみがサポートします。	
net.tcp.socket.count	パラメータに一致する TCP ソケット数を返します。 このアイテムは Linux では Zabbix agent でサポートされますが、Windows では 64 ビット Windows 上のZabbix agent 2 でのみサポートされます。	

Item key	Description	Item group
net.udp.service	サービスが稼働しており、UDP リクエストに応答しているかどうかを確認します。	
net.udp.service.perf	UDP サービスのパフォーマンスを確認します。	
net.udp.socket.count	パラメータに一致する UDP ソケット数を返します。 このアイテムは Linux では Zabbix agent でサポートされますが、Windows では 64 ビット Windows 上のZabbix agent 2 でのみサポートされます。	
proc.get	OS プロセスとそのパラメータの一覧です。 cmdline パラメータは Windows ではサポートされません。	Processes
proc.num	プロセス数です。 Windows では name と user パラメータのみサポートされます。	
system.cpu.discovery	検出された CPU/CPU コアの一覧です。	System
system.cpu.load	CPU 負荷です。 Zabbix agent で collector プロセスが開始されると、次のパフォーマンスカウンターが初期化され、このアイテムで使用されます: \System\Processor Queue Length	
system.cpu.num	CPU 数です。	
system.cpu.util	CPU 使用率です。 値は *Processor Time* パフォーマンスカウンターを使用して取得されます。なお、Windows 8 以降のタスクマネージャーは *Processor Utility* パフォーマンスカウンターに基づいて CPU 使用率を表示しますが、それ以前のバージョンでは *Processor Time* カウンターが使用されていました（詳細 を参照）。 Windows でサポートされる type パラメータは *system* のみです。	
system.hostname	システムのホスト名です。 Windows では、このアイテムの値は GetComputerName()（**netbios** の場合）、GetComputerNameExA()（**fqdn** の場合）、または gethostname()（**host** の場合）関数から取得されます。 より詳細な説明 も参照してください。	
system.localtime	システム時刻です。	
system.run	指定したコマンドをホスト上で実行します。	
system.sw.arch	ソフトウェアアーキテクチャ情報です。	
system.sw.os	オペレーティングシステム情報です。	
system.sw.os.get	オペレーティングシステムの詳細情報（バージョン、種類、ディストリビューション名、マイナー/メジャーバージョンなど）です。	
system.swap.size	スワップ領域のサイズをバイト単位、または全体に対する割合で返します。 pused 型パラメータは Linux では Zabbix agent でサポートされますが、WindowsではZabbix agent 2 でのみサポートされます。 仮想化された（VMware ESXi、VirtualBox）Windows プラットフォームでは、このキーがスワップ領域のサイズ/割合を誤って報告する場合があることに注意してください。この場合、正しいスワップ領域の割合を取得するには perf_counter[\700(_Total)\702] キーを使用できます。	
system.uname	システムの識別情報です。 Windows では、このアイテムの値は Win32_OperatingSystem および Win32_Processor WMI クラスから取得されます。OS 名（エディションを含む）は、ユーザーの表示言語に翻訳される場合があります。Windows の一部のバージョンでは、商標記号や余分なスペースが含まれます。	
system.uptime	システム稼働時間を秒単位で返します。	
vfs.dir.count	ディレクトリエントリ数です。 Windows では、ディレクトリのシンボリックリンクはスキップされ、ハードリンクは 1 回だけカウントされます。	Virtual file systems
vfs.dir.get	ディレクトリエントリの一覧です。 Windows では、ディレクトリのシンボリックリンクはスキップされ、ハードリンクは 1 回だけカウントされます。	
vfs.dir.size	ディレクトリサイズです。 Windows では、すべてのシンボリックリンクはスキップされ、ハードリンクは 1 回だけ考慮されます。	
vfs.file.cksum	UNIX の cksum アルゴリズムで計算されたファイルチェックサムです。	
vfs.file.contents	ファイルの内容を取得します。	

Item key	Description	Item group
vfs.file.exists	ファイルが存在するかどうかを確認します。 Windows では、コマンドラインユーティリティで zabbix_get.exe または agent2 を呼び出す際、ダブルクォートはバックスラッシュ \\ でエスケープし、アイテムキー全体をダブルクォートで囲む必要があります。 存在しないディレクトリ内でディレクトリを検索すると、Windows ではアイテムがサポート対象外になる場合があることに注意してください。たとえば vfs.file.exists[C:\no\dir,dir]（ここで'no' は存在しない）などです。	
vfs.file.get	ファイルに関する情報を返します。 Windows でサポートされるファイルタイプ: 通常ファイル、ディレクトリ、シンボリックリンク	
vfs.file.md5sum	ファイルの MD5 チェックサムです。	
vfs.file.owner	ファイルの所有者を取得します。	
vfs.file.regexp	ファイル内の文字列を取得します。	
vfs.file.regmatch	ファイル内の文字列を検索します。	
vfs.file.size	ファイルサイズです。	
vfs.file.time	ファイルの時刻情報です。 Windows XP では vfs.file.time[file,change] が vfs.file.time[file,access] と等しくなる場合があります。	
vfs.fs.discovery	マウント済みファイルシステムの一覧と、その種類およびマウントオプションです。 {#FSLABEL} マクロは Windows でサポートされます。	
vfs.fs.get	マウント済みファイルシステムの一覧と、その種類、利用可能なディスク容量、inode 統計、およびマウントオプションです。 {#FSLABEL} マクロは Windows でサポートされます。	
vfs.fs.size	ディスク容量をバイト単位、または全体に対する割合で返します。	
vm.memory.size	メモリサイズをバイト単位、または全体に対する割合で返します。	Virtual memory
web.page.get	Web ページの内容を取得します。	Web monitoring
web.page.perf	Web ページ全体の読み込み時間です。	
web.page.regexp	Web ページ上の文字列を検索します。	
agent.hostmetadata	エージェントのホストメタデータです。	Zabbix
agent.hostname	エージェントのホスト名です。	
agent.ping	エージェントの到達性を確認します。	
agent.variant	Zabbix agent の種類（Zabbix agent または Zabbix agent 2）です。	
agent.version	Zabbix agent のバージョンです。	
zabbix.stats	Zabbix サーバーまたはプロキシの内部メトリクスのセットをリモートで返します。	
zabbix.stats (queue)	Zabbix サーバーまたはプロキシ上で遅延している監視対象アイテムのキュー内件数をリモートで返します。	

Windows 固有のアイテム

この表は、Windows Zabbix エージェントでのみサポートされているアイテムキーの詳細を示しています。

Windows 固有のアイテムは、同様のエージェントアイテムの概算的な対応物である場合があります。たとえば、Windows でサポートされている proc_info は、Windows でサポートされていない proc.mem アイテムに大まかに対応します。

アイテムキーは、完全なアイテムキーの詳細へのリンクです。

アイテムキー	説明	アイテムグループ
eventlog	Windows イベントログの監視。	ログ監視
eventlog.count	Windows イベントログ内の行数。	
net.if.list	ネットワークインターフェースのリスト（インターフェースタイプ、ステータス、IPv4 アドレス、説明を含む）。	ネットワーク
perf_counter	任意の Windows パフォーマンスカウンターの値。	パフォーマンスカウンター
perf_counter_en	英語での任意の Windows パフォーマンスカウンターの値。	
perf_instance.discovery	Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。	
perf_instance_en.discovery	英語のオブジェクト名を使用して検出された Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。	
proc_info	特定のプロセスに関するさまざまな情報。	プロセス

アイテムキー	説明	アイテムグループ
registry.data	Windows レジストリキー内の指定された値名のデータを返します。	レジストリ
registry.get	指定されたキーにある Windows レジストリの値またはキーのリスト。	
service.discovery	Windows サービスのリスト。	サービス
service.info	サービスに関する情報。	
services	サービスのリスト。	
vm.vmemory.size	バイト単位または合計に対するパーセンテージでの仮想メモリサイズ。	仮想メモリ
wmi.get	WMI クエリを実行し、最初に選択されたオブジェクトを返します。	WMI
wmi.getall	WMI クエリを実行し、全レスポンスを返します。	

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 イベントログの監視。
 戻り値: *Log*。

パラメータ:

- **name** - イベントログチャネルの名前（Event Viewer GUI の *Log Name*）;
- **regexp** - 必要なパターンを記述する正規表現（大文字小文字を区別）;
- **severity** - 深刻度を記述する正規表現（大文字小文字を区別しない）。このパラメータは、次の値に基づく正規表現を受け付けます: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose”（Windows Vista 以降で動作）;
- **source** - ソース識別子を記述する正規表現（大文字小文字を区別しない）;
- **eventid** - イベント識別子を記述する正規表現（複数可）（大文字小文字を区別）;
- **maxlines** - エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新しい行の最大数。このパラメータは、zabbix_agentd.conf の MaxLinesPerSecond の値を上書きします。
- **mode** - 指定可能な値:
  - *all*（デフォルト）;
  - *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。

コメント:

- このアイテムはアクティブチェック として設定する必要があります;
- エージェントは”Forwarded events” ログのイベントを送信できません;
- Windows Eventing 6.0 がサポートされています;
- このアイテムで Log 以外の情報の種類 を選択すると、ローカルタイムスタンプに加えて、ログの深刻度とソース情報も失われます;
- ログ監視 に関する追加情報も参照してください。

例:

```
eventlog[Application]
eventlog[Microsoft-Windows-Application-Experience/Program-Compatibility-Assistant]
eventlog[Security,,"Failure Audit",,^(529|680)$]
eventlog[System,,"Warning|Error"]
eventlog[System,,,,^1$]
eventlog[Windows PowerShell,,,,,,skip]
eventlog[System,,,,@TWOSHORT] # ここでは `TWOSHORT` という名前のカスタム正規表現が参照されています（*Matches regular
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Windows イベントログ内の行数をカウントします。
 戻り値: *Integer*。

パラメーター:

- **name** - イベントログチャネルの名前（Event Viewer GUI の *Log Name*）;
- **regexp** - 必要なパターンを記述する正規expression（大文字小文字を区別）;
- **severity** - 深刻度を記述する正規表現（大文字小文字を区別しない）。このパラメーターは、次の値に基づく正規表現を受け付けます: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose”（Windows Vista 以降で実行時）。
- **source** - ソース識別子を記述する正規表現（大文字小文字を区別しない）;
- **eventid** - イベント識別子を記述する正規表現（大文字小文字を区別）;
- **maxproclines** - エージェントが 1 秒あたりに解析する新しい行の最大数（10000 を超えることはできません）。デフォルト値はzabbix_agentd.conf の MaxLinesPerSecond の 10 倍です。
- **mode** - 可能な値:
  - *all*（デフォルト）;

- *skip* - 古いデータの処理をスキップします（新しく作成されたアイテムにのみ影響します）。

コメント:

- このアイテムはactive check として設定する必要があります;
- エージェントは”Forwarded events” ログのイベントを送信できません;
- Windows Eventing 6.0 がサポートされています;
- このアイテムで Log 以外のtype of information を選択すると、ローカルタイムスタンプに加えて、ログの深刻度とソース情報が失われます;
- log monitoring に関する追加情報も参照してください。

例:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 ネットワークインターフェースのリスト (インターフェースの種類、状態、IPv4 アドレス、説明を含む)。
 戻り値: テキスト。

コメント:

- マルチバイトのインターフェース名をサポート;
- 無効なインターフェースはリストされません;
- 一部のコンポーネントの有効化/無効化により、Windows のインターフェース名の順序が変更される場合があります;
- 一部の Windows バージョン (例: Server 2008) では、インターフェース名で非 ASCII 文字をサポートするために最新の更新プログラムが必要な場合があります。

perf_counter[counter,<interval>]

 任意の Windows パフォーマンスカウンターの値。
 戻り値: *Integer*、*float*、*string* または *text*（要求内容による）。

パラメーター:

- **counter** - カウンターへのパス。
- **interval** - 平均値を保存するための直近 N 秒。`interval` は 1 から 900 秒（両端を含む）の範囲で指定する必要があり、デフォルト値は 1 です。

コメント:

- このアイテムは UNC パスに従うため、セキュリティリスク となる可能性があります。
- `interval` は、CPU 使用率のように複数回のサンプルを必要とするカウンターに使用されます。そのため、チェックは毎回、直近”interval” 秒の平均値を返します。
- 利用可能なカウンターの一覧は Performance Monitor で取得できます。
- 参照: Windows performance counters。

perf_counter_en[counter,<interval>]

 英語表記の Windows パフォーマンスカウンターの値。
 戻り値: *Integer*、*float*、*string* または *text*（要求内容による）。

パラメーター:

- **counter** - 英語表記のカウンターへのパス。
- **interval** - 平均値を保存する直近 N 秒。`interval` は 1 から 900 秒まで（両端を含む）である必要があり、デフォルト値は 1 です。

コメント:

- このアイテムは UNC パスに従うため、セキュリティリスク となる可能性があります。
- `interval` は、CPU 使用率のように複数回のサンプルを必要とするカウンターに使用され、チェックのたびに直近”interval” 秒の平均値を返します。
- このアイテムは **Windows Server 2008/Vista** 以降でのみサポートされます。
- 英語文字列の一覧は､次のレジストリキーを参照することで確認できます: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`。

perf_instance.discovery[object]

 Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。
 ローレベルディスカバリで使用されます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **object** - オブジェクト名（ローカライズ済み）。

perf_instance_en.discovery[object]

 英語のオブジェクト名を使用して検出された Windows パフォーマンスカウンターのオブジェクトインスタンスのリスト。ローレベルディスカバリで使用されます。
 戻り値：*JSON* オブジェクト。

パラメータ:

- **object** - オブジェクト名（英語）。

proc_info[process,<attribute>,<type>]

 特定のプロセスに関するさまざまな情報を取得します。
 戻り値: *Float*。

パラメーター:

- **process** - プロセス名;
- **attribute** - 要求するプロセス属性;
- **type** - 表現タイプ（同じ名前のプロセスが複数存在する場合に有効）

コメント:

- 次の `attributes` がサポートされています:
  - *vmsize* (default) - プロセスの仮想メモリサイズ（Kbytes）;
  - *wkset* - プロセスのワーキングセットサイズ（プロセスが使用している物理メモリ量）（Kbytes）;
  - *pf* - ページフォールト数;
  - *ktime* - プロセスのカーネル時間（ミリ秒）;
  - *utime* - プロセスのユーザー時間（ミリ秒）;
  - *io_read_b* - I/O 操作中にプロセスが読み取ったバイト数;
  - *io_read_op* - プロセスが実行した読み取り操作数;
  - *io_write_b* - I/O 操作中にプロセスが書き込んだバイト数;
  - *io_write_op* - プロセスが実行した書き込み操作数;
  - *io_other_b* - 読み取りおよび書き込み以外の操作中にプロセスが転送したバイト数;
  - *io_other_op* - 読み取りおよび書き込み以外の、プロセスが実行した I/O 操作数;
  - *gdiobj* - プロセスが使用している GDI オブジェクト数;
  - *userobj* - プロセスが使用している USER オブジェクト数。
- 有効な `types` は次のとおりです:
  - *avg* (default) - <process> という名前のすべてのプロセスの平均値;
  - *min* - <process> という名前のすべてのプロセスの最小値;
  - *max* - <process> という名前のすべてのプロセスの最大値;
  - *sum* - <process> という名前のすべてのプロセスの値の合計。
- 64 ビットシステムでは、このアイテムを正しく動作させるには 64 ビット版の Zabbix エージェントが必要です。

例:

```
proc_info[iexplore.exe,wkset,sum] # すべての Internet Explorer プロセスが使用している物理メモリ量を取得
proc_info[iexplore.exe,pf,avg] #Internet Explorer プロセスのページフォールト数の平均を取得
```

registry.data[key,<value name>]

 Windows レジストリの指定したキーにある、指定した値名のデータを返します。
 戻り値: *Integer*, *string* または *text* (値の型による)

パラメーター:

- **key** - ルートキーを含むレジストリキー。ルートの省略形 (例: HKLM) を使用できます;
- **value name** - キー内のレジストリ値名 (既定では空文字列"")。値名が指定されていない場合は、既定値が返されます。

コメント:

- サポートされるルートの省略形:
  - HKCR - HKEY_CLASSES_ROOT
  - HKCC - HKEY_CURRENT_CONFIG
  - HKCU - HKEY_CURRENT_USER
  - HKCULS - HKEY_CURRENT_USER_LOCAL_SETTINGS
  - HKLM - HKEY_LOCAL_MACHINE
  - HKPD - HKEY_PERFORMANCE_DATA
  - HKPN - HKEY_PERFORMANCE_NLSTEXT
  - HKPT - HKEY_PERFORMANCE_TEXT
  - HKU - HKEY_USERS
- スペースを含むキーは二重引用符で囲む必要があります。

例:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] # このキーの既定値のデ
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] # このキー内の "Enable
```

registry.get[key,<mode>,<name regexp>]

 指定されたキーにある Windows レジストリの値またはキーの一覧です。
 戻り値: *JSON object*。

パラメータ:

- **key** - ルートキーを含むレジストリキー。ルートの省略形（例: HKLM）を使用できます（省略形の完全な一覧については `registry.data[]` のコメントを参照してください）。
- **mode** - 指定可能な値:
values（デフォルト）または *keys*;
- **name regexp** - 名前が正規表現に一致する値のみを検出します（デフォルト - すべての値を検出）。*mode* が *values* の場合にのみ使用できます。

スペースを含むキーは、二重引用符で囲む必要があります。

例:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
JSON には、キー、最後のサブキー、値名、値の種類、および値データの詳細が含まれます。
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] # このキー内のすべての値のデータを
JSON には、キー、最後のサブキー、値名、値の種類、および値データの詳細が含まれます。
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] # このキーのすべてのサブキーを返しま
JSON には、キーと最後のサブキーの詳細が含まれます。
```

service.discovery

 Windows サービスのリスト。
 ローレベルディスカバリで使用します。
 戻り値: *JSON* オブジェクト。

service.info[service,<param>]

 サービスに関する情報。
 戻り値: - *Integer* - `param` が *state*、*startup* の場合;

 state の場合は次のとおり: - 0 - 実行中 - 1 - 一時停止中 - 2 - 開始保留中 - 3 - 一時停止保留中 - 4 - 継続保留中 - 5 - 停止保留中 - 6 - 停止済み - 7 - 不明 - 255 - 該当するサービスなし

 startup の場合は次のとおり: - 0 - 自動 - 1 - 自動（遅延開始） - 2 - 手動 - 3 - 無効 - 4 - 不明 - 5 - 自動トリガー開始 - 6 - 自動（遅延）トリガー開始 - 7 - 手動トリガー開始 - *String* - `param` が *displayname*、*path*、*user* の場合; - *Text* - `param` が *description* の場合。

パラメーター:

- **service** - 実際のサービス名、または MMC の Services スナップインで表示される表示名;
- **param** - *state*（デフォルト）、*displayname*、*path*、*user*、*startup*、または *description*。

コメント:

- `service.info[service,state]` と `service.info[service]` は同じ情報を返す;
- `param` が *state* の場合のみ、このアイテムは存在しないサービスに対する値（255）を返す。

例:

```
service.info[SNMPTRAP] - SNMPTRAP サービスの状態;
service.info[SNMP Trap] - 同じサービスの状態。ただし、表示名を指定した場合;
service.info[EventLog,startup] - EventLog サービスの起動種類
```

services[<type>,<state>,<exclude>]

 サービスの一覧。
 戻り値: *0* - 空の場合; *Text* - サービスのリスト (改行で区切られる)。

パラメータ:

- **type** - *all* (デフォルト), *automatic*, *manual*, または *disabled*;
- **state** - *all* (デフォルト), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending*, または *paused*;
- **exclude** - 結果から除外するサービス。除外するサービスはダブルクォーテーションで囲み、カンマで区切り、スペースは入れません。

例:

```
services[,started] # 起動しているサービスのリストを返します;
services[automatic, stopped] # 起動しているべきなのに停止しているサービスのリストを返します;
services[automatic, stopped, "service1,service2,service3"] # 起動しているべきなのに停止しているサービスのリストを返
```

vm.vmemory.size[<type>]

 バイト単位または合計からのパーセンテージでの仮想メモリサイズ。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。

パラメータ:

- **type** - 指定可能な値: *available* (利用可能な仮想メモリ), *pavailable* (利用可能な仮想メモリ、パーセンテージ), *pused* (使用中の仮想メモリ、パーセンテージ), *total* (合計仮想メモリ、デフォルト), *used* (使用中の仮想メモリ)

コメント:

- 仮想メモリ統計の監視は以下に基づきます:
  - Windows 上の合計仮想メモリ (物理メモリ合計 + ページファイルサイズ);
  - Zabbix エージェントがコミットできる最大メモリ量;
  - システムまたは Zabbix エージェントの現在のコミット済みメモリ制限 (小さい方)。

例:

```
vm.vmemory.size[pavailable] # 利用可能な仮想メモリをパーセンテージで返す
```

wmi.get[<namespace>,<query>]

 WMI クエリを実行し、最初に選択されたオブジェクトを返します。
 戻り値: 整数、浮動小数点数、文字列またはテキスト (リクエストによる)。

パラメータ:

- **namespace** - WMI 名前空間;
- **query** - 単一オブジェクトを返す WMI クエリ。

WMI クエリはWQLで実行されます。

例:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] # 最初の物理ディスク
```

wmi.getall[<namespace>,<query>]

 WMI クエリを実行し、応答全体を返します。ローレベルディスカバリで使用できます。
 戻り値: *JSON object*

パラメータ:

- **namespace** - WMI 名前空間。
- **query** - WMI クエリ。

コメント:

- WMI クエリは WQL を使用して実行されます。
- 返された JSON 内のより具体的な値を指定するために、JSONPath の前処理を使用できます。

例:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #returns status inf
```

Windows サービスの監視

このチュートリアルでは、Windows サービスの監視を設定する手順を説明します。Zabbix サーバーとエージェントが設定され、稼働していることを前提としています。

ステップ 1

サービス名を取得します。

MMC の Services スナップインを開き、対象サービスのプロパティを表示すると、サービス名を確認できます。*General* タブに、*Service name* という項目があります。その後に表示される値が、監視用のアイテムを設定する際に使用する名前です。たとえば、"workstation" サービスを監視したい場合、サービス名は **lanmanworkstation** になります。

ステップ 2

サービスの監視のためにアイテムを設定します。

アイテム `service.info[service,<param>]` は、特定のサービスに関する情報を取得します。必要な情報に応じて、`param` オプションを指定します。`param` には *displayname*、*state*、*path*、*user*、*startup*、*description* のいずれかの値を指定できます。`param` を指定しない場合 (`service.info[service]`)、デフォルト値は *state* です。

返される値の型は、選択した `param` によって異なります。*state* および *startup* の場合は整数、*displayname*、*path*、*user* の場合は文字列、*description* の場合はテキストです。

例:

- キー: `service.info[lanmanworkstation]`
- 情報の型: 数値（整数）

アイテム `service.info[lanmanworkstation]` は、サービスの状態を数値として取得します。フロントエンドで数値をテキスト表現にマッピングするには（例: "0" を"Running"、"1" を"Paused" など）、アイテムが設定されているホストで値のマッピングを設定できます。これを行うには、*Windows services by Zabbix agent* または *Windows services by Zabbix agent active* テンプレートをホストにリンクするか、または該当テンプレートで設定されている *Windows service state* の値のマッピングを基に、ホストで新しい値のマッピングを設定します。

なお、上記の両テンプレートには、サービスを自動的に検出するディスカバリルールが設定されています。これを望まない場合は、テンプレートをホストにリンクした後、ホストレベルでディスカバリルールを無効化できます。

Windows サービスのディスカバリ

ローレベルディスカバリは、コンピュータ上のさまざまなエンティティのアイテム、トリガー、グラフを自動的に作成する方法を提供します。Zabbix は、サービスの正確な名前を知る必要や、各サービスごとに手動でアイテムを作成することなく、マシン上の Windows サービスの監視を自動的に開始できます。フィルタを使用して、関心のあるサービスのみの実際のアイテム、トリガー、グラフを生成することができます。

ログファイル監視

概要

Zabbix は、ログローテーション対応の有無にかかわらず、ログファイルの集中監視と分析に使用できます。

通知を使用すると、ログファイルに特定の文字列または文字列パターンが含まれている場合にユーザーへ警告できます。

ログファイルを監視するには、次のものが必要です。

- ホスト上で Zabbix エージェントが稼働していること。
- ログ監視アイテムが設定されていること。

> **Attention:**
> 監視対象のログファイルのサイズ上限は、large file support に依存します。

設定

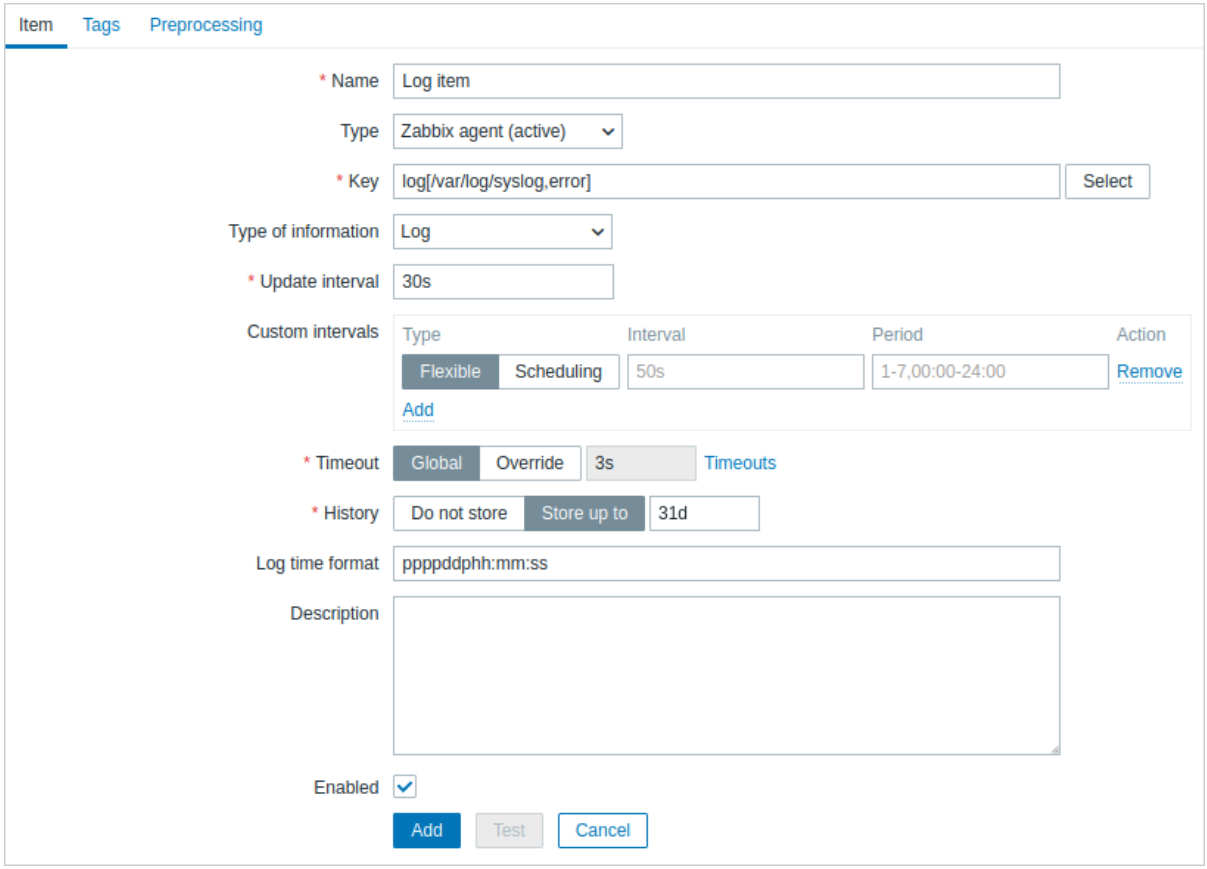
エージェントパラメータの確認

エージェント設定ファイルで、次を確認してください。

- `Hostname` パラメータが Web インターフェース上のホスト名と一致していること。
- `ServerActive` パラメータのサーバーが、アクティブチェックの処理対象として指定されていること。

アイテムの設定

ログ監視アイテムを設定します。



すべての必須入力フィールドには、赤いアスタリスクが付いています。

ログ監視アイテムでは、具体的に次の項目を入力します。

タイプ	ここでは **Zabbix** エージェント（アクティブ）を選択します。
キー	次のいずれかのアイテムキーを使用します。 `log[]` または `logrt[]`: これら 2 つのアイテムキーを使用すると、ログを監視し、指定されている場合は内容の正規表現によってログエントリをフィルターできます。 例: `log[/var/log/syslog,error]`。ファイルに'zabbix' ユーザーの読み取り権限があることを確認してください。そうでない場合、アイテムのステータスは「サポートされていない」になります。 **log.count[]** または **logrt.count[]**: これら 2 つのアイテムキーを使用すると、一致する行数のみを返すことができます。 これらのアイテムキーとパラメータの使用方法については、サポートされているZabbix エージェントアイテムキーのセクションを参照してください。
データ型	自動的に入力されます。 `log[]` または `logrt[]` アイテムの場合 - ログ; `log.count[]` または `logrt.count[]` アイテムの場合 - 数値（符号なし）。 オプションで `output` パラメータを使用する場合は、ログ以外の適切なデータ型を手動で選択できます。 ログ以外のデータ型を選択すると、ローカルタイムスタンプが失われることに注意してください。
更新間隔（秒）	このパラメータでは、Zabbix エージェントがログファイルの変更を確認する頻度を定義します。1 秒に設定すると、新しいレコードをできるだけ早く取得できます。

ログの時間形式	このフィールドでは、ログ行のタイムスタンプを解析するパターンを任意で指定できます。サポートされているプレースホルダー: * **y**: 年（*1970-2038*） * **M**: 月（*01-12*） * **d**: 日（*01-31*） * **h**: 時（*00-23*） * **m**: 分（*00-59*） * **s**: 秒（*00-59*） 空白のままにすると、タイムスタンプは Unix 時間の 0 に設定され、`January 1, 1970` を表します。 たとえば、次の Zabbix エージェントログファイルの行を考えてみます。 `23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).` この行は PID 用の 6 文字の位置で始まり、その後に日付、時刻、残りのメッセージが続きます。 この行のログ時間形式は”pppppp:yyyyMMdd:hhmmss” になります。 ”p” と”:” の文字はプレースホルダーであり、”yMdhms” 以外の任意の文字を使用できることに注意してください。

重要な注意事項

- サーバーとエージェントは、監視対象ログのサイズと最終更新時刻（`logrt` の場合）を 2 つのカウンターで保持します。さらに:
  - エージェントは、ログファイルが切り詰められたりローテーションされたりした際の判定を改善するために、内部的に inode 番号（UNIX/GNU/Linux）、ファイルインデックス（Microsoft Windows）、およびログファイル先頭 512 バイトの MD5 合計値も使用します。
  - UNIX/GNU/Linux システムでは、ログファイルが保存されているファイルシステムが inode 番号を報告し、それを使ってファイルを追跡できるものと想定されています。
  - Microsoft Windows では、Zabbix エージェントがログファイルの存在するファイルシステム種別を判定し、次を使用します:
    * NTFS ファイルシステムでは 64 ビットのファイルインデックス。
    * ReFS ファイルシステムでは（Microsoft Windows Server 2012 以降のみ）128 ビットのファイル ID。
    * ファイルインデックスが変化するファイルシステム（例: FAT32、exFAT）では、ログファイルのローテーションにより最終更新時刻が同じ複数のログファイルが生成された場合に、不確実な条件下でも妥当な判断を行うためのフォールバックアルゴリズムを使用します。
  - inode 番号、ファイルインデックス、および MD5 合計値は、Zabbix エージェントによって内部的に収集されます。これらは Zabbix サーバーには送信されず、Zabbix エージェントの停止時に失われます。
  - ログファイルの最終更新時刻を変更しないでください（たとえば `touch` を使用しないでください）。また、監視対象のログファイルをコピーして元の名前で置き換えないでください（これにより新しい inode が作成されます）。いずれの場合も、Zabbix はそのファイルを別のファイルとして扱い、先頭から再読み込みすることがあり、その結果、重複した障害通知が発生する可能性があります。
  - `logrt[]` アイテムに一致するログファイルが複数あり、Zabbix エージェントがそのうち最新のものを追跡していて、その最新のログファイルが削除された場合、警告メッセージ `"there are no files matching "<regexp mask>" in "<directory>"` が記録されます。Zabbix エージェントは、チェック対象の `logrt[]` アイテムについて、エージェントがこれまでに見た最も新しい最終更新時刻よりも古い更新時刻のログファイルを無視します。
- エージェントは、前回停止した位置からログファイルの読み取りを開始します。
- すでに解析済みのバイト数（サイズカウンター）と最終更新時刻（時間カウンター）は Zabbix データベースに保存され、エージェントが起動直後である場合や、以前は無効化されていた、または未サポートだったアイテムを受信した場合に、エージェントがこの位置からログファイルの読み取りを開始できるように送信されます。ただし、エージェントがサーバーからゼロ以外のサイズカウンターを受信していても、`logrt[]` または `logrt.count[]` アイテムが一致するファイルを見つけられない場合、後でファイルが現れたときに先頭から解析できるよう、サイズカウンターは 0 にリセットされます。
- ログファイルがエージェントに認識されているログサイズカウンターより小さくなった場合、カウンターは 0 にリセットされ、エージェントは時間カウンターを考慮しながらログファイルの先頭から読み取りを開始します。
- ディレクトリ内に最終更新時刻が同じ一致ファイルが複数ある場合、エージェントは同じ更新時刻を持つすべてのログファイルを正しく解析し、データの取りこぼしや同じデータの二重解析を避けようとしますが、すべての状況で保証されるわけではありません。エージェントは特定のログファイルローテーション方式を前提とせず、またそれを判定もしません。同じ最終更新時刻を持つ複数のログファイルがある場合、エージェントは辞書順の降順で処理します。そのため、ローテーション方式によってはログファイルが元の順序で解析・報告されます。別のローテーション方式では元のログファイル順が維持されず、結果として一致したログファイルのレコードが変更された順序で報告されることがあります（ログファイルの最終更新時刻が異なれば、この問題は発生しません）。
- Zabbix エージェントは、ログファイルの新しいレコードを *Update interval* 秒ごとに 1 回処理します。
- Zabbix エージェントは、1 秒あたり `maxlines` を超えるログファイルを送信しません。この制限はネットワークおよび CPU リソースの過負荷を防ぎ、agent configuration file の `MaxLinesPerSecond` パラメータで指定されるデフォルト値を上書き

します。
- 必要な文字列を見つけるために、Zabbix は MaxLinesPerSecond で設定された値の 10 倍の新しい行を処理します。したがって、たとえば log[] または logrt[] アイテムの *Update interval* が 1 秒の場合、デフォルトではエージェントは 200 件を超えるログファイルレコードを解析せず、1 回のチェックで Zabbix サーバーに送信する一致レコードは 20 件を超えません。エージェント設定ファイルで MaxLinesPerSecond を増やすか、アイテムキーで maxlines パラメータを設定すると、1 回のチェックで解析されるログファイルレコードは最大 10000 件、Zabbix サーバーに送信される一致レコードは最大 1000 件まで増やせます。*Update interval* を 2 秒に設定すると、1 回のチェックに対する制限は *Update interval* が 1 秒の場合の 2 倍になります。
- さらに、log および log.count の値は、非ログ値が含まれていなくても、常にエージェント送信バッファサイズの 50% に制限されます。そのため、maxlines の値を 1 回の接続で送信するには（複数回の接続に分かれないようにするには）、エージェントの BufferSize パラメータは少なくとも maxlines x 2 である必要があります。Zabbix エージェントはログ収集中にデータをアップロードしてバッファを解放できますが、Zabbix エージェント 2 はデータがアップロードされてバッファが解放されるまでログ収集を停止します。これは非同期で実行されます。
- ログアイテムが存在しない場合、エージェントバッファサイズ全体が非ログ値に使用されます。ログ値が入ってくると、必要に応じて古い非ログ値が置き換えられ、指定された 50% まで使用されます。
- 256kB を超えるログファイルレコードでは、正規表現との照合は先頭 256kB のみが対象となり、残りのレコードは無視されます。ただし、Zabbix エージェントが長いレコードを処理している最中に停止すると、エージェント内部状態は失われ、再起動後にその長いレコードが再度、かつ異なる方法で解析される可能性があります。
- \ のパス区切り文字に関する特記事項: file_format が file\.log の場合、. がエスケープされているのか、ファイル名の先頭文字なのかを曖昧なく定義できないため、file ディレクトリは存在してはなりません。
- logrt の正規表現はファイル名に対してのみサポートされ、ディレクトリの正規表現マッチングはサポートされません。
- UNIX プラットフォームでは、ログファイルが存在すると想定されるディレクトリが存在しない場合、logrt[] アイテムは NOTSUPPORTED になります。
- Microsoft Windows では、ディレクトリが存在しない場合でもアイテムは NOTSUPPORTED になりません（たとえば、アイテムキー内のディレクトリ名が誤っている場合）。
- logrt[] アイテムに対してログファイルが存在しないことは、NOTSUPPORTED にはなりません。logrt[] アイテムのログファイル読み取りエラーは Zabbix エージェントのログファイルに警告として記録されますが、アイテムを NOTSUPPORTED にはしません。
- Zabbix エージェントのログファイルは、log[] または logrt[] アイテムが NOTSUPPORTED になった理由を調べるのに役立ちます。Zabbix は、DebugLevel=4 または DebugLevel=5 の場合を除き、エージェントのログファイルを監視できます。
- 正規表現で疑問符を検索する場合、たとえば \? は、テキストファイルに NUL 文字が含まれていると誤検出を引き起こす可能性があります。これは、Zabbix が改行文字まで行の処理を続けるために、それらを ? に置き換えるためです。

正規表現の一致部分の抽出

正規表現に一致した場合に行全体を返すのではなく、対象ファイルから必要な値だけを抽出したい場合があります。

ログアイテムには、一致した行から必要な値を抽出する機能があります。これは、log および logrt アイテムの追加の **output** パラメータによって実現されます。

「output」パラメータを使用すると、一致結果のうち、必要な「キャプチャグループ」を指定できます。

たとえば、次のように指定します。

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

これにより、次の内容に含まれるエントリ数が返されます。

```
Fr Feb 07 2014 11:07:36.6690 */ Thread Id 1400 (GLEWF) large result
buffer allocation - /Length: 437136/Entries: 5948/Client Ver: >=10/RPC
ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement
```

\1 は最初で唯一のキャプチャグループ（[0-9]+）を参照するため、数値だけが返されます。

また、数値を抽出して返す機能により、その値を使用してトリガーを定義できます。

maxdelay パラメータの使用

ログアイテムの maxdelay パラメータを使用すると、ログファイル内の一部の古い行を無視して、maxdelay 秒以内に解析される最新の行を取得できます。

> **Warning:**
> maxdelay > 0 を指定すると、重要なログファイルの記録を無視したり、障害通知を見逃したりする可能性があります。必要な場合にのみ、自己責任で慎重に使用してください。

デフォルトでは、ログ監視用のアイテムはログファイルに現れるすべての新しい行を追跡します。ただし、状況によっては、ログファイルに大量のメッセージを書き込み始めるアプリケーションがあります。たとえば、データベースや DNS サーバーが利用できない場合、そのようなアプリケーションは通常の動作が復旧するまで、ほぼ同一のエラーメッセージを何千件もログファイルに出力しま

す。デフォルトでは、これらのメッセージはすべて忠実に解析され、log および logrt アイテムで設定されたとおりに一致した行がサーバーへ送信されます。

過負荷に対する組み込みの保護として、設定可能な maxlines パラメータ（サーバーを大量の一致ログ行の受信から保護します）と、10*maxlines の上限（1 回のチェックでエージェントがホストの CPU と I/O を過負荷にしないよう保護します）があります。それでも、組み込み保護には 2 つの問題があります。1 つ目は、情報量の少ない可能性のあるメッセージが大量にサーバーへ報告され、データベースの容量を消費することです。2 つ目は、1 秒あたりに解析できる行数が限られているため、エージェントが最新のログ記録から数時間遅れる可能性があることです。おそらく、古い記録を何時間も追いかけるよりも、ログファイルの現在の状況をより早く把握したいはずです。

この 2 つの問題を解決するには、maxdelay パラメータを使用します。maxdelay > 0 を指定すると、各チェックで処理済みバイト数、残りバイト数、および処理時間が測定されます。これらの数値から、エージェントは推定遅延、つまりログファイル内の残りのすべての記録を解析するのに何秒かかるかを計算します。

遅延が maxdelay を超えない場合、エージェントは通常どおりログファイルの解析を続けます。

遅延が maxdelay より大きい場合、エージェントはログファイルの一部を「飛ばして」無視し、残りの行を maxdelay 秒以内に解析できるように、新しい推定位置へ移動します。

なお、エージェントは無視した行をバッファーに読み込むことすらせず、ファイル内で移動するおおよその位置を計算します。

ログファイルの行をスキップした事実は、エージェントのログファイルに次のように記録されます。

```
14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,,120.0]"
logfile:"/home/zabbix32/test1.log" skipping 679858 bytes
(from byte 75653115 to byte 76332973) to meet maxdelay
```

to byte の数値は概算です。なぜなら、「ジャンプ」後にエージェントはファイル内の位置をログ行の先頭に調整しますが、その位置はファイル内でさらに後ろ、または前になる可能性があるためです。

ログの増加速度と解析速度の比較によっては、「ジャンプ」がまったく発生しない場合もあれば、まれに発生する場合、頻繁に発生する場合、大きい「ジャンプ」や小さい「ジャンプ」が起こる場合、あるいは各チェックで小さな「ジャンプ」が発生する場合もあります。システム負荷やネットワーク遅延の変動も遅延の計算に影響し、その結果、maxdelay パラメータに追従するために先へ「ジャンプ」する動作にも影響します。

maxdelay < update interval の設定は推奨されません（頻繁に小さな「ジャンプ」が発生する可能性があります）。

'copytruncate' ログファイルローテーションの取り扱いに関する注意事項

logrt の copytruncate オプションは、異なるログファイルには異なるレコードが含まれている（少なくともタイムスタンプが異なる）ことを前提としているため、先頭ブロック（最初の 512 バイトまで）の MD5 合計値は異なります。先頭ブロックの MD5 合計値が同じ 2 つのファイルがある場合、そのうちの 1 つが元のファイルで、もう 1 つはコピーです。

copytruncate オプション付きの logrt は、重複を報告せずにログファイルのコピーを正しく処理するよう努めます。ただし、同じタイムスタンプを持つ複数のログファイルコピーを作成すること、logrt[] アイテムの更新間隔よりも頻繁にログファイルをローテーションすること、エージェントを頻繁に再起動することは推奨されません。エージェントはこれらの状況すべてに対して妥当に対処しようとしますが、すべての状況で良好な結果が保証されるわけではありません。

ログ *[] アイテムの永続ファイルに関する注意事項

永続ファイルの目的

Zabbix エージェントが起動すると、Zabbix サーバーまたはプロキシからアクティブチェックの一覧を受け取ります。
log*[] メトリクスについては、ログ監視をどこから開始するかを判断するために、処理済みのログサイズと更新時刻を受け取ります。
ファイルシステムから報告される実際のログファイルサイズと更新時刻に応じて、エージェントは処理済みのログサイズからログ監視を継続するか、ログファイルを先頭から再解析するかを判断します。

稼働中のエージェントは、チェック間で監視対象のすべてのログファイルを追跡するために、より多くの属性を保持します。
このメモリ上の状態は、エージェントが停止すると失われます。

新しいオプションパラメーター persistent_dir は、この log[]、log.count[]、logrt[] または logrt.count[] アイテムの状態をファイルに保存するためのディレクトリを指定します。
Zabbix エージェントの再起動後、ログアイテムの状態は永続ファイルから復元されます。

主な用途は、ミラーリングされたファイルシステム上にあるログファイルの監視です。
ある時点までは、ログファイルは両方のミラーに書き込まれます。
その後、ミラーは分離されます。
アクティブ側のコピーではログファイルは引き続き増加し、新しいレコードが追加されます。
Zabbix エージェントはそれを解析し、処理済みのログサイズと更新時刻をサーバーに送信します。
パッシブ側のコピーではログファイルは変化せず、アクティブ側のコピーよりかなり遅れた状態のままです。
その後、オペレーティングシステムと Zabbix エージェントはパッシブ側のコピーから再起動されます。

Zabbix エージェントがサーバーから受け取る処理済みのログサイズと更新時刻は、パッシブ側のコピーの状況には適していない場合があります。
ファイルシステムのミラー分離時点でエージェントが中断した場所からログファイル監視を継続するために、エージェントは永続ファイルから状態を復元します。

永続ファイルを使用したエージェントの動作

起動時、Zabbix エージェントは永続ファイルについて何も知りません。
Zabbix サーバー（プロキシ）からアクティブチェックの一覧を受信して初めて、いくつかのログアイテムが指定されたディレクトリ内の永続ファイルによって保持されるべきであることを認識します。

エージェントの動作中、永続ファイルは書き込み用に開かれ（fopen(filename, "w")を使用）、最新データで上書きされます。
上書きとファイルシステムのミラー分割が同時に発生した場合に永続ファイルのデータを失う可能性は非常に低く、特別な対処は不要です。
永続ファイルへの書き込み後に、ストレージ媒体への強制的な同期（fsync()の呼び出し）は行われません。

最新データへの上書きは、Zabbix サーバーへの一致するログファイルレコード、またはメタデータ（処理済みログサイズおよび更新時刻）の報告が成功した後に行われます。
ログファイルが変化し続ける場合、これは各アイテムチェックごとに発生することがあります。

エージェントのシャットダウン時に特別な処理はありません。

アクティブチェックの一覧を受信すると、エージェントは削除対象の古い永続ファイルに印を付けます。
永続ファイルは、次の場合に古くなります。

1. 対応するログアイテムが監視されなくなった場合。
2. ログアイテムが再設定され、persistent_dir の場所が以前と異なる場合。

削除は 24 時間の遅延後に行われます。これは、NOTSUPPORTED 状態のログファイルはアクティブチェックの一覧に含まれないものの、後で SUPPORTED になる可能性があり、その場合は永続ファイルが役立つためです。

24 時間が経過する前にエージェントが停止した場合、Zabbix エージェントは Zabbix サーバーからその場所に関する情報を受け取れなくなるため、古いファイルは削除されません。

Warning:
エージェントが停止している間に、ユーザーが古い永続ファイルを削除せずにログアイテムの persistent_dir を元の persistent_dir の場所へ戻して再設定すると、古い永続ファイルからエージェントの状態が復元され、その結果、メッセージの取りこぼしや誤検知の障害が発生します。

永続ファイルの名前と場所

Zabbix エージェントは、キーによってアクティブチェックを区別します。たとえば、logrt[/home/zabbix/test.log] と logrt[/home/zabbix/test.log,] は異なるアイテムです。Web インターフェースでアイテム logrt[/home/zabbix/test.log,,,10] を logrt[/home/zabbix/test.log,,,20] に変更すると、エージェントのアクティブチェック一覧からアイテム logrt[/home/zabbix/test.log,,,10] が削除され、アイテム logrt[/home/zabbix/test.log,,,20] が作成されます（Web インターフェース/サーバーでの変更時には一部の属性が引き継がれますが、エージェントでは引き継がれません）。

ファイル名は、衝突の可能性を低減するため、アイテムキーの MD5 ハッシュにアイテムキーの長さを付加して構成されます。たとえば、アイテム logrt[/home/zabbix50/test.log,,,,,,,,/home/zabbix50/agent_private] の状態は、永続ファイル c963ade4008054813bbc0a650bb8e09266 に保持されます。

複数のログアイテムで同じ persistent_dir を使用できます。

persistent_dir は、特定のファイルシステムレイアウト、マウントポイントとマウントオプション、およびストレージのミラーリング設定を考慮して指定します。永続ファイルは、監視対象のログファイルと同じミラーリングされたファイルシステム上に配置する必要があります。

persistent_dir ディレクトリを作成できない、ディレクトリが存在しない、または Zabbix エージェントにファイルの作成、書き込み、読み取り、削除に必要な権限がない場合、ログアイテムは NOTSUPPORTED になります。

エージェントの稼働中に永続ストレージファイルへのアクセス権が削除された場合や、その他のエラー（ディスク容量不足など）が発生した場合、エラーはエージェントのログファイルに記録されますが、ログアイテムは NOTSUPPORTED にはなりません。

I/O の負荷

アイテムの永続ファイルは、アイテムのデータを含む各データバッチをサーバーへ正常に送信した後に更新されます。
たとえば、デフォルトの BufferSize は 100 です。
ログアイテムで 70 件の一致レコードが見つかった場合、最初の 50 件のレコードは 1 つ目のバッチで送信され、永続ファイルが更新されます。その後、残りの 20 件のレコードは 2 つ目のバッチで送信され（より多くのデータが蓄積されるまで少し遅延する場合があります）、永続ファイルが再度更新されます。

エージェントとサーバー間の通信が失敗した場合の動作

log[] および logrt[] アイテムの各一致行と、各 log.count[] および logrt.count[] アイテムのチェック結果には、エージェント送信バッファ内の指定された 50% 領域に空きスロットが必要です。
バッファ内の要素は定期的にサーバー（またはプロキシ）へ送信され、その後バッファスロットは再び空きになります。

エージェント送信バッファの指定されたログ領域に空きスロットがある間にエージェントとサーバー（またはプロキシ）間の通信が失敗すると、ログ監視結果は送信バッファに蓄積されます。
これにより、短時間の通信障害を緩和できます。

通信障害が長引くと、すべてのログスロットが使用中になり、次の動作が行われます。

- log[] および logrt[] アイテムのチェックは停止されます。
  通信が回復し、バッファに空きスロットがある場合、チェックは前回の位置から再開されます。
  一致した行は失われず、後で報告されます。
- log.count[] および logrt.count[] のチェックは、maxdelay = 0（デフォルト）の場合に停止されます。
  動作は上記の log[] および logrt[] アイテムと同様です。
  これは log.count[] および logrt.count[] の結果に影響する可能性があることに注意してください。たとえば、あるチェックでログファイル内の一致行を 100 行数えたものの、バッファに空きスロットがないためチェックが停止されたとします。
  通信が回復すると、エージェントは同じ 100 行の一致行を数え、さらに 70 行の新しい一致行も数えます。
  エージェントは、あたかも 1 回のチェックで見つかったかのように count = 170 を送信します。
- log.count[] および logrt.count[] の maxdelay > 0 のチェックでは、チェック中に”jump” が発生しなかった場合、動作は上記と同様です。
  ただし、ログファイル行をまたぐ”jump” が発生した場合は、”jump” 後の位置が保持され、数えられた結果は破棄されます。
  つまり、通信障害が発生した場合でも、エージェントは増加するログファイルに追従しようとします。

正規表現のコンパイルエラーおよび実行時エラーの処理

log[]、logrt[]、log.count[]、または logrt.count[] アイテムで使用されている正規表現を PCRE または PCRE2 ライブラリでコンパイルできない場合、アイテムはエラーメッセージとともに「未サポート」状態になります。ログアイテムの監視を継続するには、正規表現を修正する必要があります。

正規表現のコンパイルが正常に完了したものの、実行時に（一部またはすべてのログレコードで）失敗する場合、ログアイテムはサポート状態のままとなり、監視は継続されます。実行時エラーは、Zabbix エージェントのログファイルに記録されます（ログファイルのレコードは記録されません）。

Zabbix エージェントが自身のログファイルを監視できるように、ログの記録頻度は 1 回のチェックにつき 1 件の実行時エラーに制限されます。たとえば、10 件のレコードを解析し、そのうち 3 件で正規表現の実行時エラーが発生した場合、エージェントのログには 1 件のレコードが出力されます。

例外として、MaxLinesPerSecond=1 かつ update interval=1 の場合（1 回のチェックで解析できるレコードが 1 件のみの場合）、正規表現の実行時エラーは記録されません。

zabbix_agentd は実行時エラーが発生した場合にアイテムキーをログに記録し、zabbix_agent2 はどのログアイテムで実行時エラーが発生したかを特定できるようにアイテム ID を記録します。実行時エラーが発生した場合は、正規表現を再設計することを推奨します。

シンプルチェック

概要

シンプルチェックは、サービスのリモートエージェントレスチェックに通常使用されます。

シンプルチェックには Zabbix エージェントは必要ありません。Zabbix サーバー/プロキシがシンプルチェックの処理 (外部接続など) を担当します。

シンプルチェックの使用例:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

> **Note:**
> シンプルチェックアイテムの設定にあるユーザー名とパスワードのフィールド (255 文字まで) は、VMware 監視アイテムで使用されます。それ以外の場合は無視されます。

サポートされるチェック

アイテムキーは、オプションのパラメーターと追加情報を省略して記載しています。アイテムキーをクリックすると、詳細を表示できます。

VMware 監視のアイテムキーも参照してください。

Item key	Description
icmpping	ICMP ping によるホストの到達性。
icmppingloss	失われたパケットの割合。
icmppingretry	再試行付きの ICMP ping によるホストの到達性。
icmppingsec	ICMP ping の応答時間。
net.tcp.service	サービスが稼働しており、TCP 接続を受け付けているかを確認します。
net.tcp.service.perf	TCP サービスのパフォーマンスを確認します。
net.udp.service	サービスが稼働しており、UDP 要求に応答しているかを確認します。
net.udp.service.perf	UDP サービスのパフォーマンスを確認します。

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 ICMP ping によるホストの到達性。
 戻り値: *0* - ICMP ping 失敗; *1* - ICMP ping 成功。

パラメーター:

- **target** - ホストの IP または DNS 名;
- **packets** - パケット数;
- **interval** - 連続するパケット間の時間（ミリ秒）;
- **size** - パケットサイズ（バイト）;
- **timeout** - タイムアウト（ミリ秒）;
- **options** - リダイレクトを許可するために使用します。空の場合（デフォルト値）、リダイレクトされた応答は対象ホストがダウンしているものとして扱われます。*allow_redirect* に設定されている場合、リダイレクトされた応答は対象ホストが稼働しているものとして扱われます。

デフォルト値 の表も参照してください。

コメント:

- packets パラメーターで定義されたパケットのうち少なくとも 1 つが返された場合、アイテムは 1 を返します。すべてのパケットが返らなかった場合、アイテムは 0 を返します。

例:

```
icmpping[,4] #4 つのパケットのうち少なくとも 1 つが返された場合、アイテムは 1 を返します。
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 失われたパケットの割合。
 戻り値: *Float*。

パラメータ:

- **target** - ホストの IP または DNS 名
- **packets** - パケット数
- **interval** - 連続するパケット間の時間 (ミリ秒)
- **size** - パケットサイズ (バイト)
- **timeout** - タイムアウト (ミリ秒)
- **options** - リダイレクトを許可する場合に使用: 空の場合 (デフォルト値)、リダイレクトされた応答はターゲットホストがダウンしているとみなされる; *allow_redirect* を設定した場合、リダイレクトされた応答はターゲットホストが稼働しているとみなされる

デフォルト値の表も参照してください。

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

 ICMP ping によるホストの到達性をリトライ付きでチェックします。最初のパケットが成功した場合は停止し、パケットが失敗した場合は retries パラメータで定義された最大試行回数までリトライします。このアイテムは、ネットワーク上で送信されるパケット数を減らすのに役立ちます。
 戻り値: *0* - ICMP ping 失敗; *1* - ICMP ping 成功。

パラメータ:

- **target** - ホストの IP または DNS 名
- **retries** - ターゲットへの ping を試行する回数 (最初の試行は含まない、0 以上; デフォルト 1)
- **backoff** - 各リクエストごとに待機時間を乗算する数値 (1.0-5.0 の範囲; デフォルト 1.0)
- **size** - パケットサイズ (バイト単位)

- **timeout** - タイムアウト (ミリ秒)
- **options** - リダイレクトを許可する場合に使用: 空 (デフォルト値) の場合、リダイレクトされた応答はターゲットホストがダウンとみなされます。*allow_redirect* を設定すると、リダイレクトされた応答はターゲットホストがアップとみなされます。

デフォルト値の表も参照してください。

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 ICMP ping の応答時間（秒単位）。
 戻り値：*Float*。

パラメータ:

- **target** - ホストの IP または DNS 名
- **packets** - パケット数
- **interval** - 連続するパケット間の時間（ミリ秒単位）
- **size** - パケットサイズ（バイト単位）
- **timeout** - タイムアウト（ミリ秒単位）
- **mode** - 可能な値：*min*、*max*、または *avg*（デフォルト）
- **options** - リダイレクトを許可する場合に使用：空（デフォルト値）の場合、リダイレクトされた応答はターゲットホストがダウンしていると見なされる。*allow_redirect* を設定した場合、リダイレクトされた応答はターゲットホストがアップしていると見なされる。

コメント:

- ロストまたはタイムアウトしたパケットは計算に使用されません
- ホストが利用できない場合（タイムアウトに達した場合）、アイテムは 0 を返します
- 戻り値が 0.0001 秒未満の場合、値は 0.0001 秒に設定されます
- デフォルト値の表も参照してください

net.tcp.service[service,<ip>,<port>]

 サービスが稼働しており、TCP 接続を受け付けているかどうかをチェックします。
 戻り値: *0* - サービスは停止しています; *1* - サービスは稼働しています。

パラメータ:

- **service** - 使用可能な値: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

コメント:

- *tcp* サービスの場合、ポートの指定は必須です;
- これらのチェックは、システムデーモンのログファイルに追加のメッセージを生成する場合があります (通常、SMTP や SSH セッションが記録されます);
- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには `net.tcp.service[tcp,<ip>,port]` を使用してください。

例:

```
net.tcp.service[ftp,,45] #このアイテムは、TCPポート45でFTPサーバーの可用性をテストするために使用できます。
```

> **Attention:**
> SELinux が強制モードで実行されている場合、カスタム TCP/UDP シンプルチェックはポリシーによってブロックされることがあります。新しい送信接続を確認して許可するには、`grep denied /var/log/audit/audit.log` で監査拒否を確認してください。

net.tcp.service.perf[service,<ip>,<port>]

 TCP サービスのパフォーマンスをチェックします。
 戻り値: *Float*: *0.000000* - サービスがダウンしている; 秒 - サービスへの接続にかかった秒数。

パラメータ:

- **service** - 使用可能な値: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

コメント:

- *tcp* サービスの場合、ポートの指定は必須です;
- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには `net.tcp.service[tcp,<ip>,port]` を使用してください。

例:

```
net.tcp.service.perf[ssh] #このアイテムはSSHサーバーからの初期応答速度のテストに使用できます。
```

net.udp.service[service,<ip>,<port>]

 サービスが稼働しており、UDP リクエストに応答しているかどうかをチェックします。
 戻り値: *0* - サービスはダウンしています; *1* - サービスは稼働しています。

パラメータ:

- **service** - 可能な値: *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

例:

```
net.udp.service[ntp,,45] #このアイテムはUDPポート45でNTPサービスの可用性をテストするために使用できます。
```

net.udp.service.perf[service,<ip>,<port>]

 UDP サービスのパフォーマンスをチェックします。
 戻り値: *Float*: *0.000000* - サービスがダウンしている; 秒 - サービスからの応答を待った秒数。

パラメータ:

- **service** - 可能な値: *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準サービスのポート番号が使用されます)。

例:

```
net.udp.service.perf[ntp] #このアイテムはNTPサービスからの応答時間をテストするために使用できます。
```

Attention:
LDAP の simple check（例: `net.tcp.service[ldap]`）で SourceIP をサポートするには、OpenLDAP バージョン 2.6.1 以降が必要です。

タイムアウト処理

柔軟なアイテムタイムアウトは、シンプルチェックでもサポートされていますが、`icmpping*` および VMware アイテムには適用されません。柔軟なタイムアウトサポートを参照してください。

ICMP ping

Zabbix は、ICMP ping（**icmpping**、**icmppingloss**、**icmppingretry**、**icmppingsec**）を処理するために外部ユーティリティ**fping**を使用します。

インストール

fping は Zabbix には含まれておらず、別途インストールする必要があります。

- さまざまな Unix 系プラットフォームでは、fping パッケージがデフォルトのリポジトリにありますが、事前にインストールされていません。この場合、パッケージマネージャーを使用して fping をインストールできます。
- Zabbix は RHEL およびその派生ディストリビューション向けにfping パッケージを提供しています。これらのパッケージは公式サポートなしで提供されていることにご注意ください。
- fping はソースからコンパイルすることもできます。

設定

Zabbix サーバー/プロキシの設定ファイルの *FpingLocation* パラメーターに fping の場所を指定します
(IPv6 アドレスを使用する場合は *Fping6Location* パラメーターを使用します)。

fping は Zabbix サーバー/プロキシが実行されるユーザーによって実行可能である必要があり、このユーザーには十分な権限が必要です。

関連情報: 3.10 より前の fping バージョンでシンプルチェックを処理する場合の既知の問題 を参照してください。

デフォルト値

ICMP チェックパラメータの値に関するデフォルト値、制限、および説明:

パラメータ	単位	説明	Fping のフラグ	デフォルト値の設定元		Zabbix による許容範囲	
				fping	**Zabbix**	最小	最大
packets	数	ターゲットに送信するリクエストパケット数	-C		3	1	10000
interval	ミリ秒	個々のターゲットに連続してパケットを送信する間隔	-p	1000		20	無制限
size	バイト	パケットサイズ（バイト単位） x86 では 56 バイト、x86_64 では 68 バイト	-b	56 または 68		24	65507
timeout	ミリ秒	**fping v3.x** - 最後のパケット送信後に待機するタイムアウト。-C フラグの影響を受けます **fping v4.x** - 各パケットの個別のタイムアウト	-t	**fping v3.x** - 500 **fping v4.x** 以降 - -p フラグから継承されますが、2000 を超えません		50	無制限
retries	数	最初の試行を除き、ターゲットへの ping を試行する回数	-r	3	1	0	無制限
backoff factor	数	後続の各リクエストで待機時間を乗算する値	-B	1.5	1.0	1.0	5.0

デフォルト値は、プラットフォームとバージョンによって若干異なる場合があります。

さらに、Zabbix は fping オプション `-i interval ms`（上記の表に記載されているアイテムパラメータ `interval` と混同しないでください。このパラメータは fping オプション `-p` に対応します）および `-S source IP address`（古い fping バージョンでは `-I`）を使用します。これらのオプションは、異なるオプションの組み合わせでチェックを実行して自動検出されます。Zabbix は、0、1、10 の 3 つの値を試行して、fping が `-i` オプションに対して許可するミリ秒単位の最小値を検出します。最初に成功した値が、その後の ICMP チェックで使用されます。この処理は、各ICMP pingerプロセスによって個別に実行されます。

自動検出された fping オプションは 1 時間ごとに無効化され、次回 ICMP チェックを実行する際に再検出されます。この処理の詳細をサーバーまたはプロキシのログファイルで確認するには、DebugLevel>=4 を設定します。

Zabbix は、いずれかの `icmpping*` キーでチェックする IP アドレスを一時ファイルに書き込み、そのファイルを fping に渡します。アイテムのキーのパラメータが異なる場合、キーのパラメータが同一のものだけが 1 つのファイルに書き込まれます。1 つのファイルに書き込まれたすべての IP アドレスは fping によって並列にチェックされるため、ファイル内の IP アドレス数に関係なく、Zabbix の ICMP pinger プロセスが費やす時間は一定です。

VMware 監視アイテムキー

VMware 監視のitem キーのリストは、VMware 監視セクションに移動されました。

SNMP エージェント

概要

プリンター、ネットワークスイッチ、ルーター、UPS など、通常は SNMP に対応しており、完全なオペレーティングシステムと Zabbix エージェントを導入するのが現実的ではないデバイスでは、SNMP 監視を使用したい場合があります。

これらのデバイス上の SNMP エージェントが提供するデータを取得できるようにするには、Zabbix サーバーを `--with-net-snmp` フラグを指定して SNMP サポート付きで初期設定 する必要があります。アイテム値が正しい形式で表示されるように、MIB ファイルのインストール も推奨されます。MIB ファイルがない場合、UTF-8 の代わりに HEX で値が表示される、またはその逆といった書式の問題が発生することがあります。

SNMP チェックは UDP プロトコルでのみ実行されます。

Zabbix サーバーおよびプロキシのデーモンは、誤った SNMP 応答を受信した場合、次のような行をログに記録します。

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

これらは問題のあるすべてのケースを網羅するものではありませんが、複合リクエストを無効にすべき個別の SNMP デバイスを特定するのに役立ちます。

Zabbix サーバー/プロキシは、SNMP walk および get アイテムに対して最大 5 回再試行します。この再試行メカニズムは DNS 解決の失敗には適用されません。

従来の SNMP チェック（単一の OID 番号または文字列）では、Zabbix サーバー/プロキシは、クエリの試行が失敗した後に少なくとも 1 回再試行します。これは、SNMP ライブラリの再試行メカニズム、または内部の複合処理 メカニズムのいずれかによって行われます。

Warning:
SNMPv3 デバイスを監視する場合、msgAuthoritativeEngineID（snmpEngineID または”Engine ID” とも呼ばれます）が 2 台のデバイス間で共有されないようにしてください。RFC 2571（3.1.1.1 節）によると、これは各デバイスで一意でなければなりません。

Warning:
RFC3414 では、SNMPv3 デバイスが engineBoots を永続化することを要求しています。一部のデバイスはこれを行わず、その結果、再起動後に SNMP メッセージが古いものとして破棄されます。このような場合、SNMP キャッシュをサーバー/プロキシ上で手動でクリアする（-R snmp_cache_reload を使用する）か、サーバー/プロキシを再起動する必要があります。

Note:
Zabbix は SNMPv3 の EngineID → IP マッピングをキャッシュし、毎回プローブを送信する代わりに、後続のチェックでキャッシュされた EngineID を再利用してネットワークトラフィックを削減します。EngineID を再利用できない場合は、新しい EngineID を検出するためにプローブ付きの再試行が実行されます。

SNMP 監視の設定

SNMP を介してデバイスの監視を開始するには、次の手順を実行する必要があります。

ステップ 1

監視したいアイテムの SNMP 文字列（または OID）を確認します。

SNMP 文字列の一覧を取得するには、snmpwalk コマンド（Zabbix のインストール時に一緒にインストールしているはずの net-snmp ソフトウェアの一部）または同等のツールを使用します。

```
snmpwalk -v 2c -c public <host IP> .
```

ここで 2c は SNMP バージョンを表すため、デバイス上の SNMP Version 1 を示す 1 に置き換えることもできます。

これにより、SNMP 文字列の一覧とその最後の値が表示されます。表示されない場合は、SNMP の「community」が標準の public とは異なる可能性があるため、その値を確認する必要があります。

その後、監視したい文字列が見つかるまで一覧を確認します。たとえば、ポート 3 のスイッチに流入するバイト数を監視したい場合は、次の行の IF-MIB::ifHCInOctets.3 という文字列を使用します。

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

次に、snmpget コマンドを使用して IF-MIB::ifHCInOctets.3 の数値 OID を確認できます。

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

文字列の最後の数字は、監視対象のポート番号であることに注意してください。あわせて参照: Dynamic indexes。

すると、次のような結果が得られます。

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

ここでも、OID の最後の数字はポート番号です。

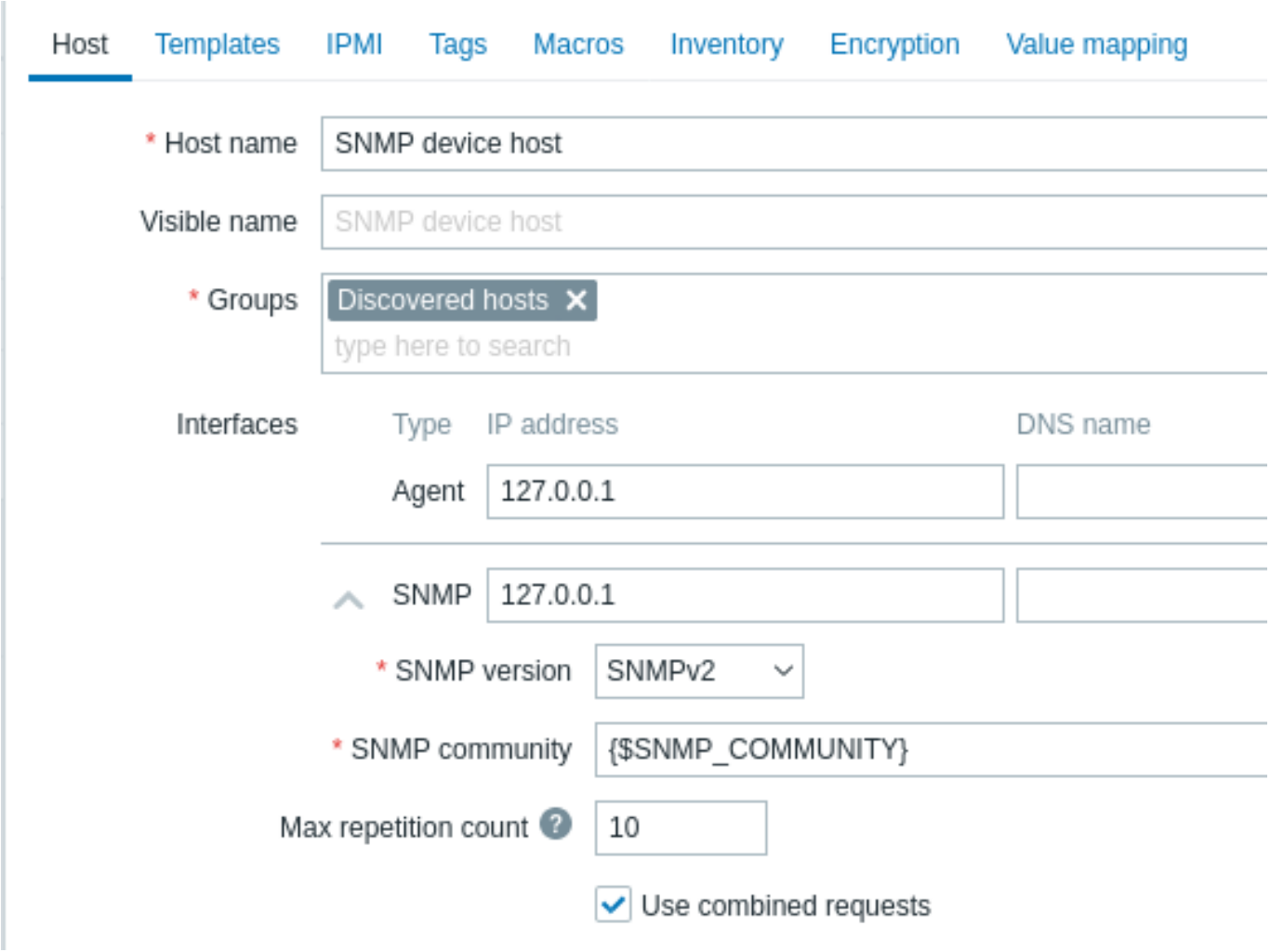
Note:
よく使用される SNMP OID の一部は、Zabbix により数値表現へ自動的に変換されます。

上の最後の例では値の型は”Counter64” で、内部的には ASN_COUNTER64 型に対応します。対応している型の完全な一覧は、ASN_COUNTER、ASN_COUNTER64、ASN_UINTEGER、ASN_UNSIGNED64、ASN_INTEGER、ASN_INTEGER64、ASN_FLOAT、ASN_DOUBLE、ASN_TIMETICKS、ASN_GAUGE、ASN_IPADDRESS、ASN_OCTET_STR、ASN_OBJECT_ID です。これらの型は、**snmpget** の出力にお

ける”Counter32”、”Counter64”、”UInteger32”、”INTEGER”、”Float”、”Double”、”Timeticks”、”Gauge32”、”IpAddress”、”OCTET STRING”、”OBJECT IDENTIFIER” におおむね対応しますが、表示ヒントの有無によっては”STRING”、”Hex-STRING”、”OID” などとして表示されることもあります。

ステップ 2

デバイスに対応するホストを作成します。



ホストに SNMP インターフェースを追加します。

1. IP アドレス/DNS 名とポート番号を入力します。
2. ドロップダウンから *SNMP version* を選択します。
3. 選択した SNMP バージョンに応じてインターフェース認証情報を追加します。
    - SNMPv1、v2 ではコミュニティー名のみが必要です（通常は’public’）。
    - SNMPv3 ではより詳細なオプションが必要です。SNMPv3 を参照してください。
4. ネイティブ SNMP バルクリクエスト（GetBulkRequest-PDU）に対する最大繰り返し値（デフォルト: 10）を指定します。これは SNMPv2 および v3 の `discovery[]` と `walk[]` アイテムにのみ適用されます。この値を高く設定しすぎると、SNMP エージェントチェックのタイムアウトが発生する場合があります。
5. *Use combined requests* チェックボックスをオンにすると、SNMP リクエストのcombined processing を有効にできます（ネイティブ SNMP バルクリクエストの”walk” および”get” とは関係ありません）。

提供されている SNMP テンプレートのいずれかを使用すると、アイテムのセットが自動的に追加されます。テンプレートを使用する前に、そのテンプレートがホストと互換性があることを確認してください。

Add をクリックしてホストを保存します。

SNMPv3

SNMPv3 では、以下のパラメータが必要です。

- *Context name* - SNMP サブネット上でアイテムを識別するためのコンテキスト名を入力します。
 このフィールドではユーザーマクロが解決されます。
- *Security name* - セキュリティ名を入力します。
 このフィールドではユーザーマクロが解決されます。
- *Security level* - セキュリティレベルを選択します。
    - *noAuthNoPriv* - 認証プロトコルもプライバシープロトコルも使用しません。

    - *AuthNoPriv* - 認証プロトコルを使用し、プライバシープロトコルは使用しません。
    - *AuthPriv* - 認証プロトコルとプライバシープロトコルの両方を使用します。
- *Authentication protocol* - 認証プロトコルを選択します: *MD5*、*SHA1*。net-snmp 5.8 以降では *SHA224*、*SHA256*、*SHA384*、または *SHA512* も使用できます。
- *Authentication passphrase* - 認証パスフレーズを入力します。
 このフィールドではユーザーマクロが解決されます。
- *Privacy protocol* - プライバシープロトコルを選択します: *DES*、*AES128*、*AES192*、*AES256*、*AES192C* (Cisco)、または *AES256C* (Cisco)。
プライバシープロトコルのサポート に関する注意事項を参照してください。
- *Privacy passphrase* - プライバシーパスフレーズを入力します。
 このフィールドではユーザーマクロが解決されます。

SNMPv3 の認証情報（security name、authentication protocol/passphrase、privacy protocol）が誤っている場合:

- Zabbix は net-snmp から ERROR を受け取ります。ただし、*Privacy passphrase* が誤っている場合は例外で、その場合 Zabbix は net-snmp から TIMEOUT エラーを受け取ります。
- SNMP インターフェースの利用可否は赤色（利用不可）に切り替わります。

Note:
Security name を変更せずに *Authentication protocol*、*Authentication passphrase*、*Privacy protocol*、または *Privacy passphrase* を変更した場合、通常は対応する SNMPv3 インターフェースが Zabbix で更新されると自動的に適用されます。*Security name* も変更された場合は、すべてのパラメータが直ちに更新されます。

プライバシープロトコルのサポート

お使いのオペレーティングシステムと net-snmp の設定によっては、一部のプライバシープロトコルが利用できない場合があります。

- 一部の新しいオペレーティングシステム（たとえば RHEL9）では、`net-snmp` パッケージで DES のサポートが削除されています。
- AES192 およびそれ以上の暗号化プロトコルは、RHEL 8、CentOS 8、Oracle Linux 8、Debian 12、Ubuntu LTS 22.04、openSUSE Leap 15.5 より古いオペレーティングシステムでは、標準ではサポートされていません。

net-snmp ライブラリが AES192+ をサポートしているか確認するには、次のいずれかの方法を使用してください。

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

出力に `--enable-blumenthal-aes` が含まれている場合、AES192+ はサポートされています。

なお、`net-snmp-config` は SNMP 用の開発パッケージの一部です（Debian/Ubuntu では `libsnmp-dev`、CentOS/RHEL/OL/SUSE では `net-snmp-devel`）であり、デフォルトではインストールされていない場合があります。

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

出力に `Invalid privacy protocol specified after -3x flag: AES-256` が含まれている場合、AES192+ はサポートされていません。出力に `No hostname specified.` が含まれている場合、AES192+ はサポートされています。

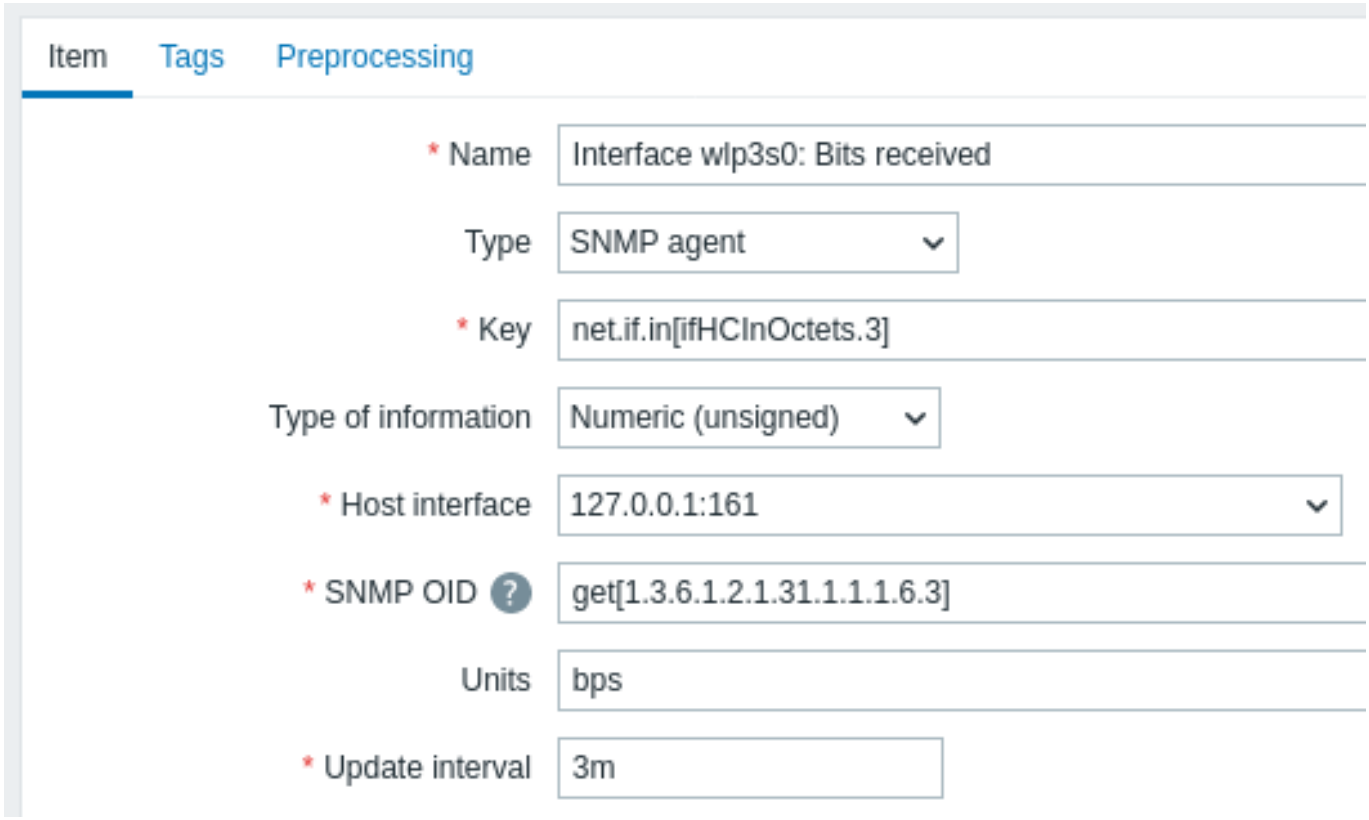
お使いの net-snmp ライブラリが AES192 以上のプロトコルをサポートしていない場合は、`--enable-blumenthal-aes` オプションを指定して net-snmp を再コンパイルし、その後 `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-config` オプションを指定して Zabbix サーバーを再コンパイルしてください。

ステップ 3

監視用のアイテムを作成します。

ここで、Zabbix に戻り、先ほど作成した SNMP ホストの アイテムをクリックします。ホストの作成時にテンプレートを使用したかどうかによって、ホストに関連付けられた SNMP アイテムの一覧、または空の一覧が表示されます。ここでは、snmpwalk と snmpget を使用して収集した情報を基に、自分でアイテムを作成するものとします。そのため、アイテムの作成をクリックします。

新しいアイテムのフォームで、必須パラメータを入力します。



パラメータ	説明
名前	アイテム名を入力します。
タイプ	ここでは **SNMP** エージェントを選択します。
キー	意味のあるキーを入力します。
ホストインターフェース	スイッチやルーターなどの SNMP インターフェースを選択していることを確認します。

パラメータ	説明
SNMP OID	OID の値を入力するには、次のいずれかのサポートされている形式を使用します。 **walk[OID1,OID2,...]** - 値のサブツリーを取得します。 例: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]` このオプションは、ネイティブ SNMP バルクリクエスト（GetBulkRequest-PDU）を非同期で使用します。 このアイテムのタイムアウト設定は、アイテム設定フォームで設定できます。デバイスに到達できない場合に長時間待機しないよう、短いタイムアウト値を設定することを検討してください。先行するリクエストがタイムアウトまたは失敗した場合、最大 5 回再試行されます（例えば、3 秒のタイムアウトでは 15 秒待機する可能性があります）。 このアイテムをマスターアイテムとして使用し、前処理によってマスターアイテムからデータを抽出する従属アイテムを設定できます。 1 回の snmp walk で複数の OID を指定できます。例えば `walk[OID1,OID2,...]` のように指定すると、各 OID が非同期で処理されます。 バルクリクエストが結果を返さない場合は、バルクリクエストを使用せずに単一レコードの取得が試行されます。 MIB 名をパラメータとして使用できます。そのため、`walk[1.3.6.1.2.1.2.2.1.2]` と `walk[ifDescr]` は同じ出力を返します。 複数の OID/MIB を指定した場合、つまり `walk[ifDescr,ifType,ifPhysAddress]` の場合、出力は連結されたリストになります。 SNMPv2 および v3 インターフェースでは GetBulk リクエストが、SNMPv1 インターフェースでは GetNext が使用されます。バルクリクエストの最大繰り返し回数は、インターフェースレベルで設定します。 最大繰り返し回数パラメータは、1 回のバルクレスポンスで返される OID の最大数を決定することで、バルクリクエストに影響します。 値を大きくするとバルクレスポンスが大きくなり、必要な送信回数が減少します。ただし、非常に大きな値をサポートしていないデバイスもあり、問題が発生する可能性があります。 このアイテムは、-Oe -Ot -On パラメータを指定した snmpwalk ユーティリティの出力を返します。 このアイテムは、SNMP ディスカバリのマスターアイテムとして使用できます。 **get[OID]** - 単一の値を非同期で取得します。 例: `get[1.3.6.1.2.1.31.1.1.1.6.3]` このアイテムのタイムアウト設定は、アイテム設定フォームで設定できます。デバイスに到達できない場合に長時間待機しないよう、短いタイムアウト値を設定することを検討してください。先行するリクエストがタイムアウトまたは失敗した場合、最大 5 回再試行されます（例えば、3 秒のタイムアウトでは 15 秒待機する可能性があります）。 **OID** - （レガシー）単一の値を同期的に取得するため、単一のテキスト形式または数値形式の OID を入力します。必要に応じて、他の値と組み合わせることもできます。 例: `1.3.6.1.2.1.31.1.1.1.6.3` このオプションでは、アイテムチェックのタイムアウトはサーバー設定ファイルで設定された値と同じになります。 パフォーマンスを向上させるため、`walk[OID]` および `get[OID]` アイテムを使用することを 推奨します。すべての `walk[OID]` および `get[OID]` アイテムは非同期で実行されるため、他のチェックを開始する前に 1 つのリクエストへの応答を受信する必要はありません。DNS 解決も非同期で行われます。 非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPoller で定義）。非同期 SNMP ポーラーの数は、StartSNMPPollers パラメータで定義します。 いずれかの方法で返されるネットワークトラフィック統計については、前処理タブで *1* 秒あたりの変化量のステップを追加する必要があります。追加しない場合、最新の変化量ではなく、SNMP デバイスからの累積値が取得されます。

すべての必須入力フィールドには、赤いアスタリスクが付いています。

ここでアイテムを保存し、SNMP データを確認するために 監視 > 最新データに移動します。

例 1

一般的な例:

パラメータ	説明
OID	1.2.3.45.6.7.8.0 (または.1.2.3.45.6.7.8.0)
キー	< トリガーの参照として使用する一意の文字列 > 例: ”my_param”。

OID は数値または文字列のいずれかの形式で指定できます。ただし、場合によっては文字列 OID を数値表現に変換する必要があります。この目的には snmpget ユーティリティを使用できます:

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

例 2

稼働時間の監視:

パラメータ	説明
OID	MIB::sysUpTime.0
キー	router.uptime
値の型	浮動小数点
単位	uptime
前処理ステップ**:** カスタム乗数	0.01

ネイティブ SNMP バルクリクエスト

walk[OID1,OID2,...] アイテムを使用すると、SNMP バージョン 2/3 で利用可能な、バルクリクエスト（GetBulkRequest-PDU）用のネイティブ SNMP 機能を使用できます。

SNMP の GetBulk リクエストは、複数の GetNext リクエストを実行し、結果を 1 つのレスポンスで返します。これにより、ネットワークのラウンドトリップを最小限に抑えながら、通常の SNMP アイテムおよび SNMP ディスカバリに使用できます。

SNMP **walk[OID1,OID2,...]** アイテムは、1 つのリクエストでデータを収集するマスターアイテムとして使用できます。依存アイテムでは、前処理を使用して必要に応じてレスポンスを解析できます。

ネイティブ SNMP バルクリクエストの使用は、SNMP リクエストを結合するオプションとは関係ありません。後者は、複数の SNMP リクエストを結合する Zabbix 独自の方法です（次のセクションを参照）。

SNMP バルクアイテムでは、パケットが 1 つ失われた場合の失敗を防ぐため、最大 5 回の再試行が行われます。`get` および `walk` を使用する SNMP アイテムのタイムアウト（アイテム設定フォームで設定）は、セッション全体に適用されます。データを完全に取得できたかどうかにかかわらず、タイムアウトが適用されます。データを部分的に受信した場合（たとえば、複数の OID のうち 1 つだけについてデータの収集に成功した場合）、アイテムは「Only partial data received」というメッセージとともにサポートされていない状態になります。タイムアウトに達すると再試行が行われ、タイムアウトがリセットされます。また、最後のリクエストが再送信されるため、1 つのパケットが失われた場合や到着が遅れた場合でも、最後のリクエストからセッションを継続できます。デバイスに到達できない場合の長時間の遅延を避けるため、低いタイムアウト値を設定することを検討してください。以前の試行がタイムアウトまたは失敗した場合、最大 5 回の再試行が行われるためです（たとえば、3 秒のタイムアウトでは待ち時間が 15 秒になる可能性があります）。

結合処理の内部動作

Zabbix サーバーとプロキシは、1 つのリクエストで SNMP デバイスに複数の値を問い合わせることができます。これは、次の種類の SNMP アイテムに影響します。

- 通常の SNMP アイテム
- 動的インデックスを使用するSNMP アイテム
- SNMP ローレベルディスカバリルール

同一インターフェース上にあり、パラメータが同一のすべての SNMP アイテムは、同時に問い合わせるようスケジュールされます。最初の 2 種類のアイテムは、最大 128 アイテムのバッチ単位でポーラーによって取得されます。一方、ローレベルディスカバリルールは、これまでどおり個別に処理されます。

より低いレベルでは、値を問い合わせるために実行される操作は 2 種類あります。指定した複数のオブジェクトを取得する操作と、OID ツリーをウォークする操作です。

「取得」では、最大 128 個の可変バインディングを含む GetRequest-PDU が使用されます。「ウォーク」では、SNMPv1 に対して GetNextRequest-PDU が使用され、SNMPv2 および SNMPv3 に対しては、`max-repetitions` フィールドを最大 128 に設定した GetBulkRequest が使用されます。

したがって、各 SNMP アイテムタイプに対する結合処理の利点は、次のようになります。

- 通常の SNMP アイテムは、「取得」の改善によるメリットを得られます。

- 動的インデックスを使用する SNMP アイテムは、「取得」と「ウォーク」の両方の改善によるメリットを得られます。「取得」はインデックスの検証に使用され、「ウォーク」はキャッシュの構築に使用されます。
- SNMP ローレベルディスカバリルールは、「ウォーク」の改善によるメリットを得られます。

ただし、すべてのデバイスが 1 回のリクエストで 128 個の値を返せるわけではないという技術的な問題があります。常に適切なレスポンスを返すデバイスもありますが、潜在的なレスポンスが一定の制限を超えると、`tooBig(1)` エラーを返すか、まったく応答しなくなるデバイスもあります。

特定のデバイスに対して問い合わせるオブジェクト数を最適化するため、Zabbix は次の戦略を使用します。まず、慎重に 1 回のリクエストで 1 つの値を問い合わせます。成功した場合は、1 回のリクエストで 2 つの値を問い合わせます。これも成功した場合は、1 回のリクエストで 3 つの値を問い合わせ、その後も同様に、問い合わせるオブジェクト数を 1.5 倍にしていきます。その結果、リクエストサイズは次の順序になります: 1、2、3、4、6、9、13、19、28、42、63、94、128。

ただし、デバイスが適切なレスポンスを返さなくなった場合（たとえば、42 個の変数に対して）、Zabbix は 2 つの処理を行います。

まず、現在のアイテムバッチでは、1 回のリクエストに含めるオブジェクト数を半分にして、21 個の変数を問い合わせます。デバイスが稼働している場合、28 個の変数が正常に処理できることが分かっており、21 個はそれより大幅に少ないため、ほとんどの場合は問い合わせが成功します。それでも失敗する場合、Zabbix は値を 1 つずつ問い合わせる処理にフォールバックします。この時点でも失敗する場合、デバイスが応答していないことは明らかであり、リクエストサイズは問題ではありません。

次に、後続のアイテムバッチに対しては、最後に成功した変数の数（この例では 28）から開始し、上限に達するまでリクエストサイズを 1 ずつ増やします。たとえば、最大レスポンスサイズが 32 個の変数であるとすると、後続のリクエストサイズは 29、30、31、32、33 になります。最後のリクエストは失敗し、Zabbix がサイズ 33 のリクエストを再度発行することはありません。それ以降、Zabbix はこのデバイスに対して最大 32 個の変数を問い合わせます。

この変数数で大きなリクエストが失敗する場合、2 つの可能性があります。デバイスがレスポンスサイズを制限する正確な基準は分かりませんが、Zabbix では変数の数を使ってその基準を近似しようとします。1 つ目の可能性は、この変数数が、通常のケースにおけるデバイスの実際のレスポンスサイズ制限付近であることです。レスポンスが制限未満になる場合もあれば、制限を超える場合もあります。2 つ目の可能性は、いずれかの方向の UDP パケットが単純に失われたことです。このため、Zabbix は問い合わせに失敗すると、デバイスが余裕を持って処理できる範囲をさらに深く探るため、試行する変数の最大数を減らします。ただし、減らすのは最大 2 回までです。

上記の例で、32 個の変数を含む問い合わせがたまたま失敗した場合、Zabbix は数を 31 に減らします。それでも失敗した場合、Zabbix は数を 30 に減らします。ただし、Zabbix は数を 30 未満には減らしません。これ以上の失敗は、デバイスの制限ではなく、UDP パケットの損失が原因であると判断するためです。

ただし、別の理由でデバイスが結合リクエストを適切に処理できず、上記のヒューリスティックが機能しない場合は、各インターフェースにある「結合リクエストを使用」設定を使用して、そのデバイスの結合リクエストを無効にできます。

結合リクエストによって部分的または不正なレスポンスが発生し、1 秒あたりの計算（差分計算）が不正確になる場合（たとえば、インターフェースカウンターに見かけ上のスパイクが発生する場合）は、影響を受けるインターフェースで結合リクエストを使用を無効にして、アイテムごとに個別の問い合わせを強制してください。これにより、誤ったスパイクを防止できることがよくあります。または、非同期で実行され、インターフェースごとの結合リクエストを使用によるバッチ処理の対象にならない、非同期の `get[]` または `walk[]` アイテムの使用を検討してください。これらは、結合リクエストに関連する問題を回避するために、従来の同期 OID チェックの代わりに使用できます。影響を受けるデバイスを特定するには、概要セクションに示されているものと同様のサーバー/プロキシログエントリを確認してください。

さらに、インターフェースが頻繁に使用不可になる場合は、リクエストの頻度を減らすために、Zabbix サーバーまたはZabbix プロキシの設定ファイルにある `UnavailableDelay` パラメータの値を増やす必要がある場合があります。ディスカバリまたは OID ウォーク中に部分的なデータを受信すると、アイテムがサポートされていない状態になることがあります。

動的インデックス

概要

必要なインデックス番号（例えばネットワークインターフェースのもの）は SNMP OID の中から見つけられる場合がありますが、インデックス番号が常に同じままであるとは限りません。

インデックス番号は動的である可能性があり、時間の経過とともに変化して、その結果アイテムが動作しなくなることがあります。

このような状況を避けるために、インデックス番号が変化する可能性を考慮した OID を定義できます。

例えば、Cisco デバイス上の **GigabitEthernet0/1** インターフェースに対応する **ifInOctets** に付加するインデックス値を取得する必要がある場合は、次の OID を使用します。

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

構文

OID には特別な構文が使用されます:

< データの **OID>["index","<** インデックスのベース **OID>","<** 検索する文字列 **>"]**

Parameter	Description
データの OID	アイテムでデータ取得に使用するメイン OID。
index	処理方法。現在は 1 つの方法のみサポートされています: **index** - インデックスを検索し、それをデータ OID に追加します
インデックスのベース OID	この OID を参照して、文字列に対応するインデックス値を取得します。
検索する文字列	参照時に値と完全一致させるために使用する文字列。大文字と小文字は区別されます。

例

apache プロセスのメモリ使用量を取得します。

この OID 構文を使用する場合:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

インデックス番号はここで検索されます:

```
...
HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"
HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"
...
```

これで、インデックス 5388 が得られました。
目的の値を受け取るために、このインデックスがデータ OID に追加されます:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

インデックス検索キャッシュ

動的インデックスアイテムが要求されると、Zabbix は一致がより早く見つかる場合でも、インデックス用のベース OID 配下の SNMP テーブル全体を取得してキャッシュします。
これは、後で別のアイテムが同じベース OID を参照する場合に備えて行われます。この場合、Zabbix は監視対象のホストに再度問い合わせる代わりに、キャッシュ内でインデックスを検索します。
なお、各ポーラープロセスはそれぞれ独立したキャッシュを使用します。

以降のすべての値取得処理では、見つかったインデックスのみが検証されます。
変更されていなければ、値が要求されます。
変更されていた場合はキャッシュが再構築されます。インデックスの変更を検出した各ポーラーは、インデックス SNMP テーブルを再度 walk します。

特殊 OID

よく使用される SNMP OID の一部は、Zabbix によって数値表現に自動的に変換されます。たとえば、**ifIndex** は **1.3.6.1.2.1.2.2.1.1** に、**ifIndex.0** は **1.3.6.1.2.1.2.2.1.1.0** に変換されます。

この表には特殊 OID の一覧が含まれています。

Special OID	Identifier	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	各インターフェースに対する一意の値。
ifDescr	1.3.6.1.2.1.2.2.1.2	インターフェースに関する情報を含むテキスト文字列。この文字列には、製造元名、製品名、およびハードウェアインターフェースのバージョンを含める必要があります。
ifType	1.3.6.1.2.1.2.2.1.3	プロトコルスタック内でネットワーク層の直下にある物理/リンクプロトコルに基づいて区別される、インターフェースの種類。
ifMtu	1.3.6.1.2.1.2.2.1.4	インターフェース上で送受信できる最大データグラムのサイズ。オクテット単位で指定されます。
ifSpeed	1.3.6.1.2.1.2.2.1.5	インターフェースの現在の帯域幅の推定値。ビット/秒で表されます。
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	プロトコルスタック内でネットワーク層の直下にあるプロトコル層におけるインターフェースのアドレス。
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	インターフェースの現在の管理状態。
ifOperStatus	1.3.6.1.2.1.2.2.1.8	インターフェースの現在の動作状態。

Special OID	Identifier	Description
ifInOctets	1.3.6.1.2.1.2.2.1.10	フレーミング文字を含む、インターフェースで受信したオクテットの総数。
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	上位層プロトコルに配信されたサブネットワークユニキャストパケットの数。
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	上位層プロトコルに配信された非ユニキャスト（つまり、サブネットワークブロードキャストまたはサブネットワークマルチキャスト）パケットの数。
ifInDiscards	1.3.6.1.2.1.2.2.1.13	エラーは検出されなかったものの、上位層プロトコルに配信されるのを防ぐために破棄対象として選択された受信パケットの数。こうしたパケットが破棄される理由の 1 つとして、バッファ領域の解放が考えられます。
ifInErrors	1.3.6.1.2.1.2.2.1.14	上位層プロトコルに配信できない原因となるエラーを含んでいた受信パケットの数。
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	不明または未サポートのプロトコルのために破棄された、インターフェース経由で受信したパケットの数。
ifOutOctets	1.3.6.1.2.1.2.2.1.16	フレーミング文字を含む、インターフェースから送信されたオクテットの総数。
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	上位層プロトコルが送信を要求し、このサブレイヤーでマルチキャストまたはブロードキャストアドレス宛てではなかったパケットの総数。破棄されたものや送信されなかったものも含みます。
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	上位層プロトコルが送信を要求し、このサブレイヤーでマルチキャストまたはブロードキャストアドレス宛てだったパケットの総数。破棄されたものや送信されなかったものも含みます。
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	エラーは検出されなかったものの、送信されるのを防ぐために破棄対象として選択された送信パケットの数。こうしたパケットが破棄される理由の 1 つとして、バッファ領域の解放が考えられます。
ifOutErrors	1.3.6.1.2.1.2.2.1.20	エラーのために送信できなかった送信パケットの数。
ifOutQLen	1.3.6.1.2.1.2.2.1.21	出力パケットキューの長さ（パケット数）。

MIB ファイル

はじめに

MIB は、Management Information Base（管理情報ベース）を意味します。MIB ファイルを使用すると、OID（オブジェクト識別子）をテキスト形式で表現できます。Zabbix で SNMP デバイスを監視する際は、OID をそのまま使用することもできますが、テキスト形式の表現を使用する方が使いやすい場合は、MIB ファイルをインストールする必要があります。

たとえば、

```
ifHCOutOctets
```

は、次の OID のテキスト形式の表現です。

```
1.3.6.1.2.1.31.1.1.1.10
```

MIB ファイルのインストール

Debian 系システムの場合:

```
apt install snmp-mibs-downloader
download-mibs
```

RedHat 系システムの場合:

```
dnf install net-snmp-libs
```

MIB ファイルの有効化

RedHat 系のシステムでは、MIB ファイルはデフォルトで有効になっています。Debian 系のシステムでは、/etc/snmp/snmp.conf ファイルを編集し、mibs :と記載されている行をコメントアウトする必要があります。

```
# snmpパッケージはライセンス上の理由によりMIBファイルなしで提供されるため、
# デフォルトではMIBのロードは無効になっています。MIBを追加した場合は、
# 以下の行をコメントアウトすることでロードを再度有効にできます。
mibs :
```

MIB ファイルのテスト

SNMP MIB のテストは、snmpwalk ユーティリティを使用して行うことができます。インストールされていない場合は、以下の手順を使用してください。

Debian 系システムの場合:

```
apt install snmp
```

RedHat 系システムの場合:

```
dnf install net-snmp-utils
```

その後、ネットワークデバイスにクエリを実行したときに、次のコマンドでエラーが発生しないことを確認してください:

```
$ snmpwalk -v 2c -c public <NETWORK DEVICE IP> ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 176137634
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 240375057
IF-MIB::ifInOctets.4 = Counter32: 220893420
[...]
```

Zabbix での MIB の使用

最も重要な点は、Zabbix のプロセスは MIB ファイルに加えられた変更を認識しないということです。そのため、MIB ファイルを変更した後は、必ず Zabbix サーバーまたはプロキシを再起動する必要があります。例：

```
systemctl restart zabbix-server
```

これで、MIB ファイルに加えた変更が有効になります。

カスタム MIB ファイルの使用

すべての GNU/Linux ディストリビューションには標準の MIB ファイルが付属しています。しかし、一部のデバイスベンダーは独自の MIB ファイルを提供しています。

たとえば、CISCO-SMI MIB ファイルを使用したいとします。以下の手順でダウンロードおよびインストールできます。

```
wget ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my -P /tmp
mkdir -p /usr/local/share/snmp/mibs
grep -q '^mibdirs +/usr/local/share/snmp/mibs' /etc/snmp/snmp.conf 2>/dev/null || echo "mibdirs +/usr/loca
cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

これで使用できるようになります。MIB ファイルからオブジェクト *ciscoProducts* の名前を OID に変換してみましょう。

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

OID の代わりにエラーが表示される場合は、前述のコマンドがすべてエラーなく実行されたことを確認してください。

オブジェクト名の変換が成功した場合、カスタム MIB ファイルを使用する準備ができています。クエリで使用されている MIB 名のプレフィックス（*CISCO-SMI::*）に注意してください。これはコマンドラインツールや Zabbix で使用する際にも必要になります。

この MIB ファイルを Zabbix で使用する前に、Zabbix サーバー/プロキシを再起動することを忘れないでください。

> **Attention:**
> MIB ファイルには依存関係がある場合があることに注意してください。つまり、ある MIB が別の MIB を必要とする場合があります。これらの依存関係を満たすためには、影響を受けるすべての MIB ファイルをインストールする必要があります。

SNMP トラップ

概要

SNMP トラップの受信は、SNMP 対応デバイスに問い合わせるのとは逆の動作です。

この場合、情報は SNMP 対応デバイスから snmptrapd に送信され、Zabbix サーバーまたは Zabbix プロキシによってファイルから収集、つまり「トラップ」されます。

通常、トラップは何らかの状態変化時に送信され、エージェントはポート 162 でサーバーに接続します（問い合わせに使用されるエージェント側のポート 161 とは異なります）。トラップを使用すると、問い合わせ間隔の間に発生して問い合わせデータでは見逃される可能性のある短時間の障害を検出できる場合があります。

Zabbix における SNMP トラップの受信は、snmptrapd と、トラップを Zabbix に渡すためのいずれかの仕組み、つまり Bash または Perl スクリプト、あるいは SNMPTT と連携するように設計されています。

Note:
Zabbix の設定後にトラップ監視を構成する最も簡単な方法は Bash スクリプトのソリューションを使うことです。Perl と SNMPTT は現代のディストリビューションでは含まれていないことが多く、より複雑な設定が必要になるためです。ただし、このソリューションでは `traphandle` として設定されたスクリプトを使用します。本番環境でより高い性能を得るには、組み込み Perl のソリューション（`do perl` オプションを使うスクリプト、または SNMPTT）を使用してください。

トラップ受信のワークフロー:

1. `snmptrapd` がトラップを受信します。
2. `snmptrapd` はトラップを受信側スクリプト（Bash、Perl）または SNMPTT に渡します。
3. 受信側はトラップを解析し、整形してファイルに書き込みます。
4. Zabbix の SNMP trapper がトラップファイルを読み取り、解析します。
5. 各トラップについて、Zabbix は受信したトラップのアドレスに一致するホストインターフェースを持つすべての SNMP trapper アイテムを検索します。この照合では、ホストインターフェースで選択された *IP* または *DNS* のみが使用されることに注意してください。
6. 見つかった各アイテムについて、トラップは `snmptrap[regexp]` の正規表現と比較されます。トラップは一致したすべてのアイテムの値として設定されます。一致するアイテムが見つからず、`snmptrap.fallback` アイテムがある場合は、トラップはその値として設定されます。
7. トラップがいずれのアイテムの値としても設定されなかった場合、Zabbix は既定で一致しなかったトラップをログに記録します。（これは 管理 > 一般 > その他の 一致しない *SNMP* トラップをログに記録で設定されます。）

HA フェイルオーバーに関する注意事項

高可用性 (HA) ノードの切り替え中、Zabbix は最後の ISO 8601 タイムスタンプ内の最後のレコードの後に処理を続行します。同じレコードが見つからない場合は、タイムスタンプのみを使用して最後の位置を識別します。

SNMP トラップの設定

このアイテムタイプには、次の Web インターフェース設定が必要です。

手順 1: ホストに SNMP インターフェースを作成する

1. データ収集 > ホストで、ホストを作成するか、既存のホストを編集します。
2. インターフェースフィールドで、*SNMP* インターフェースタイプを選択します。
3. IP アドレス/DNS 名とポート番号を入力します。
4. ドロップダウンから *SNMP* バージョンを選択し、(*SNMPv1*, *SNMPv2* または*SNMPv3*)、選択した SNMP バージョンに応じてインターフェースの認証情報を追加します。

受信した各トラップのアドレスは、対応するホストを見つけるために、すべての SNMP インターフェースの IP アドレスおよび DNS アドレスと比較されます。

手順 2: アイテムを設定する

1. ホストに対して、アイテムを作成するか、既存のアイテムを編集します。
2. キーフィールドで、次の SNMP トラップキーのいずれかを使用します。
   - `snmptrap[regexp]` - `regexp` パラメータで指定された正規表現に一致するすべての SNMP トラップを取得します。`regexp` が指定されていない場合は、任意のトラップを取得します。
 パラメータではユーザーマクロとグローバル正規表現がサポートされています。
 戻り値: SNMP トラップ。
 このアイテムは SNMP インターフェースに対してのみ設定できます。
   - `snmptrap.fallback` - そのインターフェースの `snmptrap[]` アイテムのいずれにも取得されなかったすべての SNMP トラップを取得します。
 戻り値: SNMP トラップ。
 このアイテムは SNMP インターフェースに対してのみ設定できます。

Note:
現時点では、複数行の正規表現マッチングはサポートされていません。

3. タイムスタンプを解析するには、情報の種類を ログに設定します。数値などの他の形式も使用できますが、カスタムトラップハンドラーが必要になる場合があります。

SNMP トラップ監視の設定

Zabbix サーバー/プロキシの設定

トラップを読み取るには、Zabbix サーバーまたはプロキシで SNMP trapper プロセスを起動するように設定し、SNMPTT または Bash/Perl のトラップ受信側によって書き込まれるトラップファイルを指定する必要があります。
そのためには、設定ファイル (`zabbix_server.conf` または`zabbix_proxy.conf`) を編集します。

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
systemd のパラメータ PrivateTmp が使用されている場合、このファイルは /tmp では動作しない可能性が高いです。

Bash トラップ受信側の設定

要件: snmptrapd のみ。

Bash トラップ受信側の スクリプト を使用して、trapper ファイルを介し、snmptrapd から Zabbix サーバーへトラップを渡すことができます。
これを設定するには、snmptrapd の設定ファイル (snmptrapd.conf) に traphandle オプションを追加します。
例 を参照してください。

Note:
設定の変更を反映するには、snmptrapd の再起動が必要になる場合があります。

Perl トラップ受信側の設定

要件: Perl、--enable-embedded-perl 付きでコンパイルされた Net-SNMP（Net-SNMP 5.4 以降では既定で有効）

Perl トラップ受信側（misc/snmptrap/zabbix_trap_receiver.pl を参照）は、snmptrapd から Zabbix サーバーへトラップを直接渡すために使用できます。
設定するには、次の手順を実行します。

- Perl スクリプトを snmptrapd の設定ファイル（snmptrapd.conf）に追加します。例:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- 受信側を設定します。例:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
設定の変更を反映するには、snmptrapd の再起動が必要な場合があります。

Note:
スクリプト名が引用符で囲まれていない場合、snmptrapd は起動を拒否し、次のようなメッセージを表示します。


```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

SNMPTT の設定

まず、snmptrapd を SNMPTT を使用するように設定する必要があります。

Note:
最適なパフォーマンスを得るには、SNMPTT をデーモンとして設定し、snmptthandler-embedded を使用してトラップを渡すようにしてください。手順については、SNMPTT の設定を参照してください。

SNMPTT がトラップを受信するように設定したら、snmptt.ini を設定します。

1. NET-SNMP パッケージの Perl モジュールの使用を有効にします。

```
net_snmp_perl_enable = 1
```

2. Zabbix が読み取るトラップファイルにトラップを記録します。

```
log_enable = 1
log_file = [TRAP FILE]
```

3. 日時形式を設定します。

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
net-snmp-perl パッケージは RHEL 8.0-8.2 では削除されており、RHEL 8.3 で再追加されました。詳細については、既知の問題を参照してください。

次に、Zabbix が認識できるようにトラップを整形します（snmptt.conf を編集します）。

1. 各 FORMAT 文は ZBXTRAP [address] で始める必要があります。ここで [address] は、Zabbix 上の SNMP インターフェースの IP アドレスおよび DNS アドレスと比較されます。例:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. SNMP トラップ形式の詳細については、以下を参照してください。

Attention:
不明なトラップは使用しないでください。Zabbix はそれらを認識できません。不明なトラップは、snmptt.conf で一般イベントを定義することで処理できます。


```
EVENT general .* "General event" Normal
```

SNMP トラップ形式

すべてのカスタマイズされた Perl トラップ受信側と SNMPTT のトラップ設定は、トラップを次の形式で出力する必要があります。

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

ここで

- [timestamp] - %Y-%m-%dT%H:%M:%S%z 形式のタイムスタンプ。
- ZBXTRAP - この行で新しいトラップが始まることを示すヘッダー。
- [address] - このトラップのホストを見つけるために使用される IP アドレス。

ZBXTRAP と [address] は、処理中にメッセージから切り取られることに注意してください。トラップが別の形式で出力されると、Zabbix がトラップを予期しない方法で解析する可能性があります。

トラップの例:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link
```

これは、IP=192.168.1.1 の SNMP インターフェースに対して、次のトラップになります。

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

システム要件

Note:
アイテム値が正しい形式で表示されるように、MIB ファイルをインストールすることを推奨します。MIB ファイルがない場合、値が UTF-8 ではなく HEX で表示される、またはその逆など、フォーマットの問題が発生する可能性があります。

大きなファイルのサポート

Zabbix は SNMP トラッパーファイルの大きなファイルをサポートしています。Zabbix が読み取れる最大ファイルサイズは 2^63 (8 EiB) です。ファイルシステムによっては、ファイルサイズにより低い上限が課される場合があることに注意してください。

ログローテーション

Zabbix はログローテーションの仕組みを提供していません。これはユーザー側で対応する必要があります。ログローテーションでは、まず古いファイルの名前を変更し、その後で削除するようにしてください。そうすることで、トラップが失われることを防げます。

1. Zabbix は最後に認識していた場所でトラップファイルを開き、手順 3 に進みます。
2. Zabbix は、現在開いているファイルの inode 番号を定義されたトラップファイルの inode 番号と比較して、ローテーションされたかどうかを確認します。開いているファイルがない場合、Zabbix は最後の場所をリセットして手順 1 に戻ります。
3. Zabbix は現在開いているファイルからデータを読み取り、新しい場所を設定します。
4. 新しいデータが解析されます。これがローテーションされたファイルであった場合、ファイルは閉じられ、手順 2 に戻ります。
5. 新しいデータがない場合、Zabbix は 1 秒間スリープし、手順 2 に戻ります。

ファイルシステム

trap ファイルの実装のため、Zabbix ではファイルを区別するために inode をサポートするファイルシステムが必要です（この情報は stat() 呼び出しによって取得されます）。

異なる SNMP プロトコルバージョンを使用したセットアップ例

この例では、snmptrapd と Bash の受信スクリプトを使用して、トラップを Zabbix サーバーに渡します。

セットアップ:

1. Zabbix で SNMP トラッパーを起動し、トラップファイルを設定します。zabbix_server.conf に以下を追加します。

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Bash スクリプトを/usr/sbin/zabbix_trap_handler.sh にダウンロードします。

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/temp
```

必要に応じて、スクリプト内の ZABBIX_TRAPS_FILE 変数を調整します。デフォルト値を使用する場合は、先に親ディレクトリを作成します。

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. snmtrapd.conf に以下を追加します（動作する例を参照してください）。

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> 設定の変更を反映するため、snmptrapd の再起動が必要になる場合があります。

4. SNMP アイテム TEST を作成します（最初に設定要件を満たす必要があることに注意してください）。

   Type: *SNMP trap*
 Type of information: *Log*
 Host interface: *SNMP 127.0.0.1*
 Key: snmptrap["linkUp"]
 Log time format: yyyyMMdd.hhmmss

ISO 8601 の日付と時刻の形式が使用されることに注意してください。

5. 次に、選択した SNMP プロトコルバージョン用に snmptrapd を設定し、snmptrap ユーティリティを使用してテストトラップを送信します。

SNMPv1、SNMPv2

SNMPv1 および SNMPv2 プロトコルは、「コミュニティ文字列」認証に依存します。以下の例では、コミュニティ文字列として「secret」を使用します。SNMP トラップ送信元でも同じ値を設定する必要があります。

SNMPv2 は現在も本番環境で広く使用されていますが、暗号化や実際の送信元認証は提供しません。データはプレーンテキストで送信されるため、これらのプロトコルバージョンはプライベートネットワークなどの安全な環境でのみ使用し、パブリックネットワークや第三者のネットワーク経由では決して使用しないでください。

SNMP バージョン 1 は 64 ビットカウンターをサポートしていないため、現在ではほとんど使用されておらず、レガシープロトコルと見なされています。

SNMPv1 または SNMPv2 トラップの受信を有効にするには、snmptrapd.conf に以下の行を追加します。secret を、SNMP トラップ送信元で設定した SNMP コミュニティ文字列に置き換えてください。

```
authCommunity log,execute,net secret
```

次に、snmptrap を使用してテストトラップを送信します。この例では、一般的な「リンクアップ」OID を使用します。

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 は SNMPv1/v2 のセキュリティ上の問題に対処し、認証と暗号化を提供します。認証方式として MD5 または複数の SHA 方式を、暗号方式として DES または複数の AES 方式を使用できます。

snmptrapd.conf に以下の行を追加して、SNMPv3 トラップを受信するように snmptrapd を設定します。

```
##### 認証とプライバシーを使用する SNMPv3 USM ユーザーを作成
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
##### ユーザーによるトラップのログ記録とトラップハンドラースクリプトの実行を許可
authUser log,execute <securityName>
```

例:

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
このユーザーセキュリティモデルでスクリプトを実行できるようにする「execute」キーワードに注意してください。

SNMPv3 トラップの設定をテストするには、以下のコマンドを使用してテストトラップを送信します。

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x8000000
```

Warning:
AES192 や AES256 などの強力な暗号化方式を使用する場合は、バージョン 5.8 以降の net-snmp を使用してください。configure オプションに --enable-blumenthal-aes を指定して再コンパイルする必要がある場合があります。古いバージョンの net-snmp は AES192/AES256 をサポートしていません。詳細はStrong Authentication or Encryptionも参照してください。

確認

どちらの例でも、/var/lib/zabbix/snmptraps/snmptraps.log に次のような行が表示されます。

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Zabbix のアイテム値は次のようになります。

```
2024-01-30 10:04:23 2024-01-30 10:04:21

2024-01-30T10:04:21+0200 UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Perl を使用した例:

```
2024-01-30T11:42:54+0200 ZBXTRAP 127.0.0.1
PDU INFO:
  receivedfrom                    UDP: [127.0.0.1]:58649->[127.0.0.1]:162
  notificationtype                TRAP
  version                         1
  community                       public
  errorstatus                     0
  transactionid                   1
  requestid                       2101882550
  messageid                       0
  errorindex                      0
VARBINDS:
  DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (457671) 1:16:16.71
  SNMPv2-MIB::snmpTrapOID.0       type=6  value=OID: IF-MIB::linkUp.0
```

参照

- Zabbix blog article on SNMP traps
- Configuring snmptrapd (net-snmp の公式ドキュメント)
- Configuring snmptrapd to receive SNMPv3 notifications (net-snmp の公式ドキュメント)

Zabbix 内部

概要

内部チェックを使用すると、Zabbix の内部プロセスを監視できます。つまり、Zabbix サーバーまたは Zabbix プロキシで発生している処理を監視できます。

内部チェックは、次の場所で計算されます。

- Zabbix サーバー上 - ホストがサーバーによって監視されている場合
- Zabbix プロキシ上 - ホストがプロキシによって監視されている場合

内部チェックは、ホストのメンテナンスステータスに関係なく、サーバーまたはプロキシによって処理されます。

このアイテムを使用するには、**Zabbix internal** アイテムタイプを選択します。

Note:
内部チェックは、Zabbix ポーラーによって処理されます。

パフォーマンス

一部の内部アイテムは、パフォーマンスに悪影響を及ぼす可能性があります。これらのアイテムは以下の通りです:

- zabbix[host,,items]
- zabbix[host,,items_unsupported]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[queue,,]
- zabbix[requiredperformance]
- zabbix[stats,,,queue,,]
- zabbix[triggers]

システム情報およびキューの Web インターフェースセクションも影響を受けます。

サポートされているチェック

アイテムキーは、カスタマイズ可能なパラメータおよび追加情報を省略して一覧表示されています。アイテムキーをクリックすると、詳細を確認できます。

アイテムキー	説明
zabbix[apm,]	APM コレクターが受信したテレメトリデータの統計。
zabbix[boottime]	Zabbix サーバーまたは Zabbix プロセスの起動時間（秒）。
zabbix[cluster,discovery,nodes]	高可用性クラスターのノードを検出します。
zabbix[connector_queue]	コネクターキューに登録された値の数。
zabbix[discovery_queue]	ディスカバリキューに登録されたネットワークチェックの数。
zabbix[host,,items]	ホスト上で有効になっているアイテム（サポート対象および未サポート）の数。
zabbix[host,,items_unsupported]	ホスト上で有効になっている未サポートのアイテム数。
zabbix[host,maintenance]	ホストの現在のメンテナンスステータス。
zabbix[host,active_agent,available]	ホスト上のアクティブエージェントチェックの可用性。
zabbix[host,discovery,interfaces]	Zabbix Web インターフェースで設定されているホストのすべてのインターフェースの詳細。
zabbix[host,available]	ホスト上の特定の種類のチェックに使用されるメインインターフェースの可用性。
zabbix[hosts]	監視対象ホストの数。
zabbix[items]	有効になっているアイテム（サポート対象および未サポート）の数。
zabbix[items_unsupported]	未サポートのアイテム数。
zabbix[java,]	Zabbix Java ゲートウェイに関する情報。
zabbix[lld_queue]	ローレベルディスカバリ処理キューに登録された値の数。
zabbix[preprocessing]	前処理マネージャーが受信した値の統計。
zabbix[preprocessing_queue]	前処理キューに登録された値の数。
zabbix[process,,]	特定の Zabbix プロセス、または <type> と <mode> で識別されるプロセスグループが <state> に費やした時間の割合。
zabbix[proxy,]	Zabbix プロキシに関する情報。
zabbix[proxy,discovery]	Zabbix プロキシの一覧。
zabbix[proxy group,available]	プロキシグループ内でオンラインになっているプロキシの数。
zabbix[proxy group,pavailable]	プロキシグループ内でオンラインになっているプロキシの割合。
zabbix[proxy group,proxies]	プロキシグループ内の Zabbix プロキシの一覧。
zabbix[proxy group,state]	プロキシグループの状態。
zabbix[proxy group,discovery]	設定データおよびリアルタイムデータを含むプロキシグループの一覧を返します。
zabbix[proxy_buffer,buffer,]	プロキシメモリバッファーの使用状況の統計を返します。
zabbix[proxy_buffer,state,changes]	開始以降のディスク/メモリバッファーモード間の状態変更回数を返します。
zabbix[proxy_buffer,state,current]	新しいデータが保存されている現在の動作状態を返します。
zabbix[proxy_history]	サーバーへの送信待ちとなっている、プロキシのヒストリーテーブル内の値の数。
zabbix[queue,]	<from> 秒以上、<to> 秒未満遅延している、キュー内の監視対象アイテム数。
zabbix[rcache,]	Zabbix 設定キャッシュの可用性の統計。

アイテムキー	説明
zabbix[requiredperformance]	1 秒あたりに想定される新しい値の数で表した、Zabbix サーバーまたは Zabbix プロキシに必要なパフォーマンス。
zabbix[stats,,]	Zabbix サーバーまたはプロキシの内部メトリクスを返します。<ip> と <port> が指定されている場合はリモートインスタンスから、それ以外の場合はローカルインスタンスからメトリクスを取得します。
zabbix[stats,,,queue,]	Zabbix サーバーまたはプロキシの内部キューメトリクスを返します。<ip> と <port> が指定されている場合はリモートインスタンスから、それ以外の場合はローカルインスタンスからメトリクスを取得します。
zabbix[tcache,]	Zabbix トレンド関数キャッシュの有効性の統計。
zabbix[triggers]	有効なホスト上ですべてのアイテムが有効になっている場合の、Zabbix データベース内の有効なトリガー数。
zabbix[uptime]	Zabbix サーバーまたはプロキシプロセスの稼働時間（秒）。
zabbix[vcache,buffer,]	Zabbix value cache の可用性の統計。
zabbix[vcache,cache,]	Zabbix value cache の有効性の統計。
zabbix[version]	Zabbix サーバーまたはプロキシのバージョン。
zabbix[vmware,buffer,]	Zabbix VMware キャッシュの可用性の統計。
zabbix[vps,written]	データベースに書き込まれたヒストリー値の総数。
zabbix[wcache,]	Zabbix 書き込みキャッシュの統計および可用性。

アイテムキーの詳細

- 山括弧のないパラメータは必須であり、そのまま使用する必要があります（例：zabbix[host,<type>,available] の”host” と”available”）。
- 山括弧（**< >**）付きのパラメータは、有効な値に置き換える必要があります。パラメータにデフォルト値がある場合、省略できます。
- ” プロキシではサポートされていません” とラベル付けされたアイテムやアイテムパラメータの値は、ホストがサーバーによって監視されている場合にのみ取得できます。逆に、” サーバーではサポートされていません” の値は、ホストがプロキシによって監視されている場合にのみ取得できます。

zabbix[apm,<type>]

 APM コレクターが受信したテレメトリデータの統計情報:

- *requests* - APM コレクターが受信したリクエスト数（カウンター）：
  - *accepted* - 処理のために受け入れられたリクエスト数;
  - *dropped* - 破棄されたリクエスト数;
- *rows* - データ型ごとにデータソースへ書き込まれたテレメトリデータ行数（カウンター）。

戻り値: *JSON* オブジェクト。

パラメータ:

- **type** - *traces*、*metrics*、または *logs*。指定しない場合、すべてのタイプの統計情報が返されます。

コメント:

- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[apm] の戻り値の例:

```
{
  "requests": {
    "accepted": 15230,
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
    "metrics_gauge": 31250,
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```

zabbix[boottime]

 Zabbix サーバーまたは Zabbix プロキシプロセスの起動時刻 (秒単位)。
 戻り値: 整数。

zabbix[cluster,discovery,nodes]

 高可用性クラスターのノードを検出します。
 戻り値: *JSON* オブジェクト。

コメント:

- このアイテムはローレベルディスカバリで使用できます。
- 返されるフィールドは、主に高可用性ノードオブジェクトのプロパティと一致します。
- 追加フィールド:
  - `db_timestamp` - レスポンスに含まれる現在のサーバー時刻（Unix タイムスタンプ）
  - `lastaccess_age` - 最後のノードハートビートからの経過秒数（db_timestamp - lastaccess）
  - `status` - ノードの状態:
    * 0 - スタンバイ
    * 1 - 手動停止
    * 2 - 使用不可
    * 3 - アクティブ

戻り値の例:

```
[
  {
    "id": "ckvupihk70001z8mkpw5cg0u3",
    "name": "zabbix-prod-01",
    "status": 3,
    "address": "10.0.4.12:10051",
    "port": 10051,
    "lastaccess": 1756115995,
    "db_timestamp": 1756116000,
    "lastaccess_age": 5
  },
  {
    "id": "ckvx2a9k70004b1nq2hz9d7f",
    "name": "zabbix-standby-02",
    "status": 0,
    "address": "10.0.4.13:10051",
    "port": 10051,
    "lastaccess": 1756115550,
    "db_timestamp": 1756116000,
    "lastaccess_age": 450
  },
  {
    "id": "ckw0bq3l70007y4r1a0m5kz8",
    "name": "zabbix-backup-eu1",
    "status": 1,
    "address": "backup.example.com:10051",
    "port": 10051,
    "lastaccess": 1756105080,
    "db_timestamp": 1756116000,
    "lastaccess_age": 10920
  }
]
```

zabbix[connector_queue]

 コネクタキューにエンキューされた値の数。
 戻り値: 整数。

zabbix[discovery_queue]

 ディスカバリキューに入っているネットワークチェックの数。
 戻り値: 整数。

zabbix[host,,items]

 ホスト上で有効なアイテム数（サポートされているものとサポートされていないものの両方）。
 戻り値: 整数。

zabbix[host,,items_unsupported]

 ホストで有効なサポートされていないアイテムの数。
 戻り値: 整数。

zabbix[host,,maintenance]

 ホストの現在のメンテナンスステータス。
 戻り値: *0* - 通常状態; *1* - データ収集ありのメンテナンス; *2* - データ収集なしのメンテナンス。

コメント:

- このアイテムは、ホストの場所（サーバーまたはプロキシ上）に関係なく、常に Zabbix サーバーによって処理されます。プロキシはこのアイテムを設定データとして受信しません。
- 2 番目のパラメータは空でなければならず、将来の使用のために予約されています。

zabbix[host,active_agent,available]

 ホスト上のアクティブエージェントチェックの可用性。
 戻り値: *0* - 不明; *1* - 利用可能; *2* - 利用不可。

zabbix[host,discovery,interfaces]

 Zabbix フロントエンドで設定されたホストのすべてのインターフェースの詳細。
 戻り値: *JSON* オブジェクト。

コメント:

- このアイテムはローレベルディスカバリで使用できます。
- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[host,<type>,available]

 ホスト上の特定のタイプのチェックにおけるメインインターフェースの可用性。
 戻り値: *0* - 利用不可; *1* - 利用可能; *2* - 不明。

パラメータ:

- **type** - *agent*、*snmp*、*ipmi*、または *jmx*。

コメント:

- アイテムの値は、ホストの到達不能/利用不可に関する設定パラメータに従って計算されます。

zabbix[hosts]

 監視対象ホストの数。
 戻り値: 整数。

zabbix[items]

 有効なアイテム数（サポートされているものとサポートされていないもの）。
 戻り値: 整数。

zabbix[items_unsupported]

 サポートされていないアイテムの数。
 戻り値: 整数。

zabbix[java,,<param>]

 Zabbix Java ゲートウェイに関する情報。
 戻り値: *1* - <param> が *ping* の場合; *Java gateway version* - <param> が *version* の場合（例: ”8.0.0”）。

Parameters:

- **param** - *ping* または *version*。

Comments:

- このアイテムは、nodata() トリガー関数を使用して Java ゲートウェイの可用性を確認するために使用できます。
- 2 番目のパラメータは空である必要があり、将来の使用のために予約されています。

zabbix[lld_queue]

 低レベルディスカバリ処理キューにエンキューされた値の数。
 戻り値: 整数。

コメント:

- このアイテムは、低レベルディスカバリ処理キューの長さを監視するために使用できます。

zabbix[preprocessing]

 前処理マネージャーによって受信された値の統計情報:

- *queued* - 前処理が必要なキューイングされた値の数とサイズ (カウンター)
- *direct* - 前処理が不要なキューイングされた値の数とサイズ (カウンター)
- *queue* - 前処理キューにキューイングされた値の数 (zabbix[preprocessing_queue] と同じ)

戻り値: *JSON*。

戻り値の例:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 前処理キューに入れられた値の数。
 戻り値: 整数。

コメント:

- このアイテムは前処理キューの長さを監視するために使用できます。

zabbix[process,<type>,<mode>,<state>]

 特定の Zabbix プロセス、またはプロセスのグループ（<type> および <mode> で識別）が <state> で費やした時間の割合です。直前の 1 分間のみを対象に計算されます。
 戻り値: *Float*。

パラメータ:

- **type**:
  - サーバープロセスの場合: *agent poller*、*alert manager*、*alert syncer*、*alerter*、*availability manager*、*browser poller*、*configuration syncer*、*configuration syncer worker*、*connector manager*、*connector worker*、*discovery manager*、*discovery worker*、*escalator*、*ha manager*、*history poller*、*history syncer*、*housekeeper*、*http agent poller*、*http poller*、*icmp pinger*、*internal poller*、*ipmi manager*、*ipmi poller*、*java poller*、*lld manager*、*lld worker*、*odbc poller*、*poller*、*preprocessing manager*、*preprocessing worker*、*proxy group manager*、*proxy poller*、*self-monitoring*、*service manager*、*snmp poller*、*snmp trapper*、*task manager*、*timer*、*trapper*、*trigger housekeeper*、*unreachable poller*、*vmware collector*。
  - プロキシプロセスの場合: *agent poller*、*apm manager*、*apm worker*、*availability manager*、*browser poller*、*configuration syncer*、*data sender*、*discovery manager*、*discovery worker*、*history syncer*、*housekeeper*、*http agent poller*、*http poller*、*icmp pinger*、*internal poller*、*ipmi manager*、*ipmi poller*、*java poller*、*odbc poller*、*poller*、*preprocessing manager*、*preprocessing worker*、*self-monitoring*、*snmp poller*、*snmp trapper*、*task manager*、*trapper*、*unreachable poller*、*vmware collector*。
- **mode**:
  - *avg* - 指定したタイプのすべてのプロセスの平均値（デフォルト）。
  - *count* - 指定したプロセスタイプのフォーク数を返します。<state> は指定しないでください。
  - *max* - 最大値。
  - *min* - 最小値。
  - *<process number>* - プロセス番号（1 から、フォーク前に作成されたインスタンス数までの範囲。たとえば、4 つの trapper が実行されている場合、値は 1 から 4 の範囲です）。
- **state**:
  - *busy* - プロセスがビジー状態にある場合。たとえば、リクエストを処理している状態です（デフォルト）。
  - *idle* - プロセスが何もせずアイドル状態にある場合。

コメント:

- <mode> に、実行されていない Zabbix プロセス番号を指定した場合（たとえば、5 つの poller が実行されている状態で <mode> に 6 を指定した場合）、このようなアイテムはサポート対象外になります。
- 最小値と最大値は、単一プロセスの使用率を指します。したがって、3 つの poller のグループでプロセスごとの使用率が 2、18、66 だった場合、min は 2 を返し、max は 66 を返します。
- プロセスは共有メモリに自身の処理内容を報告し、セルフモニタリングプロセスがそのデータを毎秒集計します。状態の変更（busy/idle）は変更時に登録されます。そのため、busy になったプロセスは busy として登録され、idle になるまで状態を変更または更新しません。これにより、完全にハングしたプロセスも 100% busy として正しく登録されます。
- 現在、「busy」は「スリープしていない」ことを意味しますが、将来的には、ロック待ち、データベースクエリの実行中など、追加の状態が導入される可能性があります。非同期 poller は、サーバー/プロキシの設定パラメータ MaxConcurrentChecksPerPoller で設定された上限に達している場合、busy とみなされることに注意してください。

- Linux およびその他のほとんどのシステムでは、分解能は 1/100 秒です。

例:

```
zabbix[process,poller,avg,busy] # 直前の 1 分間に poller プロセスが何らかの処理に費やした時間の平均
zabbix[process,"icmp pinger",max,busy] # 直前の 1 分間にいずれかの ICMP pinger プロセスが何らかの処理に費やした時間の
zabbix[process,"history syncer",2,busy] # 直前の 1 分間に 2 番目の history syncer が何らかの処理に費やした時間
zabbix[process,trapper,count] # 現在実行中の trapper プロセス数
```

zabbix[proxy,<name>,<param>]

 Zabbix プロキシに関する情報です。
 戻り値: *Integer*。

パラメータ:

- **name** - プロキシ名
- **param**:
  - *lastaccess* - プロキシから最後に受信したハートビートメッセージのタイムスタンプ
  - *delay* - 収集された値が未送信のままになっている時間。「プロキシの遅延」+ (" 現在のサーバー時刻" - " プロキシの lastaccess") として計算されます。ここで「プロキシの遅延」は、現在のプロキシ時刻と、プロキシ上で未送信となっている最も古い値のタイムスタンプとの差です。

コメント:

- このアイテムは、プロキシまたはプロキシグループによって監視されているホストではサポートされません。
- プロキシの可用性を確認するには、`fuzzytime()`関数を使用できます。

例:

```
zabbix[proxy,"Germany",lastaccess] #「Germany」プロキシから最後に受信したハートビートメッセージのタイムスタンプ
```

zabbix[proxy,discovery]

 名前、モード、暗号化、圧縮、バージョン、最終確認時刻、ホスト数、アイテム数、必要な 1 秒あたりの値（vps)、バージョンステータス（最新/古い/サポート対象外)、アイテムタイプごとのタイムアウト、プロキシグループ名（プロキシがグループに所属している場合)、状態（不明/オフライン/オンライン）を含む Zabbix プロキシの一覧です。
 戻り値: *JSON* オブジェクト。

コメント:

- このアイテムは、プロキシまたはプロキシグループによって監視されているホストではサポートされません。

戻り値の例:

```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
```

```
  }
]
```

zabbix[proxy group,<name>,available]

 プロキシグループ内のオンラインプロキシ数。
 戻り値: *Integer*。

パラメータ:

- **name** - プロキシグループ名。

コメント:

- このアイテムは、プロキシまたはプロキシグループによって監視されているホストではサポートされません。

zabbix[proxy group,<name>,pavailable]

 プロキシグループ内のオンラインプロキシの割合。
 戻り値: *Float*。

パラメータ:

- **name** - プロキシグループ名。

コメント:

- このアイテムは、プロキシまたはプロキシグループによって監視されているホストではサポートされません。

zabbix[proxy group,<name>,proxies]

 名前、モード、暗号化、圧縮、バージョン、最終確認時刻、ホスト数、アイテム数、必要な 1 秒あたりの値（vps）、バージョンステータス（最新/古い/サポート対象外）、タイムアウト、プロキシグループ名、状態（不明/オフライン/オンライン）を含む、プロキシグループ内の Zabbix プロキシの一覧です。
 戻り値: *JSON*。

パラメーター:

- **name** - プロキシグループ名。

コメント:

- このアイテムは、プロキシまたはプロキシグループによって監視されているホストではサポートされません。

zabbix[proxy group,<name>,state]

 プロキシグループの状態。
 戻り値: *0* - 不明; *1* - オフライン; *2* - 復旧中; *3* - オンライン; *4* - 劣化中。

パラメータ:

- **name** - プロキシグループ名。

コメント:

- このアイテムは、プロキシまたはプロキシグループによって監視されているホストではサポートされません。

zabbix[proxy group,discovery]

 構成データおよびリアルタイムデータを含むプロキシグループの一覧を返します。構成データには、プロキシグループ名、フェイルオーバー遅延、およびオンラインであることが必要なプロキシの最小数が含まれます。リアルタイムデータには、プロキシグループの状態（詳細についてはコメントを参照）、オンラインプロキシの数、およびオンラインプロキシの割合が含まれます。
 戻り値: *JSON*。

コメント:

- プロキシまたはプロキシグループによって監視されているホストでは、このアイテムはサポートされません。
- このアイテムは、グループに所属していないプロキシを返しません。
- `failover_delay` または `min_online` に無効な値が設定されている場合、それを示す特別な値 −1 が報告されます。構成にマクロを使用していて、そのマクロを有効な値に展開できない場合に、無効な値が発生することがあります。
- プロキシグループの状態は整数で報告されます: *0* - 不明、*1* - オフライン、*2* - 復旧中、*3* - オンライン、*4* - 劣化。

戻り値の例:

```
{
    "groups": [
       { "name": "Riga", "failover_delay": 60, "min_online": 1 },
       { "name": "Tokyo", "failover_delay": 60, "min_online": 2 },
       { "name": "Porto Alegre", "failover_delay": 60, "min_online": 3 }
    ],
    "details": {
```

```
        "Riga": { "state": 3, "available": 10, "pavailable": 20 },
        "Tokyo": { "state": 3, "available": 10, "pavailable": 20 },
        "Porto Alegre": { "state": 1, "available": 0, "pavailable": 0 }
    }
}
```

zabbix[proxy_buffer,buffer,<mode>]

 プロキシのメモリバッファー使用状況の統計情報。
 戻り値: *Integer*（サイズの場合）、*Float*（パーセンテージの場合）。

パラメータ:

- **mode**:
  - *total* - バッファーの合計サイズ（メモリバッファーが有効かどうかの確認に使用できます）;
  - *free* - 空きバッファーのサイズ;
  - *pfree* - 空きバッファーのパーセンテージ;
  - *used* - 使用中のバッファーのサイズ;
  - *pused* - 使用中のバッファーのパーセンテージ。

コメント:

- メモリバッファーが無効な場合、「Proxy memory buffer is disabled」エラーを返します;
- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[proxy_buffer,state,changes]

 起動後、ディスク/メモリバッファモード間の状態変更回数を返します。
 戻り値: 整数; *0* - メモリバッファが無効です。

コメント:

- 状態変更が頻繁に発生する場合は、メモリバッファのサイズまたは有効期間を増やす必要があります。
- メモリバッファの状態を監視する頻度が低い場合 (例えば 1 分に 1 回) は、バッファが状態を切り替えても記録されないことがあります。

zabbix[proxy_buffer,state,current]

 新しいデータが保存されている現在の動作状態を返します。
 戻り値: *0* - ディスク、*1* - メモリ。

コメント:

- メモリバッファーが無効になっている場合も 0 が返されます。

zabbix[proxy_history]

 サーバーに送信されるのを待機しているプロキシ履歴テーブル内の値の数。
 戻り値: *Integer*。

コメント:

- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[queue,<from>,<to>]

 キュー内で監視されているアイテムのうち、<from> 秒以上遅延しているが、<to> 秒未満の遅延であるものの数。
 戻り値: *Integer*。

パラメーター:

- **from** - 少なくともこれだけ遅延している（デフォルトは 6 秒）;
- **to** - 最大でこれだけ遅延している（デフォルトは無限大）。

コメント:

- パラメーターではTime suffixes (`s,m,h,d,w`) がサポートされています。

> **Attention:**
> Zabbix プロキシはメンテナンス期間を認識しません。詳細はCalculation of queues during maintenance を参照してください。

zabbix[rcache,<cache>,<mode>]

 Zabbix 設定キャッシュの利用可能状況の統計情報。
 戻り値: *Integer*（サイズの場合）; *Float*（割合の場合）。

パラメーター:

- **cache** - *buffer*;
- **mode**:

  - *total* - buffer の合計サイズ;
  - *free* - 空き buffer のサイズ;
  - *pfree* - 空き buffer の割合;
  - *used* - 使用中 buffer のサイズ;
  - *pused* - 使用中 buffer の割合。

zabbix[requiredperformance]

 Zabbix サーバーまたは Zabbix プロキシに必要なパフォーマンスを、想定される 1 秒あたりの新規値の数で返します。
 戻り値: *Float*。

コメント:

- *Reports* > *System information* の「Required server performance, new values per second」におおむね対応します。

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

 Zabbix サーバーまたはプロキシの内部メトリクスを返します。<ip> と <port> が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON object*。

パラメータ:

- **ip** - リモートで問い合わせるサーバー/プロキシの IP/DNS/ネットワークマスクのリスト（デフォルトは 127.0.0.1）;
- **port** - リモートで問い合わせるサーバー/プロキシのポート（デフォルトは 10051）。

コメント:

- stats リクエストは、対象インスタンスの StatsAllowedIP server/proxy パラメータに □□□□ されているアドレスからのみ受け付けられます。
- このアイテムは、内部メトリクスの選択されたセットを返します。詳細は、Zabbix stats のリモート監視 を参照してください。

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

 Zabbix サーバーまたはプロキシの内部キューメトリクスを返します（zabbix[queue,<from>,<to>] を参照）。
<ip> と <port> が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON object*。

パラメータ:

- **ip** - リモートで問い合わせるサーバー/プロキシの IP/DNS/ネットワークマスクのリスト（デフォルトは 127.0.0.1）;
- **port** - リモートで問い合わせるサーバー/プロキシのポート（デフォルトは 10051）;
- **from** - 少なくとも（デフォルトは 6 秒）遅延;
- **to** - 最大で（デフォルトは無限大）遅延。

コメント:

- stats リクエストは、対象インスタンスの StatsAllowedIP server/proxy パラメータに列挙されたアドレスからのみ受け付けられます。
- このアイテムによって、内部メトリクスの選択されたセットが返されます。
  詳細については、Zabbix stats のリモート監視 を参照してください。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細については、メンテナンス中のキューの計算 を参照してください。

zabbix[tcache,cache,<parameter>]

 Zabbix のトレンド機能キャッシュの有効性統計です。
 戻り値: *Integer*（サイズの場合）; *Float*（割合の場合）。

パラメーター:

- **parameter**:
  - *all* - キャッシュ要求の合計（デフォルト）;
  - *hits* - キャッシュヒット数;
  - *phits* - キャッシュヒット率;
  - *misses* - キャッシュミス数;
  - *pmisses* - キャッシュミス率;
  - *items* - キャッシュされたアイテム数;
  - *requests* - キャッシュされたリクエスト数;
  - *pitems* - キャッシュされたアイテム数が、キャッシュされたアイテム数 + リクエスト数に占める割合。低い割合は、キャッシュサイズを削減できる可能性が高いことを意味します。
  - *paccessed* - 過去 24 時間にアクセスされたエントリの割合。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[triggers]

 有効なホスト上のすべてのアイテムが有効な Zabbix データベース内の有効なトリガーの数。
 戻り値: 整数。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[uptime]

 Zabbix サーバーまたはプロキシプロセスの稼働時間 (秒単位)。
 戻り値: 整数。

zabbix[vcache,buffer,<mode>]

 Zabbix value cache の利用可能状況の統計情報。
 戻り値: *Integer* (サイズの場合); *Float* (割合の場合)。

パラメーター:

- **mode**:
  - *total* - バッファーの合計サイズ;
  - *free* - 空きバッファーのサイズ;
  - *pfree* - 空きバッファーの割合;
  - *used* - 使用済みバッファーのサイズ;
  - *pused* - 使用済みバッファーの割合。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[vcache,cache,<parameter>]

 Zabbix value cache の有効性統計です。
 戻り値: *Integer*。<parameter> に *mode* を指定した場合、戻り値は次のとおりです: *0* - 通常モード; *1* - 低メモリモード。

パラメーター:

- **parameter**:
  - *requests* - リクエストの総数;
  - *hits* - キャッシュヒット数（キャッシュから取得された履歴値）;
  - *misses* - キャッシュミス数（データベースから取得された履歴値）;
  - *mode* - value cache の動作モード。

コメント:

- 低メモリモードが一度有効になると、このモードを引き起こした障害がより早く解消されても、value cache は 24 時間この状態のまま維持されます。
- 値/秒の統計を取得するために、このキーを *Change per second* の前処理ステップと組み合わせて使用できます。
- このアイテムは Zabbix プロキシではサポートされていません。

zabbixversion

 Zabbix サーバーまたはプロキシのバージョン。
 戻り値: *String*。例: `8.0.0`。

zabbix[vmware,buffer,<mode>]

 Zabbix vmware cache の利用可能性統計です。
 戻り値: *Integer* (サイズの場合); *Float* (割合の場合)。

パラメーター:

- **mode**:
  - *total* - バッファーの合計サイズ;
  - *free* - 空きバッファーのサイズ;
  - *pfree* - 空きバッファーの割合;
  - *used* - 使用済みバッファーのサイズ;
  - *pused* - 使用済みバッファーの割合。

zabbix[vps,written]

 データベースに書き込まれた履歴値の合計数。
 戻り値: 整数。

zabbix[wcache,<cache>,<mode>]

 Zabbix 書き込みキャッシュの統計情報と可用性。
 戻り値: *Integer*（数値/サイズの場合）; *Float*（割合の場合）。

パラメーター:

- **cache** - *values*、*history*、*index*、または *trend*;
- **mode**:
  - *values* キャッシュパラメーターの場合:
    - *all* - Zabbix サーバー/プロキシによって処理された値の総数（サポートされていないアイテムを除く）（カウンター）（デフォルト）;
    - *float* - 処理された浮動小数値の数（カウンター）;
    - *uint* - 処理された符号なし整数値の数（カウンター）;
    - *str* - 処理された文字列値の数（カウンター）;
    - *log* - 処理されたログ値の数（カウンター）;
    - *text* - 処理されたテキスト値の数（カウンター）;
    - *bin* - 処理されたバイナリ値の数（カウンター）;
    - *json* - 処理された JSON 値の数（カウンター）;
    - *not supported* - アイテムの処理の結果、そのアイテムがサポートされない状態になった、またはその状態を維持した回数（カウンター）;
  - *history*、*index*、*trend* キャッシュパラメーターの場合:
    - *pfree* - 空きバッファーの割合（デフォルト）;
    - *total* - バッファーの合計サイズ;
    - *free* - 空きバッファーのサイズ;
    - *used* - 使用中バッファーのサイズ;
    - *pused* - 使用中バッファーの割合。

コメント:

- `trend` キャッシュパラメーターは Zabbix プロキシではサポートされていません。
- history キャッシュはアイテム値を保存するために使用されます。低い値は、データベース側のパフォーマンス問題を示します。
- history index キャッシュは、history キャッシュに保存された値を索引付けするために使用されます。
- history キャッシュがいっぱいになってからクリアされた後も、history index キャッシュには一部のデータが残ります。この動作は想定どおりであり、メモリーのサイズを常に変更するために必要な追加処理を避けることで、システムをより効率的に動作させるのに役立ちます。
- trend キャッシュは、データを受信するすべてのアイテムについて、現在の 1 時間分の集計値を保存します。
- 値/秒の統計情報を取得するには、前処理ステップ *Change per second* と組み合わせて `zabbix[wcache,values]` キーを使用できます。

Zabbix トラッパー

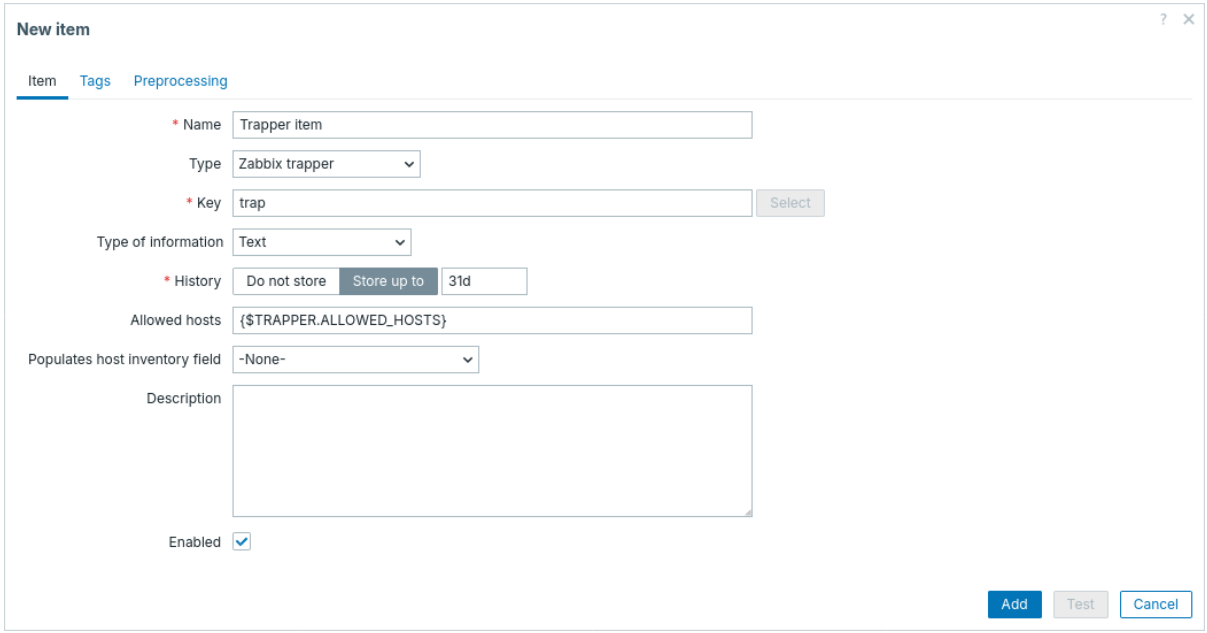
概要

トラッパーアイテムは、データを取得する代わりに受信データを受け付けます。これは、Zabbix に送信したいあらゆるデータに有用です。

設定

トラッパーアイテムを設定するには:

1. データ収集 > ホストに移動します。
2. ホストの行にある アイテムをクリックします。
3. アイテムの作成をクリックします。
4. 設定フォームでアイテムのパラメータを入力します。



必須入力フィールドには赤いアスタリスクが表示されます。

トラッパーアイテムで特定の情報の入力が必要なフィールドは次のとおりです:

Type	「Zabbix trapper」を選択します。
Key	Zabbix サーバーにデータを送信する際にアイテムを識別するために使用するキーを入力します。
Type of information	送信されるデータの形式に対応する情報の種類を選択します。
Allowed hosts	カンマ区切りの IP アドレス（必要に応じて CIDR 表記）または DNS 名の一覧を指定します。 デフォルト値はuser macro `{$TRAPPER.ALLOWED_HOSTS}` です。デフォルトでは、`{$TRAPPER.ALLOWED_HOSTS}` は `127.0.0.1,::1` です。アドレスをフィールドに直接入力することも、マクロ値を変更することもできます。空欄のままにすると、すべてのホストからの受信接続が拒否されます。指定した場合、受信接続はここに列挙されたホストからのみ受け入れられます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。 "IPv4 互換 IPv6 アドレス"（0000::/96 プレフィックス）はサポートされていますが、RFC4291 により非推奨です。 例: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} スペース、ユーザーマクロ、およびホストマクロ {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} がサポートされています。

Note:
値を送信する前に、アイテムを保存してから Zabbix サーバーが設定キャッシュの更新から変更を取得するまで、最大 60 秒待つ必要がある場合があります。

データの送信

Zabbix サーバーまたはプロキシへのデータ送信は、Zabbix senderユーティリティまたは Zabbix sender protocolを使用して行えます。Zabbix サーバーへのデータ送信は、`history.push` API メソッドを使用して行うこともできます。

Zabbix sender

Zabbix sender ユーティリティを使用して Zabbix サーバーまたはプロキシにデータを送信するには、次のコマンドを実行して"test value" を送信できます。

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

"test value" を送信するために、次のコマンドオプションが使用されます。

- `-z` は Zabbix サーバーの IP アドレスを指定します

- -p は Zabbix サーバーのポート番号を指定します（既定値は 10051）
- -s はホストを指定します（表示名のhost name ではなく、技術名を使用してください）
- -k は trapper アイテムで設定 されたアイテムのキーを指定します
- -o は送信する値を指定します

Attention:
Zabbix trapper プロセスは、対象ホストに対して対応するアイテムキーの存在を確認するために、アイテムキーで使用されているマクロを展開しません。

Zabbix sender と Zabbix サーバーまたはプロキシ間の通信の詳細については、*Zabbix sender protocol* を参照してください。

history.push

history.push API メソッドを使用して Zabbix サーバーにデータを送信するには、以下のようなテスト値を含む HTTP POST リクエストを作成できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

リクエストが正しい場合、API から返されるレスポンスは以下のようになります。

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

レスポンスデータ内のエラーは、特定のアイテムへのデータ送信が Zabbix サーバーによる検証に失敗したことを示します。これは以下の理由で発生する可能性があります。

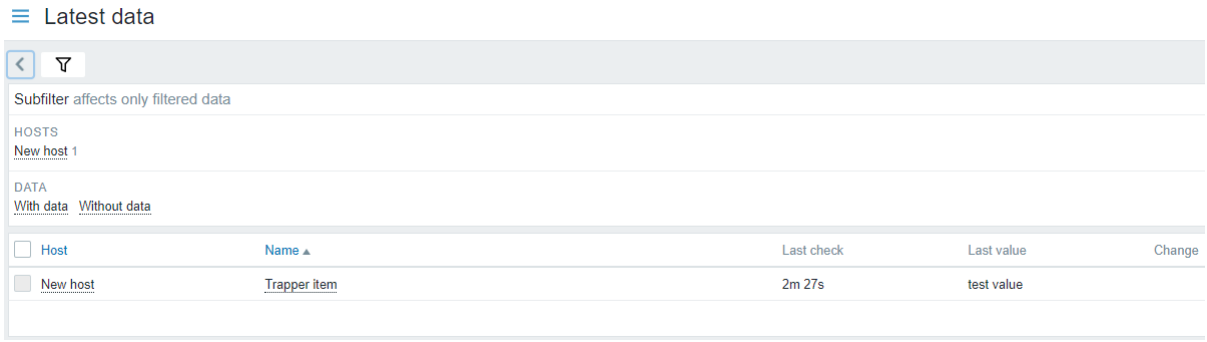
- データを送信するユーザーがアイテムのホストに対する読み取り権限を持っていない
- ホストが無効化されている、またはデータ収集なしのメンテナンス中である
- アイテムが存在しない、またはまだサーバーの設定キャッシュに含まれていない
- アイテムが無効化されている、またはタイプが Zabbix トラッパーまたはHTTP エージェント（トラッピング有効）以外である
- ユーザーの IP または DNS がアイテムの許可ホストリストに設定されていない
- 他のアイテムがナノ秒レベルでタイムスタンプが重複する値を持っている

エラーがない場合は、送信された値が処理のために受け入れられたことを示します。これには（存在する場合）前処理、トリガー処理、データベースへの保存が含まれます。受け入れられた値の処理も失敗する場合があることに注意してください（たとえば、前処理中など）。その場合、値は破棄されます。

Zabbix API の操作方法の詳細については、APIを参照してください。

データの表示

データが送信されると、監視 > 最新データ に移動して結果を確認できます:



Note:
単一の数値が送信された場合、データのグラフには、その値の時点を中心に左右へ水平線が表示されます。

外部チェック

概要

外部チェックは、Zabbix サーバーがシェルスクリプトの実行 またはバイナリを実行して行うチェックです。ただし、ホストが Zabbix プロキシによって監視されている場合、外部チェックはプロキシによって実行されます。

外部チェックでは、監視対象のホスト上でエージェントを実行する必要はありません。

アイテムキーの構文は次のとおりです。

```
script[<parameter1>,<parameter2>,...]
```

各項目の意味は次のとおりです。

ARGUMENT	DEFINITION
script	シェルスクリプトまたはバイナリの名前。
parameter(s)	省略可能なコマンドラインパラメーター。

スクリプトにパラメーターを渡したくない場合は、次を使用できます。

```
script[] or
script
```

Zabbix サーバーまたはプロキシは、外部スクリプト用に指定されたディレクトリを検索し、コマンドを実行します（Zabbix server/proxy 設定ファイルの ExternalScripts パラメーターを参照してください）。コマンドは Zabbix サーバー/プロキシと同じユーザー権限で実行されるため、必要に応じて、アクセス権限や環境変数はラッパースクリプトで処理してください。コマンドの権限も、そのユーザーが実行できるように設定されている必要があります。指定されたディレクトリ内のコマンドのみが実行可能です。

Warning:
外部チェックを使いすぎないでください。各スクリプトの実行ごとに Zabbix サーバー/プロキシは fork プロセスを開始する必要があり、多数のスクリプトを実行すると Zabbix のパフォーマンスが大幅に低下する可能性があります。

使用例

スクリプト check_oracle.sh を最初のパラメータ '-h' 付きで実行します。2 番目のパラメータは、ホストのプロパティでの選択に応じて、IP アドレスまたは DNS 名に置き換えられます。

```
check_oracle.sh["-h","{HOST.CONN}"]
```

ホストが IP アドレスを使用するように設定されている場合、Zabbix サーバー/プロキシは次を実行します。

```
check_oracle.sh '-h' '192.168.1.4'
```

外部チェックの結果

外部チェックの戻り値は、チェックによって生成された標準出力と標準エラーです。

Attention:
テキスト (文字列、ログ、テキスト型の情報) を返すアイテムは、標準エラー出力があっても未サポートにはなりません。

戻り値は 16MB(末尾の空白も含み、切り捨てられます) に制限されます。データベースの制限も適用されます。

要求されたスクリプトが見つからない場合や、Zabbix サーバー/プロキシに実行権限がない場合、アイテムは未サポートとなり、対応するエラーメッセージが表示されます。

タイムアウトの場合、アイテムは未サポートとなり、対応するエラーメッセージが表示され、スクリプト用に fork されたプロセスは終了します。

データベースモニター

概要

Zabbix の Web インターフェースにある *Database monitor* アイテムタイプは、ODBC 監視（ODBC チェック）に使用されます。

ODBC は、データベース管理システム（DBMS）にアクセスするための C プログラミング言語向けミドルウェア API です。ODBC の概念は Microsoft によって開発され、その後ほかのプラットフォームにも移植されました。

Zabbix は、ODBC でサポートされている任意のデータベースを問い合わせることができます。そのために、Zabbix はデータベースへ直接接続するのではなく、ODBC で設定された ODBC インターフェースとドライバーを使用します。これにより、複数の目的（たとえば、特定のデータベースキューの確認、使用統計の取得など）に対して、さまざまなデータベースをより効率的に監視できます。

Zabbix は、最も広く使用されているオープンソースの ODBC API 実装の 1 つである unixODBC をサポートしています。

> **Attention:**
> ODBC チェックについては、既知の問題も参照してください。

unixODBC のインストール

unixODBC をインストールする推奨方法は、Linux オペレーティングシステムのデフォルトのパッケージリポジトリを使用することです。
一般的な Linux ディストリビューションでは、unixODBC は既定でパッケージリポジトリに含まれています。
パッケージが利用できない場合は、unixODBC のホームページからソースファイルを入手できます: http://www.unixodbc.org/download.html。

unixODBC をインストールするには、使用するシステムに応じたパッケージマネージャーを使用します:

```
##### Ubuntu/Debian 系システム:
apt install unixodbc unixodbc-dev

##### RedHat/Fedora 系システム:
dnf install unixODBC unixODBC-devel

##### SUSE 系システム:
zypper in unixODBC-devel
```

> **Attention:**
> unixodbc-dev または unixODBC-devel パッケージは、unixODBC サポート付きで Zabbix をコンパイルするために必要です。
> ODBC サポートを有効にするには、Zabbix を次のconfiguration option 付きでコンパイルする必要があります:

> --with-unixodbc[=ARG] # unixODBC パッケージに対して ODBC ドライバーを使用します。

unixODBC ドライバーのインストール

監視対象のデータベースに対して unixODBC データベースドライバーをインストールする必要があります。サポートされているデータベースとドライバーの一覧については、unixODBC のホームページを参照してください: http://www.unixodbc.org/drivers.html。

> **Note:**
> 一部の Linux ディストリビューションでは、データベースドライバーがパッケージリポジトリに含まれています。

MySQL

MySQL unixODBC データベースドライバをインストールするには、お使いのシステムに応じたパッケージマネージャを使用してください:

```
##### Ubuntu/Debian システムの場合:
apt install odbc-mariadb
```

```
##### RedHat/Fedora ベースのシステムの場合:
dnf install mariadb-connector-odbc

##### SUSE ベースのシステムの場合:
zypper install mariadb-connector-odbc
```

パッケージマネージャを使用せずにデータベースドライバをインストールするには、mysql-connector-odbc についてはMySQL ドキュメント、mariadb-connector-odbc についてはMariaDB ドキュメントを参照してください。

PostgreSQL

PostgreSQL unixODBC データベースドライバーをインストールするには、お使いのシステムのパッケージマネージャーを使用します:

```
##### Ubuntu/Debian システムの場合:
apt install odbc-postgresql

##### RedHat/Fedora ベースのシステムの場合:
dnf install postgresql-odbc

##### SUSE ベースのシステムの場合:
zypper install psqlODBC
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールするには、PostgreSQL ドキュメントを参照してください。

Oracle

unixODBC データベースドライバーのインストールについては、Oracle ドキュメントを参照してください。

MSSQL

MSSQL unixODBC データベースドライバーをインストールするには、お使いのシステムに応じてパッケージマネージャーを使用してください:

```
##### Ubuntu/Debian システムの場合:
apt install tdsodbc

##### RedHat/Fedora ベースのシステムの場合 (EPEL パッケージ: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### SUSE ベースのシステムの場合:
zypper install libtdsodbc0
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールするには、FreeTDS ユーザーガイドを参照してください。

unixODBC の設定

unixODBC を設定するには、odbcinst.ini と odbc.ini ファイルを編集する必要があります。これらのファイルの場所は、次のコマンドを実行して確認できます。

```
odbcinst -j
```

コマンドの結果は、次のような情報が含まれているはずです。

```
unixODBC 2.3.9
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
```

odbcinst.ini

odbcinst.ini ファイルには、インストールされている ODBC データベースドライバーが一覧表示されます。odbcinst.ini が存在しない場合は、手動で作成する必要があります。

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

パラメータ	説明
TEST_MYSQL	データベースドライバー名
Description	データベースドライバーの説明
Driver	データベースドライバーライブラリの場所
FileUsage	データベースドライバーがローカルファイルへのアクセスをサポートせずにデータベースサーバーへの接続をサポートするか (0)、ファイルからのデータの読み取りをサポートするか (1)、ファイルへのデータの書き込みをサポートするか (2) を決定します。
Threading	スレッドの直列化レベル。PostgreSQL でサポートされています。 1.6 以降、ドライバーマネージャーがスレッドサポート付きでビルドされている場合は、別のドライバーエントリを追加できます。

odbc.ini

`odbc.ini` ファイルはデータソースの設定に使用されます。サポートされているパラメータの一覧はデータベースドライバによって異なることに注意してください（例えば、Oracle データベースでは Server の代わりに ServerName を使用する場合があります）。

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

パラメータ	説明
TEST_MYSQL	データソース名（DSN）。
Description	データソースの説明。
Driver	データベースドライバ名（`odbcinst.ini` で指定されたもの）。
Server	データベースサーバの IP/DNS。
User	接続用のデータベースユーザー。
Password	データベースユーザーのパスワード。
Port	データベース接続ポート。
Socket	データベース接続ソケット。
Database	データベース名。

その他の設定パラメータのオプションについては、MySQL ドキュメントを参照してください。

PostgreSQL 用の `odbc.ini` ファイルには、追加のパラメータを含めることができます:

```
[TEST_PSQL]
Description=PostgreSQL Test Database
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=8.0+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

パラメータ	説明
ReadOnly	データベース接続が読み取り操作（SELECT クエリ）のみを許可し、変更操作（INSERT、UPDATE、DELETE ステートメント）を制限するかどうかを指定します。データを変更せずに保持する必要があるシナリオで便利です。
Protocol	PostgreSQL バックエンドプロトコルバージョン（SSL 接続を使用している場合は無視されます）。
ShowOidColumn	SQLColumns に Object ID（OID）を含めるかどうかを指定します。
FakeOidIndex	OID に偽の一意インデックスを作成するかどうかを指定します。
RowVersioning	他のユーザーによってデータが変更された場合に、行を更新しようとしたときにアプリケーションがそれを検出できるようにするかどうかを指定します。このパラメータを有効にすると、WHERE 句で全てのカラムを指定する必要がなくなるため、更新処理が高速化される場合があります。
ShowSystemTables	データベースドライバがシステムテーブルを SQLTables で通常のテーブルとして扱うかどうかを指定します。システムテーブルへのアクセス性を高めるために有用です。
Fetch	ドライバが SELECT ステートメントを自動的に declare cursor/fetch で処理し、100 行のキャッシュを維持するかどうかを指定します。
BoolsAsChar	Boolean 型のマッピングを制御します。 「Yes」に設定すると、Bool は SQL_CHAR にマッピングされ、それ以外の場合は SQL_BIT にマッピングされます。
SSLmode	接続の SSL モードを指定します。
ConnSettings	接続時にバックエンドに送信される追加設定。

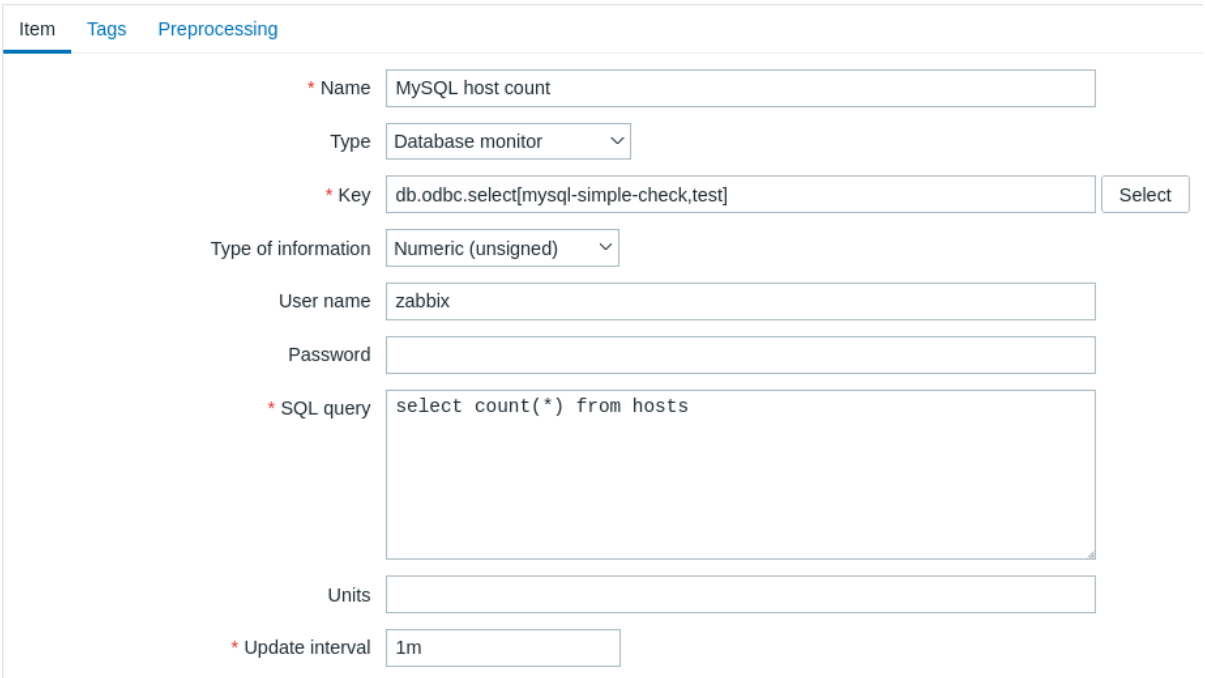
ODBC 接続のテスト

ODBC 接続が正常に動作しているかどうかをテストするには、isql ユーティリティ（unixODBC パッケージに含まれています）を使用できます。

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Zabbix Web インターフェースでのアイテム設定

Database monitoring アイテムを設定します。



必須の入力フィールドには赤いアスタリスクが付いています。

データベース監視アイテムでは、次の項目を指定する必要があります。

Type	ここで「Database monitor」を選択します。
Key	サポートされているアイテムキーのいずれかを入力します: **db.odbc.select[]** - このアイテムは 1 つの値を返します（SQL クエリ結果の最初の行の最初の列）; **db.odbc.get[]** - このアイテムは JSON 形式で複数の行/列を返します; **db.odbc.discovery[]** - このアイテムは low-level discovery データを返します。
User name	データベースのユーザー名を入力します（最大 255 文字）。 このパラメータは、データベースのユーザー名が `odbc.ini` ファイルで指定されている場合は省略できます。 接続文字列を使用し、*User name* フィールドが空でない場合は、`UID=<user>` として接続文字列に追加されます。
Password	データベースのユーザーパスワードを入力します（最大 255 文字）。 このパラメータは、パスワードが `odbc.ini` ファイルで指定されている場合は省略できます。 接続文字列を使用し、*Password* フィールドが空でない場合は、`PWD=<password>` として接続文字列に追加されます。 このフィールドでは特殊文字を使用できます。Oracle を除くすべてのデータベースでは、`%` を含むパスワードは波括弧 `{}` で囲む必要があります。 パスワードは、たとえば `UID=<username>;PWD=P?;)*word` のように、ユーザー名の後ろに接続文字列へ追加されます。 生成された文字列をテストするには、次のコマンドを実行できます: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	SQL クエリを入力します。 `db.odbc.select[]` では、クエリは 1 つの値のみを返す必要があることに注意してください。
Type of information	ここで、クエリによって返される情報の種類を選択します。 情報の種類を誤って選択すると、アイテムはサポート対象外になります。

重要な注意事項

- データベース監視アイテムは、サーバーまたはプロキシの設定で *odbc poller* プロセスが起動されていない場合、サポート対象外になります。ODBC poller を有効にするには、Zabbix サーバー 設定ファイルで `StartODBCPollers` パラメータを設定するか、プロキシによるチェックの場合は Zabbix プロキシ 設定ファイルで設定します。
- アイテム設定 フォームの *Timeout* パラメータ値は、ODBC のログインタイムアウトおよびクエリ実行タイムアウトとして使用されます。なお、インストールされている ODBC ドライバがこれらをサポートしていない場合、これらのタイムアウト設定は無視されることがあります。
- SQL コマンドは、`select` ステートメントを使用する任意のクエリと同様に、結果セットを返す必要があります。クエリ構文は、それを処理する RDBMS に依存します。ストアドプロシージャへの要求の構文は、`call` キーワードで始める必要があります。

アイテムキーの詳細

山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

db.odbc.select[<unique short description>,<dsn>,<connection string>]

 SQL クエリ結果の最初の行の最初の列、つまり 1 つの値を返します。
 戻り値: SQL クエリに依存します。

パラメータ:

- **unique short description** - アイテムを識別するための一意の短い説明 (トリガーなどで使用);
- **dsn** - データソース名 (`odbc.ini` で指定されたもの);
- **connection string** - 接続文字列 (ドライバー固有の引数を含めることができます)。

コメント:

- `dsn` と `connection string` は省略可能なパラメータですが、少なくともどちらか一方は必須です。両方が定義されている場合は、`dsn` は無視されます。
- クエリが複数の列を返す場合、最初の列のみが読み取られます。クエリが複数の行を返す場合、最初の行のみが読み取られます。

db.odbc.get[<unique short description>,<dsn>,<connection string>]

 SQL クエリの結果を JSON 配列に変換します。
 戻り値: *JSON* オブジェクト。

パラメーター:

- **unique short description** - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（`odbc.ini` で指定）;
- **connection string** - 接続文字列（ドライバー固有の引数を含めることができます)。

コメント:

- dsn と connection string はどちらも任意のパラメーターですが、少なくともどちらか一方は必須です。両方が定義されている場合、dsn は無視されます。
- JSON 形式で複数の行/列を返すことができます。このアイテムは、1 回のシステムコールですべてのデータを収集するマスターアイテムとして使用できます。また、依存アイテムでは JSONPath 前処理を使用して個々の値を抽出できます。詳しくは、低レベルディスカバリで使用される返却形式の例を参照してください。

例:

```
# Connection for MySQL ODBC driver 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<unique short description>,<dsn>,<connection string>]

 SQL クエリの結果を JSON 配列に変換し、ローレベルディスカバリで使用します。クエリ結果のカラム名は、ディスカバリされたフィールド値とペアになったローレベルディスカバリマクロ名に変換されます。これらのマクロは、アイテム、トリガーなどのプロトタイプ作成時に使用できます。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **unique short description** - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（odbc.ini で指定）;
- **connection string** - 接続文字列（ドライバ固有の引数を含めることができます）。

コメント:

- dsn と connection string はどちらも省略可能なパラメータですが、少なくともどちらか一方は必須です。両方が定義されている場合は、dsn は無視されます。

エラーメッセージ

ODBC エラーメッセージは、詳細な情報を提供するためにフィールドに構造化されています。例えば、エラーメッセージは次のようになります。

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "Cannot execute ODBC query" - Zabbix メッセージ
- "[SQL_ERROR]" - ODBC リターンコード
- "[42601]" - SQLState
- "[7]" - ネイティブエラーコード
- "[ERROR: syntax error at or near ";"; Error while executing the query]" - ネイティブエラーメッセージ

エラーメッセージの長さは 2048 バイトに制限されているため、メッセージが切り捨てられる場合があることに注意してください。ODBC 診断レコードが複数ある場合、Zabbix は長さ制限の範囲内でそれらを（| で区切って）連結しようとします。

HTTP エージェント

概要

このアイテムタイプでは、HTTP/HTTPS プロトコルを使用してデータをポーリングできます。Zabbix sender protocol を使用してデータを Zabbix サーバーまたはプロキシに送信するか、history.push API メソッドを使用して Zabbix サーバーにデータを送信することで、トラッピングも可能です。

HTTP アイテムのチェックは Zabbix サーバーによって実行されます。ただし、ホストが Zabbix プロキシで監視されている場合、HTTP アイテムのチェックはプロキシによって実行されます。

HTTP アイテムのチェックには、監視対象のホスト上で稼働しているエージェントは必要ありません。

HTTP エージェントは HTTP と HTTPS の両方をサポートします。Zabbix は必要に応じてリダイレクトを追跡します（以下の リダイレクトを追跡オプションを参照）。リダイレクトの最大数は 10 に固定されています（cURL オプション CURLOPT_MAXREDIRS を使用）。

> **Attention:**
> Zabbix サーバー/プロキシは、最初に cURL（libcurl）サポート付きで設定されている必要があります。

HTTP チェックは非同期で実行されます。1 つのリクエストの応答を受信してから他のチェックを開始する必要はありません。DNS 解決も同様に非同期です。

非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPoller により定義されます）。

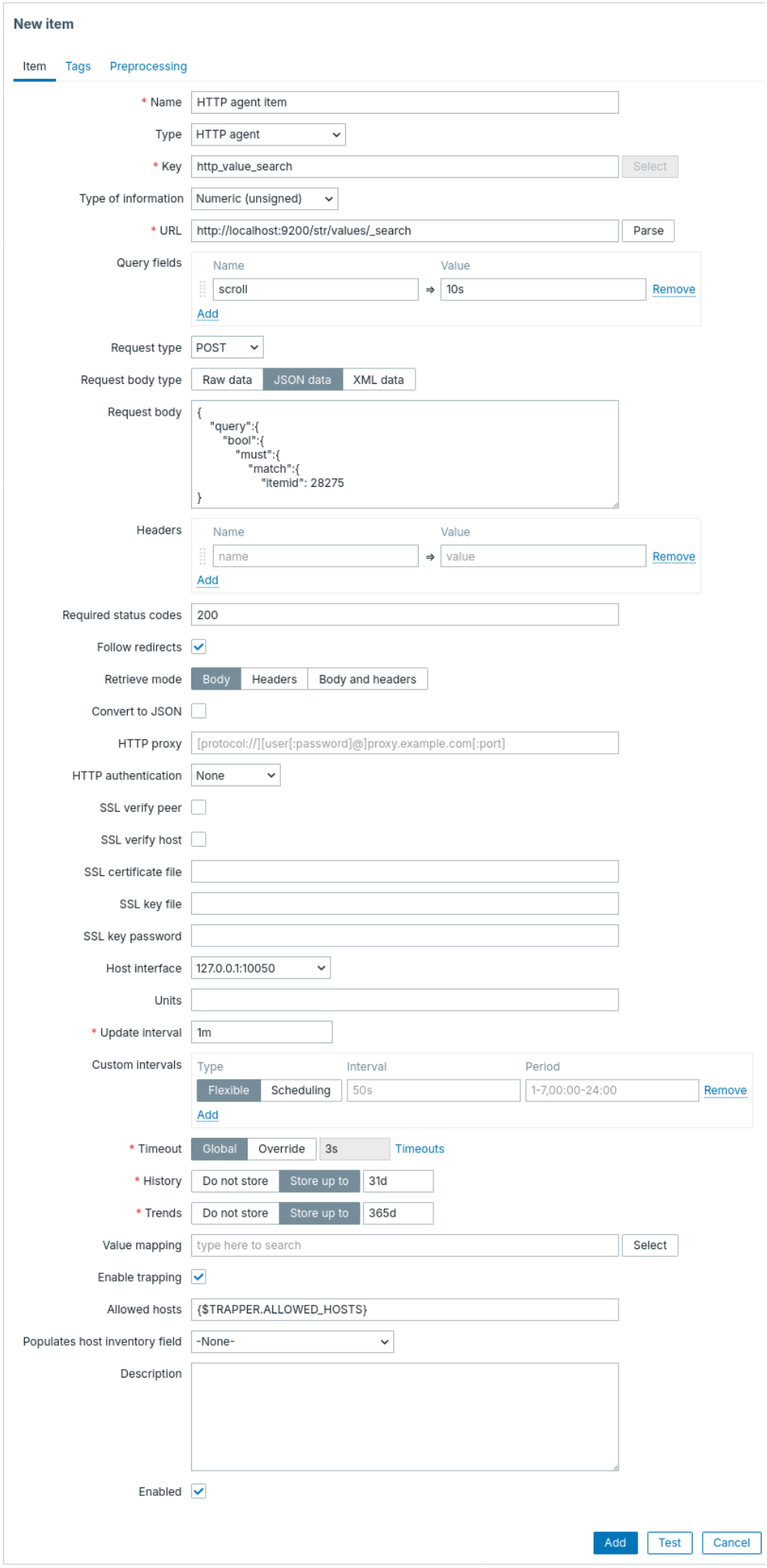
非同期 HTTP エージェント poller の数は、StartHTTPAgentPollers パラメータで定義されます。

永続接続の cURL 機能は、Zabbix 7.0 以降、HTTP エージェントのチェックに追加されました。

設定

HTTP アイテムを設定するには、次の手順を実行します。

1. データ収集 > ホストに移動します。
2. ホストの行にある アイテムをクリックします。
3. アイテムの作成をクリックします。
4. フォームにアイテムのパラメータを入力します。


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

必須入力フィールドには、赤いアスタリスクが付いています。

HTTP アイテムで固有の情報が必要なフィールドは次のとおりです。

パラメータ	説明
タイプ	ここで **HTTP** エージェントを選択します。
キー	一意のアイテムキーを入力します。

URL	接続してデータを取得する URL。例:
	`https://www.example.com`
	`http://www.example.com/download`
	ドメイン名には Unicode 文字を指定できます。HTTP チェックの実行時に、自動的に ASCII の Punycode へ変換されます。
	解析ボタンを使用すると、URL からオプションのクエリフィールド（?key=value&limit=10 など）を分離し、属性と値を クエリフィールドに移動して自動的に URL エンコードできます。
	セキュリティ上の理由から、URL に実際のユーザー名やパスワードを含めないでください。認証情報が公開されるのを避けるため、代わりにユーザーマクロを使用してください。
	2048 文字まで指定できます。
	サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。
	cURL オプション CURLOPT_URL を設定します。

クエリフィールド	URL の変数（上記を参照）。 属性と値のペアとして指定します。 値は自動的に URL エンコードされます。マクロの値は解決された後、自動的に URL エンコードされます。 サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLOPT_URL を設定します。
リクエストタイプ	リクエストメソッドのタイプとして、*GET*、*POST*、*PUT*、*HEAD* のいずれかを選択します。
リクエストボディタイプ	リクエストボディのタイプを選択します。 **Raw** データ - カスタム HTTP リクエストボディ。マクロは置換されますが、エンコードは実行されません。 **JSON** データ - JSON 形式の HTTP リクエストボディ。マクロは文字列、数値、true、false として使用できます。文字列として使用するマクロは二重引用符で囲む必要があります。マクロの値は解決された後、自動的にエスケープされます。ヘッダーに「Content-Type」が指定されていない場合、デフォルトで「Content-Type: application/json」になります。 **XML** データ - XML 形式の HTTP リクエストボディ。マクロはテキストノード、属性、CDATA セクションとして使用できます。テキストノードおよび属性内のマクロの値は、解決された後、自動的にエスケープされます。ヘッダーに「Content-Type」が指定されていない場合、デフォルトで「Content-Type: application/xml」になります。
リクエストボディ	リクエストボディを入力します。 サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。
ヘッダー	リクエストの実行時に送信されるカスタム HTTP ヘッダー。 属性と値のペアとして指定します。 サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLOPT_HTTPHEADER を設定します。
必須ステータスコード	想定される HTTP ステータスコードの一覧。Zabbix が一覧にないコードを取得した場合、アイテムはサポートされていない状態になります。空の場合、チェックは実行されません。 例: 200,201,210-299。 一覧でサポートされるマクロ: ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLINFO_RESPONSE_CODE を使用します。
リダイレクトに従う	HTTP リダイレクトに従うには、チェックボックスを選択します。 cURL オプション CURLOPT_FOLLOWLOCATION を設定します。
取得モード	取得するレスポンスの部分を選択します。 ボディ - ボディのみ ヘッダー - ヘッダーのみ ボディとヘッダー - ボディとヘッダー

パラメータ	説明
JSON に変換	取得モードが ボディに設定されていない場合、ヘッダーは「header」キーの下に属性と値のペアとして保存されます。 「Content-Type: application/json」が検出された場合、ボディはオブジェクトとして保存されます。それ以外の場合は文字列として保存されます。例: `{ "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> }`
HTTP プロキシ	[protocol://][username[:password]@]proxy.example.com[:port] の形式で、使用する HTTP プロキシを指定できます。 オプションの protocol:// プレフィックスを使用して、代替プロキシプロトコル（https、socks4、socks5 など）を指定できます（ドキュメントを参照。プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました）。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。誤ったプロトコルを指定すると接続に失敗し、アイテムはサポートされていない状態になります。 デフォルトでは、ポート 1080 が使用されます。 指定した場合、プロキシは http_proxy、HTTPS_PROXY などのプロキシ関連環境変数を上書きします。指定しない場合、プロキシはプロキシ関連環境変数を上書きしません。入力した値は「そのまま」渡され、妥当性チェックは行われません。 HTTP プロキシでサポートされる認証は、単純認証のみです。 サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLOPT_PROXY を設定します。
HTTP 認証	認証オプションを選択します。 なし - 認証を使用しません。 **Basic** - Basic 認証を使用します。 **NTLM** - NTLM（Windows NT LAN Manager）認証を使用します。 **Kerberos** - Kerberos 認証を使用します（Zabbix での Kerberos の設定も参照）。 **Digest** - Digest 認証を使用します。 cURL オプション CURLOPT_HTTPAUTH を設定します。
ユーザー名	ユーザー名を入力します（最大 255 文字）。 *HTTP* 認証が *Basic*、*NTLM*、*Kerberos*、または *Digest* に設定されている場合に使用できます。ユーザーマクロとローレベルディスカバリマクロがサポートされます。
パスワード	ユーザーパスワードを入力します（最大 255 文字）。 *HTTP* 認証が *Basic*、*NTLM*、*Kerberos*、または *Digest* に設定されている場合に使用できます。ユーザーマクロとローレベルディスカバリマクロがサポートされます。
SSL ピアの検証	Web サーバーの SSL 証明書を検証するには、チェックボックスを選択します。サーバー証明書は、システム全体の CA ロケーションにある信頼された認証局（CA）証明書に対して検証されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータ SSLCALocation を使用して上書きできます。 cURL オプション CURLOPT_SSL_VERIFYPEER を設定します。
SSL ホストの検証	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスを選択します。 cURL オプション CURLOPT_SSL_VERIFYHOST を設定します。
SSL 証明書ファイル	クライアント認証に使用する SSL 証明書ファイルの名前。証明書ファイルは PEM[1] 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は、*SSL* 鍵ファイルフィールドを空のままにします。鍵が暗号化されている場合は、*SSL* 鍵パスワードフィールドにパスワードを指定します。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ SSLCertLocation で指定します。 サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLOPT_SSLCERT を設定します。

パラメータ	説明
SSL 鍵ファイル	クライアント認証に使用する SSL 秘密鍵ファイルの名前。秘密鍵ファイルは PEM[1] 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ SSLKeyLocation で指定します。 サポートされるマクロ: {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLOPT_SSLKEY を設定します。
SSL 鍵パスワード	SSL 秘密鍵ファイルのパスワード。 サポートされるマクロ: ユーザーマクロ、ローレベルディスカバリマクロ。 cURL オプション CURLOPT_KEYPASSWD を設定します。
タイムアウト	Zabbix は、設定された時間を超えて URL の処理に時間を費やしません（1～600 秒）。実際には、このパラメータは URL への接続確立および HTTP リクエスト実行の最大時間を定義します。そのため、Zabbix が 1 回のチェックに費やす時間は タイムアウトの 2 倍を超えません。 cURL オプション CURLOPT_TIMEOUT を設定します。 タイムアウトパラメータの詳細については、一般的なアイテム属性を参照してください。
トラッピングを有効にする	このチェックボックスを選択すると、アイテムはトラッパーアイテムとしても機能し、Zabbix senderユーティリティまたは Zabbix sender プロトコルを使用して Zabbix サーバーまたはプロキシに送信されたデータを受け入れます。また、history.push API メソッドを使用して Zabbix サーバーに送信されたデータも受け入れます。データの送信については、トラッパーアイテムを参照してください。
許可されたホスト	トラッピングを有効にするチェックボックスが選択されている場合にのみ表示されます。 カンマ区切りの IP アドレス（CIDR 表記も使用可能）または DNS 名の一覧。 デフォルト値はユーザーマクロ {$TRAPPER.ALLOWED_HOSTS} です。デフォルトでは、{$TRAPPER.ALLOWED_HOSTS} は 127.0.0.1,::1 です。フィールドにアドレスを直接入力するか、マクロの値を変更できます。空のままにすると、すべてのホストからの着信接続が拒否されます。指定した場合、ここに一覧表示されたホストからの着信接続のみが受け入れられます。 IPv6 サポートが有効な場合、127.0.0.1、::127.0.0.1、::ffff:127.0.0.1 は同じものとして扱われ、::/0 は任意の IPv4 または IPv6 アドレスを許可します。 0.0.0.0/0 を使用すると、任意の IPv4 アドレスを許可できます。 「IPv4 互換 IPv6 アドレス」（0000::/96 プレフィックス）もサポートされますが、RFC4291では非推奨です。 例: 127.0.0.1、192.168.1.0/24、192.168.3.1-255、192.168.1-10.1-255、::1、2001:db8::/32、mysqlserver1、zabbix.example.com、{HOST.HOST} このフィールドではスペースとユーザーマクロを使用できます。 ホストマクロ {HOST.HOST}、{HOST.NAME}、{HOST.IP}、{HOST.DNS}、{HOST.CONN} を使用できます。

Note:
HTTP プロキシフィールドを空のままにした場合、HTTP プロキシを使用する別の方法として、プロキシ関連の環境変数を設定できます。
HTTP の場合は、Zabbix サーバーユーザーの http_proxy 環境変数を設定します。例:
 http_proxy=http://proxy_ip:proxy_port
HTTPS の場合は、HTTPS_PROXY 環境変数を設定します。例:
 HTTPS_PROXY=http://proxy_ip:proxy_port 詳細については、シェルコマンド *# man curl* を実行してください。

Attention:
[1] Zabbix は、PEM 形式の証明書ファイルと秘密鍵ファイルのみをサポートします。証明書と秘密鍵のデータが PKCS #12 形式のファイル（通常は拡張子 *.p12 または *.pfx）に含まれている場合は、次のコマンドを使用して PEM ファイルを生成できます。

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

例

例 1

Elasticsearch などのサービスからデータを取得するには、単純な GET リクエストを送信します。

- URL に localhost:9200/?pretty を指定した GET アイテムを作成します
- 応答を確認します:

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "kH4CYqh5QfqgeTsjh2F9zg",
  "version" : {
    "number" : "6.1.3",
    "build_hash" : "af51318",
    "build_date" : "2018-01-26T18:22:55.523Z",
    "build_snapshot" : false,
    "lucene_version" : "7.1.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You know, for search"
}
```

- 次に、JSONPath の前処理ステップ `$.version.number` を使用してバージョン番号を抽出します

例 2

Elasticsearch などのサービスからデータを取得するために、シンプルな POST リクエストを送信します。

- URL が `http://localhost:9200/str/values/_search?scroll=10s` の POST アイテムを作成します
- プロセッサ負荷（1 コアあたりの 1 分平均）を取得するために、次の POST 本文を設定します

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- 受信結果:

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
    "_shards": {
        "total": 5,
        "successful": 5,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": 1,
        "max_score": 1.0,
        "hits": [{
            "_index": "dbl",
            "_type": "values",
            "_id": "dqX9VWEBV6sEKSMyk6sw",
            "_score": 1.0,
            "_source": {
```

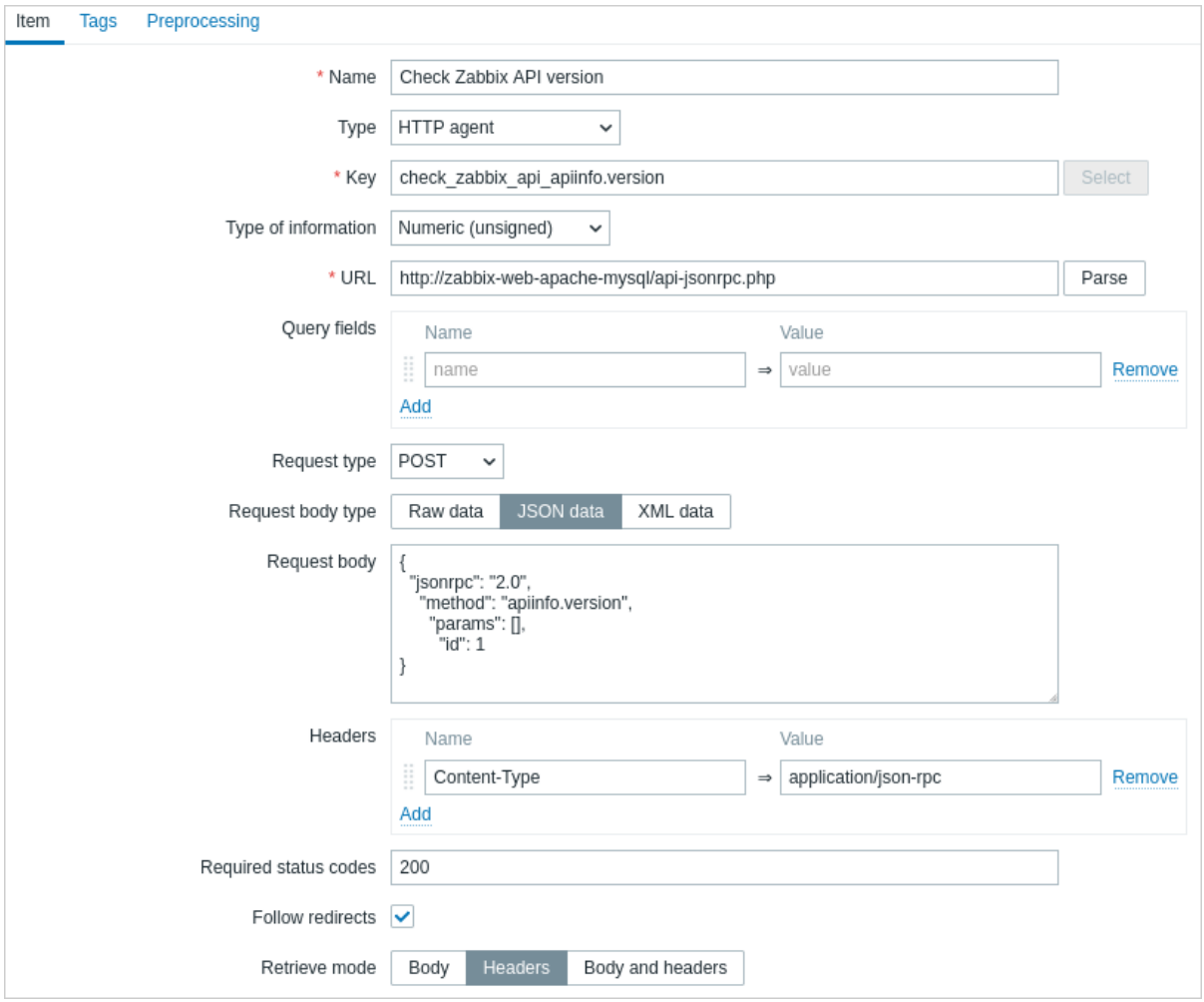
```
                    "itemid": 28275,
                    "value": "0.138750",
                    "clock": 1517566136,
                    "ns": 25388713,
                    "ttl": 604800
                }
            }]
        }
    }
```

- 次に、JSONPath の前処理ステップを使用してアイテムの値を取得します: `$.hits.hits[0]._source.value`

例 3

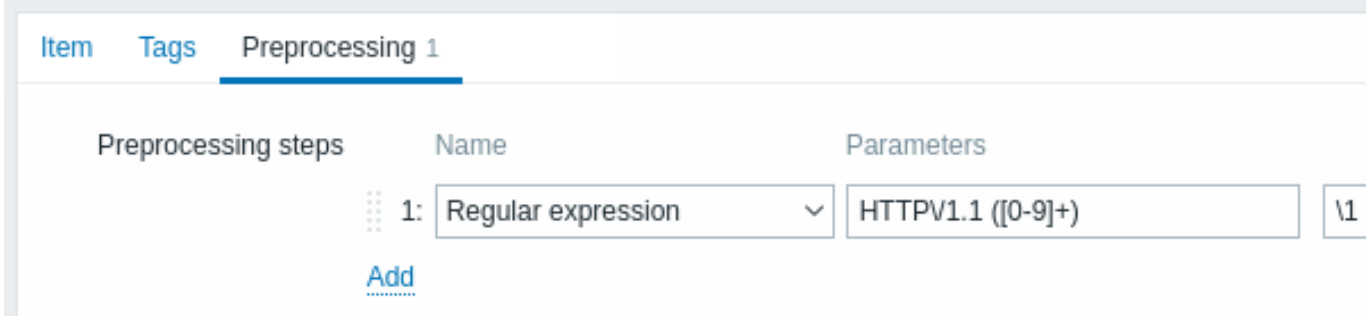
apiinfo.version を使用して、Zabbix API が稼働しているかを確認します。

- アイテムの設定:

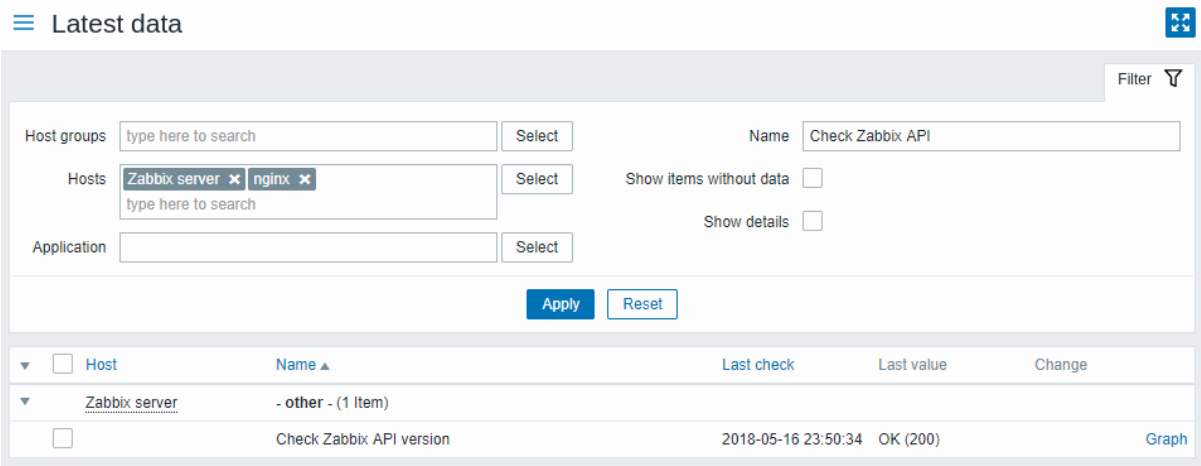


JSON データを使用した POST メソッドを使い、リクエストヘッダーを設定して、ヘッダーのみを返すように指定している点に注意してください。

- HTTP コードを取得するための正規表現によるアイテム値の前処理:



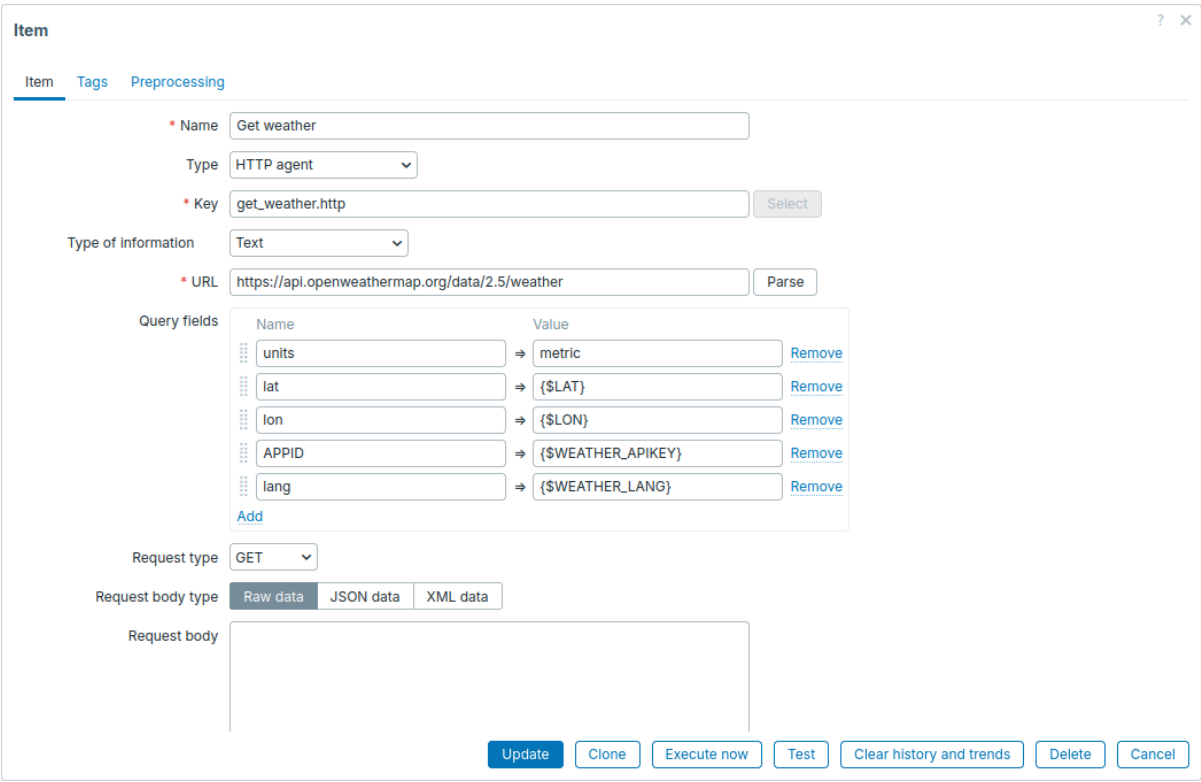
- 最新データで結果を確認します:



例 4

Openweathermap の公開サービスに接続して、気象情報を取得します。

- 単一の JSON で一括データ収集を行うためのマスターアイテムを設定します:



クエリフィールドでのマクロの使用に注意してください。
入力方法については Openweathermap API を参照してください。

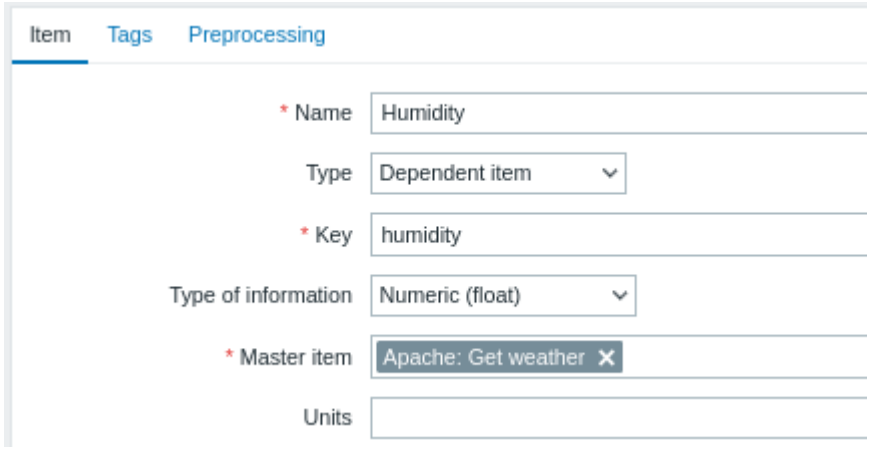
HTTP エージェントに対する応答として返されるサンプル JSON:

```
{
    "body": {
        "coord": {
            "lon": 40.01,
            "lat": 56.11
        },
        "weather": [{
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02n"
        }],
        "base": "stations",
```

```
        "main": {
            "temp": 15.14,
            "pressure": 1012.6,
            "humidity": 66,
            "temp_min": 15.14,
            "temp_max": 15.14,
            "sea_level": 1030.91,
            "grnd_level": 1012.6
        },
        "wind": {
            "speed": 1.86,
            "deg": 246.001
        },
        "clouds": {
            "all": 20
        },
        "dt": 1526509427,
        "sys": {
            "message": 0.0035,
            "country": "RU",
            "sunrise": 1526432608,
            "sunset": 1526491828
        },
        "id": 487837,
        "name": "Stavrovo",
        "cod": 200
    }
}
```

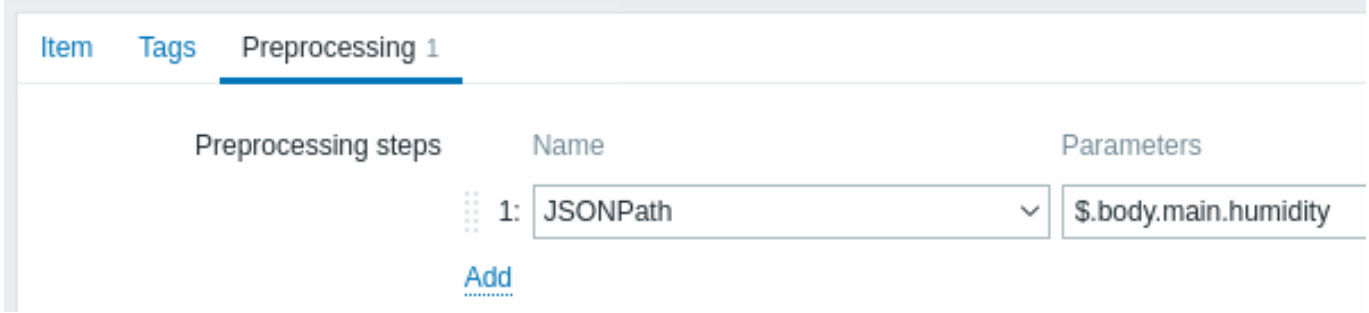

次のタスクは、JSON からデータを抽出する依存アイテムを設定することです。

- 湿度用のサンプル依存アイテムを設定します:

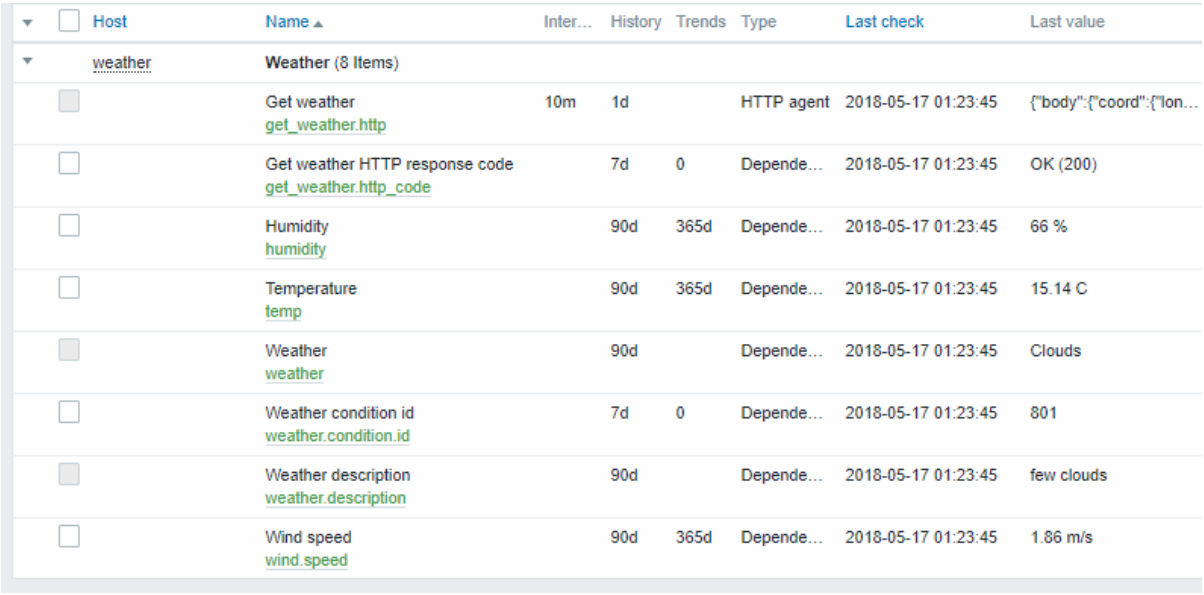


「Temperature」などの他の気象メトリクスも同様の方法で追加します。

- JSONPath を使用したサンプル依存アイテムの値の前処理:



- *Latest data* で気象データの結果を確認します:

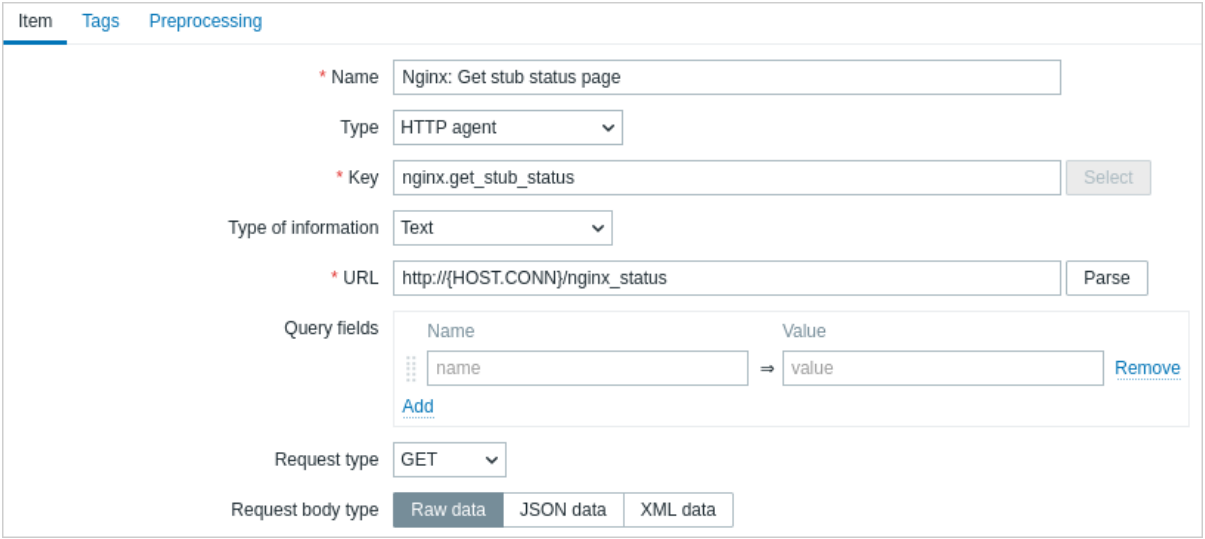


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

例 5

NGINX ステータスページに接続し、メトリクスを一括で取得します。

- 公式ガイドに従って NGINX を設定します。
- 一括データ収集用のマスターアイテムを設定します。

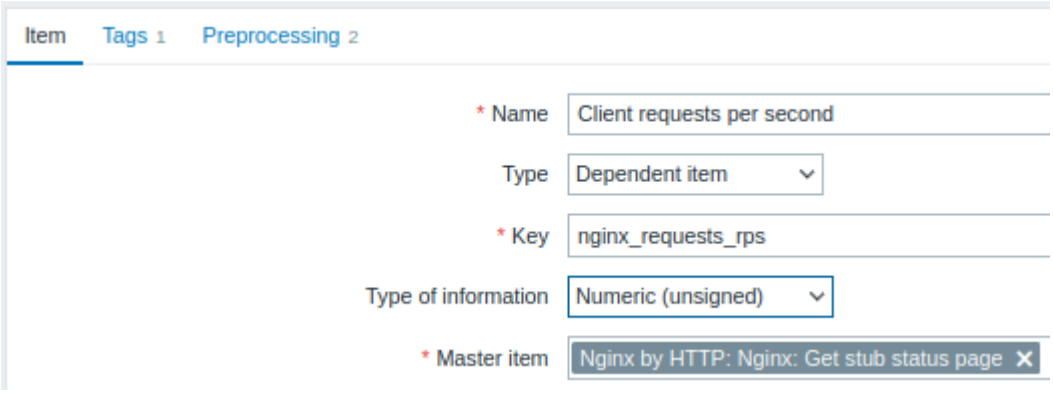


NGINX stub status の出力例:

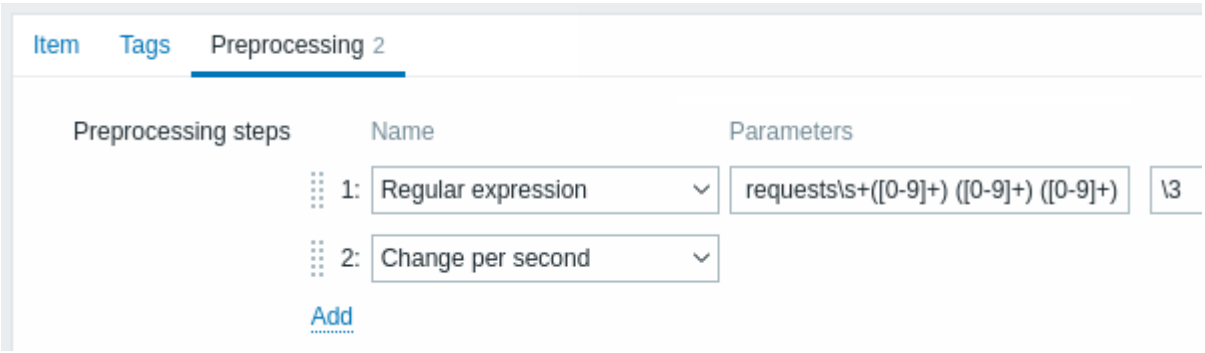
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

次に、データを抽出する依存アイテムを設定します。

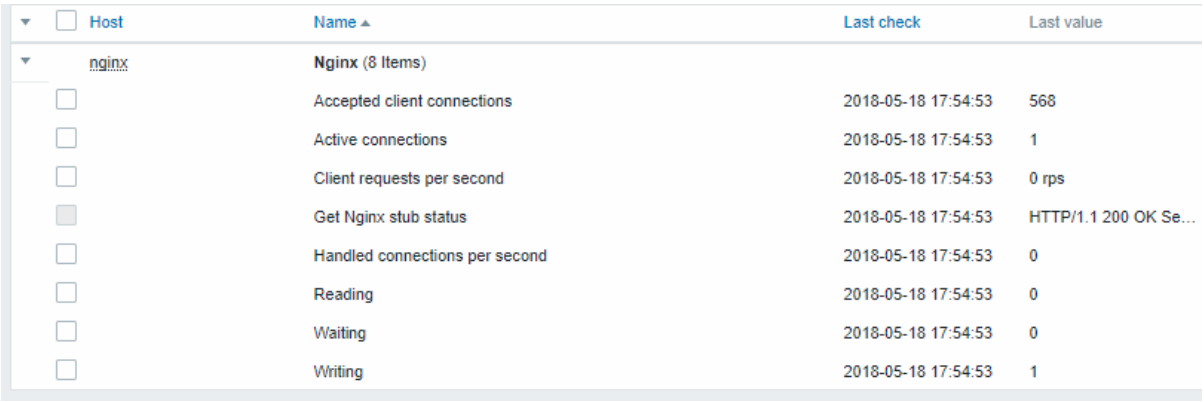
- 1 秒あたりのリクエスト数を取得する依存アイテムの例を設定します。



- 正規表現 `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)` を使用した、依存アイテムの値の前処理例:



- 最新データで stub モジュールから取得した完全な結果を確認します。



Prometheus チェック

概要

Zabbix は、Prometheus のライン形式で公開されるメトリクスを取得できます。

Prometheus データの収集を開始するには、次の 2 つの手順が必要です。

- 適切なデータエンドポイントを指すHTTP マスターアイテム。例: `https://<prometheus host>/metrics`
- マスターアイテムで収集したメトリクスから必要なデータを取得するために、Prometheus の前処理オプションを使用する依存アイテム

Prometheus のデータ前処理オプションには 2 種類あります。

- *Prometheus pattern* - 通常のアイテムで Prometheus データを取得するために使用します
- *Prometheus to JSON* - 通常のアイテムおよびローレベルディスカバリで使用します。この場合、取得した Prometheus データは JSON 形式で返されます。

一括処理

一括処理は、依存アイテムでサポートされています。キャッシュとインデックス作成を有効にするには、*Prometheus pattern* の前処理を 最初の前処理ステップにする必要があります。*Prometheus pattern* が最初の前処理ステップである場合、解析された Prometheus データは、*Prometheus pattern* 前処理ステップ内の最初の `<label>==<value>` 条件によってキャッシュされ、インデックスが作

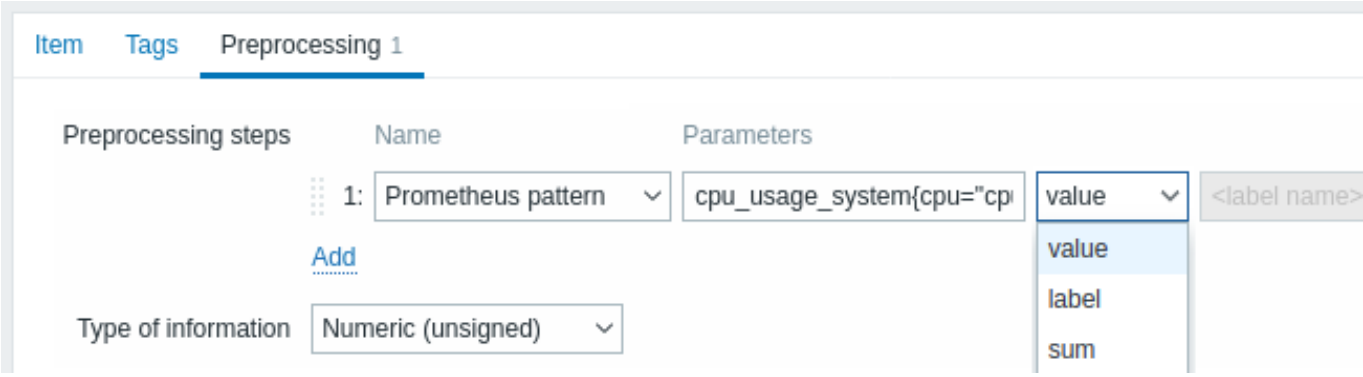
成されます。このキャッシュは、このバッチ内の他の依存アイテムを処理する際に再利用されます。最適なパフォーマンスを得るには、最初のラベルは値の種類が最も多いものにする必要があります。

最初のステップの前に他の前処理を行う必要がある場合は、それをマスターアイテムに移動するか、依存アイテムのマスターアイテムとして使用される新しい依存アイテムに移動してください。

設定

HTTP マスターアイテムが設定されている場合、Prometheus の前処理ステップを使用する依存アイテムを作成する必要があります。

1. 設定フォームで依存アイテムの一般的なパラメータを入力します。
2. 前処理タブに移動します。
3. Prometheus の前処理オプション（*Prometheus* パターンまたは *Prometheus to JSON*）を選択します。



以下のパラメータは、*Prometheus* パターンの前処理オプションに固有のものです。

パラメータ	説明	例
パターン	必要なデータパターンを定義するには、Prometheus クエリ言語に類似したクエリ言語（比較表を参照）を使用できます。例: <metric name> - メトリック名で選択 {__name__="<metric name>"} - メトリック名で選択 {__name__=~"<regex>"} - 正規表現に一致するメトリック名で選択 {<label name>="<label value>",...} - ラベル名で選択 {<label name>=~"<regex>",...} - 正規表現に一致するラベル名で選択 {__name__=~".*"}==<value> - メトリック値で選択 または、上記の組み合わせ: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> ラベル値には UTF-8 文字の任意の並びを使用できます。ただし、バックスラッシュ、二重引用符、改行文字は、それぞれ\\、\"、\n としてエスケープする必要があります。それ以外の文字はエスケープしないでください。	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu="cpu-total"}* *cpu_usage_system{cpu=".*"}* *cpu_usage_system{cpu="cpu-total",host="* *wmi_service_state{name="dhcp"}==1* *wmi_os_timezone{timezone=~".*"}==1*

パラメータ	説明	例
結果処理	値、ラベルを返すか、適切な関数を適用するかを指定します（パターンが複数の行に一致し、結果を集計する必要がある場合）。 **value** - メトリック値を返します（複数の行に一致した場合はエラー） **label** - ラベルフィールドで指定したラベルの値を返します（複数のメトリックに一致した場合はエラー） **sum** - 値の合計を返します **min** - 最小値を返します **max** - 最大値を返します **avg** - 平均値を返します **count** - 値の件数を返します このフィールドは、*Prometheus* パターンオプションでのみ使用できます。	以下のパラメータの使用例も参照してください。
出力	ラベル名を定義します（任意）。この場合、ラベル名に対応する値が返されます。 このフィールドは、結果処理フィールドで「ラベル」が選択されている場合の、*Prometheus* パターンオプションでのみ使用できます。	

パラメータの使用例

1. 最も一般的な使用例は、値を返すことです。次のデータから/var/db の値を返すには:

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096`

次のパラメータを使用します。

- パターン - `node_disk_usage_bytes{path="/var/db"}`
- 結果処理 - 「value」を選択

2. すべての `node_disk_usage_bytes` パラメータの平均値を取得することもできます。

- パターン - `node_disk_usage_bytes`
- 結果処理 - 「avg」を選択

3. Prometheus は数値データのみをサポートしますが、関連するテキスト形式の説明も返すために、回避策が一般的に使用されます。これは、フィルターを使用してラベルを指定することで実現できます。したがって、次のデータから「color」ラベルの値を返すには:

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

次のパラメータを使用します。

- パターン - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- 結果処理 - 「label」を選択
- ラベル - 「color」を指定

数値「1」に基づくフィルターは該当する行に一致し、ラベルはヘルスステータスの説明（現在は「green」ですが、「red」または「yellow」の可能性もあります）を返します。

Prometheus から JSON へ

Prometheus のデータは、低レベルディスカバリに使用できます。この場合、JSON 形式のデータが必要であり、*Prometheus to JSON* 前処理オプションを使用すると、まさにその形式が返されます。

詳細については、Prometheus データを使用したディスカバリを参照してください。

クエリ言語の比較

次の表は、PromQL と Zabbix Prometheus 前処理クエリ言語の違いと共通点を示しています。

PromQL instant vector selector	Zabbix Prometheus preprocessing
相違点	

PromQL instant vector selector		Zabbix Prometheus preprocessing
クエリ対象	Prometheus server	Prometheus exposition format のプレーンテキスト
戻り値	Instant vector	メトリックまたはラベル値（Prometheus パターン） JSON では単一値のメトリック配列（Prometheus to JSON）
ラベル一致演算子	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
ラベル名またはメトリック名の一致に使用される正規表現	RE2	PCRE
比較演算子	一覧を参照	値のフィルタリングでは **==**（等しい）のみサポートされます
共通点		
文字列と等しいメトリック名による選択	<metric name> または {__name__="<metric name>"}	<metric name> または {__name__="<metric name>"}
正規表現に一致するメトリック名による選択	{__name__=~"<regex>"}	{__name__=~"<regex>"}
<label name> の値が文字列と等しいことによる選択	{<label name>="<label value>",...}	{<label name>="<label value>",...}
<label name> の値が正規表現に一致することによる選択	{<label name>=~"<regex>",...}	{<label name>=~"<regex>",...}
値が文字列と等しいことによる選択	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

IPMI エージェント

概要

Zabbix では、Intelligent Platform Management Interface（IPMI）デバイスの状態と可用性を監視できます。IPMI チェックを実行するには、Zabbix サーバーをあらかじめ IPMI サポート付きで設定する必要があります。

IPMI は、コンピューターシステムをリモートから「ライトアウト」または「帯域外」で管理するための標準化されたインターフェースです。オペレーティングシステムやマシンの電源が入っているかどうかに関係なく、いわゆる「帯域外」管理カードからハードウェアの状態を直接監視できます。

Zabbix の IPMI 監視は、IPMI をサポートするデバイス（HP iLO、DELL DRAC、IBM RSA、Sun SSP など）でのみ機能します。

IPMI マネージャープロセスは、IPMI ポーラーによる IPMI チェックをスケジュールします。一度に 1 つの IPMI ポーラーだけがホストをポーリングするため、BMC コントローラーへのオープン接続数を減らすことができます。そのため、BMC コントローラーの過負荷を心配せずに、IPMI ポーラーの数を増やしても安全です。IPMI マネージャープロセスは、少なくとも 1 つの IPMI ポーラーが起動すると自動的に開始されます。

IPMI チェックについては、既知の問題も参照してください。

設定

ホストの設定

IPMI チェックを処理するために、ホストを設定する必要があります。
 IPMI インターフェイスをを追加し、それぞれの IP アドレスおよびポート番号を指定し、IPMI 認証パラメーターを定義する必要があります。

詳細は、configuration of hosts を参照してください。

サーバー設定

デフォルトでは、Zabbix サーバーは IPMI ポーラーを起動するように設定されていないため、追加した IPMI アイテムは動作しません。これを変更するには、Zabbix サーバーの設定ファイル (`zabbix_server.conf`) を root で開き、次の行を探します。

```
# StartIPMIPollers=0
```

このコメントを外し、ポーラー数をたとえば 3 に設定して、次のようにします。

```
StartIPMIPollers=3
```

ファイルを保存し、その後 zabbix_server を再起動します。

item の設定

ホストレベルでconfiguring an itemを行う場合:

- *Type* として'IPMI エージェント' を選択します。

- ホスト内で一意である item key (例えば、ipmi.fan.rpm) を入力します。
- *Host interface* には、関連する IPMI インターフェース (IP アドレスおよびポート番号) を選択します。IPMI インターフェースはホスト上に存在する必要があることに注意してください。
- IPMI センサー * (Dell Poweredge の'FAN MOD 1A RPM' など) からメトリックを取得するように指定します。デフォルトでは、センサー ID を指定する必要があります。接頭辞を使用することも可能です。
  - `id:` - センサーの ID を指定します。
  - `name:` - センサーのフルネームを指定します。これは、フルネームを指定することによってのみセンサを区別できる場合に役立ちます。
- それぞれの情報の種類を選択します (この場合は、'Numeric (float)' ディスクリートセンサーの場合は'Numeric (unsigned)')、単位（多くの場合'rpm')、およびその他の必要な項目属性を選択します。

サポートされているチェック

IPMI エージェントは、IPMI センサー関連の情報を返す組み込みアイテム **ipmi.get** をサポートしており、IPMI センサーのディスカバリ に使用できます。
 戻り値: *JSON object*

タイムアウトとセッションの終了

IPMI メッセージのタイムアウトとリトライ回数は、OpenIPMI ライブラリで定義されています。OpenIPMI の現在の設計上、これらの値を Zabbix で設定可能にすることはできません。インターフェースレベルでもアイテムレベルでも同様です。

LAN 用の IPMI セッション非アクティブタイムアウトは 60±3 秒です。現在のところ、OpenIPMI で Activate Session コマンドの定期的な送信を実装することはできません。Zabbix から特定の BMC への IPMI アイテムチェックが、BMC で設定されたセッションタイムアウトよりも長い間行われない場合、タイムアウト後の次の IPMI チェックは、個々のメッセージタイムアウト、リトライ、または受信エラーによりタイムアウトします。その後、新しいセッションが開かれ、BMC のフルリスキャンが開始されます。新しいセッションを管理するために新しい UDP ポートが開かれる場合があります。非アクティブとは、送信リクエストと受信レスポンスの両方がない状態を指します。不要な BMC のリスキャンを避けたい場合は、BMC で設定された IPMI セッション非アクティブタイムアウトよりも短い IPMI アイテムのポーリング間隔を設定することを推奨します。

IPMI 離散センサーに関する注意事項

ホスト上のセンサーを見つけるには、**DebugLevel=4** を有効にして Zabbix サーバーを起動します。数分待って、Zabbix サーバーログファイルでセンサー検出の記録を探します。

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

IPMI センサーのタイプや状態をデコードするには、IPMI 2.0 仕様書のコピーが利用できます（IPMI 仕様書の今後の更新は予定されていませんのでご注意ください)。

最初に確認するパラメータは"reading_type" です。仕様書の"Table 42-1, Event/Reading Type Code Ranges" を使用して"reading_type" コードをデコードします。例のほとんどのセンサーは"reading_type:0x1" で、これは"threshold" センサーを意味します。"Table 42-3, Sensor Type Codes" では"type:0x1" は温度センサー、"type:0x2" は電圧センサー、"type:0x4" はファンなどを示します。しきい値センサーは、温度、電圧、回転数などの連続パラメータを測定するため、" アナログ" センサーと呼ばれることもあります。

別の例として、"reading_type:0x3" のセンサーがあります。"Table 42-1, Event/Reading Type Code Ranges" では、リーディングタイプコード 02h-0Ch は"Generic Discrete" センサーを意味すると記載されています。離散センサーは最大 15 の状態（言い換えれば最大 15 の意味のあるビット）を持つことができます。たとえば、"type:0x7" の'CATERR' センサーの場合、"Table 42-3, Sensor Type

Codes" ではこのタイプは"Processor" を意味し、個々のビットの意味は次のとおりです：00h（最下位ビット）- IERR、01h - Thermal Trip など。

例の中には"reading_type:0x6f" のセンサーもいくつかあります。これらのセンサーについては、"Table 42-1, Event/Reading Type Code Ranges" で"Table 42-3, Sensor Type Codes" を使用してビットの意味をデコードするようにアドバイスされています。たとえば、'Power Unit Stat' センサーは"type:0x9" で、これは"Power Unit" を意味します。オフセット 00h は"PowerOff/Power Down" を意味します。言い換えれば、最下位ビットが 1 の場合、サーバーの電源がオフになっています。このビットをテストするには、マスク'1' を使ってbitand 関数を使用できます。トリガー式は次のようになります。

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

これにより、サーバーの電源オフを警告できます。

OpenIPMI-2.0.16, 2.0.17, 2.0.18, 2.0.19 のディスクリートセンサーの名前に関する注意事項

OpenIPMI-2.0.16, 2.0.17, 2.0.18 のディスクリートセンサーの名前には"0" (または他の数字や文字) を
 末尾に付加していることがよくあります。
 例えば、ipmitool と OpenIPMI-2.0.19 では、センサーの名前は"PhysicalSecurity" や"CATERR" のように表示されます。
 OpenIPMI-2.0.16, 2.0.17, 2.0.18 では"PhysicalSecurity0" または"CATERR0" と表示されます。

OpenIPMI-2.0.16、2.0.17、2.0.18 を使用して Zabbix サーバで IPMI item を設定する場合、
 IPMI エージェントアイテムの *IPMI sensor* フィールドにこれらの名前を 0 として使用してください。
 Zabbix サーバを OpenIPMI-2.0.19(またはそれ以降) を使用する新しい Linux ディストリビューションにアップグレードした場合、
 これらの IPMI センサーのアイテムは「NOT SUPPORTED」になります。そのため
 IPMI センサー名を変更し（末尾の'0' を削除）、再び"Enabled " になる前に、しばらく待つ必要があります。

しきい値センサーと離散センサーの同時利用に関する注意事項

一部の IPMI エージェントは、同じ名前でしきい値センサーと離散センサーの両方を提供します。常にしきい値センサーが優先されます。

接続終了時の注意事項

IPMI チェックが行われなかった場合 (何らかの理由: ホストの IPMI 項目が全て無効/サポートされていない、
 ホストが無効/削除された、ホストがメンテナンス中であるなど)
 Zabbix server / proxy を起動した時間に応じて 3～4 時間後に Zabbix server / proxy から IPMI 接続が切断されます。

SSH エージェント

概要

SSH チェックはエージェントレス監視として実行されます。SSH チェックに Zabbix エージェントは必要ありません。

SSH チェックを実行するには、Zabbix サーバーを最初に SSH2 サポート（libssh または libssh2）付きで設定する必要があります。あわせて要件も参照してください。

> **Attention:**
> RHEL 8 以降では、libssh のみがサポートされます。その他のディストリビューションでは、libssh2 よりも libssh の使用が推奨されます。

設定

パスフレーズ認証

SSH チェックは、ユーザー/パスワードのペアとキーファイルベースの 2 つの認証方法を提供します。

キーを使用しない場合は、ソースからビルドする場合に libssh または libssh2 を Zabbix にリンクする以外に、追加の設定は必要ありません。

キーファイル認証

SSH アイテムで鍵ベース認証を使用するには、サーバー設定にいくつかの変更が必要です。

Zabbix サーバー設定ファイル (zabbix_server.conf) を root として開き、次の行を探します。

```
##### SSHKeyLocation=
```

この行のコメントアウトを解除し、公開鍵と秘密鍵を配置するフォルダのフルパスを設定します。

```
SSHKeyLocation=/home/zabbix/.ssh
```

ファイルを保存し、その後 Zabbix サーバーを再起動します。

ここでの /home/zabbix は *zabbix* ユーザーアカウントのホームディレクトリであり、.ssh は、通常、ssh-keygen コマンドによってホームディレクトリ内に公開鍵と秘密鍵が生成されるディレクトリです。

通常、さまざまな OS ディストリビューション向けの Zabbix サーバーのインストールパッケージでは、*zabbix* ユーザーアカウントのホームディレクトリが別の場所、たとえば /var/lib/zabbix（システムアカウントの場合）に作成されます。

鍵を生成する前に、ホームディレクトリを /home/zabbix に再配置して、上記の Zabbix サーバー設定パラメータ SSHKeyLocation と一致させることができます。

> **Note:**
> *installation section* に従って *zabbix* アカウントを手動で追加した場合は、以下の手順を省略できます。その場合、*zabbix* アカウントのホームディレクトリはおそらくすでに /home/zabbix になっています。

zabbix ユーザーアカウントのホームディレクトリを変更するには、それを使用しているすべての稼働中プロセスを停止する必要があります。

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

ホームディレクトリの場所を変更し、必要に応じて（存在する場合は）移動するには、次のコマンドを実行します。

```
usermod -m -d /home/zabbix zabbix
```

旧しい場所にホームディレクトリが存在しない場合もあるため、新しい場所に作成する必要があります。安全に作成するには、次のようにします。

```
test -d /home/zabbix || mkdir /home/zabbix
```

すべてを安全にするため、ホームディレクトリの権限を設定する追加コマンドを実行できます。

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

以前停止したプロセスを再度起動できます。

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

これで、公開鍵と秘密鍵を生成する手順を次のコマンドで実行できます（見やすくするため、コマンドプロンプトはコメントアウトしています）。

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |        .        |
##### |      o          |
##### | .     o         |
##### |+     . S        |
##### |.+   o =         |
##### |E .   * =        |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
公開鍵と秘密鍵（id_rsa.pub と id_rsa）は、デフォルトで /home/zabbix/.ssh ディレクトリに生成されます。これは Zabbix サーバーの SSHKeyLocation 設定パラメータに対応しています。

Attention:
”rsa” 以外の鍵タイプも ssh-keygen ツールおよび SSH サーバーでサポートされる場合がありますが、Zabbix が使用する libssh2 ではサポートされない可能性があります。

シェル設定フォーム

この手順は、SSH チェックで監視される各ホストごとに 1 回だけ実行する必要があります。

以下のコマンドを使用すると、公開鍵ファイルをリモートホスト *10.10.10.10* にインストールでき、*root* アカウントで SSH チェックを実行できるようになります（見やすくするため、コマンドプロンプトはコメントアウトしています）。

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
##### RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Are you sure you want to continue connecting (yes/no)?
yes
##### Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
##### root@10.10.10.10's password:
<Enter root password>
##### Now try logging into the machine, with "ssh 'root@10.10.10.10'",
##### and check to make sure that only the key(s) you wanted were added.
```

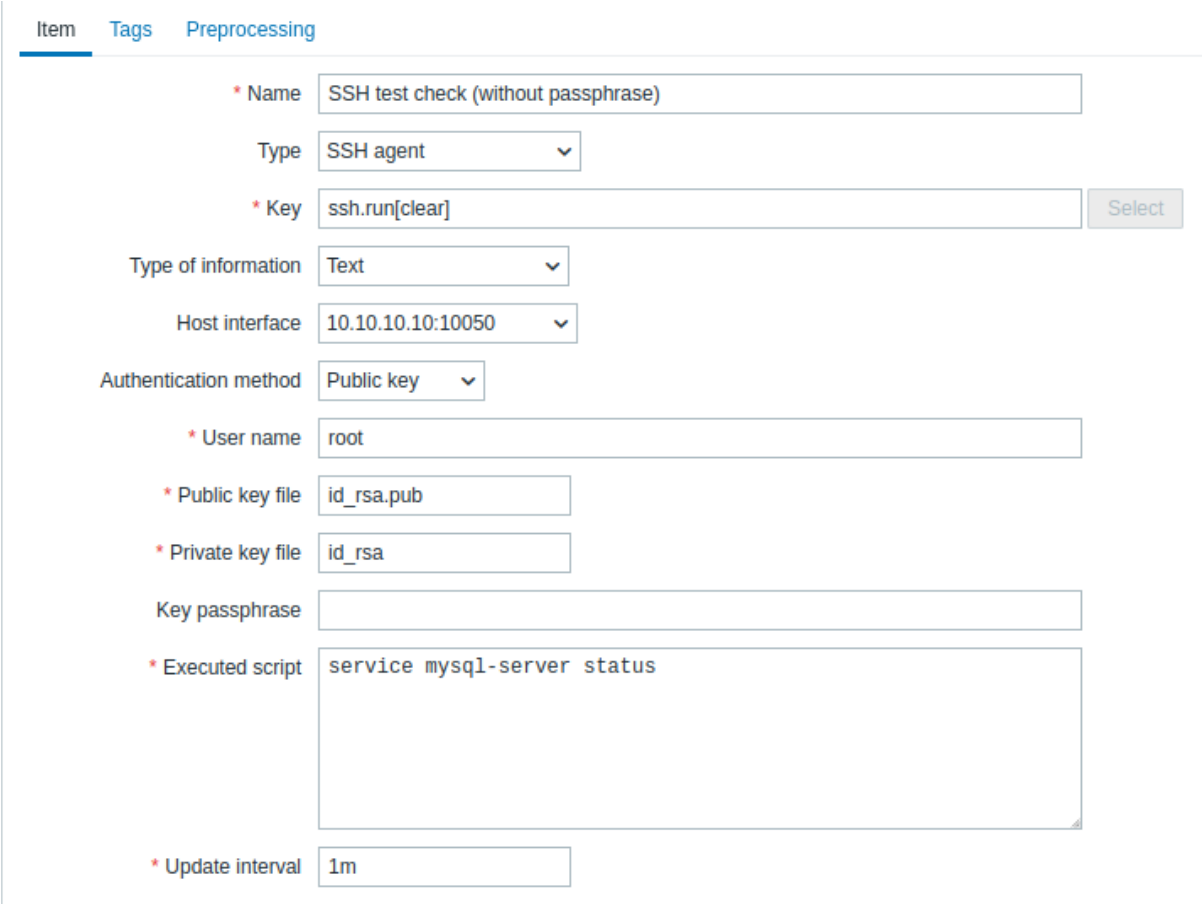
これで、*zabbix* ユーザーアカウントに対して、デフォルトの秘密鍵 (/home/zabbix/.ssh/id_rsa) を使用して SSH ログインを確認できるようになります。

```
sudo -u zabbix ssh root@10.10.10.10
```

ログインに成功した場合、シェル側の設定は完了です。リモートの SSH セッションを閉じてください。

アイテムの設定

実行する実際のコマンドは、アイテムの設定にある 実行するスクリプトフィールドに入力する必要があります。複数のコマンドは、それぞれを新しい行に入力することで、順番に実行できます。この場合、返される値も複数行としてフォーマットされます。



必須入力フィールドには、赤いアスタリスクが付いています。

SSH アイテムで固有の情報が必要なフィールドは次のとおりです。

パラメーター	説明	コメント
タイプ	ここで **SSH** エージェントを選択します。	

パラメーター	説明	コメント
キー	ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>] 形式の、ホストごとに一意なアイテムキー	**unique short description** は必須であり、ホストごとの各 SSH アイテムで一意にする必要があります。 デフォルトのポートは 22 です。このアイテムが割り当てられているインターフェースで指定されたポートではありません。 **ssh options** を使用すると、key1=value1;key2=value2,value3 形式で追加の SSH オプションを渡せます。1 つのキーに複数の値を渡す場合は、コンマで区切ります（この場合、パラメーターを引用符で囲む必要があります）。複数のオプションキーを渡す場合は、セミコロンで区切ります。 サポートされているオプションキーは、KexAlgorithms、HostkeyAlgorithms、Ciphers、MACs、PubkeyAcceptedKeyTypes です。オプションキーと値のサポート状況は SSH ライブラリによって異なります（たとえば、PubkeyAcceptedKeyTypes は libssh でのみサポートされています）。オプションがサポートされていない場合はエラーが返され、アイテムはサポートされない状態になります。 GnuTLS や OpenSSL のように暗号設定を追加するための「+」記号や、特定の暗号設定を無効にするための「!」記号はサポートされていないことに注意してください。 例: => ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128- => ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-he => ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey **subsystem** を使用すると SSH サブシステムを渡すことができ、サブシステムで許可された特定の操作に SSH 接続を制限できます（たとえば、SFTP を使用したファイル転送や、NETCONF を使用したネットワークデバイス管理）。サブシステムを使用する場合、実行するスクリプトパラメーターで特定のスクリプト構文も必要になることに注意してください。 例: => ssh.run[SFTPBackup,192.0.2.18,,,,sftp] => ssh.run[Cisco1234,192.0.2.18,,,,netconf]
認証方式	「Password」または「Public key」のいずれか。	
ユーザー名	リモートホストで認証するユーザー名（最大 255 文字）。必須。	
公開鍵ファイル	認証方式が「Public key」の場合の公開鍵のファイル名。必須。	例: *id_rsa.pub* - ssh-keygen コマンドで生成されるデフォルトの公開鍵ファイル名。
秘密鍵ファイル	認証方式が「Public key」の場合の秘密鍵のファイル名。必須。	例: *id_rsa* - デフォルトの秘密鍵ファイル名。

パラメーター	説明	コメント
パスワードまたは 鍵のパスフレーズ	認証に使用するパスワード（最大 255 文字）、または 秘密鍵に使用されている場合のパスフレーズ。	パスフレーズを使用していない場合は、鍵のパスフレーズフィールドを空のままにします。 パスフレーズの使用については、既知の問題も参照してください。

実行するスクリプト	SSH リモートセッションを使用して実行するシェルコマンド。	実行したシェルコマンドの戻り値は 16MB に制限されます（末尾の空白文字を含み、末尾の空白文字は切り捨てられます）。データベースの制限も適用されます。
		
		libssh2 ライブラリでは、実行可能なスクリプトが約 32kB に切り詰められる場合があることに注意してください。
		
		例:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		例（NETCONF サブシステムの場合）:
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

Telnet エージェント

概要

Telnet チェックはエージェントレス監視として実行されます。Zabbix エージェントは必要ありません。

設定可能なフィールド

実際に実行するコマンドは、アイテム設定の **Executed script** フィールドに記述する必要があります。
 複数のコマンドは、新しい行に記述することで順番に実行できます。この場合、返される値も複数行形式になります。

シェルプロンプトの末尾としてサポートされる文字:

- $
- #
- >
- %

Note:
これらの文字のいずれかで終わる telnet プロンプト行は返り値から削除されますが、これはコマンド一覧の最初のコマンドに対してのみ、つまり telnet セッションの開始時にのみ行われます。

Key	Description
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	telnet 接続を使用してリモートデバイス上でコマンドを実行します

Attention:
telnet チェックが非 ASCII 文字を含み、かつ非 UTF8 エンコーディングの値を返す場合は、キーの *<encoding>* パラメータを適切に指定する必要があります。詳細は返り値のエンコーディングページを参照してください。

JMX エージェント

概要

JMX 監視は、Java アプリケーションの JMX カウンタを監視するために使用できます。

JMX 監視は、”Zabbix Java gateway” と呼ばれる Zabbix デーモンの形で Zabbix にネイティブサポートされています。

ホスト上の特定の JMX カウンタの値を取得するために、Zabbix サーバーは Zabbix **Java gateway** にクエリを送り、Java gateway はJMX 管理 APIを使用して、対象のアプリケーションにリモートでクエリを送ります。

詳細およびセットアップについては、Zabbix Java gatewayのセクションを参照してください。

> **Warning:**
> Java gateway と監視対象の JMX アプリケーション間の通信は、ファイアウォールで遮断しないでください。

Java アプリケーションのリモート JMX 監視の有効化

Java アプリケーションには追加のソフトウェアをインストールする必要はありませんが、リモート JMX 監視をサポートするために、以下に指定するコマンドラインオプションを付けて起動する必要があります。

最低限、ローカルホスト上のシンプルな Java アプリケーションをセキュリティなしで監視したい場合は、次のオプションで起動します。

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.registry.ssl=false \
-jar /path/to/your/application.jar
```

これにより、Java はローカルホストからのみポート 12345 で JMX 接続を受け付け、認証や SSL を要求しないようになります。

別のインターフェースで接続を許可したい場合は、-Djava.rmi.server.hostname パラメータにそのインターフェースの IP を指定します。

セキュリティをより厳格にしたい場合は、他にも多くの Java オプションが利用できます。例えば、次の例では、より柔軟なオプションセットでアプリケーションを起動し、ローカルホストだけでなく広いネットワークに公開します。

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /path/to/your/application.jar
```

これらの設定のほとんど（すべてではないにしても）は、$JRE/lib/management/management.properties（またはシステム上のそのファイルがある場所）に指定できます。

SSL を使用する場合は、Java ゲートウェイがキーストアやトラストストアの場所を認識できるように、-Djavax.net.ssl.* オプションを追加して startup.sh スクリプトを修正する必要があることに注意してください。

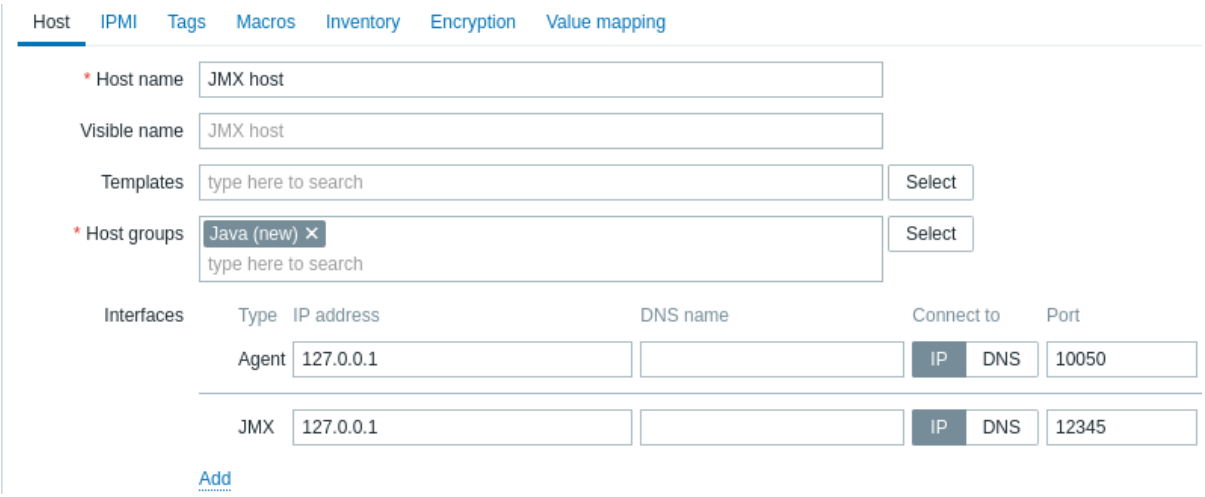
詳細については、Monitoring and Management Using JMXを参照してください。

Zabbix フロントエンドでの JMX インターフェースと item の設定

Java gateway が動作し、serveer が Java gateway の場所を認識し、リモート JMX 監視をサポートする Java アプリケーションが
 起動したら、Zabbix GUI でインタフェースと item を設定する必要があります。

JMX インターフェースの設定

まず、対象のホストに JMX タイプのインターフェースを作成します。

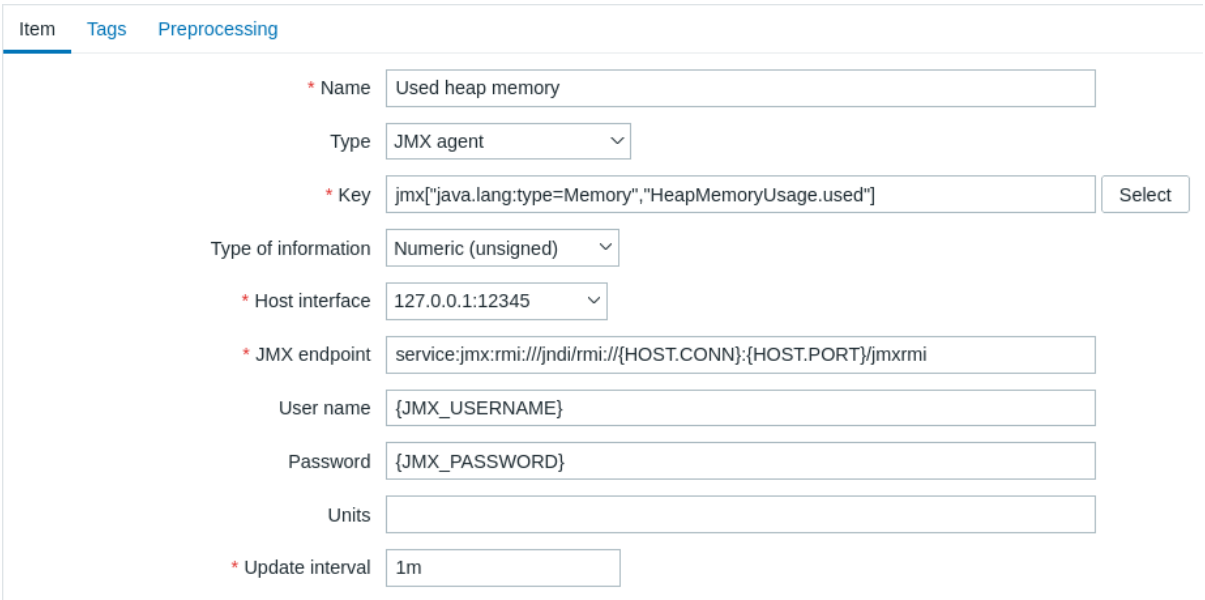


必須の入力フィールドには、赤いアスタリスクが付いています。

JMX エージェントアイテムの追加

関心のある各 JMX カウンターごとに、そのインターフェースに関連付けられた **JMX** エージェントアイテムを追加します。

以下のスクリーンショットのキーは `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]` です。



必須の入力フィールドには、赤いアスタリスクが付いています。

JMX アイテムに特有の情報が必要なフィールドは次のとおりです。

Type	ここで **JMX** エージェントを設定します。
Key	`jmx[]` アイテムキーには 3 つのパラメータが含まれます: **object name** - MBean のオブジェクト名 **attribute name** - ドットで区切られた任意の複合データフィールド名を含む MBean 属性名 **unique short description** - ホスト上で同じオブジェクト名と属性名を持つ複数の JMX アイテムを許可する一意の説明（任意） JMX アイテムキーの詳細については、以下を参照してください。 `jmx.discovery[]` low-level discovery アイテムを使用して、MBean と MBean 属性を検出できます。
JMX endpoint	カスタム JMX エンドポイントを指定できます。JMX エンドポイントの接続パラメータが JMX インターフェースと一致していることを確認してください。これは、デフォルトの JMX エンドポイントで使用されているように `{HOST.*}` マクロを使用することで実現できます。 `{HOST.*}` macros とユーザーマクロがサポートされています。
User name	Java アプリケーションで認証を設定している場合は、ユーザー名（最大 255 文字）を指定します。 ユーザーマクロがサポートされています。

Password	Java アプリケーションで認証を設定している場合は、パスワード（最大 255 文字）を指定します。 ユーザーマクロがサポートされています。

”true” または”false” のいずれかを返す Boolean カウンターを監視する場合は、情報の種類を”Numeric (unsigned)” に指定し、Preprocessing タブで”Boolean to decimal” の前処理ステップを選択します。サーバーは Boolean 値をそれぞれ 1 または 0 として保存します。

JMX item キーの詳細

シンプルな属性

MBean オブジェクト名は、Java アプリケーションで定義する文字列にすぎません。一方、属性名はより複雑になる場合があります。属性がプリミティブデータ型（整数、文字列など）を返す場合は、特に心配する必要はありません。キーは次のようになります。

```
jmx[com.example:Type=Hello,weight]
```

この例では、オブジェクト名は”com.example:Type=Hello”、属性名は”weight” で、返される値の型はおそらく” 数値（浮動小数点）” であるべきです。

複合データを返す属性

例えば、属性名が”apple” で、”weight” 、”color” などのパラメータを表すハッシュを返す場合、属性はより複雑になります。キーは次のようになります。

```
jmx[com.example:Type=Hello,apple.weight]
```

このように、属性名とハッシュ・キーは、ドット記号で区切られます。同じように、属性がネストした複合データを返す場合は、ドットで区切ります。

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

表形式データを返す属性

表形式データ属性は、1 つまたは複数の複合属性から構成されます。属性名パラメータにそのような
 属性が指定された場合、この item 値はその属性の完全な構造を JSON 形式で返します。
 表形式データ属性の中の個々の要素値は、プリプロセスによって取得することができます。

表形式データ属性の例:

Item 値:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

ドットに関する問題

ここまでは順調です。しかし、属性名やハッシュキーにドット記号が含まれている場合はどうでしょうか？以下はその例です。

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

これは問題です。属性名が”all.fruits” であることを、Zabbix にどのように伝えればよいのでしょうか？属性名とハッシュキーを区切るドットと、名前の一部であるドットをどのように区別すればよいのでしょうか？

これは可能です。名前の一部であるドットをバックスラッシュでエスケープするだけです。

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

同様に、ハッシュキーにドットが含まれている場合もエスケープします。

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

その他の問題

属性名内のバックスラッシュ文字はエスケープする必要があります。

```
jmx[com.example:type=Hello,c:\\documents]
```

JMX アイテムキー内のその他の特殊文字の扱いについては、アイテムキー形式のセクション を参照してください。

実際のところ、これで以上です。JMX 監視をお楽しみください。

非プリミティブデータ型

toString() メソッドをオーバーライドする非プリミティブデータ型を返すカスタム MBean を使用することができます。

JBoss EAP 6.4 でカスタムエンドポイントを使用する

カスタムエンドポイントを使用すると、デフォルトの RMI 以外のさまざまなトランスポートプロトコルを利用できます。

この可能性を示すために、例として JBoss EAP 6.4 の監視を設定してみましょう。まず、いくつか前提条件を置きます。

- すでに Zabbix Java ゲートウェイをインストールしていること。まだの場合は、documentation に従ってインストールできます。
- Zabbix サーバーと Java ゲートウェイは /usr/local/ をプレフィックスとしてインストールされていること
- JBoss はすでに /opt/jboss-eap-6.4/ にインストールされており、standalone モードで動作していること
- これらのコンポーネントはすべて同じホスト上で動作しているものとすること
- ファイアウォールと SELinux は無効化されている（または適切に設定されている）こと

zabbix_server.conf に簡単な設定を追加します。

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

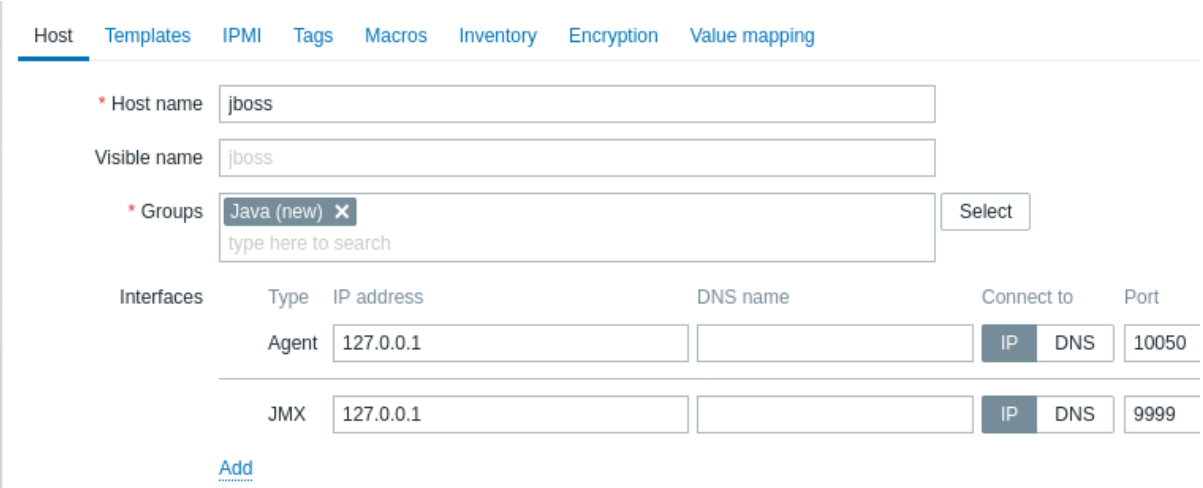
また、zabbix_java/settings.sh 設定ファイル（または zabbix_java_gateway.conf）に次を設定します。

```
START_POLLERS=5
```

JBoss が標準の管理ポートで待ち受けていることを確認します。

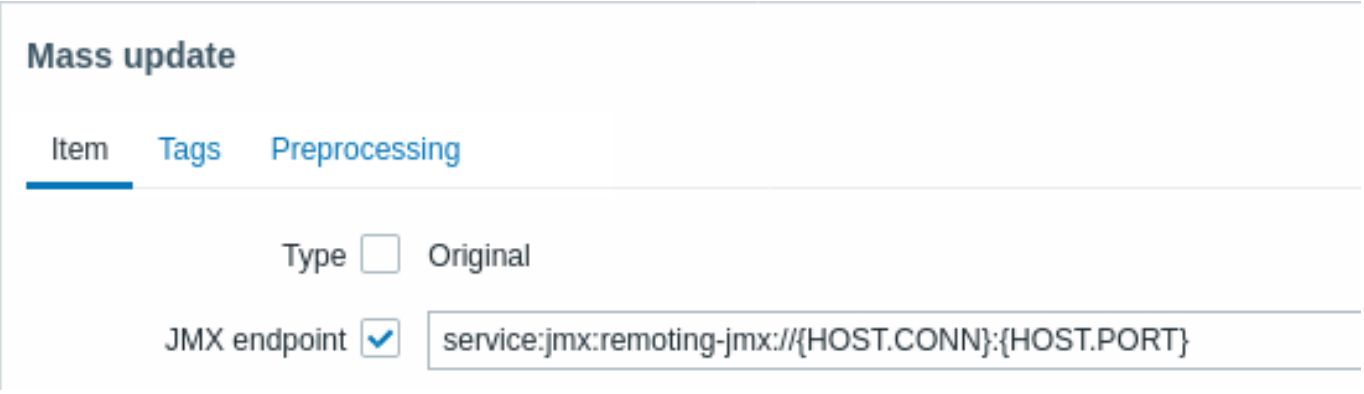
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

次に、Zabbix で JMX インターフェース 127.0.0.1:9999 を持つホストを作成します。



このバージョンの JBoss は RMI ではなく JBoss Remoting プロトコルを使用するため、JMX テンプレート内のアイテムに対して JMX エンドポイントパラメータを一括更新できます。

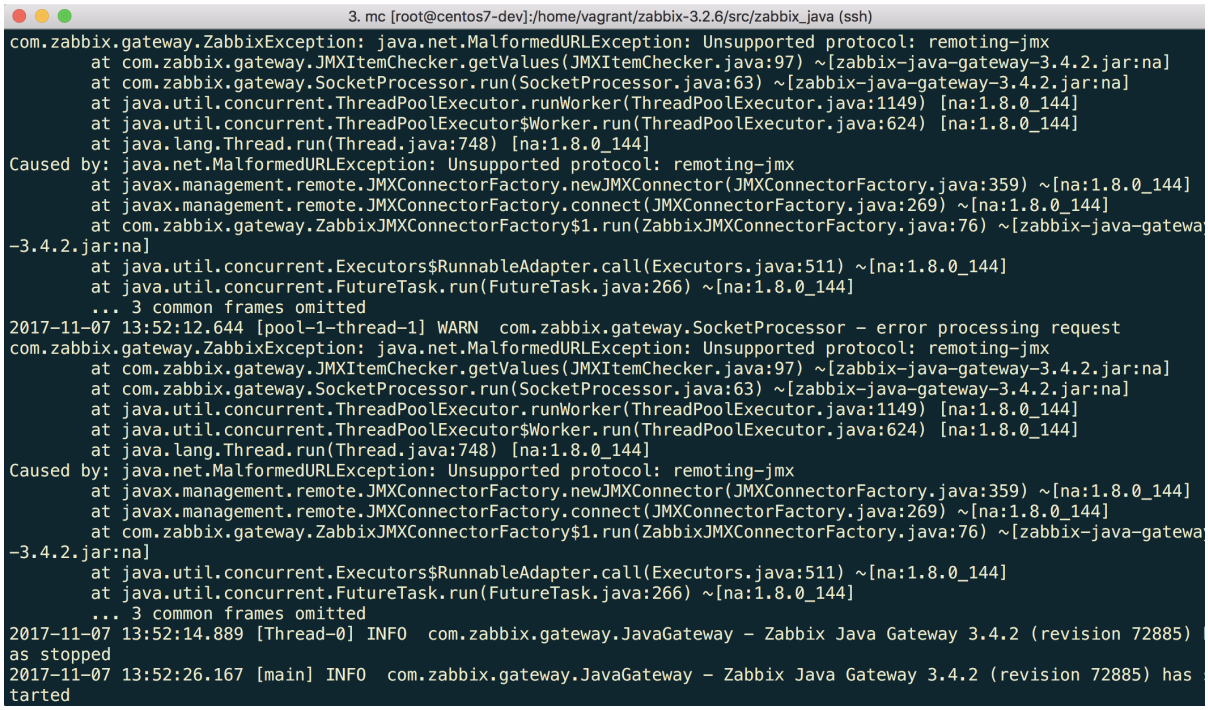
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



設定キャッシュを更新します。

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

最初にエラーが発生する場合があることに注意してください。



”Unsupported protocol: remoting-jmx” は、Java ゲートウェイが指定されたプロトコルの扱い方を知らないことを意味します。これは、次の内容を持つ ~/needed_modules.txt ファイルを作成することで解決できます。

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

その後、次のコマンドを実行します。

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

これで、Java ゲートウェイは jmx-remoting を扱うために必要なすべてのモジュールを持つことになります。あとは Java ゲートウェイを再起動し、少し待てば、正しく設定できていれば JMX 監視データが Zabbix に届き始めるのを確認できます（あわせて参照: Latest data）。

計算アイテム

概要

このアイテムタイプを使用すると、すでに存在するメトリクスに対して計算を実行できます。計算アイテム自体はデータを収集しません。

たとえば、次のような処理を実行できます。

- 2 つの値を合計する。
- 複数のアイテムの値の平均を計算する。
- アイテムのグループの合計値を計算する。

計算結果は、他のアイテムと同様に Zabbix データベースに保存されます。履歴値とトレンド値の両方が保存され、グラフを生成できます。

計算は数式に基づいて行われます。数式の構文は、トリガーの条件式と共通です。

実際には、複雑なデータ整形ロジックを計算アイテムに実装し、アイテムキーを使用して計算アイテムをトリガーから参照することで、トリガーをシンプルに保つことができます。

計算のデータソースを指定する方法は、次の 2 つです。

1. 数式内でアイテムを指定する。
2. アイテムを選択するフィルターに基づいてデータ集計を行う。

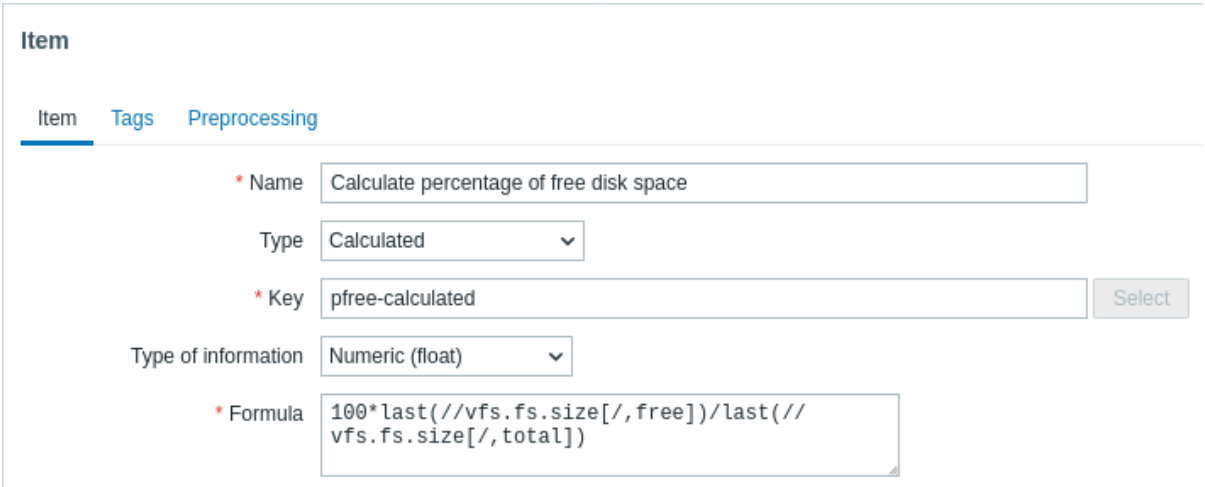
2 つ目の方法では、グループの平均値や合計値などを計算できます。

すべての計算は Zabbix サーバーによって実行され、`history poller` プロセスによって処理されます。プロキシやエージェントは一切関与しません。長い期間を対象とする計算アイテムを大量に使用すると、サーバーのパフォーマンスに影響する可能性がある点に注意してください。

計算アイテムは、計算アイテムを作成するユーザーがアクセス権を持たないアイテムを含め、他のアイテムの履歴にアクセスできる点に注意してください。この動作が望ましくない場合は、`StartHistoryPollers`設定パラメータを 0 に設定して、計算アイテムを無効にできます。

設定可能なフィールド

計算アイテムを使用するには、アイテムの種類として **Calculated** を選択します。



キーは、ホストごとに一意のアイテム識別子です。サポートされている記号を使用して、任意のキー名を作成できます。キーを使用すると、トリガー内で計算アイテムを参照できます。

計算式は **Formula** フィールドに入力する必要があります。

計算式とキーの間に関連性はありません。キーのパラメータは、計算式では一切使用されません。

トリガーの式とは異なり、Zabbix は新しい値を受信したときではなく、アイテムの更新間隔に従って計算アイテムを処理します。

情報の種類として *Numeric (unsigned)* を選択し、計算結果が浮動小数点値の場合は、整数に切り捨てられます。

単純計算式

単純な計算式の構文は次のとおりです:

`function(/host/key,<parameter1>,<parameter2>,...)`

各要素の説明:

Element	Description
`function`	サポートされている関数のいずれか: last、min、max、avg、count など
`host`	計算に使用されるアイテムのホスト。 現在のホストは省略できます（つまり、`function(//key,parameter,...)` のように指定できます）。

Element	Description
`key`	計算に使用されるアイテムのキー。 バイナリまたは JSON のデータ型の値を返すアイテムはサポートされていません。 参照されているアイテムのキーを変更した場合、計算式内のキーも手動で更新する必要があります。 履歴関数によって参照されるすべてのアイテムは、存在しており、データ収集を行っている必要があります。
`parameter(s)`	必要に応じて指定する関数のパラメータ。 時間サフィックスおよびメモリサイズサフィックスがサポートされています。

より複雑な計算式では、関数、演算子、括弧を組み合わせて使用できます。

計算アイテムの計算式の例を参照してください:

- (1) インターフェースの総トラフィック
- (2) 空きディスク容量の割合
- (3) 受信トラフィックの割合
- (4) 処理済み値の平均数

一部の変数は、ユーザーマクロを使用して計算式に渡すことができます。
ユーザーマクロについては、次の点に注意してください:

- 関数パラメータ、アイテムフィルターパラメータ、または定数の参照に使用された場合は、展開されます。
- 関数、ホスト名、アイテムキー、アイテムキーパラメータ、または演算子を参照する場合は、展開されません。

文字列との比較も可能です。

キャッシュ内に新しいデータがなく、かつ関数内で問い合わせ期間が定義されていない場合、Zabbix はデフォルトで最大 1 週間前までさかのぼってデータベースから履歴値を問い合わせます。

計算アイテムは、いくつかのケースで未サポートになる場合があります:

- 参照されているアイテムが
  - 見つからない
  - 無効化されている、または無効化されたホストに属している
  - 未サポートである（ただし、nodata() 関数および unknown 値を持つ演算子の場合を除く）
- 関数を計算するためのデータがない
- ゼロによる除算
- 構文が正しくない

データ集計

データ集計は、複数のアイテムを選択するフィルタに基づいています。データ集計も計算アイテムの一種ですが、数式では少し異なる構文を使用します。

集計値を取得するには、サポートされている集計関数のいずれかを使用します: `avg`、`max`、`min`、`sum` など。次に、必要なアイテムを選択するために、唯一のパラメータとして **foreach** 関数とそのアイテムフィルタを追加します:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

foreach 関数（例: *avg_foreach*、*count_foreach* など）は、選択された各アイテムに対して 1 つの集計値を返します。アイテムは、アイテム履歴から、アイテムフィルタ（`/host/key?[group="host group"]`）を使用して選択されます。詳細については、foreach 関数を参照してください。

要求された期間に対してデータを持たないアイテムがある場合、それらは計算で無視されます。どのアイテムにもデータがない場合、関数はエラーを返します。

または、集計のパラメータとして複数のアイテムを列挙することもできます:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

ここでの `function` は、履歴/トレンド関数でなければならないことに注意してください。

集計計算の例を参照してください:

- (5) ホストの受信トラフィック合計
- (6) グループのディスク容量合計
- (7) グループの平均プロセッサ負荷
- (8) グループおよびタグごとの平均プロセッサ負荷
- (9) 処理済み値の平均数

- (10) グループの平均クエリ数
- (11) 成功した DNS チェックの総数
- (12) グループの未サポートアイテム総数
- (13) グループのインターフェース統計

ユーザーマクロおよびローレベルディスカバリマクロは、以下でサポートされています:

- アイテムキーのパラメータ
- 関数パラメータ
- フィルタ条件（ホストグループ名およびタグ名）
- 式の定数

以下の場合、集計計算は未サポートになることがあります:

- 参照されているアイテムが 1 つも見つからない場合（これは、アイテムキーが正しくない、いずれのアイテムも存在しない、または含まれるすべてのグループが正しくない場合に発生する可能性があります）
- 関数を計算するためのデータがない場合

計算アイテムの例

(1) インターフェース上の総トラフィック

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

eth0 の総帯域幅を計算します。

(2) 空きディスク容量の割合

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix は、空きディスク容量と総ディスク容量（/上）の最新の値を取得し、指定された式に従って割合を計算します。

(3) 受信トラフィックの割合

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

総トラフィックに対する受信トラフィックの割合を計算します。

(4) 処理された値の平均数

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Zabbix によって処理された値の数の 10 分間平均を計算します。

(5) ホストの受信トラフィック合計

```
sum(last_foreach(/host/net.if.in[*]))
```

ホスト上で net.if.in[*] に一致するすべてのアイテムの合計を計算します。

(6) グループの合計ディスク容量

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

すべての MySQL Servers の合計ディスク容量を計算します。

(7) グループの平均プロセッサー負荷

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

すべての MySQL Servers の平均プロセッサー負荷を計算します。

(8) グループおよびタグの平均プロセッサー負荷

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

特定のタグを持つ複数のホストグループ内のすべてのホストにおける平均 CPU 負荷を計算します。

(9) グループのクエリ平均数

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

MySQL Servers の 1 秒あたりのクエリ数の 5 分間平均を計算します。

(10) 成功した DNS チェックの合計数

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

すべてのホストにわたる成功した DNS チェックの合計数を計算します。
表示されているアイテム形式は、1 つの可能なキーの例として net.dns[192.0.2.1,example.com,A] です。

ワイルドカードは、キー内のパラメータ数と一致する必要があることに注意してください。ここではnet.dns に 3 つのパラメータがあります: ip, name, type。

(11) グループの未サポートアイテム総数

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Zabbix servers の未サポートアイテムの総数を計算します。

(12) グループのインターフェース統計

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

video グループのインターフェース統計を計算します。

正しい構文と誤った構文の例

履歴、トレンド、または foreach 関数 のパラメータとして、式（関数呼び出しを含む）を使用することはできません。ただし、それらの関数自体は、他の（履歴以外の）関数パラメータで使用できます。

式	例
有効	avg(last(/host/key1),last(/host/key2)*10,last(/host/key1)*100)
	max(avg(avg_foreach(/*/system.cpu.load?[group="Servers A"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers B"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers C"],5m)))
無効	sum(/host/key,10+2)
	sum(/host/key, avg(10,2))
	sum(/host/key,last(/host/key2))

次のような式の場合は注意してください。

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

(resptime[*] は、resptime[1]、resptime[2] などのキーに一致するアイテムキーのパターンの例です。複数のアクティブなクラスターノードから応答時間を取得するために使用されます)、式の両方の部分が常に同じ値のセットを持つとは限りません。式の一方の部分が評価されている間に、要求された期間の新しい値が到着すると、式のもう一方の部分では異なる値のセットが使用されることになります。

依存アイテム

概要

1 つのアイテムが一度に複数のメトリックを収集する場合や、関連するメトリックを同時に収集した方が理にかなっている場合があります。例えば:

- 個々のコアの CPU 使用率
- 受信/送信/合計ネットワークトラフィック

一括メトリック収集と複数の関連アイテムでの同時利用を可能にするために、Zabbix は従属アイテムをサポートしています。従属アイテムは、データを同時に 1 つのクエリで収集するマスターアイテムに依存します。マスターアイテムに新しい値が入ると、自動的に従属アイテムの値も設定されます。従属アイテムは、マスターアイテムと異なる更新間隔を持つことはできません。

Zabbix の前処理オプションを使用して、マスターアイテムのデータから従属アイテムに必要な部分を抽出できます。

前処理は preprocessing manager プロセスによって管理され、ワーカースレッドが前処理ステップを実行します。前処理付きのすべての値は、異なるデータ収集プロセスから受信され、ヒストリキャッシュに追加される前に前処理マネージャを通過します。データ収集プロセス（ポーラー、トラッパーなど）と前処理プロセス間の通信には、ソケットベースの IPC 通信が使用されます。

Zabbix サーバーまたは Zabbix プロキシ（ホストがプロキシで監視されている場合）が前処理ステップと従属アイテムの処理を実行します。

どのタイプのアイテムでも、従属アイテムであっても、マスターアイテムとして設定できます。既存の従属アイテムの値からさらに小さな部分を抽出するために、追加の従属アイテム階層を使用することもできます。

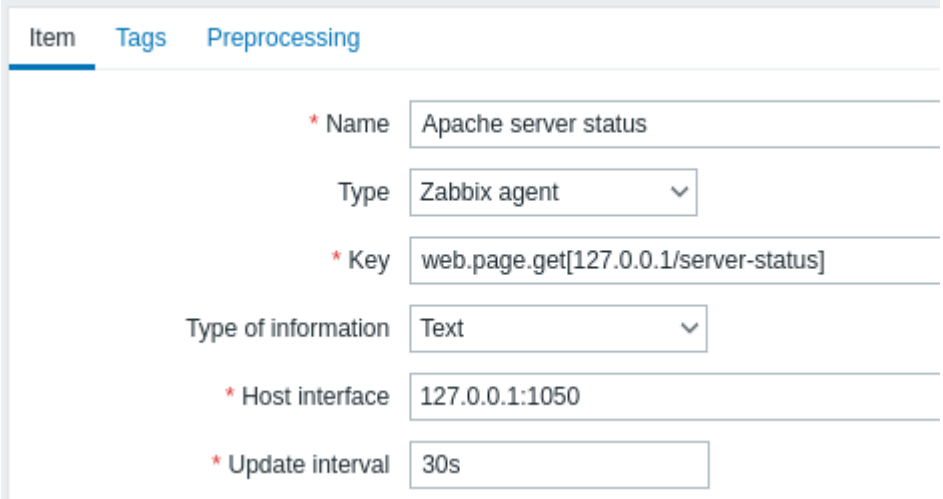
制限事項

- 同じホスト（テンプレート）間の依存関係のみ許可されます
- アイテムプロトタイプは、同じホストの他のアイテムプロトタイプまたは通常のアイテムに依存できます
- テンプレートからのマスターアイテムを持つホスト上の従属アイテムは、XML にエクスポートされません

アイテムの設定

依存アイテムは、データを取得するためにマスターアイテムに依存します。そのため、まずマスターアイテムを設定する（または存在させる）必要があります。

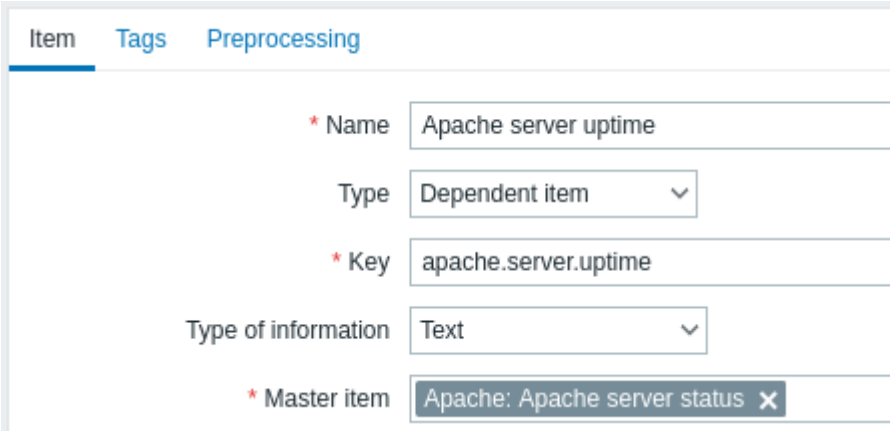
1. データ収集 > ホストに移動します。
2. ホストの行にある アイテムをクリックします。
3. アイテムの作成をクリックします。
4. フォームにアイテムのパラメータを入力します。



必須入力フィールドには、赤いアスタリスクが付いています。

追加をクリックしてマスターアイテムを保存します。

その後、依存アイテムを設定できます。

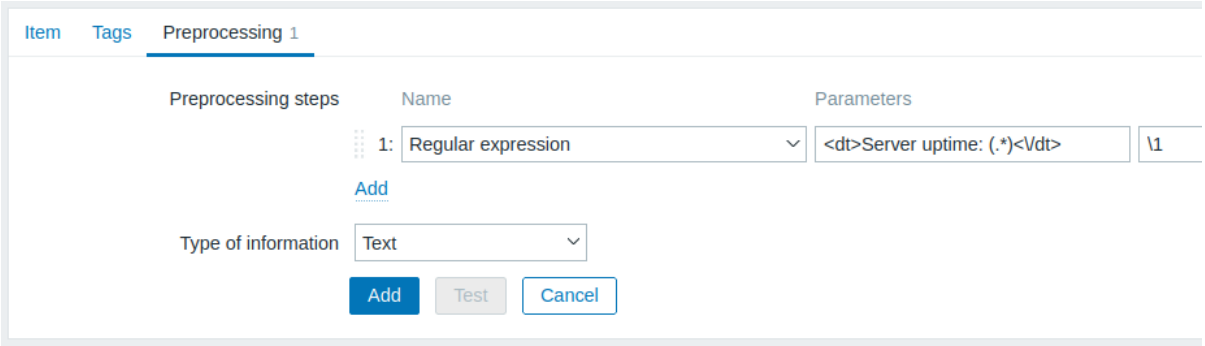


必須入力フィールドには、赤いアスタリスクが付いています。

依存アイテムに固有の情報を指定する必要があるフィールドは次のとおりです。

Type	ここで **Dependent item** を選択します。
Key	アイテムを識別するために使用するキーを入力します。
Master item	マスターアイテムを選択します。マスターアイテムの値が依存アイテムの値の入力に使用されます。
Type of information	保存されるデータの形式に対応する情報の種類を選択します。

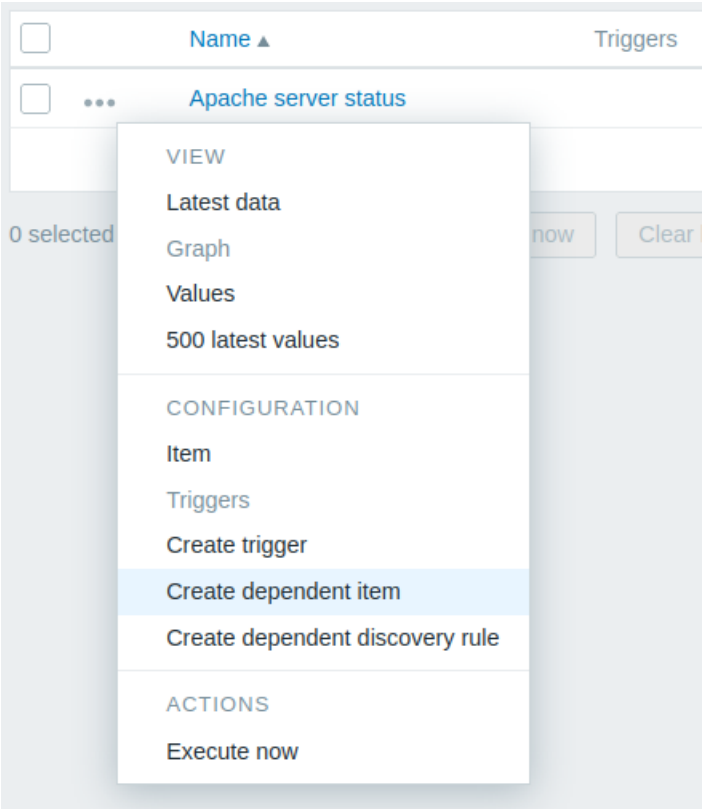
アイテム値の前処理を使用して、マスターアイテムの値から必要な部分を抽出できます。



前処理を行わない場合、依存アイテムの値はマスターアイテムの値と完全に同じになります。

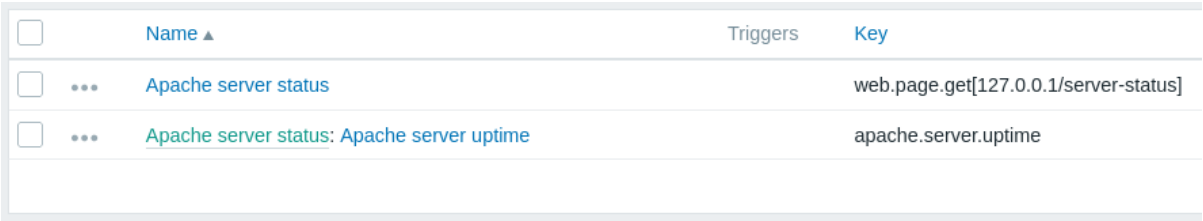
追加をクリックして依存アイテムを保存します。

依存アイテムをより簡単に作成するショートカットは、アイテム一覧で ••• ボタンをクリックし、依存アイテムの作成を選択すると利用できます。



表示

アイテムリストでは、依存アイテムはマスターアイテム名をプレフィックスとして表示されます。



マスターアイテムが削除された場合、そのアイテムに依存する依存アイテムもすべて削除されます。

スクリプトアイテム

概要

スクリプトアイテムは、ユーザー定義の JavaScript コードを実行してデータを収集するために使用でき、HTTP/HTTPS 経由でデータを取得することも可能です。スクリプトに加えて、オプションでパラメータのリスト（名前と値のペア）やタイムアウトを指定できます。

このアイテムタイプは、複数のステップや複雑なロジックを必要とするデータ収集シナリオで役立ちます。例えば、スクリプトアイテムを設定して HTTP コールを実行し、最初のステップで受信したデータを何らかの方法で処理し、変換した値を 2 回目の HTTP コールに渡すことができます。

スクリプトアイテムは、Zabbix サーバーまたはプロキシのポーラーによって処理されます。

設定

アイテム設定フォームのタイプフィールドで「スクリプト」を選択し、必要なフィールドに入力します。

script_item.png

すべての必須入力フィールドには赤いアスタリスクが付いています。

スクリプトアイテムに固有の情報が必要なフィールドは次のとおりです。

フィールド	説明
キー	アイテムを識別するために使用される一意のキーを入力します。
パラメータ	スクリプトに渡す変数を属性と値のペアで指定します。 ユーザーマクロがサポートされています。サポートされている組み込みマクロを確認するには、サポートされているマクロテーブルで「Script-type item」を検索してください。
スクリプト	パラメータフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。このコードはメトリック値を返すためのロジックを提供する必要があります。 このコードは、すべてのパラメータおよび Zabbix によって追加された追加の JavaScript オブジェクトにアクセスできます。 参考: JavaScript ガイド。
タイムアウト	JavaScript の実行タイムアウト（1～600 秒；これを超えるとエラーが返されます）。 スクリプトによっては、タイムアウトが発生するまでに時間がかかる場合があります。 タイムアウトパラメータの詳細については、一般的なアイテム属性を参照してください。

例

シンプルなデータ収集

https://www.example.com/release_notes の内容を収集します:

- タイプが「スクリプト」のアイテムを作成します。
- スクリプトフィールドに以下を入力します:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

パラメータを使用したデータ収集

特定のページの内容を収集し、パラメータを利用します:

- タイプ「スクリプト」と 2 つのパラメータを持つアイテムを作成します:
  - **url : {$DOMAIN}**（ユーザーマクロ {$DOMAIN} は、できればホストレベルで定義してください）
  - **subpage : /release_notes**
- スクリプトフィールドに以下を入力します:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

複数の HTTP リクエスト

https://www.example.com と *https://www.example.com/release_notes* の両方の内容を収集します。

- タイプが「スクリプト」のアイテムを作成します。
- スクリプトフィールドに以下を入力します。

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

ロギング

Zabbix server ログに”Log test” エントリを追加し、アイテム値”1” を返信します。

- タイプ"Script" の item を作成します。
- *Script* フィールドに、以下のコードを入力します。

```
Zabbix.log(3, 'Log test');
return 1;
```

ブラウザーアイテム

概要

Browser アイテムを使用すると、ブラウザーを使って複雑な Web サイトや Web アプリケーションを監視できます。

> **Attention:**
> Browser アイテムのサポートは現在、実験的な機能です。

Browser アイテムは、ユーザー定義の JavaScript コードを実行し、HTTP/HTTPS 経由でデータを取得することでデータを収集します。このアイテムは、クリック、テキスト入力、Web ページ間の移動、その他の Web サイトや Web アプリケーションに対するユーザー操作など、ブラウザー関連の動作をシミュレートできます。

スクリプトに加えて、オプションのパラメーター一覧（名前と値のペア）およびタイムアウトを指定できます。

> **Attention:**
> このアイテムは、Selenium Server またはプレーンな WebDriver（たとえば ChromeDriver）を Web テストのエンドポイントとして使用し、W3C WebDriver standardを部分的に実装しています。このアイテムを動作させるには、Zabbix のサーバー/プロキシ設定ファイルで `WebDriverURL` パラメータをエンドポイントに設定してください（ChromeDriver を使用する場合は、Security Considerations を参照してください）。より良いパフォーマンスを得るには、Web テスト環境専用のサーバーを使用することを検討してください。

Browser アイテムのチェックは、Zabbix サーバーまたはプロキシの browser poller によって実行および処理されます。必要に応じて、Zabbix のサーバー/プロキシ設定ファイルの `StartBrowserPollers` パラメータで、事前生成される browser poller インスタンス数を調整できます。

複雑な Web サイトや Web アプリケーションの監視には、Website by Browser テンプレートをout-of-the-box templateとして利用できます。

設定

アイテム設定フォームのタイプフィールドで「ブラウザー」を選択し、必要なフィールドに入力します。

すべての必須入力フィールドには赤いアスタリスクが付いています。

ブラウザーアイテムに固有の情報が必要なフィールドは次のとおりです。

フィールド	説明
キー	アイテムを識別するために使用される一意のキーを入力します。
パラメーター	スクリプトに渡す変数を属性と値のペアで指定します。 ユーザーマクロがサポートされています。サポートされている組み込みマクロを確認するには、サポートされているマクロテーブルで「ブラウザータイプのアイテム」を検索してください。
スクリプト	パラメーターフィールドをクリックするか、その横の鉛筆アイコンをクリックすると開くモーダルエディターに JavaScript コードを入力します。このコードはメトリック値を返すロジックを提供する必要があります。 コードは、すべてのパラメーター、すべての追加 JavaScript オブジェクト、および Zabbix によって追加されたブラウザーアイテム JavaScript オブジェクトにアクセスできます。 参考: JavaScript ガイド。
タイムアウト	JavaScript 実行のタイムアウト（1～600 秒；これを超えるとエラーが返されます）。 スクリプトによっては、タイムアウトが発生するまでに時間がかかる場合があります。 タイムアウトパラメーターの詳細については、一般的なアイテム属性を参照してください。

例

Website by Browser テンプレートを使用して Web サイトを監視するための Zabbix の設定例については、Browser アイテムを使用して Web サイトを監視する を参照してください。

デフォルトスクリプト

以下のスクリプトは、次の処理を行います。

1. ブラウザーセッションを初期化します。
2. 指定された URL に移動します。
3. パフォーマンスエントリとセッション統計を収集し、JSON 文字列として返します。

スクリプトフィールドに以下を入力します。

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

カスタム機能でブラウザを初期化する

以下のスクリプトは:

1. スクリプト内で指定された順序で最初に一致するブラウザに基づいて、利用可能なブラウザのセッションを初期化します。
2. ページロード戦略や、Chrome、Firefox、Microsoft Edge ブラウザのヘッドレスモードなど、各ブラウザ固有のオプションを含むブラウザ機能を定義します。

スクリプトフィールドに以下を入力します:

```
var browser = new Browser({
    "capabilities":{
        "firstMatch":[
            {
                "browserName":"chrome",
                "pageLoadStrategy":"normal",
                "goog:chromeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"firefox",
                "pageLoadStrategy":"normal",
                "moz:firefoxOptions":{
                    "args":[
                        "--headless"
                    ]
                }
            },
            {
                "browserName":"MicrosoftEdge",
                "pageLoadStrategy":"normal",
                "ms:edgeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"safari",
                "pageLoadStrategy":"normal"
            }
        ]
    }
});
```

GUI でブラウザを初期化する

デフォルトでは、ブラウザセッション（Safari を除く）はヘッドレスモードで初期化されます。つまり、ブラウザのグラフィカルユ

ーザーインターフェース（GUI）は表示されません。

以下のスクリプトは、GUI を有効にしたブラウザセッションを初期化します。

WebDriver がブラウザのバイナリを見つけられない場合は、パスを手動で指定できます。

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// GUI で Firefox セッションを初期化するには、以下の行のコメントを外してください。
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// GUI で Microsoft Edge セッションを初期化するには、以下の行のコメントを外してください。
// var opts = Browser.edgeOptions();
// opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = 'usr/bin/microsoft-edge';
// opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];

var browser = new Browser(opts);
```

Note:
テストがリモートサーバーやコンテナで実行されている場合は、VNC（Virtual Network Computing）クライアントを使用してマシンの VNC サーバーに接続できます。これにより、ブラウザの GUI をリモートで表示および操作できます。

スクリーンショットを撮る

以下のスクリプトは:

1. ブラウザセッションを初期化します。
2. ブラウザのビューポートサイズを設定してスクリーンショットのサイズを決定します（パラメータで指定、下記参照）。
3. URL に移動します（パラメータで指定、下記参照）。
4. セッション統計情報を収集し、スクリーンショットを撮影して収集した統計情報に追加します。
5. エラーが発生した場合は、エラーメッセージとスクリーンショットを取得します。
6. 収集した結果を JSON 文字列として返します。

このスクリプトは、アイテム設定フォームからのパラメータも使用します:

- webURL - http://example.com
- width - 1920
- height - 1080

スクリプトフィールドに以下を入力します:

```
var browser, result;

var browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Zabbix から渡されたパラメータを含む JSON 文字列を解析します。

    browser.setScreenSize(Number(params.width), Number(params.height))

    browser.navigate(params.webURL);

    result = browser.getResult();
    result.screenshot = browser.getScreenshot();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
```

```
    return JSON.stringify(result);
}
```

Zabbix ログインのチェック

以下のスクリプトは、

1. ブラウザセッションを初期化します。
2. ページに移動します（パラメータで指定、下記参照）。
3. ユーザー名とパスワードを入力します（パラメータで指定、下記参照）。
4. ログインボタンを見つけてクリックします。
5. ログアウトボタンを見つけてクリックします。
6. ログイン前後およびログアウト後のパフォーマンスデータを収集します。
7. エラーが発生した場合、エラーメッセージとスクリーンショットを取得します。
8. 収集した結果を JSON 文字列として返します。

また、アイテム設定フォームのパラメータも使用します。

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

スクリプトフィールドには、以下を入力します。

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from Zabbix.

    browser.navigate(params.webURL);
    browser.collectPerfEntries("open page");

    var el = browser.findElement("xpath", "//input[@id='name']");
    if (el === null) {
        throw Error("cannot find name input field");
    }
    el.sendKeys(params.username);

    el = browser.findElement("xpath", "//input[@id='password']");
    if (el === null) {
        throw Error("cannot find password input field");
    }
    el.sendKeys(params.password);

    el = browser.findElement("xpath", "//button[@id='enter']");
    if (el === null) {
        throw Error("cannot find login button");
    }
    el.click();

    browser.collectPerfEntries("login");

    el = browser.findElement("link text", "Sign out");
    if (el === null) {
        throw Error("cannot find logout button");
    }
    el.click();

    browser.collectPerfEntries("logout");

    result = browser.getResult();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
```

```
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

リンクの検索

以下のスクリプトは:

1. ブラウザセッションを初期化します。
2. 配列から重複要素を削除する関数を定義します（ステップ 5 を参照）。
3. ページに移動します（パラメータで指定、下記参照）。
4. ページ上のリンクを検索します。
5. リンクの重複を削除し、一意にします。
6. "http" で始まるリンクのみを抽出します。
7. 抽出したリンクを特定の構造にフォーマットします。
8. エラーが発生した場合、エラーメッセージとスクリーンショットを取得します。
9. 収集した結果を JSON 文字列として返します。

また、アイテム設定フォームのパラメータも使用します:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

スクリプトフィールドに以下を入力します:

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value);  // Zabbix から渡されたパラメータを含む JSON 文字列を解析します。

    function uniq(a) {
        return a.sort().filter(function (item, pos, ary) {
            return !pos || item != ary[pos - 1];
        });
    }

    browser.navigate(params.scheme + '://' + params.domain + params.path);

    var el = browser.findElements("link text", "");
    var links = [];
    for (var n = 0; n < el.length; n++) {
        links.push(el[n].getAttribute('href'));
    }

    links = uniq(links);

    result = [];
    for (i = 0; i < links.length; i++) {
        if (links[i].match(/^http.*/)) {
            var row = {};
            row["{#URL}"] = links[i];
            result.push(row);
        }
    }
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
```

```
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

テレメトリクエリ

このアイテムタイプを使用すると、テレメトリデータを取得または集計できます。テレメトリデータは、アプリケーションパフォーマンス監視（APM）に使用されます。APM により、アプリケーションやサービスを通過するリクエストの流れを確認し、パフォーマンスを測定して、処理が遅い、または失敗した原因を特定できます。

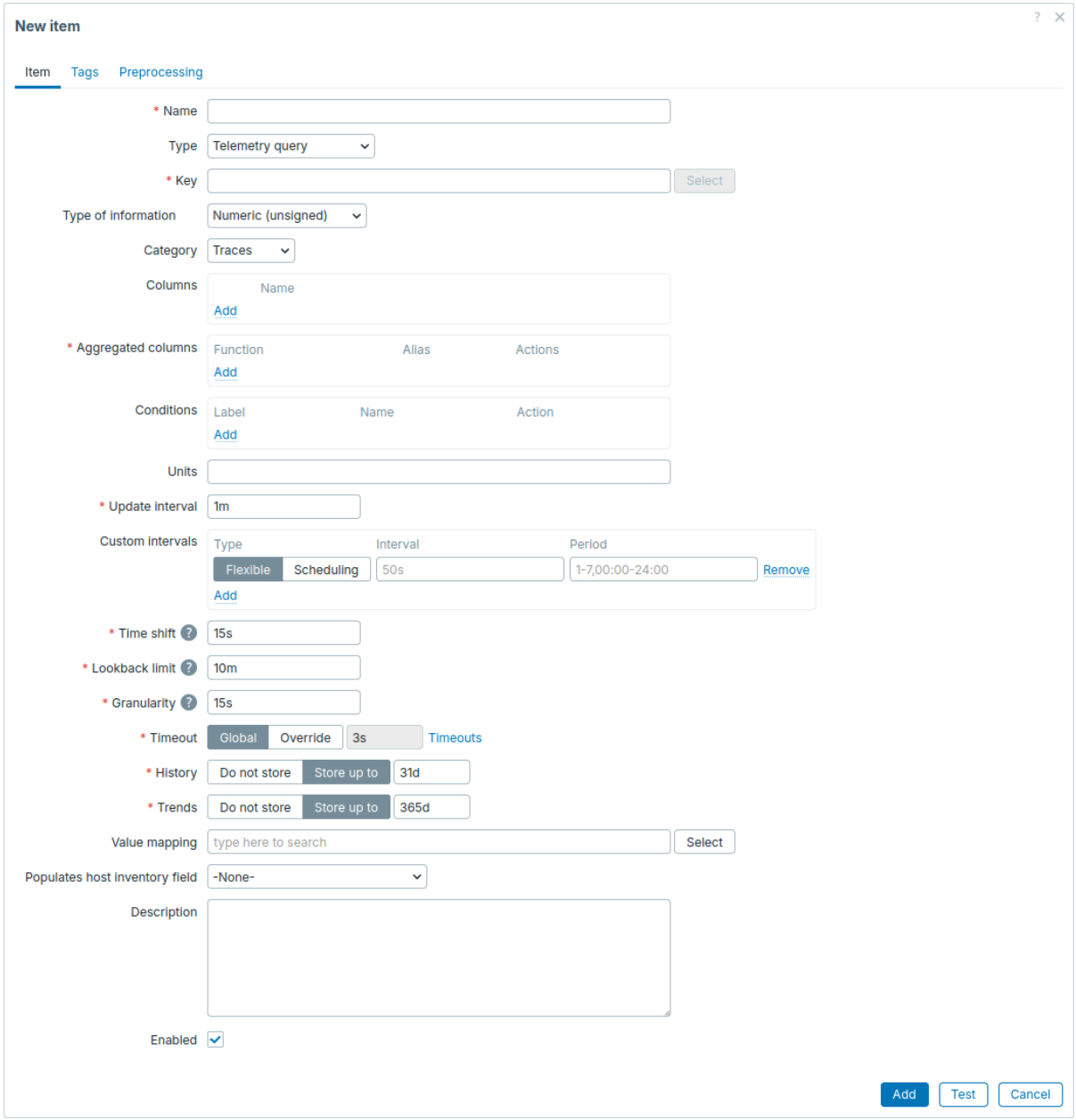
テレメトリデータには、トレース（アプリケーションやサービスを通過するリクエストの経路）、メトリクス（応答時間などの測定値）、ログ（イベントのテキスト記録）の 3 種類があります。テレメトリクエリアイテムを使用すると、たとえば、サービスがエラーを返したすべてのトレースを検索したり、メトリクスから平均応答時間を取得したり、特定のエラーメッセージをログから検索したりできます。

テレメトリクエリアイテムを設定するときは、どのテレメトリデータを要求するか、どのようにフィルターするか、どのように集計するかを Zabbix に指示する一連の命令（クエリ）を定義します。アイテムのチェックが実行されると、Zabbix はこのクエリをテレメトリデータソースに送信し、クエリに基づいて結果を取得します。

このアイテムを使用するには、テレメトリデータソースを指定し、アイテムのホストを監視するのがZabbix サーバーとZabbix プロキシのどちらであるかに応じて、その設定で `StartTelemetryQueryPollers` パラメータを設定します。

設定

アイテムを作成するときに、タイプを *Telemetry query* に設定します。



パラメータ	説明
名前	すべてのアイテムに共通するパラメータを参照してください。
タイプ	*Telemetry query* を選択します。
キー	すべてのアイテムに共通するパラメータを参照してください。
情報のタイプ	
カテゴリー	クエリするシグナルのタイプとして、*Traces*、*Metrics*、または *Logs* を選択します。
メトリクスポイント	クエリするメトリクスポイントのタイプとして、*Sum*、*Gauge*、*Histogram*、または *Exponential histogram* を選択します。 カテゴリーが *Metrics* に設定されている場合に使用できます。
列	クエリするテレメトリデータ列を追加します。キーと値のペアを保持する列の場合は、キー名も入力します。これらの列のレコードは、そのままアイテムの値に含まれます。
集計列	クエリするテレメトリデータ列と、そのレコードから値を計算する方法を追加します。 関数 - 列のレコードに適用する関数として、*min*、*max*、*avg*、*count*、*sum*、または *percentile* を選択します。 パーセンタイル - 列のレコードから計算するパーセンタイルを入力します（関数が *percentile* に設定されている場合）。 列 - 計算対象の列を選択します（関数が *count* に設定されていない場合）。 エイリアス - 計算された値を識別するために使用する一意の名前を入力します。この名前はアイテムの値で使用されます。

パラメータ	説明
計算タイプ	下記の条件パラメータに設定した条件の評価方法を選択します。 **And/Or** - すべての条件を満たす必要があります。同じタイプと属性キーを持つ条件は *Or* で結合されます。 **And** - すべての条件を満たす必要があります。 **Or** - 少なくとも 1 つの条件を満たす必要があります。 カスタム式 - 例として `A and (B or C)` のように、数式で指定した条件を満たす必要があります。 2 つ以上の条件を設定した場合に使用できます。
条件	テレメトリデータ列のレコードをフィルタリングする条件を追加します。 タイプ - 条件を適用するテレメトリデータ列を選択します。 属性キー - 条件を適用するキーを入力します（タイプがキーと値のペアを持つ列に設定されている場合）。 演算子 - レコードの値と属性値の間で使用する比較方法を選択します（下記参照）。 属性値 - レコードの値と比較する値を入力します（演算子が *Exists* に設定されていない場合）。 タイプが単一値の列に設定されている場合、条件は列のレコードに適用されます。列が値の配列を保持している場合、配列内の少なくとも 1 つの値が条件を満たすと、レコードが一致します。 タイプがキーと値のペアを保持する列に設定されている場合、条件は属性キーで指定したキーの値に適用されます。列がキーと値のペアの配列を保持している場合、指定した属性キーを持つ少なくとも 1 つのペアが条件を満たすと、レコードが一致します。 単一値の列では、次の演算子を使用できます。 **Equals** - 属性値と等しいレコードに一致します（大文字と小文字を区別）。 **Does not equal** - 属性値と等しくないレコードに一致します（大文字と小文字を区別）。 **Contains** - 属性値を含むレコードに一致します（大文字と小文字を区別）。 **Does not contain** - 属性値を含まないレコードに一致します（大文字と小文字を区別）。 キーと値のペアを保持する列では、次の演算子を使用できます。 **Equals** - 属性キーが存在し、その値が属性値と等しいレコードに一致します（大文字と小文字を区別）。 **Does not equal** - 属性キーが存在し、その値が属性値と等しくないレコードに一致します（大文字と小文字を区別）。 **Exists** - 属性キーが存在するレコードに一致します（大文字と小文字を区別）。
単位 更新間隔 カスタム間隔	すべてのアイテムに共通するパラメータを参照してください。
タイムシフト	アイテムがクエリする時間範囲の終了時点を、現在時刻よりどれだけ前にするかを入力します。秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはユーザーマクロを指定できます。 各クエリは、ある時間範囲のデータを要求します。通常、この範囲は前回のクエリの時間範囲が終了した時点から始まり、タイムシフトで設定した時点で終了します。 前回のクエリの時間範囲が、現在時刻から粒度を引いた時点より後に終了していた場合、現在のクエリの時間範囲は現在時刻から粒度を引いた時点から始まるため、前回のクエリの時間範囲と重複します。 継続元となる前回のクエリがない場合、範囲はルックバック制限で設定した時点から始まります。 このアイテムのホストが別のエンティティ（例: プロキシグループの別のプロキシ）によって監視されるようになった場合、アイテムは前回のクエリが終了した時点を認識できなくなる可能性があります。これにより、値が重複することがあります。
ルックバック制限	継続元となる前回のクエリがない場合（例: アイテムの初回実行時）、または前回のクエリが更新間隔より古い場合（例: Zabbix サーバーが停止していた場合）に、アイテムがデータをクエリする範囲を現在時刻よりどれだけ前までにするかを入力します。秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはユーザーマクロを指定できます。 テレメトリデータソースが大量のデータ（例: 1 時間あたり 100 万件を超えるレコード）を保持している場合、アイテムがこのパラメータを使用すると、大きなルックバック制限によってパフォーマンスが大幅に低下する可能性があります。
粒度	クエリの全時間範囲を分割する単位の長さを入力します。秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはユーザーマクロを指定できます。

パラメータ	説明
タイムアウト ヒストリ トレンド 値のマッピング ホストインベントリフィールドに格納 説明 有効	すべてのアイテムに共通するパラメータを参照してください。

サポートされているテレメトリデータの列

このセクションでは、これらのアイテムパラメータでサポートされている列（*Columns*、*Aggregated columns*、*Conditions*）を一覧にしています。これらは、OpenTelemetry ClickHouse exporterのスキーマに従って、ClickHouse がテレメトリデータの保存に使用する列です。

Category を選択してください。

- トレース
- メトリクス: Sum
- メトリクス: Gauge
- メトリクス: Histogram
- メトリクス: Exponential histogram
- ログ

トレース

列	列	集計列	条件
Duration	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Timestamp	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

メトリクス: 合計

列	列	集計列	条件
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–

列	列	集計列	条件
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

メトリクス: Gauge

列	列	集約された列	条件
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

メトリクス: ヒストグラム

列	列	集計列	条件
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

メトリクス: 指数ヒストグラム

列	列	集約された列	条件
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

ログ

列	列	集約列	条件
Body	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

履歴とトレンド

概要

ヒストリとトレンドは、Zabbix で収集したデータを保存する 2 つの方法です。

ヒストリは収集したすべての値を保持しますが、トレンドは 1 時間ごとの平均情報を保持するため、リソースの消費が少なくなります。

履歴の保持

履歴を何日間保持するかは、以下で設定できます。

- アイテムのプロパティフォーム
- アイテムの一括更新時

- ハウスキーパータスクを設定するとき

これより古いデータは、ハウスキーパーによって削除されます。

一般的には、履歴の保持日数は可能な限り最小限にし、そうすることで大量の履歴値によってデータベースに過度な負荷をかけないようにすることを強く推奨します。

長期間の履歴を保持する代わりに、トレンドデータをより長く保持できます。
例えば、履歴は 14 日間、トレンドは 5 年間保持できます。

履歴データとトレンドデータで必要となる容量の違いについては、データベースサイジングのページを参照すると、よく把握できます。

履歴の保持期間を短くしても、グラフでは古いデータを引き続き確認できます。これは、古いデータの表示にグラフがトレンド値を使用するためです。

Attention:
履歴が「0」に設定されている場合、アイテムは依存アイテムとインベントリのみを更新します。
トリガーの評価は履歴データのみに基づいて行われるため、トリガー関数は評価されません。

Note:
履歴を保持する別の方法として、ロード可能モジュールの履歴エクスポート機能の使用を検討してください。

トレンドの保持

トレンドは、数値データ型について、毎時ごとに最小値、最大値、平均値、値の合計数を保存する、組み込みの履歴データ削減メカニズムです。

トレンドを何日間保持するかは、以下で設定できます。

- アイテムのプロパティフォーム
- アイテムの一括更新時
- ハウスキーパータスクの設定時

トレンドは通常、履歴よりもはるかに長期間保持できます。古いデータはハウスキーパーによって削除されます。

Zabbix サーバーは、データが流入するたびに、トレンドキャッシュにトレンドデータをランタイムで蓄積します。サーバーは、以下の状況で、各アイテムの前の時間のトレンドをデータベース（Web インターフェースが参照できる場所）にフラッシュします。

- サーバーがアイテムの現在の時間の最初の値を受信したとき
- 現在の時間の残りが 5 分以下で、まだアイテムの現在の時間の値がない場合
- サーバーが停止したとき

グラフでトレンドを見るには、（アイテムが頻繁に更新される場合は）少なくとも次の時間の始まりまで、（アイテムがまれに更新される場合は）最大で次の時間の終わりまで、つまり最大 2 時間待つ必要があります。

サーバーがトレンドキャッシュをフラッシュし、その時間のトレンドがすでにデータベースに存在する場合（たとえば、サーバーが時間の途中で再起動された場合）、サーバーは単純な挿入の代わりに更新ステートメントを使用する必要があります。したがって、大規模な環境では、トレンドデータの重複を避けるために、1 時間の終わりにサーバーを停止し、次の時間の始まりに起動することが望ましいです。

履歴テーブルは、トレンド生成には一切関与しません。

Attention:
トレンドを「0」に設定すると、Zabbix サーバーはトレンドを計算も保存もしません。

Note:
トレンドは、元の値と同じデータ型で計算および保存されます。その結果、符号なしデータ型値の平均値計算は丸められ、値の間隔が小さいほど結果の精度が低くなります。たとえば、アイテムの値が 0 と 1 の場合、平均値は 0.5 ではなく 0 になります。
また、サーバーの再起動により、現在の時間の符号なしデータ型の平均値計算の精度が失われる場合があります。

ユーザーパラメーター

概要

Zabbix にあらかじめ定義されていないエージェントチェックを実行したい場合があります。そのようなときに役立つのがユーザーパラメータです。

必要なデータを取得するコマンドを作成し、それをエージェント設定ファイル の ('UserParameter' 設定パラメータ) に含めることができます。

ユーザーパラメータの構文は次のとおりです。

```
UserParameter=<key>,<command>
```

ご覧のとおり、ユーザーパラメータにはキーも含まれます。このキーは、アイテムを設定するときに必要になります。参照しやすい任意のキーを入力してください（ホスト内で一意である必要があります）。

エージェントを再起動するか、エージェントのruntime control オプションを使用して、新しいパラメータを反映します。例:

```
zabbix_agentd -R userparameter_reload
```

その後、アイテムを設定する ときに、実行したいユーザーパラメータ内のコマンドを参照するキーを入力します。

ユーザーパラメータは、Zabbix エージェントによって実行されるコマンドです。item value preprocessing の手順の前に、最大 16MB のデータを返すことができます。

UNIX オペレーティングシステムでは、コマンドラインインタープリタとして **/bin/sh** が使用されます。ユーザーパラメータはエージェントチェックのタイムアウトに従います。タイムアウトに達すると、fork されたユーザーパラメータのプロセスは終了されます。

関連項目:

- ユーザーパラメータの活用方法についてのステップバイステップチュートリアル
- コマンド実行

単純なユーザーパラメータの例

単純なコマンド:

```
UserParameter=ping,echo 1
```

agent は、'ping' キーを持つ item に対して常に'1' を返します。

より複雑な例:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

agent は、MySQL サーバが生きている場合は'1' を、そうでない場合は'0' を返します。

柔軟なユーザーパラメータ

柔軟なユーザーパラメータは、キーにパラメータを指定できます。これにより、柔軟なユーザーパラメータを複数のアイテムを作成するための基盤として使用できます。

柔軟なユーザーパラメータの構文は次のとおりです。

```
UserParameter=key[*],command
```

Parameter	Description
Key	一意のアイテムキー。[*] は、このキーが角括弧内のパラメータを受け付けることを示します。 パラメータは、アイテムを設定する際に指定します。
Command	キーの値を評価するために実行するコマンド。 柔軟なユーザーパラメータのみ: コマンド内で位置参照 $1...$9 を使用して、アイテムキー内の対応するパラメータを参照できます。 Zabbix はアイテムキーの [ ] 内に含まれるパラメータを解析し、それに応じてコマンド内の $1,...,$9 を置換します。 $0 は、実行される元のコマンド（$0,...,$9 の展開前）に置き換えられます。 位置参照は、二重引用符 (") で囲まれていても単一引用符 (') で囲まれていても解釈されます。 位置参照をそのまま使用するには、ドル記号を 2 つ指定します。たとえば、awk '{print $$2}' のようにします。この場合、$$2 はコマンド実行時に実際には $2 に変換されます。

Attention:
$ 記号を含む位置参照の検索と置換は、柔軟なユーザーパラメータに対してのみ Zabbix エージェントによって行われます。
単純なユーザーパラメータでは、この参照処理はスキップされるため、$ 記号のエスケープは不要です。

Attention:
一部の文字は、デフォルトではユーザーパラメータに使用できません。使用できない文字の完全な一覧については、UnsafeUserParameters を参照してください。

例 1

とてもシンプルなもの:

```
UserParameter=ping[*],echo $1
```

ping[something] というフォーマットを持つ監視項目は無制限に定義することができる。

- ping[0] - 常に'0' を戻す
- ping[aaa] - 常に'aaa' を戻す

例 2

もっとセンスを磨こう!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

このパラメータは、MySQL データベースの可用性を監視するために使用することができます。
 ユーザー名とパスワードを渡すことができます:

```
mysql.ping[zabbix,our_password]
```

例 3

正規表現にマッチする行はファイル中に何行あるか？

```
UserParameter=wc[*],grep -c "$2" $1
```

このパラメータは、ファイル内の行数を計算するために使用することができます。

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

コマンドの結果

コマンドの戻り値は、コマンドによって生成された標準出力と標準エラーの両方です。

Attention:
テキスト（文字列、ログ、テキスト型の情報）を返すアイテムは、標準エラー出力があっても未サポートにはなりません。

戻り値は 16MB に制限されています（末尾の空白も含み、切り捨てられます）。データベースの制限も適用されます。

テキスト（文字列、ログ、テキスト型の情報）を返すユーザーパラメータも空白を返すことができます。無効な結果の場合、アイテムは未サポートになります。

Zabbix エージェントの拡張

このチュートリアルでは、ユーザーパラメータ を使用して Zabbix エージェントの機能を拡張する方法を、手順を追って説明します。

ステップ 1

必要なパラメーターを取得するためのスクリプトまたはコマンドラインを作成します。

たとえば、MySQL サーバーによって実行されたクエリの総数を取得するために、次のコマンドを記述できます。

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

実行すると、このコマンドは SQL クエリの総数を返します。

ステップ 2

Zabbix エージェント設定ファイル（zabbix_agentd.conf）にコマンドを追加します。

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions は一意の識別子です。たとえば、*queries* のように、有効なキー識別子であれば何でも指定できます。

-t フラグを使用して Zabbix エージェントでこのパラメータをテストします（ただし、root ユーザーで実行する場合、エージェントをデーモンとして起動したときとは権限が異なる可能性があることに注意してください）。

```
zabbix_agentd -t mysql.questions
```

ステップ 3

次のコマンドを実行して、設定ファイルからユーザーパラメータを再読み込みします。

```
zabbix_agentd -R userparameter_reload
```

ランタイム制御コマンドの代わりに、エージェントを再起動してもかまいません。

zabbix_get ユーティリティを使用して、パラメータをテストします。

ステップ 4

監視対象のホストに Key=mysql.questions の新しいアイテムを追加します。アイテムのタイプは *Zabbix Agent* または *Zabbix Agent (active)* のいずれかである必要があります。

返される値のタイプは、Zabbix サーバーで正しく設定する必要があることに注意してください。そうしないと、Zabbix はそれらを受け入れません。

Windows パフォーマンスカウンター

概要

perf_counter[] キーを使用すると、Windows パフォーマンスカウンターを効果的に監視できます。

例:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

または

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

このキーまたは英語のみの同等キー perf_counter_en の使用方法については、Windows 固有のアイテムキーを参照してください。

監視に使用できるパフォーマンスカウンターの完全な一覧を取得するには、次のコマンドを実行します。

```
typeperf -qx
```

また、ローレベルディスカバリを使用して、Windows パフォーマンスカウンターの複数のオブジェクトインスタンスを検出し、複数のインスタンスオブジェクトに対する perf_counter アイテムの作成を自動化することもできます。

数値表現

Windows では、オブジェクト名およびパフォーマンスカウンター名に対して数値表現（インデックス）が保持されています。Zabbix では、perf_counter、perf_counter_en アイテムキーのパラメータ、および PerfCounter、PerfCounterEn 設定パラメータとして、これらの数値表現をサポートしています。

ただし、使用する数値インデックスが特定のホスト上で正しい文字列に対応していることを保証できる場合を除き、使用は推奨されません。異なるローカライズ版の Windows を使用するさまざまなホストで動作する移植性の高いアイテムを作成する必要がある場合は、システムのロケールに関係なく英語名を使用できる perf_counter_en キーまたは PerfCounterEn 設定パラメータを使用できます。

数値に対応する値を確認するには、regedit を実行し、HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Pe にある *Counter* を探します。

レジストリエントリには、次のような情報が含まれています。

```
1
1847
2
System
4
Memory
6
% Processor Time
10
File Read Operations/sec
12
File Write Operations/sec
14
File Control Operations/sec
16
File Read Bytes/sec
```

```
18
File Write Bytes/sec
....
```

ここでは、\System\% Processor Time のようなパフォーマンスカウンターの各文字列部分に対応する番号を確認できます。

```
System → 2
% Processor Time → 6
```

これらの番号を使用して、パスを数値で表現できます。

\2\6

パフォーマンスカウンターのパラメータ

Windows パフォーマンスカウンターを監視するために、いくつかの PerfCounter パラメータを設定できます。

例えば、Zabbix エージェントの設定ファイルに次のように追加できます。

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
または
PerfCounter=UserPerfCounter2,"\4\24",30
```

このようなパラメータを設定すると、それぞれのアイテムを作成する際に、キーとして UserPerfCounter1 または UserPerfCounter2 をそのまま使用できます。

設定ファイルを変更した後は、Zabbix エージェントを再起動してください。

一括更新

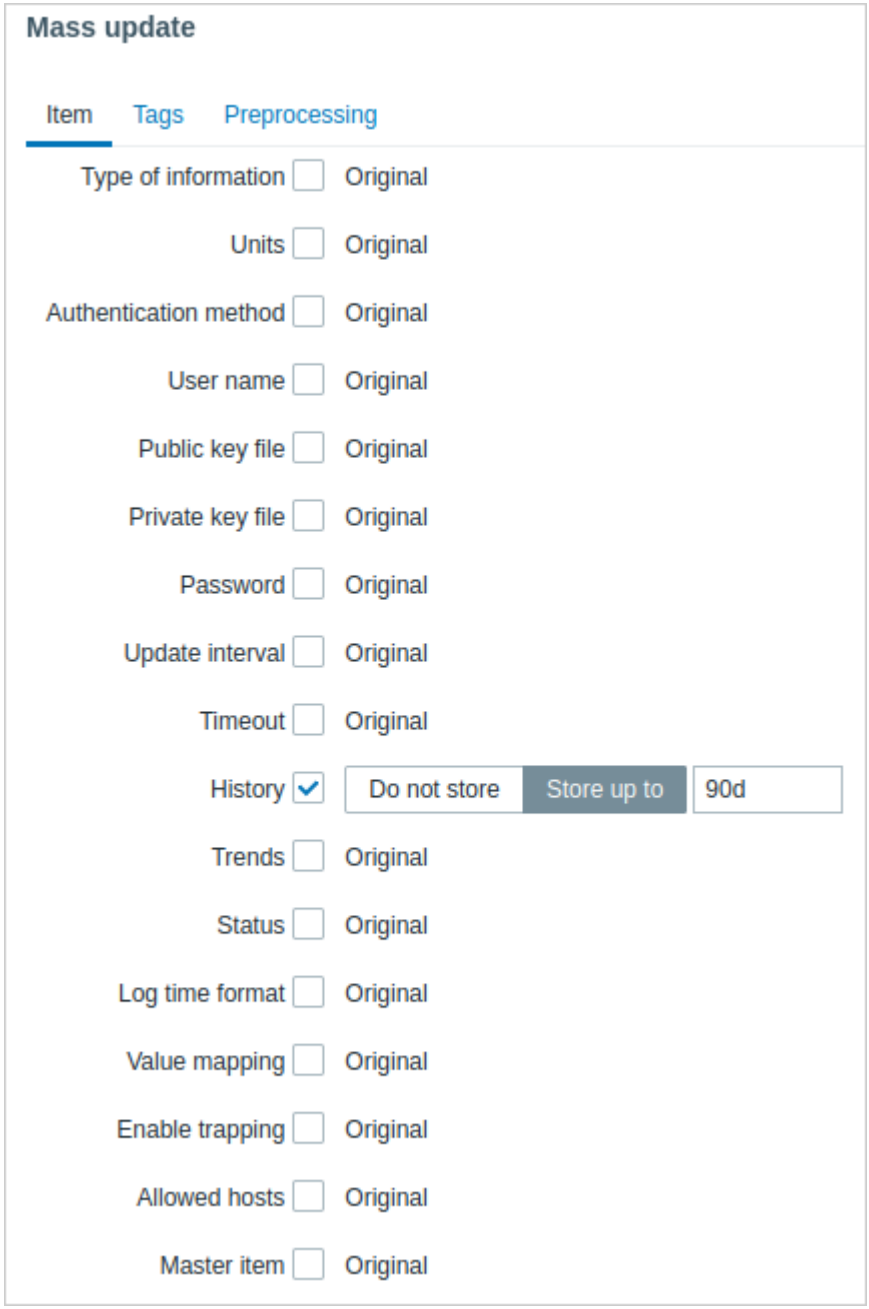
概要

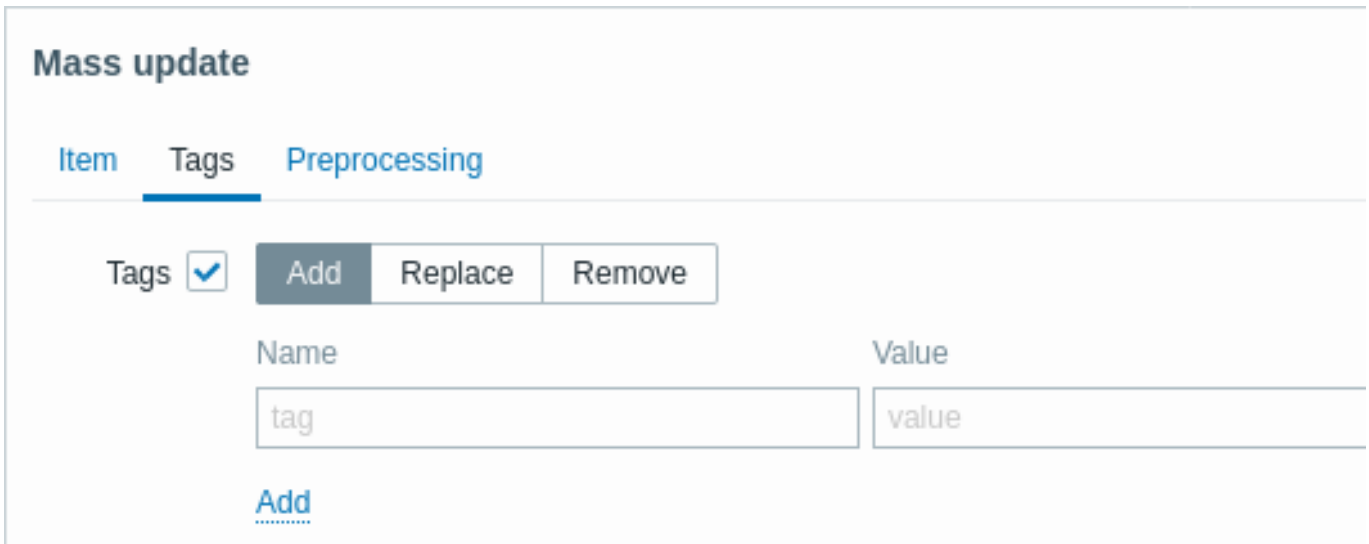
一度に複数のアイテムの属性を変更したい場合があります。個々のアイテムを開いて編集する代わりに一括更新機能を使います。

一括更新の使用

複数のアイテムを一括更新するには、次の手順を実行します。

- リストで更新するアイテムのチェックボックスを選択します
- リストの下にある 一括更新をクリックします
- 必要な属性（アイテム、タグ、または 前処理）のタブに移動します
- 更新する属性のチェックボックスを選択します
- 属性の新しい値を入力します



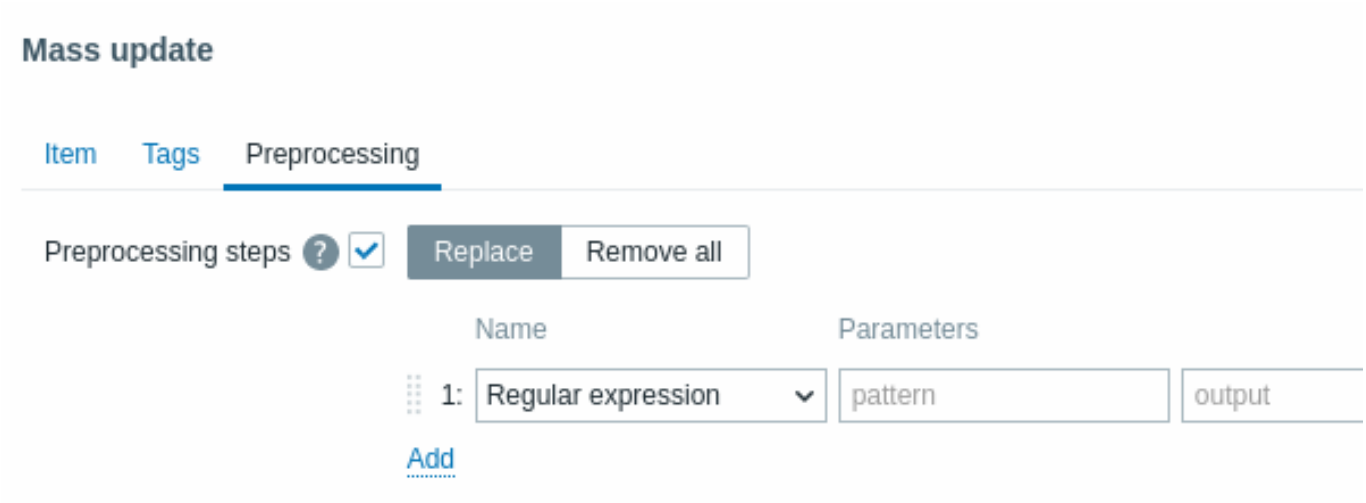


タグオプションでは、次の操作を実行できます。

- 追加 - 指定したタグをアイテムに追加します（同じ名前で値が異なるタグは「重複」とみなされないため、同じアイテムに追加できます）。
- 置換 - 指定したタグを削除し、新しい値のタグを追加します

- 削除 - 指定したタグをアイテムから削除します

ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロをタグで使用できます。



前処理ステップの一括更新を有効にするには、前処理ステップの横にあるチェックボックスを選択します。

前処理の一括更新では、次の操作を実行できます。

- 置換 - アイテムに設定されている既存の前処理ステップを、以下で指定した前処理ステップに置き換えます
- すべて削除 - アイテムからすべての前処理ステップを削除します

完了したら、更新をクリックします。

値マッピング

概要

値のマッピングを使用すると、数値/文字列の値と文字列表現の間のマッピングを使用して、受信した値をよりユーザーフレンドリーな表現に設定できます。

たとえば、アイテムの値が”0” または”1” の場合、値のマッピングを使用してこれらの値をよりユーザーフレンドリーな方法で表現できます。

- 0 → 利用不可
- 1 → 利用可能

データバックアップタイプの値のマッピングは、次のように設定できます。

- F → フル
- D → 差分
- I → 増分

電圧の値の範囲は、次のように設定できます。

- <=209 → 低
- 210-230 → OK
- >=231 → 高

値のマッピングは、Zabbix フロントエンドおよびメディアタイプによって送信される通知で使用されます。

Attention:
受信した値を設定された表現に置き換える処理は、Zabbix フロントエンドとサーバーの両方で実行されます。ただし、サーバーが置換を処理するのは、次の場合のみです。

- ホストインベントリを入力する場合
- サポートされているマクロ {ITEM.VALUE}、{ITEM.LASTVALUE}、{EVENT.OPDATA}、{EVENT.CAUSE.OPDATA} を展開する場合

値のマッピングはテンプレートまたはホストで設定します。一度設定すると、該当するテンプレートまたはホスト内のすべてのアイテムで利用できます。アイテムを設定する際に、値のマッピングパラメータで事前に設定した値のマッピング名を指定します。

Note:
値のマッピングの継承はありません。ホストやテンプレートは、リンクされたテンプレートから値のマッピングを継承しません。ホスト上のテンプレートアイテムは、テンプレートで設定された値のマッピングを引き続き使用します。

Note:
値のマッピングは、数値（整数）、数値（浮動小数点）、文字列の情報タイプを持つアイテムで使用できます。

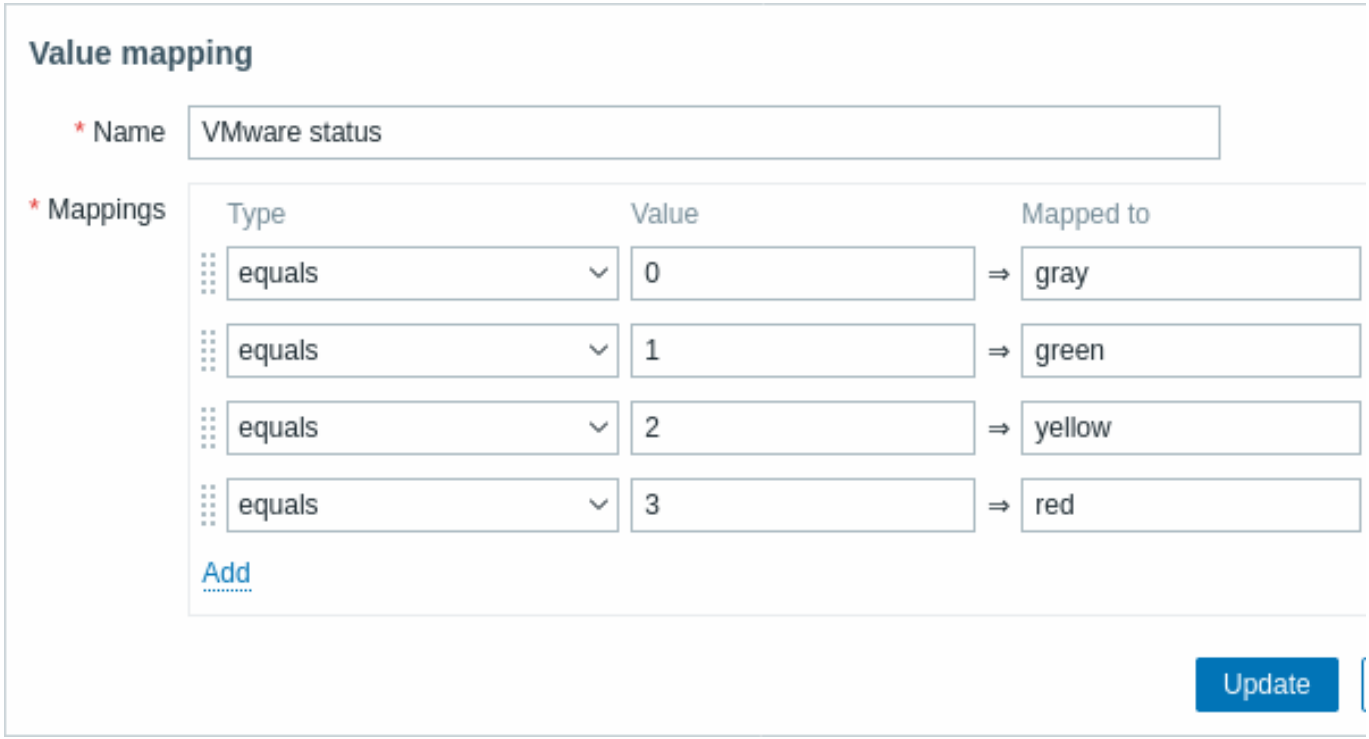
値のマッピングは、該当するテンプレートまたはホストとともにエクスポート/インポートされます。また、ホストおよびテンプレートの一括更新フォームを使用して一括更新することもできます。

設定

値のマッピングを設定するには、次の手順に従います。

1. ホストまたはテンプレートの設定フォームを開きます。

2. 値のマッピングタブで、追加をクリックして新しい値のマッピングを追加するか、既存のマッピング名をクリックして編集します。



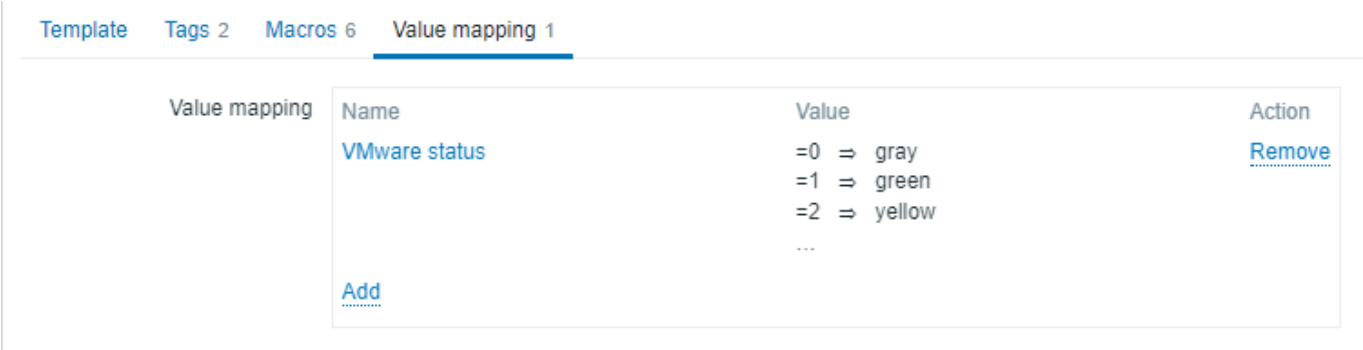
値のマッピングのパラメータ:

パラメータ	説明
名前	値のマッピングセットの一意の名前。
マッピング	数値または文字列の値を文字列表現にマッピングする個別のルール。 マッピングはルールの順序に従って適用されます。ルールはドラッグして並べ替えることができます。

パラメータ	説明
タイプ	マッピングのタイプ: **equals** - 一致する値をマッピングします。 **is greater than or equals** - 同じ値またはより大きい値をマッピングします。 **is less than or equals** - 同じ値またはより小さい値をマッピングします。 **in range** - 範囲内の値をマッピングします。範囲は <number1>-<number2> または <number> で表します。複数の範囲にも対応しています（例: 1-10,101-110,201）。 **regexp** - 正規表現に一致する値をマッピングします（グローバル正規表現には対応していません）。 **default** - 特定のマッピングが設定されていない、その他すべての値をマッピングします。 範囲のマッピングでは、数値の値タイプ（*is greater than or equals*、*is less than or equals*、*in range*）のみがサポートされます。
値	受信値（マッピングタイプに応じて、範囲または正規表現を含めることができます）。
マッピング先	受信値に対する文字列表現（最大 64 文字）。

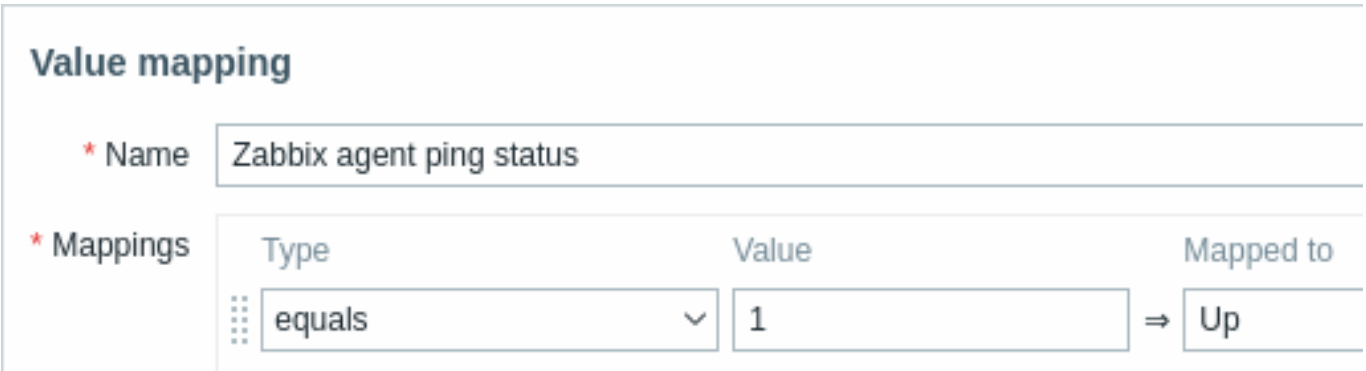
必須入力フィールドには赤いアスタリスクが付いています。

一覧で値のマッピングを表示する場合、最初の 3 つのマッピングのみが表示され、その他のマッピングが存在することを示す 3 つのドットが表示されます。



値のマッピングの例

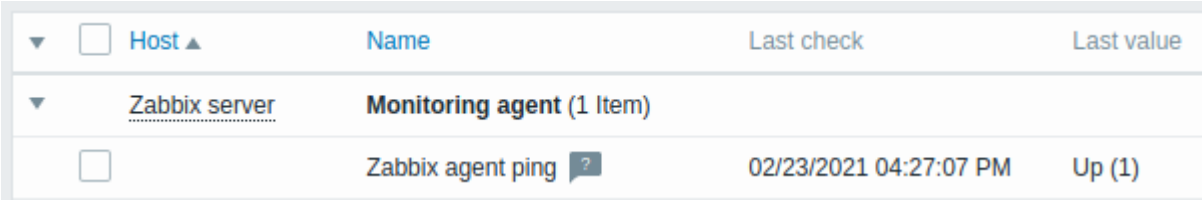
あらかじめ定義されたエージェントアイテムの 1 つである *Zabbix agent ping* は、値を表示するためにテンプレートレベルの値のマッピング「Zabbix agent ping status」を使用します。



アイテム設定フォームでは、*Value mapping* フィールドにこの値のマッピングへの参照があります。

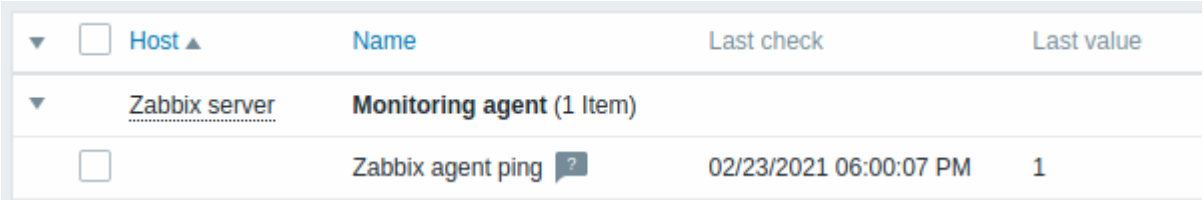


このマッピングは、監視 > 最新データセクションで「Up」（括弧内に生の値）を表示するために使用されます。



Note:
最新データセクションでは、表示される値は 20 文字に短縮されます。値のマッピングを使用する場合、この短縮はマッピングされた値には適用されず、生の値（括弧内に表示）にのみ適用されます。

あらかじめ定義された値のマッピングがない場合は、「1」としか表示されないため、理解しにくいことがあります。



キュー

概要

キューには、更新を待っているアイテムが表示されます。キューはデータの論理的な表現にすぎません。Zabbix には IPC キューやその他のキュー機構はありません。

プロキシによって監視されているアイテムもキューに含まれます。これらはプロキシの履歴データ更新期間中、キューに入っているとカウントされます。

キューには、スケジュールされた更新時間を持つアイテムのみが表示されます。つまり、以下のアイテムタイプはキューから除外されます。

- log、logrt、およびイベントログのアクティブ Zabbix エージェントアイテム
- SNMP トラップアイテム
- トラッパーアイテム
- Web 監視アイテム
- 従属アイテム

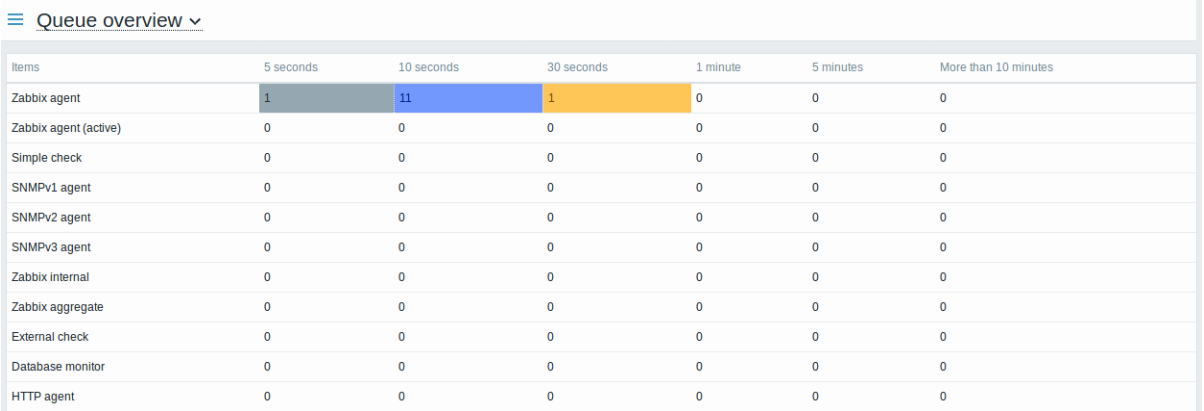
キューによって表示される統計情報は、Zabbix サーバーのパフォーマンスを示す良い指標です。

キューは、JSON プロトコルを使用して Zabbix サーバーから直接取得されます。この情報は、Zabbix サーバーが稼働している場合にのみ利用できます。

Attention:
接続の問題やエージェントの不具合によりアイテムインターフェースが利用できなくなった場合、アイテムはキューにカウントされません。

キューの確認

キューを確認するには、管理 > キュー > キューの概要に移動します。



ここに表示されている状態は一般的に「正常」であるため、サーバーは問題なく動作していると考えられます。

キューには、最大 30 秒間待機しているアイテムがいくつか表示されています。これらがどのアイテムなのかを確認できると便利です。

確認するには、タイトルのドロップダウンから キューの詳細を選択します。遅延しているアイテムの一覧が表示されます。

Queue details

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

これらの詳細から、アイテムが遅延している理由を特定できる可能性があります。

遅延しているアイテムが 1 つまたは 2 つであれば、特に心配する必要はありません。数秒後に更新される可能性があります。ただし、多数のアイテムが長時間遅延している場合は、より深刻な障害が発生している可能性があります。

関連項目: スケジューリング間隔を使用する場合のタイムゾーンの調整。

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

キューアイテム

特殊な内部アイテム `zabbix[queue,<from>,<to>]` を使用して、Zabbix のキューの状態を監視できます。設定した時間だけ遅延しているアイテムの数が返されます。詳細については、内部アイテムを参照してください。

value cache

概要

トリガー式、計算アイテム、一部のマクロの計算を高速化するために、Zabbix server ではバリューキャッシュオプションがサポートされています。

このインメモリキャッシュを利用すると、直接データベースへ SQL を発行せずに過去のデータへアクセスすることができます。過去の値がキャッシュに存在しない場合、不足分をデータベースから取得し、キャッシュの更新を行います。

アイテムの値は、次のいずれかの状態になるまでバリューキャッシュに残ります:

- アイテムが削除された場合 (キャッシュされた値は、次の構成同期後に削除されます)
- アイテムの値が、トリガーもしくは計算アイテム式で指定された時間またはカウントの範囲外になった場合 (キャッシュされた値は、新しい値を受信した時に削除されます)
- トリガー/計算アイテム式で指定された時間またはカウント範囲が変更され、計算に必要なデータが少なくなった場合 (不要なキャッシュされた値は 24 時間後に削除されます)

Note:
バリューキャッシュの状態は、サーバーのランタイムコントロールでオプション diaginfo（または diaginfo=valuecache）を実行し、診断情報のバリューキャッシュセクションを調べることで確認できます。これは、誤って構成されたトリガーまたは計算アイテムを特定する場合に役立ちます。

value cache 機能を有効にするには、Zabbix のサーバー設定でオプションの ValueCacheSize パラメータを指定します。

value cache の監視には、2 つの内部アイテムがサポートされています: zabbix[vcache,buffer,<mode>] および zabbix[vcache,cache,<parameter>]。詳細については、内部アイテムを参照してください。

今すぐ実行

概要

今すぐ実行オプションを使用すると、パッシブチェックを即座に実行できます。

Zabbix でのアイテム値の収集は、設定された更新間隔に従います。一部のアイテム（例：ローレベルディスカバリルール）は間隔が長く、すぐに新しい値を取得する必要がある場合があります。たとえば、検出可能なリソースの変更を検出する場合などです。

今すぐ実行オプションは、以下のアイテムタイプでサポートされています。

- Zabbix エージェント（パッシブ）
- シンプルチェック
- SNMP エージェント（v1/v2/v3）
- Zabbix 内部
- 外部チェック
- データベースモニター
- HTTP エージェント
- IPMI エージェント
- SSH エージェント
- TELNET エージェント
- JMX エージェント
- 計算
- 従属アイテム
- スクリプト
- ブラウザー

アイテムが「従属アイテム」タイプの場合、そのマスターアイテムも上記のいずれかのタイプである必要があります。

設定

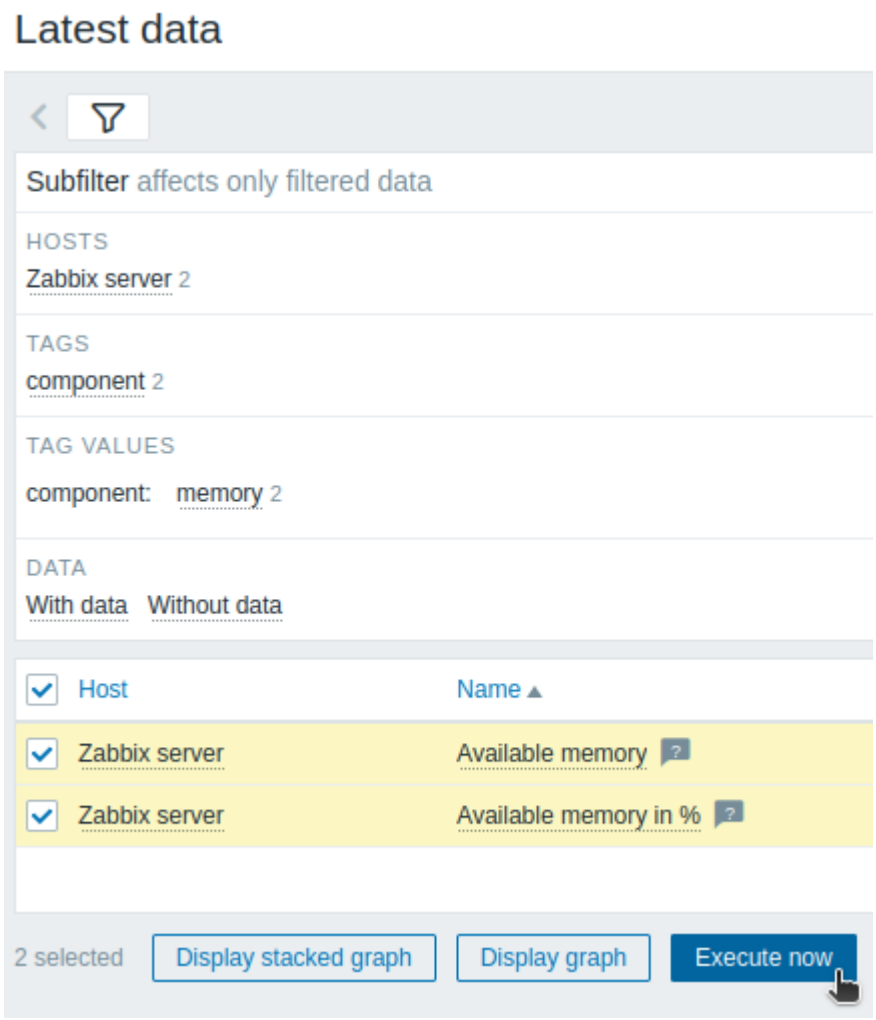
以下からパッシブチェックを実行できます。

- 監視 > 最新データ
- データ収集 > ホスト > アイテム またはディスカバリ
- アイテムメニュー
- アイテムまたは低レベルディスカバリルールを編集しているとき（それぞれの設定フォーム内）

Attention:
チェックを実行するには、設定キャッシュ内にそのチェックが存在している必要があります。詳細はCacheUpdateFrequencyも参照してください。チェック実行前にキャッシュは更新されないため、アイテムまたは低レベルディスカバリルールの設定に対する最近の変更は反映されません。作成中または新規作成したアイテムや LLD ルールをテストするには、代わりにテスト オプションを使用できます。

たとえば、監視 > 最新データ でパッシブチェックを実行するには、次の手順を行います。

1. アイテムを選択します。複数のアイテムに対して同時にチェックを実行できます。
2. 今すぐ実行をクリックします。



Note:
最新データでは、ユーザーは、読み書き権限 を持つホスト、または 読み取り権限と、ロールに対して 読み取り専用ホストで「今すぐ実行」を呼び出すアクション が有効になっているホスト上のアイテムに対して 今すぐ実行を実行できます。

エージェントチェックの制限

概要

アイテムチェック、リモートコマンド、またはスクリプトの実行時に、Zabbix エージェントまたは agent 2 が使用を許可または禁止されるアイテムキーを制御できます。

そのためには、許可/拒否ルールを定義するために、次のエージェント設定 パラメータを使用します。

- AllowKey=<pattern>
- DenyKey=<pattern>
- AllowKeyRegexp=<pattern>
- DenyKeyRegexp=<pattern>

<pattern> には単一のアイテムキーを含める必要があります。
 AllowKey と DenyKey では、<pattern> はワイルドカード (*) をサポートします。ワイルドカードは、その位置にある任意の数の任意の文字に一致し、アイテムキーまたはパラメータの一致に使用できます（例: vfs.file.*[*]）。
 AllowKeyRegexp と DenyKeyRegexp では、<pattern> は正規表現 をサポートします。正規表現パターンは、アイテムキーおよびそのパラメータ文字列の一致に使用できます。正しいエスケープ の例も参照してください。
 Zabbix エージェントと Zabbix agent 2 では、サポートされる正規表現の構文が異なる場合があることに注意してください。

Attention:
セキュリティを向上させるため、AllowKey と DenyKey ではワイルドカードではなく正確なアイテムキーを使用することを推奨します。また、AllowKeyRegexp と DenyKeyRegexp では、できるだけ具体的な正規表現パターンを使用することも推奨します。詳細は、許可/拒否ルールの保護 を参照してください。

他のエージェント設定パラメータとは異なり、AllowKey、DenyKey、AllowKeyRegexp、DenyKeyRegexp パラメータは無制限に指定できます。

重要な注意事項

- すべてのsystem.runアイテムはデフォルトで無効です（DenyKey と DenyKeyRegexp が空の場合でも）。これは、DenyKey=system.run[*] または DenyKeyRegexp=^system\.run\[.*\]$ が最後のルールとして設定されている場合と同じです。そのため、他の system.run アイテムを明示的に拒否しなくても、特定の system.run アイテムのみを許可できます。
- 可能であれば、AllowKey/AllowKeyRegexp を使用して必要なアイテムのみを許可し、それ以外はすべて拒否してください。一部のキーは、パストラバーサルを通じて意図しないファイルを読み取るために悪用される可能性があります（例: vfs.file.contents["../../../../etc/passwd"]）。また、新しい Zabbix エージェントのバージョンでは、DenyKey/DenyKeyRegexp ルールでカバーされていないキーが追加される場合があります。
- AllowKey、DenyKey、AllowKeyRegexp、および DenyKeyRegexp の設定は、HostnameItem、HostMetadataItem、または HostInterfaceItem のエージェントパラメータには影響しません。
- 拒否されたアイテムは、何のヒントやエラーメッセージもなくサポート対象外になります。たとえば、次のとおりです。
  - Zabbix エージェントの--print (-p)コマンドラインパラメータでは、拒否されたアイテムキーは表示されません。
  - Zabbix エージェントの--test (-t)コマンドラインパラメータでは、拒否されたアイテムキーに対して"Unsupported item key." が返されます。
  - ログ記録が有効化されている場合（LogRemoteCommands=1）、Zabbix エージェントのログファイルには拒否されたリモートコマンドは記録されません。

許可/拒否ルールの順序

AllowKey、DenyKey、AllowKeyRegexp、または DenyKeyRegexp のルールは無制限に指定できますが、順序が重要です。

- ルールは上から下へ、1 つずつ評価されます。
- アイテムキーがあるルールに一致すると、そのアイテムは許可または拒否され、ルールの評価は停止します。

たとえば、vfs.file.contents[/etc/passwd] を評価する場合、ルールは次のように処理されます。

```
AllowKey=vfs.file.contents[/tmp/app.log]    # アイテムキーのパターンは一致せず、エージェントは次のルールへ進みます。
AllowKey=vfs.file.contents[/etc/passwd]     # アイテムキーのパターンが一致し、エージェントはアイテムチェックを許可して
DenyKey=vfs.file.*[*]                       # 評価はすでに停止しているため、エージェントはこのルールを無視します。
```

次のルール順では、アイテムチェックは拒否されます。

```
DenyKey=vfs.file.*[*]                       # アイテムキーのパターンが一致し、エージェントはアイテムチェックを拒否して
AllowKey=vfs.file.contents[/etc/passwd]     # 評価はすでに停止しているため、エージェントはこのルールを無視します。
AllowKey=vfs.file.contents[/tmp/app.log]    # 評価はすでに停止しているため、エージェントはこのルールを無視します。
```

例

以下の例では、AllowKey、DenyKey、AllowKeyRegexp、DenyKeyRegexp の一般的な設定パターンを示します。

特定のチェックとコマンドを許可する

2 つのvfs.file アイテムチェックと 2 つのsystem.run コマンドのみを許可します。

AllowKey と DenyKey を使用する場合:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

AllowKeyRegexp と DenyKeyRegexp を使用する場合:

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
DenyKey=system.run[*] または DenyKeyRegexp=^system\.run\[.*\]$ を設定する必要はありません。その他すべての system.run コマンドはデフォルトで拒否されるためです。

スクリプトの許可

Zabbix エージェントが、利用可能なすべての方法でホスト上のスクリプトを実行できるようにします。

- Web インターフェースまたは API から実行できるグローバルスクリプト（この方法では常に system.run[myscript.sh] キーを使用します）
- アクションのオペレーション からのリモートコマンド（この方法では常に system.run[myscript.sh,nowait] キーを使用します）
- スクリプトを含むsystem.run Zabbix エージェントアイテム。たとえば、次のようになります。
  - system.run[myscript.sh]
  - system.run[myscript.sh,wait]
  - system.run[myscript.sh,nowait]

```
AllowKey=system.run[myscript.sh,*]
```

または

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

wait/nowait パラメーターを制御するには、別のルールを設定する必要があります。
たとえば、system.run[myscript.sh,wait] アイテムのみを許可し、他の方法を除外できます。

```
AllowKey=system.run[myscript.sh,wait]
```

または

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Allow/deny ルールの保護

これらの例では、過度に許可範囲の広い AllowKey/AllowKeyRegexp または DenyKey/DenyKeyRegexp ルールを安全にする方法を示します。

次のルールを考えます。

AllowKey と DenyKey を使用する場合:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

AllowKeyRegexp と DenyKeyRegexp を使用する場合:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
Windows では、AllowKey または DenyKey ではパス内のスペースをキャレット (^) で、AllowKeyRegexp または DenyKeyRegexp では \s でエスケープする必要があります。

これらのルールにはワイルドカード (*)（正規表現ベースのパターンでは .*）が含まれており、悪用される可能性があります。

- test.bat スクリプトは、意図しないものを含む任意の引数で実行できます。
- vfs.file.* (^vfs\.file\..*$) パターンは、パラメータの有無にかかわらずアイテムキーに一致します。ただし、すべてのvfs.file アイテムにはパラメータが必要です。
- system.cpu.load[*] (^system\.cpu\.load\[.*\]$) パターンは、パラメータ付きのアイテムキーにのみ一致します。ただし、system.cpu.load アイテムにはパラメータは不要です。

これらのルールを安全にするには、test.bat の実行を特定の引数に対してのみ明示的に許可し、正しいアイテムキーのパターンを拒否します。例えば、次のようにします。

AllowKey と DenyKey を使用する場合:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

AllowKeyRegexp と DenyKeyRegexp を使用する場合:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^system\.cpu\.load(\[.*\])?$
```

次のコマンドを実行してルールをテストできます。これらは ZBX_NOTSUPPORTED を返します。

```
cd "C:\Program Files\Zabbix Agent 2"
zabbix_agent2.exe -t system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat debug"]
zabbix_agent2.exe -t vfs.file.size["C:\ProgramData\MyApp\config.ini"]
zabbix_agent2.exe -t vfs.file.contents["C:\Windows\System32\drivers\etc\hosts"]
zabbix_agent2.exe -t system.cpu.load
zabbix_agent2.exe -t system.cpu.load[all,avg1]
```

パターンの例

以下の表は、アイテムキーのパターンがどのように一致するかを示しています。

- キーがパターンに一致するのは、*Matches* 列の すべての条件を満たす場合のみです。
- パラメータは必ず角括弧で完全に囲む必要があります（例: `vfs.file.contents[*` や `vfs.file.contents*utf8]` は無効なパターンです）。
- これらの例はパターン一致のみを示しており、実際のアイテムにはパラメータが必要です。

AllowKey と DenyKey の場合:

Pattern	Matches	Examples
*	パラメータの有無にかかわらず、任意のキー	
vfs.file.*	キーが vfs.file. で始まる パラメータの有無にかかわらず	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	キーが vfs. で始まる キーが .contents で終わる パラメータなし	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
vfs.file.*[*]	キーが vfs.file. で始まる 任意のパラメータ、または空のパラメータ	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
vfs.file.contents	キーが vfs.file.contents である パラメータなし	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	キーが vfs.file.contents[] である 空のパラメータ	Matches: vfs.file.contents[] Does not match: vfs.file.contents
vfs.file.contents[*]	キーが vfs.file.contents である 任意のパラメータ、または空のパラメータ	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents

Pattern	Matches	Examples
vfs.file.contents[/etc/passwd,*]	キーが vfs.file.contents である 1 つ目のパラメータが /etc/passwd 2 つ目のパラメータは任意、または空	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	キーが vfs.file.contents である 任意のパラメータで、少なくとも 1 つに passwd を含む	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	キーが vfs.file.contents である 1 つ目のパラメータに passwd を含む 2 つ目のパラメータは任意、または空	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	キーが vfs.file.contents である 1 つ目のパラメータが /etc/passwd 2 つ目のパラメータが utf8	Matches: vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

AllowKeyRegexp と DenyKeyRegexp の場合:

Pattern	Matches	Examples
^.*$	パラメータの有無にかかわらず、任意のキー	
^vfs\.file\..*$	キーが vfs.file. で始まる パラメータの有無にかかわらず	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
^vfs\..*\.contents$	キーが vfs. で始まる キーが .contents で終わる パラメータなし	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	キーが vfs.file. で始まる 任意のパラメータ、または空のパラメータ	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] Does not match: vfs.file.get.custom
^vfs\.file\.contents$	キーが vfs.file.contents である パラメータなし	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[\]$	キーが vfs.file.contents[] である 空のパラメータ	Matches: vfs.file.contents[] Does not match: vfs.file.contents

Pattern	Matches	Examples
^vfs\.file\.contents\[.*\]$	キーが vfs.file.contents である任意のパラメータ、または空のパラメータ	Matches: vfs.file.contents[/path/to/file] Does not match: vfs.file.contents
^vfs\.file\.contents\[/etc/passwd,?[^,]*\]$	キーが vfs.file.contents である 1 つ目のパラメータが /etc/passwd 2 つ目のパラメータは任意、または空	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[.*passwd.*(,[^,]*)?\]$	キーが vfs.file.contents である 任意のパラメータで、少なくとも 1 つに passwd を含む	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
^vfs\.file\.contents\[.*passwd.*,?[^,]*\]$	キーが vfs.file.contents である 1 つ目のパラメータに passwd を含む 2 つ目のパラメータは任意、または空	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
^vfs\.file\.contents\[/etc/passwd,?(utf8|windows-1252)?\]$	キーが vfs.file.contents である 1 つ目のパラメータが /etc/passwd 2 つ目のパラメータが utf8 または windows-1252	Matches: vfs.file.contents[/etc/passwd,utf8] vfs.file.contents[/etc/passwd,windows-1252] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

トリガーによる障害検出

Zabbix における障害検出は、トリガーに基づいています。

トリガーは障害条件を定義し、障害が発生した場合に Zabbix が反応できるようにします。

障害検出の全体的な流れは次のとおりです。

1. データを収集するアイテム があることを確認します。
2. このアイテムに対するトリガーを定義します。トリガーがなければ、Zabbix はデータを収集するだけで、それに対して反応することはありません。
3. 検出された障害を確認します。
   - *Monitoring* > *Problems* で確認する
   - *Problems* ウィジェット で確認する
4. 通知を受信し、リモートコマンドを実行します（定義されている場合）。

公式の Zabbix テンプレートには、あらかじめ定義されたトリガーが含まれていることに注意してください。
テンプレートを適用してLinux を監視する ガイドを参照してください。

「トリガー」とは？

トリガーは、トリガー式によって問題の条件を記述します。

実際に問題の条件が発生した場合、それがトリガーが「発火」した状態（Zabbix で問題が作成される）です。

問題の条件は、最新値、平均値、検出された文字列、その他多くの関数に基づくことができます。バイナリまたは JSON のデータ型で値を返すアイテムにはトリガーを作成できません。

単純なトリガー式は、括弧内で指定されたデータ（通常はデータアイテムと時間ウィンドウ）に対して関数（例えば、最大値の場合は **max()**）を計算し、この計算結果をしきい値と比較します。

例えば:

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

このトリガーは、5 分間にわたって測定されたディスク空き容量が常に 10GB 未満であった場合に発火します。

トリガーには以下の状態があります:

トリガーの状態	説明	Zabbix での動作
OK	トリガー式が「0」(または FALSE) と評価された	問題は作成されない (既存の場合) 問題が解決される
Problem	トリガー式が「1」(または TRUE) と評価された	問題が作成される

何らかの理由でトリガー式の評価ができない場合もあります。不明な式の状態を参照してください。

その他のリソース:

- トリガーの例
- トリガーの設定
- ベストプラクティス

計算頻度

トリガーは、Zabbix サーバーが式の一部である新しい値を受信するたびに再計算されます。新しい値が受信されると、式に含まれる各関数が再計算されます (新しい値を受信した関数だけでなく)。

さらに、式に以下のいずれかが含まれている場合、トリガーは 30 秒ごとに (history syncer によって) 再計算されます。

- 日付と時刻関数
- nodata()関数

トレンド関数のみを含むトリガーは、式内の最小の時間間隔ごとに 1 回評価されます。多くのトリガー関数はアイテム値の履歴データを使用しますが、トレンド関数はトレンドデータを使用します。

トリガーの例

このページは、トリガーの例をまとめたものです。

例はユースケースごとに分類されています。

- 利用不可のホスト
- 利用不可のプロキシ
- 利用不可の SMTP クラスター
- 到達不能なホスト
- 予期しない再起動
- 重要なファイルの変更
- DNS クエリの変更
- 異なるホスト間で一致しないソフトウェア
- 時刻が同期されていないクロック
- 古いエージェント
- フラッピングするインターフェース
- 高い受信トラフィック
- ディスク容量不足
- ディスク容量不足 (動的しきい値)
- 高い CPU 負荷
- CPU 負荷の比較
- 長期的な CPU 負荷の比較
- 特殊文字を含む文字列

利用できないホストの検出

```
max(/host/zabbix[host,agent,available],5m)=0
```

このトリガーは、ホスト上の Zabbix エージェントが 5 分間利用できない場合に発生します。
 関数: max
 データアイテム: zabbix[host,agent,available]

代替案:

```
nodata(/host/agent.ping,5m)=1
```

このトリガーは、Zabbix エージェントから 5 分間データが受信されていない場合に発生します。
 関数: nodata
 データアイテム: agent.ping

利用できないプロキシの検出

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

このトリガーは、Zabbix プロキシのデータが Zabbix サーバーの時刻より 1 分遅れている場合に発生します。
 関数: fuzzytime
 データアイテム: zabbix[proxy,{$PROXY_NAME},lastaccess]

利用できない SMTP クラスタの検出

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

両方の SMTP サーバーがダウンしている場合、このトリガーが発生します。
 関数: last
 データアイテム: net.tcp.service

到達不能なホストの検出

```
count(/host/icmpping,30m,,"0")>5
```

このトリガーは、過去 30 分間に 5 回以上 ping でホストに到達できなかった場合に発生します。
 関数: count
 データアイテム: icmpping

予期しない再起動の検出

```
change(/host/system.uptime)<0
```

システムの稼働時間の値に負の変化があった場合 (再起動を示す) にこのトリガーが発生します。
 関数: change
 データアイテム: system.uptime

重要なファイルの変更を検出する

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

このトリガーは、/etc/passwd が変更された場合に発生します。直前の/etc/passwd のチェックサムと最新のチェックサムが異なる場合に式が真になります。/etc/passwd、/etc/inetd.conf、/kernel などの重要なファイルの変更を監視するために、同様の式が役立ちます。
 関数: last
 データアイテム: vfs.file.cksum

DNS クエリの変更を検出する

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

2 番目のオペランドを囲む引用符に注意してください。

このトリガーは、クエリ結果が通常返される値と等しくない場合に発生します。

```
example.com           MX       0 mail.example.com
```

関数: last
 データアイテム: net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1]、マクロは次のように定義されています。

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

異なるホストで一致しないソフトウェアを検出する

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

このトリガーは、異なるホストで Ubuntu のバージョンが異なる場合に発生します。ここでのオペランドは文字列を返す関数であることに注意してください。
 関数: last
 データアイテム: vfs.file.contents

同期されていない時計を検出する

```
fuzzytime(/host/system.localtime,10s)=0
```

クライアントのローカル時刻と Zabbix サーバー時刻の差が 10 秒を超える場合にトリガーが発生します。
 関数: fuzzytime
 データアイテム: system.localtime

system.localtime は、Zabbix エージェントではパッシブチェック として設定する必要があることに注意してください。Zabbix agent 2 では、アクティブチェックとして設定することもできます。

古いエージェントの検出

```
find(/host/agent.version,,"like","beta")=1
```

このトリガーは、Zabbix エージェントがベータバージョンの場合に発生します。Zabbix エージェントをアップグレードする必要があります。
 関数: find
 データアイテム: agent.version

flapping インターフェースの検出

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

このトリガーは、eth0 の運用状態 (up/down/unknown) が 1 時間に 5 回以上変化した場合に発生します。
 関数: changecount
 データアイテム: vfs.file.contents

高い受信トラフィックを検出する

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

このトリガーは、過去 5 分間に eth0 で受信したバイト数が常に 100 キロバイトを超えていた場合に発生します。誰かが大きなファイルをダウンロードしている可能性があります。
 関数: min
 データアイテム: net.if.in[eth0,bytes]

ディスク容量不足の検知

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

空きディスク容量が 5 分間連続して 10GB 未満の場合にトリガーが発生します。

リカバリエクスプレッションも定義できます:

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

空きディスク容量が 10 分間連続して 40GB を超えた場合にのみ、問題が解決されます。
 関数: min
 データアイテム: vfs.fs.size

ディスク容量不足の検出（動的しきい値）

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

空き容量（割り当て単位）が 10% 未満になるとトリガーが発生します。さまざまなサイズの検出されたストレージに適用できる適応トリガーしきい値を取得するために、別のアイテムの値が使用されていることに注意してください。関数: last

高い CPU 負荷の検出

```
last(/host/system.cpu.load[all,avg1])>5
```

平均プロセッサ負荷が 1 分間 5 を超えている場合にトリガーが発生します。

バリエーション:

```
min(/host/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

```
min(/host/system.cpu.load[all,avg1],5m)>2 and not (dayofweek()=7 and time()>230000) and not (dayofweek()=1
```

このようなトリガーは 5 分間のデータを分析し、CPU 負荷が 2 未満にならない場合にのみトリガーします。さらに、これらのトリガーは以下の場合に発生します:

- 夜間のみ (00:00-06:00)
- 週の切り替え時の 2 時間 (日曜日 23:00～月曜日 01:00) を除く任意の時間

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

このトリガーは、3 つのホストのうち少なくとも 2 つでプロセッサ負荷が高すぎる場合に発生します。
 関数: last, min, dayofweek, time
 データアイテム: system.cpu.load

CPU 負荷の比較

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

このトリガーは、今日の平均負荷が昨日の同じ時間帯の平均負荷（now-1d の時間シフトを使用）を 2 倍以上上回った場合に発生します。
 関数: avg
 データアイテム: system.cpu.load

長期的な CPU 負荷の比較

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

このトリガーは、ホストの CPU 負荷が先月より 10% 以上増加した場合に発生します
 関数: trendavg
 データアイテム: system.cpu.load

また、トリガー設定のイベント名フィールドを使用して、意味のあるアラートメッセージを作成することもできます。例えば、以下のようなメッセージを受け取るには

"Exchange サーバーの負荷が 7 月に 24% 増加しました (0.69)、6 月 (0.56) と比較して"

イベント名は次のように定義する必要があります:

```
{HOST.HOST} サーバーの負荷が {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.load,1M:now/M-
```

この種の問題には、トリガー設定で手動クローズを許可することも有用です。

特殊文字を含む文字列の検出

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

/tmp/hello の内容が {$HELLO_MACRO} で定義された文字列と等しい場合、トリガーが発生します。

```
{$HELLO_MACRO} = \" //hello ?\"
```

または、直接文字列と比較することもできます。

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

特殊文字（\\と"）が直接比較される場合、どのようにエスケープされているかに注意してください。

関数: last
 データアイテム: vfs.file.contents

トリガーの設定

このページでは、Zabbix Web インターフェースでトリガーを設定する方法について説明します。

公式の Zabbix テンプレートには、あらかじめ定義されたトリガーが含まれていることに注意してください。
Linux の監視 にテンプレートを適用する手順については、ガイドを参照してください。
設定済みのトリガーを一括更新 することもできます。

その他のリソース:

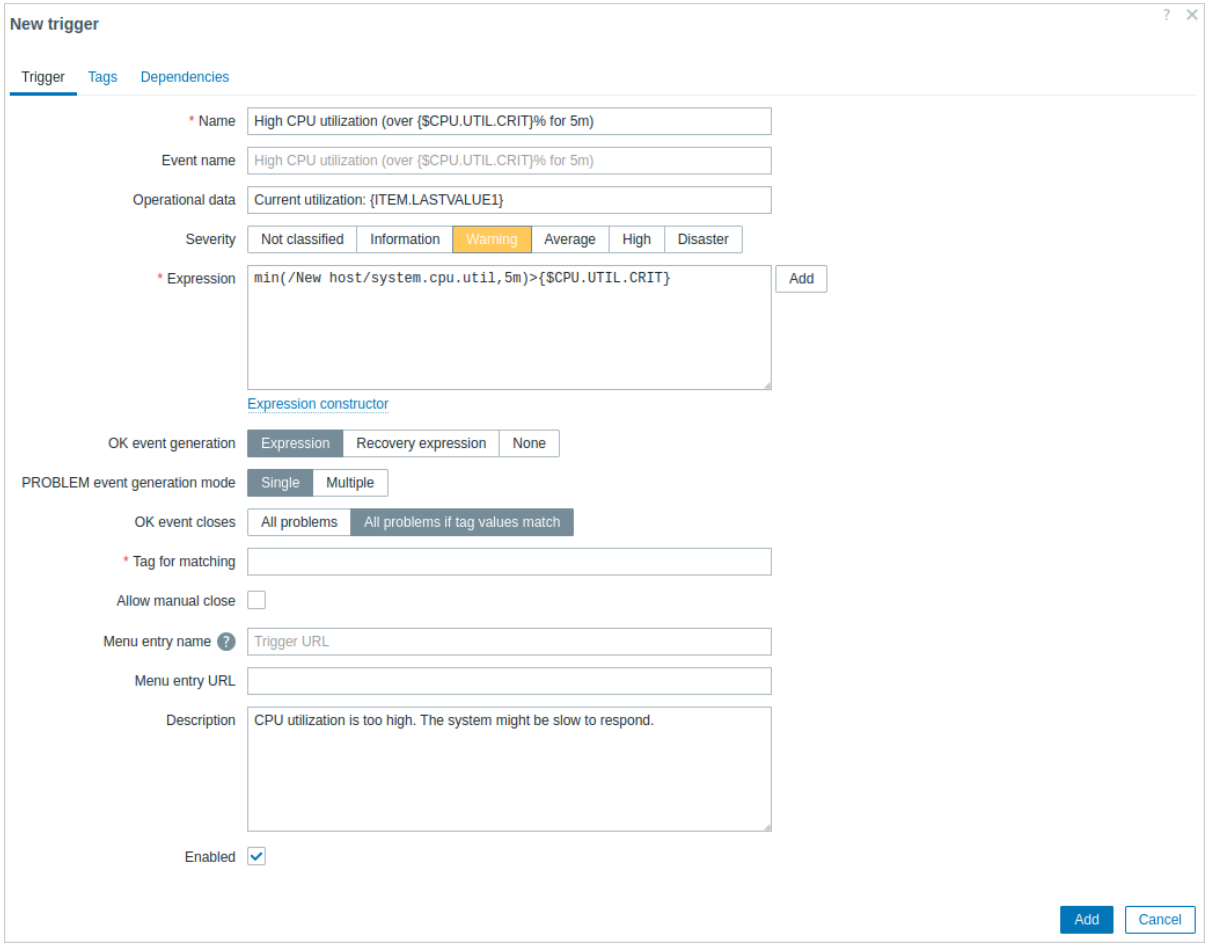
- トリガーによる障害検出（概要）
- トリガーの例（ユースケース別）
- トリガーの式と関数（構文の詳細）
- ベストプラクティス

設定

トリガーを設定するには、次の手順を実行します。

1. データ収集 > ホストに移動します。
2. ホストの行にある トリガーをクリックします。
3. 右側の トリガーの作成をクリックします（既存のトリガーを編集する場合は、トリガー名をクリックします）。
4. フォームにトリガーのパラメータを入力します。

トリガータブには、トリガーの基本属性がすべて含まれています。

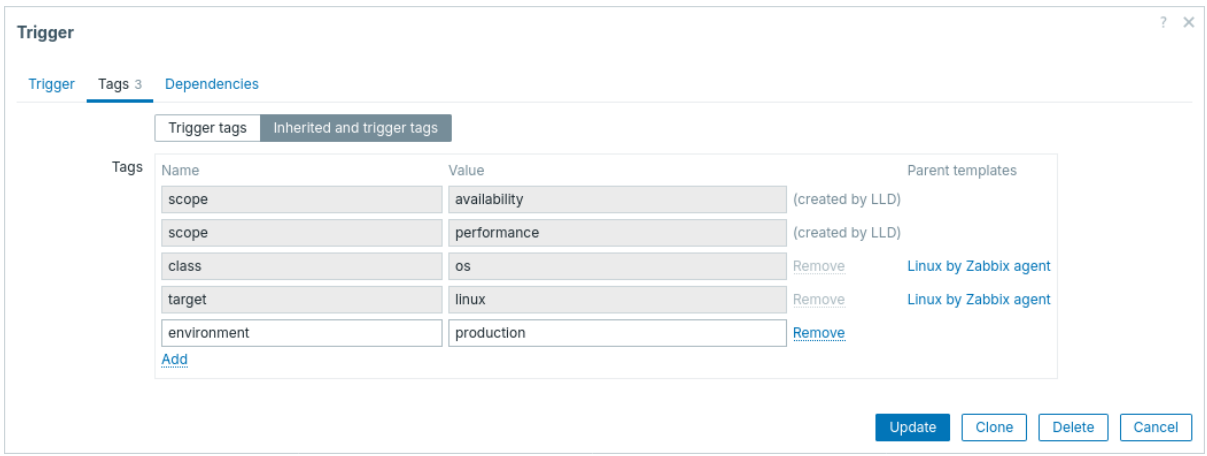


必須入力フィールドには赤いアスタリスクが付いています。

Parameter	Description
Name	トリガー名。 サポートされるマクロは次のとおりです: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {ITEM.LOG.*}, および {$MACRO} ユーザーマクロ。 **$1, $2...$9** マクロは、式の 1 番目、2 番目...9 番目の定数を参照するために使用できます。 注: $1-$9 マクロは、比較的単純で明快な式で定数を参照する場合に正しく解決されます。たとえば、名前”Processor load above $1 on {HOST.NAME}” は、式が last(/New host/system.cpu.load[percpu,avg1])>5 の場合、自動的に”Processor load above 5 on New host” に変わります。
Event name	定義すると、この名前が障害イベント名の作成に使用されます。デフォルトでは、イベント名はトリガー名と同じです。 イベント名は、障害データを含む意味のあるアラートを作成するために使用できます（例を参照）。 トリガー名と同じマクロセットに加えて、{TIME}, {TIMESTAMP}, および {?EXPRESSION} 式マクロがサポートされます。
Operational data	マクロを含む文字列を入力すると、監視 > 障害 に動的なリアルタイムデータを表示できます。あるいは、このフィールドを空欄にすると、トリガー式に含まれるすべてのアイテムの最新値が表示されます。 トリガー名と同じマクロセットがサポートされ、動的に解決できます。たとえば: {ITEM.VALUE<1-9>} は、トリガーの状態が変更された時点（障害の発生、復旧、手動クローズ、または相関によるクローズ）のアイテム値に解決されます。 {ITEM.LASTVALUE<1-9>} は最新のアイテム値に解決されます。 障害を手動でクローズしても新しい値は生成されないため、どちらのマクロも障害発生時点の値を表示し続けます。また、最新値が *Max history display period* より古い場合、どちらのマクロも *UNKNOWN* に解決されることに注意してください（*Administration > General* を参照）。
Severity	ボタンをクリックして、必要なトリガー深刻度を設定します。

Parameter	Description
Expression	障害の条件を定義するための論理式です。 時間サフィックスおよびメモリサイズサフィックスがサポートされます。 式に含まれるすべての条件が満たされると、つまり式が TRUE と評価されると障害が発生します。 *Recovery expression* に追加の復旧条件が指定されていない限り、式が FALSE と評価されるとすぐに障害は復旧します。
OK event generation	OK イベント生成オプション: **Expression** - 障害イベントと同じ式に基づいて OK イベントが生成されます; **Recovery expression** - 障害式が FALSE、かつ復旧式が TRUE と評価された場合に OK イベントが生成されます; **None** - この場合、トリガーは自動的に OK 状態に戻りません。
Recovery expression	元の障害式がすでに FALSE と評価された後、障害が復旧する前に満たす必要がある追加条件を定義する論理式（任意）です。 Recovery expression は、トリガーのヒステリシスに有用です。障害式がまだ TRUE の場合、復旧式だけで障害を復旧させることはできません。 このフィールドは、*OK event generation* で'Recovery expression' が選択されている場合にのみ使用できます。
PROBLEM event generation mode	障害イベントの生成モード: **Single** - トリガーが初めて'Problem' 状態になったときに 1 つのイベントが生成されます; **Multiple** - トリガーの'Problem' 評価ごとにイベントが生成されます。
OK event closes	OK イベントが閉じる対象を選択します: **All problems** - このトリガーのすべての障害; **All problems if tag values match** - イベントタグ値が一致するトリガー障害のみ。
Tag for matching	イベント相関に使用するイベントタグ名を入力します。 このフィールドは、*OK event closes* プロパティで'All problems if tag values match' が選択されている場合に表示され、この場合は必須です。
Allow manual close	このトリガーによって生成された障害イベントの手動クローズを許可する場合にチェックします。 障害イベントの確認時に手動クローズが可能です。
Menu entry name	空でない場合、ここで入力した名前（最大 64 文字）が、*Menu entry URL* パラメータで指定されたトリガー URL のラベルとして、いくつかの Web インターフェース上の場所で使用されます。空の場合は、既定名の *Trigger URL* が使用されます。 トリガー名と同じマクロセットに加えて、{EVENT.ID}, {HOST.ID}, {TRIGGER.ID} がサポートされます。
Menu entry URL	空でない場合、ここで入力した URL（最大 2048 文字）は、イベントメニューへのリンクとして、いくつかの Web インターフェース上の場所で利用できます。たとえば、監視 > 障害 で障害名をクリックしたときや、ダッシュボードの 障害 ウィジェットで利用できます。 トリガー名と同じマクロセットに加えて、{EVENT.ID}, {HOST.ID}, {TRIGGER.ID} がサポートされます。注: 秘密値を持つユーザーマクロは URL 内で解決されません。
Description	このトリガーに関する追加情報を提供するためのテキストフィールドです。特定の障害を修復するための手順、担当者の連絡先などを含めることができます。 トリガー名と同じマクロセットがサポートされます。
Enabled	必要に応じて、このチェックボックスをオフにするとトリガーを無効化できます。 無効化されたトリガーの障害は Web インターフェースに表示されなくなりますが、削除されるわけではありません。

タグタブでは、トリガーレベルのタグを定義できます。
このトリガーのすべての障害には、ここで入力した値がタグとして付与されます。



さらに、*Inherited and trigger tags* オプションを使用すると、トリガーがそのテンプレート由来の場合に、テンプレートレベルで定義されたタグを表示できます。
同じタグを持つテンプレートが複数ある場合、それらのタグは 1 回だけ表示され、テンプレート名はカンマで区切られます。
トリガーはホストレベルのタグを「継承」して表示することはありません。

Parameter	Description
Name/Value	トリガーイベントを識別するためのカスタムタグを設定します。 タグはタグ名と値のペアです。名前のみを使用することも、値と組み合わせることもできます。トリガーには、同じ名前で値が異なる複数のタグを持たせることができます。 ユーザーマクロ、コンテキスト付きユーザーマクロ、低レベルディスカバリマクロ、および {{ITEM.VALUE}}, {{ITEM.LASTVALUE}} を含むマクロ関数がイベントタグでサポートされます。低レベルディスカバリマクロはマクロコンテキスト内で使用できます。 {TRIGGER.ID} マクロはトリガータグ値でサポートされます。これは、トリガープロトタイプから作成されたトリガーを識別したり、たとえばメンテナンス中にこれらのトリガーからの障害を抑制したりするのに役立ちます。 展開後の値の合計長が 255 を超える場合、255 文字に切り詰められます。 イベントタグでサポートされるすべてのマクロを参照してください。 イベントタグはイベント相関、アクション条件に使用でき、監視 > 障害または 障害ウィジェットでも表示されます。

依存関係タブには、トリガーのすべての依存関係が含まれています。

新しい依存関係を追加するには、追加をクリックします。

Note:
既存のトリガーを開いて *Clone* ボタンをクリックし、別名で保存することでもトリガーを設定できます。

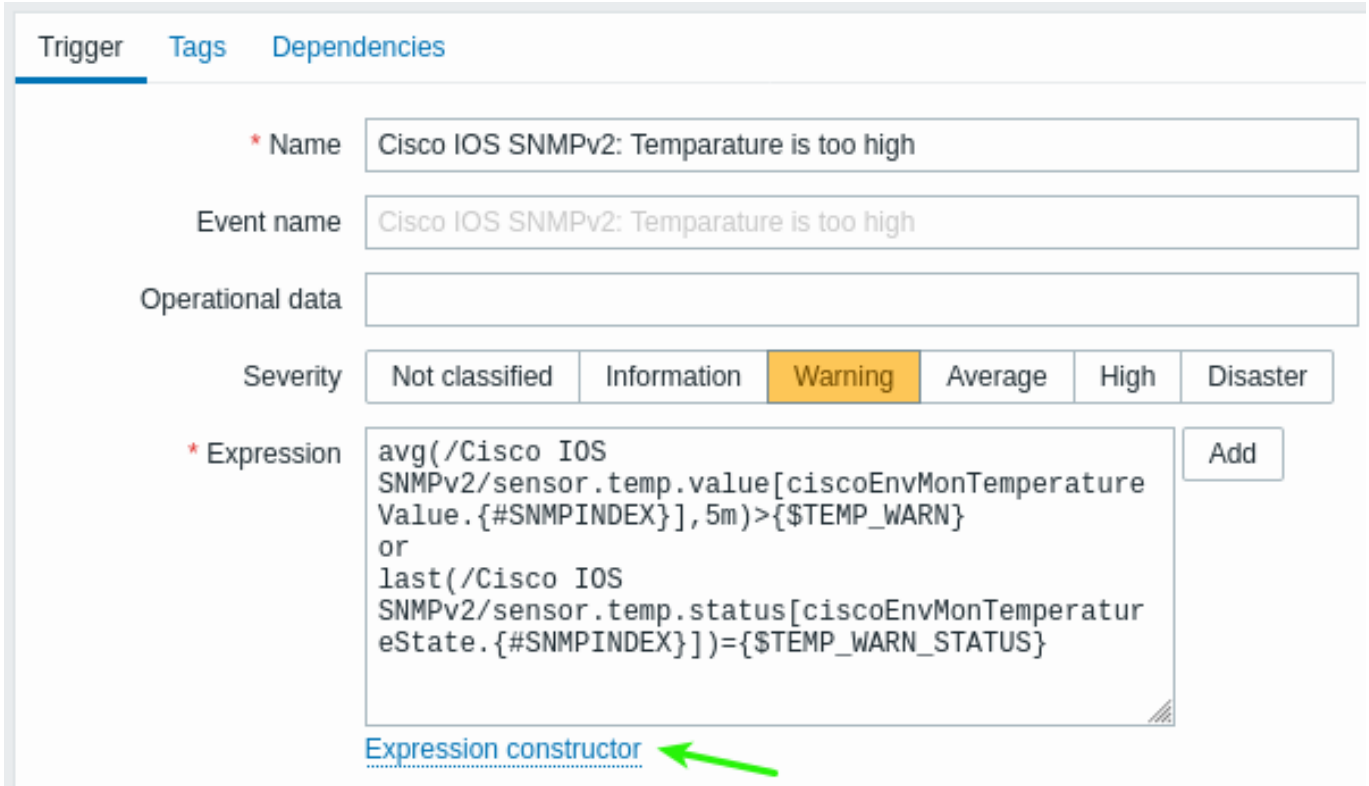
トリガー式のテスト

受信した値に応じて、式の結果がどうなるかをテストできます。

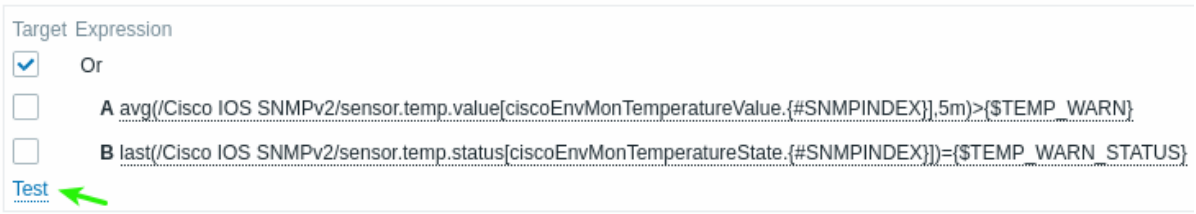
公式テンプレートの次の式を例として使用します。

```
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}],5m)>{$TEMP_WARN}
or
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.{#SNMPINDEX}])={$TEMP_WARN_STATUS}
```

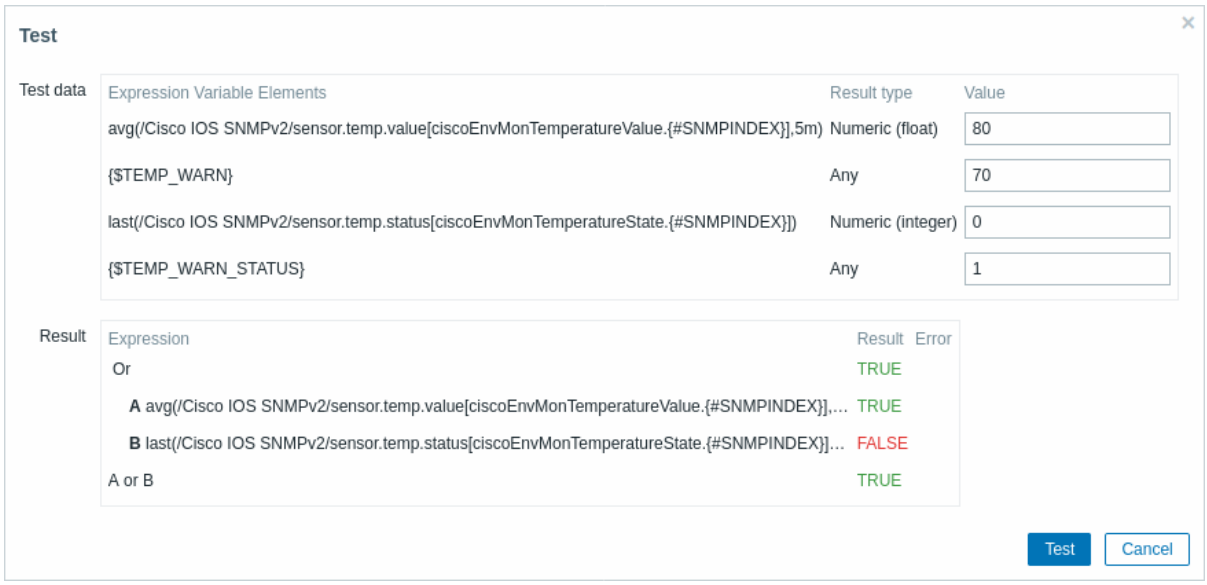
式をテストするには、式フィールドの下にある *Expression constructor* をクリックします。



Expression constructor には、個々の式がすべて一覧表示されます。
テストウィンドウを開くには、式一覧の下にある *Test* をクリックします。



テストウィンドウでは、サンプル値（この例では'80'、'70'、'0'、'1'）を入力し、*Test* ボタンをクリックすることで式の結果を確認できます。



個々の式の結果と、式全体の結果の両方を確認できます。

"TRUE" は、指定した式が正しいことを意味します。
このケース A では、"80" は指定値 {$TEMP_WARN} より大きく、この例では"70" です。
予想どおり、"TRUE" の結果が表示されます。

"FALSE" は、指定した式が正しくないことを意味します。
このケース B では、{$TEMP_WARN_STATUS} の"1" は、指定値であるこの例の"0" と等しくなければなりません。
予想どおり、"FALSE" の結果が表示されます。

選択されている式の種類は"OR" です。
指定された条件の少なくとも 1 つ（この場合は A または B）が TRUE であれば、全体の結果も TRUE になります。
つまり、現在の値が警告値を超えており、障害が発生していることを意味します。

トリガー式と関数

概要

このページでは、トリガー式の構文と評価の詳細について説明します。

単純な式の構文は次のとおりです。

```
function(/host/item,time_period)<operator><constant>
```

この式では、演算子の左側の最初のオペランドは、括弧内にパラメータ（通常はデータアイテムと期間）を持つ関数です。

関数は、設定された期間中に受信したデータを分析するために使用され、計算値が得られます。

この値は、演算子を使用して 2 番目のオペランドと比較されます。この例では 2 番目のオペランドは定数ですが、別の関数にすることもできます。

例：

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

このトリガーは、過去 5 分間に eth0 で受信したバイト数が常に 100 キロバイトを超えていた場合に発生します。この場合、式は真となり、障害が作成されます。

トリガー式は非常に柔軟です。より複雑な式では、複数の関数、演算子、定数を組み合わせることができます。

関連情報:

- トリガーの例
- トリガーによる障害検知（一般的な紹介）

関数

関数を使用すると、収集した値を分析できます。たとえば、平均値の計算や特定の文字列の検索などが可能です。

詳細を表示するには、該当する関数グループをクリックしてください。

関数グループ		関数
集計関数		avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
	Foreach 関数	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
ビット演算関数		bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
日付と時刻の関数		date, dayofmonth, dayofweek, now, time
履歴関数		change, changecount, count, countunique, find, first, firstclock, fuzzytime, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath
トレンド関数		baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
数学関数		abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate
演算子関数		between, in
予測関数		forecast, timeleft

関数グループ	関数
文字列関数	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

特に記載がない限り、これらの関数は以下でサポートされています。

- トリガーの数式
- 計算アイテム
- 数式マクロ

Foreach 関数は、集計計算でのみサポートされています。

通常、関数は比較用の数値を返します。文字列を返す場合は、**=** 演算子および **<>** 演算子を使用して比較できます（異なるホスト上の一致しないソフトウェアを検出する例を参照）。

関数パラメータ

関数パラメータでは、次の項目を指定できます。

- ホストのアイテム履歴を参照する関数のアイテムキー（/host/key として指定）
- 期間（および関数固有のその他のパラメータ）
- その他の式

アイテムのキー

参照されるアイテムは、サポートされている状態でなければなりません（ただし、**nodata()** 関数はサポートされていないアイテムにも計算されます）。

最初のパラメータでホスト名を省略する（つまり、function(//key,parameter,...) のように記述する）ことは、特定のコンテキストでのみサポートされています。

- 計算アイテムの式内
- 式マクロ内（以下の場所で使用可能）：
  - イベント名フィールド
  - グラフ名
  - ”ホスト”および”トリガー”のマップ要素のラベル

これらのコンテキストでは、{HOST.HOST}マクロも使用できます。イベント名フィールドおよび”トリガー”マップ要素の場合、トリガー式内の特定のアイテムを参照するために {HOST.HOST<1-9>} を使用できます。これらのコンテキストでホスト名を省略したり、{HOST.HOST} に置き換えたりした場合、参照はトリガー式内の最初のアイテムまたはグラフの最初のアイテムを指します。これらのサポートされているコンテキスト以外でトリガー式内のホスト名を省略すると、エラーになります。イベント名マクロでのダブルスラッシュの使用例については、長期間の CPU 負荷の比較の例を参照してください。

時間期間

関数固有のパラメータはアイテムキーの後に配置し、アイテムキーとはコンマで区切ります。

数値関数の多くは、パラメータとして時間期間を受け取ります。これにより、対象とする間隔を指定できます。時間期間（30s、10m、1h）または値の範囲（#5 - 最新の 5 つの値）として指定できます。

時間期間を示すには、秒または時間のサフィックスを使用できます。ハッシュ記号を前に付けると、パラメータの意味が変わります。

Expression	Description
sum(/host/key,**10m**)	過去 10 分間の値の合計。
sum(/host/key,**#10**)	最新 10 個の値の合計。

ハッシュ記号付きのパラメータは、**last** 関数では異なる意味を持ちます。これは N 番目に新しい値を示します。値が新しい順に 30、70、20、60、50 の場合:

- last(/host/key,#2) は「70」を返します

- last(/host/key,#5) は「50」を返します

時間期間は「現在」までの期間として測定されます。ここでいう「現在」とは、トリガーの最新の再計算時刻です（計算頻度を参照）。「現在」はサーバーの現在時刻ではありません。

時間期間は、次のいずれかを指定します。

- 「現在 - 時間期間」から「現在」までのすべての値（時間シフトを適用した場合は、「現在 - 時間シフト - 時間期間」から「現在」まで）。
- 過去から「現在」までの、指定された数の値。

指定した評価期間にデータが存在しない場合、この関数を使用するトリガーまたは計算アイテムはサポート対象外になります。

次の点に注意してください。

- トリガーで（データ履歴を参照する）単一の関数のみを使用している場合、「現在」は常に最後に受信した値になります。たとえば、最後の値を 1 時間前に受信した場合、時間期間は 1 時間前の最新の値までとして扱われます。
- 新しいトリガーは、最初の値を受信するとすぐに計算されます（履歴関数）。日付と時刻およびnodata()関数の場合は、30 秒以内に計算されます。そのため、トリガーが作成されてから、設定した時間期間（たとえば 1 時間）がまだ経過していなくても、トリガーは計算されます。また、時間範囲をたとえば最新 10 個の値に設定している場合でも、最初の値を受信した後にトリガーは計算されます。

時間シフト

関数パラメータとして、時間または値の件数とともにオプションの時間シフトを指定できます。このパラメータを使用すると、過去の期間のデータを参照できます。

時間シフトは現在時刻を指定する now で始まり、その後に時間単位を N 個加算または減算する +N<time unit> または -N<time unit> が続きます。

例えば、avg(/host/key,1h:now-1d) は、1 日前の 1 時間の平均値を返します。

Attention:
月（M）および年（y）で指定する時間シフトは、トレンド関数でのみサポートされます。その他の関数では、秒（s）、分（m）、時間（h）、日（d）、週（w）がサポートされます。

絶対時間期間での時間シフト

時間シフトパラメータでは、絶対時間期間を指定できます。例えば、1 日の場合は午前 0 時から午前 0 時まで、1 週間の場合は月曜日から日曜日まで、1 か月の場合は月初から月末までを指定できます。

絶対時間期間の時間シフトは、現在時刻を指定する now で始まり、その後に任意の数の時間操作が続きます。/<time unit> は時間単位の開始と終了を定義します。例えば、1 日の場合は午前 0 時から午前 0 時までです。+N<time unit> または -N<time unit> は、時間単位を N 個加算または減算します。

時間シフトの値は 0 以上にできますが、時間期間の最小値は 1 であることに注意してください。

Parameter	Description
1d:now/d	昨日
1d:now/d+1d	今日
2d:now/d+1d	過去 2 日間
1w:now/w	先週
1w:now/w+1w	今週

その他の式

関数のパラメータには、次の構文のように他の式を含めることができます。

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

関数がアイテムの履歴を参照する場合、他の式は使用できませんのでご注意ください。たとえば、次の構文は許可されません:

~~`min(/host/key,#5*10)`~~

関数パラメータとしての他のトリガー式は、トリガー内の非履歴関数に限定されますが、この制限は計算アイテムには適用されません。

演算子

(実行優先度の降順で）トリガーでサポートされている演算子は次のとおりです。

優先度	演算子	定義	不明な値に関する注意	オペランドを float 型に強制変換 [1]
1	**-**	単項マイナス	**-**Unknown → Unknown	はい
2	**not**	論理 NOT	**not** Unknown → Unknown	はい
3	*	乗算	0 * Unknown → Unknown (はい、Unknown、0 以外 - 算術演算で Unknown を失わないため) 1.2 * Unknown → Unknown	はい
	/	除算	Unknown / 0 → エラー Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	はい
4	+	算術加算	1.2 + Unknown → Unknown	はい
	-	算術減算	1.2 - Unknown → Unknown	はい
5	<	より小さい。演算子は次のように定義されます: A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	はい
	<=	以下。演算子は次のように定義されます: A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	はい
	>	より大きい。演算子は次のように定義されます: A>B ⇔ (A>B+0.000001)		はい
	>=	以上。演算子は次のように定義されます: A>=B ⇔ (A≥B-0.000001)		はい
6	=	等しい。演算子は次のように定義されます: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		いいえ [1]
	<>	等しくない。演算子は次のように定義されます: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		いいえ [1]
7	**and**	論理 AND	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	はい
8	**or**	論理 OR	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	はい

[1] 文字列オペランドは、以下の場合に数値型にキャストされます:

- 他のオペランドが数値型の場合
- オペランドに **=** または **<>** 以外の演算子が使用されている場合

(キャストに失敗した場合は、数値型オペランドが文字列オペランドにキャストされ、両方のオペランドが文字列として比較されます。)

not、**and**、**or** 演算子は大文字と小文字を区別し、小文字で記述する必要があります。また、スペースまたは括弧で囲む必要があります。

単項**-**および **not** を除くすべての演算子は、左結合です。単項**-**および **not** は非結合（つまり、**--1** や **not not 1** の代わりに**-(-1)** や **not (not 1)** を使用する必要があります)。

評価結果:

- **<**、**<=**、**>**、**>=**、**=**、**<>** 演算子は、指定された関係が真の場合はトリガー式で「1」、偽の場合は「0」を返します。少なくとも 1 つのオペランドが Unknown の場合、結果は Unknown になります。
- **and** は既知のオペランドの場合、両方のオペランドが「0」と等しくない場合は「1」を返し、それ以外の場合は「0」を返します。不明なオペランドの場合、**and** は一方のオペランドが「0」と等しい場合のみ「0」を返し、それ以外の場合は「Unknown」を返します。
- **or** は既知のオペランドの場合、いずれかのオペランドが「0」と等しくない場合は「1」を返し、それ以外の場合は「0」を返します。不明なオペランドの場合、**or** は一方のオペランドが「0」と等しくない場合のみ「1」を返し、それ以外の場合は「Unknown」を返します。
- 論理否定演算子 **not** の既知のオペランドに対する結果は、オペランドの値が「0」と等しくない場合は「0」、オペランドの値が「0」と等しい場合は「1」です。不明なオペランドの場合、**not** は「Unknown」を返します。

不明な式の状態

トリガー式に不明なオペランドが現れるのは、次の場合です。

- サポートされていないアイテムが使用されている
- サポートされているアイテムに対する関数の評価でエラーが発生する

この場合、トリガー式は通常 Unknown と評価されます（評価できないためです)。

不明なトリガーについて通知を受け取る ことができます。

例外

不明なオペランドがあっても、場合によってはトリガー式を既知の結果（Problem/OK）として評価できます。

- `nodata()` 関数は、参照先のアイテムがサポートされているかどうかに関係なく評価されます。
- AND/OR を含む式は、次の 2 つの場合に既知の結果として評価できます。
  - ケース **1**: `"1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ..."` は既知の結果（'1' または"Problem"）として評価される
  - ケース **2**: `"0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ..."` は既知の結果（'0' または"OK"）として評価される
- サポートされているアイテムに対する関数の評価でエラーが発生した場合、その関数値は Unknown となり、以降の式の評価では不明なオペランドとして扱われます。

不明なオペランドが「消える」のは、上記のような論理式の場合のみです。算術式では、不明なオペランドは常に Unknown につながります（0 での除算を除く)。

Attention:
不明な式の状態は、トリガーの状態（Problem/OK）を変更しません。そのため、トリガーの状態が"Problem" だった場合（ケース 1 を参照)、既知の部分が解決されて（'1' が'0' になって）も、式は Unknown と評価されるため、障害状態のまま維持されます。これはトリガーの状態を変更しないためです。

複数のサポートされていないアイテムを含むトリガー式が Unknown と評価された場合、Web インターフェースのエラーメッセージには、最後に評価されたサポートされていないアイテムが表示されます。

値のキャッシュ

トリガーの評価に必要な値は、Zabbix サーバーによってキャッシュされます。このため、サーバーの再起動後しばらくの間、トリガーの評価によってデータベースの負荷が高くなります。アイテムの履歴値が削除された場合（手動またはハウスキーパーによる)、値のキャッシュはクリアされません。そのため、サーバーはトリガー関数で定義された期間より古い値になるか、サーバーが再起動されるまでキャッシュされた値を使用します。

Note:
キャッシュに最近のデータがなく、関数でクエリ期間が定義されていない場合、Zabbix はデフォルトで過去 1 週間までさかのぼってデータベースから履歴値を取得します。

集計関数

特に明記されていない限り、ここに一覧されているすべての関数は次でサポートされています。

- トリガー式
- 計算アイテムの式
- 式マクロ

集計関数は、次のいずれかで使用できます。

- アイテムの履歴。たとえば、`min(/host/key,1h)`
- foreach 関数 を唯一のパラメータとして使用。たとえば、`min(last_foreach(/*/key))`（計算アイテムの式でのみ使用可能。トリガーで使用するには、計算アイテムをそのキーで参照します）

関数は追加情報なしで一覧されています。関数をクリックすると、詳細を表示できます。

Function	Description
avg	定義された評価期間内におけるアイテムの平均値。
bucket_percentile	ヒストグラムのバケットからパーセンタイルを計算します。
count	foreach 関数によって返される配列内の値の数。
histogram_quantile	ヒストグラムのバケットから φ-分位数を計算します。
item_count	フィルター条件に一致する、設定内に存在するアイテムの数。
kurtosis	定義された評価期間内に収集された値の「裾の厚さ」。
mad	定義された評価期間内に収集された値の中央値絶対偏差。
max	定義された評価期間内におけるアイテムの最大値。
min	定義された評価期間内におけるアイテムの最小値。
skewness	定義された評価期間内に収集された値の分布の非対称性。
stddevpop	定義された評価期間内に収集された値の母標準偏差。
stddevsamp	定義された評価期間内に収集された値の標本標準偏差。
sum	定義された評価期間内に収集された値の合計。
sumofsquares	定義された評価期間内に収集された値の二乗和。
varpop	定義された評価期間内に収集された値の母分散。
varsamp	定義された評価期間内に収集された値の標本分散。

共通パラメータ

- `/host/key` は、ホストのアイテム履歴を参照する関数に共通する必須の第 1 パラメータです。
- `(sec|#num)<:time shift>` は、ホストのアイテム履歴を参照する関数に共通する第 2 パラメータです。以下の形式で指定します。
  - **sec** - 秒単位の最大評価期間（時間のサフィックスを使用できます）。
  - **#num** - 収集された最新の値における最大評価範囲（ハッシュ記号が前に付いている場合）。
  - **time shift**（オプション）- 評価ポイントを過去の時点に移動できます。time shift の指定方法については、詳細を参照してください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます
- 関数固有のパラメータは各関数で説明されています
- `/host/key` および `(sec|#num)<:time shift>` パラメータは決して引用符で囲まないでください

avg(/host/key,(sec|#num)<:time shift>)

定義された評価期間内におけるアイテムの平均値です。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照してください。

time shift は、現在の平均値を少し前の時点の平均値と比較する必要がある場合に便利です。

例:

```
avg(/host/key,1h) # 現在までの直近 1 時間の平均値を計算
avg(/host/key,#5) # 直近 5 個の値の平均値を計算
avg(/host/key,1h:now-1d) #25 時間前から 24 時間前までの 1 時間の平均値を計算
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) # すべての MySQL サーバーの平均プロセッサ負
avg(/host/proc.num,5m)>300 # 直近 5 分間のプロセス数の平均が 300 を超えている場合にトリガー
```

bucket_percentile(item filter,time period,percentage)

ヒストグラムのバケットからパーセンタイルを計算します。

パラメータ:

- **item filter** - item filterを参照してください。
- **time period** - time periodを参照してください。
- **percentage** - パーセンテージ (0-100)。

コメント:

- 計算アイテムでのみサポートされています。
- この関数は `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))` のエイリアスです。

例:

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #5 分間の 95 パーセンタイル応答時間を計算
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #99 パーセンタイルの API レイテンシを
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

foreach 関数によって返された配列内の値の数を数えます。
 サポートされるforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメーター:

- **func_foreach** - 返された値の数を数える対象となる foreach 関数です。詳細はforeach 関数 を参照してください。count_foreach と bucket_rate_foreach は追加パラメーター をサポートしている点に注意してください。
- **item filter** - item filter を参照してください。
- **time period** - time period を参照してください。
- **operator**（二重引用符で囲む必要があります）。サポートされる operators:
eq - 等しい
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like - パターンを含む場合に一致（大文字小文字を区別）
bitand - ビット単位の AND
regexp - `pattern` で指定された正規表現に大文字小文字を区別して一致
iregexp - `pattern` で指定された正規表現に大文字小文字を区別せずに一致
- **pattern** - 必要なパターンです（文字列引数は二重引用符で囲む必要があります）。3 番目のパラメーターで *operator* が指定されている場合にサポートされます。

コメント:

- 履歴データに関連する foreach 関数（max_foreach、avg_foreach など）と **count()** を使用すると、パフォーマンスに影響する可能性があります。一方、設定データのみで動作する **exists_foreach()** を使用しても、そのような影響はありません。
- オプションのパラメーター *operator* または *pattern* は、カンマの後で空欄のままにすることはできず、完全に省略する必要があります。
- 3 番目のパラメーターとして *bitand* を指定した場合、4 番目の `pattern` パラメーターには、スラッシュ `/` で区切った 2 つの数値を指定できます: 比較対象の数値/マスク。count() は値と *mask* のビット単位の AND を計算し、その結果を 比較対象の数値と比較します。ビット単位の AND の結果が 比較対象の数値と等しい場合、その値はカウントされます。
 比較対象の数値と *mask* が等しい場合は、*mask* のみを指定できます（`/` なし）。
- 3 番目のパラメーターとして *regexp* または *iregexp* を指定した場合、4 番目の `pattern` パラメーターには、通常の正規表現またはグローバル（`@` で始まる）正規表現を指定できます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定を継承します。regexp の照合では、浮動小数点値は常に `.` の後に 4 桁の小数で表されます。また、大きな数値では、10 進表現（データベースに保存される）と 2 進表現（Zabbix サーバーで使用される）の差が、4 桁目の小数に影響する場合があることに注意してください。

例:

```
count(max_foreach(/*/net.if.in[*],1h)) #calculate the number of net.if.in items that received data in the
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #calculate the number of file systems with over 95% c
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

ヒストグラムのバケットから φ-quantile を計算します。
 サポートされるforeach 関数: *bucket_rate_foreach*。

パラメータ:

- **quantile** - 0 ≤ φ ≤ 1;
- **bucketN, valueN** - 手動で入力するパラメータのペア（2 組以上）、またはbucket_rate_foreach の応答。

コメント:

- 計算アイテムでのみサポートされます。
- 機能的には PromQL の'histogram_quantile' に対応します。
- 最後の'Infinity' バケット（"+*inf*"）の値が 0 に等しい場合は -1 を返します。

例:

```
histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]),"+Inf",last(/host/rate_bucket[Inf]))
histogram_quantile(0.5,bucket_rate_foreach(//item_key,30s))
```

item_count(item filter)

フィルター条件に一致する、設定内の既存のアイテム数です。
 サポートされる値の型: *Integer*。

パラメータ:

- **item filter** - アイテム選択の条件です。ホストグループ、ホスト、アイテムキー、タグによる参照が可能です。ワイルドカードをサポートしています。詳細はitem filterを参照してください。

コメント:

- 計算アイテムでのみサポートされます。
- *count(exists_foreach(item_filter))* 関数のエイリアスとして動作します。

例:

```
item_count(/*/agent.ping?[group="Host group 1"]) #"Host group 1" 内で agent.ping アイテムを持つホスト数を計算
```

kurtosis(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値における、確率分布の「裾の重さ」を表します。参照: Kurtosis。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #5 分間の CPU 使用率の急激なスパイクでトリガー
```

mad(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の中央値絶対偏差です。参照: 中央値絶対偏差。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

定義された評価期間内におけるアイテムの最大値。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
max(/host/key,1h) - min(/host/key,1h) # 現在までの直近 1 時間における最大値と最小値の差を計算
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) # すべての Linux サーバーにおけるディスク使用
```

min(/host/key,(sec|#num)<:time shift>)

定義された評価期間内におけるアイテムの最小値です。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
max(/host/key,1h) - min(/host/key,1h) # 現在までの直近 1 時間における最大値と最小値の差を計算します（値の差分）
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) # すべて
```

skewness(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値における確率分布の非対称性です。参照: Skewness。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
skewness(/host/vm.memory.size[used],5m)<-20 #trigger on sudden negative drops indicating crashes/restarts
```

stddevpop(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の母標準偏差です。参照: 標準偏差。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #CPU 使用率の変動が大きい場合にトリガー（10 分間の母標準偏差が 7 を超え
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の標本標準偏差です。関連項目: 標準偏差。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

この関数が動作するには、少なくとも 2 つのデータ値が必要です。

例:

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #CPU 使用率の変動が大きい場合のトリガー（10 分間の標本標準偏差が 7 を
```

sum(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の合計です。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) # すべての Linux サーバーの受信ネットワークトラフィック
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) # すべての MySQL サーバーのディスク容量の合計
sum(last_foreach(/*/net.dns[*,*,*])) # 成功した DNS チェックの総数を計算
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の二乗和です。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
sumofsquares(/host/calculated.net.if.rate,1m) # ネットワークトラフィック率（受信と送信）の二乗和
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) # すべての Linux サーバーにおける最新の受信ネ
```

varpop(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の母分散です。あわせて参照: 分散。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

例:

```
varpop(/host/system.cpu.util[,user],10m)>50 #CPU 使用率の変動が大きい場合のトリガー（10 分間の分散が 50 を超える）
```

varsamp(/host/key,(sec|#num)<:time shift>)

定義された評価期間内に収集された値の標本分散です。以下も参照してください: Variance。
 サポートされる値の型: *Float*, *Integer*。
 サポートされるforeach 関数: *last_foreach*。

パラメータ: 共通パラメータを参照してください。

この関数が動作するには、少なくとも 2 つのデータ値が必要です。

例:

```
varsamp(/host/system.cpu.util[,user],10m)>50 #trigger if CPU usage fluctuates heavily (sample variance is
```

サポートされているすべての関数を参照してください。

Foreach 関数

概要

foreach 関数は、集計計算で使用され、使用する アイテムフィルターによって選択された各アイテムごとに 1 つの集計値を返します。値の配列が返されます。

たとえば、avg_foreach 関数は、指定された時間間隔における選択されたアイテムの 平均履歴値を各値として含む配列を返します。

Attention:
foreach 関数は、計算アイテム の式でのみサポートされます。通常のアイテムキーでは呼び出せず、トリガー式を直接作成するためにも使用できません。トリガー式では、計算アイテムのキーを参照することで使用できます。

アイテムフィルター は、foreach 関数で使用される構文の一部です。アイテムフィルターではワイルドカードの使用がサポートされているため、必要なアイテムを柔軟に選択できます。

サポートされる関数

Function	Description
avg_foreach	各アイテムの平均値を返します。
bucket_rate_foreach	`histogram_quantile()` 関数で使用するのに適した、(バケットの上限, レート値) のペアを返します。ここで”bucket upper bound” は、<parameter number> parameter によって定義されるアイテムキーのパラメーター値です。
count_foreach	各アイテムの値の数を返します。
exists_foreach	有効な各アイテムに対して’1’ を返します。
last_foreach	各アイテムの最新値を返します。
max_foreach	各アイテムの最大値を返します。
min_foreach	各アイテムの最小値を返します。
sum_foreach	各アイテムの値の合計を返します。

関数構文

foreach 関数は、2 つの共通パラメータ `item filter`（詳細は以下を参照）と `time period` をサポートします。

```
foreach_function(item filter,time period)
```

例:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

は、MySQL サーバーグループ内の各’mysql.qps’ アイテムの 5 分間の平均値を返します。

一部の関数は追加のパラメータをサポートしていることに注意してください。

アイテムフィルターの構文

アイテムフィルター:

```
/host/key[parameters]?[conditions]
```

は 4 つの部分で構成されます。

- host - ホスト名
- key - アイテムキー（パラメータなし）
- parameters - アイテムキーパラメータ
- conditions - ホストグループおよび/またはアイテムタグに基づく条件（式として）

スペースは条件式の中でのみ許可されます。

ワイルドカードの使用

- ワイルドカードは、ホスト名、アイテムキー、または個々のアイテムキーパラメータの置き換えに使用できます。

- ホストまたはアイテムキーのいずれかはワイルドカードなしで指定する必要があります。したがって、/host/* や/*/key は有効なフィルターですが、/*/* は無効です。
- ワイルドカードは、ホスト名、アイテムキー、アイテムキーパラメータの一部には使用できません。
- ワイルドカードは、1 つのアイテムキーパラメータ以上には一致しません。したがって、各パラメータごとにワイルドカードを個別に指定する必要があります（例：key[abc,*,*]）。

条件式

条件式では以下がサポートされています。

- オペランド:
  - group - ホストグループ
  - tag - アイテムタグ
  - "<text>" - 文字列定数。" や\をエスケープするには\を使用
- 大文字小文字を区別する文字列比較演算子: =, <>
- 論理演算子: and, or, not
- 括弧によるグループ化: ( )

文字列定数の引用は必須です。大文字小文字を区別した完全一致のみがサポートされます。

Warning:
フィルターでタグを指定する場合（例：tag="tagname:value"）、コロン「:」は区切り文字として使用されます。それ以降はすべてタグ値と見なされます。したがって、現時点では「:」を含むタグ名の指定はサポートされていません。

例

アイテムキー、ホストグループ、タグを参照する複雑なフィルターを使用することができます。例を示します。

構文例	説明
/host/key[abc,*]	このホスト上の類似アイテムに一致します。
/*/key	任意のホストの同じアイテムに一致します。
/*/key?[group="ABC" and tag="tagname:value"]	ABC グループに属し、'tagname:value' タグを持つ任意のホストの同じアイテムに一致します。
/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]	ABC または DEF グループに属し、それぞれのタグを持つ任意のホストの類似アイテムに一致します。

参照されるすべてのアイテムが存在し、データを収集している必要があります。有効なホスト上の有効なアイテムのみが計算に含まれます。サポートされていない状態のアイテムは含まれません。

Attention:
参照アイテムのアイテムキーを変更した場合、フィルターも手動で更新する必要があります。

親ホストグループを指定すると、親グループとその配下のすべてのホストグループおよびそのアイテムが含まれます。

期間

2 番目のパラメータでは、集計する期間を指定できます。期間は時間としてのみ表現でき、値の数（**#** を前置）はサポートされていません。

このパラメータでは、利便性のためにサポートされている単位記号を使用できます。たとえば、300s（300 秒）の代わりに 5m（5 分）、86400（86400 秒）の代わりに 1d（1 日）と指定できます。

last_foreach 関数では、期間はオプションのパラメータです（Zabbix 7.0 以降でサポート）。省略できます。

```
last_foreach(/*/key?[group="host group"])
```

exists_foreach 関数では、期間はサポートされていません。

追加パラメータ

bucket_rate_foreach

bucket_rate_foreach 関数では、3 つ目のオプションパラメータがサポートされています。

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

ここで <parameter number> は、アイテムキー内の”bucket” 値の位置です。
たとえば、myItem[aaa,0.2] の”bucket” 値が 0.2 の場合、その位置は 2 です。

<parameter number> のデフォルト値は 1 です。

count_foreach

count_foreach 関数では、3 つ目と 4 つ目のオプションパラメータがサポートされています。

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

ここで:

- **operator** はアイテム値に対する条件演算子です（ダブルクォートで囲む必要があります）。
サポートされる operators:
eq - 等しい
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like - パターンを含む場合に一致（大文字小文字を区別）
bitand - ビット単位 AND
regexp - pattern で指定した正規表現に一致（大文字小文字を区別）
iregexp - pattern で指定した正規表現に一致（大文字小文字を区別しない）
- **pattern** は必須のパターンです（文字列引数はダブルクォートで囲む必要があります）。3 つ目のパラメータで *operator* が指定されている場合にサポートされます。

コメント:

- オプションパラメータの *operator* または *pattern* は、カンマの後に空欄のまま残すことはできず、完全に省略する必要があります。
- 3 つ目のパラメータとして *bitand* を指定した場合、4 つ目の pattern パラメータには、スラッシュ / で区切った 2 つの数値を指定できます: **number_to_compare_with/mask**。count_foreach() は値と *mask* に対して”bitwise AND” を計算し、その結果を *number_to_compare_with* と比較します。
”bitwise AND” の結果が *number_to_compare_with* と等しい場合、その値はカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定すればよく（/ は不要です）。
- 3 つ目のパラメータとして *regexp* または *iregexp* を指定した場合、4 つ目の pattern パラメータには通常の正規表現、またはglobal（@ で始まる）正規表現を指定できます。
global 正規表現の場合、大文字小文字の区別は global 正規表現の設定を継承します。
regexp マッチングでは、float 値は常に小数点 . の後に 4 桁で表されます。
また、大きな数値では、10 進表現（データベースに保存）と 2 進表現（Zabbix サーバーで使用）の差異が 4 桁目の小数に影響する場合があることに注意してください。

foreach 関数の使用方法の詳細と例については、集計計算 を参照してください。

可用性による動作

次の表は、ホスト/アイテムおよび履歴データの可用性が制限されている場合に、各関数がどのように動作するかを示しています。

関数	無効なホスト	データがあるが利用できないホスト	データがない利用できないホスト	無効なアイテム	未サポー
avg_foreach	無視	平均値を返す	無視	無視	無視
bucket_rate_foreach	無視	バケットレートを返す	無視	無視	無視
count_foreach	無視	カウントを返す	0	無視	無視
exists_foreach	無視	1	1	無視	1
last_foreach	無視	最新値を返す	無視	無視	無視
max_foreach	無視	最大値を返す	無視	無視	無視
min_foreach	無視	最小値を返す	無視	無視	無視
sum_foreach	無視	合計値を返す	無視	無視	無視

アイテムが無視される場合、集計には何も追加されません。

ビット演算関数

ここに一覧表示されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテムの式
- 式マクロ

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を表示できます。

Function	Description
bitand	アイテム値とマスクの「ビット単位 AND」の値。
bitlshift	アイテム値のビット左シフト。
bitnot	アイテム値の「ビット単位 NOT」の値。
bitor	アイテム値とマスクの「ビット単位 OR」の値。
bitrshift	アイテム値のビット右シフト。
bitxor	アイテム値とマスクの「ビット単位排他的 OR」の値。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- パラメータとして式を受け付けます
- オプションの関数パラメータ (またはパラメータの一部) は < > で示します

bitand(value,mask)

アイテム値とマスクの「ビットごとの AND」の値。
 サポートされる値の型：整数。

パラメータ:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビットごとに行われますが、すべての値は 10 進数で指定および返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

例:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3番目または4番目のビットがセットされているか
bitand(last(/host/key),20)=16 #3番目のビットがセットされておらず、5番目のビットがセットされている
```

bitlshift(value,bits to shift)

アイテム値のビット単位の左シフト
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **bits to shift** (必須) - シフトするビット数

比較はビット単位で行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitnot(value)

アイテム値の「ビット単位の NOT」の値。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値。

比較はビット単位で行われますが、すべての値は 10 進数で指定され、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitor(value,mask)

アイテム値とマスクの「ビットごとの OR」の値。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビットごとに行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitrshift(value,bits to shift)

アイテム値のビット単位での右シフト
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値

- **bits to shift** (必須) - シフトするビット数

比較はビット単位で行われますが、すべての値は 10 進数で指定し、10 進数で返されます。例えば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

bitxor(value,mask)

アイテム値とマスクの「ビットごとの排他的論理和」の値。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビットごとに行われますが、すべての値は 10 進数で指定し、10 進数で返されます。たとえば、3 番目のビットをチェックする場合は、100 ではなく 4 と比較します。

サポートされているすべての関数を参照してください。

日付と時刻の関数

ここに一覧表示されているすべての関数は、次の場所でサポートされています。

- トリガー式
- 計算アイテムの式
- 式マクロ

Attention:
日付と時刻の関数は、それ単体では式内で使用できません。式には、ホストのアイテムを参照する別のグループ の関数を少なくとも 1 つ含める必要があります（nodata() 関数を除く）。日付と時刻の関数が式内でどのように動作するかの詳細については、計算時刻 を参照してください。

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

Function	Description
date	現在の日付を YYYYMMDD 形式で返します。
dayofmonth	月内の日を 1 から 31 の範囲で返します。
dayofweek	曜日を 1 から 7 の範囲で返します。
now	エポック（1970 年 1 月 1 日 00:00:00 UTC）からの経過秒数。
time	現在時刻を HHMMSS 形式で返します。

関数の詳細

date

現在の日付 (YYYYMMDD 形式)。

例:

```
date()<20220101
```

dayofmonth

1 から 31 の範囲の月の日。

例:

```
dayofmonth()=1
```

dayofweek

曜日を 1 から 7 の範囲で指定します (月 - 1, 日 - 7)。

例 (平日のみ):

```
dayofweek()<6
```

例 (週末のみ):

```
dayofweek()>5
```

now

エポック（1970 年 1 月 1 日 00:00:00 UTC）からの秒数。

例:

`now()<1640998800`

time

現在の時刻 (HHMMSS 形式)。

例 (夜間のみ、00:00-06:00):

`time()<060000`

サポートされているすべての関数を参照してください。

ヒストリ関数

ここに記載されているすべての関数は、以下でサポートされています。

- トリガーの条件式
- 計算アイテムの数式
- 式マクロ

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

関数	説明
change	前回の値と最新の値の差分。
changecount	定義された評価期間内で、隣接する値の間に発生した変更の回数。
count	定義された評価期間内の値の数。
countunique	定義された評価期間内の一意の値の数。
find	定義された評価期間内で値の一致を検索します。
first	定義された評価期間内で最初（最も古い）の値。
firstclock	定義された評価期間内で最初（最も古い）の値のタイムスタンプ。
fuzzytime	パッシブエージェントの時刻と Zabbix サーバー/プロキシの時刻の差を確認します。
jsonpath	JSONPath クエリの結果を配列として返します。
last	最も新しい値。
lastclock	定義された評価期間内で、N 番目に新しい値のタイムスタンプ。
logeventid	最後のログエントリのイベント ID が正規表現に一致するかどうかを確認します。
logseverity	最後のログエントリのログ重大度。
logsource	最後のログエントリのログソースが正規表現に一致するかどうかを確認します。
logtimestamp	N 番目に新しいログアイテムの値のログメッセージのタイムスタンプ。
monodec	値が単調減少しているかどうかを確認します。
monoinc	値が単調増加しているかどうかを確認します。
nodata	データを受信していないかどうかを確認します。
percentile	期間の P パーセンタイル。P（パーセンテージ）は 3 番目のパラメータで指定します。
rate	定義された期間内に単調増加するカウンターが増加する 1 秒あたりの平均速度。
xmlxpath	XML XPath クエリの結果を配列として返します。

共通パラメータ

- `/host/key` は、ホストのアイテム履歴を参照する関数に共通する必須の第 1 パラメータです。
- `(sec|#num)<:time shift>` は、ホストのアイテム履歴を参照する関数に共通する第 2 パラメータです。次のように指定します。
  - **sec** - 秒単位の最大評価期間（時間のサフィックスを使用できます）、または
  - **#num** - 最新の収集値における最大評価範囲（ハッシュ記号が前に付いている場合）。
  - **time shift**（オプション）を使用すると、評価ポイントを過去に移動できます。time shift の指定方法については、詳細を参照してください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ (またはパラメータの一部) は < > で示します
- 関数固有のパラメータは各関数で説明します

- /host/key および（sec|#num)<:time shift> パラメータは引用符で囲まないでください

change(/host/key)

前回の値と最新の値の差分。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。
 文字列の場合は、0 - 値が等しい; 1 - 値が異なる。

パラメータ: 共通パラメータを参照。

コメント:

- 数値の差分は、以下のような受信値の例で計算されます（「前回」と「最新」の値 = 差分）:
'1' と '5' = +4
'3' と '1' = −2
'0' と '-2.5' = −2.5。
- トリガーでは、急激なスパイク（またはドロップ）、カウンタリセット、数値状態の遷移を検出するのに便利です。
- 比較にはabs も参照してください。

例:

```
change(/host/system.uptime)<0 # 前回の値からシステムの稼働時間がマイナスになった（再起動を示す）
change(/host/system.cpu.load[all,avg1])>2 #CPU ロード（1 分間）が前回の値から 2 以上増加した
change(/host/vfs.fs.size[/,free])<-1G # チェック間で空きディスク容量が 1GB 以上減少した
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

定義された評価期間内で隣接する値間の変化回数。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照
- **mode** (ダブルクォーテーションで囲む必要あり) - 指定可能な値: *all* - すべての変化をカウント (デフォルト); *dec* - 減少のみカウント; *inc* - 増加のみカウント

コメント:

- 数値以外の値の型の場合、*mode* パラメータは無視されます。
- トリガーでは、頻繁な状態変化 (不安定さを示す) を検出するのに便利です。

例:

```
changecount(/host/icmpping,10m)>5 #10 分間に ping の状態が 5 回以上変化した
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5 #eth0 の operational state が 1
changecount(/host/proc.num[httpd],15m)>10 #httpd プロセス数が 15 分間に 10 回以上変化した
changecount(/host/key,#10,"inc") # 直近 10 個の値の中で値が増加した回数（隣接値と比較）
changecount(/host/key,24h,"dec") # 現在までの直近 24 時間で値が減少した回数（隣接値と比較）
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内の値の数。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメーター:

- 共通パラメーターを参照
- **operator** (二重引用符で囲む必要があります) サポートされる operators:
eq - 等しい (integer, float のデフォルト)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (string, text, log のデフォルト) - パターンを含む場合に一致 (大文字小文字を区別)
bitand - ビット単位の AND
regexp - pattern で指定した正規表現に大文字小文字を区別して一致
iregexp - pattern で指定した正規表現に大文字小文字を区別せず一致
- **pattern** - 必要なパターン (文字列引数は二重引用符で囲む必要があります)

コメント:

- Float アイテムは 2.22e-16 の精度で一致します。
- *like* は integer 値の operator としてはサポートされていません。
- *like* と *bitand* は float 値の operator としてはサポートされていません。
- string、text、log の値では、*eq*、*ne*、*like*、*regexp*、*iregexp* のみがサポートされます。
- *bitand* を operator とする場合、4 番目の pattern パラメーターには、スラッシュ / で区切った 2 つの数値を指定できます: 比較対象の数値/マスク。count() は値と *mask* に対して「ビット単位の AND」を計算し、その結果を 比較対象の数値と比較します。「ビット単位の AND」の結果が 比較対象の数値と等しい場合、その値はカウントされます。
 比較対象の数値と *mask* が等しい場合は、*mask* だけを指定できます ("/" なし)。
- *regexp* または *iregexp* を operator とする場合、4 番目の pattern パラメーターには、通常の正規表現、またはglobal (先頭が'@') の正規表現を指定できます。global regular expression の場合、大文字小文字の区別は global regular expression の設定から継承されます。regexp の一致判定では、float 値は常に'.' の後に 4 桁の小数で表されます。また、大きな数値では、10

進数 (データベースに保存される形式) と 2 進数 (Zabbix サーバーで使用される形式) の表現差により、4 桁目の小数に影響が出る場合があります。

例:

```
count(/host/icmpping,30m,,"0")>5 #ping は 30 分間に 5 回を超えて失敗している
count(/host/key,10m,"like","error") # 現在までの直近 10 分間に 'error' を含む値の数
count(/host/key,10m,,12) # 現在までの直近 10 分間に '12' と等しい値の数
count(/host/key,10m,"gt",12) # 現在までの直近 10 分間に '12' より大きい値の数
count(/host/key,#10,"gt",12) # 現在までの直近 10 個の値のうち '12' より大きい値の数
count(/host/key,10m:now-1d,"gt",12) # 現在から 24 時間前から 24 時間前の 10 分間までの間で '12' より大きかった値の数
count(/host/key,10m,"bitand","6/7") # 現在までの直近 10 分間に、下位 3 ビットに '110' (2 進数) を持つ値の数
count(/host/key,10m:now-1d) # 現在から 24 時間前から 24 時間前の 10 分間までの間の値の数
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内の一意な値の数。
 サポートされる値の種類: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメーター:

- 共通パラメーターを参照してください
- **operator** (二重引用符で囲む必要があります)。サポートされる operators:
eq - 等しい (integer, float のデフォルト)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (string, text, log のデフォルト) - パターンを含む場合に一致します (大文字小文字を区別)
bitand - ビット単位の AND
regexp - pattern で指定された正規表現に大文字小文字を区別して一致
iregexp - pattern で指定された正規表現に大文字小文字を区別せずに一致
- **pattern** - 必要なパターン (string 引数は二重引用符で囲む必要があります)

コメント:

- Float アイテムは 2.22e-16 の精度で一致します。
- *like* は integer 値の operator としてはサポートされていません。
- *like* と *bitand* は float 値の operator としてはサポートされていません。
- string、text、log の値では *eq*、*ne*、*like*、*regexp*、*iregexp* のみがサポートされます。
- *bitand* を operator とする場合、4 番目の pattern パラメーターには、スラッシュ / で区切った 2 つの数値を指定できます: 比較対象の数値/マスク。countunique() は値と *mask* に対して「ビット単位の AND」を計算し、その結果を 比較対象の数値と比較します。結果が 比較対象の数値と等しい場合、その値はカウントされます。
 比較対象の数値と *mask* が等しい場合は、*mask* のみを指定できます ("/" なし)。
- *regexp* または *iregexp* を operator とする場合、4 番目の pattern パラメーターには通常の正規表現、またはglobal 正規表現 (@ で始まる) を指定できます。global 正規表現の場合、大文字小文字の区別は global 正規表現の設定を継承します。regexp マッチングでは、float 値は常に'.' の後に 4 桁の小数で表されます。また、大きな数値では、10 進数表記 (データベースに保存される形式) と 2 進数表記 (Zabbix サーバーで使用される形式) の差異が 4 桁目の小数に影響する場合があることに注意してください。

例:

```
countunique(/host/key,10m) # 現在から過去 10 分間の一意な値の数
countunique(/host/key,10m,"like","error") # 現在から過去 10 分間に 'error' を含む一意な値の数
countunique(/host/key,10m,,12) # 現在から過去 10 分間に '12' と等しい一意な値の数
countunique(/host/key,10m,"gt",12) # 現在から過去 10 分間に '12' より大きい一意な値の数
countunique(/host/key,#10,"gt",12) # 現在までの直近 10 個の値の中で '12' より大きい一意な値の数
countunique(/host/key,10m:now-1d,"gt",12) # 現在から 24 時間前の時点から 24 時間前の 10 分間までの間で '12' より大き
countunique(/host/key,10m,"bitand","6/7") # 現在から過去 10 分間に、下位 3 ビットに '110' (2 進数) を持つ一意な値の
countunique(/host/key,10m:now-1d) # 現在から 24 時間前の時点から 24 時間前の 10 分間までの間の一意な値の数
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内で値の一致を検索します。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。
 戻り値: 1 - 見つかった; 0 - それ以外。

パラメーター:

- 共通パラメーターを参照してください
- **sec** または **#num** (省略可) - 指定しない場合は最新の値が既定になります
- **operator** (二重引用符で囲む必要があります) サポートされる operators:
eq - 等しい (整数、浮動小数点数の既定)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (文字列、テキスト、ログの既定) - pattern で指定した文字列を含む場合に一致 (大文字小文字を区別)
bitand - ビット単位の AND
regexp - pattern で指定した正規表現に一致 (大文字小文字を区別)
iregexp - pattern で指定した正規表現に一致 (大文字小文字を区別しない)

- **pattern** - 必要なパターン (文字列引数は二重引用符で囲む必要があります); operator が *regexp*、*iregexp* の場合は Perl Compatible Regular Expression (PCRE) 正規表現

コメント:

- 複数の値が処理される場合、一致する値が 1 つでもあれば'1' が返されます。
- *like* は整数値の operator としてはサポートされていません。
- *like* と *bitand* は浮動小数点数値の operator としてはサポートされていません。
- 文字列、テキスト、ログの値では *eq*、*ne*、*like*、*regexp*、*iregexp* のみがサポートされます。
- operator として *regexp* または *iregexp* を使用する場合、4 番目の pattern パラメーターには通常の正規表現、またはglobal (先頭が'@') 正規表現を指定できます。グローバル正規表現の場合、大文字小文字の区別はグローバル正規表現の設定を継承します。

例:

```
find(/host/key,10m,"like","error") # 直近 10 分間で 'error' を含む値を検索
find(/host/agent.version,,"like","beta")=1 #Zabbix エージェントは beta 版であり、アップグレードが必要です
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 # 内部 Web サーバー エラーが見つかりました
```

first(/host/key,sec<:time shift>)

定義された評価期間内の最初 (最も古い) の値。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照

last()も参照。

例:

```
first(/host/key,1h) # 現在までの直近 1 時間で最も古い値を取得
```

firstclock(/host/key,sec<:time shift>)

定義された評価期間内の最も古い値のタイムスタンプ。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照してください。

指定された期間にデータが収集されていない場合、関数はエラーで失敗します。

lastclock()も参照してください。

例:

```
firstclock(/host/key,1h) # 直近 1 時間の最も古い値のタイムスタンプを取得
firstclock(/host/key,1h:now-24h) #1 日前の直近 1 時間の最も古い値のタイムスタンプを取得
```

fuzzytime(/host/key,sec)

パッシブエージェントの時刻が Zabbix サーバー/プロキシの時刻とどの程度異なるかを確認します。
 サポートされる値の型: *Float*, *Integer*.
 戻り値: 1 - パッシブアイテムの値（タイムスタンプとして）と Zabbix サーバー/プロキシのタイムスタンプ（値収集時刻）との差が *sec* 秒以下の場合。0 - それ以外の場合。

パラメーター:

- 共通パラメーター を参照してください。

コメント:

- 通常は'system.localtime' アイテムと組み合わせて使用し、ローカル時刻が Zabbix サーバーのローカル時刻と同期しているかを確認します。*Note* 'system.localtime' は Zabbix エージェントでパッシブチェック として設定する必要があります。Zabbix エージェント 2 では、アクティブチェックとして設定することもできます。
- vfs.file.time[/path/file,modify] キーと組み合わせて、ファイルが長時間更新されていないことを確認するためにも使用できます。
- この関数は、複雑なトリガー式 (複数のアイテムが関与するもの) での使用は推奨されません。予期しない結果を引き起こす可能性があるためです(時刻差は最も新しいメトリックで測定されます)。たとえば、fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0 のような式です。

例:

```
fuzzytime(/host/system.localtime,5m)=0 # クライアントのローカル時刻が Zabbix サーバー/プロキシの時刻より 5 分以上ずれ
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 # クライアントのローカル時刻が
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

JSONPath クエリの結果の配列を返します。
 サポートされる値のタイプ: *String*、*Text*、*Log*。

パラメータ:

- 共通パラメータを参照してください
- **pattern** - 必須のパターン（文字列の引数は二重引用符で囲む必要があります）

コメント:

- この関数は集計関数とともに使用します。
- 選択された各履歴値に対して JSONPath クエリが実行され、空でないクエリの成功結果が配列に追加されます。1 つの履歴値に対するクエリ結果に複数の一致が含まれる場合、それぞれの一致が個別に配列へ追加されます。
- 次の場合、関数はエラーで失敗します。
  - `pattern` の形式が無効な場合。
  - いずれかの履歴値に無効な JSON が含まれている場合、またはその値に対する JSONPath クエリが失敗した場合。
- 次の場合、関数は空の配列を返します。
  - 履歴値が選択されなかった場合。
  - 選択されたすべての履歴値に無効な JSON が含まれている場合。

例:

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

last(/host/key,<#num<:time shift»)

最新の値。
 サポートされている値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照
- **#num** (オプション) - N 番目に新しい値

コメント:

- ハッシュタグ付きの期間 (#N) は、他の多くの関数とは異なる動作をすることに注意してください。例えば: `last(/host/key)` は常に `last(/host/key,#1)` と等しい; `last(/host/key,#3)` は 3 番目に新しい値 (最新の 3 つの値ではない)。
- 1 秒以内に 2 つ以上の値が履歴に存在する場合、Zabbix は値の正確な順序を保証しません。
- first()も参照してください。

例:

```
last(/host/key) # 最新の値を取得
last(/host/key,#2) #1 つ前の値を取得
last(/host/key,#1)<>last(/host/key,#2) # 最新の 2 つの値が異なる
```

lastclock(/host/key,<#num<:time shift»)

定義された評価期間内で N 番目に新しい値のタイムスタンプ。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照;
- **#num** (省略可能) - N 番目に新しい値。

指定された期間にデータが収集されていない場合、または N 番目の値が収集されていない場合、関数はエラーで失敗します。

firstclock()も参照してください。

例:

```
lastclock(/host/key) # 最新値のタイムスタンプを取得
lastclock(/host/key,#2) #2 番目に新しい値のタイムスタンプを取得
now()-lastclock(/host/heartbeat.item)>300 # ハートビートアイテムが更新されてから 5 分以上経過している
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 # 直近 2 つの値
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #CPU
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

最後のログエントリのイベント ID が正規表現に一致するかどうかを確認します。
 サポートされる値の型: *Log*。
 戻り値: 0 - 一致しない; 1 - 一致する。

パラメーター:

- 共通パラメーター を参照

- **#num** (任意) - 直近から N 番目の値
- **pattern** (任意) - 必要なパターンを表す正規表現。Perl Compatible Regular Expression (PCRE) 形式（文字列引数は二重引用符で囲む必要があります）

例:

```
logeventid(/host/eventlog[Security],,"^4625$")=1 #ID が "4625" に一致するログエントリ（認証失敗）が見つかった
logeventid(/host/eventlog[System],,"^(6008|41)$")=1 #ID が "6008" または "41" に一致するログエントリが見つかった
```

logseverity(/host/key,<#num<:time shift»)

最後のログエントリのログ重大度。
 サポートされている値の型: ログ。
 戻り値: 0 - デフォルトの重大度; N - 重大度 (整数、Windows イベントログに便利: 1 - Information, 2 - Warning, 4 - Error, 7 - Failure Audit, 8 - Success Audit, 9 - Critical, 10 - Verbose)。

パラメータ:

- 共通パラメータを参照
- **#num** (オプション) - 最新から N 番目の値

Zabbix は Windows イベントログの **Information** フィールドからログ重大度を取得します。

例:

```
logseverity(/host/log[/var/log/syslog],10m)>3 # 重大度が"3" を超えるログエントリが見つかった
logseverity(/host/eventlog[System],10m)=4 # 重大度が"Error" に等しいログエントリが見つかった
```

logsource(/host/key,<#num<:time shift»,<pattern>)

最後のログエントリのログソースが正規表現に一致するかどうかを確認します。
 サポートされる値の型: *Log*。
 戻り値: 0 - 一致しない; 1 - 一致する。

パラメーター:

- 共通パラメーターを参照
- **#num** (任意) - N 番目に新しい値
- **pattern** (任意) - 必要なパターンを表す正規表現。Perl Compatible Regular Expression (PCRE) 形式（文字列引数は二重引用符で囲む必要があります）

通常、Windows のイベントログで使用されます。

例:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 # ソースが "MSSQLSERVER" に一致するログエントリが見つかっ
logsource(/host/eventlog[System],,"Service Control Manager")=1 # ソースが "Service Control Manager" に一致す
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^70
```

logtimestamp(/host/key,<#num<:time shift»)

N 番目に新しいログアイテム値のログメッセージのタイムスタンプ。
 サポートされる値の型: *Log*。

パラメータ:

- 共通パラメータを参照;
- **#num** (省略可能) - N 番目に新しい値。

コメント:

- タイムシフトの計算は、ログメッセージのタイムスタンプではなく、アイテム値のクロックに基づいて行われます;
- 次の場合、関数はエラーで失敗します:
  - ログタイプ以外のアイテムが受信された場合;
  - 指定された期間にデータが収集されていない場合;
  - N 番目の値が収集されていない場合;
  - ログメッセージにタイムスタンプ値が含まれていない場合。

例:

```
logtimestamp(/host/key) # 最新のログメッセージのタイムスタンプを取得
logtimestamp(/host/key,#2) #2 番目に新しいログメッセージのタイムスタンプを取得
logtimestamp(/host/key,#3:now-1d) #1 日前の 3 番目に新しいログメッセージのタイムスタンプを取得
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

値が単調減少しているかどうかを確認します。
 サポートされている値の型: 整数。
 戻り値: 1 - 期間内のすべての要素が連続して減少している場合; 0 - それ以外の場合。

パラメータ:

- 共通パラメータを参照
- **mode** (ダブルクォーテーションで囲む必要あり) - *weak* (すべての値が前の値以下; デフォルト) または *strict* (すべての値が減少している)

例:

```
monodec(/host/system.swap.size[all,free],60s) + monodec(/host2/system.swap.size[all,free],60s) + monodec(/
monodec(/host/proc.num[nginx],10m,"strict")=1 #nginx プロセス数が 10 分間単調減少している
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

値が単調増加しているかどうかを確認します。
 サポートされている値の型: 整数。
 戻り値: 1 - 期間内のすべての要素が連続して増加している場合; 0 - それ以外の場合。

パラメータ:

- 共通パラメータを参照
- **mode** (ダブルクォーテーションで囲む必要あり) - *weak* (すべての値が前の値以上; デフォルト) または *strict* (すべての値が増加している)

例:

```
monoinc(/host/system.localtime,#3,"strict")=0 # システムのローカル時刻が一貫して増加していない
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 # ファイル数が 24 時間停滞している場合にトリガー（増加が
```

nodata(/host/key,sec,<mode>)

データを受信していないかどうかを確認します。
 サポートされている値のタイプ: *Integer*、*Float*、*Character*、*Text*、*Log*。
 戻り値: 定義された期間中にデータを受信していない場合は 1、それ以外の場合は 0。

パラメータ:

- 共通パラメータを参照してください
- **sec** - history syncer プロセスは 30 秒ごとにこの関数を計算するため、期間は 30 秒未満にできません。`nodata(/host/key,0)` は使用できません
- **mode** - *strict*（二重引用符で囲む）に設定すると、この関数はプロキシの可用性の影響を受けなくなります（詳細はコメントを参照してください）

コメント:

- プロキシによって監視される「nodata」トリガーは、デフォルトではプロキシの可用性の影響を受けます。プロキシが利用できなくなった場合、接続が復旧しても「nodata」トリガーはすぐには発火せず、遅延期間中のデータをスキップします。なお、パッシブプロキシでは、接続が 15 秒を超えて切断され、2 秒以上経過してから復旧した場合に抑制が有効になります。アクティブプロキシでは、接続が 15 秒を超えてから復旧した場合に抑制が有効になります。プロキシの可用性の影響を受けないようにするには、3 番目のパラメータを使用します。例: `nodata(/host/key,5m,"strict")`。この場合、データがない評価期間（5 分）が経過すると、すぐに関数が発火します。
- 1 番目のパラメータで指定した期間内に次のいずれかの状況が発生した場合、この関数はエラーを表示します。
- データがなく、Zabbix サーバーが再起動された
- データがなく、メンテナンスが完了した
- データがなく、アイテムが追加または再有効化された
- エラーはトリガー一覧の *Info* 列に表示されます。
- Zabbix サーバー、プロキシ、エージェント間に時刻の差がある場合、この関数は正しく動作しないことがあります。時刻同期の要件も参照してください。

例:

```
nodata(/host/agent.ping,5m)=1 #5 分間 Zabbix エージェントからデータを受信していない場合にトリガー
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

期間の P パーセンタイル。P（パーセンテージ）は 3 番目のパラメータで指定します。
 サポートされる値の型：*Float*、*Integer*。

パラメータ:

- 共通パラメータを参照
- **percentage** - 小数点以下 4 桁までの 0 から 100（両端を含む）の浮動小数点数

例:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #1 時間のネットワーク入力（バイト/秒）の 95 パーセンタイルがし
percentile(/host/system.cpu.util,5m,95)>80 #CPU 使用率のユーザータイムパーセンテージの 95 パーセンタイルが 80 を超え
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 # ほとんどのレイテンシ測定値は 150ms 未満だが、上位 5% の値
```

```
percentile(/host/net.if.in[eth0,bytes],1h,50) #1 時間の受信ネットワークトラフィックの 50 パーセンタイル（中央値）を計
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #1 時間の値
```

rate(/host/key,sec<:time shift>)

指定した期間内の単調増加カウンターの 1 秒あたりの平均増加量。
 サポートされる値の型: *Float*, *Integer*。

パラメータ:

- 共通パラメータを参照

PromQL の'rate' に機能的に対応。

例:

```
rate(/host/key,30s) #30 秒間の単調増加が 20 の場合、この関数は 0.67 を返します。
rate(/host/net.if.in[eth0,bytes],5m)>500000 # 過去 5 分間で eth0 の受信インターフェーストラフィックレートが 500KB/s
rate(/host/app.requests.count,1m)>100 # リクエストカウントカウンターが直近 1 分間で 1 秒あたり 100 リクエストを超えて
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

XML XPath クエリの結果の配列を返します。
 サポートされる値の種類: *String*、*Text*、*Log*。

パラメータ:

- 共通パラメータを参照してください
- **pattern** - 必須のパターン（文字列の引数は二重引用符で囲む必要があります）

コメント:

- この関数は集計関数とともに使用します。
- 選択された各履歴値に対して XML XPath クエリが実行され、空でないクエリの成功結果が配列に追加されます。1 つの履歴値に対するクエリ結果に複数の一致が含まれる場合は、各一致が個別に配列へ追加されます。
- 次の場合、関数はエラーで失敗します。
  - pattern の形式が無効である。
  - Zabbix が XML サポートなしでコンパイルされている。
  - いずれかの履歴値に無効な XML が含まれている、またはその値に対する XPath クエリが失敗する。
- 次の場合、関数は空の配列を返します。
  - 履歴値が選択されていない。
  - 選択されたすべての履歴値に無効な XML が含まれている。

例:

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

サポートされているすべての関数を参照してください。

Trend functions

Trend functions は、history functionsとは異なり、計算にtrendデータを使用します。

trend は 1 時間ごとの集計値を保存します。Trend functions はこれらの 1 時間平均値を使用するため、長期分析に適しています。

Trend function の結果はキャッシュされるため、同じパラメータを持つ同一関数への複数回の呼び出しでも、データベースから情報を取得するのは 1 回だけです。trend function cache は、TrendFunctionCacheSize サーバーパラメータによって制御されます。

キャッシュが実際にどの程度使用されているかを見積もるには、`zabbix[tcache,cache,paccessed]` を使用します。paccessed が低く、pitems と pmisses も低い場合は、通常、キャッシュサイズを小さくできます。

trend functions のみを参照するトリガーは、式内で最も短い時間単位ごとに 1 回評価されます。たとえば、次のようなトリガーは

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

1 日ごとに評価されます。トリガーに trend と history の両方、またはdate and time および/またはnodata() 関数が含まれる場合は、通常の原則に従って計算されます。

ここに一覧表示されているすべての関数は、以下でサポートされています。

- Trigger expressions
- Calculated items
- Expression macros

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

Function	Description
baselinedev	最後のデータ期間と、前の季節における同じデータ期間との間の偏差数を返します。
baselinewma	加重移動平均アルゴリズムを使用して、複数の等しい期間（「season」）における同じ時間枠のデータを平均化し、ベースラインを計算します。
trendavg	定義された期間内の trend 値の平均。
trendcount	定義された期間内の trend 値を計算するために使用された、正常に取得された history 値の数。
trendmax	定義された期間内の trend 値の最大値。
trendmin	定義された期間内の trend 値の最小値。
trendstl	検出期間中の異常率を返します。これは 0 から 1 の間の小数値で、((異常値の数)/(値の総数)) です。
trendsum	定義された期間内の trend 値の合計。

共通パラメータ

- `/host/key` は共通の必須の最初のパラメータです
- `time period:time shift` は共通の 2 番目のパラメータで、以下の通りです:
  - **time period** - 期間（最小値は'1h'）、<N><time unit> で定義されます。`N` は時間単位の数、`time unit` は h（時間）、d（日）、w（週）、M（月）、y（年）です。
  - **time shift** - 期間のオフセット（関数の例を参照）

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示します
- 関数固有のパラメータは各関数で説明します
- `/host/key` および `time period:time shift` パラメータは決して引用符で囲まないでください

baselinedev(/host/key,data period:time shift,season unit,num seasons)

直近のデータ期間と過去のシーズンの同じデータ期間との間の偏差の数（stddevpop アルゴリズムによる）を返します。

パラメータ:

- 共通パラメータを参照
- **data period** - シーズン内のデータ収集期間。<N><time unit> で定義されます。
`N` - 時間単位の数
`time unit` - h（時間）、d（日）、w（週）、M（月）、y（年）のいずれか。season 以下でなければなりません。
- **season unit** - 1 シーズンを定義するカレンダー単位（h、d、w、M、y）。data period より小さくできません。
- **num seasons** - 評価するシーズンの数

例:

```
baselinedev(/host/key,1d:now/d,"M",6) # 直近 1 日と過去 6 ヶ月の同じ日との間の標準偏差（母集団）を計算します。前月に日付
baselinedev(/host/key,1h:now/h,"d",10)  # 直近 1 時間と過去 10 日間の同じ時刻との間の標準偏差（母集団）を計算します
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

加重移動平均アルゴリズムを使用して、複数の等しい期間（「シーズン」）の同じ時間枠からデータを平均化することでベースラインを計算します。

パラメータ:

- 共通パラメータを参照
- **data period** - シーズン内でのデータ収集期間。<N><time unit> で定義されます。
`N` - 時間単位の数
`time unit` - h（時間）、d（日）、w（週）、M（月）、y（年）。シーズン以下でなければなりません。
Time shift - シーズン内でのデータ収集期間の終了を定義するオフセット（例を参照）
- **season unit** - 1 シーズンを定義するカレンダー単位（h, d, w, M, y）。data period より小さくできません。
- **num seasons** - 評価するシーズン数

例:

```
baselinewma(/host/key,1h:now/h,"d",3) # 昨日で終了する直近 3 日間の同じ時刻（時単位）からベースラインを計算します。"now
baselinewma(/host/key,2h:now/h,"d",3) # 昨日で終了する直近 3 日間の同じ 2 時間枠からベースラインを計算します。"now" が
baselinewma(/host/key,1d:now/d,"M",4) # 直近の完全な月の前の 4 か月間で「昨日」と同じ日付のデータからベースラインを計算
```

trendavg(/host/key,time period:time shift)

定義された期間内のトレンド値の平均。

パラメータ:

- 共通パラメータを参照

例:

```
trendavg(/host/key,1h:now/h) # 直前 1 時間の平均 (例: 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #2 時間前の平均 (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #3 時間前の平均 (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #1 年前の直前月の平均
```

trendcount(/host/key,time period:time shift)

定義された期間内でトレンド値を計算するために使用された履歴値のうち、正常に取得された値の数。

パラメータ:

- 共通パラメータを参照

例:

```
trendcount(/host/key,1h:now/h) # 直前 1 時間の値の数 (例 : 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #2 時間前の値の数 (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #3 時間前の値の数 (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #1 年前の前月の値の数
```

trendmax(/host/key,time period:time shift)

定義された期間内のトレンド値の最大値。

パラメータ:

- 共通パラメータを参照

例:

```
trendmax(/host/key,1h:now/h) # 直前の 1 時間の最大値 (例: 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → 直前の 1 時間 (12:00-13:00) の最大値と最小値の差分
trendmax(/host/key,1h:now/h-1h) #2 時間前の最大値 (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #3 時間前の最大値 (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #1 年前の前月の最大値
```

trendmin(/host/key,time period:time shift)

定義された期間内のトレンド値の最小値。

パラメータ:

- 共通パラメータを参照

例:

```
trendmin(/host/key,1h:now/h) # 直前 1 時間の最小値 (例 : 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → 直前 1 時間 (12:00-13:00) の最大値と最小値の差分 (
trendmin(/host/key,1h:now/h-1h) #2 時間前の最小値 (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #3 時間前の最小値 (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #1 年前の直前月の最小値
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

検出期間中の異常値の割合を返します。0 から 1 までの小数値で、((異常値の数)/(値の総数)) です。

パラメータ:

- 共通パラメータを参照
- **eval period** - 分解する必要がある期間 (最小値は'1h')、<N><time unit> で定義
N - 時間単位の数
time unit - h (時間)、d (日)、w (週)、M (月)、y (年)
- **detection period** - 異常値を計算するための eval period の終了前の期間 (最小値は'1h'、eval period より長くすることはできません)、<N><time unit> で定義
N - 時間単位の数
time unit - h (時間)、d (日)、w (週)
- **season** - 繰り返しパターン (「シーズン」) が予想される最短期間 (最小値は'2h'、eval period より長くすることはできません。eval period 内のエントリ数は、得られる周波数 (season/h) の 2 倍より大きくなければなりません)、<N><time unit> で定義
N - 時間単位の数
time unit - h (時間)、d (日)、w (週)
- **deviations** - 異常値と見なす偏差の数 (devalg で計算、10 進数可) (1 以上、デフォルトは 3)

- **devalg**（ダブルクォーテーションで囲む必要あり）- 偏差アルゴリズム。*stddevpop*、*stddevsamp*、*mad*（デフォルト）から選択
- **s window** - 季節性抽出のための loess ウィンドウのスパン（ラグ数、デフォルトは"10 * eval period 内のエントリ数 + 1"）

例:

```
trendstl(/host/key,100h:now/h,10h,2h) # 直近 100 時間のトレンドデータを分析し、その期間の直近 10 時間の異常値の割合を求
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #10 時間前までの 100 時間のトレンドデータを分析し、その全期間の
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #1 日前までの 100 日間のトレンドデータを分析し、その期間の直近 10 日の
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #1 年前の前月を分析し、その期間の最終日の異常値の割合を求める。周
```

trendsum(/host/key,time period:time shift)

定義された期間内のトレンド値の合計。

パラメータ:

- 共通パラメータを参照

例:

```
trendsum(/host/key,1h:now/h) # 直前 1 時間の合計（例：12:00-13:00）
trendsum(/host/key,1h:now/h-1h) #2 時間前の合計（11:00-12:00）
trendsum(/host/key,1h:now/h-2h) #3 時間前の合計（10:00-11:00）
trendsum(/host/key,1M:now/M-1y) #1 年前の直前月の合計
```

サポートされているすべての関数を参照してください。

数学関数

ここに一覧表示されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテムの式
- 式マクロ

数学関数は、特に記載がない限り、float および integer の値型でサポートされています。

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

Function	Description
abs	値の絶対値。
acos	値の逆余弦を、ラジアンで表した角度として返します。
asin	値の逆正弦を、ラジアンで表した角度として返します。
atan	値の逆正接を、ラジアンで表した角度として返します。
atan2	指定された縦座標（value）と横座標の逆正接を、ラジアンで表した角度として返します。
avg	参照されたアイテム値の平均値。
cbrt	値の立方根。
ceil	値を切り上げて、最も近い大きいか等しい整数にします。
cos	値の余弦。ここでの値はラジアンで表した角度です。
cosh	値の双曲線余弦。
cot	値の余接。ここでの値はラジアンで表した角度です。
degrees	値をラジアンから度に変換します。
e	オイラー数（2.718281828459045）。
exp	値をべき乗したオイラー数。
expm1	値をべき乗したオイラー数から 1 を引いた値。
floor	値を切り下げて、最も近い小さいか等しい整数にします。
log	自然対数。
log10	常用対数。
max	参照されたアイテム値の最大値。
min	参照されたアイテム値の最小値。
mod	剰余。
pi	円周率定数（3.14159265358979）。
power	値のべき乗。
radians	値を度からラジアンに変換します。
rand	ランダムな整数値を返します。
round	値を小数点以下の桁数に丸めます。

Function	Description
signum	値が負の場合は'-1'、ゼロの場合は'0'、正の場合は'1' を返します。
sin	値の正弦。ここでの値はラジアンで表した角度です。
sinh	値の双曲線正弦。ここでの値はラジアンで表した角度です。
sqrt	値の平方根。
sum	参照されたアイテム値の合計。
tan	値の正接。
truncate	値を小数点以下の桁数で切り捨てます。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切られます
- パラメータとして式が使用できます
- オプションの関数パラメータ (またはパラメータの一部) は < > で示されます

abs(value)

値の絶対値（0 からの距離）を返します。

パラメータ:

- **value** - チェックする値

例えば、'3' または'-3' の絶対値は'3' になります。

例:

```
abs(last(/host/key))>10
```

acos(value)

値のアークコサイン（逆余弦）をラジアンで返します。

パラメータ:

- **value** - チェックする値

値は-1 から 1 の間でなければなりません。例えば、値'0.5' のアークコサインは'2.0943951' になります。

例:

```
acos(last(/host/key))
```

asin(value)

値のアークサイン（逆正弦）をラジアンで返します。

パラメータ:

- **value** - チェックする値

値は-1 から 1 の間でなければなりません。例えば、値'0.5' のアークサインは'-0.523598776' になります。

例:

```
asin(last(/host/key))
```

atan(value)

値の逆正接を角度としてラジアンで表したものです。

パラメータ:

- **value** - チェックする値

例えば、値'1' の逆正接は'0.785398163' になります。

例:

```
atan(last(/host/key))
```

atan2(value,abscissa)

指定された座標（縦座標（value）と横座標）のアークタンジェントをラジアンで表した角度。

パラメータ:

- **value** - チェックする値
- **abscissa** - 横座標の値

例えば、値'1' の縦座標と横座標のアークタンジェントは'2.21429744' となります。

例:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

参照されたアイテム値の平均値。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

値の立方根。

パラメータ:

- **value** - チェックする値

例えば、'64' の立方根は'4'、'63' の立方根は'3.97905721' となります。

例:

```
cbrt(last(/host/key))
```

ceil(value)

値を最も近い大きいまたは等しい整数に切り上げます。

パラメータ:

- **value** - チェックする値

例えば、'2.4' は'3' に切り上げられます。floor()も参照してください。

例:

```
ceil(last(/host/key))
```

cos(value)

値の余弦を求めます。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値'1' の余弦は'0.54030230586' になります。

例:

```
cos(last(/host/key))
```

cosh(value)

値の双曲線余弦。値を科学的記数法ではなく実数として返します。

パラメータ:

- **value** - チェックする値

例えば、値'1' の双曲線余弦は'1.54308063482' になります。

例:

```
cosh(last(/host/key))
```

cot(value)

値の余接。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値'1' の余接は'0.54030230586' になります。

例:

```
cot(last(/host/key))
```

degrees(value)

値をラジアンから度に変換します。

パラメータ:

- **value** - チェックする値

例えば、値'1' を度に変換すると'57.2957795' になります。

例:

```
degrees(last(/host/key))
```

e

ネイピア数 (2.718281828459045)。

例:

```
e()
```

exp(value)

値の累乗のネイピア数。

パラメータ:

- **value** - チェックする値

例えば、値'2' の累乗のネイピア数は'7.38905609893065' となります。

例:

```
exp(last(/host/key))
```

expm1(value)

値の累乗のオイラー数から 1 を引いた値。

パラメータ:

- **value** - チェックする値

例えば、値'2' の累乗のオイラー数から 1 を引いた値は'6.38905609893065' となります。

例:

```
expm1(last(/host/key))
```

floor(value)

値を最も近い小さいまたは等しい整数に切り捨てます。

パラメータ:

- **value** - チェックする値

例えば、'2.6' は'2' に切り捨てられます。ceil()も参照してください。

例:

```
floor(last(/host/key))
```

log(value)

自然対数。

パラメータ:

- **value** - チェックする値

例えば、値'2' の自然対数は'0.69314718055994529' になります。

例:

log(last(/host/key))

log10(value)

常用対数。

パラメータ:

- **value** - チェックする値

例えば、値'5' の常用対数は'0.69897000433' になります。

例:

log10(last(/host/key))

max(<value1>,<value2>,...)

参照されたアイテムの値の中で最も高い値です。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

max(avg(/host/key,1h),avg(/host2/key2,1h))

min(<value1>,<value2>,...)

参照されたアイテム値の中で最も低い値です。

パラメータ:

- **valueX** - アイテム履歴を扱う別の関数によって返される値。

例:

min(avg(/host/key,1h),avg(/host2/key2,1h))

mod(value,denominator)

除算の余り。

パラメータ:

- **value** - チェックする値
- **denominator** - 除算の分母

例えば、値「5」を除算の分母「2」で割った余りは「1」になります。

例:

mod(last(/host/key),2)

pi

円周率 (3.14159265358979)。

例:

pi()

power(value,power value)

値のべき乗。

パラメータ:

- **value** - チェックする値
- **power value** - 使用する N 乗

例えば、値'2' の 3 乗は'8' になります。

例:

power(last(/host/key),3)

radians(value)

値を度からラジアンに変換します。

パラメータ:

- **value** - チェックする値

たとえば、値「1」をラジアンに変換すると「0.0174532925」になります。

例:

```
radians(last(/host/key))
```

rand

ランダムな整数値を返します。時刻をシードとして使用した疑似乱数 (数学的な目的には十分ですが、暗号化には不十分) です。

例:

```
rand()
```

round(value,decimal places)

値を小数点以下の桁数に丸めます。

パラメータ:

- **value** - チェックする値
- **decimal places** - 丸める小数点以下の桁数を指定します (0 も可能です)。

たとえば、値'2.5482' を小数点以下 2 桁に丸めると'2.55' になります。

例:

```
round(last(/host/key),2)
```

signum(value)

値が負の場合は'-1'、値がゼロの場合は'0'、値が正の場合は'1' を返します。

パラメータ:

- **value** - チェックする値

例:

```
signum(last(/host/key))
```

sin(value)

値のサイン（正弦）を返します。値はラジアンで表される角度です。

パラメータ:

- **value** - チェックする値

例えば、値'1' のサインは'0.8414709848' になります。

例:

```
sin(last(/host/key))
```

sinh(value)

値の双曲線正弦です。

パラメータ:

- **value** - チェックする値

例えば、値「1」の双曲線正弦は「1.17520119364」になります。

例:

```
sinh(last(/host/key))
```

sqrt(value)

値の平方根。
 この関数は負の値では失敗します。

パラメータ:

- **value** - チェックする値

例えば、値'3.5' の平方根は'1.87082869339' になります。

例:

`sqrt(last(/host/key))`

sum(<value1>,<value2>,...)

参照されたアイテムの値の合計です。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

`sum(avg(/host/key,1h),avg(/host2/key2,1h))`

tan(value)

値のタンジェント。

パラメータ:

- **value** - チェックする値

例えば、値'1' のタンジェントは'1.55740772465' になります。

例:

`tan(last(/host/key))`

truncate(value,decimal places)

値を小数点以下で切り捨てます。

パラメータ:

- **value** - チェックする値
- **decimal places** - 切り捨てる小数点以下の桁数を指定します（0 も可能です）。

例えば、値'2.5482' を小数点以下 2 桁で切り捨てると'2.54' になります。

例:

`truncate(last(/host/key),2)`

サポートされているすべての関数を参照してください。

演算子関数

ここに記載されているすべての関数は、以下でサポートされています。

- トリガーの条件式
- 計算アイテムの数式
- 式マクロ

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

関数	説明
between	値が指定された範囲に含まれているか確認します。
in	値が一覧表示された値のうち少なくとも 1 つと等しいか確認します。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切られます
- パラメータとして式が受け入れられます

between(value,min,max)

値が指定された範囲に属しているかどうかをチェックします。
 サポートされている値の型: *Integer*, *Float*。
 戻り値: 1 - 範囲内; 0 - それ以外。

パラメータ:

- **value** - チェックする値
- **min** - 最小値

- **max** - 最大値

例:

```
between(last(/host/key),1,10)=1 #値が1から10の間ならトリガー
```

in(value,value1,value2,...valueN)

値がリストされた値のいずれかと等しいかどうかを確認します。
 サポートされている値の型: 整数、浮動小数点、文字列、テキスト、ログ。
 戻り値: 1 - 等しい場合; 0 - それ以外の場合。

パラメータ:

- **value** - チェックする値;
- **valueX** - リストされた値 (文字列値はダブルクォートで囲む必要があります)。

値は、すべての値が数値に変換できる場合は数値として比較されます。それ以外の場合は文字列として比較されます。

例:

```
in(last(/host/key),5,10)=1 #最後の値が5または10の場合にトリガー
in("text",last(/host/key),last(/host/key,#2))=1 #"text"が直近2つの値のいずれかと等しい場合にトリガー
```

サポートされているすべての関数を参照してください。

予測関数

ここに一覧表示されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテムの数式
- 式マクロ

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

Function	Description
forecast	アイテムの将来値、最大値、最小値、差分、または平均値。
timeleft	アイテムが指定したしきい値に達するまでに必要な秒数。

共通パラメータ

- /host/key は、ホストのアイテム履歴を参照する関数に共通する必須の第 1 パラメータです。
- (sec|#num)<:time shift> は、ホストのアイテム履歴を参照する関数に共通する第 2 パラメータです。次のように指定します。
  - **sec** - 秒単位の最大評価期間（時間のサフィックスを使用できます）、または
  - **#num** - 直近で収集された値における最大評価範囲（ハッシュ記号が前に付いている場合）。
  - **time shift**（オプション）- 評価時点を過去に移動できます。time shift の指定方法については、詳細を参照してください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示します
- 関数固有のパラメータは各関数で説明します
- /host/key および (sec|#num)<:time shift> パラメータは決して引用符で囲まないでください

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

アイテムの将来の値、最大値、最小値、差分、または平均値を返します。
 サポートされている値のタイプ: *Float*、*Integer*。

パラメータ:

- 共通パラメータを参照してください。
- **time** - 予測期間を秒単位で指定します（例: 1、60）。時間のサフィックスも使用できます（例: 30s、1m）。負の値もサポートされます。
- **fit**（オプション。二重引用符で囲む必要があります）- 履歴データの近似に使用する関数です。サポートされている近似方法:
linear - 線形関数（デフォルト）
polynomialN - N 次多項式（1 <= N <= 6）
exponential - 指数関数
logarithmic - 対数関数
power - べき関数
polynomial1 は *linear* と同等です。

- **mode**（オプション。二重引用符で囲む必要があります）- 要求する出力です。サポートされているモード:
value - 値（デフォルト）
max - 最大値
min - 最小値
delta - *max-min*
avg - 平均値
value は時刻 now + time におけるアイテムの値を推定します。一方、*max*、*min*、*delta*、*avg* は、now から now + time までの期間におけるアイテムの推定値を調べます。

コメント:

- 返される値が 1.7976931348623158E+308 より大きい場合、または-1.7976931348623158E+308 より小さい場合、返り値はそれぞれ 1.7976931348623158E+308 または-1.7976931348623158E+308 に切り詰められます。
- 式で誤って使用された場合（アイテムのタイプが不正、パラメータが無効など）のみサポート対象外になります。それ以外の場合、エラー時には-1 を返します。
- 予測トリガー関数の追加情報も参照してください。

例:

```
forecast(/host/key,#10,1h) #過去10個の値に基づいて、1時間後のアイテムの値を予測
forecast(/host/key,1h,30m) #過去1時間のデータに基づいて、30分後のアイテムの値を予測
forecast(/host/key,1h:now-1d,12h) #1日前の1時間分のデータに基づいて、12時間後のアイテムの値を予測
forecast(/host/key,1h,10m,"exponential") #過去1時間のデータと指数関数に基づいて、10分後のアイテムの値を予
forecast(/host/key,1h,2h,"polynomial3","max") #過去1時間のデータと3次多項式に基づいて、今後2時間にアイテム
forecast(/host/key,#2,-20m) #過去2個の値に基づいて、20分前のアイテムの値を推定（特にアイテムの更新頻度が低
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

アイテムが指定したしきい値に到達するまでに必要な秒数です。
 サポートされる値のタイプ: *Float*、*Integer*。

パラメータ:

- 共通パラメータを参照してください。
- **threshold** - 到達する値（単位のサフィックスを使用できます）。
- **fit**（オプション。二重引用符で囲む必要があります）- forecast()を参照してください。

コメント:

- 返される値が 1.7976931348623158E+308 より大きい場合、戻り値は 1.7976931348623158E+308 に切り詰められます。
- しきい値に到達できない場合は、1.7976931348623158E+308 を返します。
- 式で誤って使用した場合（アイテムのタイプが不正、パラメータが無効など）にのみサポート対象外になります。それ以外の場合、エラー時には-1 を返します。
- 予測トリガー関数の追加情報も参照してください。

例:

```
timeleft(/host/key,#10,0) #最後の10個の値に基づいて、アイテムの値が0に到達するまでの時間
timeleft(/host/key,1h,100) #過去1時間のデータに基づいて、アイテムの値が100に到達するまでの時間
timeleft(/host/key,1h:now-1d,100) #1日前の1時間のデータに基づいて、アイテムの値が100に到達するまでの時間
timeleft(/host/key,1h,200,"polynomial2") #過去1時間のデータと、アイテムが二次（2次）多項式のように動作する
```

サポートされているすべての関数を参照してください。

予測トリガー関数

概要

時には、問題が発生する前兆が現れることがあります。これらの兆候を見つけることで、問題の影響を防ぐ、あるいは少なくとも最小限に抑えるための対策を事前に講じることができます。

Zabbix には、過去のデータに基づいて監視対象システムの将来の動作を予測するためのツールがあります。これらのツールは、予測トリガー関数として実現されています。

関数

トリガーを設定する前に、問題状態が何であるか、そしてアクションを実行するのにどれだけの時間が必要かを定義する必要があります。その後、潜在的な望ましくない状況を通知するトリガーを設定する方法は 2 つあります。1 つ目は、「アクションを実行するまでの時間」後にシステムが問題状態になると予想される場合にトリガーを発火させる方法です。2 つ目は、システムが「アクションを実行するまでの時間」未満で問題状態に到達しそうな場合にトリガーを発火させる方法です。対応するトリガー関数は **forecast** と **timeleft** です。両関数で使用される統計分析は基本的に同じであることに注意してください。どちらの方法でも、ほぼ同じ結果でトリガーを設定できます。

パラメータ

両方の関数はほぼ同じパラメータセットを使用します。参照用にサポートされている関数のリストを使用してください。

時間間隔

まず、Zabbix が予測を行うために分析する履歴期間を指定する必要があります。これは、**avg**、**count**、**delta**、**max**、**min**、**sum** 関数と同様に、時間間隔パラメータとオプションの時間シフトを使用して指定します。

予測期間

(**forecast** のみ）
 パラメータ `time` は、履歴データ内で見つかった依存関係を、Zabbix が将来のどこまで外挿するかを指定します。`time_shift` を使用するかどうかにかかわらず、`time` は常に現在時刻を起点としてカウントされます。

到達すべきしきい値

(**timeleft** のみ）
 パラメータ `threshold` は、解析対象のアイテムが到達しなければならない値を指定します。上側から到達するか下側から到達するかの違いはありません。f(t) が決定されたら（以下を参照)、方程式 f(t) = `threshold` を解き、現在に最も近く、かつ現在より右側にある根を返す必要があります。そのような根が存在しない場合は 1.7976931348623158E+308 を返します。

> **Note:**
> アイテムの値がしきい値に近づいた後にそれを横切る場合、**timeleft** はその交点はすでに過去にあるとみなし、そのため、存在する場合は `threshold` レベルとの次の交点に切り替えます。ベストプラクティスとして、予測は通常の障害診断の代替ではなく、補完として使用する必要があります。[1]

フィット関数

デフォルトの `fit` は *linear* 関数です。しかし、監視対象のシステムがより複雑な場合は、他のオプションを選択することもできます。

`fit`	x = f(t)
linear	$x = a + b*t$
polynomialN[2]	$x = a_0 + a_1*t + a_2*t^2 + ... + a_n*t^n$
exponential	$x = a*exp(b*t)$
logarithmic	$x = a + b*log(t)$
power	$x = a*t^b$

モード

(**forecast** のみ)
 トリガー関数が評価されるたびに、指定された履歴期間のデータを取得し、そのデータに指定された関数を当てはめます。したがって、データがわずかに異なれば、当てはめられる関数もわずかに異なります。単純に将来の指定時刻における当てはめ関数の値を計算するだけでは、現在からその将来時点までの間に、分析対象のアイテムがどのように振る舞うと予想されるかは分かりません。`fit` の一部のオプション（*polynomial* など）では、将来の単純な値は誤解を招く可能性があります。

`mode`	**forecast** の結果
value	f(now + `time`)
max	$max_{now <= t <= now + time}$ f(t)
min	$min_{now <= t <= now + time}$ f(t)
delta	*max - min*
avg	定義 に従った f(t) の平均値（now <= t <= now + `time`）

詳細

巨大な数値での計算を避けるために、指定された期間の最初の値のタイムスタンプに 1ns を加えたものを新しいゼロタイム（現在のエポックタイムは 10^9 オーダー、エポックの 2 乗は 10^{18}、倍精度は約 10^{-16}）とみなします。対数やべき乗近似では log(t) の計算が含まれるため、すべての時間値を正にするために 1ns を加えます。時間のシフトは線形、多項式、指数（より簡単で正確な計算を除く）には影響しませんが、対数やべき乗関数の形状を変化させます。

起こりうるエラー

関数は以下のような場合に-1 を返します:

- 指定した評価期間にデータが存在しない場合
- 数学的な演算の結果が定義されていない場合[3]
- 数値的な問題（残念ながら、入力データの組み合わせによっては、倍精度浮動小数点形式の範囲や精度が不十分になる場合があります）[4]

Note:
選択したフィットが提供されたデータをうまく表現できない場合や、正確な予測に十分なデータがない場合でも、警告やエラーは表示されません。

例とエラーへの対処

ホストのディスク空き容量が不足しそうな場合に警告を受け取るには、次のようなトリガー式を使用できます。

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

ただし、エラーコード-1 が発生し、トリガーが障害状態になる場合があります。一般的には、予測が正しく機能していないことを警告してくれるので良いことですが、なぜそうなったのかを詳しく調べる必要があります。しかし、-1 は単に過去 1 時間以内にホストのディスク空き容量のデータが取得できなかったことを意味する場合もあります。誤検知のアラートが多すぎる場合は、より複雑なトリガー式の使用を検討してください [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

forecast の場合は少し状況が複雑です。まず、-1 がトリガーを障害状態にするかどうかは、`forecast(/host/item,(...))<...` のような式か、`forecast(/host/item,(...))>...` のような式かによって異なります。

さらに、アイテム値が負になることが通常であれば、-1 は有効な予測値となる場合もあります。しかし、現実の運用ではこのような状況はほとんどありません（演算子 = の動作についてはこちらを参照）。したがって、-1 を障害とみなしたい場合は `... or forecast(/host/item,(...))=-1` を、みなしたくない場合は `... and forecast(/host/item,(...))<>-1` を追加してください。

脚注

[1] 例えば、`timeleft(/host/item,1h,X) < 1h` のような単純なトリガーは、アイテム値が X に近づくと障害状態になり、値が X に達すると突然復旧する場合があります。問題がアイテム値が X 未満である場合は、`last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` を、アイテム値が X より大きい場合は `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h` を使用してください。

[2] 多項式の次数は 1 から 6 まで指定でき、*polynomial1* は *linear* と同じです。ただし、高次の多項式は注意して使用してください。評価期間内のポイント数が多項式係数の決定に必要な数より少ない場合は、多項式の次数が下げられます（例：*polynomial5* が指定されているがポイントが 4 つしかない場合、*polynomial3* が適用されます）。

[3] 例えば、*exponential* や *power* 関数のフィッティングでは、アイテム値の log() 計算が必要です。データにゼロや負の値が含まれている場合、log() は正の値にしか定義されていないためエラーになります。

[4] *linear*、*exponential*、*logarithmic*、*power* の各フィットでは、必要な計算をすべて明示的に記述できます。*polynomial* では *value* のみ追加の手順なしで計算できます。*avg* の計算には多項式の不定積分（解析的）が必要です。*max*、*min*、*delta* の計算には多項式の導関数（解析的）とその根の探索（数値的）が必要です。f(t) = 0 の解法には多項式の根の探索（数値的）が必要です。

[5] ただし、この場合-1 によってトリガーが障害状態から復旧することがあります。完全に保護したい場合は、`timeleft(/host/vfs.fs.size[/, and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1 or {TRIGGER.VALUE}=1)` を使用してください。

文字列関数

ここに記載されているすべての関数は、以下でサポートされています。

- トリガーの条件式
- 計算アイテムの数式
- 条件式マクロ

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

関数	説明
ascii	値の左端の文字の ASCII コード。
bitlength	値のビット単位の長さ。
bytelength	値のバイト単位の長さ。
char	値を ASCII コードとして解釈し、文字を返します。
concat	参照されたアイテムの値または定数値を連結した結果の文字列。
contains	値に `search value` で指定した部分文字列が含まれている場合は `1`、それ以外の場合は `0` を返します。
insert	文字列内の指定した位置から、指定した文字またはスペースを文字列に挿入します。
jsonpath	JSONPath の結果を返します。
left	値の左端の文字を返します。
length	値の文字単位の長さ。

関数	説明
ltrim	文字列の先頭から指定した文字を削除します。
mid	`start` で指定した文字位置から始まる N 文字の部分文字列を返します。この関数は `substring()` 関数の別名です。
repeat	文字列を繰り返します。
replace	値内のパターンを検索し、置換文字列に置き換えます。
right	値の右端の文字を返します。
rtrim	文字列の末尾から指定した文字を削除します。
substring	`start` で指定した文字位置から始まる N 文字の部分文字列を返します。
trim	文字列の先頭と末尾から指定した文字を削除します。
xmlxpath	XML XPath の結果を返します。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切られます
- パラメータとして式が使用できます
- 文字列パラメータはダブルクォーテーションで囲む必要があります。そうしないと誤解釈される可能性があります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます

ascii(value)

値の最左文字の ASCII コード。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値

例えば、'Abc' のような値は'65'（'A' の ASCII コード）を返します。

例:

```
ascii(last(/host/key))
```

bitlength(value)

value のビット長。
 サポートされている値の型: *String*, *Text*, *Log*, *Integer*。

パラメータ:

- **value** - チェックする値

例:

```
bitlength(last(/host/key))
```

bytelength(value)

値のバイト長。
 サポートされている値の型: *String*, *Text*, *Log*, *Integer*。

パラメータ:

- **value** - チェックする値

例:

```
bytelength(last(/host/key))
```

char(value)

値を ASCII コードとして解釈して文字を返します。
 サポートされている値の型: 整数。

パラメータ:

- **value** - チェックする値

値は 0-255 の範囲でなければなりません。例えば、'65' という値（ASCII コードとして解釈）は'A' を返します。

例:

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

参照されたアイテムの値または定数値を連結した結果の文字列。
 サポートされている値の型: *String*, *Text*, *Log*, *Float*, *Integer*。

パラメータ:

- **valueX** - ヒストリ関数のいずれかによって返される値、または定数値 (文字列、整数、または浮動小数点数)。少なくとも 2 つのパラメータを含める必要があります。

たとえば、「Zab」のような値に「bix」(定数文字列) を連結すると、「Zabbix」が返されます。

例:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15])
```

contains(value, search value)

値に search value で指定した部分文字列が含まれている場合は 1、それ以外の場合は 0 を返します。| サポートされる値のタイプ: *String*、*Text*、*Log*。

パラメータ:

- **value** - チェックする値。
- **search value** - 検索する値。

コメント:

- 大文字と小文字は区別されます。
- 値が有効で、search value が空の場合、関数は 1 を返します。

例えば、値に「Zabbix」が含まれている場合、関数は 1 を返します。

例:

```
contains(last(/host/key),"Zabbix")
```

insert(value,start,length,replacement)

指定した位置から文字列に指定した文字またはスペースを挿入します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **start** - 開始位置
- **length** - 置換する位置数
- **replacement** - 置換文字列

例えば、'Zabbbix' の値は、'bb'（開始位置 3、置換する位置数 2）を'b' に置換すると'Zabbix' に置換されます。

例:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

JSONPath の結果を返します。
 サポートされている値の型: *String*、*Text*、*Log*。

パラメータ:

- **value** - チェックする値;
- **path** - パス（引用符で囲む必要があります）;
- **default** - JSONPath クエリがデータを返さない場合のオプションのフォールバック値。他のエラーの場合は失敗が返されます (例："unsupported construct")。

例:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

値の左端の文字を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **count** - 返す文字数

たとえば、返す左端の文字数を 3 に指定することで、'Zabbix' から'Zab' を返すことができます。right()も参照してください。

例:

```
left(last(/host/key),3) #左端の3文字を返す
```

length(value)

値の文字数。
 サポートされている値の型：*String*、*Text*、*Log*。

パラメータ:

- **value** - チェックする値

例:

```
length(last(/host/key)) #最新値の文字数
length(last(/host/key,#3)) #3番目に新しい値の文字数
length(last(/host/key,#1:now-1d)) #1日前の最新値の文字数
```

ltrim(value,<chars>)

文字列の先頭から指定した文字を削除します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **chars** (省略可能) - 削除する文字を指定

省略可能な文字が指定されていない場合は、デフォルトで空白文字が左側から削除されます。詳細はrtrim()、trim() も参照してください。

例:

```
ltrim(last(/host/key)) #文字列の先頭から空白を削除
ltrim(last(/host/key),"Z") #文字列の先頭から'Z'をすべて削除
ltrim(last(/host/key)," Z") #文字列の先頭から空白と'Z'をすべて削除
```

mid(value,start,length)

「start」で指定した文字位置から始まる、N 文字の部分文字列を返します。
 この関数は `substring()` 関数のエイリアスです。
 サポートされている値のタイプ: *String*、*Text*、*Log*。

パラメータ:

- **value** - チェックする値;
- **start** - 部分文字列の開始位置;
- **length** - 部分文字列で返す文字数。

たとえば、開始位置が 2 で、返す文字数が 4 の場合、「Zabbix」のような値から「abbi」を返すことができます。

例:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

文字列を繰り返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **count** - 繰り返す回数

例:

```
repeat(last(/host/key),2) #値を2回繰り返す
```

replace(value,pattern,replacement)

値の中からパターンを見つけて、replacement で置き換えます。パターンに一致するすべての箇所が置き換えられます。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **pattern** - 検索するパターン
- **replacement** - パターンを置き換える文字列

例:

```
replace(last(/host/key),"ibb","abb") # 'ibb'をすべて'abb'に置き換える
```

right(value,count)

値の右端の文字を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **count** - 返す文字数。

例えば、右端から 3 文字を返すことで'Zabbix' から'bix' を返すことができます。left() も参照してください。

例:

```
right(last(/host/key),3) #右端から3文字を返す
```

rtrim(value,<chars>)

文字列の末尾から指定した文字を削除します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値
- **chars** (省略可能) - 削除する文字を指定

省略可能な文字が指定されていない場合は、デフォルトで空白が右側から削除されます。詳細はltrim()、trim() も参照してください。

例:

```
rtrim(last(/host/key)) #文字列の末尾から空白を削除
rtrim(last(/host/key),"x") #文字列の末尾から'x'を削除
rtrim(last(/host/key),"x ") #文字列の末尾から'x'と空白を削除
```

substring(value,start,length)

「start」で指定した文字位置から始まる、N 文字の部分文字列を返します。
 サポートされる値のタイプ: *String*、*Text*、*Log*。

パラメータ:

- **value** - チェックする値;
- **start** - 部分文字列の開始位置;
- **length** - 部分文字列で返す文字数。

たとえば、開始位置が 2 で、返す文字数が 4 の場合、「Zabbix」のような値から「abbi」を返すことができます。

例:

```
substring(last(/host/key),2,4)="abbi"
```

trim(value,<chars>)

文字列の先頭と末尾から指定した文字を削除します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **chars** (省略可能) - 削除する文字を指定します。

デフォルトでは、(省略可能な文字が指定されていない場合) 両端の空白がトリミングされます。関連項目: ltrim(), rtrim()。

例:

```
trim(last(/host/key)) #文字列の先頭と末尾から空白を削除
trim(last(/host/key),"_") #文字列の先頭と末尾から'_'を削除
```

xmlxpath(value,path,<default>)

XML XPath の結果を返します。
 サポートされている値の型: *String*, *Text*, *Log*。

パラメータ:

- **value** - チェックする値;
- **path** - パス (引用符で囲む必要があります);
- **default** - XML XPath クエリが空のノードセットを返した場合のオプションのフォールバック値。空の結果がノードセットでない場合 (つまり空文字列)、返されません。他のエラーの場合は失敗が返されます (例: "invalid expression")。

例:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

サポートされているすべての関数を参照してください。

トリガーのベストプラクティス

トリガーは強力なツールですが、不要なアラートノイズを発生させることもあります。より実際のシグナルを増やし、ノイズを減らすには、次のヒントを参考にしてください。

1. トリガーの感度を下げます。最新値（高すぎる/低すぎる）でアラートを出す代わりに、**avg**、**min**、**max** などの関数を使って、長期間の平均を分析します。
2. ランダムなスパイクや急落によるアラートを避けたい場合は、トリガーで **percentile** 関数を使用することを検討してください（95% または 5% に設定します）。
3. トリガーのフラッピングを避けるためにヒステリシス を使用します。フラッピングとは、トリガー状態が頻繁に変化すること（OK ↔ 障害）です。回復式（障害解決のための別条件）を追加することで、障害状態の継続条件を定義できます。
4. 根本原因に関連しないアラートを避けるために、トリガーの依存関係を使用します。
5. より深刻な障害のみをアラートするために、トリガーの深刻度を使用します。
6. メンテナンス期間を定義します。

ヒステリシス

単純な閾値ではなく、障害状態と復旧状態の間に間隔が必要な場合があります。たとえば、サーバールームの温度が 20°C を超えたときに障害を報告し、温度が 15°C 未満になるまで障害状態を維持したい場合、単純なトリガー閾値を 20°C に設定するだけでは不十分です。

その代わりに、まず障害イベント用のトリガー式（温度が 20°C を超えた場合）を定義する必要があります。次に、追加の復旧条件（温度が 15°C 未満）を定義する必要があります。これは、トリガーを設定する際に復旧式を定義することで実現します。

この場合、障害の復旧は 2 段階で行われます。

- まず、障害式（温度が 20°C を超えた場合）が FALSE と評価される必要があります
- 次に、復旧式（温度が 15°C 未満）が TRUE と評価される必要があります

復旧式は、障害イベントが解決された後にのみ評価されます。障害式がまだ TRUE の場合、復旧式が TRUE であるだけでは障害は解決されません！

例

サーバールームの温度が高すぎます。

障害の式:

```
last(/server/temp)>20
```

復旧の式:

```
last(/server/temp)<=15
```

> **Note:**
> 復旧の式で {TRIGGER.VALUE} マクロを使用しても非効率的です。なぜなら、この式はトリガーが「障害」状態のときだけ評価されるためです。そのため、式を評価している間、{TRIGGER.VALUE} は常に「1」（「障害」状態を示す）に解決されます。

トリガーの依存関係

トリガーの依存関係は、根本原因に関連しないアラートを回避するために使用できます。

すべてのベストプラクティスを参照してください。

概要

あるホストの可用性が別のホストに依存する場合があります。ルーターの背後にあるサーバーは、ルーターが停止すると到達不能になります。両方にトリガーを設定していると、実際にはルーターだけが原因であるにもかかわらず、2 台のホストが停止したという通知を受け取る可能性があります。

このような場合、ホスト間に依存関係を設定すると便利です。依存関係を設定すると、依存するホストの通知を保留し、ルート障害についてのみ通知を送信できます。

Zabbix はホスト間の依存関係を直接サポートしていませんが、より柔軟な別の方法であるトリガー依存関係を使用して定義できます。1 つのトリガーは、1 つ以上のトリガーに依存できます。

この簡単な例では、サーバーのトリガー設定フォームを開き、ルーターの対応するトリガーに依存するように設定します。この依存関係を設定すると、依存先のトリガーが「PROBLEM」状態である限り、サーバーのトリガーは状態を変更しません。そのため、依存するアクションは実行されず、通知も送信されません。

サーバーとルーターの両方が停止しており、「サーバー停止」トリガーと「ルーター停止」トリガーの間に依存関係が設定されている場合、Zabbix は依存するトリガーに対するアクションを実行しません。

親トリガーが PROBLEM 状態にある間、依存するトリガーは信頼できない値を報告する可能性があります。そのため、依存するトリガーは、親トリガー（上記の例ではルーター）が次のいずれかになるまで再評価されません。

- 「PROBLEM」状態から「OK」状態に戻る
- 状態が「PROBLEM」から「UNKNOWN」に変わる
- date and time関数やnodata()関数を使用して、手動、相関、または関数によってクローズされる
- 依存するトリガーに関与していないアイテムの値によって解決される
- 無効になる、無効なアイテムを持つ、または無効なアイテムのホストになる

上記のいずれの場合も、依存するトリガー（サーバー）は、新しいメトリクスを受信したときにのみ再評価されます。つまり、依存するトリガーはすぐには更新されない場合があります。

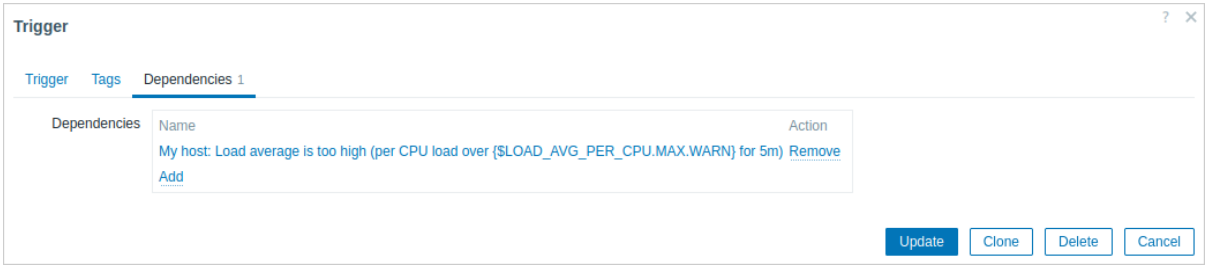
また、次の点にも注意してください。

- 循環依存にならない限り、任意のホストのトリガーから別の任意のホストのトリガーにトリガー依存関係を追加できます。
- あるテンプレートから別のテンプレートにトリガー依存関係を追加できます。テンプレート A のトリガーがテンプレート B のトリガーに依存する場合、テンプレート A はテンプレート B と一緒にのみホスト（または別のテンプレート）にリンクできます。一方、テンプレート B は単独でホスト（または別のテンプレート）にリンクできます。
- テンプレートのトリガーからホストのトリガーにトリガー依存関係を追加できます。この場合、そのテンプレートをホストにリンクすると、トリガーが依存していた同じトリガーテンプレートに依存するホストトリガーが作成されます。これにより、例えば、一部のトリガーがルーター（ホスト）のトリガーに依存するテンプレートを作成できます。このテンプレートにリンクされたすべてのホストは、その特定のルーターに依存します。
- ホストのトリガーからテンプレートのトリガーにトリガー依存関係を追加することはできません。
- トリガープロトタイプから、別のトリガープロトタイプ（同じローレベルディスカバリールール内）または実際のトリガーにトリガー依存関係を追加できます。トリガープロトタイプは、別の LLD ルールのトリガープロトタイプや、トリガープロトタイプから作成されたトリガーに依存することはできません。ホストのトリガープロトタイプは、テンプレートのトリガーに依存できません。

設定

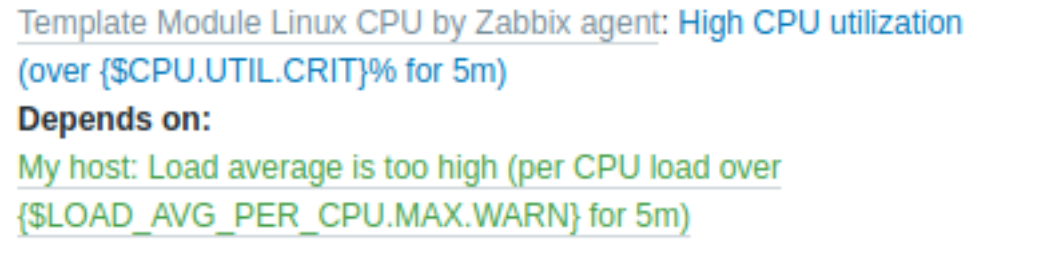
依存関係を定義するには、次の手順に従います。

1. トリガー設定フォームの依存関係タブを開きます。
2. 依存関係セクションで追加をクリックし、そのトリガーが依存する 1 つ以上のトリガーを選択します。



3. 更新をクリックします。

これで、トリガーの依存関係が一覧に表示されます。



複数の依存関係の例

たとえば、ホストは Router2 の背後にあり、Router2 は Router1 の背後にあるとします。

```
Zabbix - Router1 - Router2 - Host
```

Router1 がダウンしている場合、当然ながらホストと Router2 も到達不能ですが、ホスト、Router1、Router2 の 3 つすべてがダウンしているという通知を受け取るのは過剰です。

そのため、この場合は 2 つの依存関係を定義します。

```
the 'Host is down' trigger depends on the 'Router2 is down' trigger
the 'Router2 is down' trigger depends on the 'Router1 is down' trigger
```

'Host is down' トリガーの状態を変更する前に、Zabbix は対応するトリガーの依存関係を確認します。そのような依存関係が見つかり、そのうちのいずれかのトリガーが **Problem** 状態である場合、トリガーの状態は変更されず、アクションは実行されず、通知も送信されません。

Zabbix はこの確認を再帰的に行います。Router1 または Router2 に到達できない場合、Host のトリガーは更新されません。

トリガーの深刻度

トリガーの深刻度は、あまり重大でない問題によるアラートのノイズを回避するために使用できます。

すべてのベストプラクティスを参照してください。

概要

深刻度は、トリガーの相対的な重要度を表します。



深刻度の割り当ては、視覚的な表現だけでなく、些細な問題に関する通知を回避するためにも役立ちます。

トリガーの深刻度は、以下の目的で使用されます。

- 視覚的な表現（作成された各問題ごとに、元のトリガーの深刻度に基づいた異なる色）
- 深刻度に基づく通知の制限（アクション条件を参照）
- ユーザーメディア - 深刻度ごとに異なるメディア（通知チャネル）（例：高および致命的の深刻度のトリガーには SMS、他の深刻度のトリガーにはメール）
- グローバルアラームの音声 - 深刻度ごとに異なる音声

デフォルトでサポートされているトリガーの深刻度は以下の通りです。

深刻度	色	説明
未分類	グレー	イベントの深刻度レベルが不明、未決定、通常の監視範囲外などの場合に使用できます。例えば、初期設定時、将来の評価のためのプレースホルダー、または統合プロセスの一部として使用します。
情報	ライトブルー	即時の対応を必要としないが、有益な洞察を提供できる情報イベントに使用できます。
警告	イエロー	調査や対応が必要となる可能性があるが、重大ではない潜在的な問題を示すために使用できます。
軽度障害	オレンジ	さらなる問題を防ぐために比較的早期に対処すべき重要な問題を示すために使用できます。
重度障害	ライトレッド	重大な障害を回避するために即時の対応が必要なクリティカルな問題を示すために使用できます。
致命的	レッド	例えば、システム停止やデータ損失を防ぐために即時の対応が必要な深刻なインシデントを示すために使用できます。

トリガーの深刻度の名称と色は、管理 > 一般 > トリガー表示オプションでカスタマイズできます。
色はすべての Web インターフェーステーマで共有されます。

カスタマイズした深刻度名の翻訳

> **Attention:**
> Zabbix Web インターフェースの翻訳を使用している場合、カスタマイズした深刻度名がデフォルトで翻訳済みの名前より優先されます。

デフォルトのトリガー深刻度名は、すべてのロケールで翻訳可能です。深刻度名を変更した場合、すべてのロケールでカスタム名が使用され、追加で手動翻訳が必要になります。

カスタム深刻度名の翻訳手順:

- 必要なカスタム深刻度名（例: 'Important'）を設定する
- <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po を編集する
- 2 行追加する:

```
msgid "Important"
msgstr "<translation string>"
```

ファイルを保存します。

- <frontend_dir>/locale/README に記載されているように.mo ファイルを作成します

ここで **msgid** は新しいカスタム深刻度名と一致し、**msgstr** は特定の言語での翻訳となります。

この手順は、深刻度名を変更するたびに実施する必要があります。

可視化

概要

可視化機能では、Zabbix が収集した監視データをさまざまな方法で表示できます。グラフは傾向やメトリクスを示し、ネットワークマップはインフラストラクチャのトポロジーを表示し、ダッシュボードはカスタマイズ可能なウィジェットを組み合わせて包括的な監視ビューを提供します。

可視化ツール

- グラフ - アイテムデータをシンプルな自動生成グラフとして表示したり、カスタムグラフを作成したり、アドホックグラフで複数のアイテムを比較したりできます。
- ネットワークマップ - カスタマイズ可能な要素やインジケーターを使用して、ネットワークインフラストラクチャの視覚的な表現を作成します。
- ダッシュボード - さまざまな形式でデータを可視化する多様なダッシュボードウィジェットを組み合わせて、レイアウトをカスタマイズします。

グラフ

概要

Zabbix に大量のデータが流入すると、数値だけを見るよりも、何が起きているかを視覚的に確認できるほうが、ユーザーにとってはるかに分かりやすくなります。

そこで役立つのがグラフです。グラフを使用すると、データの流れを一目で把握したり、障害を関連付けたり、何かがいつ始まったのかを発見したり、何かが障害に発展する可能性のある時期を示したりできます。

Zabbix では、次の機能を利用できます。

- 1 つのアイテムのデータを表示する組み込みのシンプルグラフ。
- より複雑なカスタムグラフを作成する機能。
- アドホックグラフで複数のアイテムをすばやく比較する機能。
- 最新のカスタマイズ可能なベクター形式のグラフおよび円グラフ。

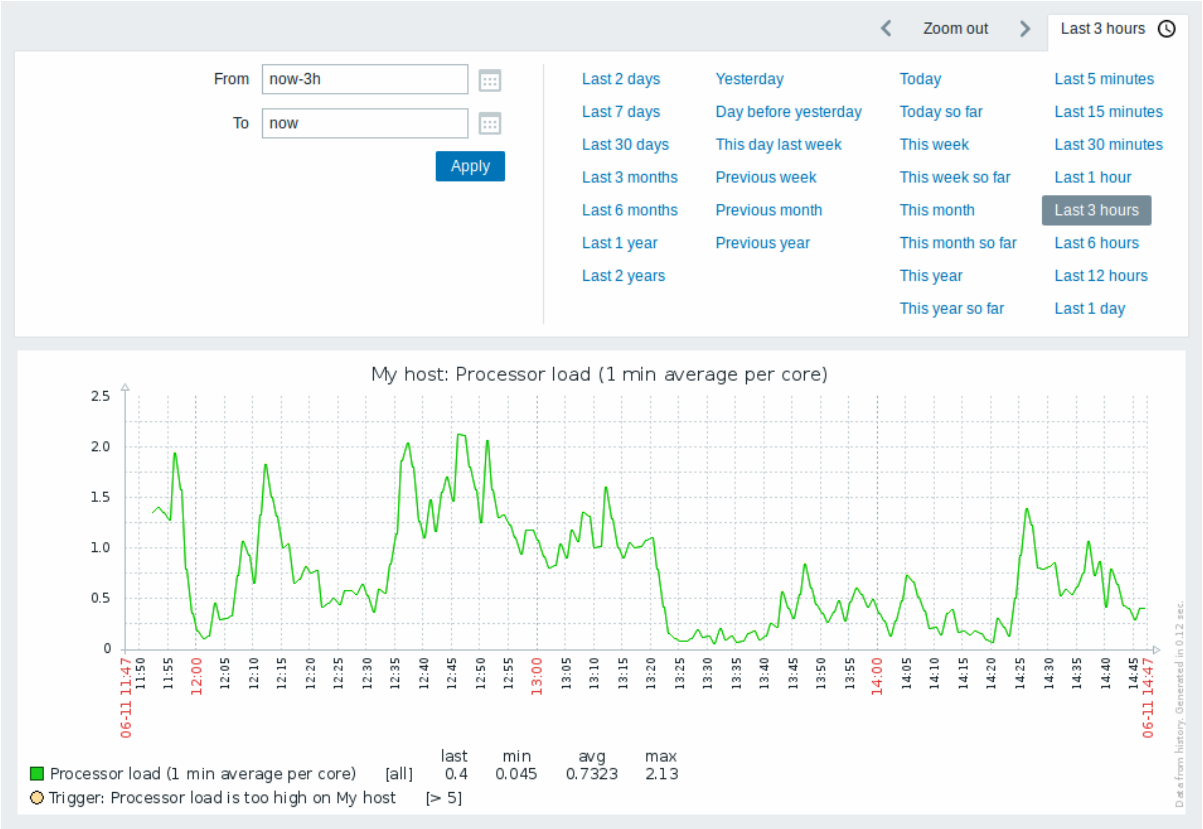
シンプルグラフ

概要

アイテムによって収集されたデータを可視化するために、シンプルグラフが提供されています。

シンプルグラフを表示するために、ユーザー側で設定を行う必要はありません。シンプルグラフは、Zabbix によって無償で提供されています。

監視 > 最新データに移動し、対象のアイテムの グラフリンクをクリックするだけで、グラフが表示されます。



Note:
シンプルグラフは、すべての数値アイテムに対して提供されます。テキストアイテムの場合は、監視 > 最新データに 履歴へのリンクが表示されます。

期間セレクター

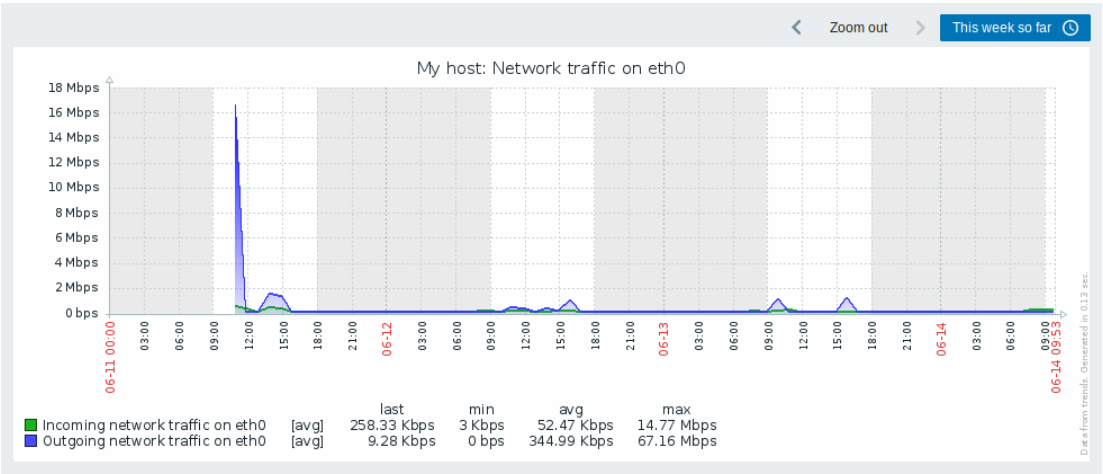
グラフ上部の期間セレクターでは、よく使用する期間をマウスで 1 回クリックするだけで選択できます。詳細については、期間およびホストセレクターを参照してください。

直近のデータと長期間のデータの違い

ごく直近のデータでは、受信した各値を結ぶ **1** 本の線が描画されます。この 1 本の線は、1 つの値に対して少なくとも 1 つの水平方向のピクセルを使用できる限り描画されます。

より長い期間を示すデータでは、**3** 本の線が描画されます。濃い緑色の線は平均値を示し、薄いピンク色と薄い緑色の線は、その時点における最大値と最小値を示します。最大値と最小値の間の領域は、黄色の背景で塗りつぶされます。

グラフでは、稼働時間（稼働日）は白い背景で表示され、非稼働時間は灰色で表示されます（デフォルトの Web インターフェーステーマ *Original blue* の場合）。



稼働時間はシンプルグラフでは常に表示されますが、カスタムグラフで表示するかどうかはユーザー設定によります。

グラフの表示期間が 3 か月を超える場合、稼働時間は表示されません。

トリガーライン

シンプルトリガーは、トリガーの深刻度の色の上に黒い破線のラインとして表示されます。グラフ上の青いラインと、凡例に表示されるトリガー情報に注目してください。
グラフには最大 3 本のトリガーラインを表示できます。トリガーがそれより多い場合は、深刻度の低いトリガーが優先されます。
トリガーはシンプルグラフでは常に表示されますが、カスタムグラフで表示するかどうかはユーザー設定によります。



履歴/トレンドからの生成

グラフは、アイテムの履歴またはトレンドに基づいて描画できます。

Web インターフェースでデバッグモードを有効にしているユーザーには、データの取得元を示す灰色の縦書きのキャプションがグラフの右下に表示されます。

履歴とトレンドのどちらを使用するかは、いくつかの要因によって決まります。

- アイテム履歴の保存期間。例えば、アイテム履歴を 14 日間保存するように設定できます。この場合、14 日より古いデータはトレンドから取得されます。
- グラフ内のデータ量。横方向のグラフの 1 ピクセルに表示される秒数が 3600/16 を超える場合、アイテム履歴が同じ期間に対してまだ利用可能であっても、トレンドデータが表示されます。
- トレンドが無効になっている場合、アイテム履歴がその期間に対して利用可能であれば、グラフの作成に使用されます。

データの欠如

通常の更新間隔を持つアイテムの場合、アイテムデータが収集されていない場合はグラフに何も表示されません。

ただし、トラッパーアイテムやスケジュールされた更新間隔（通常の更新間隔が 0 に設定されている）を持つアイテムの場合、最初に収集された値までと最後に収集された値からグラフの終わりまで直線が描画されます。この線はそれぞれ最初/最後の値のレベルにあります。

生の値への切り替え

右上のドロップダウンから、シンプルグラフを 値/最新 *500* 個の値の一覧に切り替えられます。これは、グラフを構成する数値を確認する場合に便利です。

ここに表示される値は生の値です。つまり、単位や値の前処理は適用されません。ただし、値のマッピングは適用されます。

既知の問題

グラフについては、known issuesを参照してください。

カスタムグラフ

概要

カスタムグラフは、その名の通りカスタマイズ機能を提供します。

シンプルグラフは単一のアイテムのデータを表示するのに適していますが、設定機能はありません。

したがって、グラフのスタイルや線の表示方法を変更したい場合や、複数のアイテム (例えば、1 つのグラフで受信トラフィックと送信トラフィックを比較したい場合) を比較したい場合は、カスタムグラフが必要です。

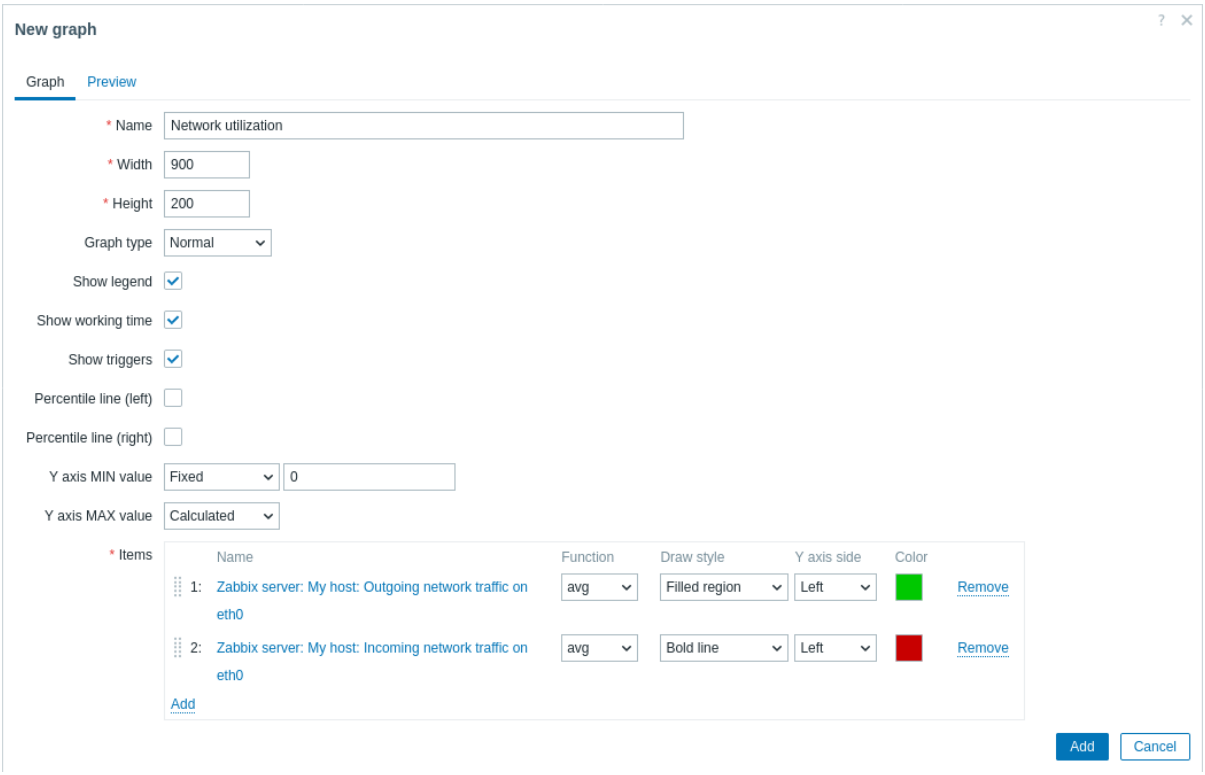
カスタムグラフは手動で設定します。

ホストまたは複数のホスト、または単一のテンプレートに対して作成できます。

カスタムグラフの設定

カスタムグラフを作成するには、次の手順を実行します。

1. データ収集 > ホスト（または テンプレート）に移動します。
2. 対象のホストまたはテンプレートの行にある *Graphs* をクリックします。
3. Graphs 画面で、*Create graph* をクリックします。
4. グラフの属性を編集します。



必須入力フィールドには赤いアスタリスクが付いています。

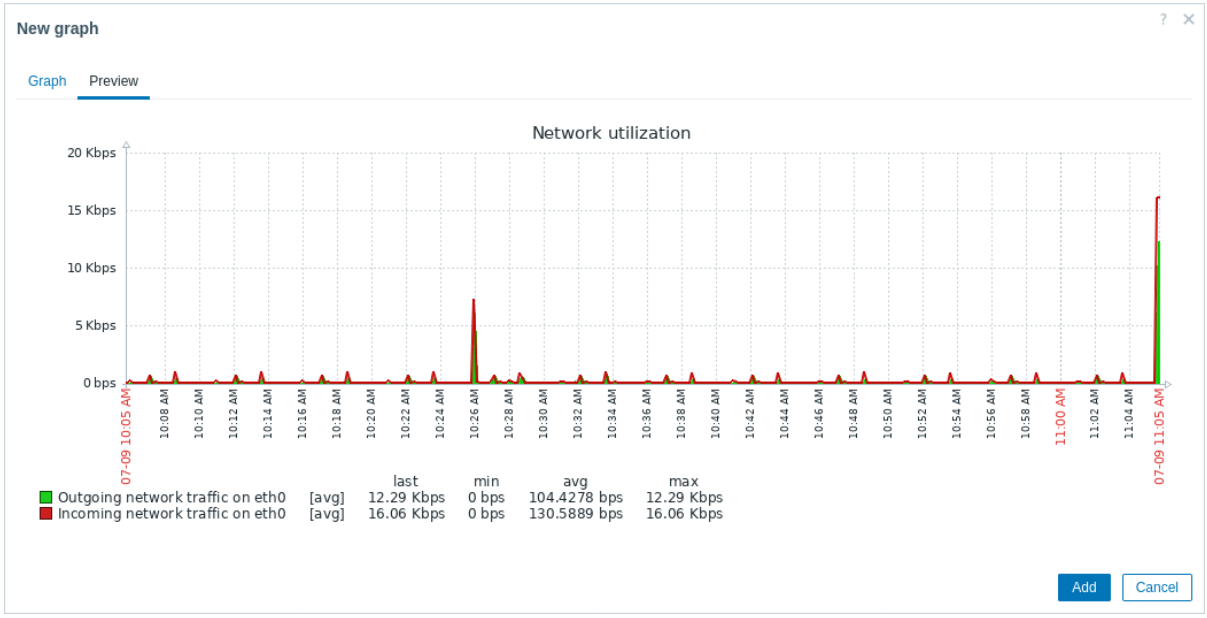
グラフの属性:

Parameter	Description
Name	一意のグラフ名。 このフィールドでは式macros がサポートされていますが、時間をパラメーターとする avg、last、min、max 関数のみ使用できます（例: {?avg(/host/key,1h)}）。 {HOST.HOST<1-9>} マクロは、このマクロ内で使用でき、グラフ内の 1 番目、2 番目、3 番目などのホストを参照します。例: {?avg(/{HOST.HOST2}/key,1h)}。なお、このマクロで 1 番目のホストを参照するのは冗長です。1 番目のホストは暗黙的に参照できるため、例: {?avg(//key,1h)}。
Width	グラフの幅（ピクセル単位、プレビューおよび円グラフ/分割円グラフのみ）。
Height	グラフの高さ（ピクセル単位）。
Graph type	グラフの種類: **Normal** - 通常のグラフ。値は線で表示されます **Stacked** - 積み上げグラフ。塗りつぶされた領域で表示されます **Pie** - 円グラフ **Exploded** - 「分割」円グラフ。各部分が円から「切り出された」ように表示されます

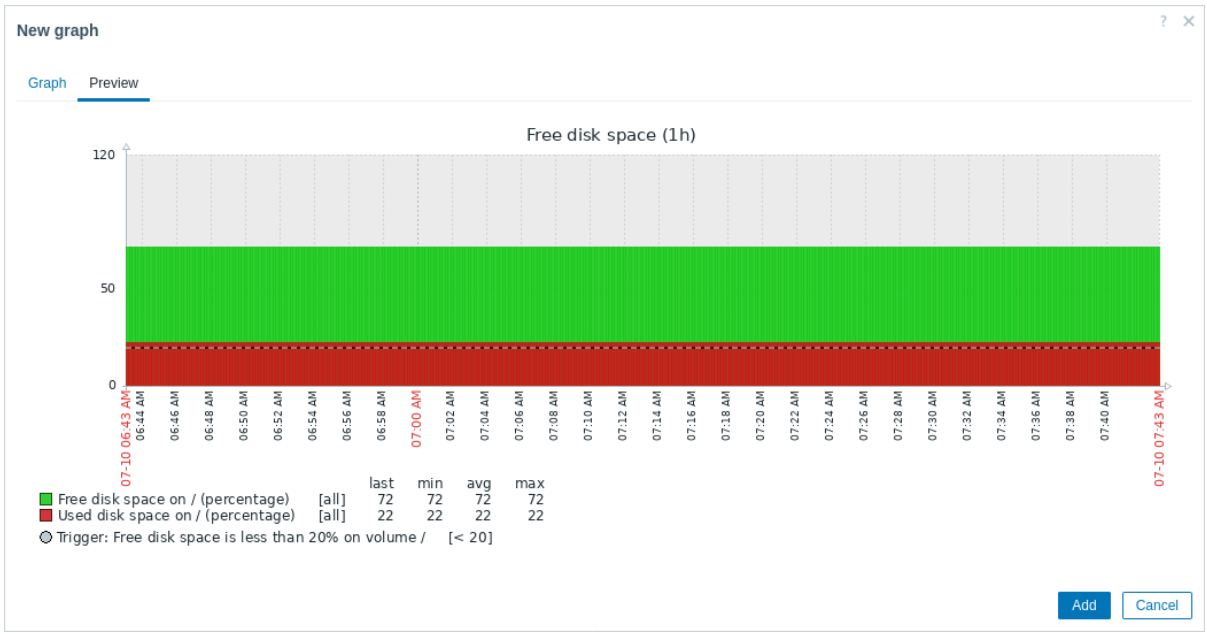
Parameter	Description
Show legend	このチェックボックスをオンにすると、グラフの凡例が表示されます。
Show working time	選択すると、非稼働時間が灰色の背景で表示されます。このパラメーターは円グラフおよび分割円グラフでは使用できません。
Show triggers	選択すると、simple triggers が、トリガーの深刻度の色の上に黒い破線付きの線として表示されます。このパラメーターは円グラフおよび分割円グラフでは使用できません。
Percentile line (left)	左 Y 軸のパーセンタイルを表示します。たとえば 95% パーセンタイルを設定すると、値の 95% がその下に収まる位置にパーセンタイル線が表示されます。明るい緑色の線で表示されます。通常のグラフでのみ使用できます。
Percentile line (right)	右 Y 軸のパーセンタイルを表示します。たとえば 95% パーセンタイルを設定すると、値の 95% がその下に収まる位置にパーセンタイル線が表示されます。明るい赤色の線で表示されます。通常のグラフでのみ使用できます。
Y axis MIN value	Y 軸の最小値: **Calculated** - Y 軸の最小値が自動計算されます。 **Fixed** - Y 軸の固定最小値。 **Item** - 選択したアイテムの最新値が最小値になります。 このパラメーターは円グラフおよび分割円グラフでは使用できません。
Y axis MAX value	Y 軸の最大値: **Calculated** - Y 軸の最大値が自動計算されます。 **Fixed** - Y 軸の固定最大値。 **Item** - 選択したアイテムの最新値が最大値になります。 このパラメーターは円グラフおよび分割円グラフでは使用できません。
3D view	3D スタイルを有効にします。円グラフおよび分割円グラフでのみ使用できます。
Items	このグラフに表示するデータのアイテムです。*Add* をクリックしてアイテムを選択します。表示オプション（関数、描画スタイル、左/右軸の表示、色）も選択できます。
Sort order (0→100)	描画順序。0 が最初に処理されます。他の線や領域の背後（または前面）に描画するために使用できます。 行の先頭にあるアイコンを使用してアイテムをドラッグ＆ドロップし、並び順や、どのアイテムを他のアイテムの前面に表示するかを設定できます。
Name	選択したアイテム名がリンクとして表示されます。リンクをクリックすると、利用可能な他のアイテムの一覧が開きます。
Type	種類（円グラフおよび分割円グラフでのみ使用可能）: **Simple** - アイテムの値が円グラフ上で比例表示されます **Graph sum** - アイテムの値が円全体を表します 「graph sum」アイテムの色は、「proportional」アイテムによって占められていない範囲にのみ表示される点に注意してください。
Function	アイテムに対して、縦方向のグラフ 1 ピクセルあたりに複数の値が存在する場合に表示する値を選択します: **all** - 可能なすべての値（最小、最大、平均）をグラフに表示します。短い期間ではこの設定は効果がなく、縦方向のグラフ 1 ピクセルにデータが集中する長い期間でのみ、'all' は最小、最大、平均値の表示を開始します。この関数は *Normal* グラフタイプでのみ使用できます。こちらも参照: 履歴/トレンドからのグラフ生成from history/trends。 **avg** - 平均値を表示します **last** - 最新値を表示します。この関数は、グラフタイプとして *Pie/Exploded pie* が選択されている場合にのみ使用できます。 **max** - 最大値を表示します **min** - 最小値を表示します
Draw style	描画スタイルを選択します（通常のグラフでのみ使用可能。積み上げグラフでは塗りつぶし領域が常に使用されます）。アイテムデータに適用するスタイルは *Line*、*Bold line*、*Filled region*、*Dot*、*Dashed line*、*Gradient line* です。
Y axis side	アイテムデータを表示する Y 軸の側を選択します - *Left*、*Right*。
Color	アイテムデータに適用する色を選択します。

グラフのプレビュー

プレビュータブでは、グラフのプレビューが表示され、作成している内容をすぐに確認できます。



テンプレートアイテムの場合、プレビューにはデータが表示されないことに注意してください。



この例では、トリガーレベルを表示する破線の太線と、凡例に表示されるトリガー情報に注目してください。

Note:
表示できるトリガーラインは 3 本までです。トリガーが 3 つ以上ある場合は、重大度の低いトリガーが優先して表示されます。

 グラフの高さが 120 ピクセル未満に設定されている場合、凡例にはトリガーが表示されません。

アドホックグラフ

概要

シンプルグラフは単一のアイテムのデータを表示するのに便利で、カスタムグラフではカスタマイズオプションを利用できますが、どちらも複数のアイテムを比較するグラフを、最小限の手間とメンテナンスで迅速に作成することはできません。

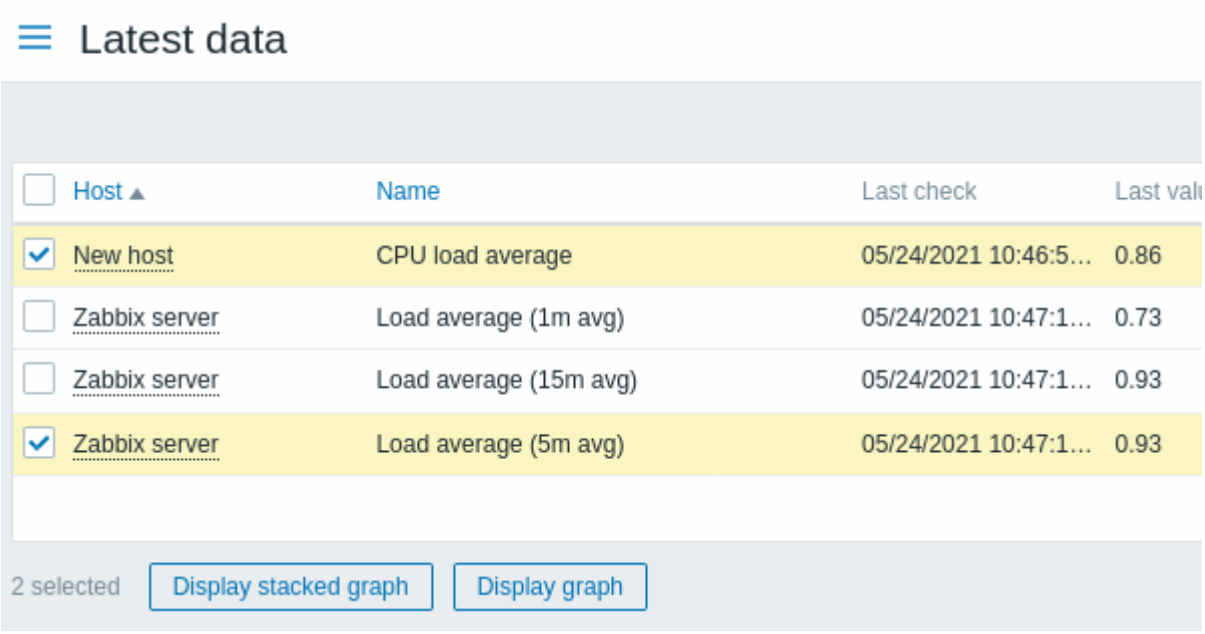
この問題に対処するため、複数のアイテムのアドホックグラフを非常に迅速に作成できます。

設定

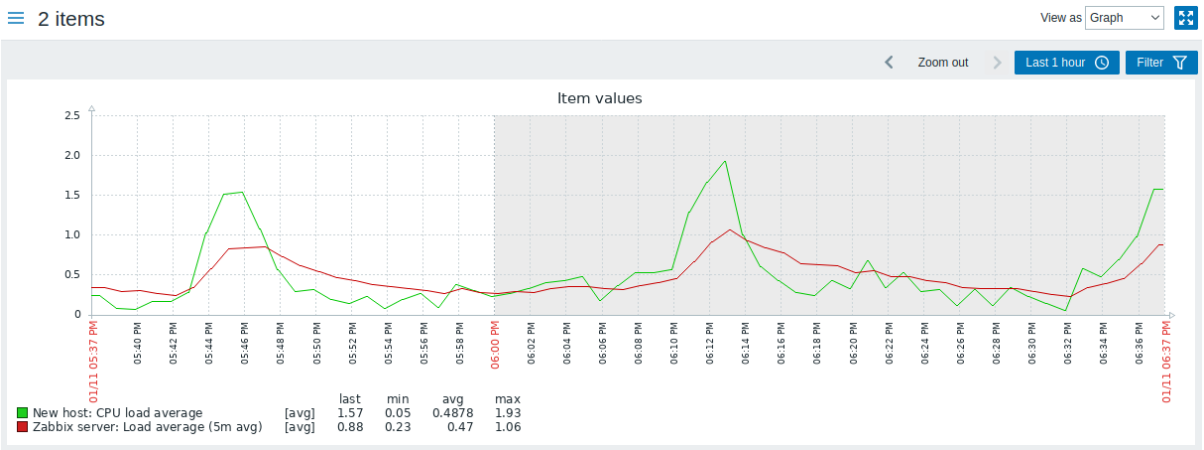
アドホックグラフを作成するには、次の手順を実行します。

1. *Monitoring* > *Latest data* に移動します。
2. フィルターを使用して、表示したいアイテムを表示します。

3. グラフ化したいアイテムのチェックボックスをオンにします。
4. *Display stacked graph* または *Display graph* ボタンをクリックします。

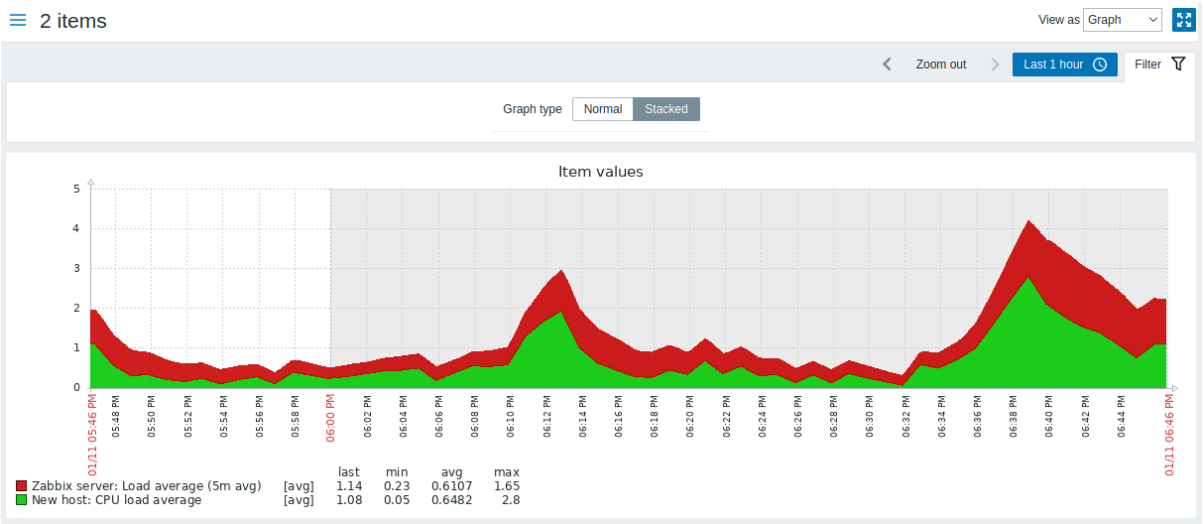


グラフは即座に作成されます。



グラフに表示される線が多くなりすぎないように、各アイテムについては平均値のみが表示されます（最小/最大値の線は表示されません）。トリガーおよびトリガー情報はグラフに表示されません。

作成されたグラフのウィンドウでは、時間期間セレクターを利用でき、” 通常” の折れ線グラフから積み上げグラフへ、またその逆へ切り替えることができます。



ネットワークマップ

概要

ネットワークの管理をしている場合、インフラの概要をどこかで把握したいと思うかもしれません。
 そのために、Zabbix ではネットワークや好きなもののマップを作成することができます。

すべてのユーザがネットワークマップを作成することができます。
 マップはパブリック (すべてのユーザが利用可能) またはプライベート (選択されたユーザが利用可能) にすることができます。

configuring a network mapに進みます。

ネットワークマップの設定

概要

Zabbix でマップを設定するには、まず一般パラメータを定義してマップを作成し、その後、実際のマップに要素とそのリンクを追加していきます。

マップには、ホスト、ホストグループ、トリガー、画像、または別のマップを要素として配置できます。

アイコンはマップ要素を表すために使用されます。アイコンに表示する情報を定義したり、最近の問題を特別な方法で表示するように設定したりできます。また、アイコン同士をリンクし、リンク上に表示する情報を定義することもできます。

アイコンをクリックしてアクセスできるカスタム URL を追加できます。これにより、ホストのアイコンをホストのプロパティに、またはマップのアイコンを別のマップにリンクできます。

マップに表示される問題数は、原因となっている問題のみを対象としています。

マップは *Monitoring* > *Maps* で管理され、そこで設定、管理、表示できます。監視ビューでは、アイコンをクリックして、いくつかのスクリプトや URL へのリンクを利用できます。

ネットワークマップはベクターグラフィックス（SVG）に基づいています。

パブリックマップとプライベートマップ

Zabbix のすべてのユーザー（管理者以外のユーザーも含む）は、ネットワークマップを作成できます。マップには所有者（作成したユーザー）がいます。マップはパブリックまたはプライベートにすることができます。

- パブリックマップはすべてのユーザーに表示されますが、表示するにはユーザーが少なくとも 1 つのマップ要素への読み取り権限を持っている必要があります。パブリックマップは、ユーザー/ユーザーグループがこのマップに対して読み書き権限を持ち、かつ対応するマップのすべての要素（リンク内のトリガーを含む）に対して少なくとも読み取り権限を持っている場合に編集できます。
- プライベートマップは、所有者と、所有者が共有したユーザー/ユーザーグループにのみ表示されます。通常の（スーパ管理者以外の）ユーザーは、自分が所属するグループおよびそのグループに所属するユーザーとしか共有できません。管理者レベルのユーザーは、所有者であるか共有ユーザーリストに含まれているかに関係なく、プライベートマップを表示できます。プライベートマップは、マップの所有者および、ユーザー/ユーザーグループがこのマップに対して読み書き権限を持ち、かつ対応するマップのすべての要素（リンク内のトリガーを含む）に対して少なくとも読み取り権限を持っている場合に編集できます。

ユーザーが読み取り権限を持たないマップ要素は、グレーアウトしたアイコンで表示され、要素に関するすべてのテキスト情報は非表示になります。ただし、ユーザーがトリガーへの権限を持っていなくても、トリガーラベルは表示されます。

マップに要素を追加するには、ユーザーがその要素に対して少なくとも読み取り権限を持っている必要があります。

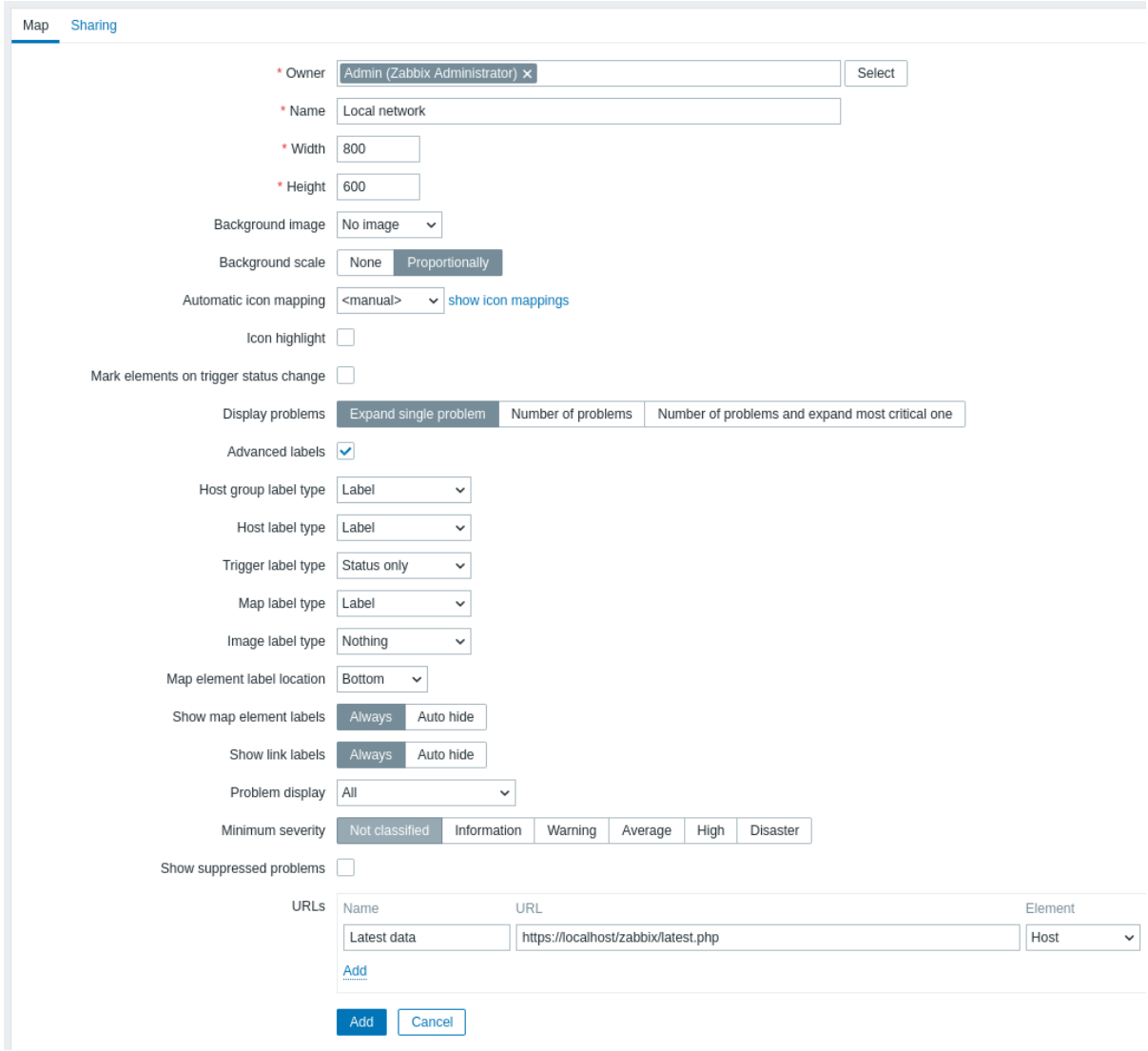
マップの作成

マップを作成するには、次の手順を実行します。

1. 監視 > マップに移動します。
2. すべてのマップが表示されるビューに移動します。
3. マップの作成をクリックします。
4. マップの詳細を設定します。

既存のマップの設定フォームにある 複製ボタンを使用して、新しいマップを作成することもできます。このマップには、一般的なレイアウト属性を含む既存のマップのすべてのプロパティと、既存のマップの要素が引き継がれます。

マップタブには、マップの一般的な属性が含まれています。



必須入力フィールドには、赤いアスタリスクが付いています。

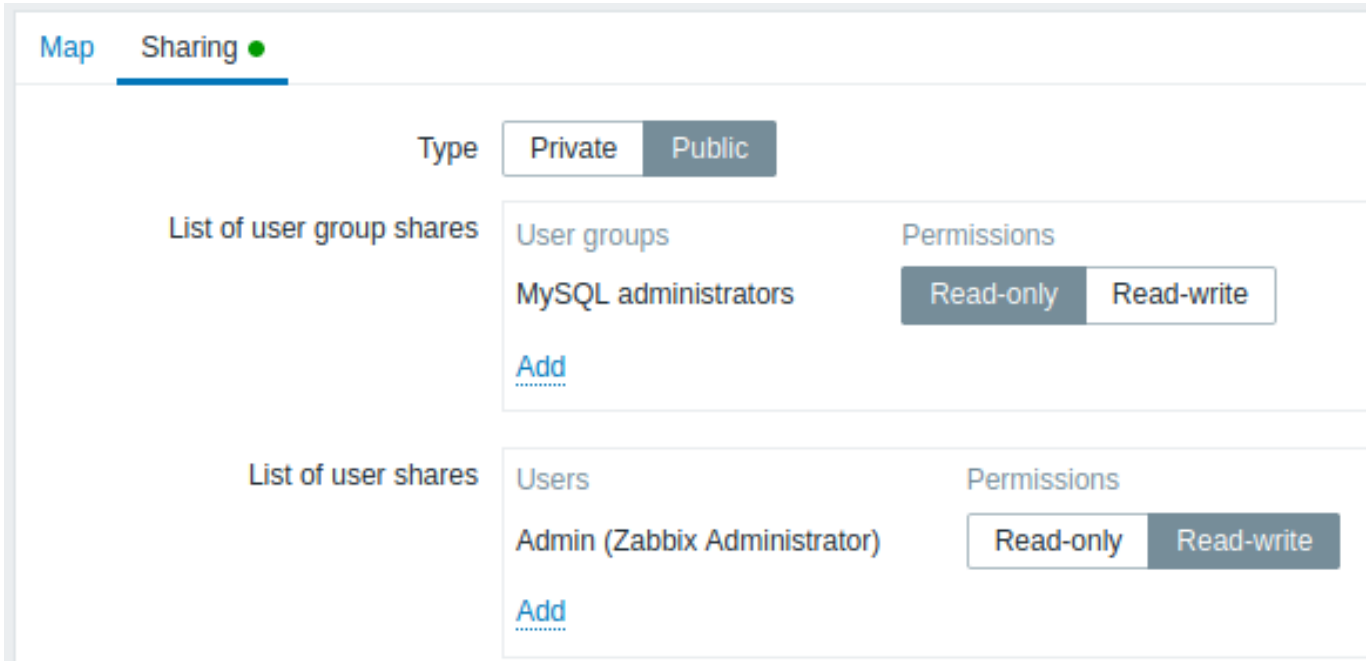
マップの一般的な属性:

パラメーター	説明
所有者	マップ所有者の名前。
名前	一意のマップ名。
幅	ピクセル単位のマップ幅。
高さ	ピクセル単位のマップの高さ。
背景画像	背景画像を使用します: 画像なし - 背景画像なし（白い背景） 画像 - 背景画像として使用する選択した画像。地理的なマップやその他の画像を使用して、マップを見やすくすることができます。
背景の拡大縮小	背景の拡大縮小を使用します: なし - 拡大縮小なし; 比例 - 画像の比率を維持したまま、マップの背景全体を覆います（デフォルト）。
自動アイコンマッピング	管理 > 一般設定 > アイコンマッピングで設定した自動アイコンマッピングを使用するように設定できます。アイコンマッピングを使用すると、特定のアイコンを特定のホストインベントリフィールドに対応付けることができます。
アイコンの強調表示	このチェックボックスをオンにすると、マップ要素が強調表示されます。 アクティブなトリガーがある要素には、最も高い深刻度のトリガーと同じ色の円形の背景が表示されます。また、すべての障害が認識済みの場合は、円の周囲に太い緑色の線が表示されます。 「無効」または「メンテナンス中」のステータスの要素には、それぞれ灰色またはオレンジ色の四角形の背景が表示されます。 関連項目: マップの表示
トリガーのステータス変更時に要素をマーク	トリガー状態の最近の変更（最近発生した障害または障害の解決）は、要素アイコンのラベルがない 3 辺に、内側を指す赤い三角形のマーカーとして強調表示されます。マーカーは 30 分間表示されます。

パラメーター	説明
障害の表示	マップ要素での障害の表示方法を選択します: 単一の障害を展開 - 障害が 1 件のみの場合は、障害名が表示されます。それ以外の場合は、障害の合計数が表示されます。 障害数 - 障害の合計数が表示されます 障害数と最も深刻な障害を展開 - 最も深刻な障害の名前と障害の合計数が表示されます。 「最も深刻」は、障害の深刻度に基づいて決定されます。深刻度が同じ場合は、障害イベント ID（ID が大きいもの、または後に発生した障害が先）に基づきます。トリガーのマップ要素では、障害の深刻度に基づき、深刻度が同じ場合はトリガーリスト内のトリガーの位置に基づきます。同じトリガーで複数の障害がある場合は、最も新しい障害が表示されます。
高度なラベル	このチェックボックスをオンにすると、要素タイプごとに個別のラベルタイプを定義できます。
ホストグループのラベルタイプ	マップ要素に使用するラベルタイプ: ラベル - マップ要素のラベル **IP** アドレス - IP アドレス 要素名 - 要素名（例: ホスト名） ステータスのみ - ステータスのみ（OK または障害） なし - ラベルを表示しません
ホストのラベルタイプ	
トリガーのラベルタイプ	
マップのラベルタイプ	
画像のラベルタイプ	
マップ要素のラベル位置	マップ要素に対するラベルの位置: 下 - マップ要素の下 左 - 左側 右 - 右側 上 - マップ要素の上
マップ要素のラベルを表示	マップ要素のラベルの表示方法を選択します: 常に表示 - 常に表示します（デフォルト）; 自動的に非表示 - ラベルにカーソルを合わせていない、または選択していない場合はラベルを非表示にします。
リンクラベルを表示	リンクラベルの表示方法を選択します: 常に表示 - 常に表示します（デフォルト）; 自動的に非表示 - ラベルにカーソルを合わせていない、または選択していない場合はラベルを非表示にします。
障害の表示	障害数の表示方法: すべて - 障害の合計数を表示します 分離 - 未認識の障害数を、障害の合計数とは分けて表示します 未認識のみ - 未認識の障害数のみを表示します
最小の深刻度	選択した最小深刻度レベル未満の障害は、マップに表示されません。 たとえば、警告を選択した場合、情報および 未分類レベルのトリガーによる変更はマップに反映されません。
抑制された障害を表示	ホストのメンテナンスによって本来抑制される（表示されない）障害を表示するには、チェックボックスをオンにします。
URL	ここでは、各要素タイプの URL を定義できます。 名前 - URL のラベル; **URL** - URL（最大 2048 文字）; 要素 - 要素タイプ（例: ホスト、ホストグループなど）。 これらは、マップ表示モードでユーザーが要素をクリックしたときにリンクとして表示されます。 マップ URL の名前と値にはマクロを使用できます。完全な一覧については、サポートされているマクロを参照し、「map URL names and values」を検索してください。

共有

共有タブには、マップの種類と、プライベートマップ向けの共有オプション（ユーザーグループ、ユーザー）が含まれます。



Parameter	Description
Type	マップの種類を選択します: **Private** - マップは選択したユーザーグループとユーザーにのみ表示されます **Public** - マップはすべてのユーザーに表示されます
List of user group shares	マップへのアクセスを許可するユーザーグループを選択します。 読み取り専用または読み取り/書き込みアクセスを許可できます。
List of user shares	マップへのアクセスを許可するユーザーを選択します。 読み取り専用または読み取り/書き込みアクセスを許可できます。

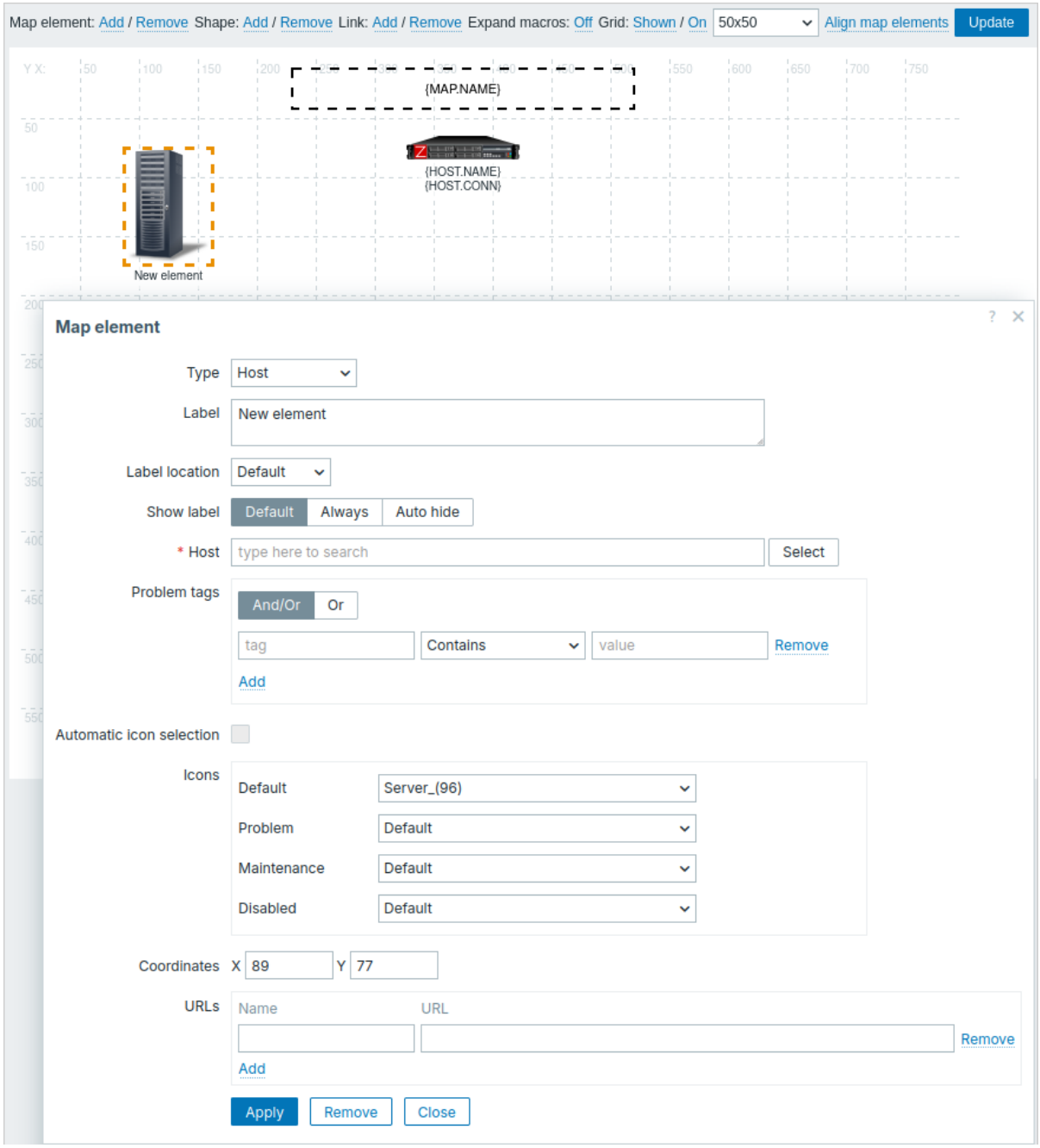
このマップを保存するために *Add* をクリックすると、名前、サイズ、およびいくつかの設定を持つ空のマップが作成されます。次に、要素を追加する必要があります。そのためには、マップ一覧で *Edit* をクリックして編集領域を開きます。

要素の追加

要素を追加するには、`Map element` の横にある *Add* をクリックします。新しい要素はマップの左上隅に表示されます。ドラッグ＆ドロップで好きな場所に配置してください。

Grid オプションが”On” の場合、要素は常にグリッドに揃えられます（ドロップダウンからさまざまなグリッドサイズを選択でき、グリッドの表示/非表示も切り替えられます）。配置を揃えずに任意の場所へ置きたい場合は、このオプションを”Off” にします。（後から *Align map elements* をクリックして、ランダムに配置した要素をグリッドに揃えることもできます。）

要素を配置できたら、名前を付けるなどして区別したくなるでしょう。要素をクリックするとフォームが表示され、要素タイプの設定、名前の指定、別のアイコンの選択などができます。



マップ要素の属性:

Parameter	Description
Type	要素のタイプ: **Host** - 選択したホストのすべてのトリガーの状態を表すアイコン **Map** - マップ内のすべての要素の状態を表すアイコン **Trigger** - 1 つ以上のトリガーの状態を表すアイコン **Host group** - 選択したグループに属するすべてのホストのすべてのトリガーの状態を表すアイコン **Image** - いずれのリソースにもリンクされていないアイコン
Label	アイコンのラベル。任意の文字列を指定できます。 マクロおよび複数行文字列を使用できます。 このフィールドでは式のmacros がサポートされていますが、`avg`、`last`、`min`、`max` 関数のみ、かつ時間をパラメータとして指定する場合に限られます（例: `{?avg(/host/key,1h)}`）。 サポートされるマクロの一覧は、supported macros を参照し、'map element labels' を検索してください。

Parameter	Description
Label location	アイコンに対するラベルの位置: **Default** - マップの既定のラベル位置; **Bottom** - アイコンの下; **Left** - 左側; **Right** - 右側; **Top** - アイコンの上。
Show label	要素ラベルの表示方法を選択します: **Default** - マップ設定の *Show map element labels* を使用します（既定）; **Always** - 常に表示します（既定）; **Auto hide** - ホバーまたは選択されていない場合はラベルを非表示にします。
Host	ホストを選択するか、ホスト名を入力します。このフィールドは自動補完に対応しているため、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。スクロールして選択してください。選択を解除するには削除（x）アイコンをクリックします。 このフィールドは *Host* タイプの要素でのみ使用できます。
Map	マップを選択するか、マップ名を入力します。このフィールドは自動補完に対応しているため、マップ名の入力を始めると一致するマップのドロップダウンが表示されます。スクロールして選択してください。選択を解除するには削除（x）アイコンをクリックします。 このフィールドは *Map* タイプの要素でのみ使用できます。
Triggers	下の *New triggers* フィールドで 1 つ以上のトリガーを選択し、*Add* をクリックします。あるいは、トリガー名の入力を開始して、一致するトリガーのドロップダウン一覧から 1 つ選択します。選択を解除するには削除（x）アイコンをクリックします。 このフィールドは *Trigger* タイプの要素でのみ使用できます。 選択したトリガーの順序は変更できますが、同じ深刻度のトリガー内に限られます。複数トリガーの選択は、編集モードと表示モードの両方で {HOST.*} マクロの解決にも影響します。 編集モードでは、最初に表示される {HOST.*} マクロは、一覧の最初のトリガー（トリガーの深刻度に基づく）に応じて解決されます。 表示モードは、General map attributes のDisplay problems パラメータに依存します: - *Expand single problem* モードが選択されている場合、最初に表示される {HOST.*} マクロは、最新に検出された問題トリガー（深刻度は関係ありません）、または問題が検出されていない場合は一覧の最初のトリガーに応じて解決されます。 - *Number of problems and expand most critical one* モードが選択されている場合、最初に表示される {HOST.*} マクロはトリガーの深刻度に応じて解決されます。
Host group	ホストグループを選択するか、ホストグループ名を入力します。このフィールドは自動補完に対応しているため、ホストグループ名の入力を始めると一致するホストグループのドロップダウンが表示されます。スクロールして選択してください。選択を解除するには削除（x）アイコンをクリックします。 このフィールドは *Host group* タイプの要素でのみ使用できます。
Problem tags	ウィジェットに表示する問題数を制限するためのタグを指定します。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件には次の演算子があります: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字小文字を区別） **Contains** - タグ値に入力文字列を含む指定タグ名を含める（部分一致、大文字小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字小文字を区別） **Does not contain** - タグ値に入力文字列を含む指定タグ名を除外する（部分一致、大文字小文字を区別しない） 条件の計算タイプは 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です このフィールドは *Host* および *Host group* タイプの要素でのみ使用できます。
Automatic icon selection	チェックボックスをオンにすると、どのアイコンを表示するかの判定にアイコンマッピングが使用されます。
Icons	次の場合に要素へ別のアイコンを表示するよう選択できます: *Default*、*Problem*、*Maintenance*、*Disabled*。 このフィールドは *Host* タイプの要素でのみ使用できます。
Coordinates	マップ要素の **X** および **Y** 座標。

Parameter	Description
URLs	ここで要素固有の URL を設定できます。 **Name** - URL のラベル; **URL** - URL（最大 2048 文字）。 これらは、ユーザーがマップ表示モードで要素をクリックしたときにリンクとして表示されます。 要素に独自の URL があり、かつそのタイプに対するマップレベルの URL が定義されている場合、それらは同じメニューにまとめて表示されます。 マップ要素名と値にはマクロを使用できます。完全な一覧はsupported macros を参照し、'map URL names and values' を検索してください。

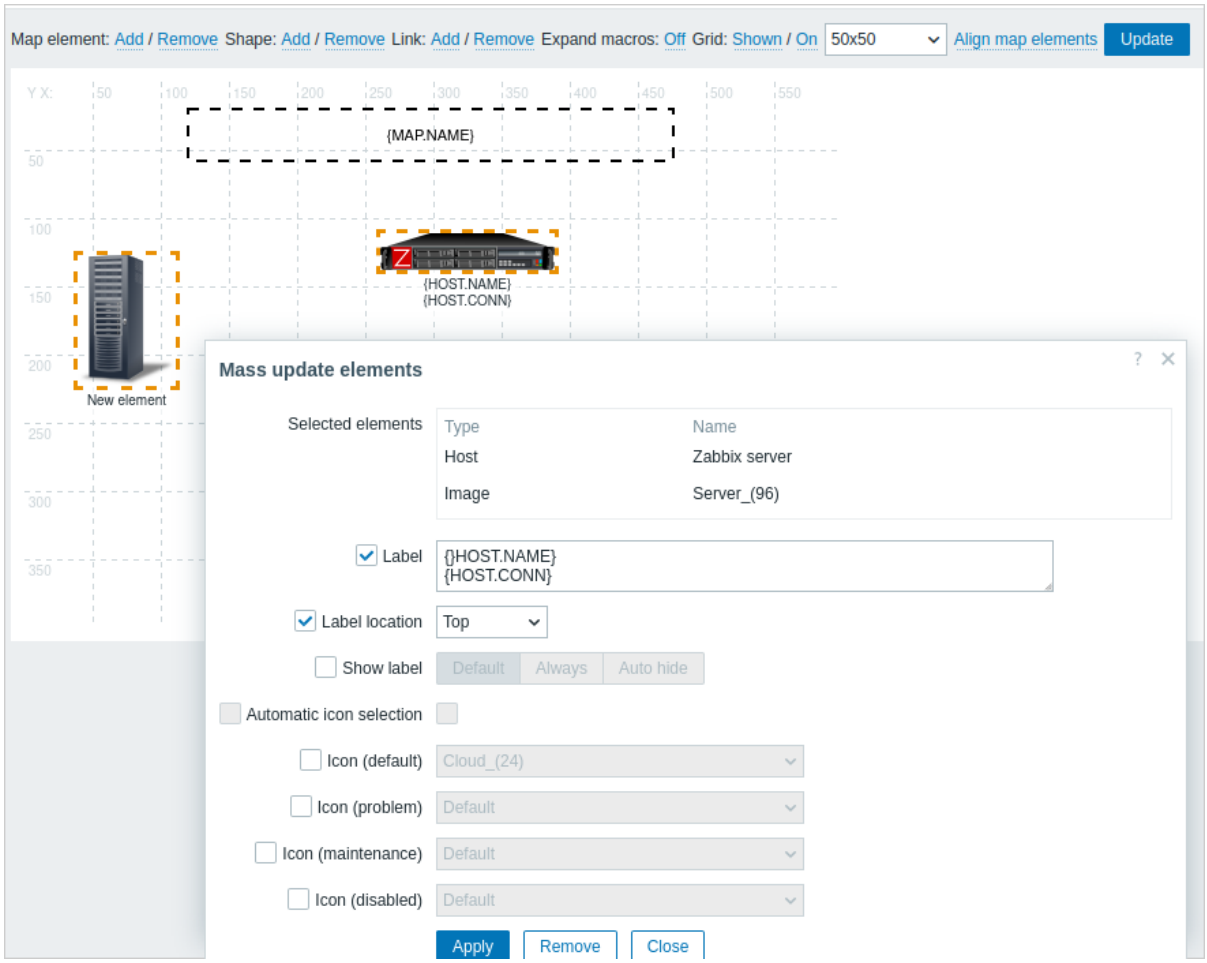
Attention:
追加した要素は自動的には保存されません。ページから移動すると、すべての変更が失われる可能性があります。
そのため、右上の **Update** ボタンをクリックすることをお勧めします。クリックすると、その後のポップアップで何を選択しても変更は保存されます。
選択したグリッドオプションも各マップとともに保存されます。

要素の選択

要素を選択するには、1 つを選択してから *Ctrl* を押しながら他の要素を選択します。

また、編集可能な領域で矩形をドラッグし、その中のすべての要素を選択することで、複数の要素を選択することができます。

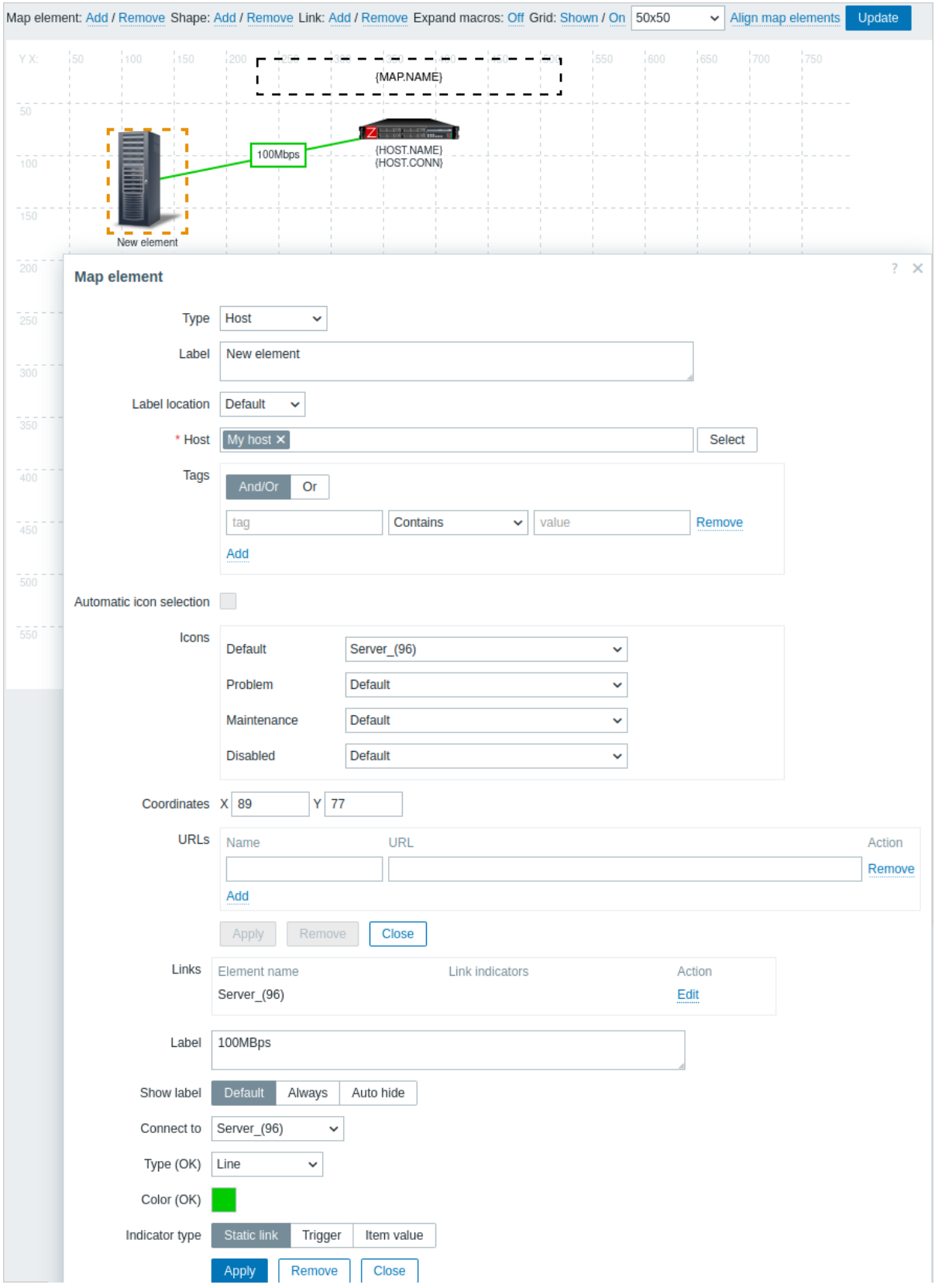
複数の要素を選択すると、要素のプロパティフォームが一括更新モードに移行し、選択した要素の属性を一括して変更する
 ことができます。そのためには、チェックボックスを使って属性をマークし、その属性に新しい値を入力します。
 ここでマクロを使用することもできます (例えば、要素のラベルに {HOST.NAME} を使用)。



要素のリンク

マップ上にいくつかの要素を配置したら、次はそれらをリンクする作業です。2 つの要素をリンクするには、まずそれらを選択する必要があります。要素を選択した状態で、Link の横にある *Add* をクリックします。

リンクを作成すると、単一要素のフォームに新しい *Links* セクションが追加されます。*Edit* をクリックしてリンク属性を編集します。



リンク属性:

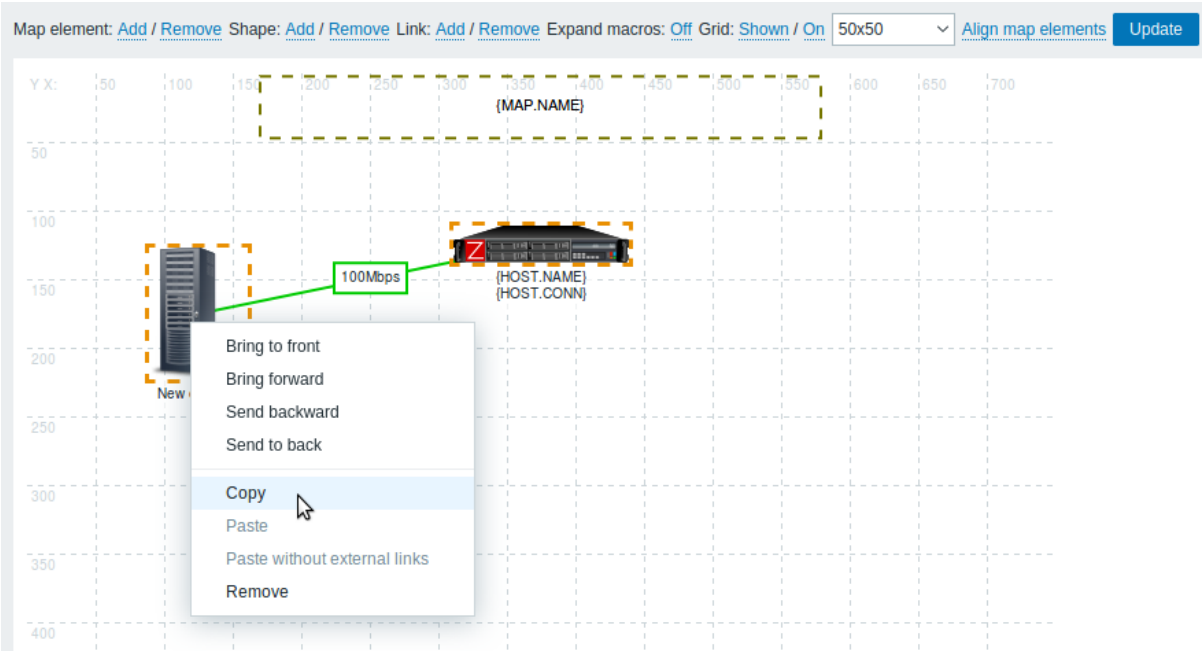
Parameter	Description
Label	リンクの上に表示されるラベルです。 このフィールドでは式macros がサポートされていますが、時間をパラメーターとする `avg`、`last`、`min`、`max` 関数のみが使用できます（例: `{?avg(/host/key,1h)}`）。
Show label	リンクラベルの表示方法を選択します: **Default** - マップ設定の *Show map element labels* 設定を使用します（デフォルト）; **Always** - 常に表示します（デフォルト）; **Auto hide** - ホバーされていない、または選択されていない場合はラベルを非表示にします。

Parameter	Description
Connect to	リンクが接続する要素です。
Type (OK)	デフォルトのリンクスタイルです。スタイルを *Line*、*Bold line*、*Dot*、または *Dashed line* から選択します。
Color (OK)	デフォルトのリンク色です。別の色を選択するには、カラースウォッチをクリックします。
Indicator type	リンクインジケーターの種類を選択します: **Static link** - インジケーターなし; **Trigger** - トリガーベースのリンクインジケーターを使用可能; **Item value** - アイテム値ベースのリンクインジケーターを使用可能。
Item	アイテムを選択します。このアイテムの値がリンクのスタイルに影響します。 このフィールドは、*Indicator type* で *Item value* を選択した場合に使用できます。
Indicators	リンクインジケーター の一覧です。 このフィールドは、*Indicator type* が *Trigger* または *Item value* に設定されている場合に使用できます。 *Trigger* に設定すると、トリガーをリンクインジケーターとして割り当てることができます。トリガーが *Problem* 状態になると、そのスタイル（選択した色と線種）がリンクに適用されます。 *Item value* に設定され、*Item* フィールドでアイテムが選択されている場合は、アイテム値をリンクインジケーターとして割り当てることができます。アイテム値が指定したしきい値に達するか（数値アイテムの場合）、パターンに一致すると（テキストアイテムの場合）、そのスタイル（選択した色と線種）がリンクに適用されます。

要素の移動とコピー＆ペースト

複数の選択した要素は、選択した要素の 1 つをクリックし、マウスボタンを押したまま、カーソルを目的の場所へ移動することで、別の場所へ移動できます。

1 つ以上の要素は、要素を選択したあと、選択した要素をマウスの右ボタンでクリックし、メニューから *Copy* を選択することでコピーできます。

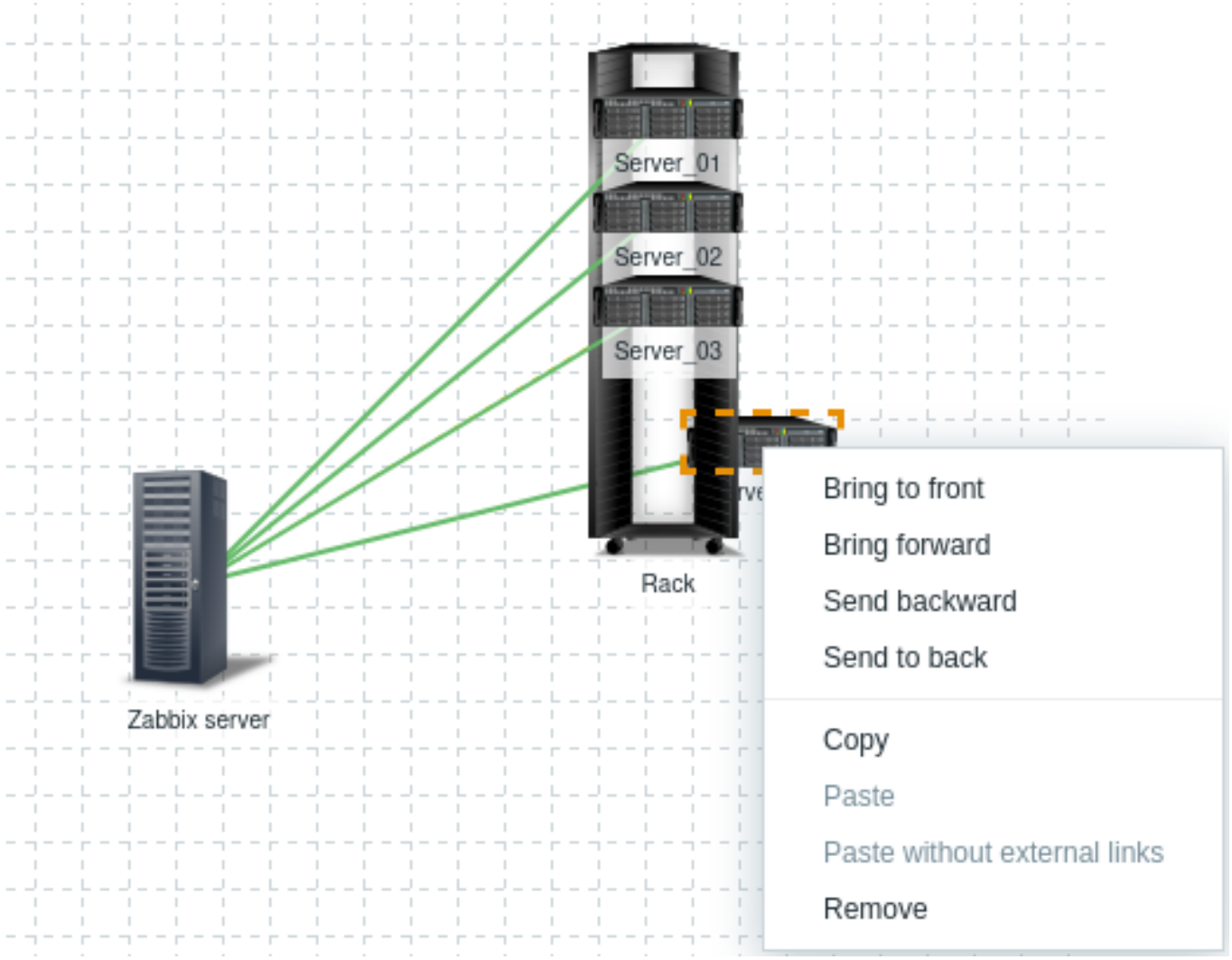


要素を貼り付けるには、マップ領域をマウスの右ボタンでクリックし、メニューから *Paste* を選択します。*Paste without external links* オプションを使用すると、選択した要素間のリンクのみを保持したまま要素を貼り付けます。

コピー＆ペーストは、同じブラウザーウィンドウ内でのみ動作します。キーボードショートカットはサポートされていません。

要素の順序変更

ある要素を別の要素の前面に移動するには（またはその逆にするには）、その要素を右クリックし、前面へ移動/最前面へ移動または背面へ移動/最背面へ移動を選択します。

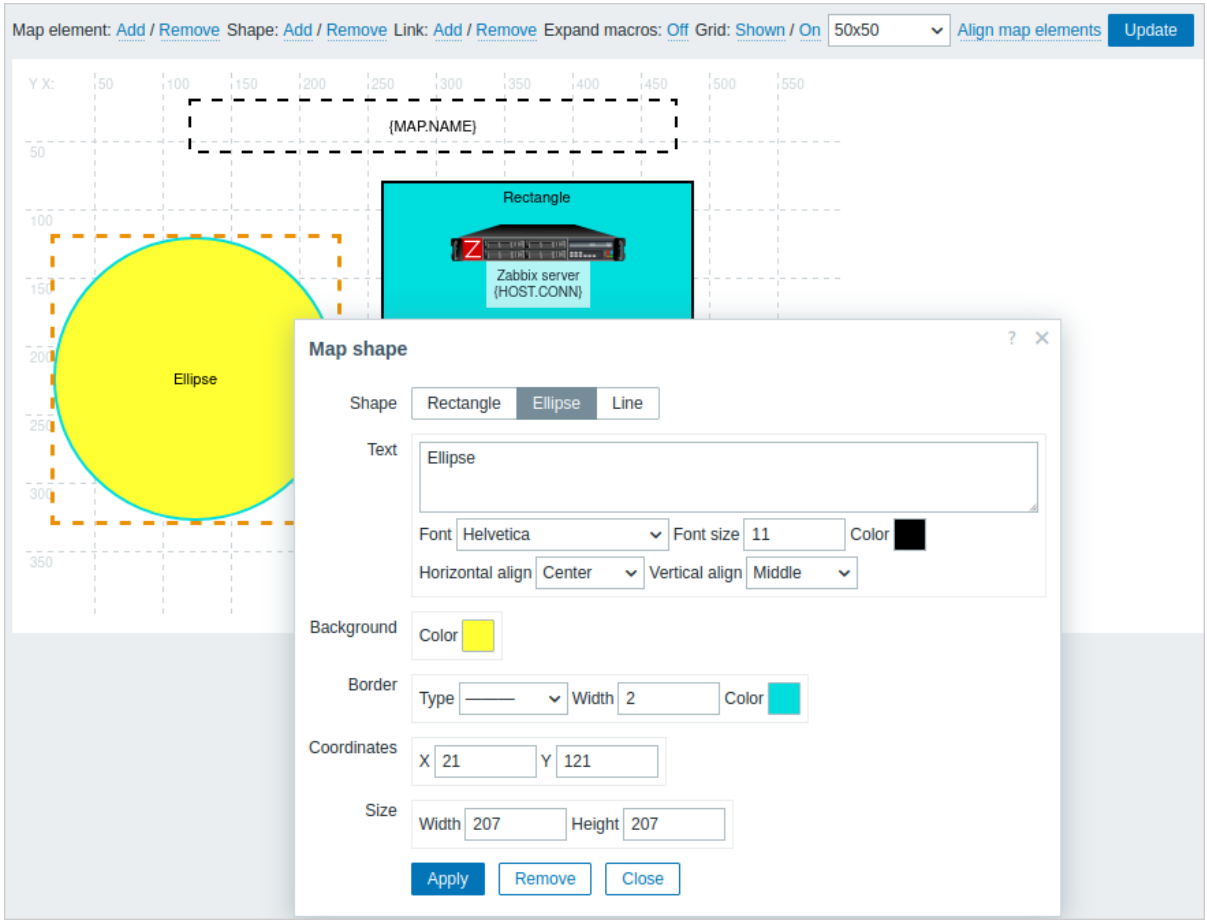


図形の追加

マップ要素に加えて、いくつかの図形を追加することもできます。図形はマップ要素ではなく、単なる視覚的な表現です。たとえば、長方形の図形は、いくつかのホストをグループ化する背景として使用できます。長方形と楕円の図形を追加できます。

図形を追加するには、Shape の横にある *Add* をクリックします。新しい図形はマップの左上隅に表示されます。好きな場所へドラッグ＆ドロップしてください。

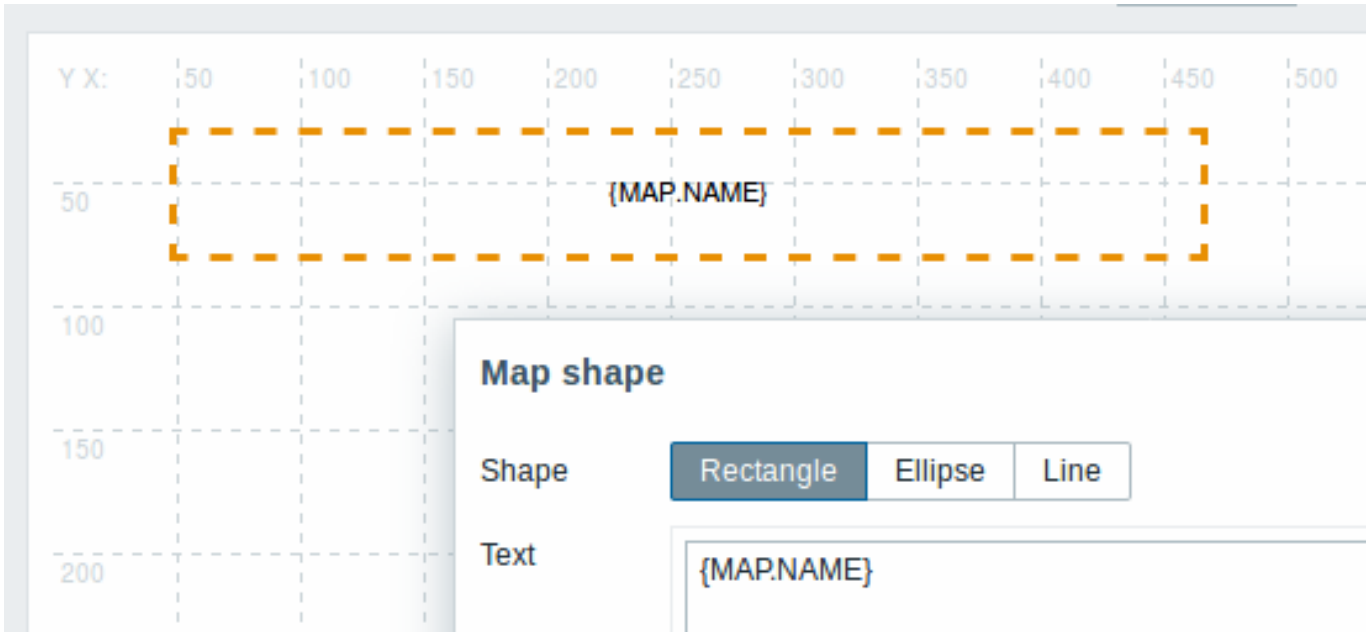
新しい図形は既定の色で追加されます。図形をクリックするとフォームが表示され、図形の見た目やテキストなどをカスタマイズできます。



図形を選択するには、1 つを選択してから *Ctrl* を押しながら他の図形を選択します。複数の図形を選択すると、要素と同様に共通プロパティを一括更新できます。

図形内にはテキストを追加できます。テキスト内では式マクロがサポートされていますが、`avg`、`last`、`min`、`max` 関数のみで、パラメータには時間を指定します（例: `{?avg(/host/key,1h)}`）。

テキストのみを表示したい場合は、図形の枠線を削除して図形を非表示にできます（*Border* フィールドで'None' を選択します）。たとえば、上のスクリーンショットに表示されている {MAP.NAME} マクロは、実際にはテキスト付きの長方形図形であり、マクロをクリックすると確認できます。



{MAP.NAME} は、マップを表示したときに設定済みのマップ名に展開されます。

テキスト内でハイパーリンクを使用している場合、マップ表示時にクリック可能になります。

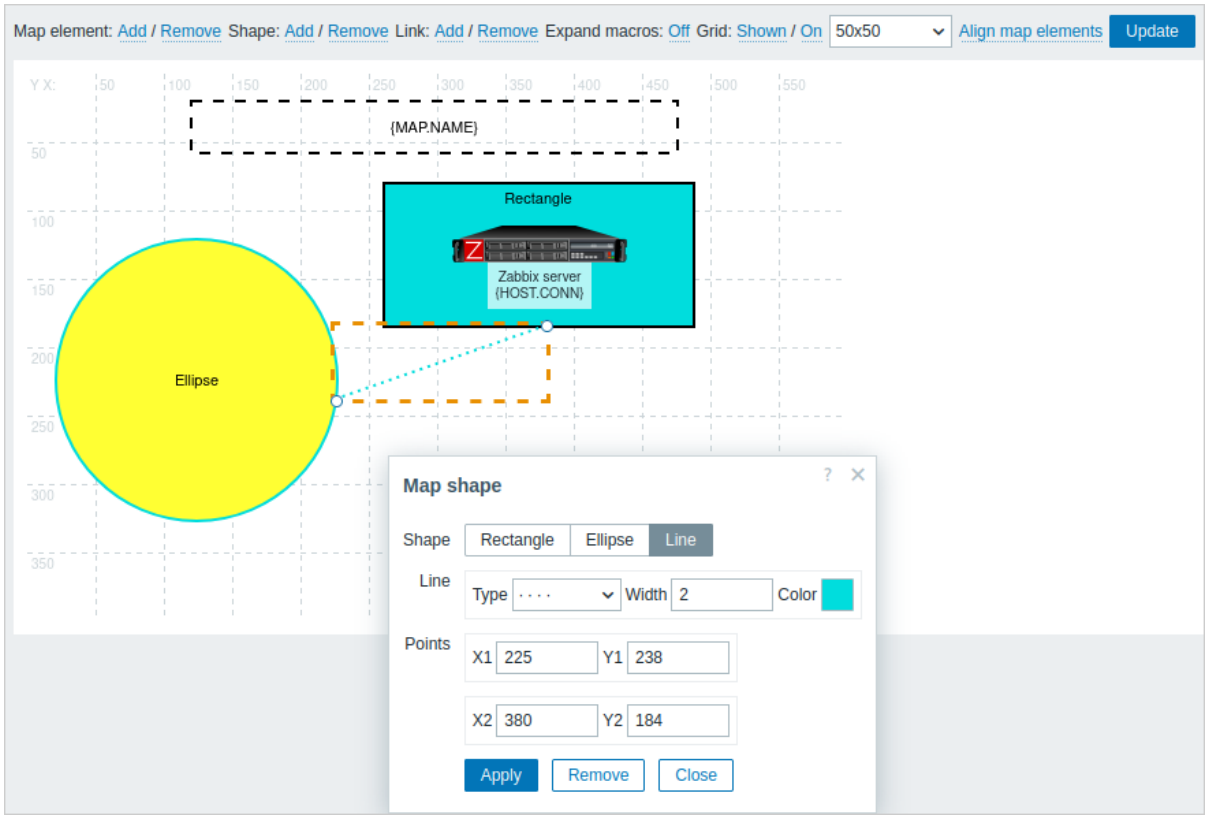
図形内のテキストの改行は常に有効です。ただし、楕円内では、行は楕円を長方形とみなした場合のように折り返されます。単語単位の折り返しは実装されていないため、長い単語（図形に収まらない単語）は折り返されず、マスクされます（マップ編集ページ）また

は切り詰められます（マップを表示する他のページ）。

線の追加

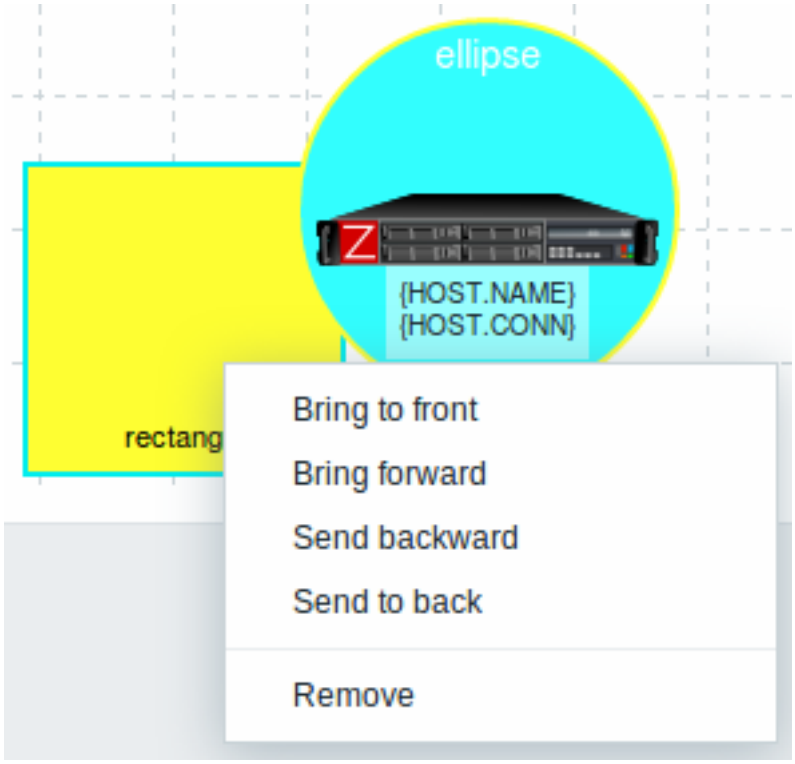
図形に加えて、いくつかの線を追加することもできます。線は、マップ内の要素や図形を結ぶために使用できます。

線を追加するには、Shape の横にある *Add* をクリックします。新しい図形がマップの左上隅に表示されます。それを選択し、編集フォームで *Line* をクリックして図形を線に変更します。その後、線種、幅、色などの線のプロパティを調整します。



図形と線の順序を変更する

ある図形を別の図形の前面に移動するには（またはその逆にするには）、図形を右クリックしてマップ図形メニューを表示します。

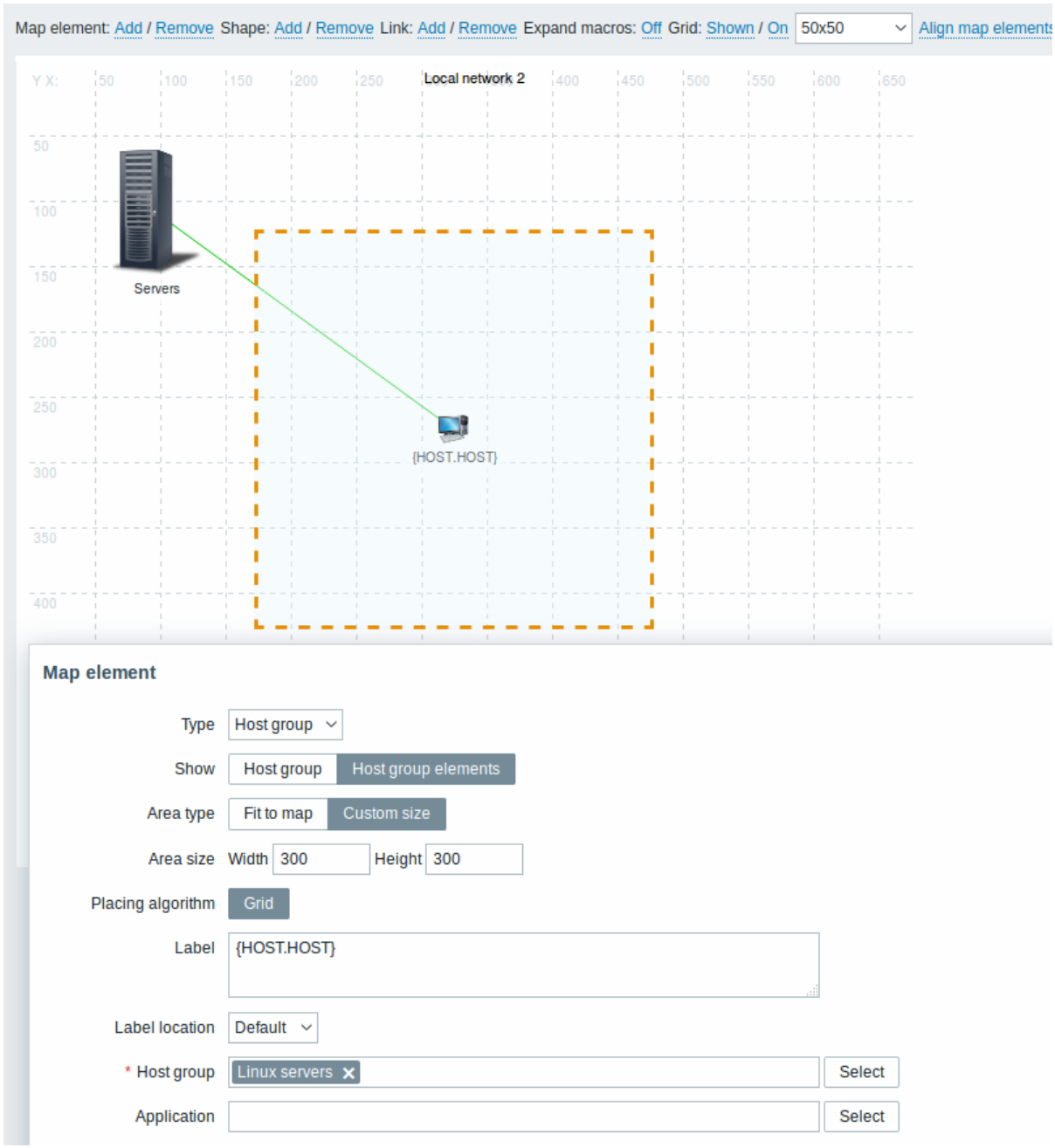


ホストグループ要素

概要

このセクションでは、ネットワークマップを設定する際にホストグループタイプの要素を追加する方法について説明します。

設定



必須入力フィールドには、赤いアスタリスクが付いています。

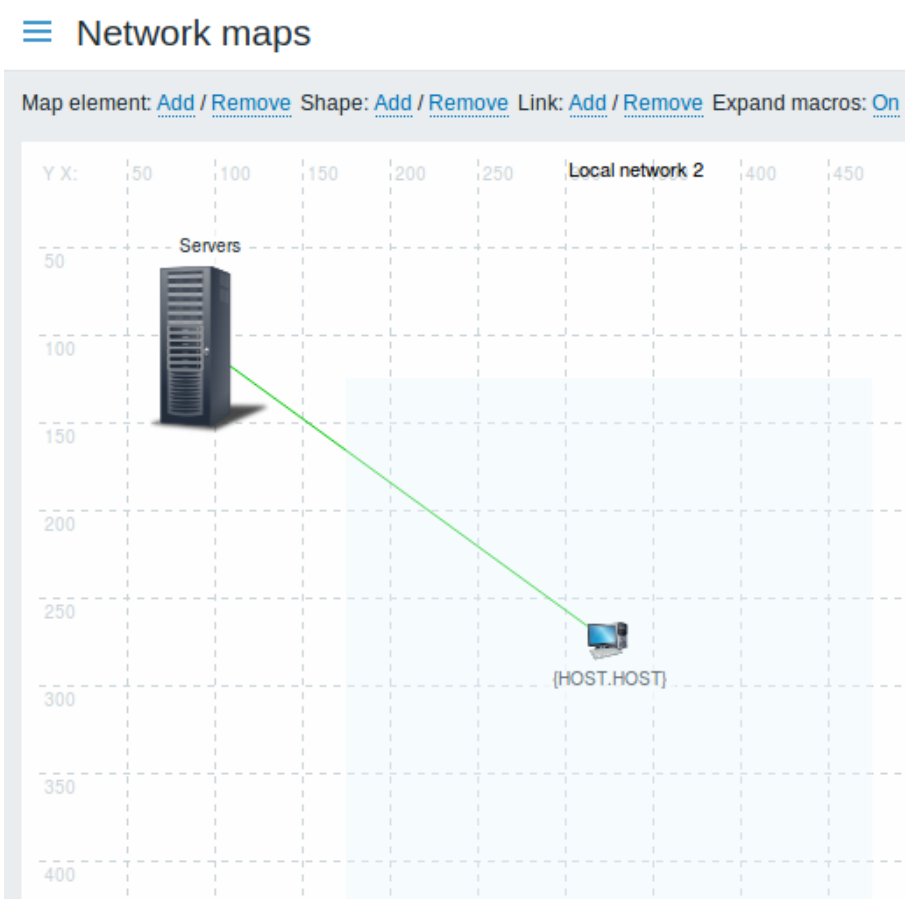
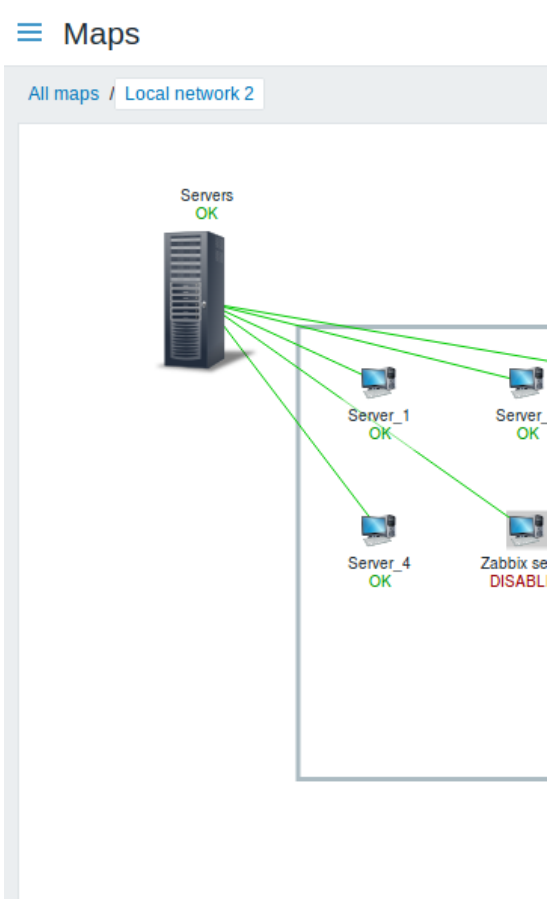
この表には、ホストグループ要素タイプに固有のパラメータが記載されています。

パラメータ	説明
Type	要素のタイプを選択します。 **Host group** - 選択したグループに属するすべてのホストのすべてのトリガーのステータスを表すアイコンです。
Show	表示オプション： **Host group** - このオプションを選択すると、特定のホストグループに関する情報を表示する単一のアイコンが表示されます。 **Host group elements** - このオプションを選択すると、特定のホストグループの各要素（ホスト）に関する情報を表示する複数のアイコンが表示されます。
Area type	*Show* が *Host group elements* に設定されている場合に使用できます。 **Fit to map** - すべてのホストグループ要素をマップ内に均等に配置します。 **Custom size** - 表示するすべてのホストグループ要素に対するマップ領域のサイズを手動で設定します。

パラメータ	説明
Area size	*Area type* が *Custom size* に設定されている場合に使用できます。 **Width** - マップ領域の幅をピクセル単位で指定します。 **Height** - マップ領域の高さをピクセル単位で指定します。
Placing algorithm	**Grid** - すべてのホストグループ要素を表示するために使用できる唯一のオプションです。
Label	アイコンのラベルです。任意の文字列を指定できます。 ラベルにはマクロおよび複数行の文字列を使用できます。 マップ要素のタイプが *Host group* の場合、特定のマクロを指定すると、各ホストに関する情報がマップビューに表示されます。たとえば、{HOST.IP} マクロを使用すると、マップの編集ビューにはマクロ {HOST.IP} 自体のみが表示されますが、マップビューには各ホスト固有の IP アドレスが含まれ、表示されます。

ホストグループ要素の表示

このオプションは、ホストグループ要素の表示オプションが選択されている場合に使用できます。
表示オプションとして ホストグループ要素を選択すると、最初はホストグループを表すアイコンが 1 つだけ表示されます。
ただし、マップを保存してからマップビューに移動すると、特定のホストグループに含まれるすべての要素（ホスト）がマップに表示されます。

マップ編集ビュー	マップビュー
	

{HOST.HOST} マクロが使用されていることに注目してください。
マップ編集ではマクロ名は解決されませんが、マップビューではすべてのホストの一意の名前が表示されます。

リンクインジケーター

概要

ネットワークマップ内の要素間のリンクにインジケーターを割り当てることができます。

インジケーターは、トリガーまたはアイテムの値に基づいて設定できます。異なるリンクのスタイルと色を表示することが可能です。

- トリガーが障害状態になったとき。

- アイテムの値が次の場合:
  - しきい値に達したとき（数値アイテムの場合）。
  - 正規表現に一致したとき（テキストアイテムの場合）。

リンクを設定する際には、デフォルトのリンクタイプと色を設定します。リンクにインジケーターを割り当てることで、リンクのスタイルと色をトリガーの状態またはアイテムの値に応じて変化させることができます。

たとえば、割り当てられたトリガーのいずれかが障害状態になると、それを反映してリンクのスタイルと色が変化します。たとえば、デフォルトのリンクが緑色の線だったとします。このとき、トリガーが障害状態になると、リンクは太い赤線になる場合があります(そのように定義している場合)。

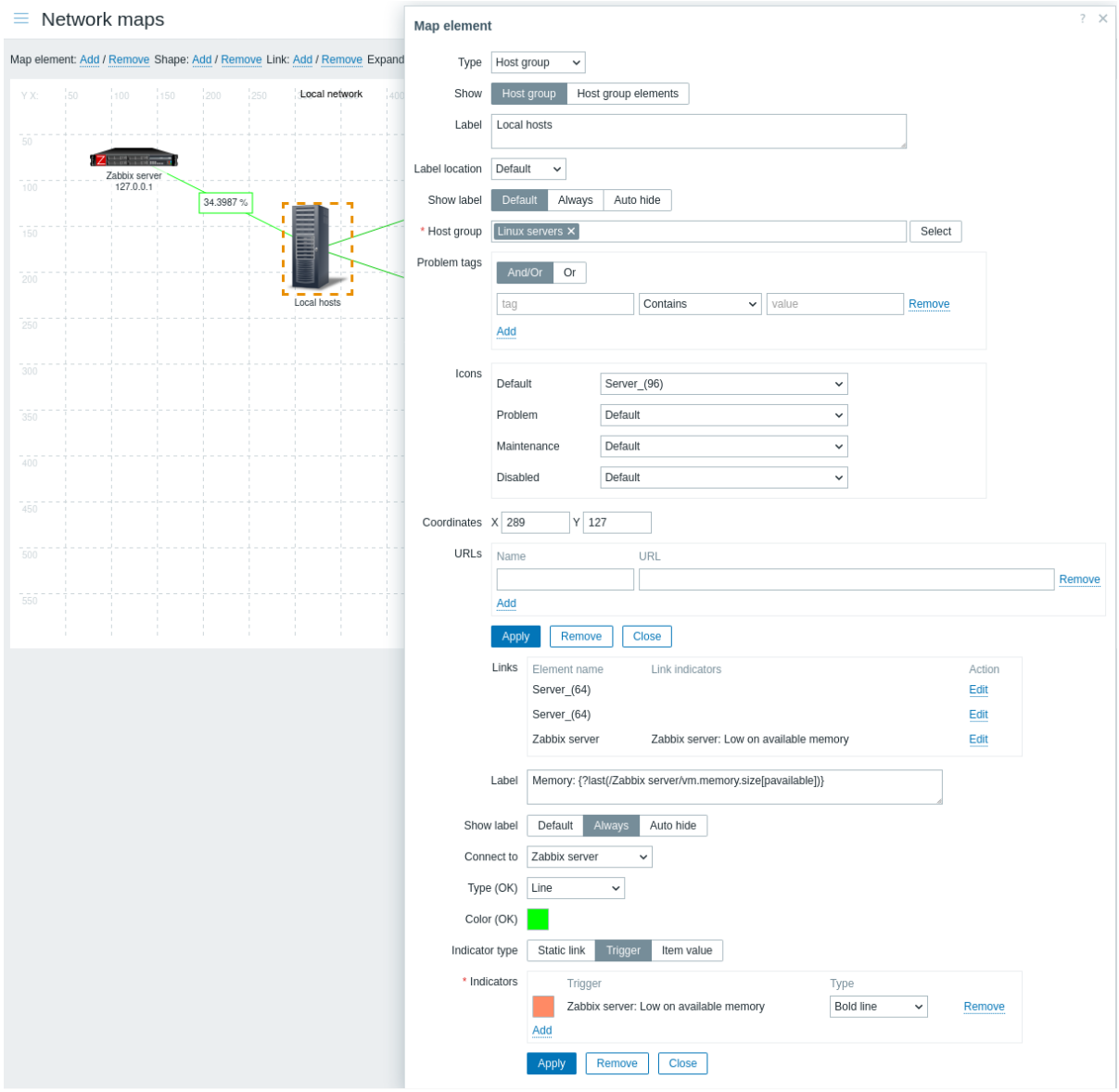
同様に、アイテムの値が指定したしきい値に達した場合や、指定した正規表現に一致した場合にも、リンクのスタイルにそれが反映されます。

設定

トリガー

トリガーをリンクのインジケーターとして割り当てるには、次の手順を実行します。

- マップ要素を選択する
- 適切なリンクの *Links* セクションで *Edit* をクリックする
- インジケータータイプとして *Trigger* を選択する
- *Indicators* ブロックで *Add* をクリックし、1 つ以上のトリガーを選択する

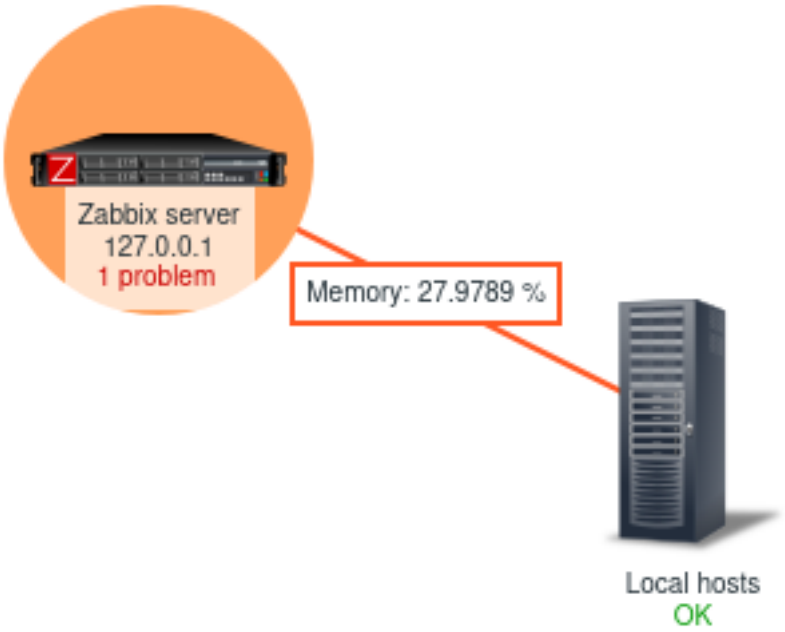


必須入力フィールドには赤いアスタリスクが付いています。

追加したトリガーは *Indicators* 一覧で確認できます。

各インジケーターのリンクタイプと色は、一覧から直接設定できます。完了したら、*Apply* をクリックしてフォームを閉じ、*Update* をクリックしてマップの変更を保存します。

Monitoring > *Maps* では、**trigger** が障害状態になると、該当するリンクタイプと色が表示されます。



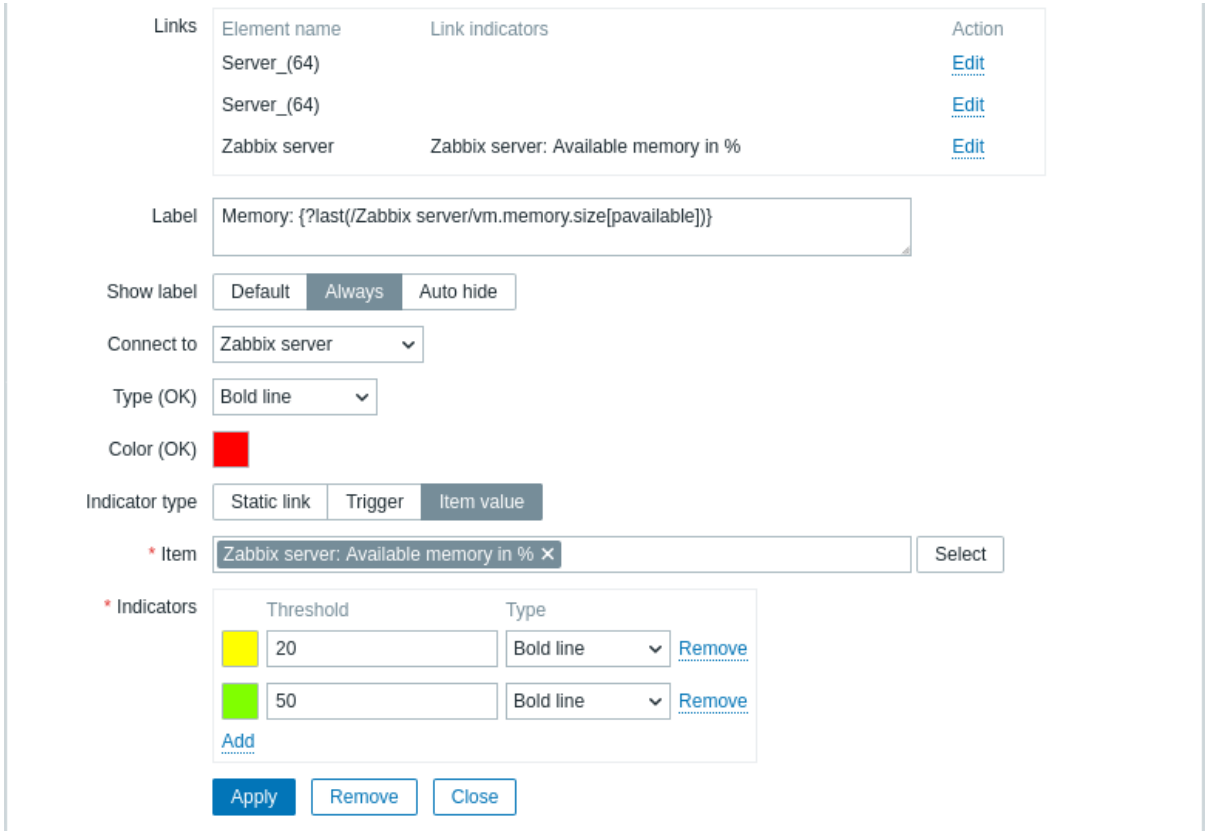
Note:
複数のトリガーが障害状態になった場合、最も重大度の高い障害がリンクのスタイルと色を決定します。同じ重大度の複数のトリガーが同じマップリンクに割り当てられている場合は、ID が最も小さいものが優先されます。なお、次の点にも注意してください。

1. マップ設定の *Minimum trigger severity* と *Show suppressed problem* の設定は、どの障害が考慮されるかに影響します。
2. 複数の障害を持つトリガー（multiple problem generation）の場合、各障害はトリガーの重大度と異なる重大度を持つことがあり（手動で変更された場合）、異なるタグを持つことがあり（マクロによる）、また抑制されることもあります。

アイテム値

アイテム値をリンクのインジケーターとして割り当てるには、次の手順を実行します。

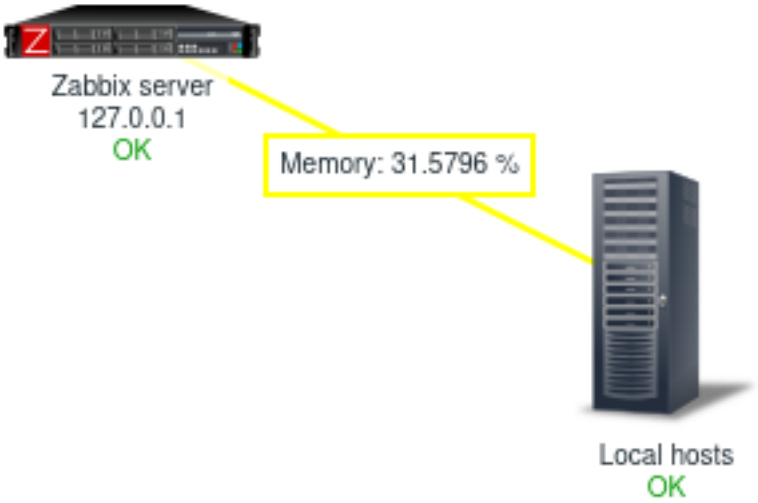
- マップ要素を選択する
- 適切なリンクの *Links* セクションで *Edit* をクリックする
- インジケータータイプとして *Item value* を選択する
- アイテムを選択する
- *Indicators* ブロックで、1 つ以上のアイテム値のしきい値またはパターンを追加する



追加したアイテムのしきい値/パターンは *Indicators* 一覧で確認できます。

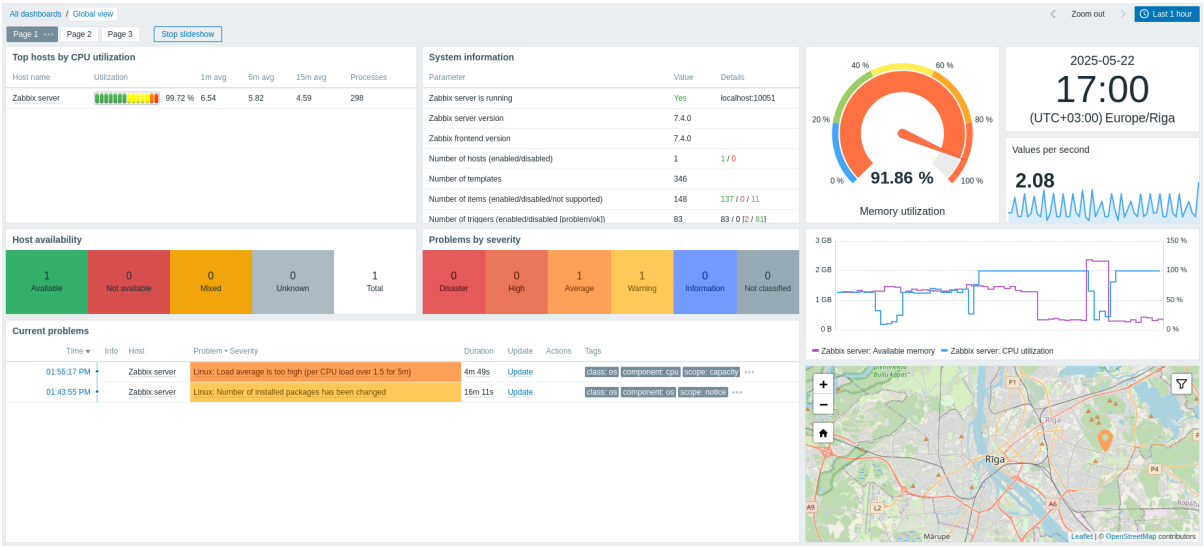
各インジケーターのリンクタイプと色は、一覧から直接設定できます。完了したら、*Apply* をクリックしてフォームを閉じ、*Update* をクリックしてマップの変更を保存します。

Monitoring > *Maps* では、**item value** が設定したしきい値に達した場合（数値データ型の場合）または正規表現パターンに一致した場合（テキストデータ型の場合）に、該当するリンクタイプと色が表示されます。



ダッシュボード

ダッシュボード - グローバルダッシュボード とホストダッシュボード の両方 - は、最新のグラフ、マップ、スライドショーなどのウィジェット やツールを備えた強力な可視化プラットフォームを提供します。



テンプレートとテンプレートグループ

概要

テンプレートを使用することで、作業負荷を軽減し、Zabbix の設定を効率化することができます。テンプレートとは、複数のホストに便利に適用できるエンティティのセットです。

エンティティには以下が含まれます:

- アイテム
- トリガー
- グラフ
- ダッシュボード
- ローレベルディスカバリルール
- Web シナリオ

実際には多くのホストが同一または非常に似ているため、1 つのホストのために作成したエンティティ（アイテム、トリガー、グラフなど）のセットは、多くのホストにも有用であることが自然にわかります。もちろん、それらを各新規ホストにコピーすることもできますが、それでは多くの手作業が必要になります。代わりに、テンプレートを使えば、それらを 1 つのテンプレートにコピーし、必要なだけ多くのホストにテンプレートを適用できます。

テンプレートがホストにリンクされると、テンプレートのすべてのエンティティ（アイテム、トリガー、グラフなど）がホストに追加されます。テンプレートは各ホストに直接割り当てられます（ホストグループには割り当てられません）。

テンプレートは、特定のサービスやアプリケーション（Apache、MySQL、PostgreSQL、Postfix など）ごとにエンティティをグループ化し、それらのサービスを実行しているホストに適用するためによく使用されます。

テンプレートを使用するもう 1 つの利点は、すべてのホストに対して何かを変更する必要がある場合です。テンプレートレベルで 1 回変更すれば、その変更はリンクされているすべてのホストに反映されます。

テンプレートはテンプレートグループで整理されます。

テンプレートの作成と設定に進みます。

標準搭載のテンプレート

概要

Zabbix には、監視対象のセットアップを簡素化し、迅速化するために、事前設定済みのテンプレートが数多く用意されています。

すぐに使用できるすべてのテンプレートは、データ収集 > テンプレートから利用できます。

Zabbix をアップグレードしても、テンプレートは自動的にはアップグレードされません。これは、ユーザーが行ったカスタム変更が上書きされるのを防ぐためです。テンプレートを手動でアップグレードまたは追加するには、以下のテンプレートのアップグレードセクションを参照してください。

特定のテンプレートタイプおよび操作要件に関する情報には、サイドバーを使用してアクセスしてください。

既存の Zabbix テンプレートで動作するように設計されたグローバルダッシュボードは、ダッシュボードリポジトリからダウンロードし、Zabbix 環境にインポートできます。各ダッシュボードの詳細な説明（前提条件を含む）は、ダッシュボードの README ファイルに記載されています。

関連項目:

- テンプレートのリンク
- テンプレートに関する既知の問題

テンプレートのアップグレード

単一のテンプレートをアップグレードまたは追加するには、次の手順を実行します。

1. Zabbix Git リポジトリの `templates` ディレクトリを開きます。

2. Zabbix のバージョンに一致するブランチ（*Source* ラベルの下）を選択します（開発バージョンの場合は *master* など）。

3. 追加またはアップグレードするテンプレートを見つけます（例: MySQL by Zabbix agent）。

4. テンプレートファイルの内容（例: `template_db_mysql_agent.yaml`）をコピーし、ローカルファイルに保存します。

5. Zabbix のユーザーインターフェースで、データ収集 > テンプレートに移動し、ローカルファイルをインポートします。

Note:
Webhook メディアタイプも同様にアップグレードできます。`media` ディレクトリから取得し、アラート > メディアタイプでインポートします。

複数のテンプレートをアップグレードまたは追加するには、次の手順を実行します。

1. Web サーバーに応じて、テンプレートバンドルのアーカイブをダウンロードします。

- Nginx - `templates_bundles-8.0-1M-nginx.zip`
- Apache - `templates_bundles-8.0-2M-apache.zip`

2. アーカイブを展開します。アーカイブには、それぞれ複数のテンプレートを含む YAML ファイルと、各 YAML ファイルに含まれるテンプレートを一覧表示するインデックスファイルがあります。

3. Zabbix のユーザーインターフェースで、データ収集 > テンプレートに移動し、YAML ファイルを 1 つずつインポートします。

Note:
Webhook メディアタイプも同様にアップグレードできます。テンプレートバンドルのアーカイブから `media_bundle-8.0-000.yaml` ファイルを取得し、アラート > メディアタイプでインポートします。

すべてのテンプレートを一度にアップグレードまたは追加するには、テンプレートバンドルのアーカイブに含まれる `import_templates.sh` スクリプトを使用できます。このスクリプトには `curl` および `jq` ユーティリティが必要で、`configuration.import` Zabbix API メソッドを使用します。スクリプトの実行時には、YAML ファイルを含むディレクトリ、Zabbix API のURL、および認証トークンを指定します。

Warning:
`import_templates.sh` スクリプトは、既存のすべてのテンプレートを上書きします。テンプレートの変更を確認したり、元に戻したりするオプションはありません。

例:

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

ホストウィザードとのテンプレート互換性

ホストウィザードは、すべての標準テンプレートに対応しています。

カスタムテンプレートをホストウィザードに対応させることもできます。

1. Zabbix の Web インターフェースで、データ収集 > テンプレートに移動し、カスタムテンプレートをエクスポートします。

2. エクスポートしたファイルを編集します。

- ホストウィザードとの互換性を示すため、`wizard_ready` テンプレート要素を追加します。
- 必要に応じて、`readme` または `config` テンプレート要素を追加します。

3. Zabbix の Web インターフェースで、データ収集 > テンプレートに移動し、更新したテンプレートファイルをインポートします。

Zabbix エージェントテンプレートの操作

Zabbix エージェントでメトリクスを収集するテンプレートを正しく動作させるための手順です。

1. Zabbix エージェントがホストにインストールされていることを確認します。アクティブチェックの場合は、エージェント設定ファイルの `ServerActive` パラメータに Zabbix サーバーまたはプロキシのアドレスが追加されていることも確認します。
2. 対象のホストにテンプレートをリンクします（Zabbix のインストール環境でテンプレートを利用できない場合は、先にテンプレートをインポートする必要があります）。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスを設定し、Zabbix とのデータ共有を許可します。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックすると表示できます）で確認できます。

以下のテンプレートを利用できます。

- Apache by Zabbix エージェント
- Apache by Zabbix エージェント active
- HAProxy by Zabbix エージェント
- IIS by Zabbix エージェント
- IIS by Zabbix エージェント active
- MariaDB by Zabbix エージェント
- MariaDB by Zabbix エージェント active
- Microsoft Exchange Server 2016 by Zabbix エージェント
- Microsoft Exchange Server 2016 by Zabbix エージェント active
- MySQL by Zabbix エージェント
- MySQL by Zabbix エージェント active
- Nginx by Zabbix エージェント
- Nginx by Zabbix エージェント active
- Percona by Zabbix エージェント
- Percona by Zabbix エージェント active
- PHP-FPM by Zabbix エージェント
- PHP-FPM by Zabbix エージェント active
- Podman by Zabbix エージェント
- Podman by Zabbix エージェント active
- PostgreSQL by Zabbix エージェント
- PostgreSQL by Zabbix エージェント active
- RabbitMQ cluster by Zabbix エージェント

Zabbix エージェント **2** テンプレートの運用

Zabbix エージェント 2 でメトリクスを収集するテンプレートを正しく運用するための手順:

1. エージェント 2 がホストにインストールされており、インストール済みのバージョンに必要なプラグインが含まれていることを確認します。場合によっては、先にエージェント 2 をアップグレードする必要があります。
2. テンプレートを対象のホストにリンクします（テンプレートが Zabbix 環境にない場合は、先にテンプレートをインポートする必要があります）。
3. 必要に応じて、テンプレートマクロの値を調整します。ユーザーマクロを使用して設定パラメータを上書きできることに注意してください。
4. 監視対象のインスタンスで、Zabbix とのデータ共有を許可するように設定します。

> **Attention:**
> Zabbix エージェント 2 のテンプレートはプラグインと連携して動作します。基本的な設定はユーザーマクロを調整するだけで行えますが、より詳細なカスタマイズはプラグイン自体を設定することで実現できます。たとえば、プラグインが名前付きセッションをサポートしている場合、設定ファイルで各エンティティごとに独自の URI、ユーザー名、パスワードを持つ名前付きセッションを指定することで、同種の複数のエンティティ（例: MySQL1 と MySQL2）を監視できます。

テンプレートの詳細な説明（マクロ、アイテム、トリガーの完全な一覧を含む）は、テンプレートの README ファイルで確認できます（テンプレート名をクリックするとアクセスできます）。

利用可能なテンプレートは次のとおりです:

- Ceph by Zabbix agent 2

- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

HTTP テンプレートの操作

HTTP エージェントでメトリクスを収集するテンプレートが正しく動作するようにするには、次の手順を実行します。

1. 監視対象に対応するホストを Zabbix に作成します。
   監視対象の IP アドレスまたは DNS 名を、ホストのメインインターフェースとして指定します。
   これは、テンプレートのアイテムで `{HOST.CONN}` マクロを正しく解決するために必要です。
2. 手順 1 で作成したホストにテンプレートをリンクします（テンプレートが Zabbix のインストール環境で利用できない場合は、先にテンプレートをインポートする必要があります）。
3. 必要に応じて、テンプレートレベルのマクロの値を調整します。
4. 監視対象のインスタンスを設定し、Zabbix とのデータ共有を許可します。

テンプレートの詳細な説明（マクロ、アイテム、トリガーの完全な一覧を含む）は、テンプレートの `README` ファイル（テンプレート名をクリックすると表示）で確認できます。

次のテンプレートを利用できます。

- Acronis Cyber Protect Cloud（HTTP）
- Apache（HTTP）
- Asterisk（HTTP）
- AWS（HTTP）
- AWS Cost Explorer（HTTP）
- AWS EC2（HTTP）
- AWS ECS Cluster（HTTP）
- AWS ECS Serverless Cluster（HTTP）
- AWS ELB Application Load Balancer（HTTP）
- AWS ELB Network Load Balancer（HTTP）
- AWS Lambda（HTTP）
- AWS RDS instance（HTTP）
- AWS S3 bucket（HTTP）
- Azure（HTTP）
- Cisco Meraki organization（HTTP）
- Cisco SD-WAN（HTTP）
- Cisco Secure Firewall Threat Defense（HTTP）
- Claude API（HTTP）
- ClickHouse（HTTP）
- Cloudflare（HTTP）
- CockroachDB（HTTP）
- Control-M enterprise manager（HTTP）
- Control-M server（HTTP）
- Cradlepoint NCM v2（HTTP）
- DELL PowerEdge R720（HTTP）
- DELL PowerEdge R740（HTTP）
- DELL PowerEdge R820（HTTP）
- DELL PowerEdge R840（HTTP）
- Domain RDAP（HTTP）
- Elasticsearch Cluster（HTTP）
- Envoy Proxy（HTTP）
- Etcd（HTTP）

- FortiGate（HTTP）
- GitHub repository（HTTP）
- GitLab（HTTP）
- GLPI（HTTP）
- Google Cloud Platform (GCP)（HTTP）
- Hadoop（HTTP）
- HAProxy（HTTP）
- HashiCorp Consul Cluster（HTTP）
- HashiCorp Consul Node（HTTP）
- HashiCorp Nomad（HTTP）
- HashiCorp Vault（HTTP）
- Hikvision camera（HTTP）
- HPE iLO（HTTP）
- HPE MSA 2040 Storage（HTTP）
- HPE MSA 2060 Storage（HTTP）
- HPE Primera（HTTP）
- HPE Synergy（HTTP）
- InfluxDB（HTTP）
- Jenkins（HTTP）
- Kubernetes API server（HTTP）、Kubernetes cluster state（HTTP）、Kubernetes Controller manager（HTTP）、Kubernetes kubelet（HTTP）、Kubernetes nodes（HTTP）、Kubernetes Scheduler（HTTP）は非推奨となり、*Kubernetes Cluster*（*HTTP*）テンプレートに置き換えられています。
- Kubernetes Cluster（HTTP）
- MantisBT（HTTP）
- Microsoft 365 reports（HTTP）
- Microsoft SharePoint（HTTP）
- NetApp AFF A700（HTTP）
- Nextcloud（HTTP）
- NGINX（HTTP）
- NGINX Plus（HTTP）
- Nutanix Prism Element（HTTP）
- OpenAI Platform（HTTP）
- OpenStack（HTTP）
- OpenWeatherMap（HTTP）
- Oracle Cloud（HTTP）
- Palo Alto PA-440（HTTP）
- PHP-FPM（HTTP）
- Podman（HTTP）
- Proxmox VE（HTTP）
- Pure Storage FlashArray v1 and v2（HTTP）
- RabbitMQ cluster（HTTP）
- Ribbon SBC Edge（HTTP）
- Ribbon SBC SWe CE（HTTP）
- Ribbon SBC SWe Core（HTTP）
- TiDB（HTTP）
- TiDB PD（HTTP）
- TiDB TiKV（HTTP）
- Travis CI（HTTP）
- Veeam Backup Enterprise Manager（HTTP）
- Veeam Backup and Replication（HTTP）
- VeloCloud SD-WAN（HTTP）
- VeloCloud SD-WAN Edge（HTTP）
- YugabyteDB（HTTP）
- ZooKeeper（HTTP）

IPMI テンプレートの操作

IPMI テンプレートには、特別な設定は必要ありません。監視を開始するには、link テンプレートを対象のホストにリンクします（テンプレートが Zabbix のインストール環境で利用できない場合は、先にテンプレートをimport する必要がある場合があります）。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイルで確認できます（テンプレート名をクリックすると開けます）。

利用可能なテンプレート:

- Chassis by IPMI

JMX テンプレートの運用

JMX によりメトリクスを収集するテンプレートを正しく運用するための手順:

1. Zabbix Java ゲートウェイ がインストールされ、正しく設定されていることを確認します。
2. テンプレートを対象のホストにリンクします。ホストには JMX インターフェースが設定されている必要があります。テンプレートが Zabbix のインストール環境で利用できない場合は、先にテンプレートをインポートする必要があるかもしれません。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスで、Zabbix とのデータ共有を許可するように設定します。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細は、テンプレートの README ファイルで確認できます（テンプレート名をクリックするとアクセスできます）。

利用可能なテンプレートは次のとおりです。

- Apache ActiveMQ by JMX
- Apache Cassandra by JMX
- Apache Kafka by JMX
- Apache Tomcat by JMX
- GridGain by JMX
- Ignite by JMX
- Jira Data Center by JMX
- WildFly Domain by JMX
- WildFly Server by JMX

ODBC テンプレートの運用

ODBC 監視を介してメトリクスを収集するテンプレートを正しく運用するための手順:

1. 必要な ODBC ドライバーが Zabbix サーバーまたはプロキシにインストールされていることを確認します。
2. リンク して、テンプレートを対象のホストに適用します（テンプレートが Zabbix のインストールに含まれていない場合は、先にテンプレートをインポート する必要がある場合があります）。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスで、Zabbix とのデータ共有を許可するように設定します。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細は、テンプレートの README ファイルで確認できます（テンプレート名をクリックするとアクセスできます）。

利用可能なテンプレートは次のとおりです:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

ネットワークデバイス向けの標準化されたテンプレート

概要

スイッチやルーターなどのネットワークデバイスを監視するために、ネットワークデバイス自体（基本的にはそのシャーシ）用とネットワークインターフェース用の、いわゆる 2 つのモデルを作成しました。

多くのネットワークデバイスファミリー向けのテンプレートが用意されています。すべてのテンプレートは、可能な場合にデバイスから取得できる以下の項目をカバーしています。

- シャーシ障害監視（電源、ファン、温度、全体状態）
- シャーシ性能監視（CPU およびメモリアイテム）
- シャーシ在庫情報の収集（シリアル番号、モデル名、ファームウェアバージョン）
- IF-MIB および EtherLike-MIB を使用したネットワークインターフェース監視（インターフェース状態、インターフェースのトラフィック負荷、Ethernet のデュプレックス状態）

テンプレートが Zabbix のインストール環境にない場合は、最初にインポートする必要があります。

新しい標準搭載テンプレートをインポートする場合は、`@Network interfaces for discovery` グローバル正規表現も次のように更新するとよいでしょう。

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

これにより、多くのシステムでループバックおよび null インターフェースを除外できます。

デバイス

テンプレートが利用可能なデバイスファミリーの一覧:

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certified
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Aruba CX スイッチシリーズ	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certified
Brocade FC SNMP	Brocade	Brocade FC スイッチ	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Performance Fault
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certified
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Performance Fault
Check Point Next Generation Firewall by SNMP	Check Point	次世代ファイアウォール	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certified
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP		CISCO-MEMORY-POOL-MIB, IF-MIB, EtherLike-MIB, SNMPv2-MIB, CISCO-PROCESS-MIB, CISCO-ENVMON-MIB, ENTITY-MIB	Certified
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T かつ 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certified
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX スイッチ	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certified
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Performance Fault Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S シリーズ	S4810		F10-S-SERIES-CHASSIS-MIB	Certified
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certified
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Performance Inventory
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certified
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Huawei AR600 by SNMP	Huawei	Huawei NetEngine AR600 シリーズルーター	Huawei AR600/AR6100/AR6200/AR6300 series routers with VRP software (V300R019/V300R021/V300R023 or later)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500V5/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certified
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infini-band デバイス	Infiniband 12300		ICS-CHASSIS-MIB	Fault Inventory
Juniper SNMP	Juniper	MX,SRX,EX モデル	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certified
Juniper MX NETCONF	Juniper	MX モデル	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	該当なし *(SSH 上の NETCONF を使用)*	Certified
Juniper MX SNMP	Juniper	MX モデル	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certified
Mellanox SNMP	Mellanox	Mellanox Infini-band デバイス	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Routers (CCR シリーズ)	MikroTik CCR1009-7G-1C-1S+, MikroTik CCR1009-7G-1C-1S+PC, MikroTik CCR1009-7G-1C-PC, MikroTik CCR1016-12G, MikroTik CCR1016-12S-1S+, MikroTik CCR1036-12G-4S-EM, MikroTik CCR1036-12G-4S, MikroTik CCR1036-8G-2S+, MikroTik CCR1036-8G-2S+EM, MikroTik CCR1072-1G-8S+, MikroTik CCR2004-16G-2S+, MikroTik CCR2004-1G-12S+2XS 用の個別の専用テンプレートが利用可能です	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (CRS シリーズ)	MikroTik CRS106-1C-5S, MikroTik CRS109-8G-1S-2HnD-IN, MikroTik CRS112-8G-4S-IN, MikroTik CRS112-8P-4S-IN, MikroTik CRS125-24G-1S-2HnD-IN, MikroTik CRS212-1G-10S-1S+IN, MikroTik CRS305-1G-4S+IN, MikroTik CRS309-1G-8S+IN, MikroTik CRS312-4C+8XG-RM, MikroTik CRS317-1G-16S+RM, MikroTik CRS326-24G-2S+IN, MikroTik CRS326-24G-2S+RM, MikroTik CRS326-24S+2Q+RM, MikroTik CRS328-24P-4S+RM, MikroTik CRS328-4C-20S-4S+RM, MikroTik CRS354-48G-4S+2Q+RM, MikroTik CRS354-48P-4S+2Q+RM 用の個別の専用テンプレートが利用可能です	RouterOS/SwitchOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (CSS シリーズ)	MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN 用の個別の専用テンプレートが利用可能です	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S 用の個別の専用テンプレートが利用可能です	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R 用の個別の専用テンプレートが利用可能です	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	MikroTik PowerBox, MikroTik PowerBox Pro 用の個別の専用テンプレートが利用可能です	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik RB<device model> SNMP	MikroTik	MikroTik RB シリーズルーター	MikroTik RB1100AHx4, MikroTik RB1100AHx4 Dude Edition, MikroTik RB2011iL-IN, MikroTik RB2011iL-RM, MikroTik RB2011iLS-IN, MikroTik RB2011UiAS-IN, MikroTik RB2011UiAS-RM, MikroTik RB260GS, MikroTik RB3011UiAS-RM, MikroTik RB4011iGS+RM, MikroTik RB5009UG+S+IN 用の個別の専用テンプレートが利用可能です	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik SNMP	MikroTik	MikroTik RouterOS デバイス	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Fault Inventory
QTech QSW SNMP	QTech	Qtech デバイス	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Performance Inventory

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Stormshield SNS by SNMP	Stormshield	Stormshield Network Security (SNS) ファイアウォール	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Certified
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Performance Inventory
Ubiquiti AirOS SNMP	Ubiquiti	Ubiquiti AirOS ワイヤレスデバイス	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Performance
Vyatta Virtual Router by SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Performance Inventory

テンプレート設計

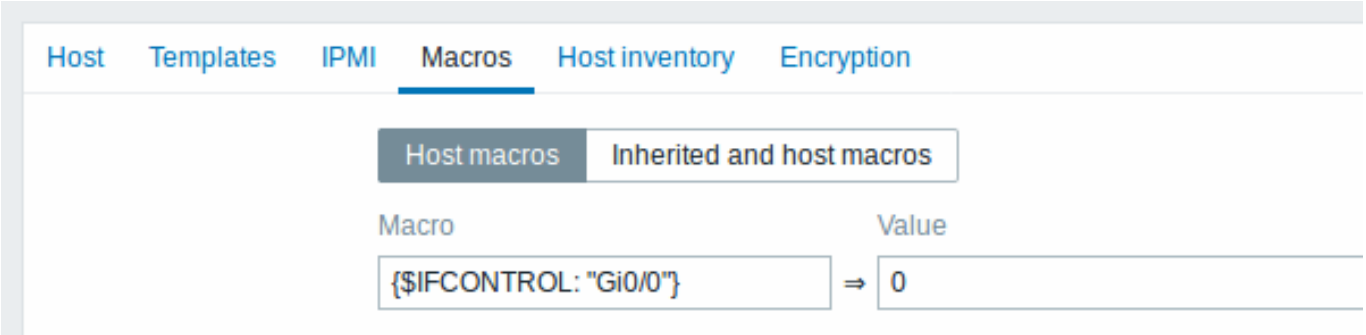
テンプレートは以下の点を考慮して設計されています。

- トリガーをユーザーが調整できるよう、できるだけユーザーマクロを使用しています。
- サポートされていないアイテムの数を最小限に抑えるため、できるだけローレベルディスカバリを使用しています。
- すべてのテンプレートは Template ICMP Ping に依存しているため、すべてのデバイスが ICMP でもチェックされます。
- アイテムでは MIB を使用せず、SNMP OID をアイテムやローレベルディスカバリで使用しています。そのため、テンプレートを動作させるために Zabbix に MIB をロードする必要はありません。
- ディスカバリ時にループバックインターフェースや ifAdminStatus = down(2) のインターフェースをフィルタリングします。
- 可能な場合は IF-MIB::ifXTable の 64bit カウンタを使用します。サポートされていない場合は、デフォルトの 32bit カウンタを使用します。

ディスカバリされたすべてのネットワークインターフェースには、その運用状態（リンク）を監視するトリガーがあります。例えば：

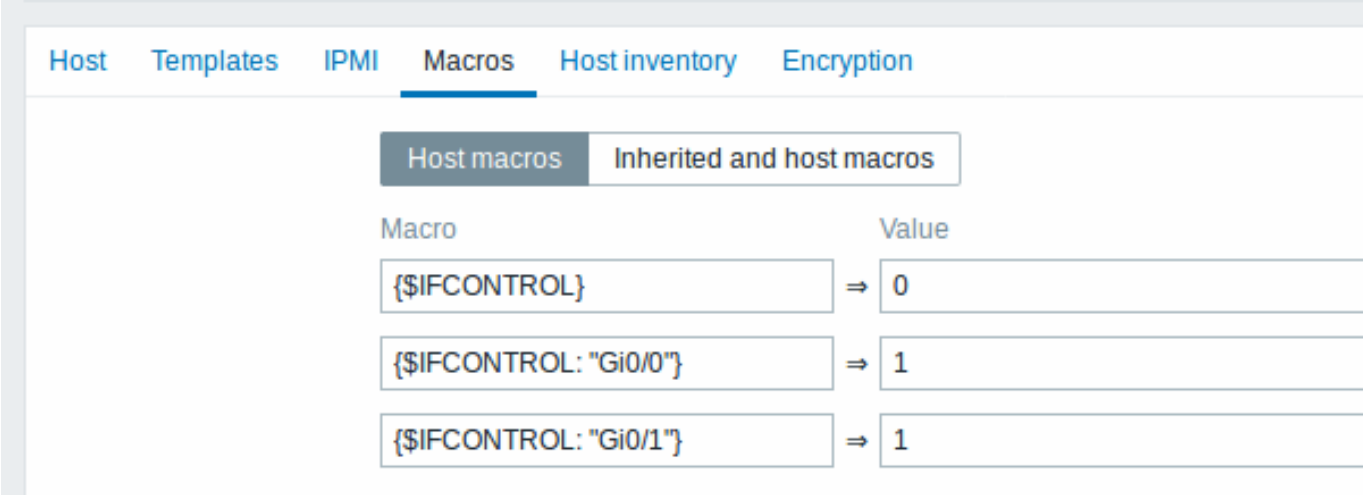
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 特定のインターフェースでこの条件を監視したくない場合は、値を 0 としたコンテキスト付きユーザーマクロを作成してください。例：



ここで、Gi0/0 は {#IFNAME} です。これにより、この特定のインターフェースではトリガーが使用されなくなります。

- すべてのトリガーが発火しないようにデフォルトの動作を変更し、アップリンクなど限られたインターフェースだけでこのトリガーを有効にすることもできます。



タグ

- パフォーマンス - デバイスファミリー MIB は、CPU およびメモリアイテムを監視する方法を提供します。
- 障害 - デバイスファミリー MIB は、少なくとも 1 つの温度センサーを監視する方法を提供します。
- インベントリ - デバイスファミリー MIB は、少なくともデバイスのシリアル番号およびモデル名を収集する方法を提供します。
- 認定済み - 上記の 3 つの主要カテゴリすべてがカバーされています。

VMware テンプレートの操作

概要

Zabbix には、VMware vCenter および ESXi ハイパーバイザー用に設計された仮想マシン監視用のすぐに使えるテンプレートが用意されています。利用可能なテンプレートは、2 つの別々のテンプレートセットに含まれています。

- VMware - 対応するマクロに UUID データを使用
- VMware FQDN - 対応するマクロに FQDN データを使用

VMware vCenter または ESXi ハイパーバイザーの監視用に設計されたテンプレートには、仮想インストールの監視用にさまざまな組み込みチェックとともに、事前構成された低レベルディスカバリールールが含まれています。

Note:
VMware FQDN テンプレートが正しく機能するためには、監視対象の各仮想マシンが FQDN のルールに従った一意の OS 名を持っている必要があります。さらに、VMware Tools/Open Virtual Machine tools をすべてのマシンにインストールする必要があります。これらの前提条件が満たされている場合は、*VMware FQDN* テンプレートの使用を推奨します。*VMware FQDN* テンプレートは、カスタムインターフェースでホストを作成する機能が導入された Zabbix 5.2 以降で利用可能です。

 従来の *VMware* テンプレートも利用可能であり、FQDN の要件が満たされていない場合に使用できます。ただし、*VMware* テンプレートには既知の問題があります。検出された仮想マシン用のホストは、vCenter に保存されている名前で作成されます（例：”VM1”、”VM2” など）。これらのホストに Zabbix agent がインストールされていて、アクティブな Zabbix agent の自動登録が有効になっている場合、自動登録プロセスは起動時に登録されたホスト名（例：”vm1.example.com”、”vm2.example.com” など）を読み取ります。このため、既存の仮想マシンに対して新しいホストが作成される可能性があり（名前の一致が見つからないため）、異なる名前の重複したホストが作成されることになります。

ご注意ください:

- テンプレート *VMware Hypervisor* は、ホストに手動でリンクすることも、ディスカバリーで使用することもできます。
- テンプレート *VMware Guest* は、ホストに手動でリンクしてはいけません。ディスカバリーでのみ使用できます。

ホストマクロの設定

VMware のシンプルチェックを使用するには、ホストに以下のユーザーマクロを定義する必要があります。

- {$VMWARE.URL} - VMware サービス（vCenter または ESXi ハイパーバイザー）SDK の URL（https://servername/sdk）
- {$VMWARE.USERNAME} - VMware サービスのユーザー名
- {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーのパスワード

OTLP テンプレートの操作

Zabbix には、*Telemetry query* アイテムを使用して OpenTelemetry のトレース、メトリクス、ログを収集および監視するための、すぐに使用できるさまざまなテンプレートが用意されています。

テンプレートを正しく動作させるための手順:

1. テンプレート固有の設定要件について、テンプレートの `README` ファイルを確認します。
2. 監視対象に対応するホストを Zabbix に作成します。
3. Zabbix のサーバーまたは Zabbix のプロキシの設定ファイルで、`TelemetryProvider` パラメータが設定されていることを確認します。
4. 作成したホストにテンプレートをリンクします（Zabbix のインストール環境でテンプレートを利用できない場合は、先にテンプレートをインポートする必要があります）。
5. 必要に応じて、テンプレートレベルのマクロの値を調整します。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレート名をクリックしてアクセスできるテンプレートの `README` ファイルに記載されています。

以下のテンプレートを利用できます:

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

イベント発生時の通知

概要

いくつかのアイテムとトリガーを構成し、トリガーの状態変更の結果としていくつかのイベントが発生していると仮定して、いくつかのアクションを検討します。

そもそも、トリガーやイベントのリストを常に眺めたくはありません。何か重大な、障害などが発生した場合に通知を受け取る方がはるかに良いでしょう。また、障害が発生した場合には、関係者全員に周知したいと考えております。

そのため、通知の送信は Zabbix が提供する主要なアクションの 1 つです。特定のイベントが発生したときに、誰に、いつ、通知するかを定義できます。

Zabbix から通知を送受信できるようにするには、次のことを行う必要があります。

- メディアの定義
- 定義されたメディアにメッセージを送信するアクションの設定

アクションは条件と操作で構成されます。基本的に条件が満たされた場合に操作が実行されます。2 つの主要な操作は、メッセージの送信 (通知) とリモートコマンドの実行です。

ディスカバリおよび自動登録によって作成されたイベントについては、いくつかの追加操作を使用できます。追加操作はホストの追加または削除、テンプレートのリンクなどです。

メディアタイプ

概要

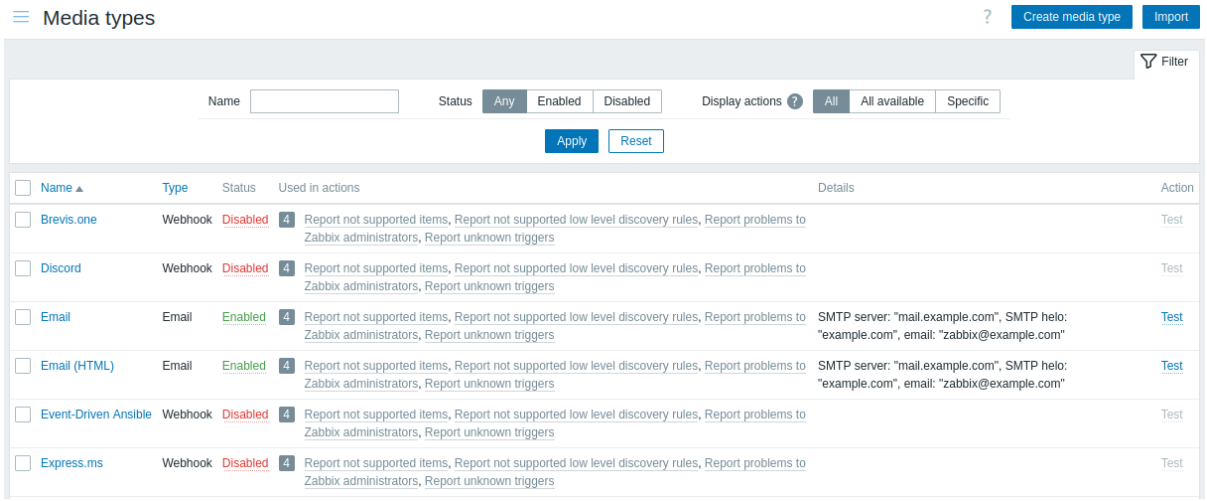
メディアタイプは、Zabbix から通知やアラートを送信するために使用する配信チャネルです。

メディアタイプでは、次の配信方法をサポートしています。

- メール

- プッシュ通知
- SMS
- カスタムスクリプト
- Webhook

メディアタイプは、*Alerts* > *Media types* で管理します。デフォルトのデータセットには、あらかじめ定義されているメディアタイプが含まれています。
それらのパラメータを調整するだけで、使用できるようになります。



メディアタイプがアラート処理の中でどのように機能するかを確認するため、Zabbix からエンドユーザーに通知を配信するために必要な 3 つの要件を見てみましょう。

1. 通知を送信するアクションの実行内容を定義する必要があります。
2. 動作するメディアタイプを定義する必要があります（SMTP を使用してアラートを送信するメールなど）。
3. ユーザーレベルの配信先情報（メールアドレス、電話番号など）をユーザーメディアで定義する必要があります。

メディアタイプのテスト

設定したメディアタイプが機能するかテストするには、メディアタイプの一覧で テストをクリックします。

テストリクエストは Zabbix サーバーに送信されます。Zabbix サーバーは、指定したメディアタイプを使用してアラートの送信を試み、その結果を Web インターフェースに返します。Web インターフェースは、サーバーから結果が返されるまで待機します。メディアタイプのテストには、デフォルトで 65 秒のタイムアウトが設定されています（管理 > 一般 > タイムアウト で設定可能）。

以下のテストの詳細も参照してください。

- メール
- プッシュ通知
- Webhook
- スクリプト

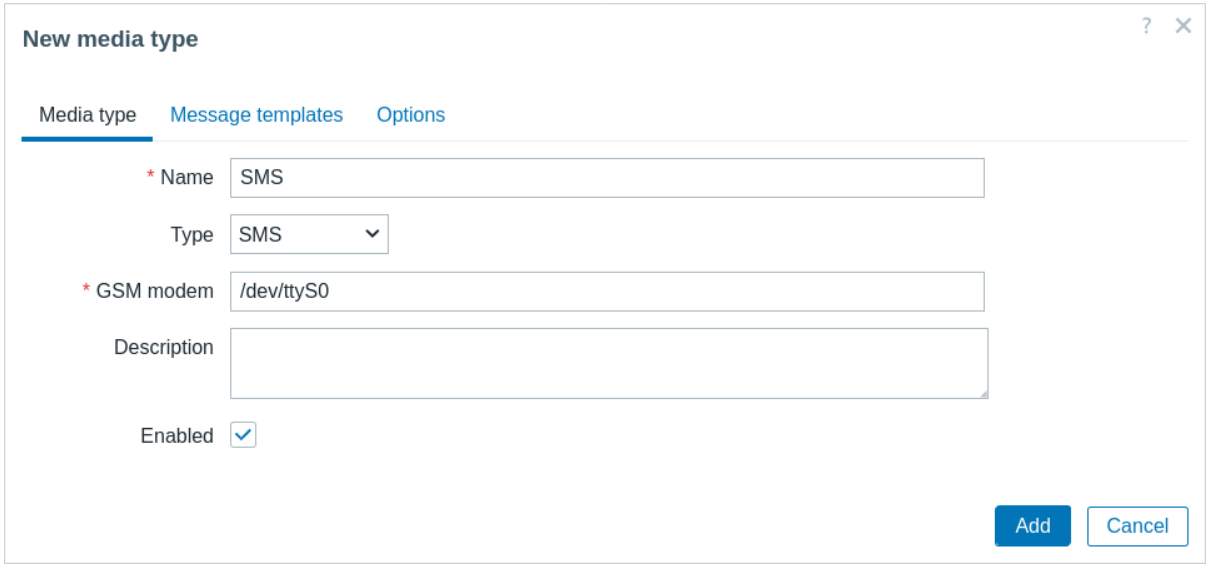
設定

Zabbix の Web インターフェースでメディアタイプを作成するには、次の手順に従います。

1. *Alerts* > *Media types* に移動します。
2. *Create media type* をクリックします。
3. フォームにメディアタイプのパラメータを入力します。

一部のパラメータは、すべての配信方法で共通です。

Note:
Web インターフェース設定ファイル（`zabbix.conf.php`）の `$ZBX_FEATURE_FLAGS['media_type_denylist']` 機能フラグで 1 つ以上の配信方法が指定されている場合、それらの配信方法は *Type* リストで使用できません。

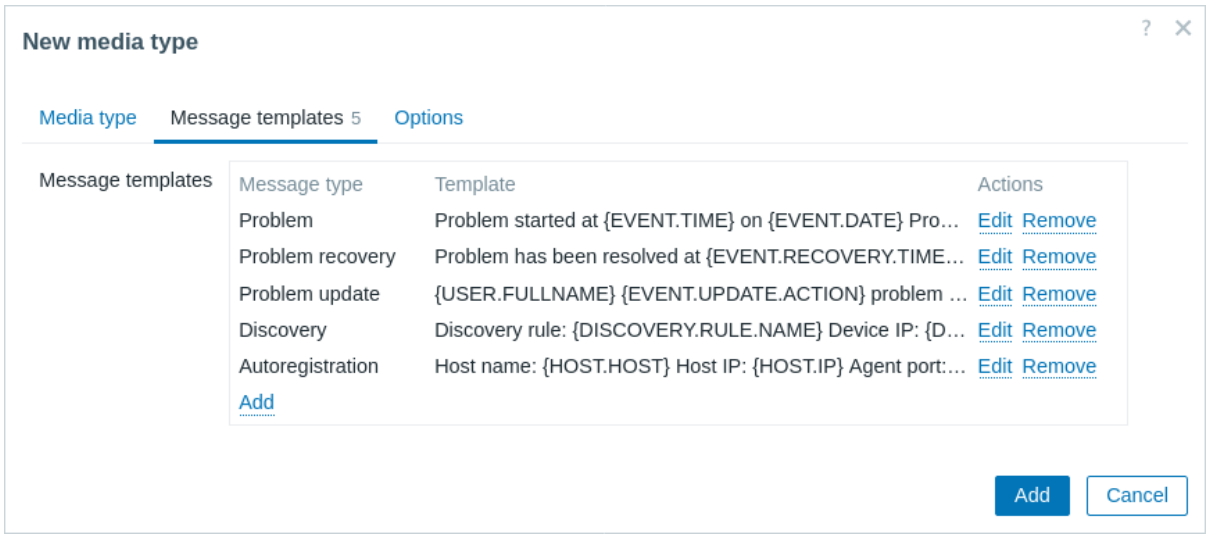


パラメータ	説明
Name	メディアタイプの名前。
Type	メディアタイプの配信方法を選択します。
Description	メディアタイプの説明を入力します。
Enabled	メディアタイプを有効にするには、チェックボックスをオンにします。

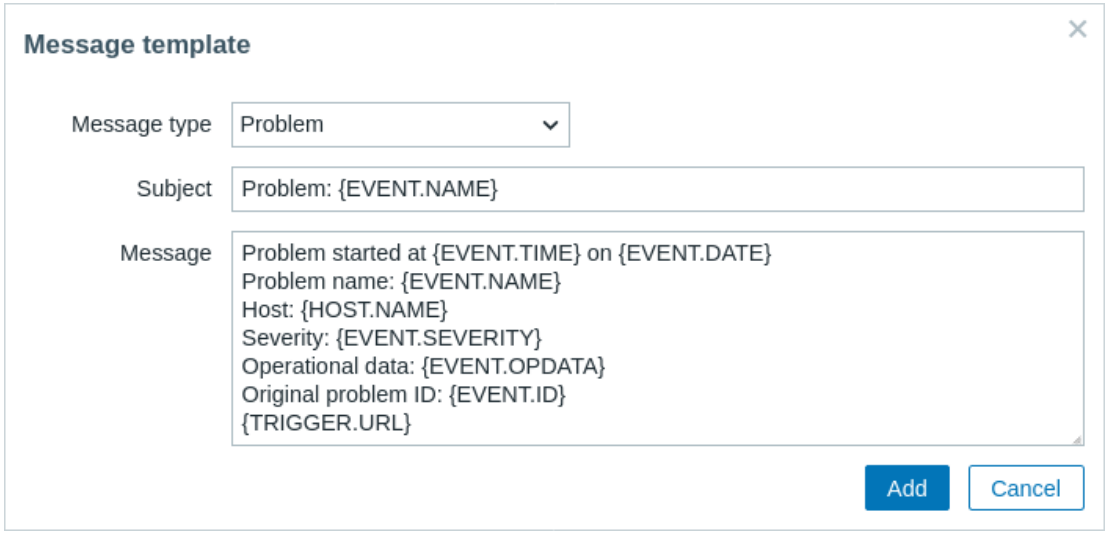
方法固有のパラメータについては、メール、プッシュ通知、SMS、カスタム alertscript、またはwebhookのページを参照してください。

メッセージテンプレート

メッセージテンプレートタブには、イベントタイプごとのデフォルトメッセージ（障害、障害復旧、ディスカバリなど）が含まれます。



追加をクリックしてデフォルトメッセージを定義するか、既存のメッセージを更新するには 編集をクリックします。



Parameter	Description
Message type	デフォルトメッセージを使用するイベントの種類。 各イベントタイプにつき、定義できるデフォルトメッセージは 1 つだけです。
Subject	デフォルトメッセージの件名。件名にはマクロを含めることができます。255 文字までです。 件名は SMS メディアタイプでは使用できません。
Message	デフォルトメッセージです。データベースの種類に応じて、使用できる文字数に制限があります（詳細はメッセージの送信 を参照してください）。 メッセージには、サポートされているマクロ を含めることができます。 障害および障害更新メッセージでは、式マクロがサポートされています（例: `{?avg(/host/key,1h)}`）。

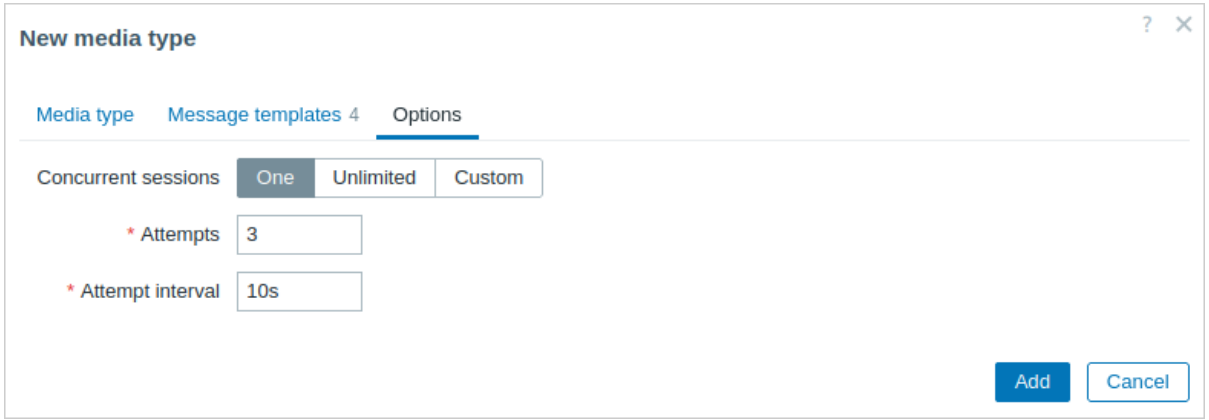
デフォルトメッセージは、アクションオペレーション で定義されている場合、カスタムメッセージによって上書きされることに注意してください。

Warning:
メッセージテンプレートの定義は、webhook やカスタムアラートスクリプトを含むすべての配信方法で必須です。これらは通知にデフォルトメッセージを使用しません。
たとえば、アクション「Send message to Pushover webhook」は、Pushover webhook の障害メッセージが定義されていない場合、障害通知の送信に失敗します。

オプション

オプションタブには、アラート処理の設定が含まれています。
同じオプションセットをメディアタイプごとに設定できます。

すべてのメディアタイプは並列で処理されます。
同時セッションの最大数はメディアタイプごとに設定できますが、
サーバー上のアラートプロセスの総数は、StartAlerters パラメータでのみ制限できます。
1 つのトリガーによって生成されたアラートは順番に処理されます。
そのため、複数の通知が同時に処理されるのは、複数のトリガーによって生成された場合のみです。



パラメータ	説明
同時セッション数	メディアタイプで使用する並列アラートセッションの数を選択します。 **1** - 1 セッション 無制限 - セッション数を無制限にする カスタム - セッション数を指定する 無制限または大きな値を設定すると、より多くのセッションを並列で実行でき、通知の送信容量が増加します。無制限または大きな値は、多数の通知を同時に送信する必要がある大規模な環境で使用してください。 同時セッション数を超える通知を送信する必要がある場合、残りの通知はキューに入れられます。通知が失われることはありません。
試行回数	通知の送信を試行する回数です。最大 100 回まで指定できます。デフォルト値は「3」です。「1」を指定した場合、Zabbix は通知を 1 回だけ送信し、送信に失敗しても再試行しません。
試行間隔	送信に失敗した場合に通知の再送を試行する間隔を秒単位で指定します（0～3600）。「0」を指定した場合、Zabbix は直ちに再試行します。 時間のサフィックスがサポートされています（例: 30s、1m、1h）。

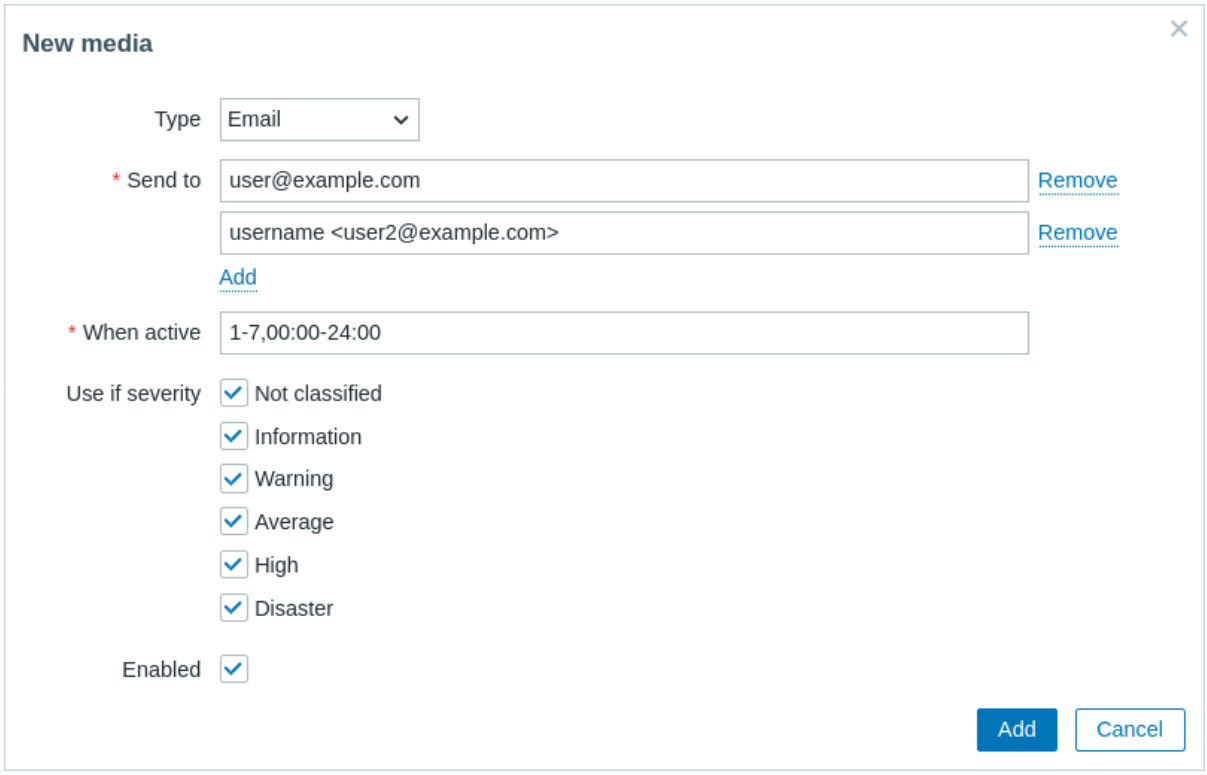
ユーザーメディア

メディアタイプでは通知の送信方法を定義するのに対し、ユーザーメディアでは通知の送信先を定義します。

配信方法にかかわらず、ユーザーメディア（メールアドレス、webhook ユーザー ID など）はユーザープロファイルで定義する必要があります。webhook *X* を使用して *Admin* ユーザーにメッセージを送信するアクションは、webhook *X* の配信詳細が Admin ユーザーのプロファイルで定義されていない場合、配信に失敗します。

ユーザーメディアを定義するには、次の手順に従います。

1. ユーザー > ユーザーに移動し、ユーザープロパティフォームを開きます。
   または、自分のユーザープロファイルの ユーザー設定 > 通知に移動します。
2. メディアタブで 追加をクリックします。
3. フォームにユーザーメディアのパラメータを入力します。



パラメータ	説明
タイプ	ドロップダウンリストには、有効になっているメディアタイプの名前が表示されます。 無効になっているメディアタイプのメディアを編集する場合、タイプは赤色で表示されます。

パラメータ	説明
送信先	メッセージの送信先となる連絡先情報を入力します。 メールメディアタイプでは、アドレスフィールドの下にある Add ボタンをクリックして、複数のアドレスを追加できます。この場合、通知は一覧にあるすべてのアドレスに送信されます。アドレスの例については、メールメディアタイプの *Email* パラメータの説明を参照してください。 プッシュ通知メディアタイプでは、ユーザーのアクティブなモバイルデバイスから選択するか、1台以上の特定のデバイスを追加できます。
有効な時間帯	メッセージを送信する時間を制限できます。たとえば、平日のみ（1-5,09:00-18:00）に設定できます。この制限は、ユーザーのタイムゾーンに基づくことに注意してください。ユーザーのタイムゾーンが変更され、システムのタイムゾーンと異なる場合、重要なメッセージを見逃さないように、この制限を適宜調整する必要があります。 形式の説明については、期間の指定ページを参照してください。 ユーザーマクロがサポートされています。
重大度に応じて使用	通知を受信するトリガーの重大度のチェックボックスを選択します。 トリガー以外のイベントに関する通知を受信する場合、デフォルトの重大度（「未分類」）に必ずチェックを入れてください。 保存すると、選択したトリガーの重大度は対応する重大度の色で表示され、選択していないものはグレー表示されます。
ステータス	ユーザーメディアのステータス。 有効 - 使用中です。 無効 - 使用されていません。

メール

概要

メッセージの配信チャネルとしてメールを設定するには、メールをメディアタイプとして設定し、特定のアドレスをユーザーに割り当てる必要があります。

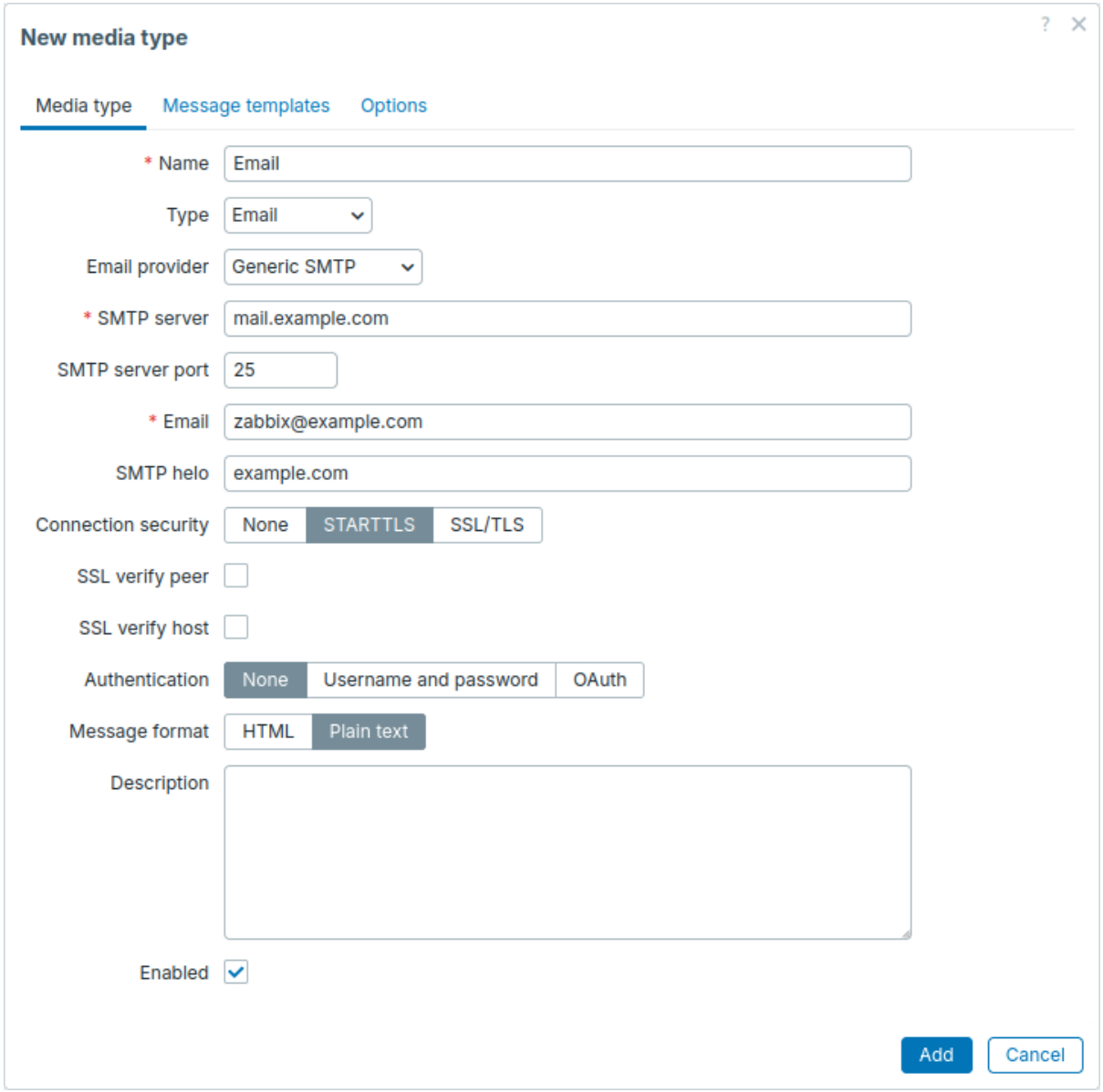
> **Note:**
> 単一イベントに対する複数の通知は、同じメールスレッドにまとめられます。

設定

メールをメディアタイプとして設定するには、次の手順を実行します。

1. *Alerts* > *Media types* に移動します。
2. *Create media type* をクリックします（または、事前定義されたメディアタイプの一覧で *Email* をクリックします）。
3. フォームにメールメディアタイプのパラメータを入力します。

Media type タブには、メディアタイプの一般属性が含まれます。



すべての必須入力フィールドには、赤いアスタリスクが付いています。

Warning:
メールメディアタイプの設定をエクスポートすると、パスワードは平文でエクスポートされます。

以下のパラメータは、メールメディアタイプ固有のものです。

パラメータ	説明
Email provider	メールプロバイダーを選択します。*Generic SMTP*、*Gmail*、*Gmail relay*、*Office365*、または *Office365 relay* から選択できます。 Gmail または Office365 関連のオプションを選択した場合、送信者のメールアドレスとパスワードのみを指定する必要があります。*SMTP server*、*SMTP server port*、*SMTP helo*、*Connection security* などのオプションは、Zabbix によって自動的に入力されます。自動化された Gmail/Office365 メディアタイプも参照してください。
SMTP server	送信メッセージを処理する SMTP サーバーを設定します。 このフィールドは、メールプロバイダーとして *Generic SMTP* が選択されている場合に使用できます。
SMTP server port	送信メッセージを処理する SMTP サーバーのポートを設定します。 このフィールドは、メールプロバイダーとして *Generic SMTP* が選択されている場合に使用できます。

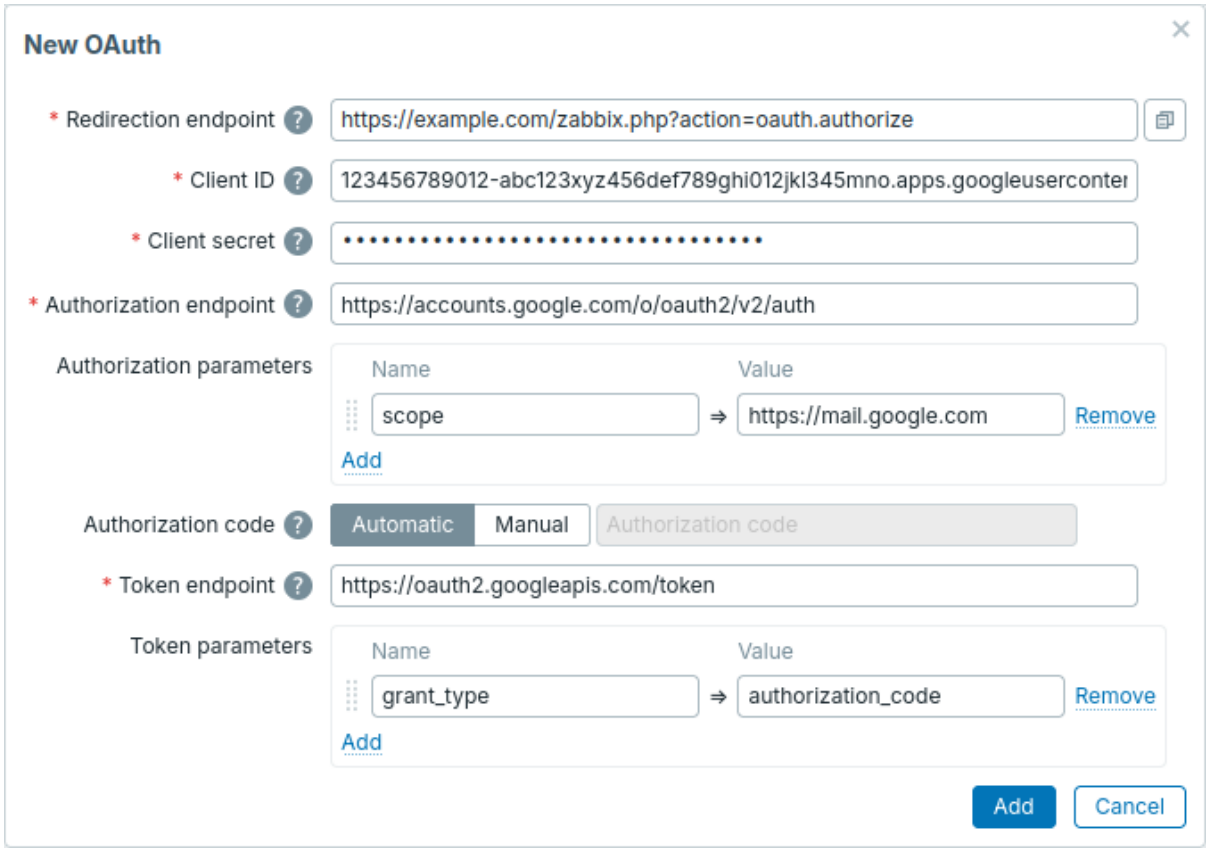
パラメータ	説明
Email	ここに入力したアドレスが、送信メッセージの **From** アドレスとして使用されます。 実際のメールアドレスに送信者の表示名を追加することもできます（例: `Zabbix_info <zabbix@company.com>`）。 Zabbix のメールにおける表示名には、RFC 5322 で許可されているものと比較して、いくつかの制限があります。以下に例を示します。 有効な例: `zabbix@company.com`（メールアドレスのみ。山括弧は不要） `Zabbix_info <zabbix@company.com>`（表示名の後に、山括弧で囲んだメールアドレスを指定） `Ω-monitoring <zabbix@company.com>`（表示名に UTF-8 文字を使用） 無効な例: `Zabbix HQ zabbix@company.com`（表示名はあるが、メールアドレスが山括弧で囲まれていない） `"Zabbix@<H(comment)Q>" <zabbix@company.com>`（RFC 5322 では有効ですが、引用符付きペアとコメントは Zabbix のメールではサポートされません）。
SMTP helo	通常はドメイン名である、正しい SMTP helo 値を設定します。 空の場合、メールのドメイン名（つまり、*Email* フィールドの @ より後の部分）が送信されます。ドメイン名を取得できない場合は、デバッグレベルの警告がログに記録され、HELO コマンドのドメインとしてサーバーのホスト名が送信されます。 このフィールドは、メールプロバイダーとして *Generic SMTP* が選択されている場合に使用できます。
Connection security	接続セキュリティのレベルを選択します。 **None** - CURLOPT_USE_SSL オプションを使用しません **STARTTLS** - CURLUSESSL_ALL 値を指定して CURLOPT_USE_SSL オプションを使用します **SSL/TLS** - CURLOPT_USE_SSL の使用は任意です。
SSL verify peer	SMTP サーバーの SSL 証明書を検証するには、チェックボックスを選択します。 証明書の検証のため、サーバー設定ディレクティブ「SSLCALocation」の値を CURLOPT_CAPATH に指定する必要があります。 これにより、cURL オプション CURLOPT_SSL_VERIFYPEER が設定されます。
SSL verify host	SMTP サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスを選択します。 これにより、cURL オプション CURLOPT_SSL_VERIFYHOST が設定されます。
Authentication	認証レベルを選択します。 **None** - cURL オプションを設定しません **Username and password** - 「AUTH=*」を指定し、認証メカニズムの選択を cURL に任せます **OAuth** - OAuth 認証 OAuth 認証は、*Office365 relay* メールプロバイダーではサポートされません。
Username	認証に使用するユーザー名です。 これにより、CURLOPT_USERNAME の値が設定されます。 ユーザーマクロがサポートされています。
Password	認証に使用するパスワードです。 これにより、CURLOPT_PASSWORD の値が設定されます。 ユーザーマクロがサポートされています。 ここに入力した値は、メディアタイプの設定をエクスポートすると、そのままエクスポートされます。エクスポートされたファイルに機密性の高い認証情報が露出するのを避けるには、リテラルパスワードの代わりにユーザーマクロ（例: `{$EMAIL_SMTP_PASSWORD}`）を使用してください。マクロ参照はエクスポートされますが、秘密の値は移行先システムで設定または再設定する必要があります。
OAuth tokens	*Configure* をクリックすると、新しいウィンドウでOAuth トークンを取得するためのパラメータを設定できます。 このフィールドは、*Authentication* フィールドで「OAuth」が選択されている場合にのみ使用できます。
Message format	メッセージ形式を選択します。 **HTML** - HTML として送信します。 **Plain text** - プレーンテキストとして送信します。

Attention:
SMTP 認証を有効にするには、Zabbix サーバーを `--with-libcurl` コンパイルオプション付きでコンパイルする必要があります。このオプションには libcurl が必要です（バージョンの詳細については、サーバーの要件を参照してください）。

デフォルトメッセージおよびアラート処理オプションの設定方法については、共通メディアタイプパラメータも参照してください。

OAuth トークン

OAuth トークンを取得するには、以下のパラメーターが必要です。



Warning:
メールメディアタイプの設定をエクスポートすると、Client secret は平文でエクスポートされます。

Parameter	Description
Redirection endpoint	OAuth サービスが OAuth 認可を返す Zabbix Web インターフェースの URL を入力します（形式: `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`）。 frontend URL が定義されている場合、新しいメールメディアタイプには自動的に設定されます。
Client ID	OAuth 認可サーバーに登録されているクライアントアプリの一意の識別子を入力します。
Client secret	OAuth 認可サーバーに登録されているクライアントアプリの秘密鍵を入力します。
Authorization endpoint	認可要求を行うための OAuth 認可サーバーの URL を入力します。
Authorization parameters	認可エンドポイントのパラメーターを入力します。
Authorization code	認可コードを入力します: **Automatic** - コードはリダイレクトページを通じて自動的に取得されます; **Manual** - 自動取得に失敗した場合は、コードを手動で入力します。
Token endpoint	認可コードをアクセストークンおよびリフレッシュトークンと交換するための OAuth 認可サーバーの URL を入力します。
Token parameters	アクセストークンのパラメーターを入力します。

Note:
OAuth トークンの取得には、**Gmail**、**Gmail relay**、**Office365** のメールプロバイダー向けに自動化機能 があります。必要なのは *Redirection endpoint*、*Client ID*、*Client secret* の各パラメーター値のみです。Zabbix が他の必要な値を自動的に入力します（プロバイダー別の OAuth URL のデフォルト値 を参照してください）。

OAuth アクセストークンとリフレッシュトークンの取得

OAuth パラメータを送信した後:

1. ブラウザのポップアップが開き、ユーザーを認可エンドポイントに誘導します。
2. ユーザーは OAuth サービスで Zabbix を認可します。

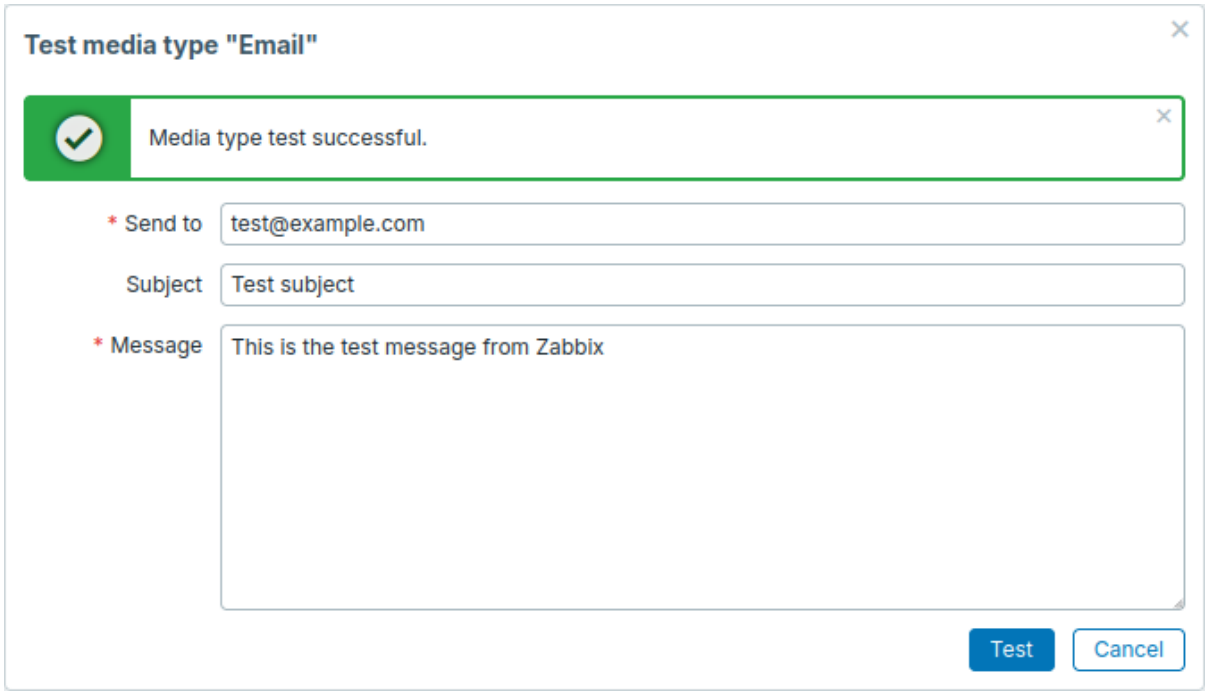
3. OAuth サービスは、認可コードとスコープ値を持って、ユーザーを Zabbix フロントエンドアクション `oauth.authorize` にリダイレクトします。
4. その応答として、Zabbix はトークンエンドポイントにリクエストを行い、認可コードをアクセストークンとリフレッシュトークンに交換します。

テスト

設定したメールメディアタイプが正しく動作するかをテストするには、次の手順を実行します。

1. 一覧 で該当するメールを見つけます。
2. 一覧の最後の列にある *Test* をクリックします（テスト用ウィンドウが開きます）。
3. *Send to* の宛先アドレス、メッセージ本文、必要に応じて件名を入力します。
4. *Test* をクリックしてテストメッセージを送信します。

テストの成功または失敗メッセージは、同じウィンドウに表示されます。



ユーザーメディア

email メディアタイプを設定したら、*Users* > *Users* セクションに移動し、ユーザープロファイルを編集してユーザーに email メディアを割り当てます。ユーザーメディアの設定手順は、すべてのメディアタイプに共通であり、Media types ページに記載されています。

自動化された Gmail/Office365 メディアタイプ

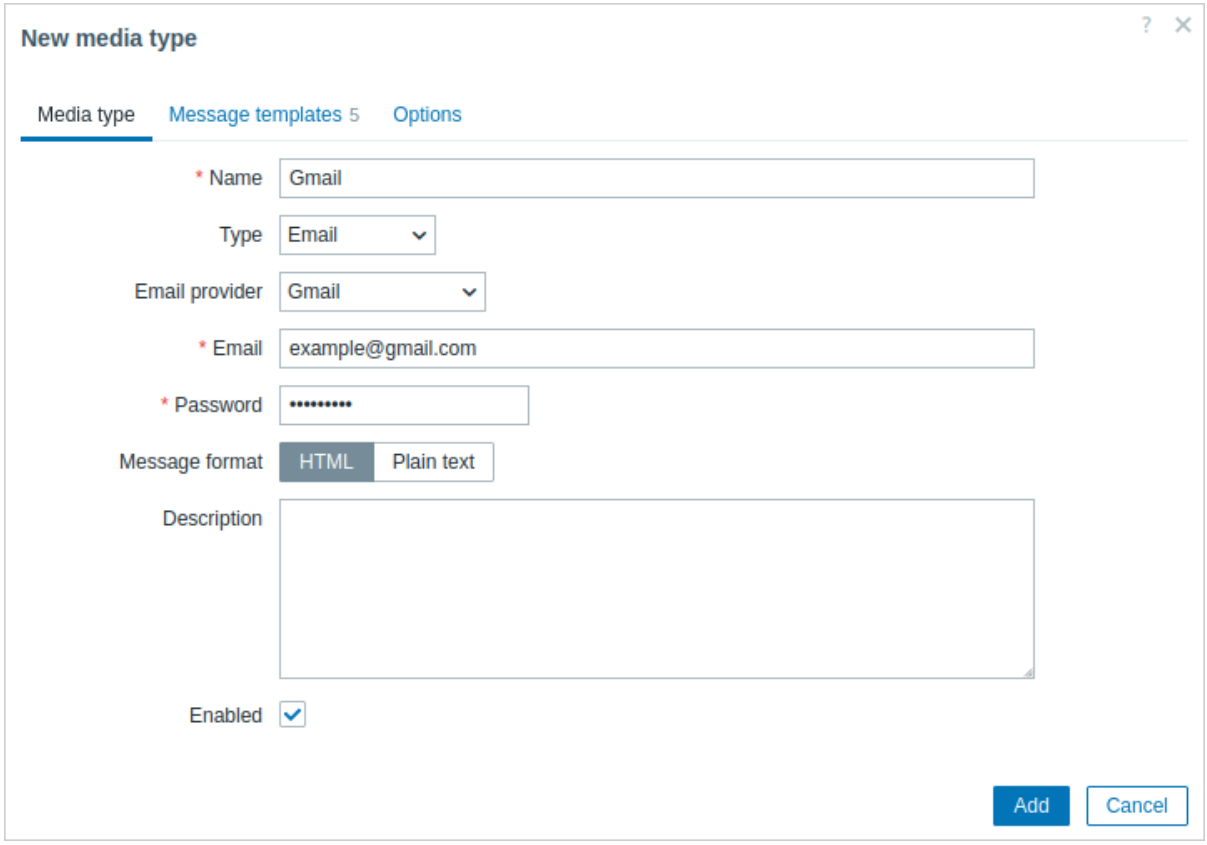
概要

Gmail や Office365 のユーザーは、メディアタイプの設定における自動化機能の恩恵を受けることができます。

設定

メールメディアタイプの設定にある *Email provider* フィールドでは、Gmail および Office 365 用に事前設定されたオプションを選択できます。

Gmail/Office365 関連のオプションを選択する場合、動作するメディアタイプを作成するために、送信者のメールアドレスとパスワードを入力するだけで済みます。



メールアドレスとパスワードを入力すると、Zabbix は Gmail/Office365 のメディアタイプに必要なすべての設定に、実際の値または推奨値を自動的に入力できます。対象となる設定は、*SMTP server*、*SMTP server port*、*SMTP helo*、*Connection security* です。この自動化により、これらのフィールドは表示されません。ただし、メディアタイプの一覧で SMTP サーバーおよびメールの詳細を確認できます（*Details* 列を参照）。

次の点にも注意してください。

- リレーオプションでは、パスワードは必要ありません。
- Office365 リレーでは、指定したメールアドレスのドメイン名が SMTP サーバーの動的な入力に使用されます（つまり、`example-com.mail.protection.outlook.com` の `"example.com"` が実際の値に置き換えられます）。

OAuth トークン

OAuth トークンの取得は、*Gmail*、*Gmail* リレー、*Office365* のメールプロバイダーに対して自動化されています。



OAuth のアクセストークンおよびリフレッシュトークンの取得には、*Redirection endpoint*、*Client ID*、*Client secret* という限られたパラメータのみを指定する必要があります。Zabbix は他の必要な値を自動的に入力します（プロバイダー別の OAuth URL デフォルト値を参照）。

これらのプロバイダーに対して OAuth 認証を設定する場合、Generic SMTPフォームを使用することも可能です。

Note:
SmtpClientAuthentication は、Office365 でユーザー/共有メールボックスごと、またはテナント全体で有効にする必要があります（デフォルト設定では有効になっていません）。

プロバイダーごとの OAuth URL のデフォルト

以下の表は、Zabbix で使用されるプロバイダーごとのデフォルトの OAuth URL 値とパラメーターを示しています。

パラメーター		Gmail	Office365	Generic SMTP
認可エンドポイント		https://accounts.google.com/o/oauth2/auth	https://login.microsoftonline.com/common/oauth2/v2.0/authorize	デフォルトなし
	scope	https://mail.google.com	https://outlook.office.com/SMTP.Send offline_access	デフォルトなし
	access_type	offline	未使用	デフォルトなし
	prompt	consent	未使用	デフォルトなし
	redirect_uri	*Redirection endpoint* フィールドの値が自動的にコードで追加されます。*Authorization endpoint* パラメーターには記載されません。		
	state	OAuth トークンで更新されるメディアを識別する一意のハッシュが自動的にコードで追加されます。*Authorization endpoint* パラメーターには記載されません。		
トークンエンドポイント		https://oauth2.googleapis.com/token	https://login.microsoftonline.com/common/oauth2/v2.0/token	デフォルトなし
	grant_type	authorization_code	authorization_code	デフォルトなし
	redirect_uri	*Redirection endpoint* フィールドの値が自動的にコードで追加されます。*Token endpoint* パラメーターには記載されません。		
	state	OAuth トークンで更新されるメディアを識別する一意のハッシュが自動的にコードで追加されます。*Token endpoint* パラメーターには記載されません。		

Office365 の個人アカウントの回避策

Office365 の個人アカウントは、リダイレクト URL のクエリ文字列をサポートしていません。

この問題を回避するために、Apache ウェブサーバーの管理者は、以下のリライトルールを .htaccess ファイルに追加できます。

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

これにより、以下のような簡略化されたリダイレクト URL が使用できるようになります。

```
http://server-name/zabbix/oauth.authorize
```

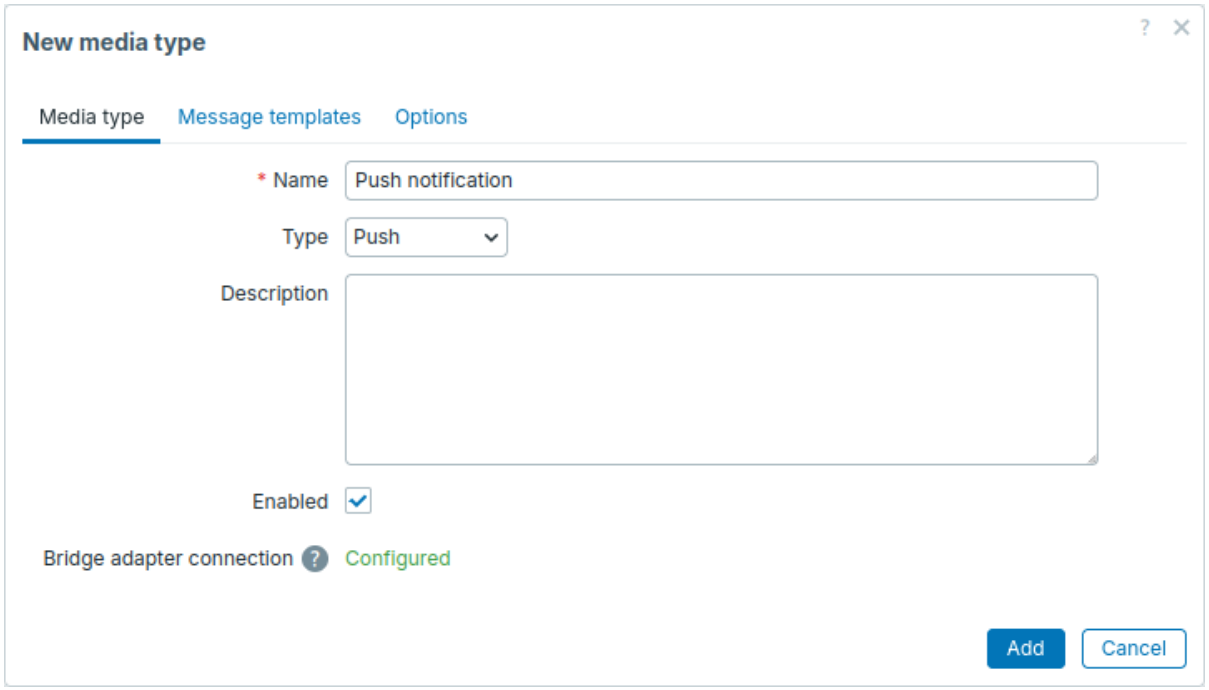
これにより、クエリパラメータ付きの URL の代わりに Office365 の個人アカウントとの互換性が確保されます。

プッシュ通知

このメディアタイプを使用すると、Zabbix サーバーからモバイルデバイス上のZabbix モバイルアプリにプッシュ通知を送信できます。

設定

メディアタイプの作成時に、*Type* として *Push* を選択します。



設定フォームには、共通のメディアタイプのパラメータ、メッセージテンプレート、およびオプションが含まれます。

また、Zabbix ブリッジアダプターが設定済みかどうかもフォームに表示されます。
 アダプターが設定されていない場合でもメディアタイプを作成できますが、機能しません。

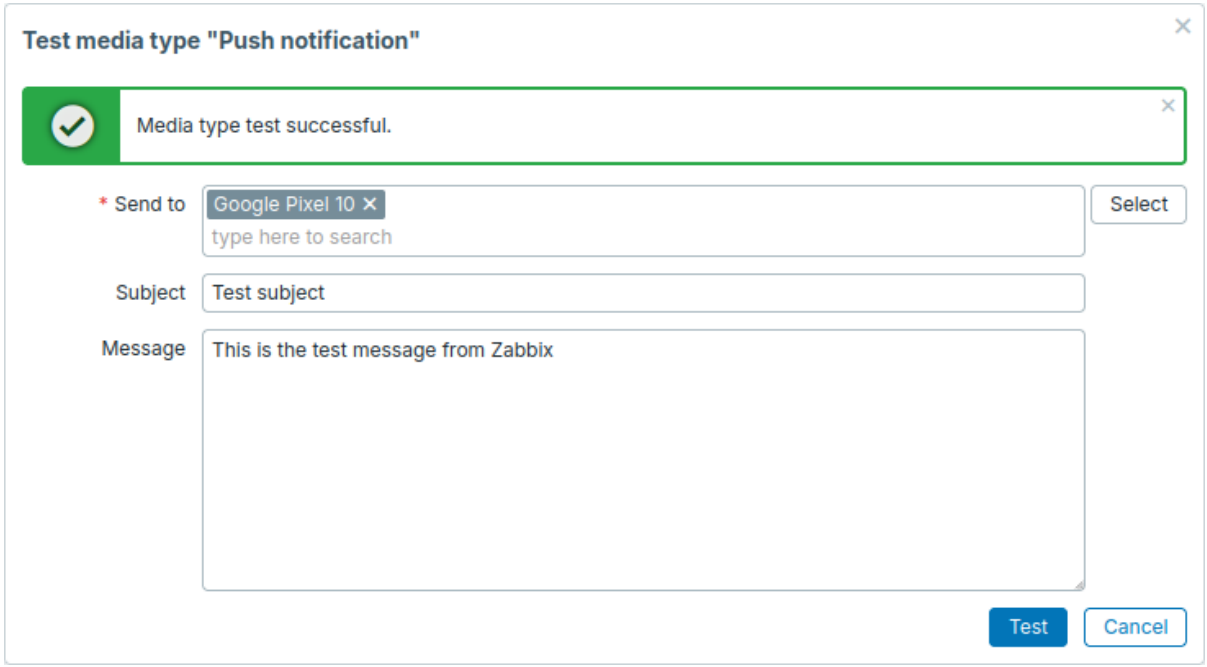
メディアタイプを作成して有効にしたら、ユーザーのデバイスを追加して、そのユーザーにメディアタイプを自動的に追加します(または、ユーザーメディアを手動で追加します)。これにより、ユーザーはモバイルデバイスでプッシュ通知を受信できるようになります。

テスト

このメディアタイプが機能するかどうかをテストするには、次の手順を実行します。

1. メディアタイプの一覧でメディアタイプを見つけます。
2. 一覧の最後の列にあるテストをクリックして、テストフォームを開きます。
3. 送信先フィールドで、デバイスを選択します。
4. テストをクリックして、選択したデバイスにテストプッシュ通知を送信します。

成功または失敗のメッセージが同じウィンドウに表示されます。



SMS

概要

Zabbix は、Zabbix サーバーのシリアルポートに接続されたシリアル GSM モデムを使用した SMS メッセージの送信をサポートしています。

以下を確認してください。

- シリアルデバイスの速度（Linux では通常 /dev/ttyS0）が GSM モデムの速度と一致していること。Zabbix はシリアルリンクの速度を設定しません。デフォルト設定を使用します。
- 'zabbix' ユーザーがシリアルデバイスに対する読み取り/書き込みアクセス権を持っていること。シリアルデバイスの現在の権限を確認するには、コマンド ls -l /dev/ttyS0 を実行してください。
- GSM モデムに PIN が入力されており、電源リセット後も保持されること。あるいは、SIM カードの PIN を無効にすることもできます。PIN は、Unix の minicom や Windows の HyperTerminal などのターミナルソフトウェアで、コマンド AT+CPIN="NNNN"（NNNN は PIN 番号、引用符は必須）を実行して入力できます。

Zabbix は、以下の GSM モデムでテストされています。

- Siemens MC35
- Teltonika ModemCOM/G10

メッセージの配信チャネルとして SMS を設定するには、メディアタイプとして SMS を設定し、ユーザーに対応する電話番号を入力する必要もあります。

設定

SMS をメディアタイプとして設定するには、次の手順を実行します。

1. *Alerts* > *Media types* に移動します。
2. *Create media type* をクリックします（または、事前定義されたメディアタイプの一覧で *SMS* をクリックします）。
3. フォームに SMS メディアタイプのパラメータを入力します。

以下のパラメータは、SMS メディアタイプ固有のものです。

Parameter	Description
GSM modem	GSM モデムのシリアルデバイス名を設定します。 ここで入力したパスは、SMSDevices サーバーパラメータ（指定されている場合）に対して検証されます。

デフォルトメッセージおよびアラート処理オプションの設定方法については、共通のメディアタイプパラメータ を参照してください。SMS 通知の送信を並列処理することはできません。

ユーザーのメディア

SMS メディアタイプを設定したら、ユーザー > ユーザーセクションに移動し、ユーザープロファイルを編集して SMS メディアをユーザーに割り当てます。すべてのメディアタイプに共通するユーザーメディアの設定手順については、メディアタイプ ページで説明しています。

カスタムアラートスクリプト

概要

アラート送信に既存のメディアタイプが適していない場合は、別の方法があります。通知を独自の方法で処理するスクリプトを作成できます。

カスタムアラートスクリプトは Zabbix サーバー上で実行されます。これらのスクリプトは、サーバー設定ファイルのAlertScriptsPathパラメータで指定されたディレクトリに配置する必要があります。

以下は、カスタムアラートスクリプトの例です。

```
#####!/bin/bash

to=$1
subject=$2
body=$3
host=$4
value=$5

cat <<EOF | mail -s "$subject" "$to"
$body

Host: $host
Value: $value
EOF
```

Attention:
Zabbix は、実行されたコマンドおよびスクリプトの終了コードを確認します。**0** 以外の終了コードは、すべてコマンド実行エラーと見なされます。そのような場合、Zabbix は失敗した実行を再試行します。

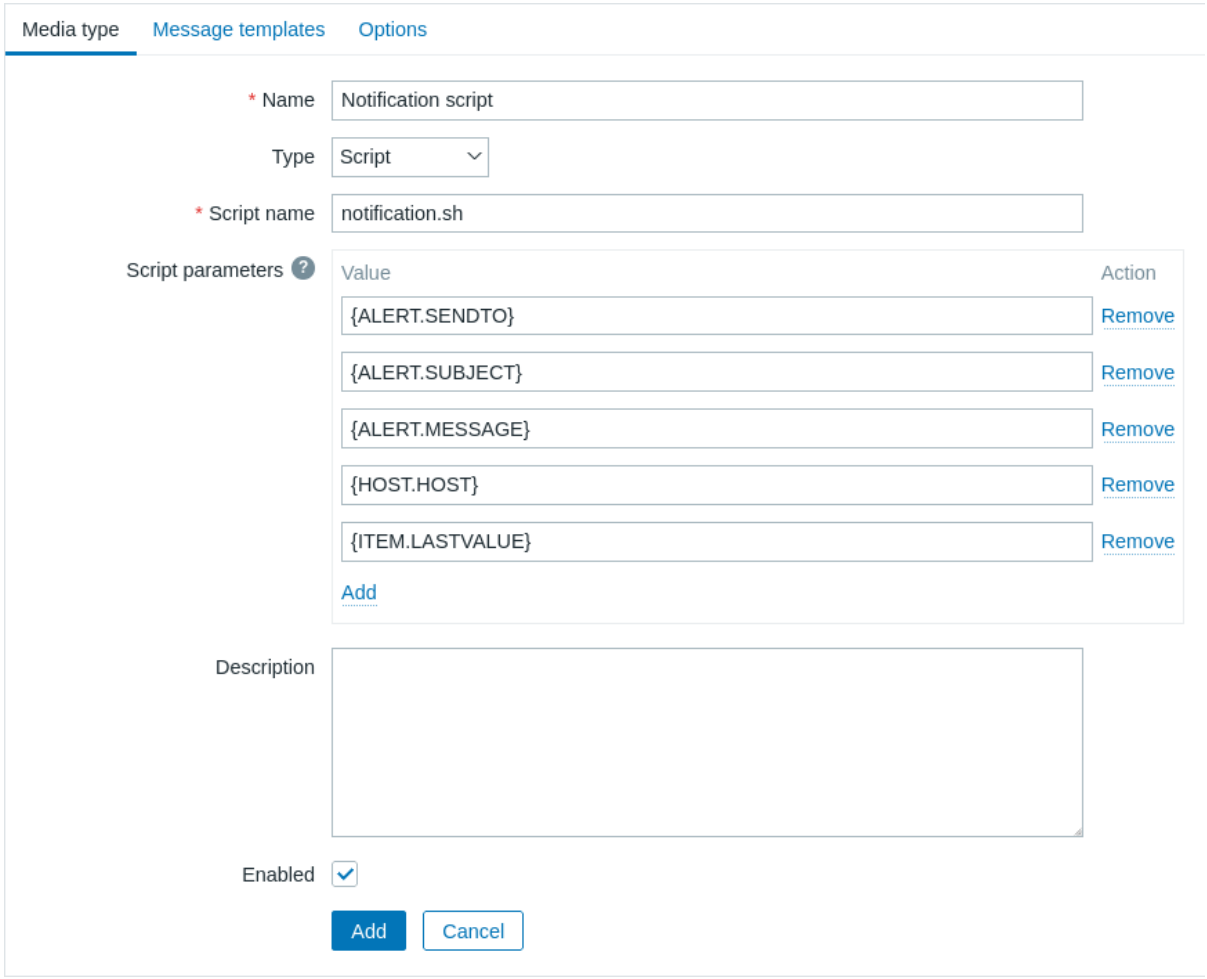
スクリプト用の環境変数は保持も作成もされないため、必要に応じて明示的に扱う必要があります。

設定

カスタムアラートスクリプトをメディアタイプとして設定するには、次の手順を実行します。

1. アラート > メディアタイプに移動します。
2. メディアタイプの作成をクリックします。
3. フォームにスクリプトメディアタイプのパラメータを入力します。

メディアタイプタブには、メディアタイプの一般的な属性が含まれています。



すべての必須入力フィールドには、赤いアスタリスクが付いています。

以下のパラメータは、スクリプトメディアタイプ固有のものです。

パラメータ	説明
スクリプト名	AlertScriptsPath サーバー設定パラメータで指定されたディレクトリにあるスクリプトファイルの名前（例: notification.sh）を入力します。
スクリプトパラメータ	スクリプトにコマンドライン引数として、定義された順序で渡すオプションのスクリプトパラメータを追加します。 スクリプトパラメータでは、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} マクロ、および通知でサポートされるすべてのマクロと、ユーザーマクロを使用できます。

デフォルトメッセージとアラート処理オプションの設定方法については、共通のメディアタイプパラメータを参照してください。

Warning:
アラートスクリプトでデフォルトメッセージを使用しない場合でも、このメディアタイプで使用される実行タイプのメッセージテンプレートを定義する必要があります。定義しない場合、通知は送信されません。

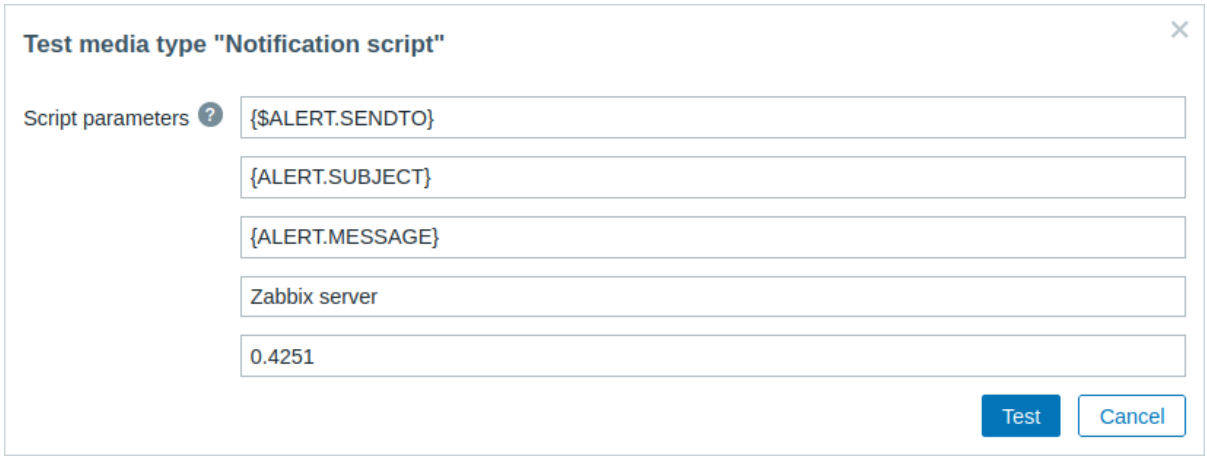
Attention:
複数のスクリプトメディアタイプが設定されている場合、これらのスクリプトはアラートプロセスによって並列処理されることがあります。アラートプロセスの合計数は、サーバー設定ファイルの`StartAlerters` パラメータによって制限されます。

テスト

設定済みのスクリプトメディアタイプをテストするには:

1. メディアタイプの一覧 から、対象のスクリプトを見つけます。
2. 一覧の最後の列にある *Test* をクリックします。テスト用フォームがポップアップウィンドウで開きます。テスト用フォームには、スクリプトメディアタイプに設定されているのと同じ数のパラメータが表示されます。

3. 必要に応じてスクリプトのパラメータ値を編集します。編集内容はテスト手順にのみ影響し、実際の値は変更されません。
4. *Test* をクリックします。



Note:
設定済みのスクリプトメディアタイプをテストする場合、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} およびユーザーマクロはそれぞれの値に展開されます。ただし、イベントに関連するマクロ (例: {HOST.HOST}、{ITEM.LASTVALUE} など) は、テスト時には詳細を取得する関連イベントがないため展開されません。{ALERT.SUBJECT} および {ALERT.MESSAGE} 内のマクロも展開されないことに注意してください。たとえば、{ALERT.SUBJECT} の値が"Problem: {EVENT.NAME}" で構成されている場合、{EVENT.NAME} マクロは展開されません。

ユーザーメディア

メディアタイプを設定したら、*Users* > *Users* セクションに移動し、このメディアタイプをユーザーに割り当ててユーザープロファイルを編集します。ユーザーメディアの設定手順は、すべてのメディアタイプに共通する内容として、Media types ページに記載されています。

ユーザーメディアを定義する際、*Send to* フィールドを空にすることはできません。このフィールドがアラートスクリプトで使用されない場合は、検証要件を回避するために、サポートされている文字の任意の組み合わせを入力してください。

webhook

概要

Webhook メディアタイプは、カスタム JavaScript コードを使用して HTTP コールを行うのに便利で、ヘルプデスクシステム、チャット、メッセンジャーなどの外部ソフトウェアとの簡単な統合が可能です。Zabbix が提供するインテグレーションをインポートするか、独自のインテグレーションをゼロから作成することができます。

インテグレーション

以下のインテグレーションを利用できます。定義済みの webhook メディアタイプを使用して、Zabbix の通知を次のサービスに送信できます。

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE

- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
ここに記載されているサービスに加えて、Zabbix は **Spiceworks** とも統合できます（webhook は不要です）。Zabbix の通知を Spiceworks のチケットに変換するには、メールメディアタイプを作成し、指定した Zabbix ユーザーのプロファイル設定に Spiceworks ヘルプデスクのメールアドレス（例: help@zabbix.on.spiceworks.com）を入力します。

設定

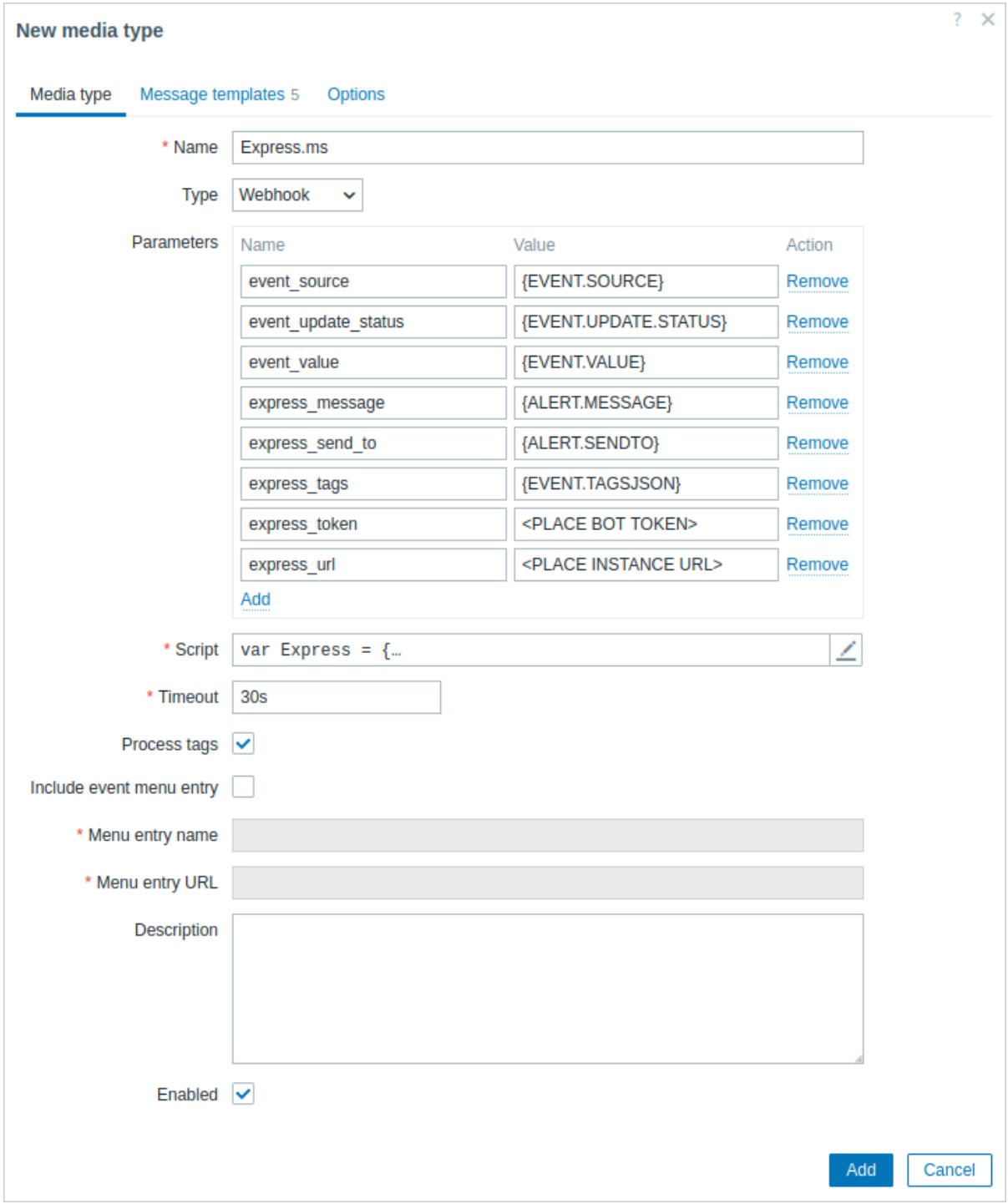
webhook 統合の使用を開始するには、次の手順に従います。

1. ダウンロードした Zabbix バージョンの `templates/media` ディレクトリで必要な `.yaml` ファイルを探すか、Zabbix のgitリポジトリからダウンロードします。
2. ファイルを Zabbix インストール環境にインポートします。
   webhook がメディアタイプの一覧に表示されます。
3. `Readme.md` ファイルの手順に従って webhook を設定します（上記の webhook 名をクリックすると、`Readme.md` にすばやくアクセスできます）。

最初からカスタム webhook を作成するには、次の手順に従います。

1. *Alerts* > *Media types* に移動します。
2. *Create media type* をクリックします。
3. フォームに webhook メディアタイプのパラメータを入力します。

Media type タブには、このメディアタイプ固有のさまざまな属性が含まれています。



必須入力フィールドには赤いアスタリスクが付いています。

webhook メディアタイプ固有のパラメータは次のとおりです。

Parameter	Description
Parameters	webhook 変数を属性と値のペアとして指定します。 事前設定済みの webhook では、パラメータの一覧はサービスによって異なります。パラメータの説明については、webhook の *Readme.md* ファイルを確認してください。 新しい webhook には、いくつかの共通変数がデフォルトで含まれています（URL:<empty>、HTTPProxy:<empty>、To:{ALERT.SENDTO}、Subject:{ALERT.SUBJECT}、Message:{ALERT.MESSAGE}）。これらはそのまま使用することも、削除することもできます。 webhook パラメータでは、ユーザーマクロ、障害通知でサポートされるすべてのマクロ、さらに {ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} マクロを使用できます。 HTTP プロキシを指定する場合、このフィールドではアイテム設定のHTTP プロキシフィールドと同じ機能を使用できます。プロキシ文字列の先頭に `[scheme]://`を付けて、使用するプロキシの種類を指定できます（例: https、socks4、socks5。ドキュメントを参照）。
Script	パラメータフィールド内またはその横の鉛筆アイコンをクリックすると開くモーダルエディターに、JavaScript コードを入力します。このコードによって webhook 操作が実行されます。 スクリプトは、パラメータと値のペアを受け取る関数コードです。値は、JSON.parse() メソッドを使用して JSON オブジェクトに変換する必要があります。例: `var params = JSON.parse(value);`。 コードはすべてのパラメータにアクセスでき、HTTP GET、POST、PUT、DELETE リクエストを実行できます。また、CONNECT、PATCH、HEAD、OPTIONS、TRACE などの追加メソッドをサポートし、HTTP ヘッダーとリクエスト本文を制御できます。 スクリプトには return 演算子を含める必要があります。含まれていない場合、スクリプトは有効になりません。スクリプトは、オプションのタグとタグ値の一覧を伴う OK ステータス（*Process tags* オプションを参照）またはエラー文字列を返すことができます。 復旧イベント（自動的に生成されたもの、手動クローズの結果として生成されたものを問わず）はサーバーによって作成され、解決済みイベントタグ（テンプレート、ホスト、トリガーから継承されたタグを含む）が含まれます。webhook スクリプトはアラートの作成後に実行されるため、webhook スクリプトが返すタグは初回アラートの作成後にのみ追加され、初回の障害メッセージまたは即時復旧メッセージの{EVENT.TAGS}および{EVENT.RECOVERY.TAGS}マクロには含まれません。 注: 各スクリプトが独自のデータを処理し、同時実行される呼び出し間の衝突を避けるため、グローバル変数（例: `global = 1`）ではなくローカル変数（例: `var local = 1`）を使用することを推奨します（既知の問題を参照）。 詳細については、Webhook 開発ガイドライン、Webhook スクリプトの例、追加の JavaScript オブジェクトも参照してください。
Timeout	JavaScript の実行タイムアウト（1～60 秒、デフォルト 30 秒）。 時間のサフィックスがサポートされています（例: 30s、1m）。
Process tags	返された JSON プロパティ値をタグとして処理するには、チェックボックスを選択します。これらのタグは、既存の障害タグに追加されます。 webhook タグを使用する場合、webhook は少なくとも空の tags オブジェクトを含む JSON オブジェクトを返す必要があります: `var result = {tags: {}};` 返すことができるタグの例: *jira-id:prod-1234*、*responsible:John Smith*、*processed:<no value>*
Include event menu entry	作成された外部チケットへのリンクを含む項目をevent menuに追加するには、チェックボックスを選択します。 有効になっていて、このチェックボックスが選択されている各 webhook について項目が追加されます。*Menu entry name* および *Menu entry URL* パラメータに{EVENT.TAGS.<tag name>}マクロが含まれている場合、これらのマクロを解決できる場合（つまり、イベントにこれらのタグが定義されている場合）にのみ項目が追加されることに注意してください。 選択した場合、この webhook を異なるユーザーへの通知送信に使用しないでください（代わりに専用ユーザーの作成を検討してください）。また、単一の障害イベントに対する複数のアラートアクションで使用しないでください。
Menu entry name	メニュー項目の名前を指定します。 {EVENT.TAGS.<tag name>}マクロがサポートされています。 *Include event menu entry* を選択した場合のみ、このフィールドは必須です。
Menu entry URL	メニュー項目のリンク先 URL を指定します。 {EVENT.TAGS.<tag name>}マクロがサポートされています。 *Include event menu entry* を選択した場合のみ、このフィールドは必須です。

デフォルトメッセージとアラート処理オプションの設定方法については、共通メディアタイプパラメータを参照してください。

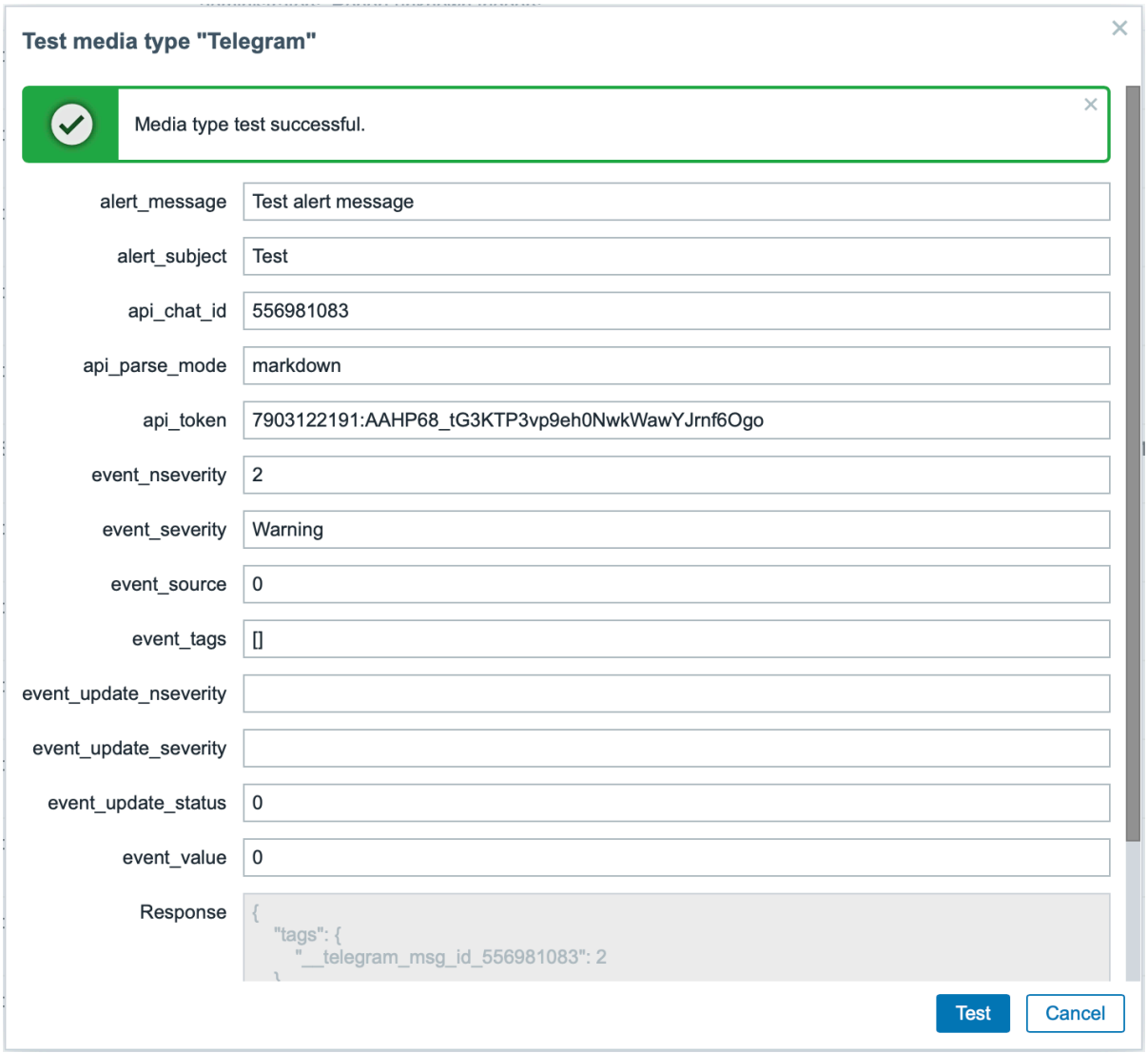
Warning:
webhook がデフォルトメッセージを使用しない場合でも、この webhook で使用される操作タイプのメッセージテンプレートを定義する必要があります。

テスト

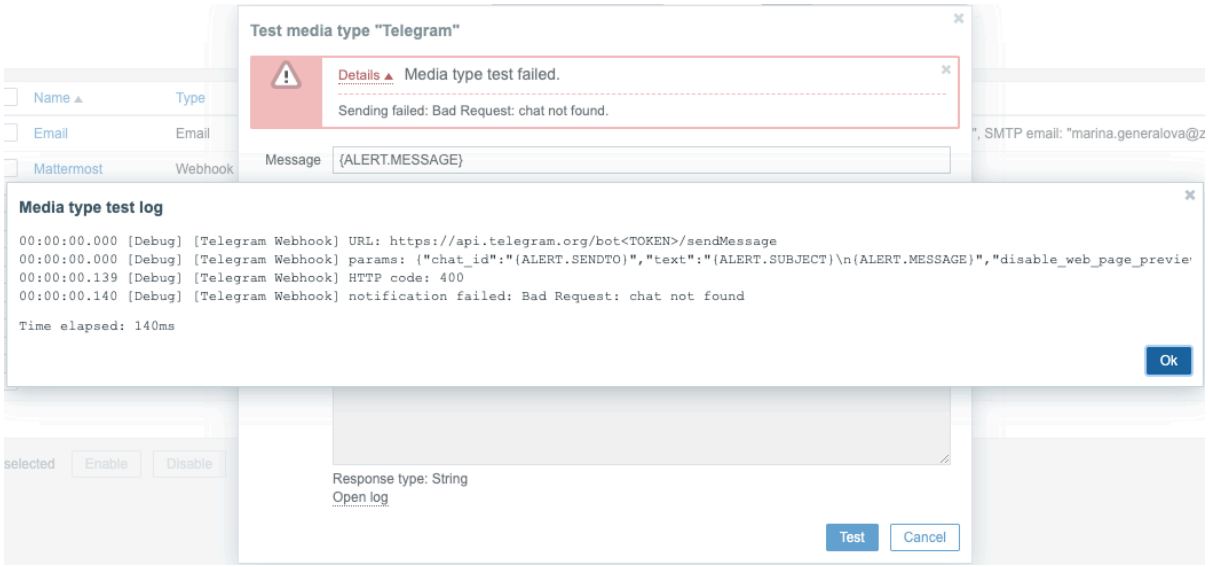
設定済みの webhook メディアタイプをテストするには、次の手順に従います。

1. メディアタイプの一覧で、該当する webhook を見つけます。
2. 一覧の最後の列にあるテストをクリックします（テストウィンドウが開きます）。
3. 必要に応じて、webhook パラメータの値を編集します。
   マクロをサンプル値に置き換えてください。置き換えない場合、マクロは解決されず、テストは失敗します。
4. テストをクリックします。

テストウィンドウで値を置き換えたり削除したりしても、テスト手順にのみ影響します。実際の webhook 属性の値は変更されません。



テストウィンドウを閉じずにメディアタイプのテストログエントリを表示するには、ログを開くをクリックします（新しいポップアップウィンドウが開きます）。



Webhook テストが成功した場合：

- *"Media type test successful."* メッセージが表示されます。
- サーバーのレスポンスがグレーの *Response* フィールドに表示されます。
- レスポンスタイプ（JSON または String）が *Response* フィールドの下に表示されます。

Webhook テストが失敗した場合**:**

- " メディアタイプのテストに失敗しました。" というメッセージが表示され、続いて追加の失敗の詳細が表示されます。

ユーザーメディア

メディアタイプを設定したら、ユーザー > ユーザーセクションに移動し、webhook メディアを既存のユーザーに割り当てるか、webhook を表す新しいユーザーを作成します。既存のユーザーにユーザーメディアを設定する手順は、すべてのメディアタイプに共通で、メディアタイプページに記載されています。

webhook がチケット\メッセージ ID の保存にタグを使用する場合、同じ webhook を異なるユーザーのメディアとして割り当てないでください。これにより webhook エラーが発生する可能性があります（イベントメニューエントリを含めるオプションを使用する大半の webhook に該当します）。この場合、webhook を表す専用ユーザーを作成することをお勧めします。

1. webhook メディアタイプを設定したら、ユーザー > ユーザーセクションに移動し、webhook を表す専用の Zabbix ユーザーを作成します。たとえば、Slack webhook の場合は、ユーザー名を *Slack* にします。このユーザーは Zabbix にログインしないため、メディア以外のすべての設定はデフォルトのままにできます。
2. ユーザープロファイルで、メディアタブに移動し、必要な連絡先情報を指定してwebhook を追加します。webhook が送信先フィールドを使用しない場合は、検証要件を満たすため、サポートされている文字を任意に組み合わせて入力します。
3. アラートを送信する必要があるすべてのホストに対して、このユーザーに少なくとも読み取り権限を付与します。

アラートアクションを設定する際は、操作の詳細のユーザーに送信フィールドにこのユーザーを追加します。これにより、Zabbix はこのアクションからの通知に webhook を使用します。

アラートアクションの設定

アクションは、webhook を介してどの通知を送信するかを決定します。webhook を含むアクションの設定 の手順は、次の例外を除き、他のすべてのメディアタイプと同じです。

- webhook がチケット\メッセージ ID を保存し、更新\解決の操作を処理するために webhook タグ を使用する場合、1 つの障害イベントに対して複数のアラートアクションで同じ webhook を使用しないでください。{EVENT.TAGS.<tag name>} が存在し、webhook 内で更新される場合、その結果の値は未定義になります。これを回避するには、更新された値を保存するために webhook 内で新しいタグ名を使用してください。これは、Zabbix が提供する Jira、Jira Service Desk、Mattermost、Opsgenie、OTRS、Redmine、ServiceNow、Slack、Zammad、Zendesk の webhook、および *Include event menu entry* オプションを利用するほとんどの webhook に適用されます。ただし、同じ webhook は、同一のアクション内の複数の操作またはエスカレーションステップで使用できます。また、異なる条件 により同じ障害イベントではトリガーされない別のアクションでも使用できます。
- 内部イベント 用のアクションで webhook を使用する場合は、必ず *Custom message* チェックボックスをオンにし、アクションの操作設定でカスタムメッセージを定義してください。そうしないと、通知は送信されません。

webhook スクリプトの例

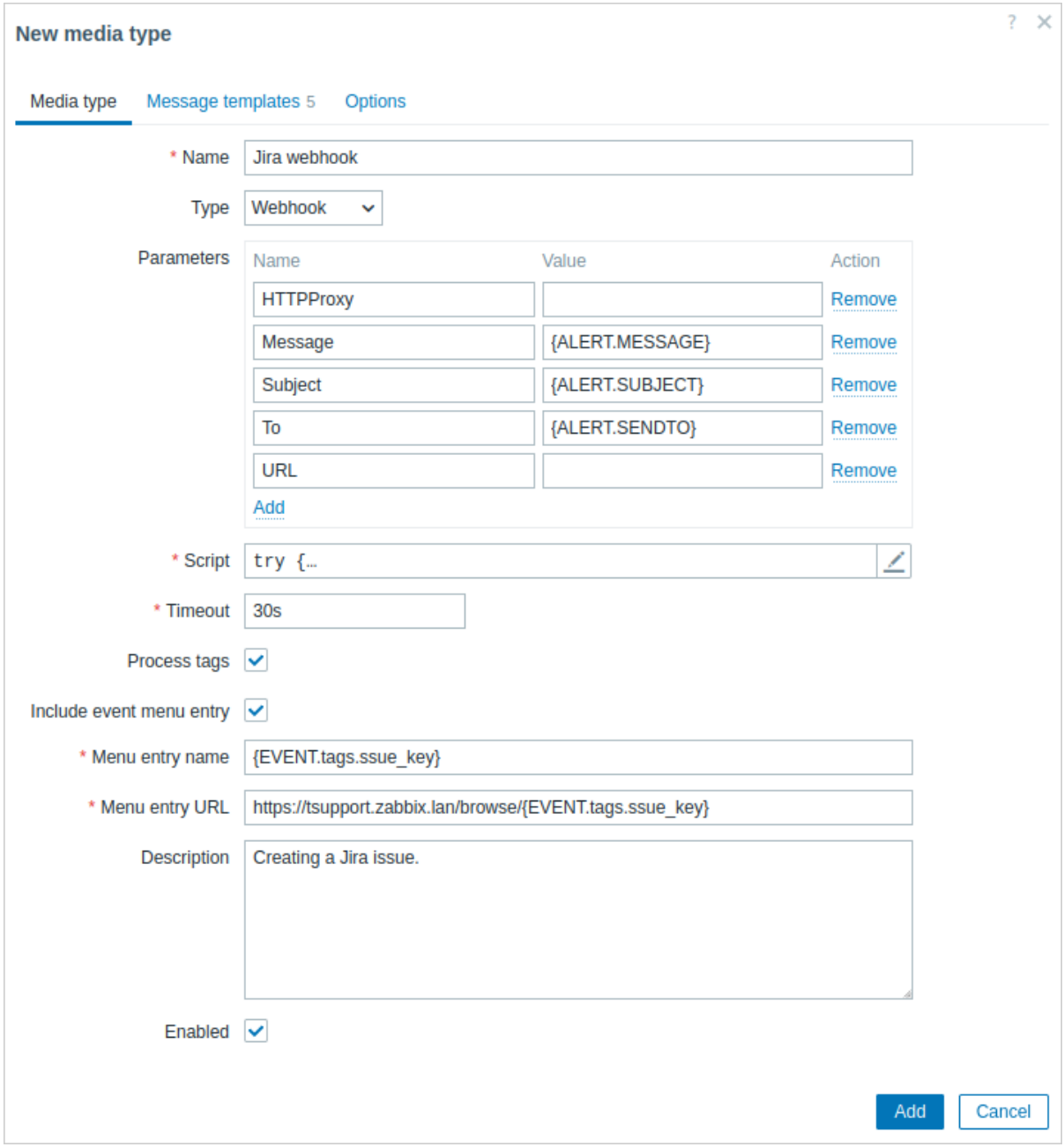
概要

Zabbix には標準で利用できる多数の webhook 連携がありますが、独自の webhook を作成したい場合もあります。このセクションでは、カスタム webhook スクリプトの例を示します（*Script* パラメータで使用します）。他の webhook パラメータの説明については、webhook セクションを参照してください。

Attention:
前処理 JavaScript では、宣言されていない代入を使用しないでください。ローカル変数の宣言には `var` を使用してください。

復旧イベント（自動生成された場合、または手動クローズ の後に生成された場合のいずれも）は、解決済みのイベントタグを含みます（テンプレート、ホスト、およびトリガーから継承されたタグを含みます）。webhook スクリプトはアラート作成後に実行されるため、webhook スクリプトが返すタグは初回のアラート作成後に適用され、webhook をトリガーしたアラートには含まれない場合があります。連携で webhook によって生成されたタグを初期通知に含める必要がある場合は、サーバーからイベントタグを取得するか（たとえばEvent API を使用）、または webhook で生成されたタグを外部の永続ストアに保存してそこで関連付けてください。

Jira webhook（カスタム）



このスクリプトは JIRA の課題を作成し、作成された課題に関する情報を返します。

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
                'endpoint': 'jira'
```

```
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

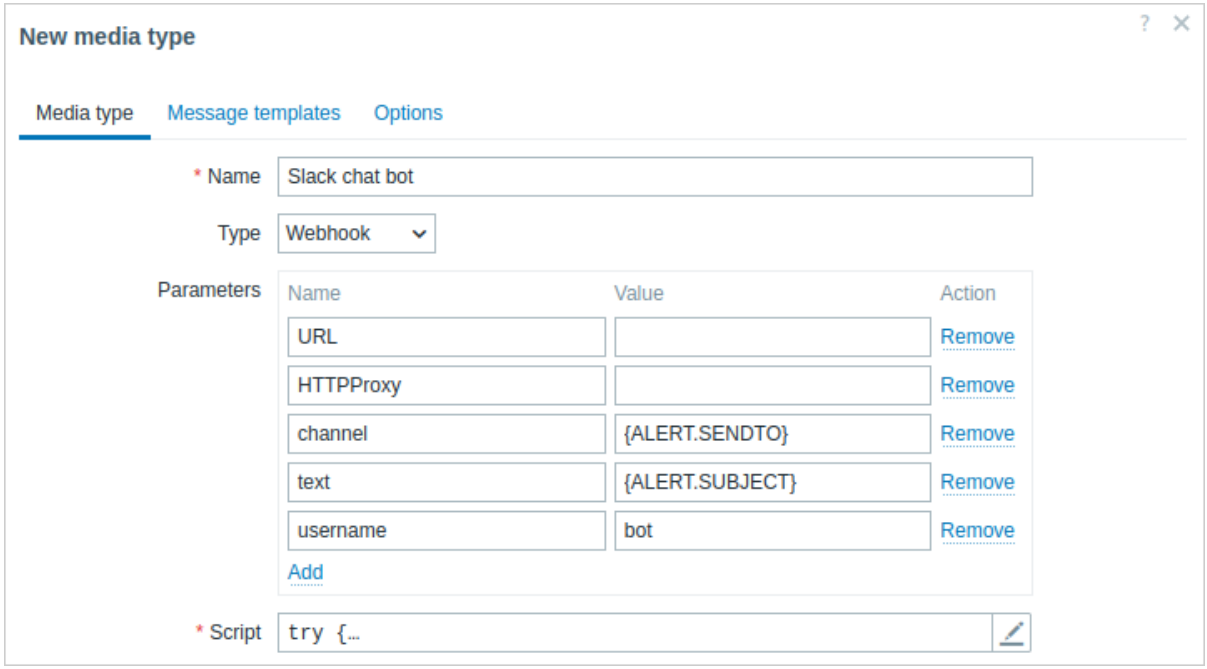
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Slack webhook (カスタム)

この webhook は、Zabbix から Slack チャンネルへ通知を転送します。




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);

    response = req.post(params.URL, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

アクション

概要

イベントの結果として何らかの操作（たとえば通知の送信など）を行いたい場合は、アクションを設定する必要があります。

アクションは、サポートされているすべてのタイプのイベントに対して定義できます。

- トリガーアクション - トリガーの状態が OK と障害の間で変化したときのイベント
- サービスアクション - サービスの状態が OK と障害の間で変化したときのイベント
- ディスカバリアクション - ネットワークディスカバリが行われたときのイベント
- 自動登録アクション - 新しいアクティブエージェントが自動登録されたとき（または登録済みのホストのメタデータが変更されたとき）のイベント
- 内部アクション - アイテムがサポート対象外になったり、トリガーが不明な状態になったときのイベント

サービスアクションの主な違いは次のとおりです。

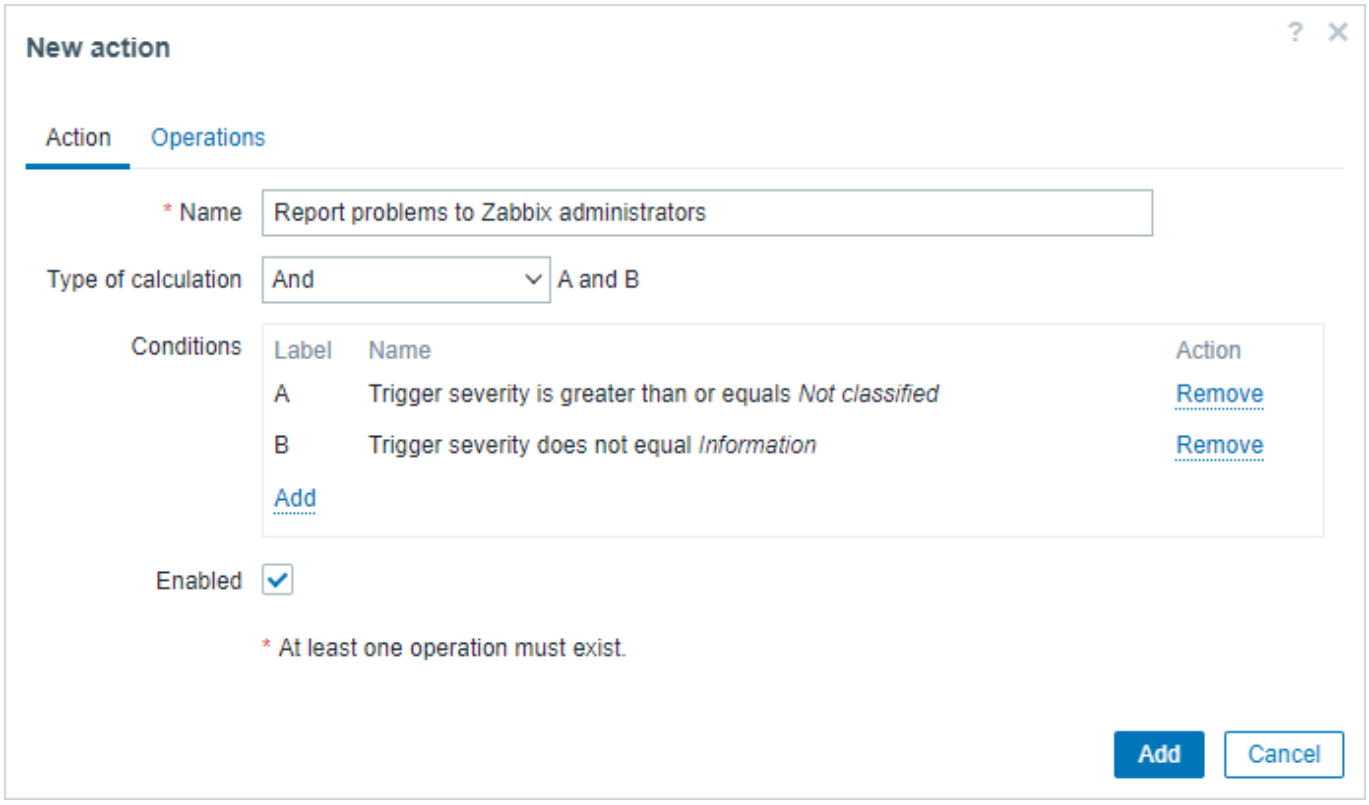
- サービスアクションへのユーザーアクセスは、ユーザーのロールによって付与されたサービスへのアクセス権に依存します
- サービスアクションは、異なる条件セットをサポートしています

アクションの設定

アクションを設定するには、次の手順を実行します。

1. アラート > アクションに移動し、サブメニューから必要なアクションタイプを選択します（後でタイトルのドロップダウンを使用して、別のタイプに切り替えることもできます）。
2. アクションの作成をクリックします。
3. アクション名を指定します。
4. 実行する操作の条件 を選択します。
5. 実行する操作 を選択します。

一般的なアクション属性:



必須の入力フィールドには赤いアスタリスクが付いています。

Parameter	Description
Name	一意のアクション名。

Parameter	Description
Type of calculation	アクション条件の評価オプション を選択します（条件が複数ある場合）。 **And** - すべての条件を満たす必要があります。 **Or** - いずれか 1 つの条件を満たせば十分です。 **And/Or** - 2 つの組み合わせです。異なる条件タイプでは AND、同じ条件タイプでは OR になります。 **Custom expression** - アクション条件を評価するための、ユーザー定義の計算式です。
Conditions	アクション条件の一覧です。 *Add* をクリックして新しい条件 を追加します。 条件が設定されていない場合、アクションは設定中のアクションタイプ に対応するすべてのイベントで実行されます。
Enabled	チェックボックスをオンにするとアクションが有効になります。オフの場合は無効になります。

条件

概要

イベントが定義された条件のセットに一致する場合にのみ、アクションが実行されるように定義できます。条件は、アクションの設定時に指定します。

条件の一致では大文字と小文字が区別されます。

トリガーアクション

トリガーベースのアクションでは、以下の条件を使用できます。

条件タイプ	サポートされる演算子	説明
ホストグループ	equals does not equal	ホストグループまたは除外するホストグループを指定します。 **equals** - イベントがこのホストグループに属します。 **does not equal** - イベントがこのホストグループに属しません。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。親グループのみを指定するには、すべてのネストされたグループを **does not equal** 演算子で追加指定する必要があります。
テンプレート	equals does not equal	テンプレートまたは除外するテンプレートを指定します。 **equals** - イベントがこのテンプレートから継承されたトリガーに属します。 **does not equal** - イベントがこのテンプレートから継承されたトリガーに属しません。
ホスト	equals does not equal	ホストまたは除外するホストを指定します。 **equals** - イベントがこのホストに属します。 **does not equal** - イベントがこのホストに属しません。
タグ名	equals does not equal contains does not contain	イベントタグまたは除外するイベントタグを指定します。 **equals** - イベントがこのタグを持ちます。 **does not equal** - イベントがこのタグを持ちません。 **contains** - イベントがこの文字列を含むタグを持ちます。 **does not contain** - イベントがこの文字列を含むタグを持ちません。
タグ値	equals does not equal contains does not contain	イベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。 **equals** - イベントがこのタグと値を持ちます。 **does not equal** - イベントがこのタグと値を持ちません。 **contains** - イベントがこの文字列を含むタグと値を持ちます。 **does not contain** - イベントがこの文字列を含むタグと値を持ちません。
トリガー	equals does not equal	トリガーまたは除外するトリガーを指定します。 **equals** - イベントがこのトリガーによって生成されます。 **does not equal** - イベントがこのトリガー以外のトリガーによって生成されます。

条件タイプ	サポートされる演算子	説明
イベント名	contains does not contain	トリガーによって生成されたイベント名に含まれる文字列、または除外する文字列を指定します。デフォルトでは、イベント名はトリガー名と一致しますが、トリガー設定でカスタムイベント名が指定されている場合はその限りではありません。 **contains** - イベント名がこの文字列を含みます。 **does not contain** - この文字列がイベント名から除外されます。 注: 入力された値は、すべてのマクロが展開されたイベント名と比較されます。
トリガーの深刻度	equals does not equal is greater than or equals is less than or equals	トリガーの深刻度を指定します。 **equals** - トリガーの深刻度と等しい。 **does not equal** - トリガーの深刻度と等しくない。 **is greater than or equals** - トリガーの深刻度以上。 **is less than or equals** - トリガーの深刻度以下。
時間帯	in not in	時間帯または除外する時間帯を指定します。 **in** - イベントの発生時刻が時間帯内。 **not in** - イベントの発生時刻が時間帯外。 書式の説明については時間帯の指定ページを参照してください。 ユーザーマクロがサポートされています。
問題が抑制されている	no yes	ホストのメンテナンスにより問題が抑制（非表示）されているかどうかを指定します。 **no** - 問題は抑制されていません。 **yes** - 問題は抑制されています。

サービスアクション

サービスアクションでは、次の条件を使用できます。

Condition type	Supported operators	Description
Service	equals does not equal	サービス、または除外するサービスを指定します。 **equals** - イベントはこのサービスに属します。 **does not equal** - イベントはこのサービスに属しません。 親サービスを指定すると、すべての子サービスが暗黙的に選択されます。親サービスのみを指定するには、ネストされたすべてのサービスを追加で **does not equal** 演算子で設定する必要があります。
Service name	contains does not contain	サービス名に含める文字列、または除外する文字列を指定します。 **contains** - イベントは、名前にこの文字列を含むサービスによって生成されます。 **does not contain** - この文字列はサービス名に含まれていません。
Service tag name	equals does not equal contains does not contain	イベントタグ、または除外するイベントタグを指定します。サービスイベントタグは、サービス設定セクションの *Tags* で定義できます。 **equals** - イベントにはこのタグがあります。 **does not equal** - イベントにはこのタグがありません。 **contains** - イベントには、この文字列を含むタグがあります。 **does not contain** - イベントには、この文字列を含むタグがありません。
Service tag value	equals does not equal contains does not contain	イベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。サービスイベントタグは、サービス設定セクションの *Tags* で定義できます。 **equals** - イベントにはこのタグと値があります。 **does not equal** - イベントにはこのタグと値がありません。 **contains** - イベントには、これらの文字列を含むタグと値があります。 **does not contain** - イベントには、これらの文字列を含むタグと値がありません。

Attention:
Alerts > *Media types* メニューで、Service actions 用のmessage templates を必ず定義してください。定義しない場合、通知は送信されません。

ディスカバリアクション

以下の条件をディスカバリーベースのイベントで使用できます。

条件タイプ	サポートされる演算子	説明
ホスト *IP*	等しい 等しくない	検出されたホストの IP アドレス範囲、または除外する範囲を指定します。 等しい - ホスト IP が範囲内にある。 等しくない - ホスト IP が範囲内にない。 以下の形式を使用できます。 単一の IP: 192.168.1.33 IP アドレスの範囲: 192.168.1-10.1-254 IP マスク: 192.168.4.0/24 リスト: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 リスト形式ではスペースも使用できます。
サービスの種類	等しい 等しくない	検出されたサービスのサービスの種類、または除外するサービスの種類を指定します。 等しい - 検出されたサービスと一致する。 等しくない - 検出されたサービスと一致しない。 使用可能なサービスの種類: SSH、LDAP、SMTP、FTP、HTTP、HTTPS、POP、NNTP、IMAP、TCP、Zabbix エージェント、SNMPv1 エージェント、SNMPv2 エージェント、SNMPv3 エージェント、ICMP ping、telnet。
サービスのポート	等しい 等しくない	検出されたサービスの TCP ポート範囲、または除外する範囲を指定します。 等しい - サービスのポートが範囲内にある。 等しくない - サービスのポートが範囲内にない。
ディスカバリールール	等しい 等しくない	ディスカバリールール、または除外するディスカバリールールを指定します。 等しい - このディスカバリールールを使用している。 等しくない - このディスカバリールール以外のディスカバリールールを使用している。
ディスカリーチェック	等しい 等しくない	ディスカバリーチェック、または除外するディスカバリーチェックを指定します。 等しい - このディスカバリーチェックを使用している。 等しくない - このディスカバリーチェック以外のディスカバリーチェックを使用している。
ディスカリーオブジェクト	等しい	検出されたオブジェクトを指定します。 等しい - 検出されたオブジェクト（デバイスまたはサービス）と等しい。
ディスカバリーステータス	等しい	**Up** - 「ホストが稼働中」および「サービスが稼働中」イベントに一致します。 **Down** - 「ホストが停止」および「サービスが停止」イベントに一致します。 **Discovered** - 「ホストを検出」および「サービスを検出」イベントに一致します。 **Lost** - 「ホストを消失」および「サービスを消失」イベントに一致します。
稼働時間/停止時間	以上 以下	「ホストが稼働中」および「サービスが稼働中」イベントでは稼働時間。「ホストが停止」および「サービスが停止」イベントでは停止時間です。 以上 - 指定値以上。パラメータは秒単位で指定します。 以下 - 指定値以下。パラメータは秒単位で指定します。
受信値	等しい 等しくない 以上 以下 含む 含まない	ディスカバリールール内のエージェント（Zabbix、SNMP）チェックから受信した値を指定します。文字列比較です。ルールに複数の Zabbix エージェントまたは SNMP チェックが設定されている場合、それぞれの受信値がチェックされます（各チェックによって新しいイベントが生成され、そのイベントがすべての条件と照合されます）。 等しい - 値と等しい。 等しくない - 値と等しくない。 以上 - 値以上。 以下 - 値以下。 含む - 部分文字列を含む。パラメータは文字列として指定します。 含まない - 部分文字列を含まない。パラメータは文字列として指定します。

条件タイプ	サポートされる演算子	説明
プロキシ	等しい 等しくない	プロキシ、または除外するプロキシを指定します。 等しい - このプロキシを使用している。 等しくない - このプロキシ以外のプロキシを使用している。
プロキシグループ	等しい 等しくない	プロキシグループ、または除外するプロキシグループを指定します。 等しい - このプロキシグループを使用している。 等しくない - このプロキシグループ以外のプロキシグループを使用している。

Note:
ディスカバリーイベントを生成するディスカバリールール内のサービスチェックは、同時には実行されません。そのため、アクションの `Service type`、`Service port`、または `Received value` 条件に複数の値が設定されている場合、それらは一度に 1 つのディスカバリーイベントと比較され、複数のイベントと同時に比較されることはありません。その結果、同じチェックタイプに複数の値を指定したアクションが正しく実行されない場合があります。

自動登録アクション

アクティブエージェントの自動登録に基づくアクションでは、次の条件を使用できます。

条件タイプ	サポートされている演算子	説明
ホストメタデータ	contains does not contain matches does not match	除外するホストメタデータまたはホストメタデータを指定します。 **contains** - ホストメタデータに文字列が含まれている場合。 **does not contain** - ホストメタデータに文字列が含まれていない場合。 ホストメタデータは、エージェント設定ファイルで指定できます。 **matches** - ホストメタデータが正規表現に一致する場合。 **does not match** - ホストメタデータが正規表現に一致しない場合。
ホスト名	contains does not contain matches does not match	除外するホスト名またはホスト名を指定します。 **contains** - ホスト名に文字列が含まれている場合。 **does not contain** - ホスト名に文字列が含まれていない場合。 **matches** - ホスト名が正規表現に一致する場合。 **does not match** - ホスト名が正規表現に一致しない場合。
プロキシ	equals does not equal	除外するプロキシまたはプロキシを指定します。 **equals** - このプロキシを使用している場合。 **does not equal** - このプロキシ以外のプロキシ（またはサーバー）を使用している場合。
プロキシグループ	equals does not equal	除外するプロキシグループまたはプロキシグループを指定します。 **equals** - このプロキシグループを使用している場合。 **does not equal** - このプロキシグループ以外のプロキシグループを使用している場合。

内部イベントアクション

内部イベントに基づくアクションには、次の条件を設定できます。

タイプ	オペレータ	説明
イベントタイプ	等しい	アイテムが”取得不可”の状態 - アイテムが’通常’から’サポートされていない’に移行するイベントに一致します ローレベルディスカバリが”取得不可”の状態 - ローレベルディスカバリルールが’通常’から’サポートされていない’に移行するイベントに一致します トリガーが”不明”の状態 - トリガーが’通常’から’不明’に移行するイベントに一致します
ホストグループ	等しい 等しくない	ホストグループまたは除外するホストグループを指定。 等しい - イベントはこのホストグループに属します。 等しくない - イベントはこのホストグループに属しません。

タイプ	オペレータ	説明
タグ名	等しい 等しくない 含む 含まない	イベントタグまたは除外するイベントタグを指定。 等しい - イベントにこのタグがある 等しくない - イベントにこのタグが無い 含む - イベントにこの文字列を含むタグがある * * 含まない ** - イベントにこの文字列を含むタグがない
タグの値	等しい 等しくない 含む 含まない	タグと値の組み合わせ、または除外するイベントタグと値の組み合わせを指定します。 等しい - イベントにこのタグと値がある 等しくない - イベントにこのタグと値がない 含む - イベントにこれらの文字列を含むタグと値がある 含まない - イベントにこれらの文字列を含むタグと値がない
テンプレート	等しい 等しくない	テンプレートまたは除外するテンプレートを指定します。 等しい - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスカバリルールに属する 等しくない - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスカバリルールに属さない
ホスト	等しい 等しくない	ホストまたは除外するホストを指定します。 等しい - イベントはこのホストに属する 等しくない - イベントはこのホストに属さない

計算タイプ

条件の計算には、以下のオプションがあります。

- **And** - すべての条件を満たす必要があります

複数のトリガーが `Trigger=` 条件として選択されている場合、それらの間で”And” 計算を使用することはできません。アクションは、1 つのトリガーのイベントに基づいてのみ実行できます。

- **Or** - 1 つの条件を満たせば十分です
- **And/Or** - 2 つの組み合わせです。異なる条件タイプでは AND、同じ条件タイプでは OR になります。例えば:

ホストグループが Oracle servers と等しい
 ホストグループが MySQL servers と等しい
 イベント名に’Database is down’ が含まれる
 イベント名に’Database is unavailable’ が含まれる

これは次のように評価されます。

(ホストグループが Oracle servers と等しい **or** ホストグループが MySQL servers と等しい**) and (**イベント名に’Database is down’ が含まれる **or** イベント名に’Database is unavailable’ が含まれる**)**

- **Custom expression** - アクション条件を評価するためのユーザー定義の計算式です。すべての条件（大文字の A、B、C、... で表されます）を含める必要があり、スペース、タブ、括弧 ( )、**and**（大文字小文字を区別）、**or**（大文字小文字を区別）、**not**（大文字小文字を区別）を含めることができます。

前述の `And/Or` の例は (A or B) and (C or D) と表されますが、カスタム式では他にもさまざまな計算方法を使用できます。

(A and B) and (C or D)
 (A and B) or (C and D)
 ((A or B) and C) or D
 (not (A or B) and C) or not D
 など。

アクションで使用されているオブジェクトの削除

アクションの条件/操作で使用されているオブジェクトを削除しようとすると、ユーザーには対応するエラーメッセージが表示されます。

アクションの条件または操作が以下の場合、アクションは無効になりません。

- 削除されたオブジェクトがある場合
- 低レベルディスカバリで検出されなくなり、(失われたリソースの削除期間後に) 削除されたオブジェクトがある場合

アクションの条件/操作に欠落しているオブジェクトがある場合、それらは削除されません。代わりに、削除されたオブジェクトは削除されたホストグループ、削除されたホスト、削除されたトリガーなどとして表示されます。アクションリストの「情報」列には、「このアクションには削除されたオブジェクトを参照する条件または操作があります。」というメッセージ付きの警告アイコンが表示されます。

操作

概要

すべてのイベントに対して、以下の実行内容を定義できます。

- メッセージを送信
- リモートコマンドを実行

Attention:
アクションの実行内容の送信先として定義されたユーザーに対して、そのホストへのアクセスが明示的に「拒否」されている場合、またはそのユーザーにそのホストに対する権限がまったく設定されていない場合、Zabbix サーバーはアラートを生成しません。

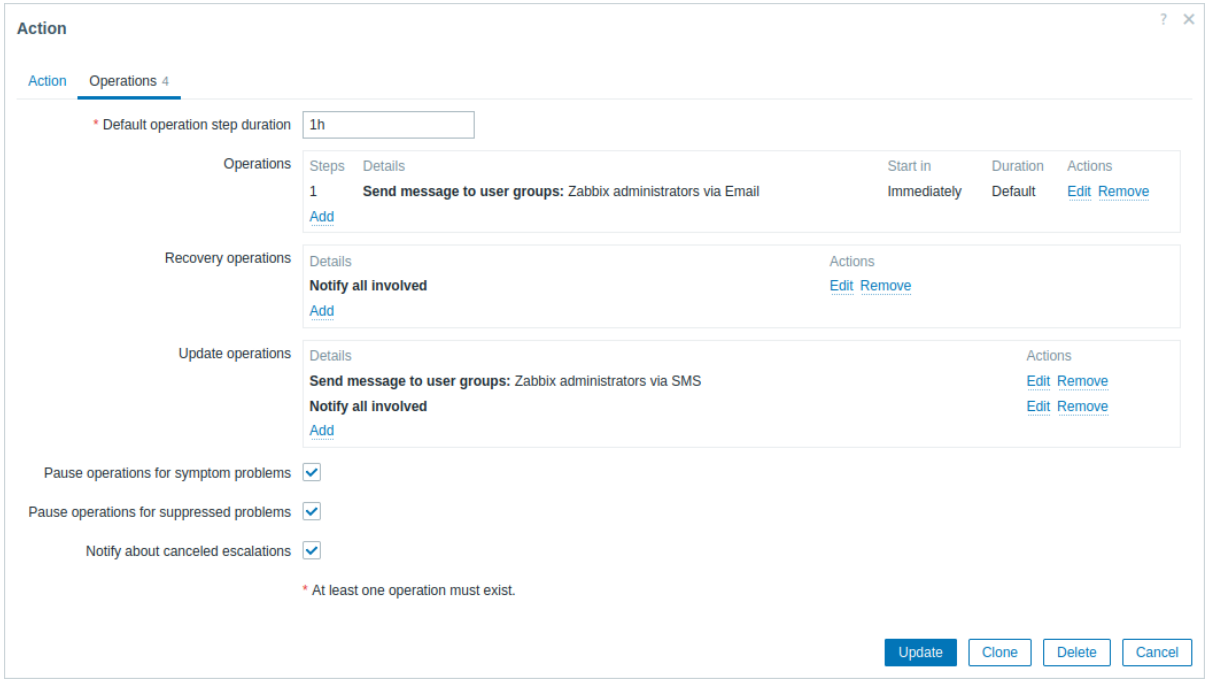
ディスカバリイベントおよび自動登録イベントでは、追加の実行内容を利用できます。

- ホストを追加
- ホストを削除
- ホストを有効化
- ホストを無効化
- ホストグループに追加
- ホストグループから削除
- ホストタグを追加
- ホストタグを削除
- テンプレートをリンク
- テンプレートのリンクを解除
- ホストインベントリモードを設定

オペレーションの設定

オペレーションを設定するには、次の手順を実行します。

1. アクションの設定で、オペレーションタブに移動します。
2. オペレーションの詳細を設定します。



オペレーションの一般的な属性:

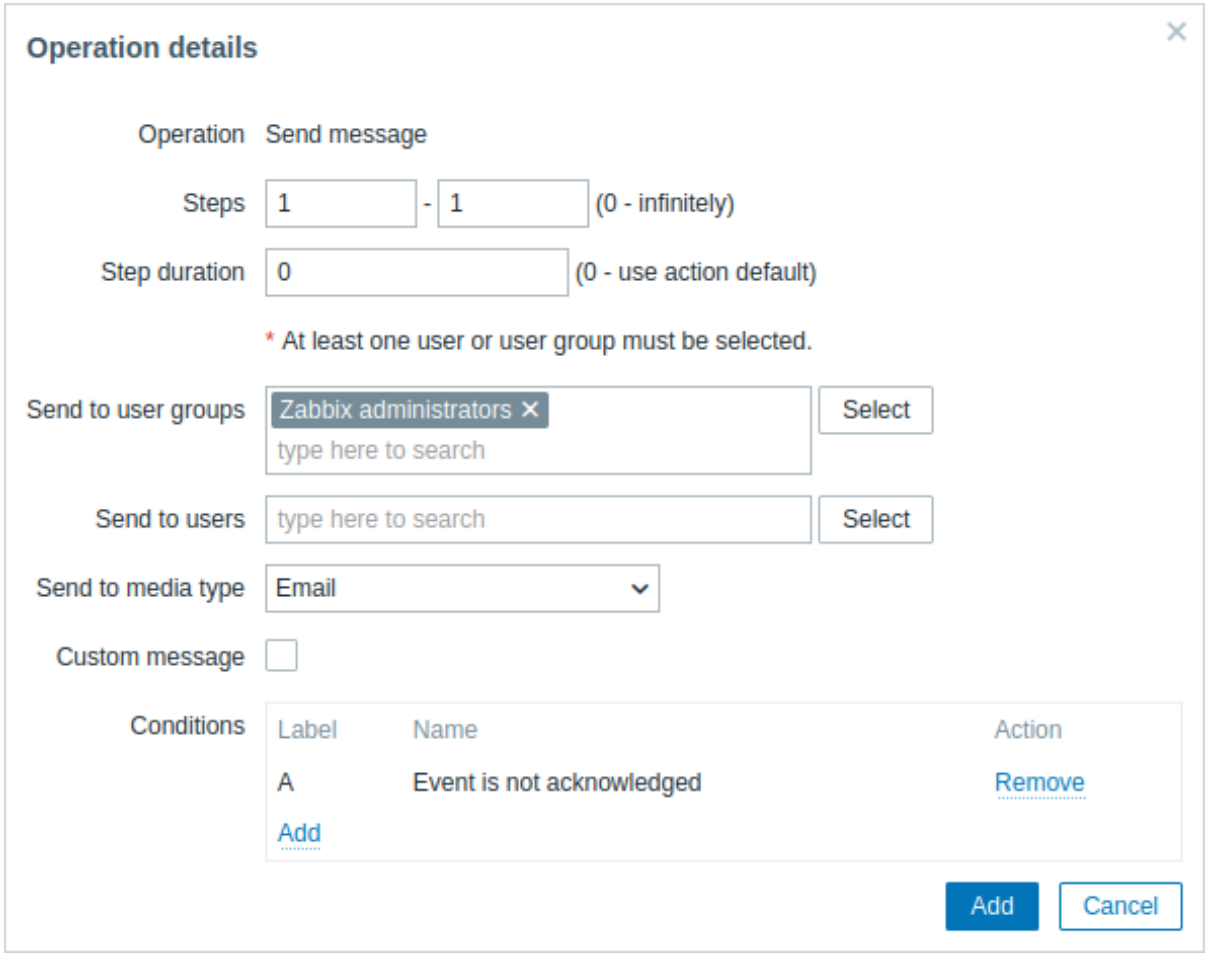
パラメータ	説明
デフォルトのオペレーションステップ期間	デフォルトでの 1 オペレーションステップの期間（60 秒から 1 週間）。 例えば、ステップ期間を 1 時間に設定した場合、オペレーションが実行されると、次のステップまでに 1 時間経過します。 時間のサフィックスがサポートされています（例: 60s、1m、2h、1d）。 ユーザーマクロがサポートされています。

パラメータ	説明
オペレーション	アクションオペレーション（存在する場合）が、次の詳細とともに表示されます。 ステップ - オペレーションが割り当てられているエスカレーションステップ。 詳細 - オペレーションのタイプと受信者/対象。 オペレーションの一覧には、使用されるメディアタイプ（メール、SMS、スクリプト）と、通知の受信者の名前および姓（ユーザー名の後ろに括弧内で表示）も表示されます。 開始時間 - イベント発生後、オペレーションが実行されるまでの時間。 期間（秒） - ステップ期間が表示されます。ステップでデフォルトの期間を使用する場合はデフォルトと表示され、カスタム期間を使用する場合は時間が表示されます。 アクション - オペレーションを編集および削除するためのリンクが表示されます。
復旧オペレーション	アクションオペレーション（存在する場合）が、次の詳細とともに表示されます。 詳細 - オペレーションのタイプと受信者/対象。 オペレーションの一覧には、使用されるメディアタイプ（メール、SMS、スクリプト）と、通知の受信者の名前および姓（ユーザー名の後ろに括弧内で表示）も表示されます。 アクション - オペレーションを編集および削除するためのリンクが表示されます。
更新オペレーション	アクションオペレーション（存在する場合）が、次の詳細とともに表示されます。 詳細 - オペレーションのタイプと受信者/対象。 オペレーションの一覧には、使用されるメディアタイプ（メール、SMS、スクリプト）と、通知の受信者の名前および姓（ユーザー名の後ろに括弧内で表示）も表示されます。 アクション - オペレーションを編集および削除するためのリンクが表示されます。
症状の障害に対するオペレーションを一時停止	症状の障害に対するオペレーション（最初のオペレーションの後）を一時停止するには、このチェックボックスをオンにします。 この設定は障害のエスカレーションにのみ影響し、復旧および更新オペレーションには影響しません。 このオプションはトリガーアクションでのみ使用できます。
抑制された障害に対するオペレーションを一時停止	メンテナンス期間中、オペレーションの開始を遅延させるには、このチェックボックスをオンにします。 メンテナンス終了後にオペレーションが開始されると、メンテナンス中に発生したイベントに対するオペレーションを含め、すべてのオペレーションが実行されます。 この設定は障害のエスカレーションにのみ影響し、復旧および更新オペレーションには影響しません。 このチェックボックスをオフにすると、メンテナンス期間中でもオペレーションは遅延なく実行されます。 このオプションはサービスアクションでは使用できません。
キャンセルされたエスカレーションについて通知	キャンセルされたエスカレーション（ホスト、アイテム、トリガー、またはアクションが無効になった場合）に関する通知を無効にするには、このチェックボックスをオフにします。

必須入力フィールドには、赤いアスタリスクが付いています。

新しいオペレーションの詳細を設定するには、オペレーションブロックの Add をクリックします。既存のオペレーションを編集するには、オペレーションの横にある Edit をクリックします。オペレーションステップの詳細を編集できるポップアップウィンドウが開きます。

操作の詳細



パラメータ	説明
操作	操作を選択します: メッセージを送信 - ユーザーにメッセージを送信します。 < リモートコマンド名 > - リモートコマンドを実行します。コマンドを実行するには、あらかじめスコープとしてアクション操作を選択してグローバルスクリプトで定義しておく必要があります。 ディスカバリおよび自動登録に基づくイベントでは、さらに多くの操作を利用できます(上記を参照)。
ステップ	エスカレーションスケジュールで操作を割り当てるステップを選択します: 開始 - このステップから実行を開始します。 終了 - このステップまで実行します（0= 無限。実行は制限されません）。
ステップの実行時間	これらのステップに対するカスタム実行時間です（0= デフォルトのステップ実行時間を使用）。 時間のサフィックスがサポートされています（例: 60s、1m、2h、1d）。 ユーザーマクロがサポートされています。 同じステップに複数の操作を割り当てることができます。これらの操作に異なるステップ実行時間が定義されている場合、最も短い時間が考慮され、ステップに適用されます。

パラメータ		説明
操作タイプ: メッセージを送信		
	ユーザーグループに送信	メッセージの送信先となるユーザーグループを選択します。 通知を受け取るには、ユーザーグループにホストへの少なくとも「読み取り」権限が必要です。
	ユーザーに送信	メッセージの送信先となるユーザーを選択します。 通知を受け取るには、ユーザーにホストへの少なくとも「読み取り」権限が必要です。
	メディアタイプに送信	利用可能なすべての（設定済みで有効な）メディアタイプ、または特定のメディアタイプだけにメッセージを送信します。
	カスタムメッセージ	選択すると、カスタムメッセージを設定できます。 webhookを介して内部イベントを通知する場合、カスタムメッセージは必須です。
	件名	カスタムメッセージの件名です。件名にはマクロを含めることができます。255 文字に制限されています。
	メッセージ	カスタムメッセージです。メッセージにはマクロを含めることができます。データベースの種類に応じて、一定の文字数に制限されます（詳細はメッセージの送信を参照）。

パラメータ		説明
操作タイプ: リモートコマンド		
	ターゲットリスト	コマンドを実行するターゲットを選択します: 現在のホスト - 障害イベントの原因となったトリガーのホスト上でコマンドを実行します。トリガーに複数のホストが含まれている場合、このオプションは機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。そのため、リモートコマンドはネストされたグループのホストでも実行されます。 ホストが複数回一致する場合（例: 複数のホストグループから、または個別とホストグループの両方から一致する場合）でも、そのホスト上のコマンドは 1 回だけ実行されます。 Zabbix サーバー上でカスタムスクリプトを実行する場合、ターゲットリストは意味を持ちません。この場合、ターゲットを追加で選択すると、サーバー上でスクリプトが実行される回数が増えるだけです。 グローバルスクリプトでは、ターゲットの選択がグローバルスクリプトの設定のホストグループ設定にも依存することに注意してください。 サービスアクションでは、リモートコマンドは常に Zabbix サーバー上で実行されるため、ターゲットリストオプションは利用できません。
条件		操作を実行する条件です: イベントが認識されていない - イベントが未認識の場合のみ実行します。 イベントが認識されている - イベントが認識済みの場合のみ実行します。 条件オプションはトリガーアクションでのみ利用できます。

完了したら、追加をクリックして操作を操作の一覧に追加します。

メッセージの送信

概要

メッセージの送信は、障害について人々に通知する最良の方法の 1 つです。これが、Zabbix が提供する主要なアクションの 1 つである理由です。

設定

Zabbix から通知を送受信できるようにするには、次の操作が必要です。

- メッセージの送信先となるメディアを定義します。

選択したメディアに対してユーザー設定で定義されている有効な時間帯以外に操作が実行された場合、メッセージは送信されません。

トリガー以外のイベント（ディスカバリ、アクティブエージェントの自動登録、内部イベントなど）の通知を受信する場合は、ユーザーのメディア設定でデフォルトのトリガー深刻度（「未分類」）を必ずチェックしてください。

- 定義済みのメディアのいずれかにメッセージを送信するアクションの実行内容を設定します。

Attention:
Zabbix は、イベントを生成したホストに対して少なくとも「読み取り」権限を持つユーザーにのみ通知を送信します。トリガーの式に含まれるホストのうち、少なくとも 1 台にはアクセスできる必要があります。

エスカレーションを使用して、メッセージ送信のカスタムシナリオを設定できます。

Zabbix からメールを正常に受信して読み取るには、メールサーバーおよびメールクライアントが標準の「SMTP/MIME メール」形式をサポートしている必要があります。Zabbix は UTF-8 データを送信するためです（件名に ASCII 文字のみが含まれる場合、UTF-8 エンコードは行われません）。「SMTP/MIME メール」形式の標準に従い、メッセージの件名と本文は Base64 エンコードされます。

すべてのマクロを展開した後のメッセージ制限は、リモートコマンドのメッセージ制限と同じです。

メッセージの追跡

Monitoring > *Problems* で送信されたメッセージのステータスを確認できます。

Actions 列では、実行されたアクションの概要情報を確認できます。ここで、緑の数字は送信されたメッセージ、赤の数字は失敗したメッセージを表します。*In progress* は、アクションが開始されたことを示します。*Failed* は、アクションが正常に実行されなかったことを示します。

イベント時刻をクリックしてイベント詳細を表示すると、*Actions* ブロックで、そのイベントにより送信されたメッセージ、または送信されなかったメッセージの詳細を確認できます。

Reports > *Action log* では、アクションが設定されているイベントについて、実行されたすべてのアクションの詳細を確認できます。

リモートコマンド

概要

リモートコマンドを使用すると、特定の事前定義済みコマンドを、ある条件が発生したときに監視対象のホスト上で自動的に実行するように定義できます。

このため、リモートコマンドは、インテリジェントでプロアクティブな監視を実現する強力な仕組みです。

この機能の最も一般的な使用例として、次のようなことができます。

- 応答しなくなった場合に、アプリケーション（Web サーバー、ミドルウェア、CRM など）を自動的に再起動する
- リクエストに応答しないリモートサーバーを、IPMI の「reboot」コマンドを使用して再起動する
- ディスク容量が不足した場合に、古いファイルを削除したり `/tmp` をクリーンアップしたりして、ディスク領域を自動的に解放する
- CPU 負荷に応じて、VM をある物理マシンから別の物理マシンへ移行する
- CPU、ディスク、メモリなどのリソースが不足した場合に、クラウド環境へ新しいノードを追加する

リモートコマンド用のアクションの設定は、メッセージ送信用のアクションの設定と似ています。唯一の違いは、Zabbix がメッセージを送信する代わりにコマンドを実行する点です。

リモートコマンドは、Zabbix サーバー、Zabbix プロキシ、または Zabbix エージェントによって実行できます。次の点に注意してください。

- Zabbix エージェント上のリモートコマンドは、Zabbix サーバーから直接、または Zabbix プロキシ経由で実行できます。
- Zabbix サーバーによって実行されるリモートコマンドは、終了コードの確認を含め、コマンドの実行で説明されている方法で実行されます。
- 対象のホストがメンテナンスモードの場合でも、リモートコマンドは実行されます。

リモートコマンドは、Zabbix エージェントと Zabbix プロキシの両方でデフォルトでは無効になっています。次のように有効にできます。

- Zabbix エージェントによって実行されるリモートコマンドの場合、エージェント設定に、許可する各コマンドに対して `AllowKey=system.run[<command>,*]` または `AllowKeyRegexp=^system\.run\[<command>,.*\]$` パラメータを追加します。`*` は、どちらの実行モード（`wait` または `nowait`）にも一致します。
- Zabbix プロキシによって実行されるリモートコマンドの場合、プロキシ設定で `EnableRemoteCommands` パラメータを 1 に設定します。

プロキシによって監視されているホスト上で Zabbix エージェントがリモートコマンドを実行する場合、Zabbix プロキシでリモートコマンドを有効にする必要はありません。

Attention:
必要な場合を除き、AllowKey=system.run[*] または AllowKeyRegexp=^system\.run\[.*\]$ は使用しないでください。これらを使用すると、ホスト上で任意のコマンドを実行できるようになります。セキュリティ上の理由から、完全に一致するコマンドのみを許可することを推奨します。system.run[*] はデフォルトで拒否されています (DenyKey=system.run[*])。

エージェントチェックの制限も参照してください。

リモートコマンドの制限

すべてのマクロを解決した後のリモートコマンドの制限は、データベースの種類と文字セット（非 ASCII 文字は格納に 1 バイト以上必要）によって異なります。

データベース	文字数の制限	バイト数の制限
MySQL	65535	65535
PostgreSQL	65535	無制限
SQLite (Zabbix プロキシのみ)	65535	無制限

リモートコマンドの実行出力（戻り値）は 16MB に制限されています（末尾の空白も含み、切り捨てられます）。IPMI リモートコマンドの制限は、インストールされている IPMI ライブラリに基づきます。データベースの制限は、すべてのリモートコマンドに適用されることに注意してください。

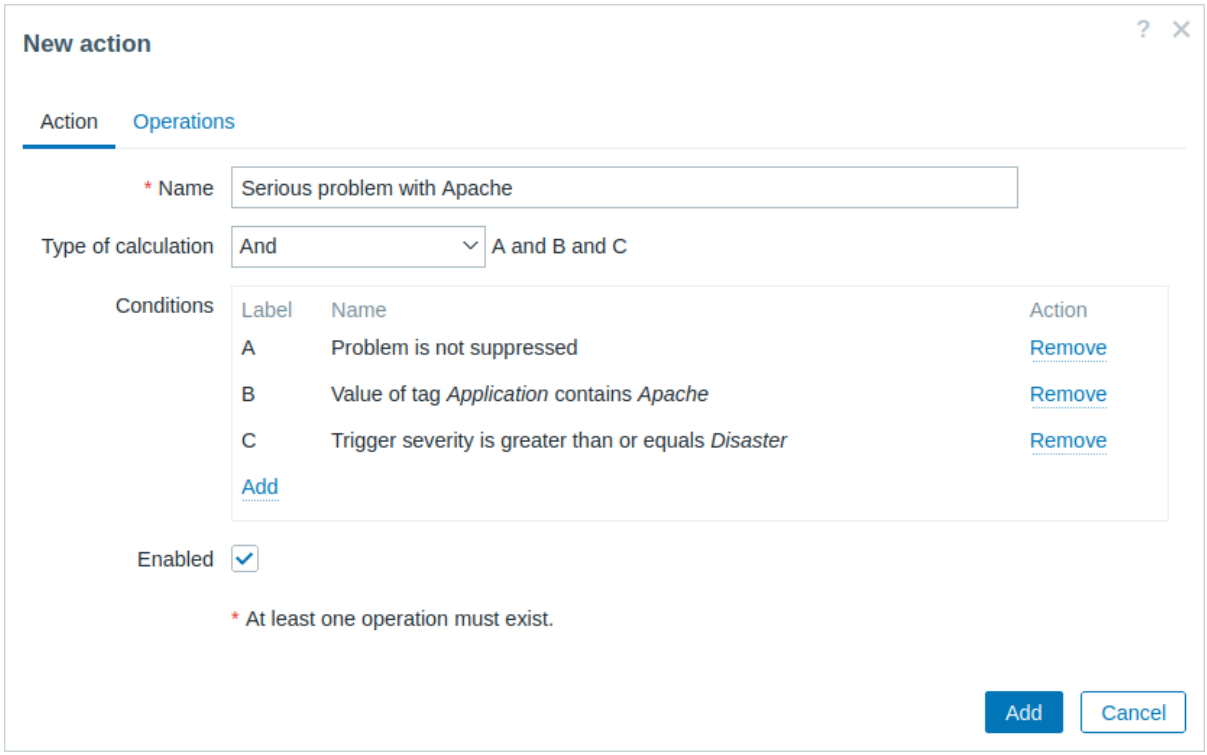
設定

Zabbix エージェントが実行するリモートコマンドは、まずAllowKey または AllowKeyRegexp パラメータを使用して、エージェントの設定 で許可する必要があります。

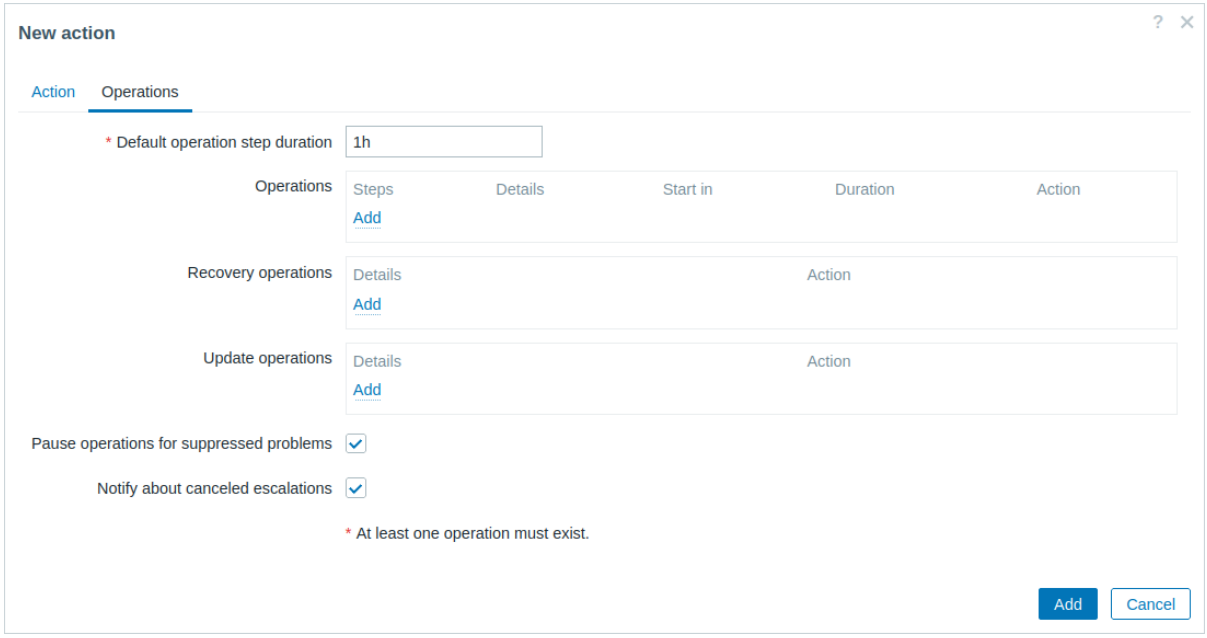
各コマンドを nowait モードで実行できるようにするには、許可するコマンドごとに AllowKey=system.run[<command>,*] または AllowKeyRegexp=^system\.run\[<command>,.*\]$ パラメータがエージェントの設定に追加されていることを確認してください。パラメータを変更した後、エージェントデーモンを再起動します。

次に、障害 > アクション > トリガーアクションで新しいアクションを設定する場合:

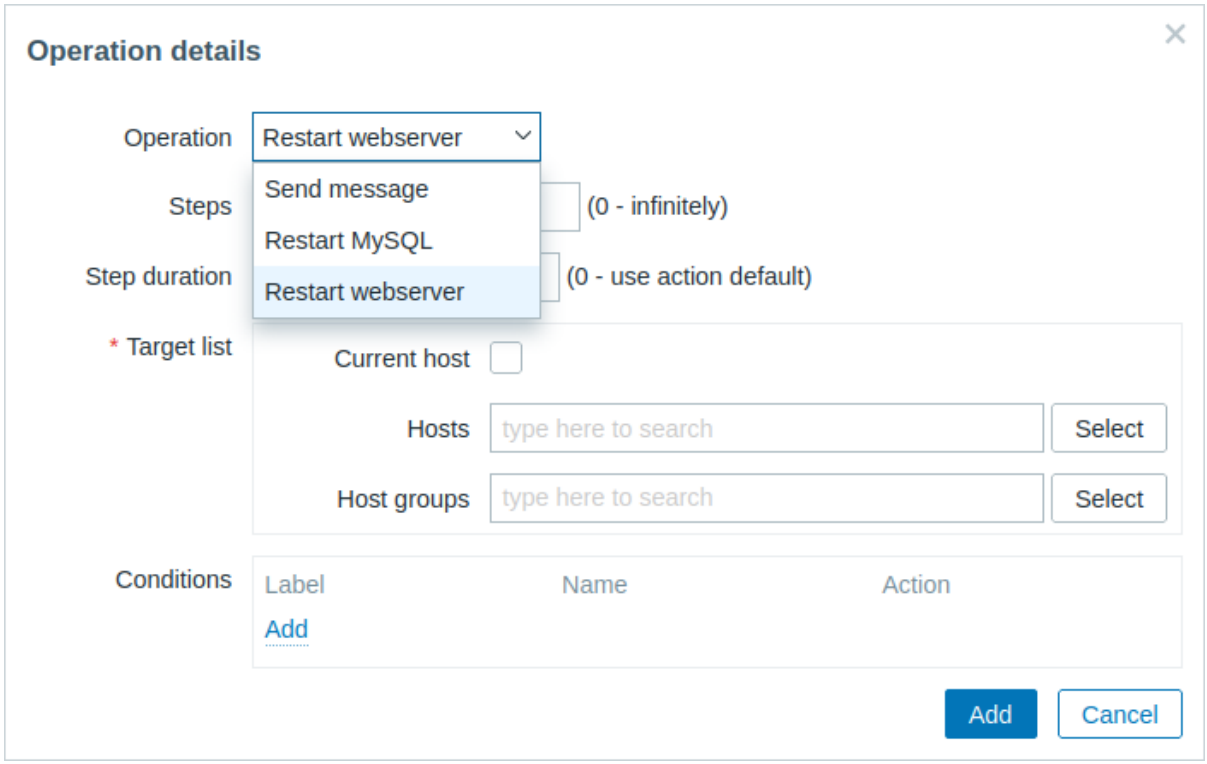
1. 適切な条件を定義します。たとえば、いずれかの Apache アプリケーションで重大度が「致命的」の障害が発生したときにアクションが有効になるように設定します。



2. 実行内容タブで、実行内容、復旧時の実行内容、または更新時の実行内容セクションの追加をクリックします。



3. 実行内容ドロップダウンリストから定義済みスクリプトのいずれかを選択し、スクリプトの対象リストを設定します。



定義済みスクリプト

アクション操作で利用できるスクリプト（webhook、script、SSH、Telnet、IPMI）は、スコープとして *Action operation* を選択したglobal scripts で定義されます。

例:

```
sudo /etc/init.d/apache restart
```

この場合、Zabbix は Apache プロセスの再起動を試みます。このコマンドでは、コマンドが Zabbix エージェント上で実行されるようにしてください（*Execute on* に対して *Zabbix agent* ボタンをクリックします）。

Attention:
sudo の使用に注意してください。Zabbix ユーザーには、デフォルトではシステムサービスを再起動する権限がありません。sudo の設定方法については、以下のヒントを参照してください。

Note:
Zabbix エージェント 7.0 以降では、アクティブモードで動作しているエージェント上でもリモートコマンドを実行できます。Zabbix エージェントは、アクティブ/パッシブのいずれであってもリモートホスト上で動作し、コマンドをバックグラウンドで実行します。

Zabbix エージェント上のリモートコマンドは、`system.run[,nowait]` キーによってタイムアウトなしで実行され、実行結果は確認されません。Zabbix サーバーおよび Zabbix プロキシでは、リモートコマンドは `zabbix_server.conf` または `zabbix_proxy.conf` ファイルの`TrapperTimeout` パラメータで設定されたタイムアウト付きで実行され、実行結果は確認 されます。詳細については、*Script timeout* を参照してください。

アクセス権限

zabbix ユーザーに、設定されたコマンドの実行権限があることを確認してください。特権コマンドへのアクセスを許可するために sudo を使用するとよい場合があります。アクセスを設定するには、root として次を実行します。

```
visudo
```

sudoers ファイルで使用できる例は次のとおりです。

```
# 'zabbix' ユーザーがパスワードなしで全コマンドを実行できるようにします。
zabbix ALL=NOPASSWD: ALL

# 'zabbix' ユーザーがパスワードなしで apache を再起動できるようにします。
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
一部のシステムでは、sudoers ファイルによりローカル以外のユーザーがコマンドを実行できないようになっています。これを変更するには、/etc/sudoers 内の `requiretty` オプションをコメントアウトしてください。

複数のインターフェースを持つリモートコマンド

ターゲットシステムが選択されたタイプ (Zabbix エージェントまたは IPMI) の複数のインターフェースを持っている場合、リモートコマンドはデフォルトのインターフェースで実行されます。

Zabbix エージェント以外のインターフェースを使用して、SSH および Telnet 経由でリモートコマンドを実行することも可能です。使用可能なインターフェースは、次の順序で選択されます:

- Zabbix エージェントのデフォルトインターフェース
- SNMP のデフォルトインターフェース
- JMX のデフォルトインターフェース
- IPMI のデフォルトインターフェース

IPMI リモートコマンド

IPMI リモートコマンドの場合、次の構文を使用する必要があります。

```
<command> [<value>]
```

内容

- <command> - スペースなしの 1 つの IPMI コマンド
- <value> - 'on', 'off' または任意の符号なし整数。<value> はオプションパラメーターです。

例

アクションの実行内容でリモートコマンドとして使用できるグローバルスクリプトの例。

例 **1**

特定の条件で Windows を再起動します。

Zabbix によって障害が検出されたときに Windows を自動的に再起動するには、次のスクリプトを定義します。

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`c:\windows\system32\shutdown.exe -r -f`

例 **2**

IPMI 制御を使用してホストを再起動します。

スクリプトパラメータ	値
スコープ	' アクションオペレーション'
タイプ	'IPMI'
コマンド	reset

例 **3**

IPMI 制御を使用してホストの電源をオフにします。

スクリプトパラメータ	値
スコープ	' アクションオペレーション'
タイプ	'IPMI'
コマンド	power off

追加の操作

概要

このセクションでは、ディスカバリ/自動登録イベントの追加操作の詳細を確認できます。

ホストの追加

ホストはディスカバリプロセス中に追加されます。ホストが検出されるとすぐに追加され、ディスカバリプロセスの最後ではありません。

Note:
ネットワークディスカバリは、多くの利用できないホスト/サービスがあるため、時間がかかる場合があります。忍耐強く、適切な IP レンジを使用することをお勧めします。

ホストを追加する際、その名前は標準の **gethostbyname** 関数によって決定されます。ホストが解決できる場合は、解決された名前が使用されます。できない場合は、IP アドレスが使用されます。また、ホスト名に IPv6 アドレスを使用する必要がある場合、ホスト名にコロン（:）は使用できないため、すべての「:」（コロン）は「_」（アンダースコア）に置き換えられます。

Attention:
プロキシによるディスカバリを実行している場合でも、現在はホスト名のルックアップは Zabbix サーバー上で行われます。

Attention:
Zabbix の設定に、検出された新しいホストと同じ名前のホストがすでに存在する場合、Zabbix はホスト名に **_N** を追加します。ここで **N** は 2 から始まる増加する番号です。

メッセージでマクロを使用する

概要

メッセージの件名やメッセージ本文では、より効率的な問題報告のためにマクロを使用できます。

多くの組み込みマクロに加えて、ユーザーマクロや式マクロもサポートされています。Zabbix でサポートされているマクロの全リストも利用できます。

例

以下の例は、メッセージでマクロを使用する方法を示しています。

例 1

メッセージの件名:

```
Problem: {TRIGGER.NAME}
```

メッセージを受信すると、メッセージの件名は次のように置き換えられます:

```
Problem: Processor load is too high on Zabbix server
```

例 2

メッセージ:

```
Processor load is: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

このメッセージを受信すると、メッセージは次のように置き換えられます:

```
Processor load is: 1.45
```

例 3

メッセージ:

```
最新値: {?last(/{HOST.HOST}/{ITEM.KEY})}
15分間の最大値: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
15分間の最小値: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

このメッセージを受信すると、次のように置き換えられます:

```
最新値: 1.45
15分間の最大値: 2.33
15分間の最小値: 1.01
```

例 4

メッセージ:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

このメッセージを受信すると、イベントの詳細ページへのリンクが含まれます。このページには、イベント、そのトリガー、および同じトリガーによって生成された最新のイベントのリストに関する情報が表示されます。

例 5

トリガー式で複数のホストからの値を通知する。

メッセージ:

```
Problem name: {TRIGGER.NAME}
Trigger expression: {TRIGGER.EXPRESSION}

1. Item value on {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Item value on {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

このメッセージを受信すると、次のように置き換えられます:

```
Problem name: Processor load is too high on a local host
Trigger expression: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[perc

1. Item value on Myhost: 0.83 (Processor load (1 min average per core))
2. Item value on Myotherhost: 5.125 (Processor load (1 min average per core))
```

例 6

リカバリメッセージで、問題イベントとリカバリエベントの両方の詳細を受信する:

メッセージ:

```
問題:

イベントID: {EVENT.ID}
イベント値: {EVENT.VALUE}
イベントステータス: {EVENT.STATUS}
イベント時刻: {EVENT.TIME}
イベント日付: {EVENT.DATE}
イベント経過時間: {EVENT.AGE}
イベントの確認: {EVENT.ACK.STATUS}
イベントの更新履歴: {EVENT.UPDATE.HISTORY}

リカバリ:
```

```
イベントID: {EVENT.RECOVERY.ID}
イベント値: {EVENT.RECOVERY.VALUE}
イベントステータス: {EVENT.RECOVERY.STATUS}
イベント時刻: {EVENT.RECOVERY.TIME}
イベント日付: {EVENT.RECOVERY.DATE}
運用データ: {EVENT.OPDATA}
```

メッセージを受信すると、マクロは次のように置き換えられます:

```
問題:

イベントID: 21874
イベント値: 1
イベントステータス: PROBLEM
イベント時刻: 13:04:30
イベント日付: 2018.01.02
イベント経過時間: 5m 0s
イベントの確認: Yes
イベントの更新履歴: 2018.01.02 13:05:51 "John Smith (Admin)"
Actions: acknowledged.

リカバリ:

イベントID: 21896
イベント値: 0
イベントステータス: OK
イベント時刻: 13:10:07
イベント日付: 2018.01.02
運用データ: Current value is 0.83
```

復旧操作

概要

復旧操作により、障害が解決されたときに通知を受けることができます。

復旧操作では、メッセージとリモートコマンドの両方がサポートされています。複数の操作を追加できますが、エスカレーションはサポートされていません。すべての操作が 1 つのステップに割り当てられるため、同時に実行されます。

使用例

リカバリー操作の使用例は以下の通りです。

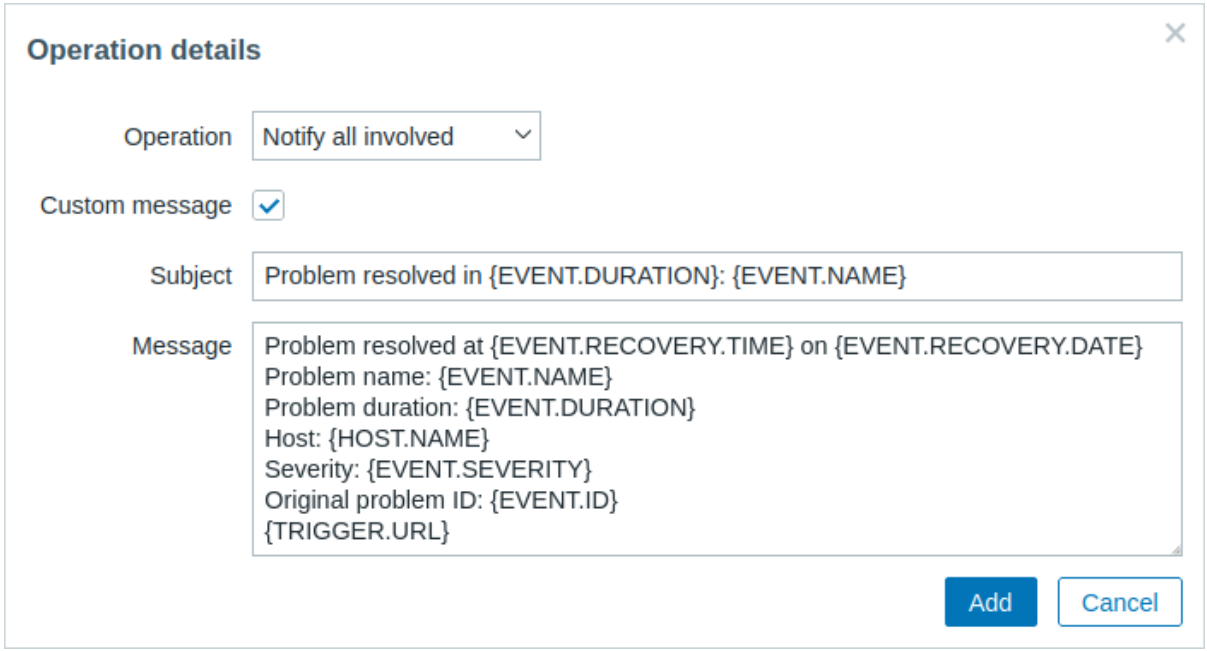
1. 問題時に通知されたすべてのユーザーにリカバリー時も通知する:
    - 操作タイプとしてすべての関係者に通知を選択します。
2. リカバリー時に複数の操作を行う: 通知を送信し、リモートコマンドを実行する:
    - メッセージ送信とコマンド実行の操作タイプを追加します。
3. 外部のヘルプデスク/チケッティングシステムでチケットをオープンし、問題が解決したらクローズする:
    - ヘルプデスクシステムと通信する外部スクリプトを作成します。
    - このスクリプトを実行してチケットをオープンする操作を持つアクションを作成します。
    - このスクリプトを他のパラメータで実行してチケットをクローズするリカバリー操作を作成します。
    - {EVENT.ID} マクロを使用して元の問題を参照します。

リカバリー操作の設定

リカバリー操作を設定するには、以下の手順に従います。

1. アクション設定フォームの操作タブに移動します。
2. 新しいリカバリー操作を設定するには、リカバリー操作セクションで追加をクリックします。既存の操作を編集するには、操作の横にある編集をクリックします。
3. 操作の詳細を設定します。

復旧操作の詳細



復旧イベントでは、次の 3 種類の操作タイプを利用できます。

- メッセージの送信 - 指定したユーザーに復旧メッセージを送信します。
- 関係者全員に通知 - 問題イベントで通知されたすべてのユーザーに復旧メッセージを送信します。
- **<** リモートコマンド名 **>** - リモートコマンドを実行します。コマンドは、事前にグローバルスクリプトでスコープとして *Action operation* を選択して定義されている場合に実行できます。

各操作タイプのパラメータについては、以下で説明します。必須の入力フィールドには、すべて赤いアスタリスクが付いています。
設定が完了したら、追加をクリックして、操作を 復旧操作の一覧に追加します。

Note:
同じ受信者が複数の操作タイプに定義されていて、カスタムメッセージが指定されていない場合、重複した通知は送信されません。

操作タイプ：メッセージの送信

パラメータ	説明
ユーザーグループに送信	リカバリーメッセージを送信するユーザーグループを選択します。 通知を受けるには、ユーザーグループがホストに対して少なくとも「読み取り」権限を持っている必要があります。
ユーザーに送信	リカバリーメッセージを送信するユーザーを選択します。 通知を受けるには、ユーザーがホストに対して少なくとも「読み取り」権限を持っている必要があります。

パラメータ		説明
メディアタイプに送信		すべての利用可能な（設定済みかつ有効な）メディアタイプ、または特定のメディアタイプのみにデフォルトのリカバリーメッセージを送信します。
カスタムメッセージ		選択すると、カスタムメッセージを定義できます。
	件名	カスタムメッセージの件名。件名にはマクロを含めることができます。
	メッセージ	カスタムメッセージ。メッセージにはマクロを含めることができます。

操作タイプ: リモートコマンド

パラメーター	説明
ターゲットリスト	コマンドを実行するターゲットを選択します。 現在のホスト - 障害イベントの原因となったトリガーのホスト上でコマンドを実行します。トリガーに複数のホストが含まれている場合、このオプションは機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。そのため、リモートコマンドはネストされたグループのホストでも実行されます。 ホストが複数回一致する場合（例えば、複数のホストグループから、または個別およびホストグループから一致する場合）でも、そのホスト上のコマンドは 1 回だけ実行されます。 コマンドが Zabbix サーバー上で実行される場合、ターゲットリストは意味を持ちません。この場合、より多くのターゲットを選択すると、サーバー上でコマンドがより多く実行されるだけです。 グローバルスクリプトの場合、ターゲットの選択はグローバルスクリプトの設定にあるホストグループ設定にも依存することに注意してください。

操作タイプ : 関係者全員に通知

パラメータ		説明
カスタムメッセージ		選択すると、カスタムメッセージを定義できます。
	件名	カスタムメッセージの件名。件名にはマクロを含めることができます。
	メッセージ	カスタムメッセージ。メッセージにはマクロを含めることができます。

更新操作

概要

更新オペレーションは、次のイベントソースを持つアクションで利用できます。

- トリガー - 他のユーザーによって障害が更新された場合。つまり、コメントが追加された、承認された、深刻度が変更された、(手動で）クローズされた場合です。

- サービス - サービスの深刻度が変更されたが、サービスはまだ復旧していない場合です。

ユーザーは自分自身の更新については通知を受け取らないことに注意してください。

更新オペレーションでは、メッセージとリモートコマンドの両方がサポートされています。複数のオペレーションを追加できますが、エスカレーションはサポートされません。すべてのオペレーションは単一のステップに割り当てられるため、同時に実行されます。

更新操作の設定

更新操作を設定するには、以下の手順に従います。

1. アクションの設定フォームのオペレーションタブに移動します。
2. 新しい更新操作を設定するには、更新操作セクションで追加をクリックします。既存の操作を編集するには、操作の横にある編集をクリックします。
3. 操作の詳細を設定します。

更新操作の詳細

更新操作では、リカバリ操作と同じパラメータセットが提供されます。

エスカレーション

概要

エスカレーションを使用すると、通知の送信やリモートコマンドの実行のためのカスタムシナリオを作成できます。

実際には、これは次のことを意味します。

- ユーザーは新しい障害について即座に通知を受けることができます。
- 障害が解決されるまで通知を繰り返すことができます。
- 通知の送信を遅らせることができます。
- 通知を別の「より上位の」ユーザーグループにエスカレーションできます。
- リモートコマンドは、即座に実行することも、障害が長時間解決されない場合に実行することもできます。

アクションは、エスカレーションステップに基づいてエスカレーションされます。各ステップには継続時間があります。

デフォルトの継続時間と、個々のステップのカスタム継続時間の両方を定義できます。1 つのエスカレーションステップの最小継続時間は 60 秒です。

通知の送信やコマンドの実行などのアクションは、任意のステップから開始できます。ステップ 1 は即時アクション用です。アクションを遅らせたい場合は、後のステップに割り当てることができます。各ステップに対して、複数のアクションを定義できます。

エスカレーションステップの数に制限はありません。

エスカレーションは、実行内容の設定時に定義します。エスカレーションは障害時の実行内容でのみサポートされ、復旧時にはサポートされません。

エスカレーション動作に関するその他の事項

アクションに複数のエスカレーションステップが含まれている場合、さまざまな状況で何が起こるかを見てみましょう。

状況	動作
初期障害通知の送信後、対象のホストがメンテナンスに入った	アクションの設定にある抑制された障害の操作を一時停止設定に応じて、残りのすべてのエスカレーションステップは、メンテナンス期間による遅延後、または遅延なしで実行されます。メンテナンス期間によって操作がキャンセルされることはありません。
初期通知の送信後、期間アクション条件で定義された時間帯が終了した	残りのすべてのエスカレーションステップが実行されます。期間条件によって操作を停止することはできません。この条件は、アクションを開始するかどうかに影響しますが、操作には影響しません。
メンテナンス中に障害が開始し、メンテナンス終了後も継続している（解決していない）	アクションの設定にある抑制された障害の操作を一時停止設定に応じて、すべてのエスカレーションステップは、メンテナンス終了時点から、または直ちに実行されます。
データなしメンテナンス中に障害が開始し、メンテナンス終了後も継続している（解決していない）	すべてのエスカレーションステップが実行される前に、トリガーが発火するまで待機する必要があります。

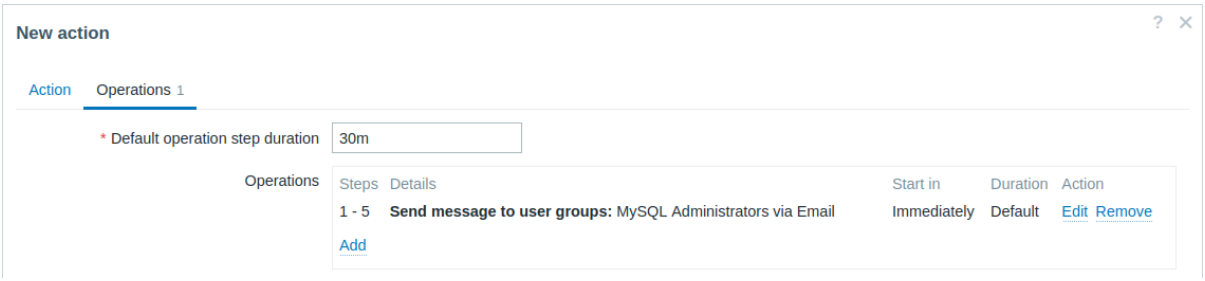
状況	動作
異なるエスカレーションが短い間隔で続けて発生し、重複した	新しいエスカレーションの実行によって、以前のエスカレーションが上書きされます。ただし、以前のエスカレーションでは、少なくとも 1 つのエスカレーションステップが常に実行されます。この動作は、トリガーの障害評価ごとに作成されるイベントに対するアクションに関係します。
エスカレーションの実行中（メッセージの送信中など）に、次のいずれかが発生した <*br*>- アクションが無効になった <*br*> トリガーイベントに基づく場合：<*br*>- トリガーが無効になった <*br*>- ホストまたはアイテムが無効になった <*br*> トリガーに関する内部イベントに基づく場合：<*br*>- トリガーが無効になった <*br*> アイテム/ローレベルディスカバリルールに関する内部イベントに基づく場合：<*br*>- アイテムが無効になった <*br*>- ホストが無効になった	実行中のメッセージが送信された後、エスカレーションに関するメッセージがもう 1 通送信されます。後続のメッセージ本文の先頭には、キャンセルの理由を示すキャンセルテキスト（注: エスカレーションがキャンセルされました）が含まれます（例: 注: エスカレーションがキャンセルされました: アクション「*<Action name>*」が無効になりました）。これにより、受信者はエスカレーションがキャンセルされ、これ以上のステップが実行されないことを把握できます。このメッセージは、それまでに通知を受信したすべての宛先に送信されます。キャンセルの理由はサーバーのログファイルにも記録されます（デバッグレベル 3=Warning 以上）。 操作が完了していても、復旧操作が設定されており、まだ実行されていない場合は、エスカレーションがキャンセルされましたメッセージが送信されることに注意してください。
エスカレーションの実行中（メッセージの送信中など）にアクションが削除された	これ以上メッセージは送信されません。この情報はサーバーのログファイルに記録されます（デバッグレベル 3=Warning 以上）。例: `escalation canceled: action id:334 deleted`

エスカレーション例

例 1

30 分ごとに繰り返し通知を送信する（合計 5 回）”MySQL Administrators” グループに送信します。設定方法は以下の通りです。

- オペレーションタブで、デフォルトのオペレーションステップの期間を”30m”（30 分）に設定します。
- エスカレーションのステップを”1” から”5” までに設定します。
- メッセージの受信者として”MySQL Administrators” グループを選択します。



問題が発生してから 0:00、0:30、1:00、1:30、2:00 の時点で通知が送信されます（もちろん、問題がそれより早く解決された場合は除きます）。

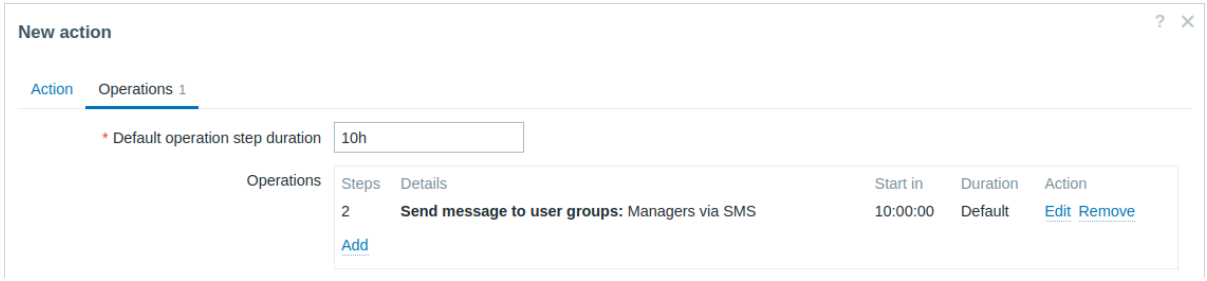
問題が解決され、リカバリーメッセージが設定されている場合、このエスカレーションシナリオで少なくとも 1 つの問題メッセージを受信した人に送信されます。

Note:
アクティブなエスカレーションを生成したトリガーが無効化された場合、Zabbix はすでに通知を受信したすべてのユーザーにその旨の情報メッセージを送信します。

例 2

長時間継続している問題について遅延通知を送信します。設定方法:

- オペレーションタブで、デフォルトのオペレーションステップの期間を「10h」(10 時間) に設定します。
- エスカレーションのステップを「2」から「2」に設定します。



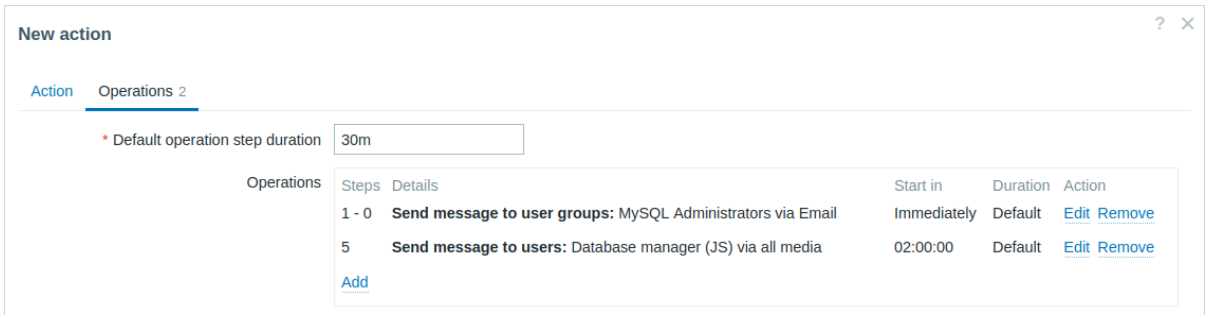
通知は、エスカレーションシナリオのステップ 2、つまり問題発生から 10 時間後にのみ送信されます。

メッセージテキストを「問題が 10 時間以上継続しています」などにカスタマイズすることもできます。

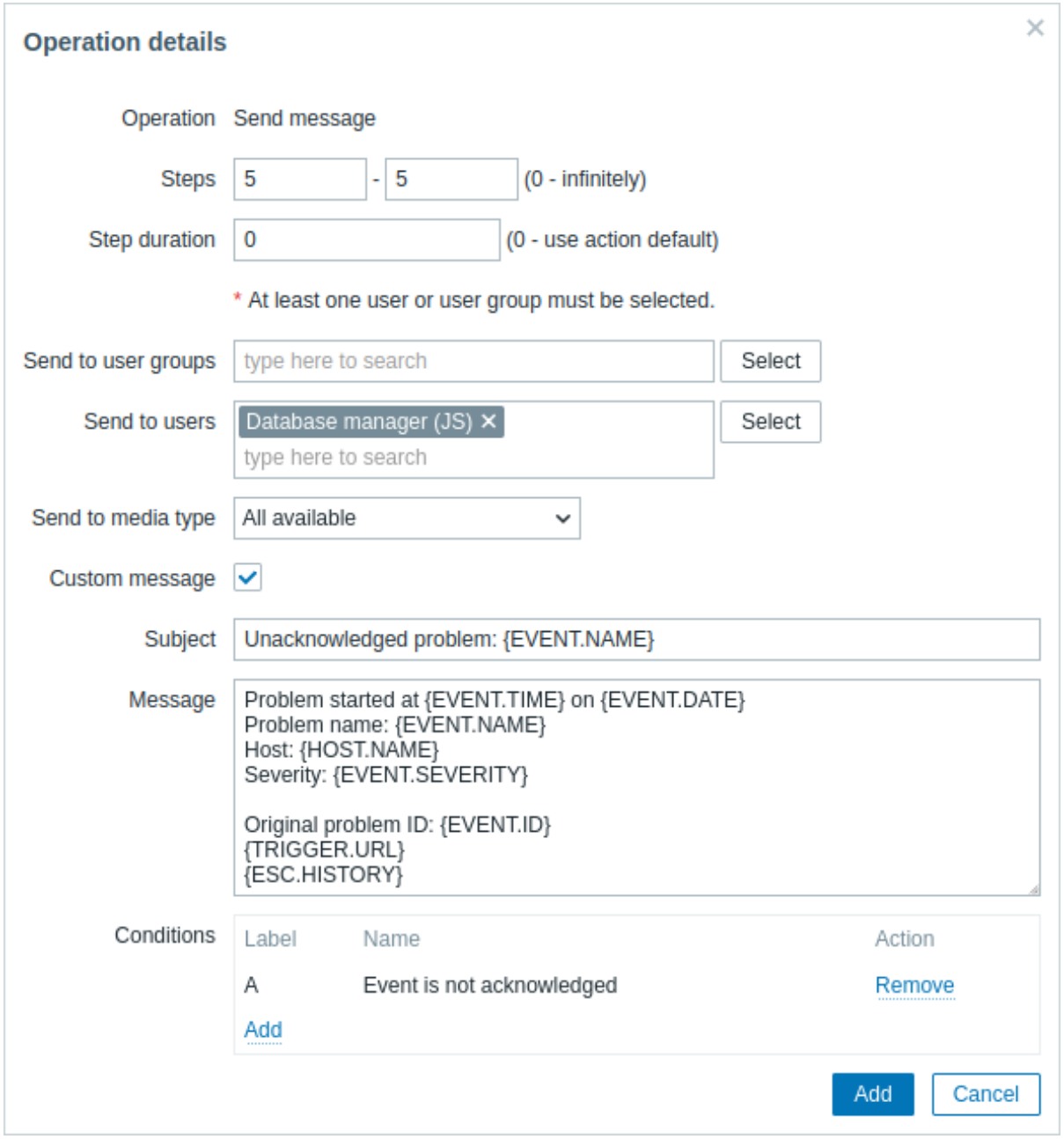
例 3

問題を上司にエスカレーションする。

上記の最初の例では、MySQL 管理者への定期的なメッセージ送信を設定しました。この場合、管理者は問題がデータベースマネージャーにエスカレーションされる前に 4 回メッセージを受け取ります。マネージャーは、問題がまだ確認されていない場合、つまり誰も対応していない場合にのみメッセージを受け取ることに注意してください。



操作 2 の詳細:



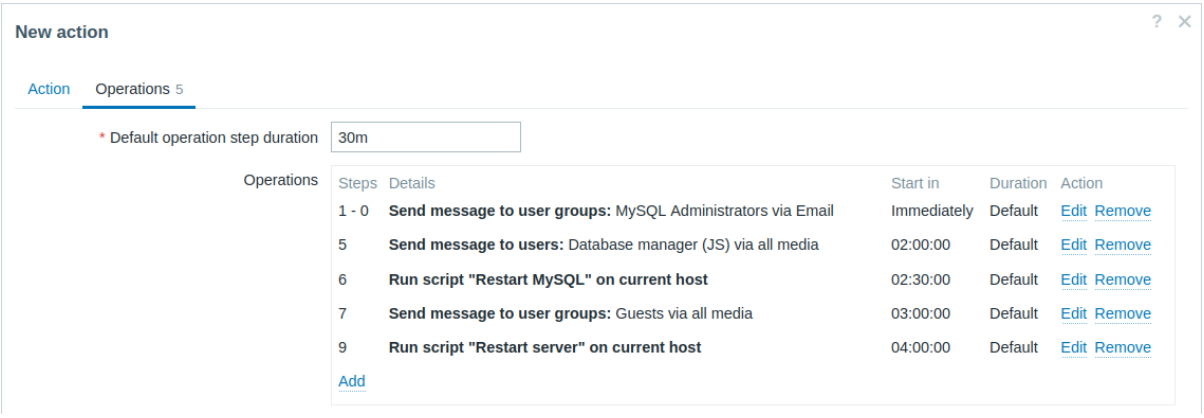
カスタマイズされたメッセージで {ESC.HISTORY} マクロを使用していることに注意してください。このマクロには、このエスカレーションで以前に実行されたすべてのステップ（送信された通知や実行されたコマンドなど）に関する情報が含まれます。

例 4

より複雑なシナリオ。MySQL 管理者への複数のメッセージとマネージャーへのエスカレーションの後、Zabbix は MySQL データベースの再起動を試みます。障害が 2:30 時間存在し、確認されていない場合に発生します。

それでも障害が解決しない場合は、さらに 30 分後に Zabbix がすべてのゲストユーザーにメッセージを送信します。

これで障害が解決しない場合は、さらに 1 時間後に、Zabbix は IPMI コマンドを使用して MySQL データベース (2 番目のリモートコマンド) でサーバーを再起動します。



例 5

重複するステップ範囲とカスタム間隔を持つ複数のオペレーションを含むエスカレーション。デフォルトのオペレーションステップの継続時間は 30 分です。

通知は以下のように送信されます。

- 問題発生後、MySQL 管理者には 0:00、0:30、1:00、1:30 に通知されます。
- データベースマネージャーには 2:00 と 2:10 に通知されます（後続のオペレーションで定義された短いカスタムステップ継続時間 10 分が、ここで設定された長いステップ継続時間 1 時間を上書きします。これは、オペレーションの詳細のステップ継続時間で、ステップが重複する場合の動作として説明されています）。
- Zabbix 管理者には、問題発生後 2:00、2:10、2:20 に通知されます（カスタムステップ継続時間 10 分が適用されます）。
- ゲストユーザーには、問題発生後 4:00 に通知されます（ステップ 8 と 11 の間はデフォルトのステップ継続時間 30 分が有効になります）。

サポートされていないアイテムに関する通知を受信する

概要

Zabbix では、サポートされていないアイテムに関する通知を受け取ることができます。

これは Zabbix の内部イベントの概念の一部であり、これらの状況でユーザーに通知できるようにします。内部イベント は状態の変化を反映します。

- アイテムが「正常」から「サポートされていない」状態に変わるとき（およびその逆）；
- トリガーが「正常」から「不明」状態に変わるとき（およびその逆）；
- 低レベルディスカバリルールが「正常」から「サポートされていない」状態に変わるとき（およびその逆）。

このセクションでは、アイテムがサポートされていない状態になったときに通知を受け取るための手順を説明します。

設定

おおまかには通知を設定するプロセスは以前に Zabbix でアラートを設定したことがある人には馴染みのあるものです。

ステップ 1

通知に使用するメディア（メール、SMS、またはスクリプトなど）を設定します。この作業を実行するには、マニュアルの対応するセクションを参照してください。

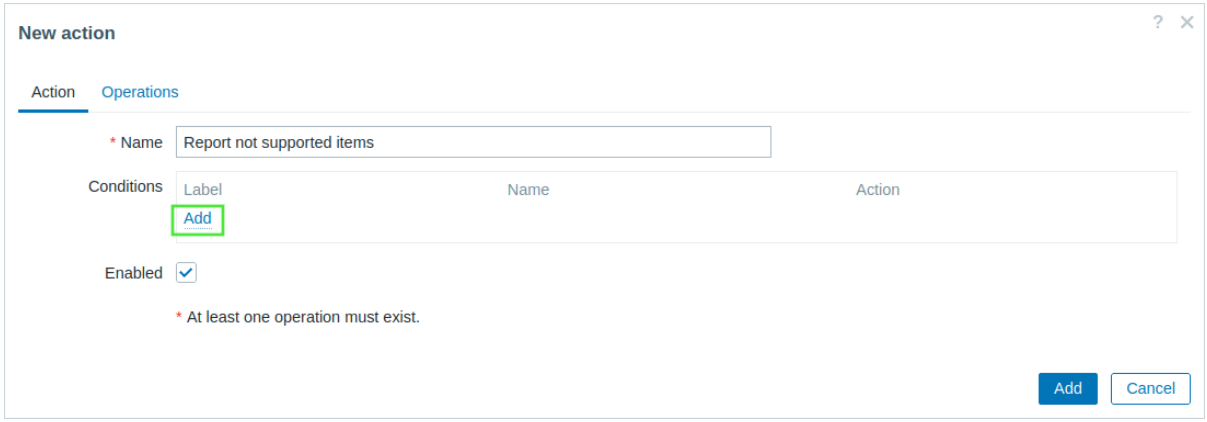
Attention:
内部イベントの通知にはデフォルトの深刻度（「未分類」）が使用されるため、内部イベントの通知を受け取りたい場合は、ユーザーメディアの設定時にこの項目をチェックしたままにしてください。

ステップ 2

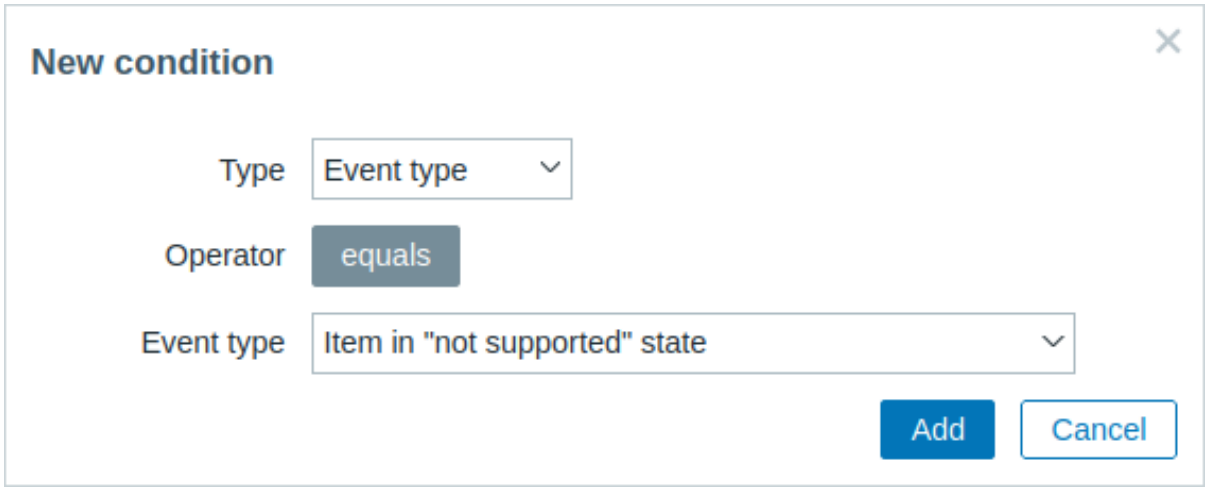
1. *Alerts* > *Actions* > *Internal actions* に移動します。
2. ページ右上の *Create action* をクリックして、アクション設定フォームを開きます。

ステップ 3

1. アクションタブでアクションの名前を入力します。
2. 条件ブロックの 追加をクリックして、新しい条件を追加します。



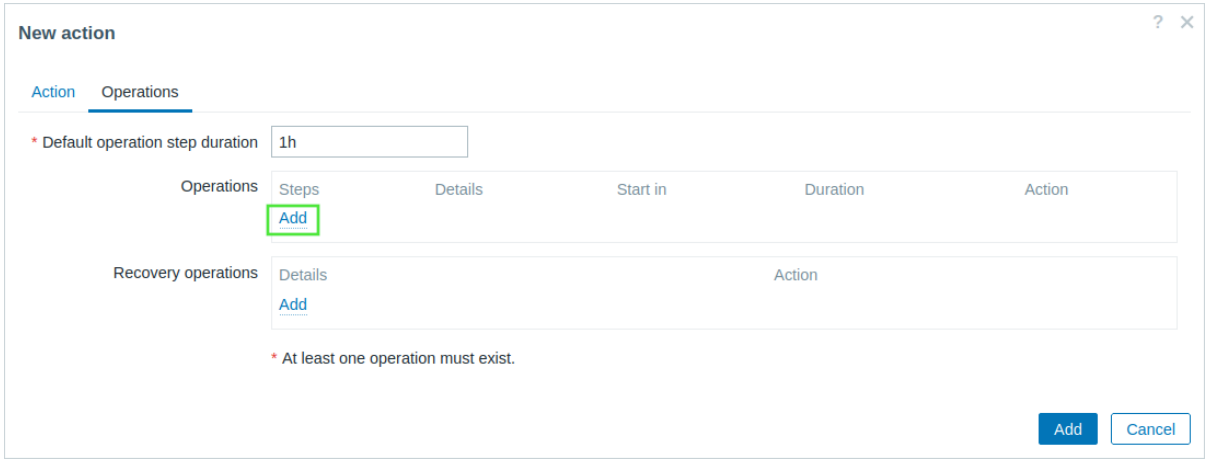
3. 新しい条件ポップアップウィンドウで条件の種類として「イベントタイプ」を選択し、イベントタイプとして「サポートされていない状態のアイテム」を選択します。



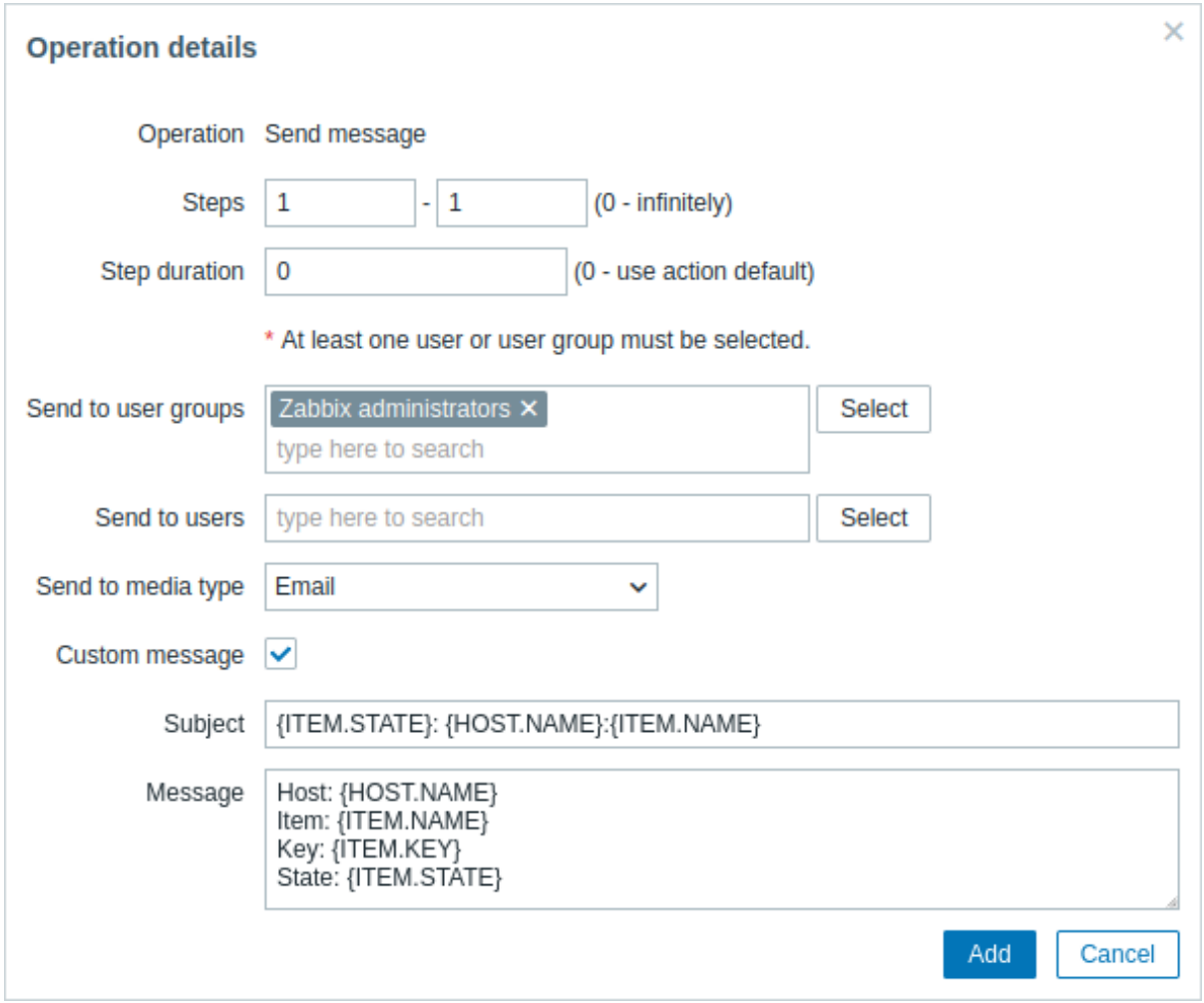
4. 追加をクリックして、条件ブロックに条件を実際に一覧表示します。

ステップ 4

1. オペレーションタブで、オペレーションブロックの 追加をクリックして、新しいオペレーションを追加します。



2. メッセージの受信者（ユーザーグループ/ユーザー）と、配信に使用するメディアタイプ（または「すべて」）を選択します。
3. 障害メッセージのカスタム件名/内容を入力する場合は、カスタムメッセージチェックボックスをオンにします。



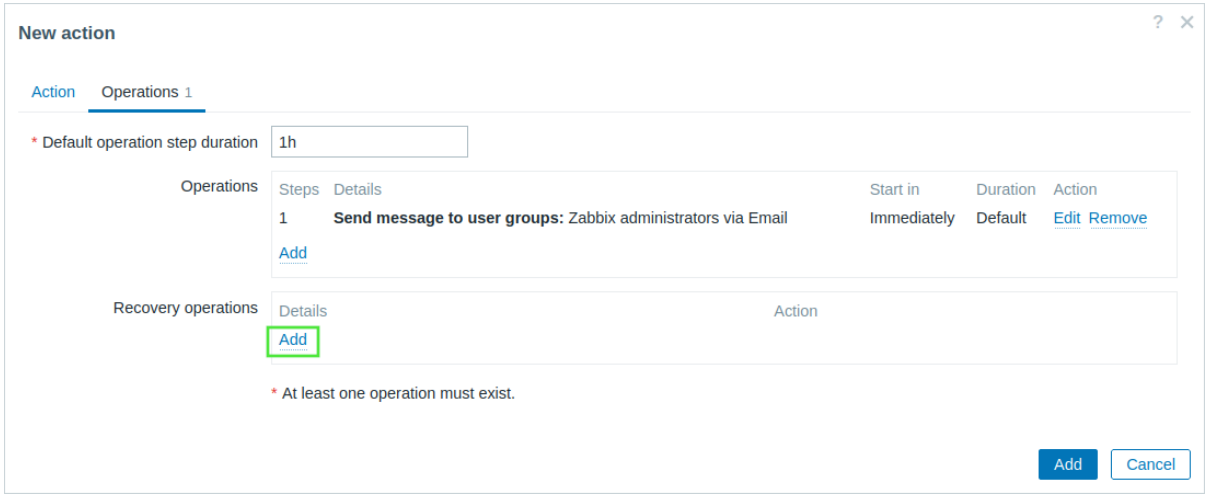
4. 追加をクリックして、オペレーションブロックにオペレーションを実際に追加します。

複数の通知を受信する場合は、オペレーションのステップ期間（メッセージ送信間隔）を設定し、別のステップを追加します。

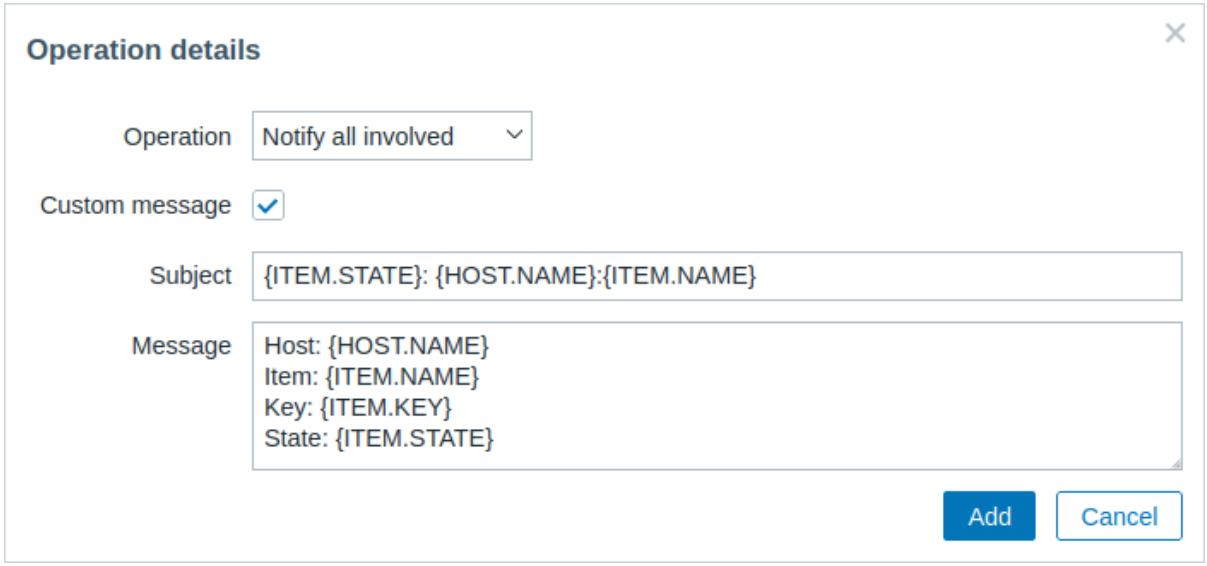
手順 5

復旧時の実行内容ブロックでは、アイテムが正常な状態に戻ったときの復旧通知を設定できます。

1. 復旧時の実行内容ブロックで 追加をクリックし、新しい復旧時の実行内容を追加します。



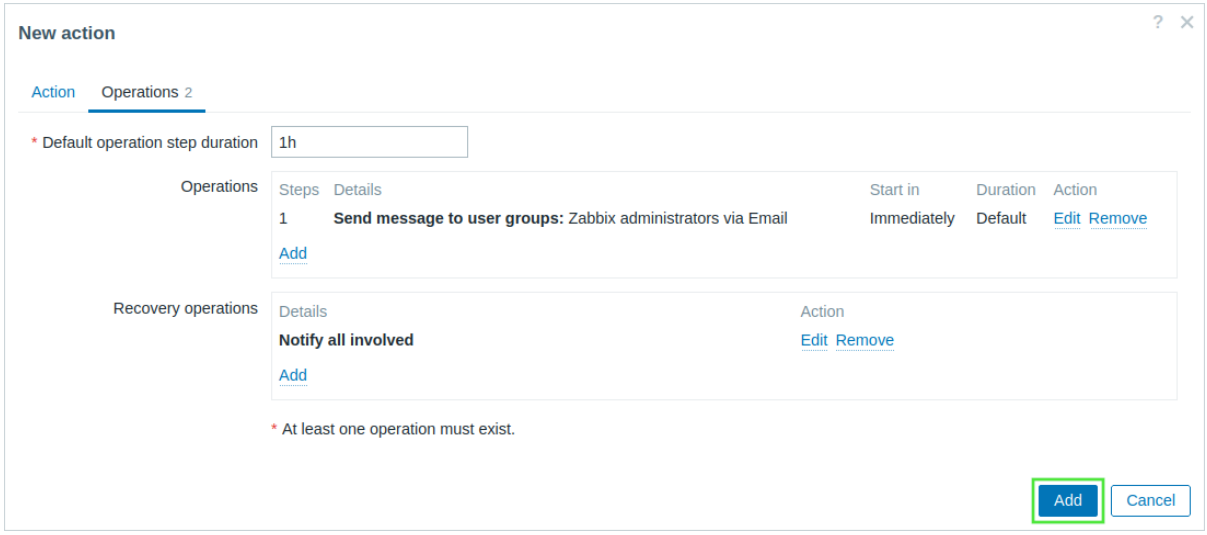
2. 実行内容のタイプとして「関係者全員に通知」を選択します。
3. 復旧メッセージの件名や本文をカスタマイズする場合は、カスタムメッセージチェックボックスを選択します。



4. 実行内容の詳細ポップアップウィンドウで 追加をクリックし、復旧時の実行内容ブロックに実行内容を実際に追加します。

ステップ 6

完了したら、フォーム下部の *Add* ボタンをクリックします。



これで完了です。設定は以上です。これで、いずれかのアイテムがサポート対象外になった場合に、Zabbix から最初の通知を受け取れるようになります。

マクロ

概要

Zabbix は、さまざまな状況で使用できる多数の組み込みマクロをサポートしています。これらのマクロは、特定の構文で識別される変数です。

`{MACRO}`

マクロは、コンテキストに応じて特定の値に解決されます。

マクロを効果的に使用することで、時間を節約し、Zabbix の設定をより分かりやすくできます。

典型的な使用例の 1 つとして、テンプレートでマクロを使用できます。たとえば、テンプレート上のトリガーに「Processor load is too high on {HOST.NAME}」という名前を付けることができます。テンプレートを Zabbix サーバーなどのホストに適用すると、Monitoring セクションでトリガーを表示した際に、名前は「Processor load is too high on Zabbix server」に解決されます。

マクロはアイテムキーのパラメータで使用できます。パラメータの一部だけにマクロを使用することもできます。たとえば、`item.key[server_{HOST.HOST}_local]` のように指定します。解決されたマクロに曖昧な特殊記号が含まれている場合でも、Zabbix が処理するため、パラメータを二重引用符で囲む必要はありません。

Zabbix には、その他の種類のマクロもあります。

Zabbix は、次のマクロをサポートしています。

- {MACRO} - 組み込みマクロ（完全な一覧を参照）
- {<macro>.<func>(<params>)} - マクロ関数
- {$MACRO} - ユーザーマクロ、オプションでコンテキスト付き
- {#MACRO} - ローレベルディスカバリ用のマクロ
- {?EXPRESSION} - 式マクロ

マクロ関数

概要

マクロ関数を使用すると、マクロの値をカスタマイズできます（たとえば、短縮したり、特定の部分文字列を抽出したりできます）。これにより、扱いやすくなります。

マクロ関数の構文は次のとおりです。

```
{macro.func(params)}
```

ここで、

- **macro** - カスタマイズするマクロ;
- **func** - 適用する関数（サポートされる関数を参照）;
- **params** - 関数パラメーターのカンマ区切りリスト。次の場合は 二重引用符で囲む必要があります:
  - スペースまたは二重引用符で始まる;
  - 閉じ括弧またはカンマを含む。

例:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

マクロ関数は、次のマクロでサポートされています。

- 組み込みマクロ
- ユーザーマクロ
- 低レベルディスカバリマクロ
- 式マクロ

マクロ関数は、列挙したマクロをサポートするすべての場所で使用できます。ただし、ホストマクロの設定時や低レベルディスカバリルールのフィルターのように、明示的にマクロのみが期待される場合はこの限りではありません。

1 つのマクロにつき 1 つの関数のみがサポートされており、複数のマクロ関数を連結することはできません。

> **Note:**
> マクロ関数が関数、アイテムキー、別のマクロなど、他のコンテキスト内で使用される場合の例については、エスケープの例を参照してください。

サポートされている関数

関数は追加情報なしで一覧表示されています。
関数をクリックすると、詳細を確認できます。

関数	説明
btoa	マクロ値を Base64 エンコードに変換します。
fmtnum	小数点以下に表示する桁数を制御するための数値フォーマット。
fmttime	時刻のフォーマット。
htmldecode	HTML エンコードされたマクロ値をデコードします。
htmlencode	マクロ値を HTML エンコードに変換します。
iregsub	正規表現の一致による部分文字列の抽出（大文字と小文字を区別しない）。
jsonpath	JSONPath の結果を返します。
lowercase	マクロ値の文字を小文字に変換します。
regrepl	マクロ値内の文字または部分文字列を置換します。
regsub	正規表現の一致による部分文字列の抽出（大文字と小文字を区別する）。

関数	説明
tr	マクロ値の文字を音訳します。
uppercase	マクロ値の文字を大文字に変換します。
urldecode	URL エンコードされたマクロ値をデコードします。
urlencode	マクロ値を URL エンコードに変換します。
xmlxpath	XML XPath の結果を返します。

関数の詳細

オプションの関数パラメータは < > で示されます。

btoa

マクロ値を Base64 エンコーディングにエンコードします。Base64 エンコーディングは、バイナリデータをテキストとして表現する方法であり、テキストベースのプロトコルでバイナリコンテンツを保存および安全に転送するのに便利です。

例:

```
{{ITEM.VALUE}.btoa()} - "zabbix" のような値を Base64 で"emFiYml4" にエンコードします
```

fmtnum(digits)

小数点以下の桁数を制御するための数値フォーマット。

パラメータ:

- **digits** - 小数点以下の桁数。有効範囲: 0-20。末尾のゼロは出力されません。

例:

```
{{ITEM.VALUE}.fmtnum(2)} - 受信値 "24.3483523" から "24.35" を返します
{{ITEM.VALUE}.fmtnum(0)} - 受信値 "24.3483523" から "24" を返します
```

fmttime(format,<time_shift>)

時刻の書式設定。
 この関数は、次のいずれかの時刻形式の値に解決されるマクロで使用できます。

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (ISO8601 標準)
- UNIX タイムスタンプ

パラメーター:

- **format** - 必須の書式文字列。`strftime` 関数の書式指定と互換性があります。
- **time_shift** (任意) - 書式設定前に時刻へ適用する時間シフト。`-<N><time_unit>` または `+<N><time_unit>` で始める必要があります。ここで:
  - `N` - 加算または減算する時間単位の数;
  - `time_unit` - h (時間), d (日), w (週), M (月), y (年)。

コメント:

- `time_shift` パラメーターは複数段階の時刻操作をサポートし、時間単位の先頭へシフトするための `/<time_unit>` を含めることができます (`/d` - 深夜、`/w` - 週の 1 日目 (月曜日)、`/M` - 月の 1 日目、など)。例: `-1w` - ちょうど 7 日前; `-1w/w` - 前週の月曜日; `-1w/w+1d` - 前週の火曜日。
- 時刻操作は優先順位なしで左から右へ計算されます。たとえば、`-1M/d+1h/w` は `((-1M/d)+1h)/w` として解釈されます。

例:

```
{{TIMESTAMP}.fmttime(%B)} - 受信した値 `1633098961` から "October" を返します
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} - 受信した値 `1633098961` から "1 September" を返します
{{EVENT.TIMESTAMP}.fmttime("%x %X")} - ロケールに応じた日付と時刻の表現を返します。たとえば、イベントのタイムスタンプ `1
{{EVENT.TIME}.fmttime("%x %X")} - イベントに関連付けられたタイムスタンプ `1784100339` に対して、ロケールに応じた日付と
```

htmldecode

マクロ値を HTML エンコードからデコードします。

以下の文字がサポートされています:

値	デコード後の値
&	&
<	

値	デコード後の値
>	>
"	"
'	'
'	'

例:

```
{{ITEM.VALUE}.htmldecode()} - "&lt;" のような値を HTML デコードして"<" にします
```

htmlencode

マクロ値を HTML エンコーディングに変換します。

以下の文字がサポートされています:

値	エンコード値
&	& <
>	>
"	"
'	'

例:

```
{{ITEM.VALUE}.htmlencode()} - "<" のような文字を "&lt;" に HTML エンコードします
```

iregsub(pattern,output)

正規表現 (大文字・小文字を区別しない) による部分文字列の抽出。

パラメータ:

- **pattern** - マッチさせる正規表現
- **output** - 出力オプション。**\1 - \9** プレースホルダーはキャプチャグループに対応。**\0** はマッチしたテキストを返します。

コメント:

- 正規表現にマッチしない場合、関数は空文字列を返します。
- 関数パターンが不正な正規表現の場合、マクロは「UNKNOWN」と評価されます (ただし、ローレベルディスカバリーマクロの場合は関数が無視され、マクロは未解決のままになります)。
- 置換文字列内で存在しないキャプチャグループへの参照は空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - "fail"、"error"、"fault"、"problem" のいずれかの
```

jsonpath(pattern,<default>)

JSONPath の結果を返します。
 サポートされている値のタイプ: *String*、*Text*、*Log*。

パラメータ:

- **pattern** - 必須のパターン;
- **default** - JSONPath クエリがデータを返さない場合のオプションのフォールバック値。その他のエラーが発生した場合は、失敗が返されます。

コメント:

- 次の場合、関数はエラーで失敗します:
  - JSON が無効な場合。
  - JSONPath クエリが失敗した場合。
  - JSONPath の結果にデータが含まれておらず、`default` パラメータが指定されていない場合。

例:

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

lowercase

すべてのマクロ値の文字を小文字に変換します。シングルバイト文字セット（ASCII など）で動作し、UTF-8 はサポートしていません。

例:

```
{{ITEM.VALUE}.lowercase()} - "Zabbix SERVER" のような値を"zabbix server"（小文字）に変換します
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

マクロ値内の文字/サブストリングを置換します。

パラメーター:

- **pattern** - 一致させる正規表現。
- **replacement** - 置換文字列。**\1 - \9** のプレースホルダーは、置換文字列内でキャプチャグループを参照するために使用できます。

コメント:

- pattern に一致するすべての出現箇所が置換されます。
- pattern と replacement は順番に処理され、各後続のペアは前の置換結果に従って適用されます。
- 置換文字列内で存在しないキャプチャグループを参照した場合は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - "oldParam" を "newParam" に置換します
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - 英字以外のすべての文字はバックスラッシュでエスケープされます
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - 末尾のバックスラッシュを削除します（たとえば、"C:\" を "C:" に置
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - アイテム値内の複数の部分を置換します
```

regsub(pattern,output)

正規表現マッチによる部分文字列の抽出（大文字と小文字を区別）。

パラメータ:

- **pattern** - マッチさせる正規表現
- **output** - 出力オプション。**\1 - \9** プレースホルダーはキャプチャグループに対応。**\0** はマッチしたテキストを返します。

コメント:

- 正規表現にマッチしない場合、関数は空文字列を返します。
- 関数パターンが不正な正規表現の場合、マクロは'UNKNOWN' に評価されます（ただし、ローレベルディスカバリーマクロの場合は関数が無視され、マクロは未解決のままになります）。
- 置換文字列で存在しないキャプチャグループへの参照は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - "123 Log line" のような値を受信した場合、"Problem ID: 123"
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - "fail"、"error"、"fault"、"problem" のいずれかの部
```

追加の例も参照してください。

tr(characters,replacement)

マクロ値の文字の音訳。

- **characters** - 置換する文字のセット
- **replacement** - 位置的に対応する置換文字のセット

例:

```
{{ITEM.VALUE}.tr(abc, xyz)} - "a" を"x"、"b" を"y"、"c" を"z" にすべて置換します
{{ITEM.VALUE}.tr(abc, xyzq)} - "a" を"x"、"b" を"y"、"c" を"z" にすべて置換します（"q" は無視されます）
{{ITEM.VALUE}.tr(abcde, xyz)} - "a" を"x"、"b" を"y"、"c" を"z"、"d" を"z"、"e" を"z" にすべて置換します（つまり x
{{ITEM.VALUE}.tr("\\'", "\/\"")} - バックスラッシュをスラッシュに、シングルクォートをダブルクォートにすべて置換します
{{ITEM.VALUE}.tr(A-Z,a-z)} - すべての文字を小文字に変換します
{{ITEM.VALUE}.tr(0-9a-z,*)} - すべての数字と小文字を"*" に置換します
{{ITEM.VALUE}.tr(0-9,ab)} - 0 を"a" に、1,2,3,4,5,6,7,8,9 を"b" にすべて置換します
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - すべての数字、"abc" 文字、A-L 範囲を"*" に置換します
{{ITEM.VALUE}.tr("\n","*")} - 行末を * に置換します
{{ITEM.VALUE}.tr("e", "\n")} - すべての"e" を行末に置換します
```

リテラル文字を含めるには:

```
バックスラッシュ - \\でエスケープする必要があります
シングルクォート - \'でエスケープする必要があります
ダブルクォート - \" でエスケープする必要があります
```

バックスラッシュによるサポートされるエスケープシーケンス:

```
\\\\ => \\ - ダブルバックスラッシュをシングルバックスラッシュに
\\a  => \a - アラート
\\b  => \b - バックスペース
\\f  => \f - フォームフィード
\\n  => \n - 改行
\\r  => \r - リターン
\\t  => \t - 水平タブ
\\v  => \v - 垂直タブ
```

uppercase

すべてのマクロ値の文字を大文字に変換します。シングルバイト文字セット (ASCII など) で動作し、UTF-8 はサポートしていません。

例:

```
{{ITEM.VALUE}.uppercase()} - "Zabbix Server" のような値を"ZABBIX SERVER"(大文字) に変換します
```

urldecode

マクロ値を URL エンコードからデコードします。

例:

```
{{ITEM.VALUE}.urldecode()} - "%2F" のような値を"/" に URL デコードします
```

urlencode

マクロ値を URL エンコーディングに変換します。

例:

```
{{ITEM.VALUE}.urlencode()} - "/" のような文字を"%2F" に URL エンコードします
```

xmlxpath(pattern,<default>)

XML XPath の結果を返します。
 サポートされている値のタイプ: *String*、*Text*、*Log*。

パラメータ:

- **pattern** - 必須のパターン;
- **default** - XPath クエリがデータを返さない場合のオプションのフォールバック値。その他のエラーが発生した場合は、失敗が返されます。

コメント:

- 次の場合、関数はエラーで失敗します。
  - XML が無効である。
  - XPath クエリが失敗する。
  - XPath の結果にデータが含まれておらず、default パラメータが指定されていない。
  - Zabbix が XML サポートなしでコンパイルされている。

例:

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

追加例

以下の表は、マクロ関数の使用例をさらに示しています。

{#IFALIAS} はLLD マクロであり、低レベルディスカバリのコンテキスト（ディスカバリルール、プロトタイプ、およびそれらから作成されたアイテム/トリガー）でのみ定義されます。LLD 以外で使用すると、トークンは展開されません。

マクロ関数	受信値	出力
{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}	123Log line	Problem
{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}	123 Log line	Problem

マクロ関数	受信値	出力
{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}	Log line	Problem ID:
{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}	MySQL crashed errno 123	Problem ID: MySQL_123
{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}	123 Log line	*UNKNOWN*（無効な正規表現）
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}	customername_1	customername
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}	customername_11	
{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}	customername_1	{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}（無効な正規表現）
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"customername"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \2)}"}	customername_1	{$MACRO:"1"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}（無効な正規表現）
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"	customername_1	"{$MACRO:"\customername\"}"
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \2)}\"}"	customername_1	"{$MACRO:"\1\"}"
"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+", \1)}\"}"	customername_1	"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}"（無効な正規表現）

アイテム値全体の表示

テキスト/ログアイテムの解決された {ITEM.VALUE} および {ITEM.LASTVALUE} マクロの長い値は、一部のフロントエンドの場所で 20 文字に切り捨てられます。これらのマクロの完全な値を表示するには、以下のようなマクロ関数を使用できます。

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

詳細は、{ITEM.VALUE} および {ITEM.LASTVALUE} のマクロ詳細も参照してください。

ユーザーマクロ

概要

Zabbix では、標準でサポートされているマクロに加えて、柔軟性を高めるためにユーザーマクロがサポートされています。

ユーザーマクロは、グローバル、テンプレート、ホストの各レベルで定義できます。これらのマクロには、次の特殊な構文があります。

{$MACRO}

Zabbix は、次の優先順位に従ってマクロを解決します。

1. ホストレベルのマクロ（最初に確認）。
2. ホストの第 1 レベルのテンプレート（ホストに直接リンクされているテンプレート）に定義されたマクロ。テンプレート ID 順に確認されます。
3. ホストの第 2 レベルのテンプレートに定義されたマクロ。テンプレート ID 順に確認されます。
4. ホストの第 3 レベルのテンプレートに定義されたマクロ。テンプレート ID 順に確認されます。以降も同様です。
5. グローバルマクロ（最後に確認）。

つまり、ホストにマクロが存在しない場合、Zabbix はテンプレートの階層を順に深くたどって、ホストのテンプレート内からマクロを検索します。それでも見つからない場合、グローバルマクロが存在すれば、それが使用されます。

> **Warning:**
> 同じレベルの複数のリンクされたテンプレートに同じ名前のマクロが存在する場合、ID が最も小さいテンプレートのマクロが使用されます。そのため、複数のテンプレートで同じ名前のマクロを使用することは、設定上のリスクになります。

Zabbix がマクロを見つけられない場合、そのマクロは解決されません。

Attention:
マクロ（ユーザーマクロを含む）は、複雑な設定をより分かりやすくするため、仕様上、設定セクション（たとえば、トリガーの一覧）では未解決のまま表示されます。

ユーザーマクロは、次の場所で使用できます。

- アイテム名
- アイテムキーのパラメータ
- アイテムの更新間隔および柔軟な間隔
- トリガー名および説明
- トリガーの式のパラメータおよび定数（例を参照）
- その他多数の場所 - 完全な一覧を参照してください。

グローバルマクロとホストマクロの一般的な使用例

- グローバルマクロを複数の場所で使用し、マクロの値を変更すると、ワンクリックですべての場所に設定変更が適用されます。
- ホスト固有の属性（パスワード、ポート番号、ファイル名、正規表現など）を持つテンプレートを活用します。

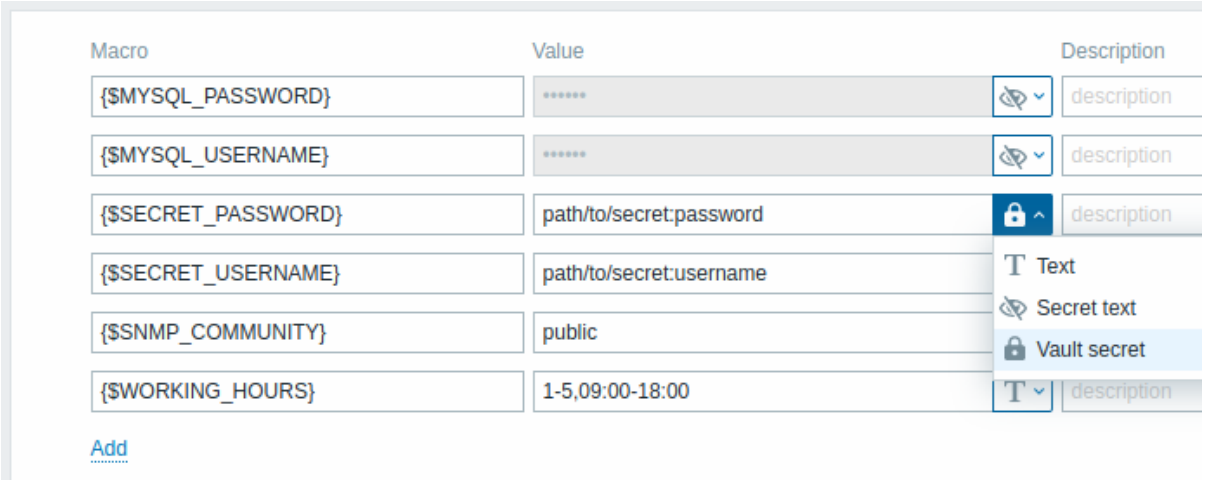
Note:
グローバルマクロの追加、更新、削除を行うと、すべてのホストに対して増分設定更新が強制されるため、グローバルマクロの代わりにホストマクロを使用することを推奨します。詳細については、Passive and active agent checksを参照してください。

設定

ユーザーマクロを定義するには、Web インターフェースの該当する場所に移動します。

- グローバルマクロの場合は、*Administration* > *Macros* を開きます。
- ホストおよびテンプレートレベルのマクロの場合は、ホストまたはテンプレートのプロパティを開き、*Macros* タブを確認します。

ユーザーマクロには、次の属性があります。



Parameter	Description
Macro	マクロ名。名前は中括弧で囲み、ドル記号で始める必要があります。 例: {$FRONTEND_URL}。マクロ名に使用できる文字は、**A-Z**（大文字のみ）、**0-9**、_、**.** です。
Value	マクロの値。3 種類の値タイプがサポートされています: **Text**（デフォルト）- プレーンテキストの値 **Secret text** - 値はアスタリスクでマスクされます **Vault secret** - 値にはvault secret へのパス/クエリが含まれます。 値タイプを変更するには、値入力フィールドの末尾にあるボタンをクリックします。 ユーザーマクロ値の最大長は 2048 文字です。
Description	このマクロに関する追加情報を記載するためのテキストフィールドです。

ユーザーマクロを設定する際は、コンテキストに応じて次の動作に注意してください。

- ユーザーマクロがテンプレートのアイテムやトリガーで使用される場合は、それらのマクロをテンプレートにも追加することを検討してください（グローバルに定義されている場合でも同様です）。そうすることで、*Text* タイプのマクロは、テンプレートを XML にエクスポートして別のシステムにインポートした後も、期待どおりに動作します（secret macro の値はエクスポートされません）。
- ユーザーマクロがトリガー式で使用される場合、それらのマクロはパラメータまたは定数を参照している場合にのみ解決されます。ホスト、アイテムキー、関数、演算子、または別のトリガー式を参照している場合は解決されません（secret macro はトリガー式では使用できません）。
- ユーザーマクロが、ホストプロトタイプを持つ低レベルディスカバリルールがあるホストで使用される場合、検出されたホスト は、そのホストで定義されたすべてのユーザーマクロを継承します。

例

例 1

「SSH デーモンのステータス」アイテムキーでのホストレベルマクロの使用:

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

このアイテムは、複数のホストに割り当てることができます。ただし、それらのホストで {$SSH_PORT} の値が定義されている必要があります。

例 2

「CPU load is too high」トリガーでのホストレベルマクロの使用:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

このようなトリガーはテンプレート上で作成し、個々のホストで編集することはありません。

Note:
値の数を関数パラメータとして使用する場合（例: max(/host/key,#3)）、次のようにマクロ定義にハッシュ記号を含めます: SOME_PERIOD => #3。

例 3

「CPU load is too high」トリガーでの 2 つのマクロの使用:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

マクロはトリガー関数のパラメータとして使用できます。この例では min() 関数が該当します。

例 4

エージェントが利用できない状態の条件を、アイテムの更新間隔と同期します。

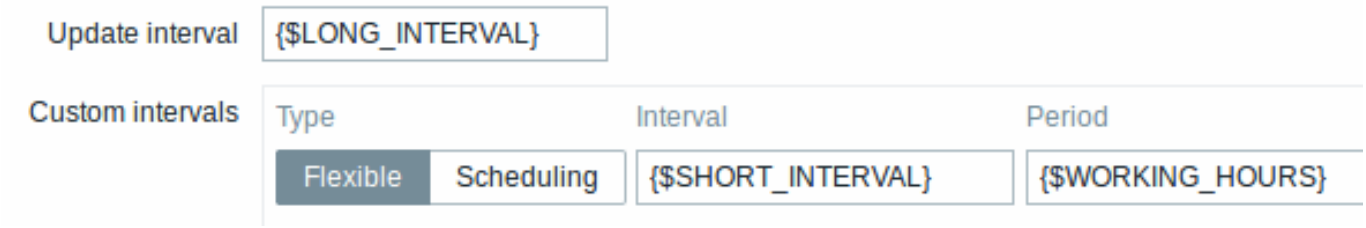
- {$INTERVAL} マクロを定義し、アイテムの更新間隔で使用する
- {$INTERVAL} をエージェントが利用できない状態のトリガーのパラメータとして使用する

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

例 5

稼働時間の設定を一元化します。

- {$WORKING_HOURS} マクロを作成し、1-5,09:00-18:00 と設定します。
- 管理 > 一般設定 > *GUI* の 稼働時間フィールドで使用します。
- ユーザーの ユーザー > ユーザーの メディアタブにある 有効な時間帯フィールドで使用します。
- 稼働時間中のアイテムのポーリング頻度を高くするために使用します。

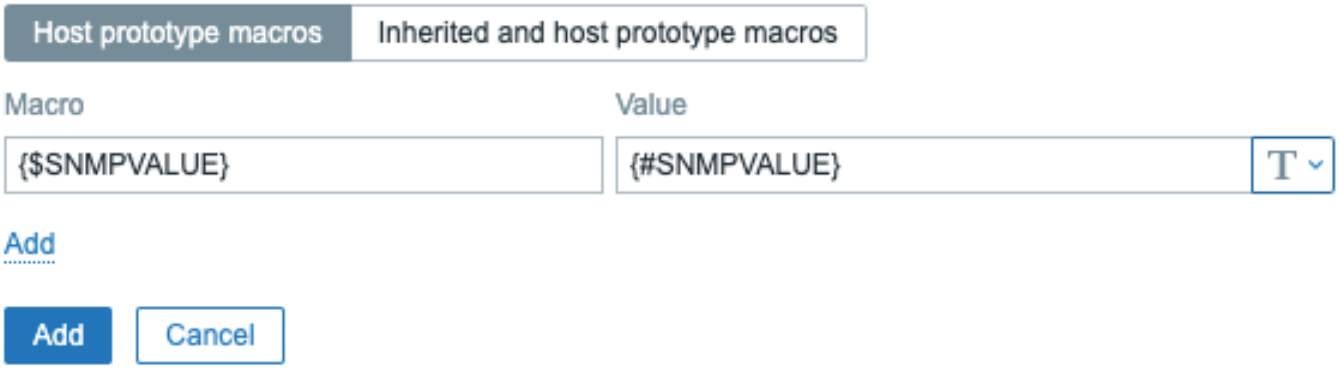


- アクションの条件の 期間で使用します。
- 必要に応じて、管理 > マクロで稼働時間を調整します。

例 6

ホストプロトタイプマクロを使用して、ディスカバリされたホストのアイテムを設定します。

1. ホストプロトタイプで、ユーザーマクロ {$SNMPVALUE} を定義し、値として {#SNMPVALUE} ローレベルディスカバリ マクロを指定します。



2. *Generic SNMPv2* テンプレートをホストプロトタイプに割り当てます。
3. *Generic SNMPv2* テンプレートのアイテムの *SNMP OID* フィールドで {$SNMPVALUE} を使用します。

ユーザーマクロのコンテキスト

コンテキスト付きユーザーマクロを参照してください。

コンテキスト付きユーザーマクロ

概要

user macroでは、オプションのコンテキストを使用できます。これにより、デフォルト値をコンテキスト固有の値で上書きできます。

コンテキストはマクロ名に追加します。構文は、コンテキストが静的なテキスト値かどうかによって異なります。

```
{$MACRO:"static text"}
```

または正規表現の場合は次のようになります。

```
{$MACRO:regex:"regular expression"}
```

正規表現コンテキストを持つマクロは、ユーザーマクロの設定でのみ定義できることに注意してください。トリガーの式など、ユーザーマクロのコンテキストとして別の場所で `regex:` プレフィックスを使用した場合は、静的コンテキストとして扱われます。

コンテキストの引用符は省略できます（重要な注意事項も参照）。

マクロコンテキストの例:

例	説明
{$LOW_SPACE_LIMIT}	コンテキストなしのユーザーマクロ。
{$LOW_SPACE_LIMIT:/tmp}	コンテキスト付きのユーザーマクロ（静的文字列）。
{$LOW_SPACE_LIMIT:regex:"^/tmp$"}	コンテキスト付きのユーザーマクロ（正規表現）。 {$LOW_SPACE_LIMIT:/tmp} と同じです。
{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}	コンテキスト付きのユーザーマクロ（正規表現）。/var/log/ で始まるすべての文字列に一致します。

ユースケース

コンテキスト付きのユーザーマクロを定義すると、トリガー式でより柔軟な閾値を実現できます（low-level discovery によって取得された値に基づく）。たとえば、次のマクロを定義できます。

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:”^/[a-z]+$”} = 30

このとき、マウントされたファイルシステムの discovery に対するトリガープロトタイプで、low-level discovery マクロをマクロコンテキストとして使用できます。

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

discovery の後は、検出されたマウントポイントやファイルシステムの種類に応じて、トリガーに異なる空き容量の閾値が適用されます。以下の場合に障害イベントが生成されます。

- /home フォルダの空きディスク容量が 20% 未満

- 正規表現パターンに一致するフォルダ（/etc、/tmp、/var など）の空きディスク容量が 30% 未満
- 正規表現パターンに一致せず、かつ /home でもないフォルダの空きディスク容量が 10% 未満

重要事項

- コンテキストを持つ複数のユーザーマクロが存在する場合、Zabbix は最初に単純なコンテキストマクロを照合し、次にコンテキストマクロを未定義の順序で正規表現と照合しようとします。

Warning:
未定義の動作を避けるために、同じ文字列に一致する別のコンテキストマクロを作成しないでください。

- コンテキストを持つマクロがホスト、リンクされたテンプレート、またはグローバルで見つからない場合、コンテキストのないマクロが検索されます。
- コンテキストでは、ローレベルディスカバリマクロのみがサポートされています。その他のマクロはすべて無視され、プレーンテキストとして扱われます。

技術的には、マクロコンテキストはitem key パラメータと同様のルールを使用して指定されますが、, 文字がある場合、マクロコンテキストは複数のパラメータとして解析されません。

- コンテキストに } 文字が含まれている場合、または " 文字で始まる場合、マクロコンテキストは " で引用する必要があります。引用されたコンテキスト内の引用符は、\ 文字でエスケープする必要があります。
- \ 文字自体はエスケープされません。つまり、\ 文字で終わる引用符付きコンテキストを持つことは不可能です - マクロ {$MACRO:"a:\b\c\"} は無効です。
- コンテキスト内の先頭のスペースは無視されますが、末尾のスペースは無視されます:
  - たとえば、{$MACRO:A} は {$MACRO: A} と同じですが、{$MACRO:A } とは異なります。
- 先頭の引用符の前と末尾の引用符の後のすべてのスペースは無視されますが、引用符内のすべてのスペースは無視されません。
  - マクロ {$MACRO:"A"}、{$MACRO: "A"}、{$MACRO:"A" }、および {$MACRO: "A" } は同じですが、マクロ {$MACRO:"A"} および {$MACRO:" A "} ではありません。

{$MACRO:A}、{$MACRO: A}、および {$MACRO:"A"} のマクロは、同じコンテキストを持っているため、すべて同等です。これは、'key[a]'、'key[a]'、および'key["a"]' が意味的に同じである項目キーとは対照的ですが、一意性のために異なります。

秘密ユーザーマクロ

概要

Zabbix は、ユーザーマクロ値の機密情報を保護するために 2 つのオプションを提供しています。

- シークレットテキスト
- Vault シークレット

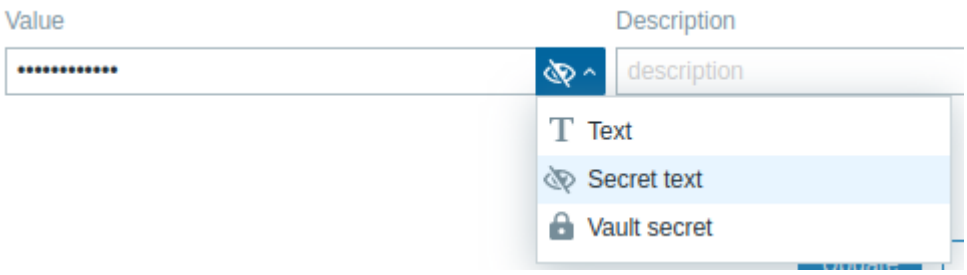
Note:
シークレットマクロの値は非表示ですが、アイテムで使用することで明らかになる場合があります。たとえば、外部スクリプトでシークレットマクロを参照する `echo` 文を使用すると、Zabbix サーバーが実際のマクロ値にアクセスできるため、マクロ値がフロントエンドに表示される場合があります。シークレットマクロ値がマスク解除される場所を参照してください。

シークレットマクロはトリガー式では使用できません。

Secret text

Secret text マクロでは、マクロ値はアスタリスクでマスクされます。

マクロ値を secret にするには、*Value* フィールドの末尾にあるボタンをクリックし、*Secret text* オプションを選択します。



設定を保存すると、値を表示できなくなります。

マクロ値を変更するには、*Value* フィールドにカーソルを合わせ、*Set new value* ボタンをクリックします（ホバー時に表示されます）。



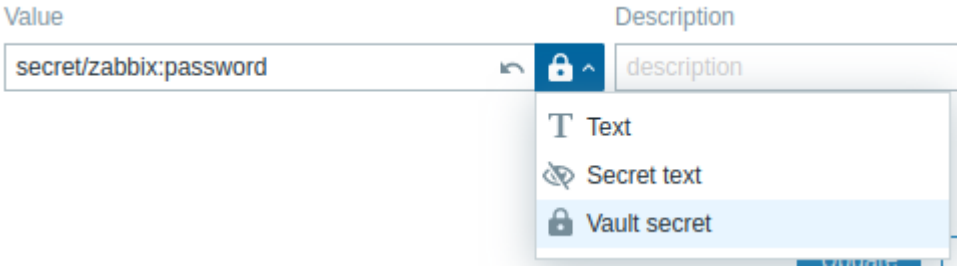
Set new value ボタンをクリックするか、マクロ値の種類を変更すると、現在の値は消去されます。元の値を復元するには、*Value* フィールドの末尾にある 矢印をクリックします（新しい設定を保存する前にのみ使用できます）。元の値を復元しても、その値が表示されることはありません。

Note:
secret マクロを含む URL は、そこに含まれるマクロが"******" として解決されるため、機能しません。

Vault secret

Vault secret マクロでは、マクロ値は外部のシークレット管理ソフトウェア（vault）に保存されます。

Vault secret マクロを設定するには、*Value* フィールドの末尾にあるボタンをクリックし、*Vault secret* オプションを選択します。



マクロ値は vault secret を指している必要があります。入力形式は vault プロバイダーによって異なります。プロバイダー固有の設定例については、次を参照してください。

- HashiCorp
- CyberArk

Vault シークレットマクロの値は、設定データが更新されるたびに、Zabbix サーバー（および、*Resolve secret vault macros by* が設定 *Zabbix server and proxy* の場合は Zabbix プロキシ）によって vault から取得され、その後設定キャッシュに保存されます。Zabbix サーバーと Zabbix プロキシは、異なる vault を使用する場合があります。

Resolve secret vault macros by が設定 *Zabbix server* の場合、vault シークレットはサーバーによってのみ取得され、Zabbix プロキシは各設定同期時に Zabbix サーバーから Vault シークレットマクロの値を受け取り、自身の設定キャッシュに保存します。これは、Zabbix プロキシが再起動後、Zabbix サーバーから設定更新を受信するまでデータ収集を開始できないことを意味します。

vault からシークレット値を手動で更新するには、`secrets_reload` ランタイム制御 オプション（サーバーのみ）を使用します。

Zabbix サーバーとプロキシ間では暗号化を有効にする必要があります。そうでない場合、サーバーの警告メッセージがログに記録されます。

Warning:
マクロ値を正常に取得できない場合、その値を使用している対応するアイテムは未サポートになります。

マスクされていない場所

この一覧は、秘密マクロ値がマスクされていないパラメーターの場所を示します。

Note:
秘密マクロ値は、間接的に参照されている場合、以下の場所でもマスクされたままになります。たとえば、メディアタイプ（Script または webhook パラメーター）で使用される {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} 組み込みマクロ は、`net.tcp.port[******,******]` のように、マスクされた秘密マクロを含むアイテムキーに解決されます。`net.tcp.port[192.0.2.1,80]` にはなりません。

Context		Parameter
アイテム、アイテムプロトタイプ、**LLD** ルール		
	アイテム	アイテムキーのパラメーター
	アイテムプロトタイプ	アイテムプロトタイプキーのパラメーター
	低レベルディスカバリルール	ディスカバリアイテムキーのパラメーター
	SNMP エージェント	*SNMP community*
		コンテキスト名 (SNMPv3)
		セキュリティ名 (SNMPv3)

Context		Parameter
		認証パスフレーズ (SNMPv3)
		プライバシーパスフレーズ (SNMPv3)
	HTTP エージェント	*URL*
		クエリフィールド
		リクエスト本文
		ヘッダー
		ユーザー名
		パスワード
		SSL キーパスワード
	Script	パラメーター
		スクリプト
	Browser	パラメーター
		スクリプト
	Database monitor	*SQL* クエリ
	TELNET エージェント	スクリプト
		ユーザー名
		パスワード
	SSH エージェント	スクリプト
		ユーザー名
		パスワード
	Simple check	ユーザー名
		パスワード
	JMX エージェント	ユーザー名
		パスワード
アイテム値の前処理		
	JavaScript 前処理ステップ	スクリプト
Web シナリオ		
	Web シナリオ	変数値
		ヘッダー値
		URL
		クエリフィールド値
		POST フィールド値
		生の *POST*
	Web シナリオ認証	ユーザー
		パスワード
		SSL キーパスワード
コネクタ		
	コネクタ	*URL*
		ユーザー名
		パスワード
		トークン
		HTTP プロキシ
		SSL 証明書ファイル
		SSL キーファイル
		SSL キーパスワード
ネットワークディスカバリ		
	SNMP	*SNMP community*
		コンテキスト名 (SNMPv3)
		セキュリティ名 (SNMPv3)
		認証パスフレーズ (SNMPv3)
		プライバシーパスフレーズ (SNMPv3)
グローバルスクリプト		
	webhook	*JavaScript* スクリプト
		JavaScript スクリプトパラメーター値
	Telnet	ユーザー名
		パスワード
	SSH	ユーザー名
		パスワード
	Script	スクリプト
メディアタイプ		
	Script	スクリプトパラメーター

Context		Parameter
IPMI 管理	Webhook	パラメーター
	ホスト	ユーザー名 パスワード

ローレベルディスカバリマクロ

概要

ローレベルディスカバリ (LLD) 機能内で使用されるマクロの種類があります:

```
{#MACRO}
```

これは LLD ルールで使用されるマクロで、ファイルシステム名、ネットワークインターフェース、SNMP OID などの実際の値を返します。

これらのマクロは、エンティティのプロトタイプ（アイテム、トリガー、グラフ、LLD ルール、ホスト、ホストグループのプロトタイプ）の作成に使用できます。
実際のファイルシステム、ネットワークインターフェース、仮想マシン などのディスカバリ時に、これらのマクロは実際の値に置き換えられ、実際のエンティティを作成するための基礎となります。

一部のローレベルディスカバリマクロは、Zabbix の LLD 機能にあらかじめ含まれています - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}。
ただし、カスタム ローレベルディスカバリルールを作成する際に、これらの名前に従う必要はありません。
その場合は、他の任意の LLD マクロ名を使用し、その名前を参照できます。

サポートされるデータ型

カスタムディスカバリルールを定義する場合、LLD マクロ用の JSON オブジェクトで返されるプロパティ値は、次のプリミティブ型のいずれかである必要があります。

- string
- number
- boolean

配列、オブジェクト、null 値はサポートされていません。このような値を参照する LLD マクロは展開されないままとなり、アイテムの前処理および作成時にそのままの文字列（例: '{#MY_MACRO}'）として表示されます。

サポートされる場所

LLD マクロは以下で使用できます:

- ローレベルディスカバリルールのフィルター内
- ネストされたローレベルディスカバリルール内の以下
  - JSONPath 前処理パラメータ
  - カスタム LLD マクロ用の JSONPath フィールド
- アイテムプロトタイプおよびディスカバリプロトタイプの以下
  - 名前
  - キーパラメータ
  - 単位
  - 更新間隔[1]
  - タイムアウト[1]
  - 履歴保存期間[1]
  - トレンド保存期間[1]
  - アイテム値の前処理ステップ
  - SNMP OID
  - IPMI センサーフィールド
  - 計算/集計アイテムの式。以下で使用可能:
    - 式の定数および関数パラメータ
    - アイテムのキーパラメータ
  - 集計アイテムのフィルター条件（ホストグループ名およびタグ名）
  - SSH スクリプトおよび Telnet スクリプト
  - データベース監視 SQL クエリ
  - JMX アイテムのエンドポイントフィールド
  - 説明
  - HTTP エージェントの URL フィールド

    - HTTP エージェントの HTTP クエリフィールドフィールド
    - HTTP エージェントのリクエストボディフィールド
    - HTTP エージェントの必要なステータスコードフィールド
    - HTTP エージェントのヘッダーフィールドのキーと値
    - HTTP エージェントの HTTP 認証ユーザー名フィールド
    - HTTP エージェントの HTTP 認証パスワードフィールド
    - HTTP エージェントの HTTP プロキシフィールド
    - HTTP エージェントの HTTP SSL 証明書ファイルフィールド
    - HTTP エージェントの HTTP SSL キーファイルフィールド
    - HTTP エージェントの HTTP SSL キーパスワードフィールド
    - タグ
- トリガープロトタイプの以下
    - 名前
    - 運用データ
    - 式（定数および関数パラメータ内のみ）
    - URL
    - 説明
    - タグ
- グラフプロトタイプの以下
    - 名前
- ホストプロトタイプの以下
    - 名前
    - 表示名
    - カスタムインターフェースフィールド: IP、DNS、ポート、SNMP v1/v2 コミュニティ、SNMP v3 コンテキスト名、SNMP v3 セキュリティ名、SNMP v3 認証パスフレーズ、SNMP v3 プライバシーパスフレーズ
    - ホストグループプロトタイプ名
    - ホストタグ値
    - ホストマクロ値
    - (完全な一覧を参照)

これらすべての場所では、ローレベルディスカバリルールのフィルターを除き、LLD マクロを静的ユーザーマクロコンテキスト内で使用できます。

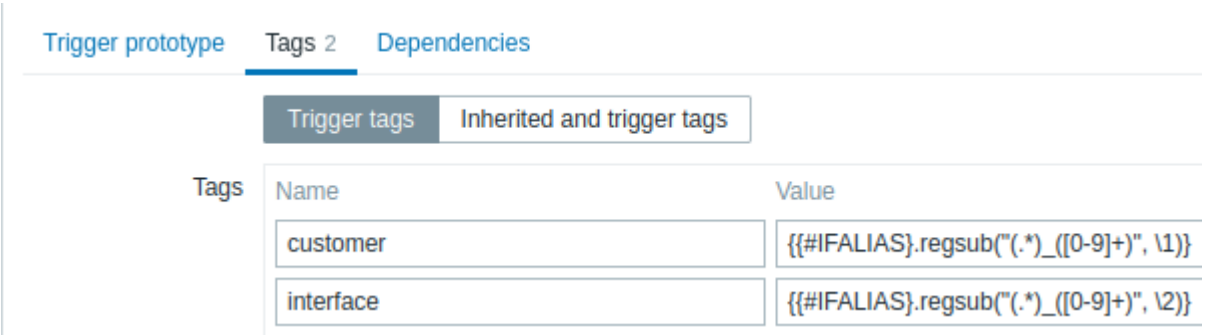
マクロ関数の使用

マクロ関数はローレベルディスカバリマクロでサポートされています（ローレベルディスカバリルールのフィルターを除く）。これにより、正規表現を使用してマクロ値の特定の部分を抽出できます。

たとえば、イベントタグ付けのために、次の LLD マクロから顧客名とインターフェース番号を抽出したい場合があります。

```
{#IFALIAS}=customername_1
```

これを行うには、トリガープロトタイプのイベントタグ値フィールドで、マクロとともに regsub マクロ関数を使用します。



引用符で囲まれていないアイテムのkey パラメータではコンマを使用できないため、マクロ関数を含むパラメータは引用符で囲む必要があります。パラメータ内の二重引用符をエスケープするには、バックスラッシュ（\）文字を使用してください。例:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

マクロ関数の構文の詳細については、マクロ関数を参照してください。

マクロ関数は Zabbix 4.0 以降、ローレベルディスカバリマクロでサポートされています。

キーパラメータのないアイテムのプロトタイプ

アイテムキーのパラメータを使用しない場合は、LLD マクロをitem keyのパラメータ括弧 [...] 内に配置して、パラメータとして扱われ、ディスカバリ時に置換されるようにします。

```
v_[{#MACRO}]
```

脚注

[1] [1] で示されたフィールドでは、単一のマクロがフィールド全体を埋める必要があります。1 つのフィールド内で複数のマクロを使用すること、またはマクロをテキストと混在させることはサポートされていません。

式マクロ

概要

Expression macro を使用すると、フィールド内で計算を実行できます。

その値は、まず内部のマクロを解決し、その後に結果の式を評価することで計算されます。

構文:

```
{?EXPRESSION}
```

EXPRESSION は、トリガー式と同じ構文を使用し、同じ関数をサポートします。

例:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Expression macro をサポートする場所については、サポートされるマクロ表で"{?EXPRESSION}" を検索してください。

使用上の注意:

- トリガー式/復旧式の関数値を参照するには、{FUNCTION.*} マクロを使用します。
- ホストおよびアイテムを参照するには、{HOST.HOST<1-9>} および{ITEM.KEY<1-9>} マクロを使用します。
- テンプレートでは、{HOST.HOST<1-9>} マクロを使用するか、最初のホストについてはホスト名を完全に省略します。たとえば {?avg(//item1,1h)} のようにします。テンプレート名はテンプレートのリンク 時にホスト名へ置換されないため、テンプレート名の代わりには使用しません。

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

イベント名で expression macro を使用する例については、トリガー式の例 も参照してください。

ユーザーとユーザーグループ

概要

Zabbix のすべてのユーザーは、Web ベースの Web インターフェースを通じて Zabbix アプリケーションにアクセスします。各ユーザーには、一意のログイン名とパスワードが割り当てられます。

すべてのユーザーパスワードは暗号化され、Zabbix データベースに保存されます。ユーザーは、UNIX 側でも適切に設定されていない限り、自分のユーザー ID とパスワードを使って UNIX サーバーに直接ログインすることはできません。Web サーバーとユーザーのブラウザー間の通信は、SSL を使用して保護できます。

柔軟なユーザー権限スキーマ を使用すると、次の権限を制限したり、区別したりできます。

- 管理用の Zabbix Web インターフェース機能へのアクセス。
- Web インターフェースでの特定の操作の実行。
- ホストグループ内の監視対象ホストへのアクセス。
- 特定の API メソッドの使用。

ユーザーの設定

概要

Zabbix を初期インストールすると、次の 2 つの定義済みユーザーが作成されます。

- *Admin* - すべての権限を持つ Zabbix のスーパーユーザー。
- *guest* - 特別な Zabbix のユーザー。

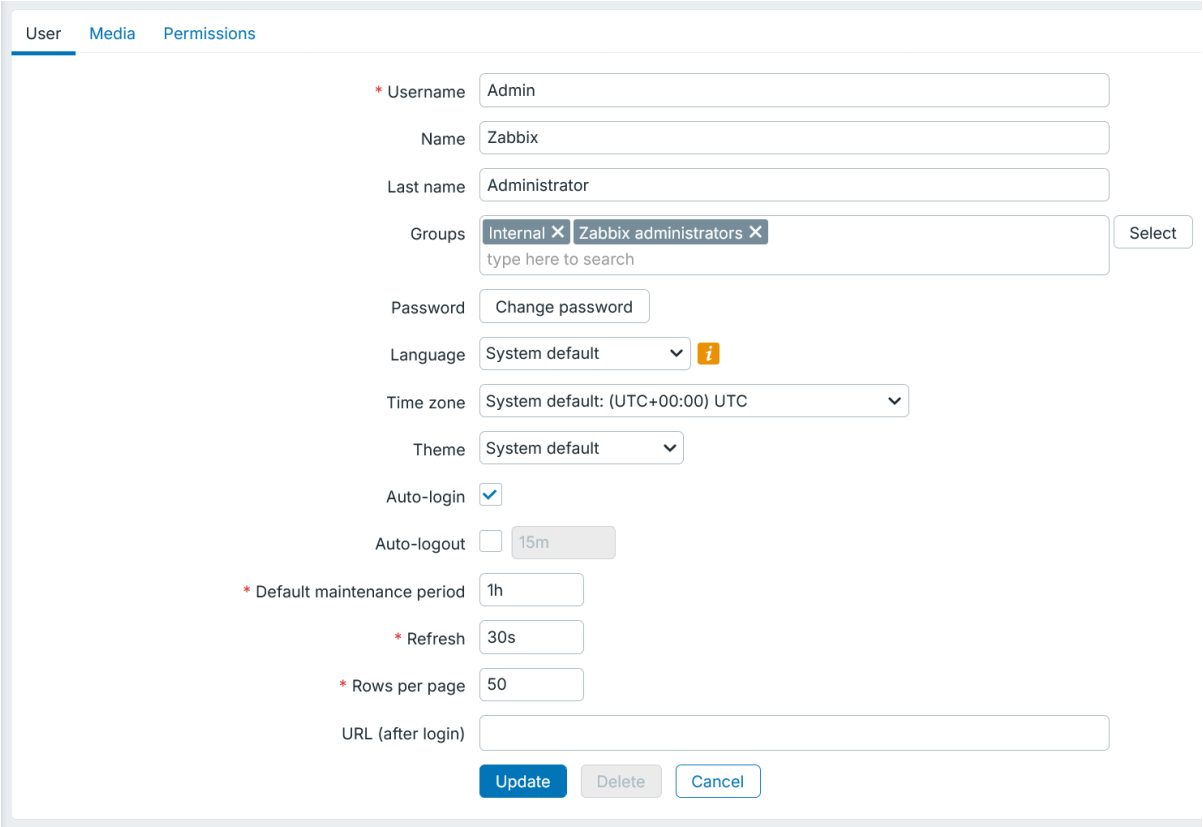
guest ユーザーは、デフォルトでは無効になっています。
このユーザーを有効にするには、ユーザーグループから定義済みの *Disabled* グループを削除します。
このユーザーでログインすると、Zabbix の監視ページにアクセスできます。
デフォルトでは、*guest* には Zabbix オブジェクトに対する権限がないことに注意してください。

ユーザーを設定するには、次の手順を実行します。

1. *Users* > *Users* に移動します。
2. *Create user* をクリックします（既存のユーザーを編集する場合は、ユーザー名をクリックします）。
3. フォームでユーザー属性を編集します。

一般属性

User タブには、ユーザーの一般属性が含まれます。



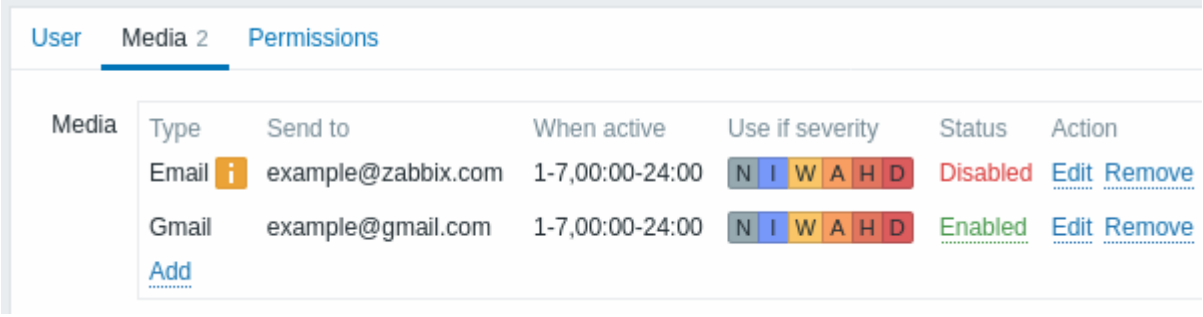
すべての必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
Username	ログイン名として使用する一意のユーザー名。
Name	ユーザーの名（任意）。 空でない場合、障害の確認情報および通知の受信者情報に表示されます。
Last name	ユーザーの姓（任意）。 空でない場合、障害の確認情報および通知の受信者情報に表示されます。
Groups	ユーザーが所属するユーザーグループを選択します。このフィールドはオートコンプリートに対応しているため、ユーザーグループ名の入力を開始すると、一致するグループがドロップダウンに表示されます。下にスクロールして選択します。または、*Select* をクリックしてグループを追加します。選択したグループを削除するには「x」をクリックします。 ユーザーグループへの所属によって、ユーザーがアクセスできるホストグループおよびホストが決まります。
Password	ユーザーパスワードを入力する 2 つのフィールド、またはユーザーがすでに存在する場合は *Change password* ボタンが表示されます。 *Change password* ボタンをクリックすると、新しいパスワードを入力する 2 つのフィールドが開きます。 *Super admin role* のユーザーが自分のパスワードを変更する場合、*Change password* ボタンをクリックすると、現在の（古い）パスワードを入力する追加フィールドが開きます。 パスワードの変更に成功すると、パスワードを変更したユーザーは、すべてのアクティブなセッションからログアウトされます。 パスワードを変更できるのは、Zabbix の内部認証を使用しているユーザーのみです。

パラメータ	説明
Language	Zabbix Web インターフェースの言語。 翻訳を機能させるには、php gettext 拡張機能が必要です。
Time zone	ユーザーレベルでグローバルタイムゾーンを上書きするタイムゾーンを選択するか、グローバルなタイムゾーン設定を使用する場合は **System default** を選択します。
Theme	Web インターフェースの外観を定義します。 **System default** - システムのデフォルト設定を使用 **Blue** - 標準の青いテーマ **Blue (classic)** - 従来のフォントを使用する標準の青いテーマ **Dark** - 代替のダークテーマ **Dark blue** - ダークブルーのテーマ **Dark (classic)** - 従来のフォントを使用する代替のダークテーマ **High-contrast light** - 高コントラストのライトテーマ **High-contrast dark** - 高コントラストのダークテーマ
Auto-login	このチェックボックスを選択すると、Zabbix がユーザーを記憶し、30 日間自動的にログインします。 *Remember for 30 days*を使用してログインした場合： - 自動ログアウトがリセットされます（セッションは 30 日間維持されます）。 - シームレスな再認証のために自動ログインが有効になります。 *Remember for 30 days* を使用せずにログインした場合： - シームレスな再認証のために自動ログインが有効になります。 - 自動ログアウトは標準のタイムアウト設定に従います。 この機能にはブラウザーの Cookie が使用されます。
Auto-logout	このチェックボックスを選択すると、設定した秒数（最小 90 秒、最大 1 日）が経過した後、ユーザーは自動的にログアウトされます。 *Remember for 30 days*が有効になっている場合、セッションが期間全体にわたって延長されるため、この設定は上書きされます。 時間のサフィックスに対応しています（例：90s、5m、2h、1d）。 このオプションは、次の場合には機能しません。 * 「Zabbix server is down の場合に警告を表示する」グローバル設定オプションが有効で、Zabbix Web インターフェースを開いたままにしている場合。 * Monitoring メニューのページで、バックグラウンドの情報更新が実行されている場合。 * *Remember me for 30 days* オプションを選択してログインした場合。
Default maintenance period	アドホックメンテナンスのデフォルトのメンテナンス期間を設定します。
Refresh	グラフやプレーンテキストデータなどに使用する更新間隔を設定します。0 に設定すると無効にできます。 時間のサフィックスに対応しています（例：90s、5m、1h）。
Rows per page	リストに表示する 1 ページあたりの行数を指定できます。
URL (after login)	ログインに成功した後、ユーザーを特定の URL（例：*Problems* ページ）に移動させることができます。

ユーザーメディア

メディアタブには、ユーザーに定義されているすべてのメディアが一覧表示されます。メディアは通知の送信に使用されます。



追加をクリックして、ユーザーにメディアを割り当てます。

メディアタイプが無効になっている場合:

- 名前の後に黄色の情報アイコンが表示されます。
- ステータス列に 無効と表示されます。

Note:
ユーザー自身のメディアの詳細を変更する権限は、ユーザーロールに基づいて付与または取り消すことができます（自分のメディアの作成と編集オプションを参照）。スーパーユーザーが他のユーザーのメディアの詳細を変更する権限も、ユーザーロールに基づいて付与または取り消すことができます（ユーザーメディアの作成と編集オプションを参照）。

ユーザーメディアの設定方法については、メディアタイプセクションを参照してください。

権限

権限タブには、以下の要素に関する情報が含まれています。

- ユーザーロール（新規作成ユーザーには必須）。スーパ管理者タイプのユーザーのみが変更できます。

Attention:
ユーザーロールがないとユーザーは作成できません（Zabbix User APIを除く）。以前に作成されたロールのないユーザーは、ロールを割り当てずに編集できます。ただし、一度ロールを割り当てると、変更はできますが削除はできません。

ロールのないユーザーは、LDAPまたはSAML認証を使用してのみ Zabbix にログインできます（LDAP/SAML 情報が Zabbix で設定されたユーザーグループのマッピングと一致する場合）。

- ユーザータイプ（ユーザー、管理者、スーパ管理者）。ユーザーロールの設定で定義されます。
- ユーザーがアクセスできるホストグループとテンプレートグループ。
  - ユーザーおよび 管理者タイプのユーザーは、デフォルトではグループ、テンプレート、ホストへのアクセス権がありません。アクセス権を付与するには、該当するエンティティへの権限を持つユーザーグループにユーザーを含める必要があります。
- Zabbix フロントエンド、モジュール、API メソッドのセクションや要素へのアクセス権。
  - アクセスが許可されている要素は緑色、拒否されている要素は薄いグレーで表示されます。
- 特定のアクションを実行する権限。
  - ユーザーが実行できるアクションは緑色、拒否されているアクションは薄いグレーで表示されます。

詳細は権限ページを参照してください。

権限

概要

Zabbix の権限は、ユーザータイプ、カスタマイズされたユーザーロール、およびユーザーグループに基づいて指定されるホストへのアクセスに依存します。

ユーザータイプ

Zabbix の権限は、主にユーザータイプによって決まります。

- *User* - メニューセクションへのアクセス権が制限されており（以下を参照）、デフォルトではどのリソースにもアクセスできません。ホストグループまたはテンプレートグループへの権限は、明示的に割り当てる必要があります。
- *Admin* - メニューセクションへのアクセス権が一部制限されています（以下を参照）。デフォルトでは、どのホストグループにもアクセスできません。ホストグループまたはテンプレートグループへの権限は、明示的に付与する必要があります。
- *Super admin* - すべてのメニューセクションにアクセスできます。すべてのホストグループおよびテンプレートグループに対して、読み書きアクセス権を持ちます。特定のグループへのアクセスを拒否しても、権限を取り消すことはできません。

メニューアクセス

Note:
一部の UI 要素へのアクセスを制限しても、そのページを開けなくなるだけであり、Web インターフェースの他の部分から基盤となるデータにアクセスする機能は削除されません。

次の表は、ユーザータイプごとの Zabbix メニューセクションへのアクセスを示しています。

メニューセクション		User	Admin	Super admin
ダッシュボード		+	+	+
監視		+	+	+
	障害	+	+	+
	ホスト	+	+	+
	最新データ	+	+	+
	マップ	+	+	+
	ディスカバリ		+	+

メニューセクション		User	Admin	Super admin
APM		+	+	+
	トレース	+	+	+
	メトリクス	+	+	+
	ログ	+	+	+
サービス		+	+	+
	サービス	+	+	+
	SLA		+	+
	SLA レポート	+	+	+
インベントリ		+	+	+
	概要	+	+	+
	ホスト	+	+	+
レポート		+	+	+
	システム情報		+	+
	スケジュールレポート		+	+
	可用性レポート	+	+	+
	トリガー上位 *100*	+	+	+
	監査ログ			+
	アクションログ			+
	通知		+	+
データ収集			+	+
	テンプレートグループ		+	+
	ホストグループ		+	+
	テンプレート		+	+
	ホスト		+	+
	メンテナンス		+	+
	イベント相関			+
	ディスカバリ		+	+
アラート			+	+
	トリガーアクション		+	+
	サービスアクション		+	+
	ディスカバリアクション		+	+
	自動登録アクション		+	+
	内部アクション		+	+
	メディアタイプ			+
	スクリプト			+
ユーザー				+
	ユーザーグループ			+
	ユーザーロール			+
	ユーザー			+
	API トークン			+
	認証			+
	デバイス			+
管理				+
	データソース			+
	一般			+
	監査ログ			+
	ハウスキーピング			+
	プロキシグループ			+
	プロキシ			+
	マクロ			+
	キュー			+

ユーザーロール

ユーザーロールを使用すると、ユーザータイプで定義された権限をカスタマイズできます。ユーザータイプの権限を超える権限を追加することはできませんが、一部の権限を取り消すことはできます。

さらに、ユーザーロールはメニューセクションだけでなく、サービス、モジュール、API メソッド、および Web インターフェースでのさまざまなアクションへのアクセスも決定します。

ユーザーロールは、スーパーユーザーがユーザー > ユーザーロールセクションで設定します。

ユーザーロールは、スーパーユーザーがユーザー設定フォームの権限タブでユーザーに割り当てます。

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

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

Add | Cancel

ホストへのアクセス

Zabbix では、任意のホストおよびテンプレートのデータへのアクセスは、ユーザーグループに対して、ホスト/テンプレートグループレベルでのみ許可されます。

これは、個々のユーザーに対してホスト（またはホストグループ）へのアクセス権を直接付与することはできないことを意味します。ホストへのアクセス権は、そのホストを含むホストグループへのアクセス権が付与されたユーザーグループのメンバーである場合にのみ付与できます。

同様に、テンプレートへのアクセス権は、そのテンプレートを含むテンプレートグループへのアクセス権が付与されたユーザーグループのメンバーである場合にのみユーザーに付与できます。

ユーザーグループ

概要

ユーザーグループを使用すると、組織上の目的やデータへの権限割り当てのために、ユーザーをグループ化できます。ホストグループおよびテンプレートグループのデータの表示と設定に関する権限は、個々のユーザーではなく、ユーザーグループに割り当てられます。

あるユーザーグループが利用できる情報と、別のユーザーグループが利用できる情報を分けることが適切な場合があります。これは、ユーザーをグループ化し、ホストグループおよびテンプレートグループに異なる権限を割り当てることで実現できます。

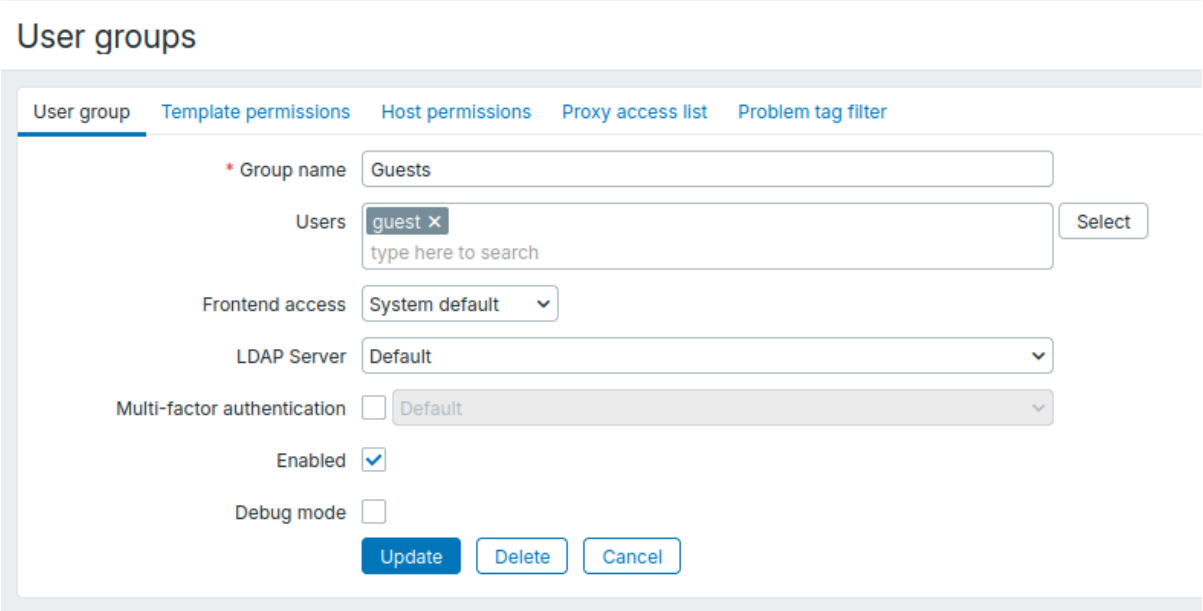
ユーザーは、任意の数のグループに所属できます。

設定

ユーザーグループを設定するには、次の手順を実行します。

1. ユーザー > ユーザーグループに移動します。
2. ユーザーグループの作成をクリックします（既存のグループを編集する場合は、グループ名をクリックします）。
3. フォームでグループ属性を編集します。

ユーザーグループタブには、グループの一般属性が表示されます。

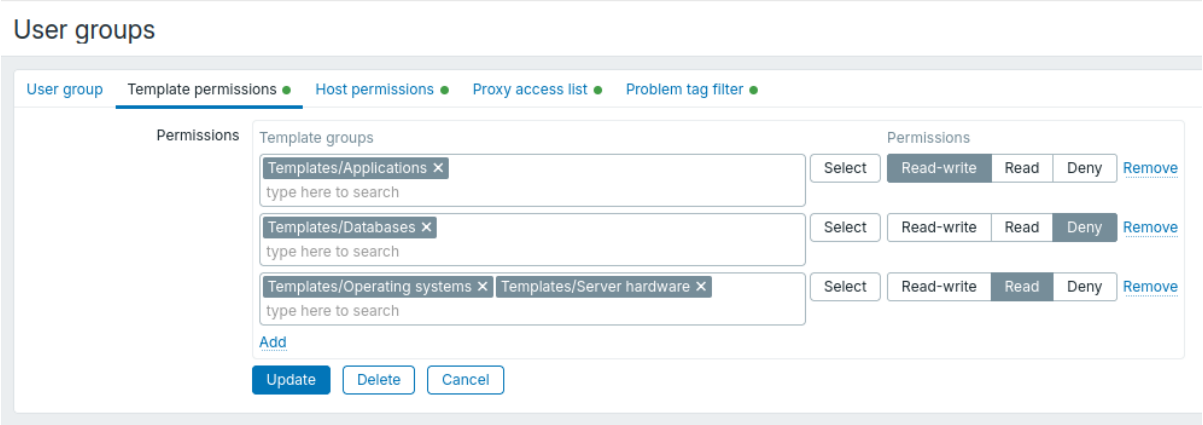


必須入力フィールドには、赤いアスタリスクが付いています。

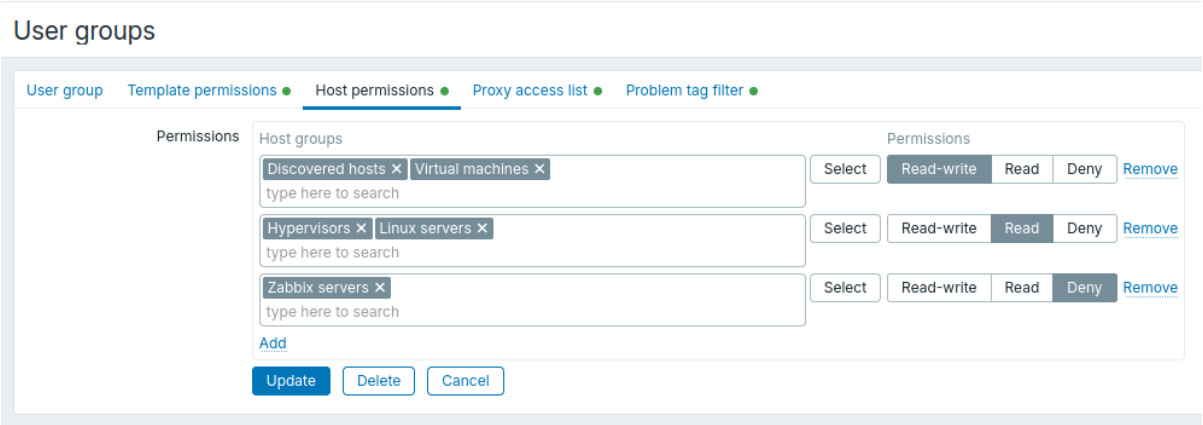
パラメータ	説明
グループ名	一意のグループ名。
ユーザー	グループにユーザーを追加するには、既存のユーザー名の入力を開始します。一致するユーザー名を含むドロップダウンが表示されたら、スクロールして選択します。 または、選択ボタンをクリックして、ポップアップでユーザーを選択できます。

パラメータ	説明
Web インターフェースへのアクセス	グループのユーザーの認証方法。 システムデフォルト - デフォルトの認証方法を使用します（グローバル設定で設定）。 内部 - Zabbix 内部認証を使用します（グローバルで LDAP 認証が使用されている場合でも同様）。 HTTP 認証がグローバルのデフォルトの場合は無視されます。 **LDAP** - LDAP 認証を使用します（グローバルで内部認証が使用されている場合でも同様）。 HTTP 認証がグローバルのデフォルトの場合は無視されます。 無効 - このグループの Web インターフェースへのアクセスを禁止します。
LDAP サーバー	ユーザーの認証に使用するLDAP サーバーを選択します。 このフィールドは、*Web* インターフェースへのアクセスが LDAP またはシステムデフォルトに設定されている場合のみ有効です。
多要素認証	ユーザーの認証に使用する多要素認証の方式を選択します。 デフォルト - MFA 設定でデフォルトに設定されている方式を使用します。MFA が有効な場合、新しいユーザーグループではデフォルトでこのオプションが選択されます。 **<** 方式名 **>** - 選択した方式を使用します（例: 「Zabbix TOTP」）。 無効 - このグループでは MFA を無効にします。MFA が無効な場合、新しいユーザーグループではデフォルトでこのオプションが選択されます。 ユーザーが MFA を有効にした複数のユーザーグループに所属している場合（または、少なくとも 1 つのグループで MFA が有効な場合）は、次の認証ルールが適用されます。いずれかのグループで「デフォルト」の MFA 方式が使用されている場合は、その方式でユーザーを認証します。それ以外の場合は、最初の方式（アルファベット順）が認証に使用されます。
有効	ユーザーグループとグループメンバーのステータス。 チェックあり - ユーザーグループとユーザーが有効です。 チェックなし - ユーザーグループとユーザーが無効です。
デバッグモード	このチェックボックスを選択すると、ユーザーのデバッグモードが有効になります。

テンプレートの権限タブでは、ユーザーグループによるテンプレートグループへのアクセスと、それらを介したテンプレートデータへのアクセスを指定できます。



ホストの権限タブでは、ユーザーグループによるホストグループへのアクセスと、それらを介したホストデータへのアクセスを指定できます。



Add をクリックして、テンプレート/ホストグループ（親グループまたはネストされたグループ）を選択し、それらに権限を割り当てます。

グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。または、選択をクリックすると、すべてのグループを一覧表示するポップアップウィンドウが開きます。

次に、オプションボタンを使用して、選択したグループに権限を割り当てます。
設定できる権限は次のとおりです。

- 読み書き - グループへの読み書きアクセス
- 読み取り - グループへの読み取り専用アクセス
- 拒否 - グループへのアクセスを拒否

同じテンプレート/ホストグループが異なる権限設定で複数の行に追加されている場合、最も厳しい権限が適用されます。

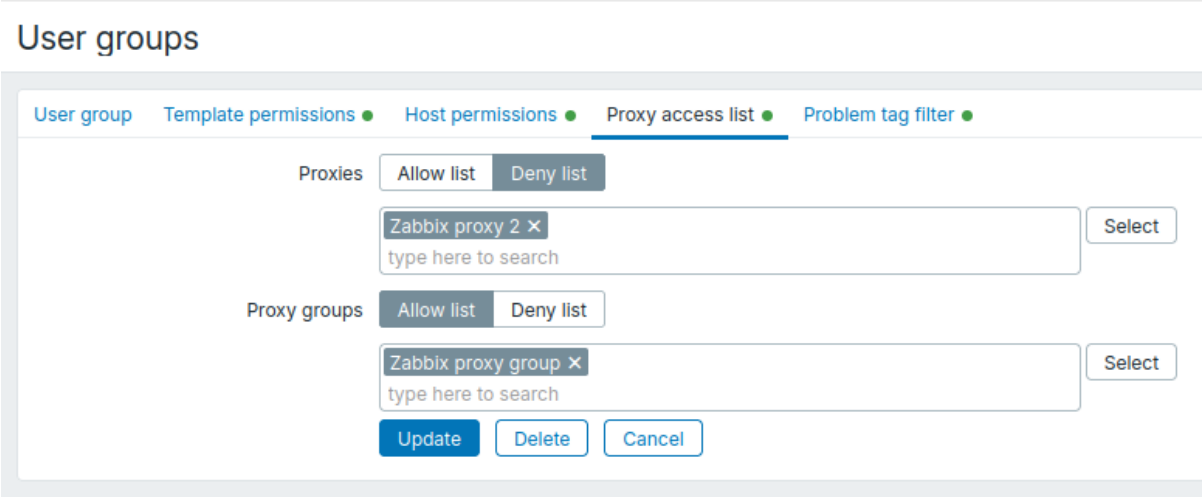
スーパーユーザーは、ネストされたグループに親グループと同じレベルの権限を強制的に設定できます。これは、ホスト/テンプレートグループの設定フォームで実行できます。

テンプレートの権限タブと ホストの権限タブでは、同じパラメータセットがサポートされています。

グループに対する現在の権限は 権限ブロックに表示され、変更または削除できます。

Note:
ユーザーグループがホストに対して 読み書き権限を持ち、そのホストにリンクされたテンプレートに対して 拒否または権限なしの場合、そのグループのユーザーはホスト上のテンプレート化されたアイテムを編集できません。また、テンプレート名は アクセスできないテンプレートと表示されます。

プロキシアクセスリストタブでは、個々のプロキシまたはプロキシグループを介して、ユーザーグループによるプロキシへのアクセスを指定できます。



プロキシグループフィールドを使用して、選択したグループに権限を適用します。
グループに属するプロキシは、そのグループを介してのみ許可または拒否できます。
どのプロキシグループにも属していない個々のプロキシを追加するには、プロキシフィールドを使用します。

プロキシと プロキシグループでは、同じパラメータセットがサポートされています。

プロキシまたは プロキシグループフィールドの横にある 選択ボタンをクリックして、それぞれプロキシまたはプロキシグループを選択します。
または、対応するフィールドにプロキシ名またはプロキシグループ名の入力を開始します。
一致するプロキシまたはプロキシグループのドロップダウンリストが表示されます。

許可リストを選択すると、ユーザーグループは選択したプロキシおよびプロキシグループにのみアクセスできます。
 許可リストで プロキシの選択が空の場合、ユーザーグループはプロキシグループに属していないプロキシにアクセスできません。
許可リストで プロキシグループの選択が空の場合、ユーザーグループはどのプロキシグループにもアクセスできません。

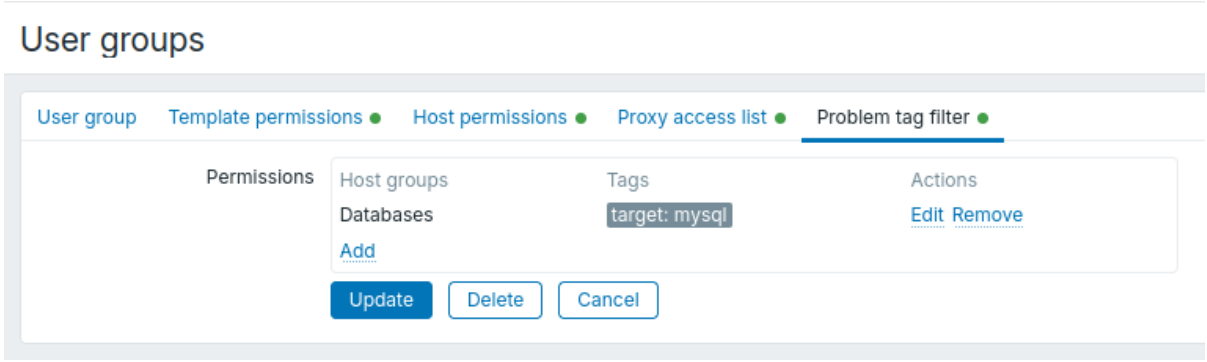
拒否リストを選択すると、ユーザーグループは選択したプロキシおよびプロキシグループにアクセスできません。
その他すべてのプロキシおよびプロキシグループへのアクセスは許可されます。
 拒否リストで プロキシの選択が空の場合、ユーザーグループはプロキシグループに属していないすべてのプロキシに完全にアクセスできます。
拒否リストで プロキシグループの選択が空の場合、ユーザーグループはすべてのプロキシグループに完全にアクセスできます。

複数のユーザーグループに所属するユーザーに対する権限の優先順位は次のとおりです。

- 拒否リストは 許可リストより優先されます。
  いずれかのグループがプロキシ/プロキシグループへのアクセスを拒否している場合、ユーザーはアクセスできません。

- 許可リストの権限は累積されます。
  複数のグループがアクセスを許可している場合、ユーザーはそれらのグループによって許可されたすべてのプロキシにアクセスできます。

障害タグフィルタータブでは、タグ名と値でフィルタリングした障害を表示するための、ユーザーグループに対するタグベースの権限を設定できます。

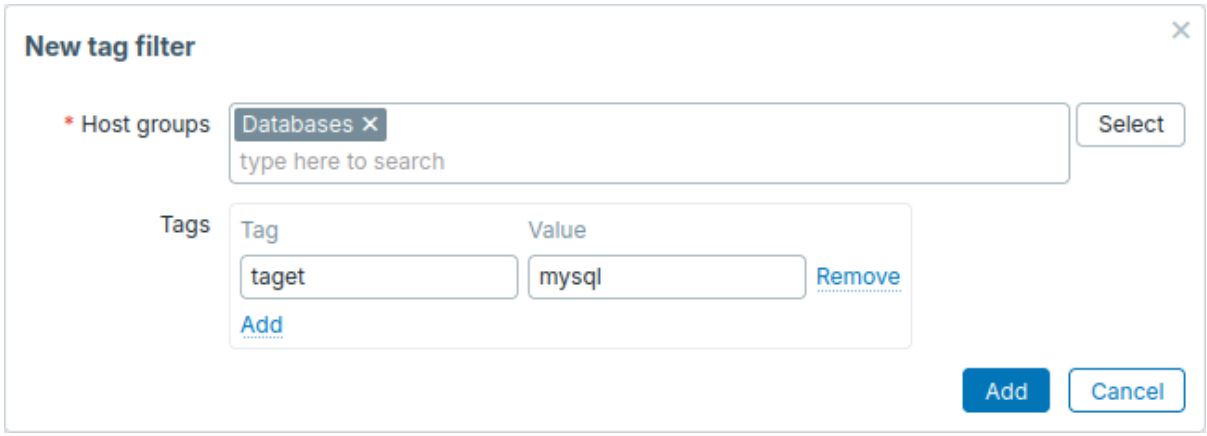


Add をクリックして、ホストグループを選択します。
タグフィルターを適用するホストグループを選択するには、選択をクリックして既存のホストグループの完全な一覧を表示するか、ホストグループ名の入力を開始して一致するグループのドロップダウンを表示します。
障害タグフィルターはテンプレートグループには適用できないため、ホストグループのみが表示されます。

値のないタグ名は追加できますが、名前のない値は追加できません。
最初の 3 つのタグ（値がある場合は値も）のみが 権限ブロックに表示されます。タグがそれ以上ある場合は、••• アイコンをクリックまたはホバーすると確認できます。

タグフィルターを使用すると、ホストグループへのアクセスと障害を表示できるかどうかを分けて設定できます。

たとえば、データベース管理者が「MySQL」データベースの障害のみを確認する必要がある場合、まずデータベース管理者用のユーザーグループを作成し、次に「target」をタグ名、「mysql」を値として指定します。



「target」をタグ名として指定し、値フィールドを空のままにすると、ユーザーグループは選択したホストグループにある「target」タグ名を持つすべての障害を表示できます。

タグ名とタグ値が正しく指定されていることを確認してください。正しく指定されていない場合、ユーザーグループには障害が表示されません。

選択した複数のユーザーグループにユーザーが所属している場合の例を確認します。
この場合、タグのフィルタリングには OR 条件が使用されます。

ユーザーグループ **A**			ユーザーグループ **B**			両方のグループに所属するユーザーに表示される
タグフィルター						
ホストグループ	タグ名	タグ値	ホストグループ	タグ名	タグ値	
データベース	target	mysql	データベース	target	oracle	target:mysql または target:oracle の障害が表示さ
障害タグフィルターで未設定			データベース	target	oracle	target:oracle の障害が表示される

複数のユーザーグループからのアクセス

ユーザーは任意の数のユーザーグループに所属することができます。これらのグループは、ホストやテンプレートに対して異なるアクセス権限を持つことができます。

したがって、権限のないユーザーが最終的にどのエンティティにアクセスできるかを知ることが重要です。次の例では、ユーザーがユーザーグループ A と B に所属している場合に、ホストグループ 1 のホスト **X** へのアクセスがさまざまな状況でどのように影響を受けるかを考えてみます。

- グループ A がホストグループ 1 に対して読込アクセスのみを持ち、グループ B がホストグループ 1 に対して読書/書込アクセスを持つ場合、ユーザーは'X' に対して読書/書込アクセスを取得します。

Attention:
" 読書/書込" 権限は" 読込" 権限よりも優先されます。

- 上記と同じシナリオで、'X' が同時にグループ A または B に拒否されているホストグループ 2 にも存在する場合、ホストグループ 1 に読書/書込アクセスがあっても、'X' へのアクセスは利用できません。
- グループ A に権限が定義されておらず、グループ B がホストグループ 1 に読書/書込アクセスを持つ場合、ユーザーは'X' に対して読書/書込アクセスを取得します。
- グループ A がホストグループ 1 に拒否アクセスを持ち、グループ B がホストグループ 1 に読書/書込アクセスを持つ場合、ユーザーは'X' へのアクセスが拒否されます。

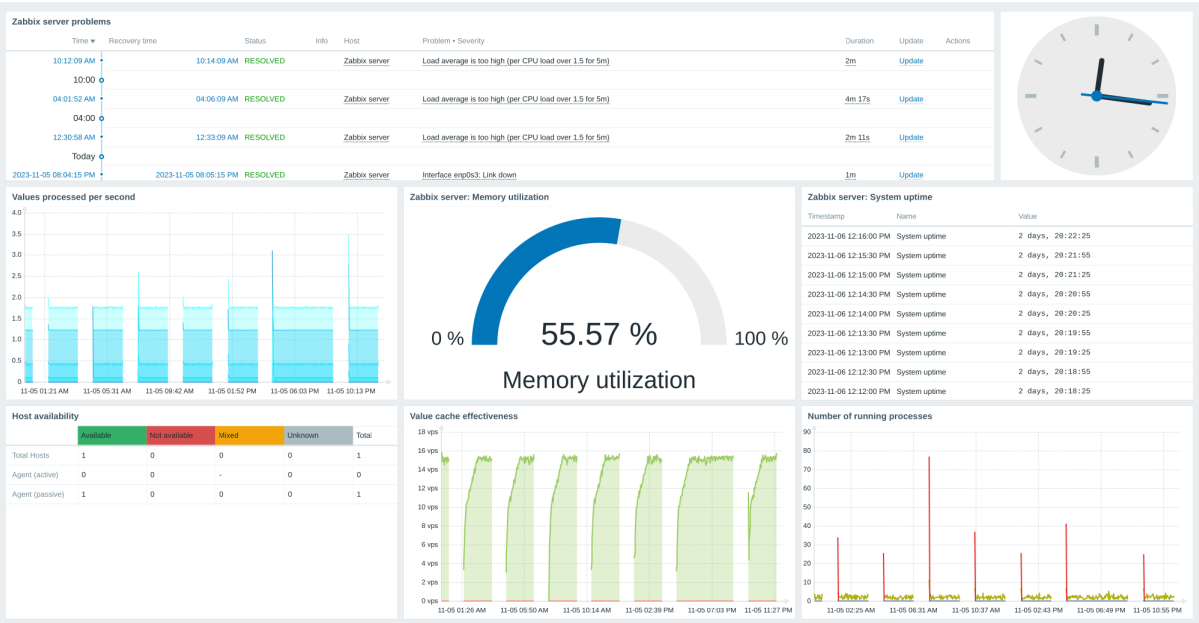
その他の詳細

- *Read-write* アクセス権を持つ Admin レベルのユーザーが、テンプレートが属するテンプレートグループへのアクセス権を持っていない場合、ホストへのテンプレートのリンク/リンク解除を行うことはできません。テンプレートグループへの *Read* アクセス権があれば、ホストへのテンプレートのリンク/リンク解除は可能ですが、テンプレートリストにはテンプレートが表示されず、他の場所でテンプレートを操作することもできません。
- *Read* アクセス権を持つ Admin レベルのユーザーは、設定セクションのホストリストでホストを見ることはできませんが、IT サービスの設定ではホストのトリガーにアクセスできます。
- Super Admin 以外のユーザー（'guest' を含む）は、マップが空であるか画像のみの場合はネットワークマップを見ることができます。マップにホスト、ホストグループ、トリガーが追加されると、権限が適用されます。
- Zabbix サーバーは、該当するホストへのアクセスが明示的に「拒否」されている場合、アクションの操作受信者として定義されたユーザーに通知を送信しません。

スケジュールされたレポート

概要

スケジュールレポート機能を使用すると、指定したダッシュボードの PDF 版を、指定した受信者に定期的に送信するよう設定できます。



前提条件:

- スケジュールレポートの生成を有効にするには、Zabbix Web サービスをインストールし、正しく設定する必要があります。手順については、スケジュールレポートの設定を参照してください。
- ユーザーには、次の権限を持つ管理者または特権管理者タイプのユーザーロールが必要です。
  - *UI* 要素へのアクセスブロックのスケジュールレポート（レポート設定を表示するため）

- アクションへのアクセスブロックのスケジュールレポートの管理（レポートを作成または編集するため）

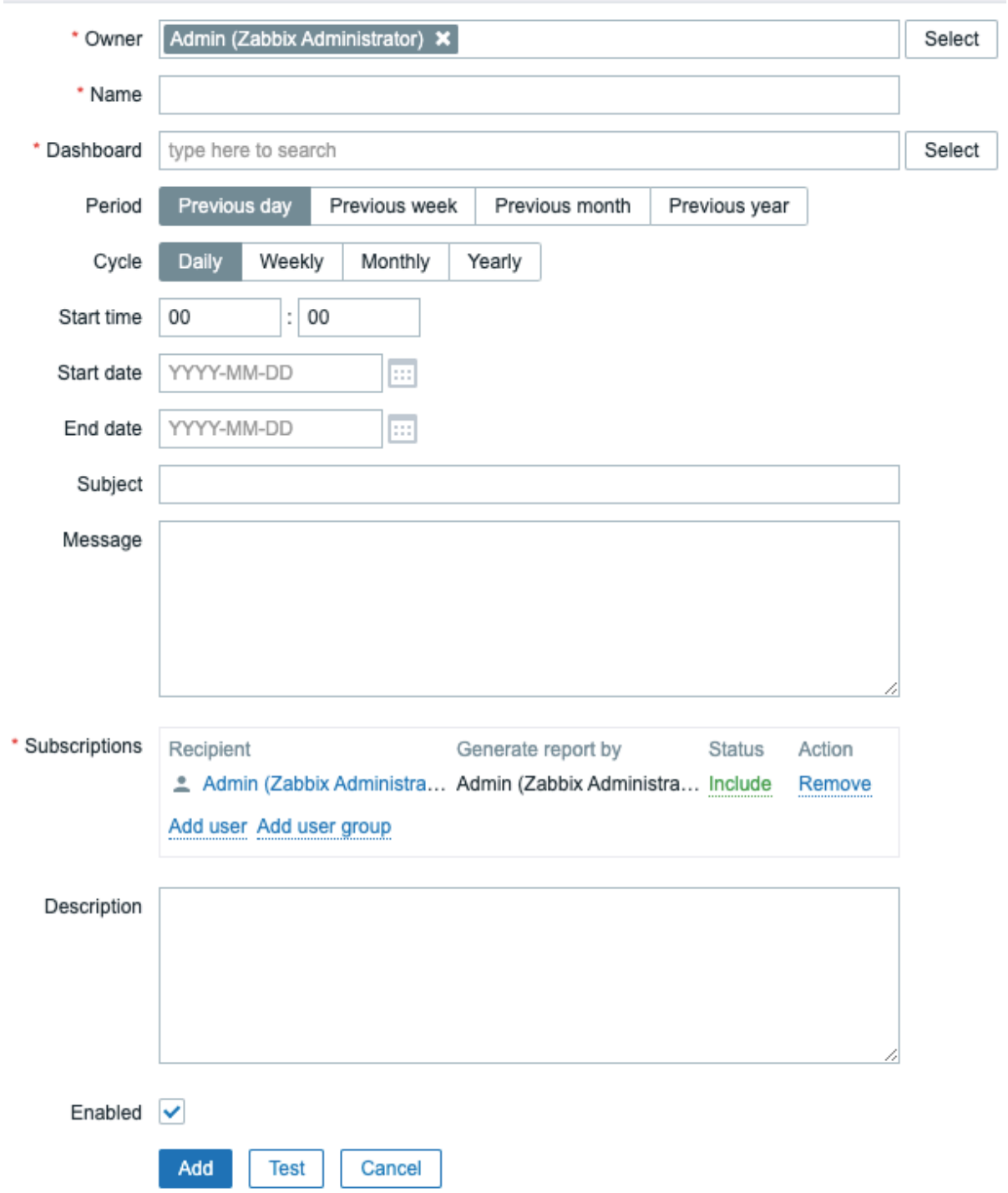
Zabbix Web インターフェースでスケジュールレポートを作成するには、次の手順を実行します。

1. レポート > スケジュールレポートに移動します。
2. 画面右上のレポートの作成をクリックします。
3. フォームにレポートのパラメータを入力します。

または、既存のレポートを開き、複製ボタンをクリックして、別の名前で保存することもできます。

設定

スケジュールレポートタブには、レポートの一般的な属性が含まれています。



すべての必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
所有者	レポートを作成するユーザー。スーパーユーザーレベルのユーザーは所有者を変更できます。管理者レベルのユーザーの場合、このフィールドは読み取り専用です。

パラメータ	説明
名前	レポートの名前。一意である必要があります。
ダッシュボード	レポートの基になるダッシュボード。一度に選択できるダッシュボードは 1 つだけです。ダッシュボードを選択するには、名前の入力を開始します。一致するダッシュボードの一覧が表示されるので、スクロールして選択します。または、フィールドの横にある 選択をクリックし、表示された一覧からダッシュボードを選択することもできます。
期間	レポートを作成する期間。前日、前週、前月、または前年を選択します。
サイクル	レポートの生成頻度。レポートは毎日、毎週、毎月、または毎年送信できます。「毎週」モードでは、レポートを送信する曜日を選択できます。
開始時刻	レポートを作成する時刻。hh:mm 形式で指定します。Zabbix サーバーのタイムゾーンが使用されることに注意してください。
繰り返し曜日	レポートを送信する曜日。このフィールドは、サイクルが「毎週」に設定されている場合にのみ使用できます。
開始日	定期的なレポート生成を開始する日付。
終了日	定期的なレポート生成を停止する日付。
件名	レポートメールの件名。サポートされているマクロ: {TIME}, {TIMESTAMP}。
メッセージ	レポートメールの本文。サポートされているマクロ: {TIME}, {TIMESTAMP}。
サブスクリプション	レポートの受信者一覧。デフォルトでは、レポートの所有者のみが含まれます。メールメディアが設定されている任意の Zabbix ユーザーを、レポートの受信者として指定できます。 受信者を追加するには、ユーザーを追加または ユーザーグループを追加をクリックします。 設定を編集するには、ユーザー名をクリックします。 レポート生成者 - 現在のユーザーまたは受信者のダッシュボード権限に基づいてレポートデータを生成するかどうかを指定します。 ステータス - ユーザーにレポートを送信するには「含める」を選択し、このユーザーへのレポート送信を防ぐには「除外」を選択します。少なくとも 1 人のユーザーに「含める」ステータスを設定する必要があります。「除外」ステータスは、含められているユーザーグループから特定のユーザーを除外するために使用できます。 権限が不十分なユーザー（つまり、管理者ユーザータイプに基づくロールを持ち、受信者またはレポート所有者と同じユーザーグループに所属していないユーザー）には、受信者および レポート生成者フィールドで実際の名前の代わりに「アクセスできないユーザー」または「アクセスできないユーザーグループ」と表示されます。ステータスおよび アクションフィールドは読み取り専用で表示されます。
有効	レポートのステータス。このチェックボックスをオフにすると、レポートは無効になります。
説明	レポートの任意の説明。この説明は内部使用を目的としており、レポートの受信者には送信されません。

フォームのボタン

フォーム下部のボタンを使用すると、いくつかの操作を実行できます。

Add	レポートを追加します。このボタンは新しいレポートの場合のみ使用できます。
Update	レポートのプロパティを更新します。
Clone	現在のレポートのプロパティに基づいて、別のレポートを作成します。
Test	現在のユーザーにレポートを送信して、レポート設定が正しいかどうかをテストします。
Delete	レポートを削除します。
Cancel	レポートのプロパティの編集をキャンセルします。

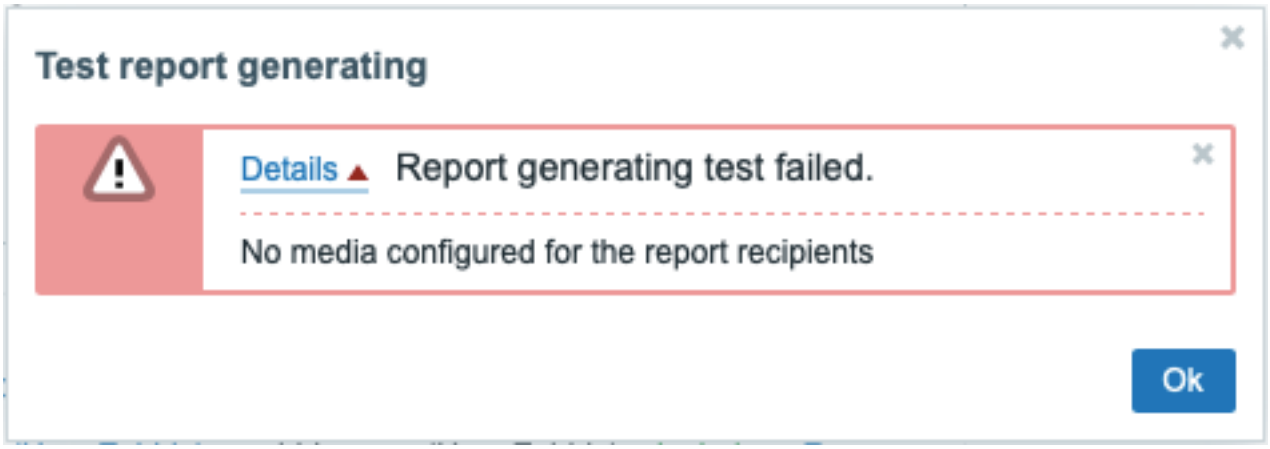
テスト

レポートをテストするには、レポート設定フォームの下部にある テストボタンをクリックします。

Note:
レポート設定フォームをダッシュボードのアクションメニューから開いた場合、テストボタンは使用できません。

設定が正しい場合、テストレポートは現在のユーザーに直ちに送信されます。テストレポートでは、購読者および レポート生成者のユーザー設定は無視されます。

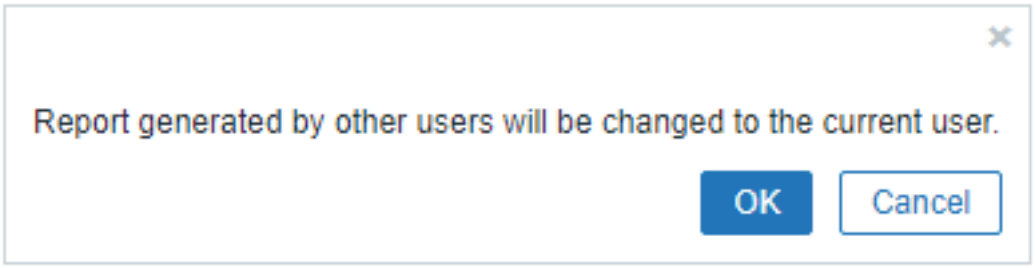
設定が正しくない場合、考えられる原因を説明するエラーメッセージが表示されます。



レポートの更新

既存のレポートを更新するには、レポート名をクリックし、必要な設定変更を行い、更新ボタンをクリックします。

既存のレポートが他のユーザーによって更新され、このユーザーがダッシュボードを変更した場合、更新ボタンをクリックすると「他のユーザーによって生成されたレポートは現在のユーザーに変更されます」という警告メッセージが表示されます。



この段階で *OK* をクリックすると、以下の変更が行われます。

- レポートの生成者の設定が、レポートを最後に編集したユーザーを表示するように更新されます（レポートの生成者が受信者に設定されていない場合）。
- 「アクセスできないユーザー」または「アクセスできないユーザーグループ」と表示されていたユーザーは、レポート購読者のリストから削除されます。

キャンセルをクリックすると、設定フォームが閉じられ、レポートの更新がキャンセルされます。

レポートのクローン作成

既存のレポートをすばやくクローンするには、既存のレポート設定フォームの下部にあるクローンボタンをクリックします。他のユーザーが作成したレポートをクローンする場合、新しいレポートの所有者は現在のユーザーになります。

ユーザーの権限に応じて、レポート設定が新しいレポート設定フォームにコピーされます。

- レポートをクローンするユーザーがダッシュボードへの権限を持っていない場合、ダッシュボードフィールドはクリアされます。
- レポートをクローンするユーザーがサブスクリプションリスト内の一部のユーザーまたはユーザーグループへの権限を持っていない場合、アクセスできない受信者はクローンされません。
- レポートの生成者の設定は、（レポートの生成者が受信者に設定されていない限り）現在のユーザーを表示するように更新されます。

必要な設定とレポート名を変更し、追加をクリックします。

データエクスポート

概要

Zabbix は、次の 2 つの方法でリアルタイムでのデータエクスポートをサポートします。

- ファイルにエクスポート
- 外部システムへのストリーミング

次のエンティティをエクスポートできます。

- トリガーイベント

- アイテムの値
- トレンド (ファイルにエクスポートのみ)

ファイルへのエクスポート

概要

トリガーイベント、アイテムの値、およびトレンドを、改行区切りの JSON 形式でリアルタイムにエクスポートするよう設定できます。

エクスポートはファイルに対して行われ、エクスポートファイルの各行は JSON オブジェクトです。値マッピングは適用されません。

エラーが発生した場合（データを書き込み先のエクスポートファイルに書き込めない、エクスポートファイルの名前を変更できない、または名前変更後に新しいファイルを作成できない場合）、そのデータ項目は破棄され、エクスポートファイルには書き込まれません。データは Zabbix データベースにのみ書き込まれます。書き込みの問題が解決すると、エクスポートファイルへのデータ書き込みは再開されます。

エクスポートファイルは、ファイル所有者のみが読み書きできる権限で作成されます。さらに、所有者グループには読み取り権限があります。それ以外の権限はすべて拒否されます。

データを受信した後、サーバーがデータをエクスポートする前にホスト/アイテムが削除された場合、ホスト/アイテムにはメタデータ（ホストグループ、ホスト名、アイテム名）が含まれないことがある点に注意してください。

設定

トリガーイベント、アイテムの値、トレンドのリアルタイムエクスポートは、エクスポートファイル用のディレクトリを指定して設定します。サーバー設定のExportDir パラメータを参照してください。

次の 2 つのパラメータも使用できます。

- ExportFileSize を使用すると、個々のエクスポートファイルに許可される最大サイズを設定できます。プロセスがファイルに書き込む必要がある場合、最初にファイルサイズを確認します。設定されたサイズ制限を超えている場合、ファイル名に.old を追加してファイル名を変更し、元の名前で新しいファイルを作成します。

Attention:
データを書き込むプロセスごとにファイルが作成されます（約 4～30 ファイル）。エクスポートファイルのデフォルトサイズは 1G であるため、大きなエクスポートファイルを保持すると、ディスク容量が急速に消費される可能性があります。

- ExportType を使用すると、エクスポートするエンティティタイプ（イベント、履歴、トレンド）を指定できます。

改行区切り JSON エクスポートプロトコル

このセクションでは、以下で使用される、改行区切り JSON 形式のエクスポートプロトコルの詳細を説明します。

- ファイルへのデータエクスポート
- 外部システムへのストリーミング

すべてのファイルは .ndjson 拡張子を持ちます。エクスポートファイルの各行は JSON オブジェクトです。

トリガーイベントのエクスポート

障害イベントについて、次の情報がエクスポートされます:

Field		Type	Description
clock		number	障害が検出された時点の、Epoch からの経過秒数（整数部）。
ns		number	正確な障害検出時刻を得るために clock に加算されるナノ秒数。
value		number	1（常に）。
eventid		number	障害イベント ID。
name		string	障害イベント名。
severity		number	障害イベントの深刻度（0 - 未分類、1 - 情報、2 - 警告、3 - 平均、4 - 高、5 - 災害）。
hosts		array	トリガー式に関与するホストの一覧。配列には少なくとも 1 要素が必要です。
	-	object	
	host	string	ホスト名。
	name	string	表示ホスト名。
groups		array	トリガー式に関与するすべてのホストのホストグループの一覧。配列には少なくとも 1 要素が必要です。
	-	string	ホストグループ名。
tags		array	障害タグの一覧（空にできます）。
	-	object	
	tag	string	タグ名。

Field	Type	Description
value	string	タグ値（空にできます）。

復旧イベントについて、次の情報がエクスポートされます:

Field	Type	Description
clock	number	障害が解決された時点の、Epoch からの経過秒数（整数部）。
ns	number	正確な障害解決時刻を得るために clock に加算されるナノ秒数。
value	number	0（常に）。
eventid	number	復旧イベント ID。
p_eventid	number	障害イベント ID。

トリガーイベントのエクスポート例

障害:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

復旧:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

障害（複数の障害イベント生成）:

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

復旧:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}
```

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

アイテム値のエクスポート

収集されたアイテム値について、以下の情報がエクスポートされます:

Field	Type	Description
host	object	アイテムのホストのホスト名。
host	string	ホスト名。
name	string	表示用ホスト名。
groups	array	アイテムのホストのホストグループの一覧。配列には少なくとも 1 つの要素が必要です。
-	string	ホストグループ名。
item_tags	array	アイテムタグの一覧（空でも可）。
-	object	
tag	string	タグ名。
value	string	タグの値（空でも可）。
itemid	number	アイテム ID。
name	string	表示用アイテム名。
clock	number	値が収集された時点の、Epoch からの経過秒数（整数部）。
ns	number	正確な値の収集時刻を得るために clock に加算するナノ秒数。
timestamp (*Log* only)	number	利用できない場合は 0。
source (*Log* only)	string	利用できない場合は空文字列。
severity (*Log* only)	number	利用できない場合は 0。

Field	Type	Description
eventid (*Log* only)	number	利用できない場合は 0。
value	number (for numeric items) or string (for text items)	収集されたアイテム値。
type	number	収集された値のタイプ: 0 - 数値浮動小数点, 1 - 文字, 2 - ログ, 3 - 数値符号なし, 4 - テキスト, 5 - バイナリ, 6 - JSON

アイテム値のエクスポート例

数値（符号なし）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

数値（浮動小数）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

文字、テキスト値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

ログ値:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

トレンドのエクスポート

計算されたトレンド値について、次の情報がエクスポートされます:

Field			Type	Description
host			object	アイテムのホストのホスト名。
	host		string	ホスト名。
	name		string	表示用ホスト名。
groups			array	アイテムのホストのホストグループの一覧。配列には少なくとも 1 つの要素が必要です。
	-		string	ホストグループ名。
item_tags			array	アイテムタグの一覧（空にすることもできます）。
	-		object	
		tag	string	タグ名。
		value	string	タグ値（空にすることもできます）。
itemid			number	アイテム ID。
name			string	表示用アイテム名。
clock			number	値が収集された時点の Epoch からの経過秒数（整数部）。
count			number	指定した 1 時間に収集された値の数。
min			number	指定した 1 時間のアイテム値の最小値。
avg			number	指定した 1 時間のアイテム値の平均値。
max			number	指定した 1 時間のアイテム値の最大値。
type			number	値のタイプ: 0 - numeric float, 3 - numeric unsigned

トレンドエクスポートの例

数値（符号なし）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

数値（浮動小数）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

外部システムへのストリーミング

概要

Zabbix から外部システムへ、HTTP 経由でアイテムの値やイベントをストリーミングできます（プロトコルの詳細を参照）。

タグフィルターを使用すると、アイテムの値やイベントのサブセットをストリーミングできます。

データストリーミングは、2 種類の Zabbix サーバープロセスタイプ（connector manager と connector worker）によって処理されます。Zabbix 内部アイテム zabbix[connector_queue] を使用すると、コネクターキューに登録された値の数を監視できます。

設定

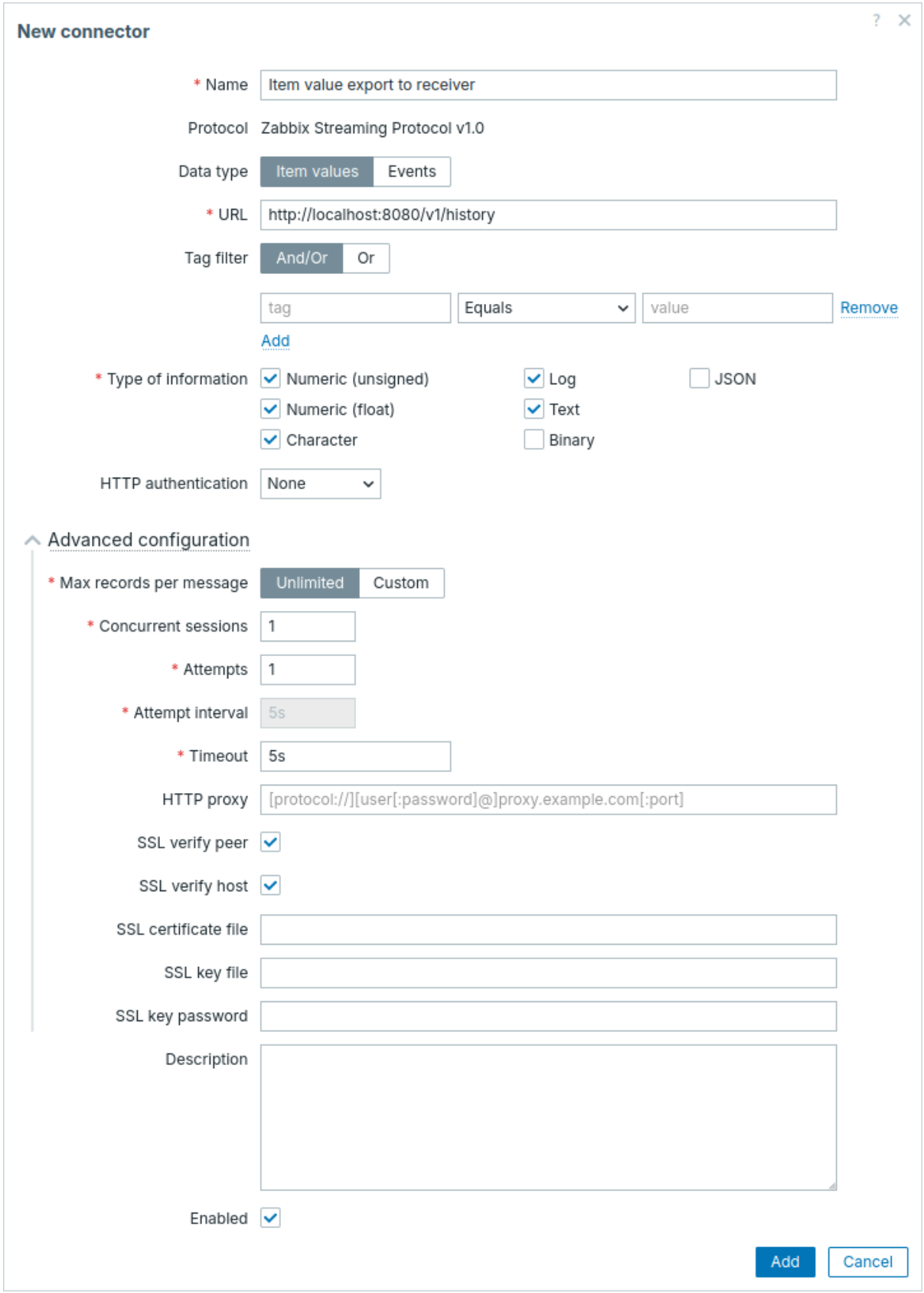
外部システムへのデータストリーミングを設定するには、次の手順が必要です。

1. Zabbix からデータを受信するリモートシステムをセットアップします。この目的のために、次のツールを利用できます。

- 受信した情報を events.ndjson および history.ndjson ファイルに記録する、シンプルな receiver の例。
- Kafka connector for Zabbix server - Zabbix サーバーから Kafka ブローカーへアイテムの値とイベントを転送するよう設計された、Go で記述された軽量サーバー。

2. zabbix_server.conf のStartConnectors パラメータを調整して、Zabbix で必要な数のコネクターのワーカーを設定します。コネクターのワーカー数は、Zabbix Web インターフェースで設定したコネクター数と一致させる必要があります（同時セッション数が 1 より多い場合は、それ以上にします）。その後、Zabbix サーバーを再起動します。

3. Zabbix Web インターフェースで新しいコネクターを設定し（管理 > 一般設定 > コネクター）、zabbix_server -R config_cache_reload コマンドでサーバーキャッシュを再読み込みします。



必須フィールドにはアスタリスクが付いています。

パラメータ	説明
名前	コネクター名を入力します。
データ型	ストリーミングするデータ型を選択します。 アイテムの値 - Zabbix から外部システムへアイテムの値をストリーミングします。 イベント - Zabbix から外部システムへイベントをストリーミングします。
URL	受信側の URL を入力します。ユーザーマクロを使用できます。

パラメータ	説明
タグフィルター	タグフィルターに一致するアイテムの値またはイベントのみをエクスポートします。設定しない場合は、すべてをエクスポートします。 特定のタグおよびタグの値を含めたり除外したりできます。複数の条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグの値に入力した文字列が含まれる、指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグの値に入力した文字列が含まれる、指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には次の 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - 1 つの条件を満たせば十分です。
情報のタイプ	コネクターがストリーミングするアイテムの値をフィルタリングする情報のタイプ（数値（整数）、数値（浮動小数）、文字など）を選択します。 データ型が「アイテムの値」に設定されている場合に使用できます。
HTTP 認証	認証オプションを選択します。 なし - 認証を使用しません。 **Basic** - Basic 認証を使用します。 **NTLM** - NTLM（Windows NT LAN Manager）認証を使用します。 **Kerberos** - Kerberos 認証を使用します（Zabbix での Kerberos の設定も参照）。 **Digest** - Digest 認証を使用します。 **Bearer** - Bearer 認証を使用します。
ユーザー名	ユーザー名を入力します（最大 255 文字）。ユーザーマクロを使用できます。 *HTTP* 認証が「Basic」、「NTLM」、「Kerberos」、または「Digest」に設定されている場合に使用できます。
パスワード	ユーザーパスワードを入力します（最大 255 文字）。ユーザーマクロを使用できます。 *HTTP* 認証が「Basic」、「NTLM」、「Kerberos」、または「Digest」に設定されている場合に使用できます。
Bearer トークン	Bearer トークンを入力します。ユーザーマクロを使用できます。 *HTTP* 認証が「Bearer」に設定されている場合に使用でき、必須です。
詳細設定	詳細設定ヘッダーをクリックすると、詳細設定オプションが表示されます（以下を参照）。
メッセージあたりの最大レコード数	1 つのメッセージ内でストリーミングできる値またはイベントの最大数を指定します。
同時セッション数	このコネクターで実行する送信プロセスの数を選択します。最大 100 セッションまで指定できます。 デフォルト値は「1」です。
試行回数	データストリーミングを試行する回数です。最大 5 回まで指定できます。デフォルト値は「1」です。
試行間隔	データストリーミングに失敗した後、コネクターが待機する時間を指定します。最大 10 秒まで指定できます。デフォルト値は「5s」です。 試行回数が「2」以上に設定されている場合に使用できます。 接続の確立に失敗した場合、または HTTP レスポンスコードが 200、201、202、203、204 以外の場合は、試行に失敗したとみなされます。通信エラーが発生した場合、または HTTP レスポンスコードが 200、201、202、203、204、400、401、403、404、405、415、422 以外の場合、再試行がトリガーされます。リダイレクトは追従されるため、302 -> 200 は成功レスポンスですが、302 -> 503 の場合は再試行がトリガーされます。
タイムアウト	メッセージのタイムアウトを指定します（1～60 秒、デフォルトは 5 秒）。 時間のサフィックスを使用できます（例：30s、1m）。ユーザーマクロを使用できます。

パラメータ	説明
HTTP プロキシ	次の形式で使用する HTTP プロキシを指定できます。 `[protocol://][username[:password]@]proxy.example.com[:port]` ユーザーマクロを使用できます。 オプションの `protocol://` プレフィックスを使用して、代替プロキシプロトコルを指定できます(プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました)。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。デフォルトでは、ポート 1080 が使用されます。 *HTTP* プロキシを指定すると、`http_proxy` や `HTTPS_PROXY` などのプロキシ関連の環境変数がプロキシによって上書きされます。指定しない場合、プロキシ関連の環境変数は上書きされません。 入力した値はそのまま渡され、妥当性の確認は行われません。 SOCKS プロキシのアドレスも入力できます。誤ったプロトコルを指定すると、コネクターは Zabbix からアイテムの値またはイベントをストリーミングできません。 HTTP プロキシでサポートされるのは、単純な認証のみです。
SSL ピアの検証	Web サーバーの SSL 証明書を検証するには、チェックボックスを選択します。 サーバー証明書は、システム全体の CA の場所にある信頼された認証局（CA）証明書に対して検証されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータ`SSLCALocation` で上書きできます。
SSL ホストの検証	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスを選択します。 これにより、`CURLOPT_SSL_VERIFYHOST` cURL オプションが設定されます。
SSL 証明書ファイル	クライアント認証に使用する SSL 証明書ファイルの名前です。証明書ファイルは PEM[1] 形式である必要があります。ユーザーマクロを使用できます。 証明書ファイルに秘密鍵も含まれている場合は、*SSL* キーファイルフィールドを空のままにします。 鍵が暗号化されている場合は、*SSL* キーパスワードフィールドにパスワードを指定します。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ`SSLCertLocation` で指定します。
SSL キーファイル	クライアント認証に使用する SSL 秘密鍵ファイルの名前です。秘密鍵ファイルは PEM[1] 形式である必要があります。ユーザーマクロを使用できます。 このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ`SSLKeyLocation` で指定します。
SSL キーパスワード	SSL 秘密鍵ファイルのパスワードです。ユーザーマクロを使用できます。
説明	コネクターの説明を入力します。
有効	コネクターを有効にするには、チェックボックスを選択します。

Kafka コネクターを、ブートストラップブローカーのアドレスをカンマ区切りで指定して設定した場合(例：`Kafka.URL=kafka1.example.com`
Kafka クライアントは最初に応答したブローカーに接続し、そのブローカーのクラスター metadata を使用します。リストに異なる Kafka クラスターのアドレスが含まれている場合、最も速く応答したクラスターのみが使用され、その他のアドレスは利用不可としてログに記録されます。その結果、コネクターが接続されている場合でも、次のような起動時の警告が表示されることがあります。

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

一部の環境（プライベートネットワーク、コンテナネットワーク、または標準以外の DNS/hosts 設定）では、ホスト名または IP がループバックアドレス（例：`127.0.0.1/localhost`）に解決されたり、クライアントによって正規化されたりすることがあります。そのため、このような警告が誤解を招く場合があります。混乱を避けるには、すべての `Kafka.URL` アドレスが同じ Kafka クラスターに属していることを確認し、コネクターのホストおよびブローカーの `advertised.listeners` から DNS 解決を検証し、ブローカーの advertised アドレスに解決されるアドレスを使用してください。

プロトコル

サーバーと受信側の間の通信は、REST API、NDJSON、”Content-Type: application/x-ndjson” を使用して HTTP 経由で行われます。

詳細については、Newline-delimited JSON export protocol を参照してください。

サーバーリクエスト

ストリーミングアイテム値の例:

```
POST /v1/history HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
```

```
Content-Length: 628
Content-Type: application/x-ndjson

{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"b
```

ストリーミングイベントの例:

```
POST /v1/events HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 333
Content-Type: application/x-ndjson

{"clock":1673454303,"ns":800155804,"value":1,"eventid":5,"name":"trigger for foo being 0","severity":0,"ho
{"clock":1673454303,"ns":832290669,"value":0,"eventid":6,"p_eventid":5}
```

レシーバーのレスポンス

レスポンスは、HTTP レスポンスのステータスコードと JSON ペイロードで構成されます。正常に処理されたリクエストでは、HTTP レスポンスのステータスコードは”200”、”201”、”202”、”203”、または”204” である必要があります。失敗したリクエストでは、それ以外のコードになります。

成功時の例:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

エラー時の例:

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

SNMP ゲートウェイ

概要

Zabbix SNMP ゲートウェイは、SNMP ポーリングとトラッピングの両方をサポートする snmpd 用の AgentX 拡張です。

Zabbix SNMP ゲートウェイを使用すると、SNMP プロトコルを使用して以下の情報を取得できます:

- トリガーのデータ
- 問題トリガーのデータ
- ホストグループのステータス (グループごとのトリガーステータスごとのトリガー数)

データは OID によって取得されます。OID は共通のベースと特定のサフィックスの組み合わせです。共通のベースは SNMP ゲートウェイの設定ファイルで設定されます。例:

- `BaseOID=1.3.6.1.4.1.3043.7.55` - 任意のトリガーデータ用
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - 問題トリガーデータ用
- `BaseOID=1.3.6.1.4.1.3046.7.55` - ホストグループのステータス用

OID のサフィックスは、Web インターフェースでホストのトリガーの設定時にタグ (例: `OIDSuffix:3`) として設定します。

この場合、トリガーに関するすべての情報は `OID=1.3.6.1.4.1.3043.7.55.X.3` で利用できます。ここで”X” はトリガーデータフィールドの番号です (例: 1 - サフィックス、2 - ID、3 - 式、4 - 説明など)。

より詳細な説明や設定ファイルの例については、SNMP ゲートウェイの readme ファイルを参照してください。

インストールとセットアップ

インストール手順については、SNMP ゲートウェイのreadmeファイルを参照してください。

- snmpd のインストールと設定
- AgentX サポートの有効化
- Zabbix SNMP ゲートウェイの設定
- トリガー状態変更のための SNMP トラップの設定

データの取得

すべてが正しく設定されていれば、snmpwalk および snmpget コマンドを使用してデータを取得できます。

```
[user@localhost ~]# snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55.2.3
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
```

```
[user@localhost ~]# snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55
SNMPv2-SMI::enterprises.3043.7.55.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3043.7.55.1.3 = INTEGER: 3
SNMPv2-SMI::enterprises.3043.7.55.1.4 = INTEGER: 4
SNMPv2-SMI::enterprises.3043.7.55.1.5 = INTEGER: 5
SNMPv2-SMI::enterprises.3043.7.55.1.6 = INTEGER: 6
SNMPv2-SMI::enterprises.3043.7.55.1.10 = INTEGER: 10
SNMPv2-SMI::enterprises.3043.7.55.2.1 = INTEGER: 15367
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
SNMPv2-SMI::enterprises.3043.7.55.2.4 = INTEGER: 15365
SNMPv2-SMI::enterprises.3043.7.55.2.5 = INTEGER: 15366
SNMPv2-SMI::enterprises.3043.7.55.2.6 = INTEGER: 13493
SNMPv2-SMI::enterprises.3043.7.55.2.10 = INTEGER: 13503
...
```

フィルタリングオプション

SNMP ゲートウェイの設定で、問題トリガー情報を以下のように制限できます。

- 深刻度で制限（デフォルトは ProblemMinSeverity=-1）
- 確認済みの問題を非表示にする（デフォルトは ProblemHideAck=false）

SNMP ゲートウェイの設定で、ホストグループごとの問題数を以下のように制限できます。

- 不明な状態のトリガーで制限（デフォルトは CountUnknown=false）
- 確認済み/未確認/すべての問題のトリガーで制限（デフォルトは CountAcknowledgeStatus=all）

設定のエクスポート/インポート

概要

Zabbix のエクスポート/インポート機能を使用すると、ある Zabbix システムと別の Zabbix システムの間で、さまざまな設定エンティティを交換できます。

この機能の一般的な使用例は次のとおりです。

- テンプレートやネットワークマップを共有する - Zabbix ユーザーは自分の設定パラメータを共有できます
- Zabbix Community templates にテンプレートをアップロードする。その後、他のユーザーはテンプレートをダウンロードし、ファイルを Zabbix にインポートできます。
- サードパーティ製ツールと統合する - 汎用的な YAML、XML、JSON 形式により、サードパーティ製ツールやアプリケーションとの統合、およびデータのインポート/エクスポートが可能になります

エクスポート/インポートできるもの

エクスポート/インポートできるオブジェクトは次のとおりです。

- ホストグループ (*Zabbix API* 経由のみ)
- テンプレートグループ (*Zabbix API* 経由のみ)
- テンプレート
- ホスト
- ネットワークマップ
- メディアタイプ

- 画像
- ダッシュボード
- 正規表現

エクスポート形式

データは Zabbix Web インターフェースまたはZabbix API を使用してエクスポートできます。対応しているエクスポート形式は YAML、XML、JSON です。

エクスポートの詳細

- サポートされているすべての要素は、1 つのファイルにエクスポートされます。
- リンクされたテンプレートから継承されたホストおよびテンプレートのエンティティ（アイテム、トリガー、グラフ、ディスカバリルール）はエクスポートされません。これらのエンティティに対してホストレベルで行われた変更（たとえば、アイテムの間隔の変更、正規表現の修正、または low-level discovery ルールへのプロトタイプの追加）は、エクスポート時に失われます。インポート時には、リンクされたテンプレートのすべてのエンティティが元のリンク先テンプレートと同じように再作成されます。
- low-level discovery によって作成されたエンティティ、およびそれらに依存するエンティティはエクスポートされません。たとえば、LLD ルールによって生成されたアイテム用に作成されたトリガーはエクスポートされません。
- エクスポート対象のホスト/テンプレートにタイムアウトをサポートするエンティティが含まれている場合、それらのエンティティに個別のタイムアウトが設定されていれば、そのタイムアウト値もエクスポートされます。

インポートの詳細

- インポートは最初のエラーで停止します。
- 画像インポート中に既存の画像を更新する場合、”imagetype” フィールドは無視されます。つまり、インポートによって画像タイプを変更することはできません。
- ”Delete missing” オプションを使用してホスト/テンプレートをインポートする場合、インポートファイルに存在しないホスト/テンプレートマクロは、インポート後にホスト/テンプレートから削除されます。
- アイテム、トリガー、グラフ、discoveryRules、itemPrototypes、triggerPrototypes、graphPrototypes の空のタグは意味を持ちません。つまり、存在しない場合と同じです。
- インポートされたホスト/テンプレートのエンティティに独自のタイムアウト設定がある場合は、それが適用されます。そうでない場合は、プロキシ/グローバルのタイムアウトが適用されます。
- インポートは YAML、XML、JSON をサポートします。インポートファイルには正しい拡張子が必要です。YAML は.yaml および.yml、XML は.xml、JSON は.json です。サポートされる XML バージョンについては、互換性情報 を参照してください。
- インポートは UTF-8 エンコーディングの設定ファイルのみをサポートします (BOM の有無は問いません)。その他のエンコーディング (UTF16LE、UTF16BE、UTF32LE、UTF32BE など) は、インポート変換エラーになります。

YAML ベース形式

YAML エクスポート形式には、次のノードが含まれます。

- Zabbix YAML エクスポートのルートノード
- エクスポートバージョン

```
zabbix_export:
  version: '8.0'
```

その他のノードは、エクスポートされるオブジェクトに依存します。

XML 形式

XML エクスポート形式には、次のタグが含まれます。

- XML ドキュメントのデフォルトヘッダー
- Zabbix XML エクスポートのルートタグ
- エクスポートバージョン

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>8.0</version>
</zabbix_export>
```

その他のタグは、エクスポートされるオブジェクトに依存します。

JSON 形式

JSON エクスポート形式には、次のオブジェクトが含まれます。

- Zabbix JSON エクスポートのルートオブジェクト
- エクスポートバージョン

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

その他のオブジェクトは、エクスポートされるオブジェクトに依存します。

テンプレートグループ

概要

Web インターフェースでは、テンプレートグループはエクスポートでテンプレートのエクスポート時にのみエクスポートできます。テンプレートをエクスポートすると、そのテンプレートが属するすべてのグループも自動的に一緒にエクスポートされます。

API では、テンプレートグループをテンプレートとは独立してエクスポートできます。

エクスポート形式

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Network devices'
```

エクスポートされた要素

Element	Type	Description
uuid	string	このテンプレートグループの一意の識別子。
name	string	グループ名。

ホストグループ

概要

Web インターフェースでは、ホストグループはホストのエクスポートと一緒の場合にのみエクスポートできます。ホストをエクスポートすると、そのホストが所属するすべてのグループも自動的に一緒にエクスポートされます。

API では、ホストとは独立してホストグループをエクスポートできます。

エクスポート形式

```
   host_groups:
    - uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Zabbix servers'
```

エクスポートされた要素

Element	Type	Description
uuid	string	このホストグループの一意の識別子。
name	string	グループ名。

テンプレート

概要

テンプレートは、多くの関連オブジェクトおよびオブジェクト間の関係とともにエクスポートされます。

テンプレートのエクスポートには、次の内容が含まれます。

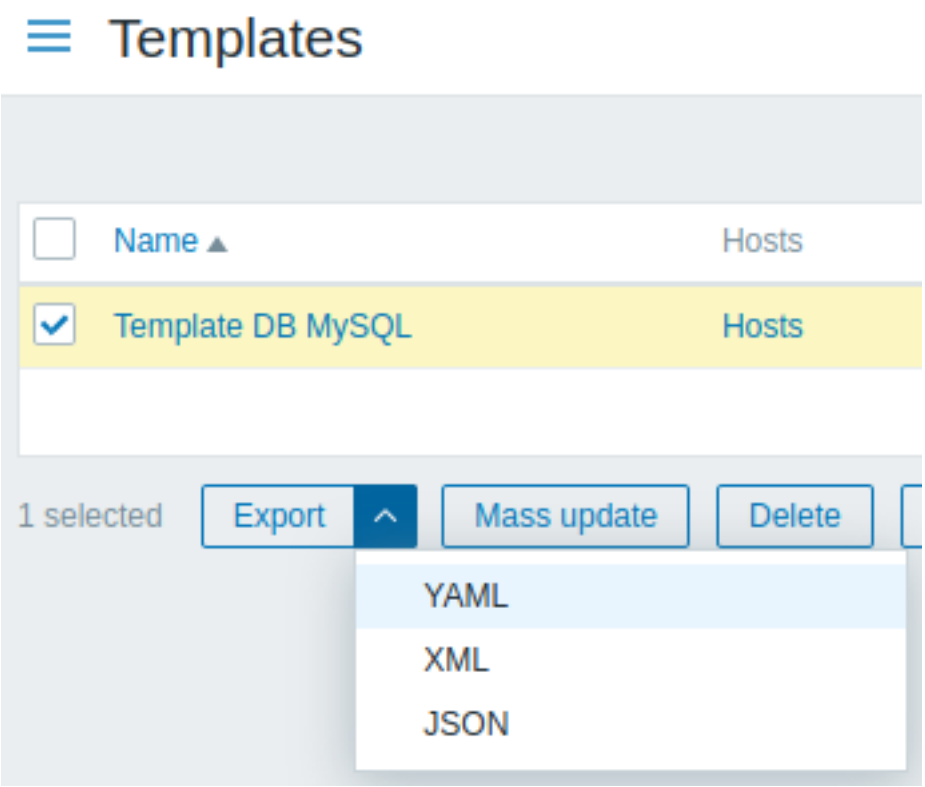
- リンクされたテンプレートグループ
- リンクされたホストグループ（ホストプロトタイプの設定で使用されている場合）
- テンプレートデータ
- 他のテンプレートへのリンク
- テンプレートグループへのリンク
- 直接リンクされたアイテム
- 直接リンクされたトリガー

- 直接リンクされたグラフ
- 直接リンクされたダッシュボード
- すべてのプロトタイプを含む直接リンクされたディスカバリルール
- 直接リンクされた Web シナリオ
- 値マップ

エクスポート

テンプレートをエクスポートするには、次の手順を実行します。

1. データ収集 > テンプレートに移動します。
2. エクスポートするテンプレートのチェックボックスをオンにします。
3. 一覧の下にある エクスポートをクリックして、デフォルトの YAML 形式でエクスポートするか、矢印をクリックして別のエクスポート形式を選択します。



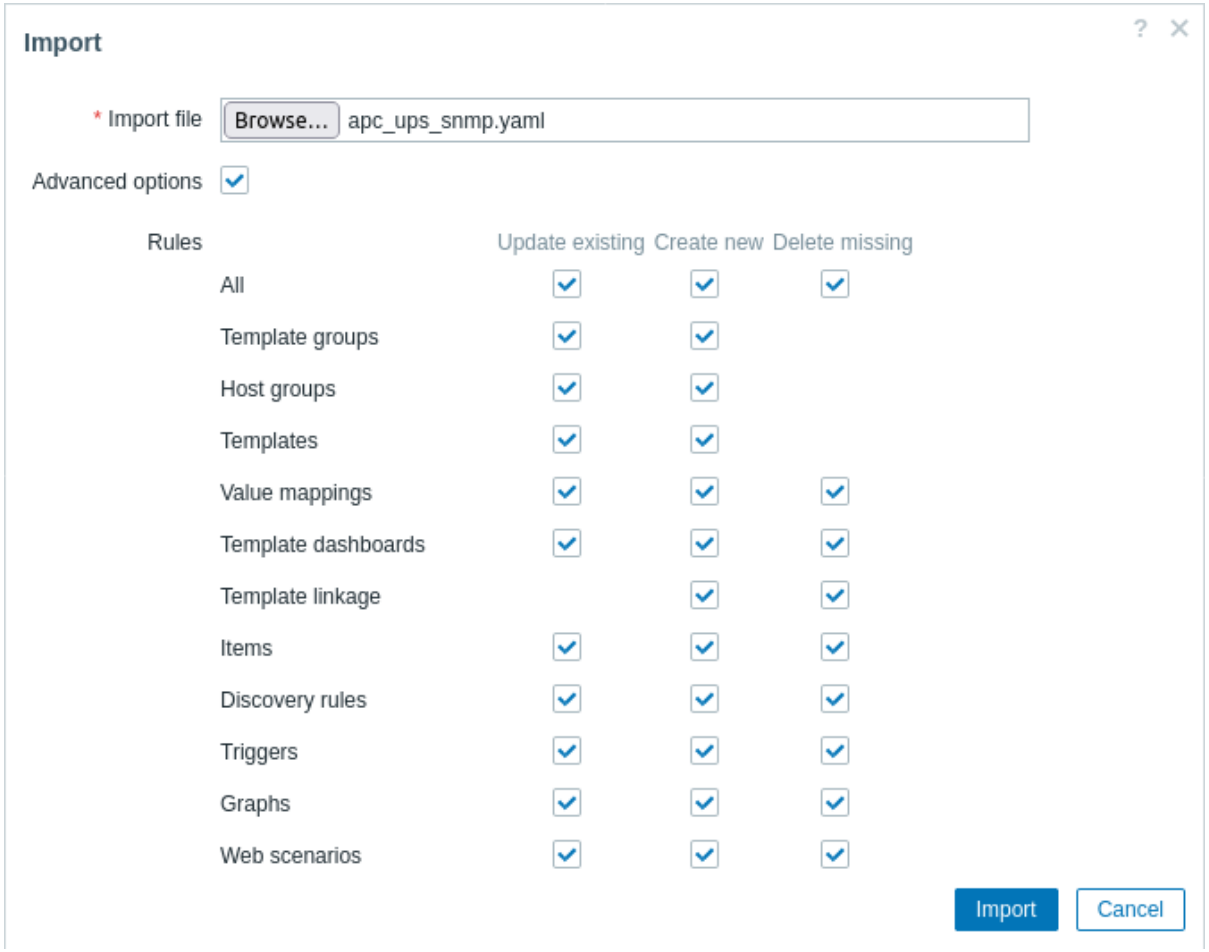
選択した形式に応じて、テンプレートは次のデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_templates.yaml` - YAML エクスポート;
- `zabbix_export_templates.xml` - XML エクスポート;
- `zabbix_export_templates.json` - JSON エクスポート。

インポート

テンプレートをインポートするには、次の手順を実行します。

1. データ収集 > テンプレートに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. 適用するインポートルールのチェックボックスにチェックを入れます（以下を参照）。
5. 設定フォームの右下隅にある インポートをクリックします。



詳細オプションチェックボックスにチェックを入れると、インポート可能なすべての要素の詳細な一覧が表示されます。必要に応じて各インポートルールにチェックを入れる、またはチェックを外します。

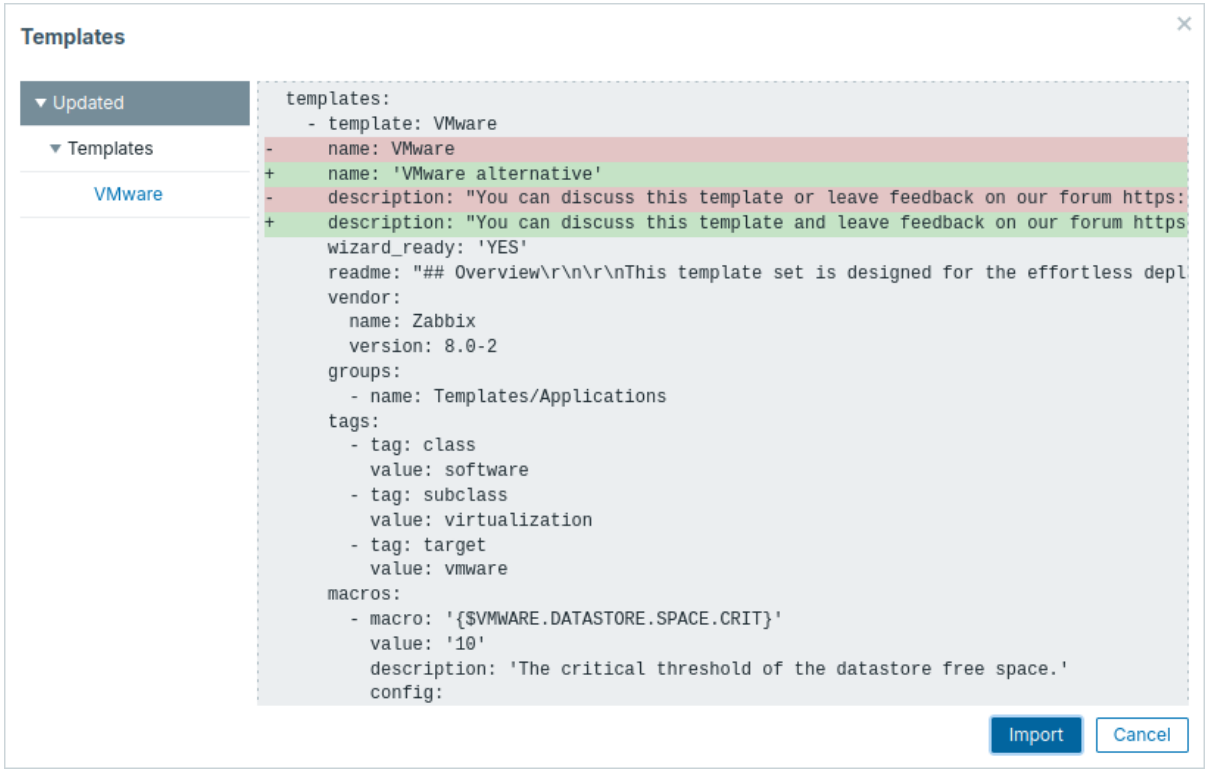
すべて行のチェックボックスをクリックすると、その下にあるすべての要素にチェックを入れる、またはチェックを外すことができます。

インポートルール:

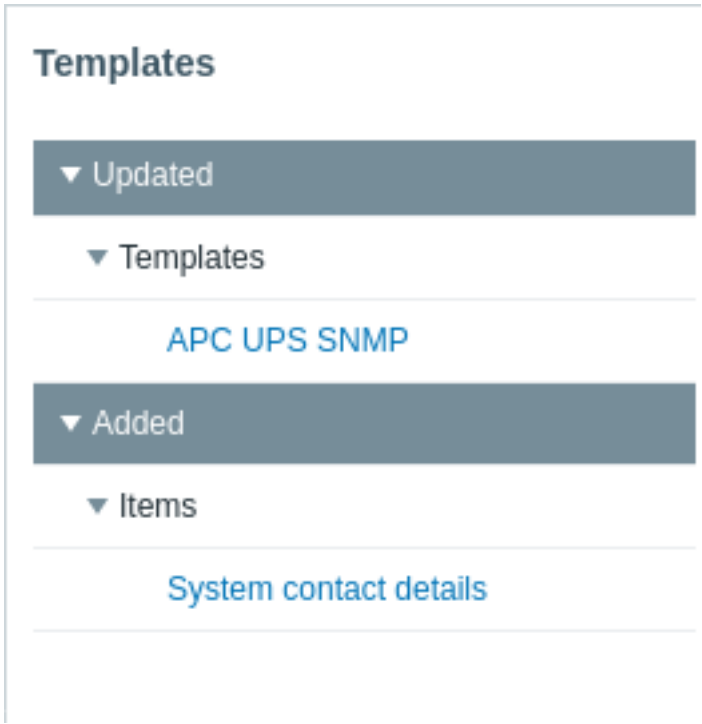
ルール	説明
既存の要素を更新	インポートファイルのデータを使用して既存の要素を更新します。それ以外の場合、既存の要素は更新されません。
新規作成	インポートファイルのデータを使用して新しい要素を作成します。それ以外の場合、新しい要素は作成されません。
存在しない要素を削除	インポートファイルに存在しない既存の要素を削除します。それ以外の場合、既存の要素は削除されません。 テンプレートのリンクに対して 存在しない要素を削除にチェックを入れた場合、インポートファイルに存在しない現在のテンプレートリンクは解除されます。リンクを解除したテンプレートから継承されたエンティティ（アイテム、トリガー、グラフなど）は削除されません（各エンティティでも 存在しない要素を削除オプションを選択した場合を除く）。

Attention:
同じ名前のテンプレートがすでに存在する場合、クリーンなインポートを実行するには、インポート時に 存在しない要素を削除オプションを選択する必要があります。これにより、更新されたテンプレートに存在しなくなった古いアイテムが削除されます（これらの古いアイテムの履歴が失われることに注意してください）。

次の画面では、インポートするテンプレートの内容を確認できます。新しいテンプレートの場合、すべての要素が緑色で表示されます。既存のテンプレートを更新する場合、新しいテンプレート要素は緑色で強調表示され、削除されたテンプレート要素は赤色で強調表示されます。変更されていない要素は灰色の背景に表示されます。



左側のメニューを使用して、変更一覧を移動できます。更新セクションには、既存のテンプレート要素に加えられたすべての変更が表示されます。追加セクションには、新しいテンプレート要素が一覧表示されます。各セクションの要素は要素タイプごとにグループ化されています。灰色の矢印をクリックすると、要素グループを展開または折りたたむことができます。



テンプレートの変更内容を確認し、インポートをクリックしてテンプレートをインポートします。インポートの成功または失敗を示すメッセージが Web インターフェースに表示されます。

エクスポート形式　テンプレートがエクスポートされると、Zabbix は構造化された YAML、JSON、または XML 形式を生成します。
エクスポートには、テンプレートのメタデータ、アイテム、マクロ、トリガー、ダッシュボードなどのテンプレート要素が含まれます。

各要素には特定の役割があり、入れ子の要素を含む場合があります。

以下のセクションでは、エクスポート形式における各要素について説明します。
例では Linux by Zabbix agent テンプレートを使用しています。

省略記号（`...`）は、簡潔にするために省略された要素を示します。
注記（`see table below`）は、後続のセクションでその要素がより詳しく説明されることを示します。

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (see table below)
  triggers: (see table below)
  graphs: (see table below)
```

テンプレートメタデータ

Element		Type	Description
version		string	(必須) エクスポートバージョン。
template_groups			(必須) テンプレートグループのルート要素。
	uuid	string	(必須) このテンプレートグループの一意の識別子。
	name	string	(必須) テンプレートグループ名。
host_groups			ホストプロトタイプで使用されるホストグループのルート要素。
	uuid	string	(必須) このホストグループの一意の識別子。
	name	string	(必須) ホストグループ名。
templates			テンプレート のルート要素。
triggers			テンプレートのトリガー要素のルート要素。これはテンプレートアイテムのトリガー と同じです。
graphs			テンプレートのグラフ要素のルート要素。これはホストのグラフ と同じです。

テンプレート

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix エージェント'
      name: 'Linux by Zabbix エージェント'
      description: |
        これは公式の Linux テンプレートです。Zabbix エージェント 8.0 以降が必要です。(...)
      wizard_ready: 'YES'
      readme: |
        ## 概要

        これは公式の Linux テンプレートです。Zabbix エージェント 8.0 以降が必要です。(...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'エージェントが利用不可と見なされるまでのタイムアウトです。Zabbix サーバー/プロキシから到達可能なエ
          config: (see table below)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'CPU 使用率の重大しきい値 (%) です。'
          config: (see table below)
        (...)
      dashboards: (see table below)
```

```
      valuemaps: (see table below)
    (...)
```

Element	Type	Description
uuid	string	(required) このテンプレートの一意の識別子。
template	string	(required) 一意のテンプレート名。
name	string	表示用のテンプレート名。
description	text	テンプレートの説明。
wizard_ready	text	ホストウィザードで選択可能かどうか。 可能な値:[1] NO (0, default), YES (1). 参照: Template object (`wizard_ready`)。
readme	text	ホストウィザード に表示する、テンプレート固有の設定手順。Markdown 書式に対応しています。
vendor		テンプレートベンダーのルート要素（エクスポートされたテンプレートにベンダーデータが含まれる場合に存在します）。
name	string	(required) テンプレートベンダー名。
version	string	(required) テンプレートのバージョン。 標準テンプレート では、バージョンは次の形式で表示されます: Zabbix のメジャーバージョン、区切り文字 (”-”)、リビジョン番号（テンプレートの新しいバージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされます）。例: 7.0-0, 7.0-3, 8.0-0, 8.0-3。
templates		リンクされたテンプレートのルート要素。
name	string	(required) テンプレート名。
groups		テンプレートグループのルート要素。
name	string	(required) テンプレートグループ名。
items		テンプレートアイテム のルート要素。
discovery_rules		テンプレートの低レベルディスカバリルール のルート要素。
httptests		テンプレートの Web シナリオ のルート要素。
tags		テンプレートタグのルート要素。
tag	string	(required) タグ名。
value	string	タグの値。
macros		テンプレートユーザーマクロのルート要素。
macro	string	(required) ユーザーマクロ名。
type	string	ユーザーマクロの種類。 可能な値:[1] TEXT (0, default), SECRET_TEXT (1), VAULT (2). 参照: User macro object (`type`)。
value	string	ユーザーマクロの値。
description	string	ユーザーマクロの説明。
config		テンプレートマクロ設定 のルート要素で、ホストウィザード でマクロをどのように表示するかを制御します。
dashboards		テンプレートダッシュボード のルート要素。
valuemaps		テンプレート値マップ のルート要素。

テンプレートアイテム

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix エージェントの ping'
          key: agent.ping
          description: 'エージェントはこのアイテムに対して常に "1" を返します。可用性チェックのために `nodata()` と組み合
          valuemap:
            name: 'Zabbix エージェントの ping ステータス'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'インストールされているパッケージ数'
          key: system.sw.packages.get
```

```
      delay: 1h
      preprocessing: （下表を参照）
      tags:
        - tag: component
          value: os
      triggers: （下表を参照）
    - uuid: 403cebed115441369e94d35d070ca7b8
      name: 'メモリ使用率'
      type: DEPENDENT
      key: vm.memory.utilization
      value_type: FLOAT
      units: '%'
      description: '使用中メモリの割合は `100-pavailable` として計算されます。'
      preprocessing: （下表を参照）
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: （下表を参照）
    (...)
```

要素	タイプ	説明
uuid	string	(必須) このアイテムの一意の識別子。
name	string	(必須) アイテム名。
type	string	アイテムのタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0、デフォルト)、TRAP (2)、SIMPLE (3)、INTERNAL (5)、ZABBIX_ACTIVE (7)、EXTERNAL (10)、ODBC (11)、IPMI (12)、SSH (13)、TELNET (14)、CALCULATED (15)、JMX (16)、SNMP_TRAP (17)、DEPENDENT (18)、HTTP_AGENT (19)、SNMP_AGENT (20)、ITEM_TYPE_SCRIPT (21)、ITEM_TYPE_BROWSER (22)、TELEMETRY_QUERY (24)。
snmp_oid	string	(SNMP_AGENT アイテムでは必須) SNMP オブジェクト ID。
key	string	(必須) アイテムキー。
delay	string	アイテムの更新間隔。 デフォルト: 1m。TRAP アイテムの場合、値は常に 0 です。
history	string	履歴データを保存する期間（時間のサフィックス、ユーザーマクロ、またはLLD マクロ を使用)。 デフォルト: 31d。
trends	string	トレンドデータを保存する期間（時間のサフィックス、ユーザーマクロ、またはLLD マクロ を使用)。 デフォルト: 365d。
status	string	アイテムのステータス。 指定可能な値:[1] ENABLED (0、デフォルト)、DISABLED (1)。
value_type	string	受信値のタイプ。 指定可能な値:[1] FLOAT (0)、CHAR (1)、LOG (2)、UNSIGNED (3、デフォルト)、TEXT (4)、BINARY (5)、JSON (6)。
allowed_hosts	string	アイテムのデータ送信を許可するホストの IP アドレスをカンマ区切りで指定したリスト。 TRAP および HTTP_AGENT アイテムでサポートされます。
units	string	受信値の単位（bps、B など)。
params	text	アイテムのタイプに応じた追加パラメータ（SSH および TELNET アイテムで実行するスクリプト、ODBC アイテムの SQL クエリ、CALCULATED アイテムの数式、ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムのスクリプト)。
ipmi_sensor	string	IPMI センサー。 IPMI アイテムでサポートされます。
authtype	string	認証タイプ。 SSH および HTTP_AGENT アイテムでサポートされます。 SSH アイテムで指定可能な値:[1] PASSWORD (0、デフォルト)、PUBLIC_KEY (1)。 HTTP_AGENT アイテムで指定可能な値:[1] NONE (0、デフォルト)、BASIC (1)、NTLM (2)、Kerberos (3)、または Digest (4)。

要素	タイプ	説明
username	string	(SSH および TELNET アイテムでは必須) 認証用のユーザー名。 SIMPLE、ODBC、JMX、および HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、password（下記参照）も指定するか、両方の要素を空欄にする必要があります。
password	string	(SSH および TELNET アイテムでは必須) 認証用のパスワード。 SIMPLE、ODBC、JMX、および HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、username（上記参照）も指定するか、両方の要素を空欄にする必要があります。
publickey	string	(SSH アイテムでは必須) 公開鍵ファイルの名前。
privatekey	string	(SSH アイテムでは必須) 秘密鍵ファイルの名前。
description	text	アイテムの説明。
inventory_link	string	アイテムによって値が設定されるホストインベントリフィールド。 指定可能な値:[1] NONE (0)、ALIAS (4) など（サポートされるフィールドについてはホストインベントリ を参照）。
valuemap		アイテムの値マップのルート要素。
name	string	(必須) アイテムに使用する値マップの名前。
logtimefmt	string	ログエントリの時刻形式。 LOG 値タイプのアイテムでサポートされます。
preprocessing		アイテム値の前処理のルート要素。
step		テンプレートアイテム値の前処理ステップのルート要素。
jmx_endpoint	string	JMX エンドポイント。 JMX アイテムでサポートされます。
master_item		(DEPENDENT アイテムでは必須) 依存アイテムのマスターアイテムのルート要素。
key	string	(必須) 依存アイテムのマスターアイテムキー。
timeout	string	アイテムデータのポーリング要求のタイムアウト。 タイムアウトリストにあるアイテムタイプでサポートされます。
url	string	(HTTP_AGENT アイテムでは必須) URL 文字列。
query_fields		クエリパラメータのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムでは必須) クエリパラメータ名。
value	string	クエリパラメータの値。 HTTP_AGENT アイテムでサポートされます。
parameters		ユーザー定義パラメータのルート要素。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
name	string	(ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでは必須) ユーザー定義パラメータ名。
value	string	ユーザー定義パラメータの値。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
posts	string	HTTP(S) リクエスト本文のデータ。 HTTP_AGENT アイテムでサポートされます。
status_codes	string	必要な HTTP ステータスコードの範囲をカンマ区切りで指定したもの。 HTTP_AGENT アイテムでサポートされます。
follow_redirects	string	データのポーリング中にレスポンスのリダイレクトに従うかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0)、YES (1、デフォルト)。
post_type	string	POST データ本文のタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0、デフォルト)、JSON (2)、XML (3)。
http_proxy	string	HTTP(S) プロキシ接続文字列。 HTTP_AGENT アイテムでサポートされます。
headers		HTTP(S) リクエストヘッダーのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムでは必須) ヘッダー名。
value	string	(HTTP_AGENT アイテムでは必須) ヘッダー値。
retrieve_mode	string	保存するレスポンスの部分。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] BODY (0、デフォルト)、HEADERS (1)、BOTH (2)。
request_method	string	リクエストメソッドのタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] GET (0、デフォルト)、POST (1)、PUT (2)、HEAD (3)。

要素	タイプ	説明
output_format	string	レスポンスの処理方法。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0、デフォルト)、JSON (1)。
allow_traps	string	trapper アイテムと同様の方法で値を設定できるようにするかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
ssl_cert_file	string	公開 SSL キーのファイルパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_file	string	秘密 SSL キーのファイルパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_password	string	SSL キーファイルのパスワード。 HTTP_AGENT アイテムでサポートされます。
verify_peer	string	ホストの証明書が正当なものであることを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
verify_host	string	接続先のホスト名がホストの証明書に記載されたものと一致することを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
time_shift	string	アイテムがクエリする時間範囲の終了時点を、現在時刻よりどれだけ過去にするか。 TELEMETRY_QUERY アイテムでサポートされます。
lookback_limit	string	継続元となる以前のクエリが存在しない場合（アイテムの初回実行時など）、または以前のクエリが `delay` より古い場合（Zabbix サーバーが停止していた場合など）に、アイテムがデータをクエリする過去の範囲。 TELEMETRY_QUERY アイテムでサポートされます。
granularity	string	クエリの全時間範囲を分割する各区間の長さ。 TELEMETRY_QUERY アイテムでサポートされます。
query		(TELEMETRY_QUERY アイテムでは必須) テレメトリクエリのルート要素。アイテムがデータソースからクエリする内容、データのフィルタリング方法、および結果の集計方法を定義します。
signal_type	string	(必須) アイテムがクエリする シグナルのタイプ。 TELEMETRY_QUERY アイテムでサポートされます。 指定可能な値:[1] TRACES (0)、METRICS (1)、LOGS (2)。
metric_point_type	string	(METRICS シグナルタイプでは必須) アイテムがクエリする メトリックポイントのタイプ。 TELEMETRY_QUERY アイテムでサポートされます。 指定可能な値:[1] SUM (0)、GAUGE (1)、HISTOGRAM (2)、EXPONENTIAL_HISTOGRAM (3)。
columns		(必須) アイテムがクエリするテレメトリデータ列のルート要素。 この要素には次の要素があります。 `column`（`string`）- (必須) アイテムがクエリするテレメトリデータ列の名前。 `attribute_key`（`string`）- (キーと値のペアを保持する列では必須) キーと値のペアを保持する列内で、アイテムがクエリするキーの名前。
aggregated_columns		(必須) アイテムがクエリするテレメトリデータの集計列のルート要素。 この要素には次の要素があります。 `column`（`string`）- (MIN、MAX、AVG、SUM、および PERCENTILE 関数では必須) アイテムがクエリするテレメトリデータ列の名前。 `function`（`string`）- (必須) アイテムが列のレコードに適用する関数。指定可能な値:[1] MIN (1)、MAX (2)、AVG (3)、COUNT (4)、SUM (5)、PERCENTILE (8)。 `parameters`（`array`）- (PERCENTILE 関数では必須) アイテムが列のレコードから計算するパーセンタイル。 `alias`（`string`）- (必須) 計算値を識別するためにアイテム値で使用する、一意の名前。

要素		タイプ	説明
	filter		(必須) クエリ内の列レコードに対するテレメトリクエリフィルターのルート要素。 この要素には次の要素があります。 `evaltype`（string）- (必須) フィルター条件の評価方法。指定可能な値:[1] AND_OR (0)、AND (1)、OR (2)、FORMULA (3)。 `formula`（string）- (FORMULA 評価方法では必須) カスタム式を使用してフィルター条件を評価するためのユーザー定義式。 `conditions`（string）- (必須) テレメトリデータ列内のレコードが満たす必要のあるテレメトリデータフィルター条件のルート要素。 `conditions` 要素には次の要素があります。 `column`（string）- (必須) フィルター条件を適用するテレメトリデータ列の名前。 `attribute_key`（string）- (キーと値のペアを保持する列では必須) キーと値のペアを保持する列内で、条件を適用するキーの名前。 `operator`（string）- (必須) フィルター条件の演算子。単一値の列で指定可能な値:[1] EQUAL (0)、NOT_EQUAL (1)、LIKE (2)、NOT_LIKE (3)。キーと値のペアを保持する列で指定可能な値:[1] EQUAL (0)、NOT_EQUAL (1)、EXISTS (12)。 `formulaid`（string）- (FORMULA 評価方法では必須) カスタム式から条件を参照するために使用する任意の一意の ID。 `value`（string）- (EQUAL、NOT_EQUAL、LIKE、および NOT LIKE 操作では必須) 比較対象の値。
tags			アイテムタグのルート要素。
	tag	string	(必須) タグ名。
	value	string	タグの値。
triggers			テンプレートアイテムのトリガーのルート要素。

Note:
関連項目: アイテムオブジェクト（同じ名前の該当するプロパティを参照）。

テンプレートアイテムの値の前処理ステップ

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Element	Type	Description
type	string	(必須) アイテム値の前処理ステップのタイプ。 可能な値:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, 計算値: 受信した値 - 直前の値), CHANGE_PER_SECOND (10, 計算値: (受信した値 - 直前の値)/(現在時刻 - 最後のチェック時刻)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(必須) アイテム値の前処理ステップのパラメーターのルート要素。
parameter	string	アイテム値の前処理ステップの個別パラメーター。
error_handler	string	前処理ステップの失敗時に使用するアクションタイプ。 可能な値:[1] ORIGINAL_ERROR (0, デフォルト), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	エラーハンドラーのパラメーター。

Note:
あわせて参照: アイテム前処理オブジェクト（対応する名前の関連プロパティを参照してください）。

テンプレートアイテムのトリガー

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: インストール済みパッケージ数が変更されました'
              priority: WARNING
              manual_close: 'YES'
              tags:
                - tag: scope
                  value: notice
        (...)
        - uuid: 403cebed115441369e94d35d070ca7b8
          (...)
          triggers:
            - uuid: cfd395b1cde74ef18a5e5f840bd5142a
              expression: 'min(/Linux by Zabbix agent/vm.memory.utilization,5m)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: メモリ使用率が高い'
              event_name: 'Linux: メモリ使用率が高い（5m 間で>{$MEMORY.UTIL.MAX}%）'
              priority: AVERAGE
              description: 'システムの空きメモリが不足しています。'
              dependencies:
                - name: 'Linux: 利用可能メモリ不足'
                  expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
              tags:
                - tag: scope
                  value: capacity
                - tag: scope
                  value: performance
        (...)
```

Element	Type	Description
uuid	string	(required) このトリガーの一意の識別子。
expression	string	(required) トリガー式。
recovery_mode	string	OK イベントを生成するための基準。 可能な値:[1] EXPRESSION (0, default), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	トリガーの復旧式。
correlation_mode	string	相関モード（イベント相関なし、またはタグによるイベント相関）。 可能な値:[1] DISABLED (0, default), TAG_VALUE (1).
correlation_tag	string	イベント相関に使用するタグ名。
name	string	(required) トリガー名。
event_name	string	イベント名。
opdata	string	運用データ。
url_name	string	トリガーに関連付けられた URL のラベル。
url	string	トリガーに関連付けられた URL。
status	string	トリガーのステータス。 可能な値:[1] ENABLED (0, default), DISABLED (1).
priority	string	トリガーの深刻度。 可能な値:[1] NOT_CLASSIFIED (0, default), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	トリガーの説明。
type	string	イベント生成タイプ（単一の障害イベント、または複数の障害イベント）。 可能な値:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	障害イベントの手動クローズ。 可能な値:[1] NO (0, default), YES (1).
dependencies		依存関係のルート要素。
name	string	(required) 依存トリガー名。
expression	string	(required) 依存トリガー式。
recovery_expression	string	依存トリガーの復旧式。
tags		トリガータグのルート要素。
tag	string	(required) タグ名。
value	string	タグ値。

Note:
こちらも参照: Trigger object（該当する名前のプロパティを参照してください）。

テンプレートの低レベルディスカバリルール

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'The discovery of network interfaces.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
テンプレートの低レベルディスカバリルール要素のほとんどは、テンプレートアイテムと同じです。以下の表では、テンプレートアイテムと異なる要素について説明します。

Element	Type	Description
type	string	アイテムタイプ。 可能な値:[1] ZABBIX_PASSIVE (0, default), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(必須) 低レベルディスカバリルールのキー。
filter		テンプレートの低レベルディスカバリルールフィルターのルート要素。
lifetime	string	期間（秒、time suffix、またはuser macroを使用)。この期間を過ぎると、もはや検出されなくなったリソースは削除されます。 デフォルト: 7d.
lifetime_type	string	失われた LLD リソースを削除するシナリオ。 可能な値: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	期間（秒、time suffix、またはuser macroを使用)。この期間を過ぎると、もはや検出されなくなったリソースは無効化されます。
enabled_lifetime_type	string	失われた LLD リソースを無効化するシナリオ。 可能な値: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		テンプレートアイテムプロトタイプ要素のルート要素。これはテンプレートアイテムと同じです。
trigger_prototypes		テンプレートトリガープロトタイプ要素のルート要素。これはテンプレートアイテムトリガーと同じです。
graph_prototypes		テンプレートグラフプロトタイプ要素のルート要素。これはホストグラフと同じです。
host_prototypes		テンプレートホストプロトタイプ要素のルート要素。これはホストと同じです。
parent_discovery_rule		低レベルディスカバリルールプロトタイプの親となる低レベルディスカバリルール（またはルールプロトタイプ）のルート要素。このプロパティは、参照先のルール/ルールプロトタイプの直接の子である LLD ルールプロトタイプであることを示します。
key	string	(必須) 親となる低レベルディスカバリルール（またはルールプロトタイプ）のキー。
master_item	string	(DEPENDENT ルールで必須) 依存ルールのマスターアイテムのルート要素。
lld_macro_paths		低レベルディスカバリルールのマクロパスのルート要素。
lld_macro	string	(必須) 低レベルディスカバリルールのマクロ名。
path	string	(必須) 値のセレクター。この値が対応するマクロに割り当てられます。
preprocessing		低レベルディスカバリルールの値の前処理のルート要素。
step		低レベルディスカバリルールの値の前処理ステップ要素のルート要素。これはテンプレートアイテムの値の前処理ステップと同じですが、使用可能な値はより少なくなります。参照先: LLD rule preprocessing object。
overrides		低レベルディスカバリルールのオーバーライドルールのルート要素。
name	string	(必須) 一意のオーバーライド名。
step	string	(必須) オーバーライドの一意の順序番号。
stop	string	一致した場合、次のオーバーライドの処理を停止します。
filter		テンプレートの低レベルディスカバリルールオーバーライドルールフィルター要素のルート要素。これはテンプレートの低レベルディスカバリルールフィルターと同じです。
operations		テンプレートの低レベルディスカバリルールオーバーライド操作のルート要素。

Note:
こちらも参照: LLD rule object（対応する名前の関連プロパティを参照してください)。

テンプレートの低レベルディスカバリルールのフィルター

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.NOT_MATCHES}'
```

```
            operator: NOT_MATCHES_REGEX
        (...)
      - uuid: 2bbdc79f082d4c618e01bec625e9c90a
        (...)
        filter:
          evaltype: AND
          conditions:
            - macro: '{#DEVNAME}'
              value: '{$VFS.DEV.DEVNAME.MATCHES}'
            - macro: '{#DEVNAME}'
              value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
              operator: NOT_MATCHES_REGEX
            - macro: '{#DEVTYPE}'
              value: disk
        (...)
```

Element	Type	Description
evaltype	string	フィルター条件の評価方法を上書きします。 可能な値:[1] AND_OR (0, default), AND (1), OR (2), FORMULA (3).
formula	string	フィルター条件のカスタム計算式。
conditions		フィルター条件のルート要素。
macro	string	(required) チェック対象の低レベルディスカバリマクロ名。
value	string	比較する値。
operator	string	条件演算子。 可能な値:[1] MATCHES_REGEX (8, default), NOT_MATCHES_REGEX (9).
formulaid	string	(required) カスタム式から条件を参照するために使用される任意の一意な ID。大文字のみ使用できます。この ID は、フィルター条件を変更する際にユーザーが指定する必要がありますが、その後取得する際には新たに生成されます。

Note:
あわせて参照: LLD rule filter object (該当する名前のプロパティを参照してください)。

テンプレートの低レベルディスカバリルールのオーバーライド操作

Element	Type	Description
operationobject	string	オーバーライド操作が適用されるオブジェクト。 可能な値:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	オーバーライド条件の演算子。 可能な値:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	オーバーライド条件の演算子に対する正規表現または文字列。
status	string	オーバーライド操作時のオブジェクトのステータス。
discover	string	ディスカバリの結果としてオブジェクトが追加されるかどうか。
delay	string	オーバーライド操作時にアイテムプロトタイプに設定される更新間隔。
history	string	オーバーライド操作時にアイテムプロトタイプに設定される履歴保存期間。
trends	string	オーバーライド操作時にアイテムプロトタイプに設定されるトレンド保存期間。
severity	string	オーバーライド操作時に設定されるトリガープロトタイプの深刻度。
tags		オーバーライド操作時にオブジェクトに設定されるタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグの値。
templates		オーバーライド操作時にホストプロトタイプにリンクされるテンプレートのルート要素。
name	string	(必須) テンプレート名。
inventory_mode	string	オーバーライド操作時に設定されるホストプロトタイプのインベントリモード。

Note:
参照: LLD rule override operation object (対応する名前のプロパティを参照してください)。

テンプレートの Web シナリオ

Element	Type	Description
uuid	string	(必須) この Web シナリオの一意の識別子。
name	string	(必須) Web シナリオ名。
delay	string	Web シナリオを実行する頻度（秒、time suffix、またはuser macroを使用)。 デフォルト: 1m。
attempts	integer	Web シナリオのステップを実行する試行回数。 指定可能な値:[1] 1-10（デフォルト: 1)。
agent	string	クライアントエージェント。Zabbix は選択したブラウザーになりすまします。 これは、Web サイトがブラウザーごとに異なるコンテンツを返す場合に便利です。 デフォルト: Zabbix。
http_proxy	string	Web シナリオで使用するプロキシ。形式: `http://[username[:password]@]proxy.example.com[:port]`
variables		シナリオのステップで使用できる Web シナリオ変数のルート要素。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信する HTTP ヘッダーのルート要素。 ヘッダーは、HTTP プロトコルで表示されるのと同じ構文で列挙する必要があります。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
status	string	Web シナリオの状態。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
authentication	string	認証方式。 指定可能な値:[1] NONE (0, デフォルト), BASIC (1), NTLM (2), Kerberos (3), Digest (4)。
http_user	string	BASIC (HTTP), NTLM, Kerberos, または Digest 認証で使用するユーザー名。
http_password	string	BASIC (HTTP), NTLM, Kerberos, または Digest 認証で使用するパスワード。
verify_peer	string	Web サーバーの SSL 証明書を検証します。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
verify_host	string	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証します。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
ssl_cert_file	string	クライアント認証に使用する SSL 証明書ファイル名（PEM 形式である必要があります)。
ssl_key_file	string	クライアント認証に使用する SSL 秘密鍵ファイル名（PEM 形式である必要があります)。
ssl_key_password	string	SSL 秘密鍵ファイルのパスワード。
steps		テンプレートの Web シナリオステップのルート要素（必須)。
tags		Web シナリオタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
関連情報: Web シナリオオブジェクト（該当する名前のプロパティを参照してください)。

テンプレートの Web シナリオステップ

Element	Type	Description
name	string	(必須) Web シナリオステップ名。
url	string	(必須) 監視対象の URL。
query_fields		クエリパラメータのルート要素（リクエスト実行時に URL へ追加される HTTP フィールドの配列)。
name	string	(必須) クエリパラメータ名。
value	string	クエリパラメータ値。
posts		HTTP POST 変数のルート要素（文字列（raw post data）または HTTP フィールドの配列(フォームフィールドデータ))。
name	string	(必須) Post フィールド名。
value	string	(必須) Post フィールド値。
variables		このステップの後に適用されるステップレベルの変数（マクロ）のルート要素。 変数値に `regex:` プレフィックスがある場合、その値は `regex:` プレフィックスの後に続く正規表現パターンに従って、このステップから返されたデータから抽出されます
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。

Element	Type	Description
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
follow_redirects	string	HTTP リダイレクトを追跡します。 可能な値:[1] NO (0), YES (1, デフォルト)。
retrieve_mode	string	HTTP レスポンスの取得モード。 可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
timeout	string	ステップ実行のタイムアウト（秒、time suffix、またはuser macroを使用)。 デフォルト: 15s。
required	string	レスポンスに含まれている必要があるテキスト（空の場合は無視されます)。
status_codes	string	許可される HTTP ステータスコードのカンマ区切りリスト（例: 200-201,210-299。空の場合は無視されます)。

Note:
こちらも参照してください: Web scenario step object（対応する名前の関連プロパティを参照してください)。

テンプレートマクロの設定

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: '最後に Zabbix エージェントが検出されてからの経過秒数'
            description: 'エージェントが利用不可と見なされるまでのタイムアウト。'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'CPU 使用率のしきい値'
            description: 'CPU 使用率の重大なしきい値を % で表します。0 から 100 までの範囲です。'
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'インターフェース制御'
            description: 'インターフェースの動作状態が "Link down" に変化したときにトリガーを発生させます。'
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Element	Type	Description
type	string	(必須) マクロ入力フィールドの種類。 可能な値:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	マクロ一覧におけるマクロの位置。
section_name	string	マクロがグループ化される折りたたみセクションのラベル。
label	string	(TEXT, LIST, CHECKBOX マクロで必須) マクロのラベル。
description	text	マクロのヘルプテキスト。Markdown 書式をサポートします。

Element	Type	Description
required	string	マクロを必須としてマークします。 可能な値:[1] NO (0), YES (1). TEXT および LIST マクロでサポートされます。
regex	string	テキストボックスフィールドでユーザー入力を検証するための正規表現。 TEXT マクロでサポートされます。
options		LIST 項目または CHECKBOX 値のルート要素。
value	string	(LIST マクロで必須) LIST 項目の値。
text	string	(LIST マクロで必須) LIST 項目のテキスト。
checked	string	(CHECKBOX マクロで必須) チェック済み状態を表す値。 可能な値:[1] 0 (false), 1 (true). CHECKBOX マクロでサポートされます。
unchecked	string	(CHECKBOX マクロで必須) 未チェック状態を表す値。 可能な値:[1] 0 (false), 1 (true). CHECKBOX マクロでサポートされます。

Note:
あわせて参照: マクロ設定オブジェクト (該当する名前のプロパティを参照してください)。

テンプレートダッシュボード

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Element		Type	Description
uuid		string	(必須) このダッシュボードの一意の識別子。
name		string	(必須) テンプレートダッシュボード名。
display period		integer	ダッシュボードページの表示期間。
auto_start		string	スライドショーの自動開始。 可能な値:[1] NO (0), YES (1, default).
pages			テンプレートダッシュボードページのルート要素。
	name	string	ページ名。
	display period	integer	スライドショーの間隔。
	sortorder	integer	ページの並び順。
widgets			テンプレートダッシュボードウィジェット のルート要素。

Note:
関連情報: Template dashboard object (該当する同名のプロパティを参照してください)。

テンプレートダッシュボードウィジェット

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
```

```
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                    - type: STRING
                      name: reference
                      value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Element	Type	Description
type	string	(required) ウィジェットの種類。
name	string	ウィジェット名。
x	integer	テンプレートダッシュボードの左端からの水平方向の位置。 可能な値:[1] 0-71.
y	integer	テンプレートダッシュボードの上端からの垂直方向の位置。 可能な値:[1] 0-63.
width	integer	ウィジェットの幅。 可能な値:[1] 1-72.
height	integer	ウィジェットの高さ。 可能な値:[1] 1-64.
hide_header	string	ウィジェットのヘッダーを非表示にする。 可能な値:[1] NO (0, デフォルト), YES (1).
fields		テンプレートダッシュボードウィジェットのフィールドのルート要素。
type	string	(required) ウィジェットフィールドの種類。 可能な値:[1] INTEGER (0), STRING (1), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(required) ウィジェットフィールド名。
value	mixed	(required) フィールドの種類に応じたウィジェットフィールドの値。

Note:
参照: Template dashboard widget object（対応する名前の関連プロパティを参照してください）。

テンプレートの値マップ

```
zabbix_export:
  (...)
```

```
templates:
  (...)
    valuemaps:
      - uuid: 8c048c6cca8248f2860c208e8db0f59e
        name: ifOperStatus
        mappings:
          - value: '0'
            newvalue: unknown
          - value: '1'
            newvalue: notpresent
          - value: '2'
            newvalue: down
          - value: '3'
            newvalue: lowerlayerdown
          - value: '4'
            newvalue: testing
          - value: '5'
            newvalue: dormant
          - value: '6'
            newvalue: up
      (...)
```

Element	Type	Description
uuid	string	(必須) この値マップの一意の識別子。
name	string	(必須) 値マップ名。
mapping		マッピングのルート要素。
type	string	マッピングの一致タイプ。 可能な値:[1] EQUAL (0, default), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	元の値。
newvalue	string	(必須) 元の値がマッピングされる値。

Note:
参照: Value map object (該当する名前のプロパティを参照してください)。

脚注 [1] 括弧内の API の整数値、たとえば ENABLED (0) は、参照用としてのみ記載されています。詳細については、表の項目内にあるリンク先の API オブジェクトページ、または各セクションの末尾を参照してください。

ホスト

概要

ホストは、多くの関連オブジェクトおよびオブジェクト間の関連付けとともにエクスポートされます。

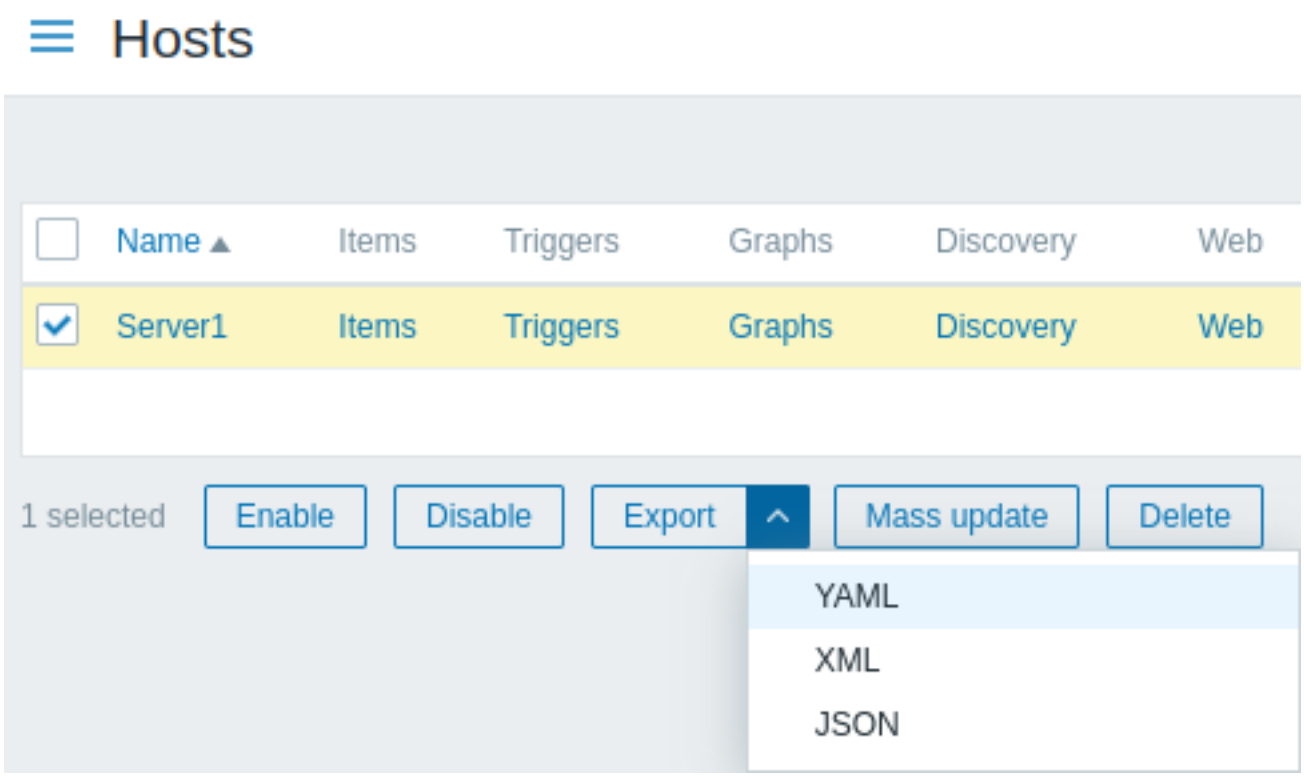
ホストのエクスポートには、次の内容が含まれます。

- リンクされたホストグループ
- ホストデータ
- テンプレートのリンク
- ホストグループのリンク
- ホストインターフェース
- 直接リンクされたアイテム
- 直接リンクされたトリガー
- すべてのプロトタイプを含む、直接リンクされたディスカバリルール
- 直接リンクされた Web シナリオ
- ホストマクロ
- ホストインベントリデータ
- 値マップ
- リンクされたグラフ

エクスポート

ホストをエクスポートするには、次の手順を実行します。

1. *Data collection* > ホストに移動します。
2. エクスポートするホストのチェックボックスを選択します。
3. 一覧の下にある *Export* をクリックすると、デフォルトの YAML 形式でエクスポートされます。別のエクスポート形式を選択するには、矢印をクリックします。



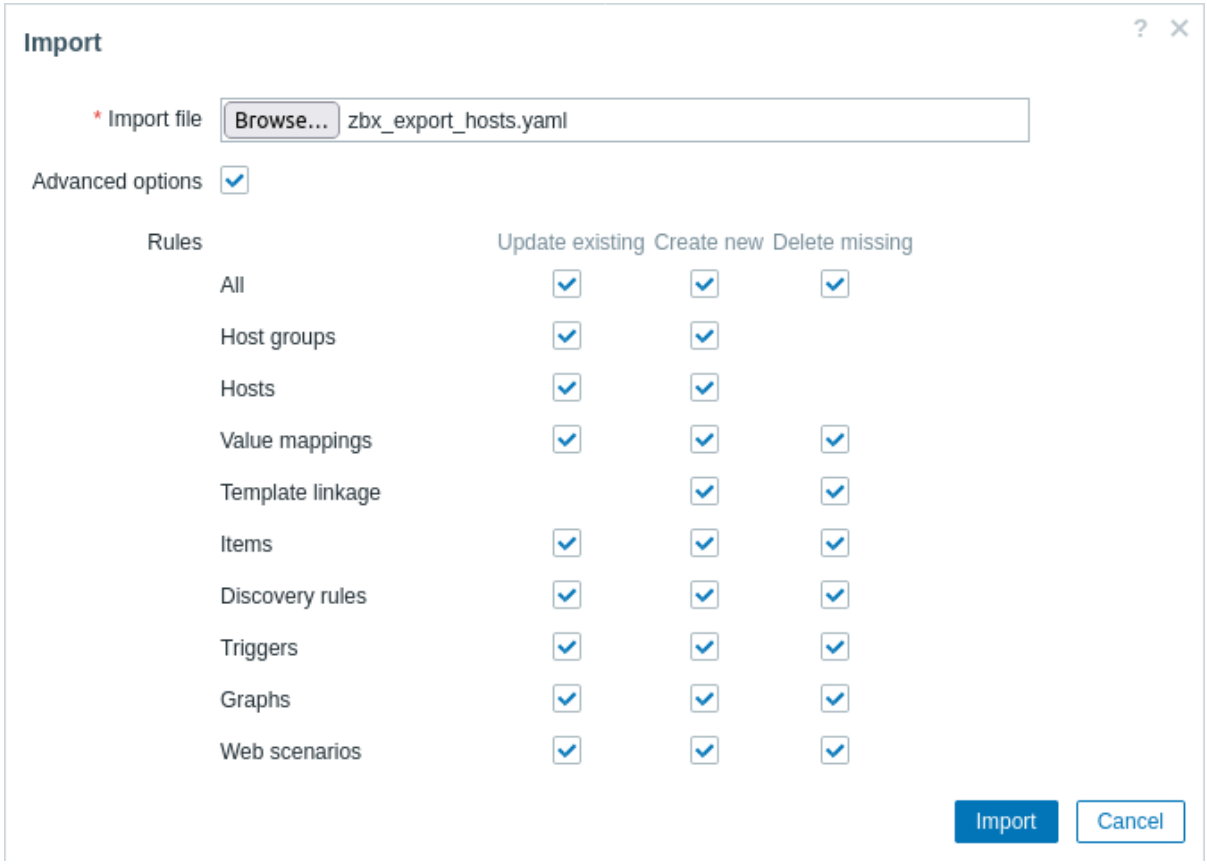
選択した形式に応じて、ホストは既定の名前でローカルファイルにエクスポートされます。

- `zabbix_export_hosts.yaml` - YAML エクスポート;
- `zabbix_export_hosts.xml` - XML エクスポート;
- `zabbix_export_hosts.json` - JSON エクスポート。

インポート

ホストをインポートするには、次の手順を実行します。

1. データ収集 > ホストに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. 適用するインポートルールのチェックボックスにチェックを入れます（以下を参照）。
5. 設定フォームの右下隅にある インポートをクリックします。



詳細オプションチェックボックスにチェックを入れると、インポート可能なすべての要素の詳細な一覧が表示されます。必要に応じて、各インポートルールにチェックを入れるか、チェックを外します。

すべて行のチェックボックスをクリックすると、その下にあるすべての要素にチェックが入るか、チェックが外れます。

インポートルール:

ルール	説明
既存を更新	既存の要素をインポートファイルのデータで更新します。選択しない場合、既存の要素は更新されません。
新規を作成	インポートファイルのデータを使用して新しい要素を作成します。選択しない場合、新しい要素は作成されません。
不足しているものを削除	インポートファイルに存在しない既存の要素を削除します。選択しない場合、既存の要素は削除されません。 テンプレートのリンクに対して 不足しているものを削除にチェックを入れた場合、インポートファイルに存在しない現在のテンプレートリンクは解除されます。リンクを解除したテンプレートから継承されたエンティティ（アイテム、トリガー、グラフなど）は削除されません（各エンティティに対しても 不足しているものを削除オプションを選択している場合を除く）。

インポート中にホストへの変更を確認することはできません。インポートの成功または失敗を示すメッセージが Web インターフェースに表示されます。

エクスポート形式

ホストをエクスポートすると、Zabbix は構造化された YAML、JSON、または XML 形式を生成します。
エクスポートには、ホストのメタデータ、アイテム、マクロ、トリガーなどのホスト要素が含まれます。

各要素には特定の役割があり、入れ子の要素を含む場合があります。

以下のセクションでは、エクスポート形式内の各要素について説明します。
例では、MySQL by Zabbix agent 2 テンプレートを持つホストを使用しています。
要素の説明のため、ホスト作成後にテンプレートのリンクを解除しています。
リンク解除では、すべての要素が設定内に保持されます（リンク解除してクリアした場合は要素が削除されます）。

省略記号（`...`）は、簡潔にするために省略された要素を示します。
注記（`see table below`）は、その要素が後続のセクションでより詳しく説明される場合に使用されます。

```
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (see table below)
  graphs: (see table below)
```

ホストメタデータ

Element		Type	Description
version		string	(必須) Zabbix のバージョン。
host_groups			(必須) ホストグループのルート要素。
	uuid	string	(必須) このホストグループの一意の識別子。
	name	string	(必須) ホストグループ名。
hosts			ホストのルート要素。
graphs			ホストグラフのルート要素。

ホスト

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL user password.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(必須) 一意のホスト名。
name	string	表示用ホスト名。
description	text	ホストの説明。
monitored_by	string	ホストの監視方法。 可能な値:[1] SERVER (0, default), PROXY (1), or PROXY_GROUP (2).
proxy		プロキシのルート要素。
name	string	(必須) ホストを監視するプロキシの名前（ある場合）。
proxy_group		プロキシグループのルート要素。
name	string	(必須) ホストの監視に使用するプロキシグループの名前（ある場合）。
status	string	ホストのステータス。 可能な値:[1] ENABLED (0, default), DISABLED (1).
ipmi_authtype	string	IPMI セッションの認証タイプ。 可能な値:[1] DEFAULT (-1, default), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).

Element	Type	Description
ipmi_privilege	string	IPMI セッションの権限レベル。 可能な値:[1] CALLBACK (1), USER (2, default), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	IPMI チェック用のユーザー名。
ipmi_password	string	IPMI チェック用のパスワード。
templates		リンクされたテンプレートのルート要素。
name	string	(必須) テンプレート名。
groups		ホストが所属するホストグループのルート要素。
name	string	(必須) ホストグループ名。
interfaces		ホストインターフェースのルート要素。
items		ホストアイテムのルート要素。
discovery_rules		ホストの低レベルディスカバリルールのルート要素。
http_tests		ホストの Web シナリオのルート要素。
tags		ホストタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグの値。
macros		ホストマクロのルート要素。
macro	string	(必須) ユーザーマクロ名。
type	string	ユーザーマクロのタイプ。 可能な値:[1] TEXT (0, default), SECRET_TEXT (1), VAULT (2).
value	string	ユーザーマクロの値。
description	string	ユーザーマクロの説明。
inventory		ホストインベントリのルート要素。
<inventory_property>	string	インベントリプロパティ。すべてのプロパティにはそれぞれ対応する要素があります (type, name, os など。例: Export format を参照)。
inventory_mode	string	インベントリモード。 可能な値:[1] DISABLED (-1), MANUAL (0, default), AUTOMATIC (1).
valuemaps		ホストの値マップのルート要素。

Note:
参照: Host object（該当する名前のプロパティを参照してください）。

ホストインターフェース

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	説明
default	string	これがプライマリホストインターフェースかどうか。1 つのホスト上で、同じ種類のプライマリインターフェースは 1 つしか持てないことに注意してください。 可能な値:[1] NO (0), YES (1, default).
type	string	インターフェースの種類。 可能な値:[1] ZABBIX (1, default), SNMP (2), IPMI (3), JMX (4).
useip	string	ホストへの接続に IP をインターフェースとして使用するかどうか（使用しない場合は DNS が使用されます）。 可能な値:[1] NO (0), YES (1, default).
ip	string	(IP 接続に必須) IP アドレス (IPv4 または IPv6)。
dns	string	(DNS 接続に必須) DNS 名。
port	string	ポート番号。
details		インターフェース詳細のルート要素。
version	string	この SNMP バージョンを使用します。 可能な値:[1] SNMPV1 (1), SNMP_V2C (2, default), SNMP_V3 (3).
community	string	(SNMPv1 および SNMPv2 アイテムに必須) SNMP community。

Element	Type	説明
max_repetitions	string	ネイティブ SNMP bulk リクエスト（GetBulkRequest-PDUs）における最大繰り返し値。 SNMPv2 および SNMPv3 アイテム（`discovery[]` および `walk[]` アイテム）でサポートされます。 デフォルト: 10。
contextname	string	SNMPv3 のコンテキスト名。 SNMPv3 アイテムでサポートされます。
securityname	string	SNMPv3 のセキュリティ名。 SNMPv3 アイテムでサポートされます。
securitylevel	string	SNMPv3 のセキュリティレベル。 SNMPv3 アイテムでサポートされます。 可能な値:[1] NOAUTHNOPRIV (0, default), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	SNMPv3 の認証プロトコル。 SNMPv3 アイテムでサポートされます。 可能な値:[1] MD5 (0, default), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	SNMPv3 の認証パスフレーズ。 SNMPv3 アイテムでサポートされます。
privprotocol	string	SNMPv3 のプライバシープロトコル。 SNMPv3 アイテムでサポートされます。 可能な値:[1] DES (0, default), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	SNMPv3 のプライバシーパスフレーズ。 SNMPv3 アイテムでサポートされます。
bulk	string	SNMP で bulk リクエストを使用するかどうか。 可能な値:[1] NO (0), YES (1, default).
interface_ref	string	アイテムで使用するインターフェース参照名（形式: `if<N>`）。

Note:
参照: ホストインターフェースオブジェクト（該当する名前のプロパティを参照してください）。

ホストアイテム

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: '`binlog_cache_size` より大きく、通常のバイナリログキャッシュに収まらないため、一時ディスクキャッ
          preprocessing: （下表を参照）
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Buffer pool utilization'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
          params: |
            ( last(//mysql.innodb_buffer_pool_pages_total) -
            last(//mysql.innodb_buffer_pool_pages_free) ) /
            ( last(//mysql.innodb_buffer_pool_pages_total) +
            ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
            ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
```

```
      description: 'バッファプール内の使用済みページ数と総ページ数の比率。'
      tags:
        - tag: component
          value: memory
      triggers: （下表を参照）
    (...)
    - name: Uptime
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'サーバーが稼働している秒数。'
      preprocessing: （下表を参照）
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: （下表を参照）
    (...)
```

Element	Type	Description
name	string	(必須) アイテム名。
type	string	アイテムのタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0、デフォルト)、TRAP (2)、SIMPLE (3)、INTERNAL (5)、ZABBIX_ACTIVE (7)、EXTERNAL (10)、ODBC (11)、IPMI (12)、SSH (13)、TELNET (14)、CALCULATED (15)、JMX (16)、SNMP_TRAP (17)、DEPENDENT (18)、HTTP_AGENT (19)、SNMP_AGENT (20)、ITEM_TYPE_SCRIPT (21)、ITEM_TYPE_BROWSER (22)、ITEM_TYPE_NESTED (23)、TELEMETRY_QUERY (24)。
snmp_oid	string	(SNMP_AGENT アイテムで必須) SNMP オブジェクト ID。
key	string	(必須) アイテムキー。
delay	string	アイテムの更新間隔。 デフォルト: 1m。TRAP アイテムの場合、値は常に 0 です。
history	string	履歴データを保存する期間（時間のサフィックス、ユーザーマクロ、またはLLD マクロ を使用）。 デフォルト: 31d。
trends	string	トレンドデータを保存する期間（時間のサフィックス、ユーザーマクロ、またはLLD マクロ を使用）。 デフォルト: 365d。
status	string	アイテムのステータス。 指定可能な値:[1] ENABLED (0、デフォルト)、DISABLED (1)。
value_type	string	受信値のタイプ。 指定可能な値:[1] FLOAT (0)、CHAR (1)、LOG (2)、UNSIGNED (3、デフォルト)、TEXT (4)、BINARY (5)、JSON (6)。
allowed_hosts	string	アイテムのデータ送信を許可するホストの、カンマ区切りの IP アドレス一覧。 TRAP および HTTP_AGENT アイテムでサポートされます。
units	string	受信値の単位（bps、B など）。
params	text	アイテムのタイプに応じた追加パラメータ（SSH および TELNET アイテムで実行するスクリプト、ODBC アイテムの SQL クエリ、CALCULATED アイテムの計算式、ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムのスクリプト）。
ipmi_sensor	string	IPMI センサー。 IPMI アイテムでサポートされます。
authtype	string	認証タイプ。 SSH および HTTP_AGENT アイテムでサポートされます。 SSH アイテムで指定可能な値:[1] PASSWORD (0、デフォルト)、PUBLIC_KEY (1)。 HTTP_AGENT アイテムで指定可能な値:[1] NONE (0、デフォルト)、BASIC (1)、NTLM (2)、Kerberos (3)、Digest (4)。
username	string	(SSH および TELNET アイテムで必須) 認証用のユーザー名。 SIMPLE、ODBC、JMX、および HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、password（下記参照）も指定するか、両方の要素を空白のままにする必要があります。

Element	Type	Description
password	string	(SSH および TELNET アイテムで必須) 認証用のパスワード。 SIMPLE、ODBC、JMX、および HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、`username`（上記参照）も指定するか、両方の要素を空白のままにする必要があります。
publickey	string	(SSH アイテムで必須) 公開鍵ファイルの名前。
privatekey	string	(SSH アイテムで必須) 秘密鍵ファイルの名前。
description	text	アイテムの説明。
inventory_link	string	アイテムによって値が設定されるホストインベントリフィールド。 指定可能な値:[1] NONE (0)、ALIAS (4) など（サポートされるフィールドについてはホストインベントリ を参照）。
valuemap		アイテムの値マップのルート要素。
name	string	(必須) アイテムで使用する値マップの名前。
logtimefmt	string	ログエントリの時刻形式。 LOG 値タイプのアイテムでサポートされます。
preprocessing		アイテム値の前処理のルート要素。
step		ホストアイテム値の前処理ステップのルート要素。
interface_ref	string	ホストインターフェースへの参照（形式: `if<N>`）。
jmx_endpoint	string	JMX エンドポイント。 JMX アイテムでサポートされます。
master_item		(DEPENDENT アイテムで必須) 依存アイテムのマスターアイテムのルート要素。
key	string	(必須) 依存アイテムのマスターアイテムキー。
timeout	string	アイテムデータのポーリング要求のタイムアウト。 アイテムタイプのタイムアウト リストでサポートされます。
url	string	(HTTP_AGENT アイテムで必須) URL 文字列。
query_fields		クエリパラメータのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムで必須) クエリパラメータ名。
value	string	クエリパラメータの値。 HTTP_AGENT アイテムでサポートされます。
parameters		ユーザー定義パラメータのルート要素。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
name	string	(ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムで必須) ユーザー定義パラメータ名。
value	string	ユーザー定義パラメータの値。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
posts	string	HTTP(S) リクエスト本文のデータ。 HTTP_AGENT アイテムでサポートされます。
status_codes	string	必要な HTTP ステータスコードの範囲（カンマ区切り）。 HTTP_AGENT アイテムでサポートされます。
follow_redirects	string	データのポーリング中にレスポンスのリダイレクトに従うかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0)、YES (1、デフォルト)。
post_type	string	POST データ本文のタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0、デフォルト)、JSON (2)、XML (3)。
http_proxy	string	HTTP(S) プロキシ接続文字列。 HTTP_AGENT アイテムでサポートされます。
headers		HTTP(S) リクエストヘッダーのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムで必須) ヘッダー名。
value	string	(HTTP_AGENT アイテムで必須) ヘッダー値。
retrieve_mode	string	保存するレスポンスの部分。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] BODY (0、デフォルト)、HEADERS (1)、BOTH (2)。
request_method	string	リクエストメソッドのタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] GET (0、デフォルト)、POST (1)、PUT (2)、HEAD (3)。
output_format	string	レスポンスの処理方法。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0、デフォルト)、JSON (1)。

Element	Type	Description
allow_traps	string	trapper アイテムと同じ方法で値を設定できるようにするかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
ssl_cert_file	string	公開 SSL 鍵ファイルのパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_file	string	秘密 SSL 鍵ファイルのパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_password	string	SSL 鍵ファイルのパスワード。 HTTP_AGENT アイテムでサポートされます。
verify_peer	string	ホストの証明書が正当なものであることを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
verify_host	string	接続先のホスト名がホストの証明書に記載されたものと一致することを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
time_shift	string	アイテムがクエリする時間範囲の終了時点を、現在時刻よりどれだけ過去にするか。 TELEMETRY_QUERY アイテムでサポートされます。
lookback_limit	string	継続元となる以前のクエリがない場合（アイテムの初回実行時など）、または以前のクエリが `delay` より古い場合（Zabbix サーバーが停止していた場合など）に、アイテムがデータをクエリする過去の期間。 TELEMETRY_QUERY アイテムでサポートされます。
granularity	string	クエリの全時間範囲を分割する各区間の長さ。 TELEMETRY_QUERY アイテムでサポートされます。
query		(TELEMETRY_QUERY アイテムで必須) テレメトリクエリのルート要素。データソースからアイテムがクエリする内容、データのフィルタリング方法、および結果の集計方法を定義します。
signal_type	string	(必須) アイテムがクエリする シグナル のタイプ。 TELEMETRY_QUERY アイテムでサポートされます。 指定可能な値:[1] TRACES (0)、METRICS (1)、LOGS (2)。
metric_point_type	string	(METRICS シグナルタイプで必須) アイテムがクエリする メトリックポイント のタイプ。 TELEMETRY_QUERY アイテムでサポートされます。 指定可能な値:[1] SUM (0)、GAUGE (1)、HISTOGRAM (2)、EXPONENTIAL_HISTOGRAM (3)。
columns		(必須) アイテムがクエリするテレメトリデータ列のルート要素。 この要素には次の要素があります: `column`（`string`）- (必須) アイテムがクエリするテレメトリデータ列の名前。 `attribute_key`（`string`）- (キーと値のペアを保持する列で必須) キーと値のペアを保持する列内で、アイテムがクエリするキーの名前。
aggregated_columns		(必須) アイテムがクエリするテレメトリデータの集計列のルート要素。 この要素には次の要素があります: `column`（`string`）- MIN、MAX、AVG、SUM、および PERCENTILE 関数で必須。アイテムがクエリするテレメトリデータ列の名前。 `function`（`string`）- (必須) アイテムが列のレコードに適用する関数。指定可能な値:[1] MIN (1)、MAX (2)、AVG (3)、COUNT (4)、SUM (5)、PERCENTILE (8)。 `parameters`（`array`）- PERCENTILE 関数で必須。アイテムが列のレコードから計算するパーセンタイル。 `alias`（`string`）- (必須) 計算値の一意の名前。アイテム値内でその値を識別するために使用されます。

Element	Type	Description
filter		(必須) クエリ内の列レコードに対するテレメトリクエリフィルターのルート要素。 この要素には次の要素があります: `evaltype`（`string`）- (必須) フィルター条件の評価方法。指定可能な値:[1] AND_OR (0)、AND (1)、OR (2)、FORMULA (3)。 `formula`（`string`）- FORMULA 評価方法で必須。カスタム式を使用してフィルター条件を評価するためのユーザー定義式。 `conditions`（`string`）- テレメトリデータ列内のレコードが満たす必要のあるテレメトリデータフィルター条件のルート要素。 `conditions` 要素には次の要素があります: `column`（`string`）- (必須) フィルター条件を適用するテレメトリデータ列の名前。 `attribute_key`（`string`）- (キーと値のペアを保持する列で必須) キーと値のペアを保持する列内で、条件を適用するキーの名前。 `operator`（`string`）- (必須) フィルター条件の演算子。単一値の列で指定可能な値:[1] EQUAL (0)、NOT_EQUAL (1)、LIKE (2)、NOT_LIKE (3)。キーと値のペアを保持する列で指定可能な値:[1] EQUAL (0)、NOT_EQUAL (1)、EXISTS (12)。 `formulaid`（`string`）- FORMULA 評価方法で必須。カスタム式から条件を参照するために使用する任意の一意の ID。 `value`（`string`）- EQUAL、NOT_EQUAL、LIKE、および NOT LIKE 操作で必須。比較対象の値。
tags		アイテムタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグの値。
triggers		ホストアイテムトリガーのルート要素。

Note:
関連項目: Item object（同じ名前の該当するプロパティを参照）。

ホストアイテム値の前処理ステップ

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

Element	Type	Description
type	string	(必須) アイテム値の前処理ステップのタイプ。 可能な値:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9, calculated: received value - previous value), CHANGE_PER_SECOND (10, calculated: (received value - previous value)/(time now - time of last check)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30).
parameters		(必須) アイテム値の前処理ステップのパラメーターのルート要素。
parameter	string	アイテム値の前処理ステップの個別パラメーター。
error_handler	string	前処理ステップの失敗時に使用するアクションタイプ。 可能な値:[1] ORIGINAL_ERROR (0, default), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	エラーハンドラーのパラメーター。

Note:
あわせて参照: Item preprocessing object (対応する名前の関連プロパティを参照してください)。

ホストアイテムのトリガー

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL: 情報データの取得に失敗しました'
              event_name: 'MySQL: 情報データの取得に失敗しました (または 30 分間データなし)'
              priority: INFO
              description: 'Zabbix は過去 30 分間、アイテムのデータを受信していません。'
              dependencies:
                - name: 'MySQL: サービスが停止しています'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: サービスが再起動されました'
              event_name: 'MySQL: サービスが再起動されました (稼働時間 < 10 分)'
              priority: INFO
              description: 'MySQL の稼働時間は 10 分未満です。'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Element	Type	Description
uuid	string	(必須) このトリガーの一意の識別子。
expression	string	(必須) トリガーの式。
recovery_mode	string	OK イベントを生成する基準。 可能な値:[1] EXPRESSION (0, default), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	トリガーの復旧式。

Element	Type	Description
correlation_mode	string	相関モード（イベント相関なし、またはタグによるイベント相関）。 可能な値:[1] DISABLED (0, default), TAG_VALUE (1).
correlation_tag	string	イベント相関に使用するタグ名。
name	string	(必須) トリガー名。
event_name	string	イベント名。
opdata	string	運用データ。
url_name	string	トリガーに関連付けられた URL のラベル。
url	string	トリガーに関連付けられた URL。
status	string	トリガーの状態。 可能な値:[1] ENABLED (0, default), DISABLED (1).
priority	string	トリガーの深刻度。 可能な値:[1] NOT_CLASSIFIED (0, default), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	トリガーの説明。
type	string	イベント生成タイプ（単一の障害イベント、または複数の障害イベント）。 可能な値:[1] SINGLE (0, default), MULTIPLE (1).
manual_close	string	障害イベントの手動クローズ。 可能な値:[1] NO (0, default), YES (1).
dependencies		依存関係のルート要素。
name	string	(必須) 依存トリガー名。
expression	string	(必須) 依存トリガーの式。
recovery_expression	string	依存トリガーの復旧式。
tags		トリガータグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
こちらも参照: Trigger object（該当する名前のプロパティを参照してください）。

ホストの低レベルディスカバリルール

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'DBMS 内のデータベースをスキャンします。'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'データベースサイズ。'
              preprocessing:
                - type: DISCARD_UNCHANGED_HEARTBEAT
                  parameters:
                    - 1h
              interface_ref: if1
              tags:
                - tag: component
                  value: storage
                - tag: database
                  value: '{#DATABASE}'
          lld_macro_paths:
            - lld_macro: '{#DATABASE}'
```

```
            path: $.Database
          preprocessing:
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 1d
        (...)
```

Attention:
ホストの低レベルディスカバリルールの要素のほとんどは、ホストのアイテムと同じです。以下の表では、ホストのアイテムと異なる要素について説明します。

Element	Type	Description
type	string	アイテムタイプ。 可能な値:[1] ZABBIX_PASSIVE (0, default), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22).
key	string	(必須) 低レベルディスカバリルールのキー。
filter		ホストの低レベルディスカバリルールフィルターのルート要素。
lifetime	string	期間（秒、time suffix、またはuser macroを使用）。この期間を過ぎると、もはや検出されなくなったリソースは削除されます。 デフォルト: 7d.
lifetime_type	string	失われた LLD リソースを削除するシナリオ。 可能な値: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	期間（秒、time suffix、またはuser macroを使用）。この期間を過ぎると、もはや検出されなくなったリソースは無効化されます。
enabled_lifetime_type	string	失われた LLD リソースを無効化するシナリオ。 可能な値: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		ホストのアイテムプロトタイプ要素のルート要素で、ホストのアイテムと同じです。
trigger_prototypes		ホストのトリガープロトタイプ要素のルート要素で、ホストのアイテムトリガーと同じです。
graph_prototypes		ホストのグラフプロトタイプのルート要素で、ホストのグラフと同じです。
host_prototypes		ホストプロトタイプのルート要素で、ホストと同じです。
parent_discovery_rule		低レベルディスカバリルールプロトタイプの親となる低レベルディスカバリルール（またはルールプロトタイプ）のルート要素です。このプロパティは、参照先のルール/ルールプロトタイプの直接の子である LLD ルールプロトタイプであることを示します。
key	string	(必須) 親の低レベルディスカバリルール（またはルールプロトタイプ）のキー。
master_item	string	(DEPENDENT ルールで必須) 依存ルールのマスターアイテムのルート要素。
lld_macro_paths		低レベルディスカバリルールのマクロパスのルート要素。
lld_macro	string	(必須) 低レベルディスカバリルールのマクロ名。
path	string	(必須) 対応するマクロに割り当てる値のセレクター。
preprocessing		低レベルディスカバリルールの値の前処理のルート要素。
step		低レベルディスカバリルールの値の前処理ステップ要素のルート要素で、ホストのアイテムの値の前処理ステップと同じですが、使用可能な値はより少なくなります。参照先: LLD rule preprocessing object。
overrides		低レベルディスカバリルールのオーバーライドルールのルート要素。
name	string	(必須) 一意のオーバーライド名。
step	string	(必須) オーバーライドの一意の順序番号。
stop	string	一致した場合、次のオーバーライドの処理を停止します。
filter		ホストの低レベルディスカバリルールフィルターと同じ、低レベルディスカバリルールのオーバーライドルールフィルター要素のルート要素。
operations		ホストの低レベルディスカバリルールのオーバーライド操作のルート要素。

Note:
こちらも参照: LLD rule object（該当する名前のプロパティを参照してください）。

ホストの低レベルディスカバリルールのフィルター

```
zabbix_export:
  (...)
```

```
hosts:
  - host: 'MySQL server'
    (...)
    discovery_rules:
      - name: 'Database discovery'
        (...)
        filter:
          evaltype: AND
          conditions:
            - macro: '{#DATABASE}'
              value: '{$MYSQL.DBNAME.MATCHES}'
            - macro: '{#DATABASE}'
              value: '{$MYSQL.DBNAME.NOT_MATCHES}'
              operator: NOT_MATCHES_REGEX
        (...)
```

Element	Type	Description
evaltype	string	フィルター条件の評価方法を上書きします。 使用可能な値:[1] AND_OR (0, default), AND (1), OR (2), FORMULA (3).
formula	string	フィルター条件のカスタム計算式。
conditions		フィルター条件のルート要素。
macro	string	(required) チェック対象となる低レベルディスカバリマクロの名前。
value	string	比較する値。
operator	string	条件演算子。 使用可能な値:[1] MATCHES_REGEX (8, default), NOT_MATCHES_REGEX (9).
formulaid	string	(required) カスタム式から条件を参照するために使用される、任意の一意な ID。大文字の英字のみを含めることができます。フィルター条件を変更する際にはユーザーがこの ID を指定する必要がありますが、その後に取得する際には新たに生成されます。

Note:
あわせて参照: LLD rule filter object (対応する名前のプロパティを参照してください)。

ホストの低レベルディスカバリルールのオーバーライド操作

Element	Type	Description
operationobject	string	オーバーライド操作が適用されるオブジェクト。 可能な値:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4).
operator	string	オーバーライド条件の演算子。 可能な値:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	オーバーライド条件の演算子に対する正規表現または文字列。
status	string	オーバーライド操作時のオブジェクトのステータス。
discover	string	ディスカバリの結果としてオブジェクトが追加されるかどうか。
delay	string	オーバーライド操作時にアイテムプロトタイプに設定される更新間隔。
history	string	オーバーライド操作時にアイテムプロトタイプに設定される履歴保存期間。
trends	string	オーバーライド操作時にアイテムプロトタイプに設定されるトレンド保存期間。
severity	string	オーバーライド操作時に設定されるトリガープロトタイプの深刻度。
tags		オーバーライド操作時にオブジェクトに設定されるタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグの値。
templates		オーバーライド操作時にホストプロトタイプにリンクされるテンプレートのルート要素。
name	string	(必須) テンプレート名。
inventory_mode	string	オーバーライド操作時に設定されるホストプロトタイプのインベントリモード。

Note:
参照: LLD rule override operation object (対応する名前の関連プロパティを参照してください)。

ホストの Web シナリオ

Element	Type	Description
uuid	string	(required) この Web シナリオの一意の識別子。
name	string	(required) Web シナリオ名。
delay	string	Web シナリオを実行する頻度（秒、time suffix、またはuser macroを使用）。 Default: 1m.
attempts	integer	Web シナリオのステップを実行する試行回数。 Possible values: 1-10 (default: 1).
agent	string	クライアントエージェント。Zabbix は選択したブラウザになりすまします。 これは、Web サイトがブラウザごとに異なるコンテンツを返す場合に便利です。 Default: Zabbix.
http_proxy	string	Web シナリオで使用するプロキシ。形式: `http://[username[:password]@]proxy.example.com[:port]`
variables		Web シナリオのステップで使用できる変数のルート要素。
name	string	(required) 変数名。
value	text	(required) 変数値。
headers		リクエスト実行時に送信する HTTP ヘッダーのルート要素。 ヘッダーは HTTP プロトコルに表示されるのと同じ構文で列挙してください。
name	string	(required) ヘッダー名。
value	text	(required) ヘッダー値。
status	string	Web シナリオの状態。 Possible values:[1] ENABLED (0, default), DISABLED (1).
authentication	string	認証方式。 Possible values:[1] NONE (0, default), BASIC (1), NTLM (2), Kerberos (3), or Digest (4).
http_user	string	BASIC (HTTP)、NTLM、Kerberos、または Digest 認証で使用するユーザー名。
http_password	string	BASIC (HTTP)、NTLM、Kerberos、または Digest 認証で使用するパスワード。
verify_peer	string	Web サーバーの SSL 証明書を検証します。 Possible values:[1] NO (0, default), YES (1).
verify_host	string	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証します。 Possible values:[1] NO (0, default), YES (1).
ssl_cert_file	string	クライアント認証に使用する SSL 証明書ファイル名（PEM 形式である必要があります）。
ssl_key_file	string	クライアント認証に使用する SSL 秘密鍵ファイル名（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵ファイルのパスワード。
steps		(required) ホストの Web シナリオステップ のルート要素。
tags		Web シナリオタグのルート要素。
tag	string	(required) タグ名。
value	string	タグ値。

Note:
See also: Web scenario object (refer to the relevant property with a matching name).

ホストの Web シナリオステップ

Element	Type	Description
name	string	(必須) Web シナリオステップ名。
url	string	(必須) 監視対象の URL。
query_fields		クエリパラメータのルート要素（リクエスト実行時に URL へ追加される HTTP フィールドの配列）。
name	string	(必須) クエリパラメータ名。
value	string	クエリパラメータ値。
posts		HTTP POST 変数のルート要素（文字列（raw post data）または HTTP フィールドの配列（フォームフィールドデータ））。
name	string	(必須) POST フィールド名。
value	string	(必須) POST フィールド値。
variables		このステップの後に適用されるステップレベル変数（マクロ）のルート要素。 変数値に `regex:` プレフィックスがある場合、その値は `regex:` プレフィックスの後に続く正規表現パターンに従って、このステップから返されたデータから抽出されます
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。

Element	Type	Description
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
follow_redirects	string	HTTP リダイレクトを追従します。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
retrieve_mode	string	HTTP レスポンスの取得モード。 指定可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
timeout	string	ステップ実行のタイムアウト（秒、time suffix、またはuser macroを使用)。 デフォルト: 15s。
required	string	レスポンスに含まれている必要があるテキスト（空の場合は無視されます)。
status_codes	string	受け入れ可能な HTTP ステータスコードのカンマ区切りリスト（例: 200-201,210-299。 空の場合は無視されます)。

Note:
こちらも参照: Web scenario step object（対応する名前のプロパティを参照してください)。

ホストグラフ

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (see table below)
    (...)
```

Element	Type	Description
uuid	string	グラフの一意の識別子。
name	string	(必須) グラフ名。
width	integer	グラフの幅（ピクセル単位)。 プレビューおよび円グラフ/分割円グラフで使用されます。 指定可能な値:[1] 20-65535 (default: 900).
height	integer	グラフの高さ（ピクセル単位)。 プレビューおよび円グラフ/分割円グラフで使用されます。 指定可能な値:[1] 20-65535 (default: 900).
yaxismin	double	Y 軸の最小値。 Y 軸の FIXED 最小値でサポートされます。 デフォルト: 0。
yaxismax	double	Y 軸の最大値。 X 軸の FIXED 最小値でサポートされます。 デフォルト: 0。
show_work_period	string	非稼働時間を強調表示します。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] NO (0), YES (1, default).
show_triggers	string	単純なトリガー値を線として表示します。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] NO (0), YES (1, default).
type	string	グラフの種類。 指定可能な値:[1] NORMAL (0, default), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	グラフの凡例を表示します。 指定可能な値:[1] NO (0), YES (1, default).
show_3d	string	3D スタイルを有効にします。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] NO (0, default), YES (1).
percent_left	double	左軸のパーセンタイル線を表示します。 NORMAL グラフでサポートされます。 デフォルト: 0。
percent_right	double	右軸のパーセンタイル線を表示します。 NORMAL グラフでサポートされます。 デフォルト: 0。

Element	Type	Description
ymin_type_1	string	Y 軸の最小値。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] CALCULATED (0, default), FIXED (1), ITEM (2).
ymin_item_1		(ymin_type_1 が ITEM に設定されている場合に必須) 個別アイテムの詳細のルート要素。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムキー。
ymax_type_1	string	Y 軸の最大値。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] CALCULATED (0, default), FIXED (1), ITEM (2).
ymax_item_1		(ymax_type_1 が ITEM に設定されている場合に必須) 個別アイテムの詳細のルート要素。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムキー。
graph_items		(必須) ホストグラフアイテム のルート要素。

Note:
参照: Graph object（対応する名前の関連プロパティを参照してください）。

ホストグラフのアイテム

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items:
        - drawtype: GRADIENT_LINE
          color: 199C0D
          item:
            host: 'MySQL server'
            key: mysql.bytes_received.rate
        - sortorder: '1'
          drawtype: GRADIENT_LINE
          color: F63100
          item:
            host: 'MySQL server'
            key: mysql.bytes_sent.rate
    (...)
```

Element	Type	Description
sortorder	integer	描画順序。値が小さいものから先に描画されます。線や領域を他の線や領域の背後に、または前面に描画するために使用できます。
drawtype	string	グラフアイテムの描画スタイル。 NORMAL グラフでサポートされます。 指定可能な値:[1] SINGLE_LINE (0, default), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color	string	要素の色（6 文字の 16 進数）。
yaxisside	string	グラフアイテムの Y 軸スケールを描画するグラフの側。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] LEFT (0, default), RIGHT (1).
calc_fnc	string	アイテムに複数の値が存在する場合に描画するデータ。 指定可能な値:[1] MIN (1), AVG (2, default), MAX (4), ALL (7; 最小値、平均値、最大値。simple graphs でサポート), LAST (9, pie/exploded graphs でサポート).
type	string	グラフアイテムのタイプ。 指定可能な値:[1] SIMPLE (0, default), GRAPH_SUM (2; アイテムの値が円グラフ全体を表す。pie/exploded graphs でサポート).
item		(必須) 個別のアイテム。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムのキー。

Note:
こちらも参照: Graph item object（該当する名前のプロパティを参照してください）。

ホストの値マップ

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element	Type	Description
uuid	string	(required) この値マップの一意な識別子。
name	string	(required) 値マップ名。
mapping		マッピングのルート要素。
type	string	マッピングの一致タイプ。 Possible values:[1] EQUAL (0, default), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	元の値。
newvalue	string	(required) 元の値がマッピングされる先の値。

Note:
See also: Value map object (refer to the relevant property with a matching name).

脚注 [1] 角括弧内の API の整数値、たとえば ENABLED (0) は、参考情報としてのみ記載されています。詳細については、表の項目内にあるリンク先の API オブジェクトページ、または各セクションの末尾を参照してください。

ネットワークマップ

概要

ネットワークマップのエクスポートには、以下が含まれます。

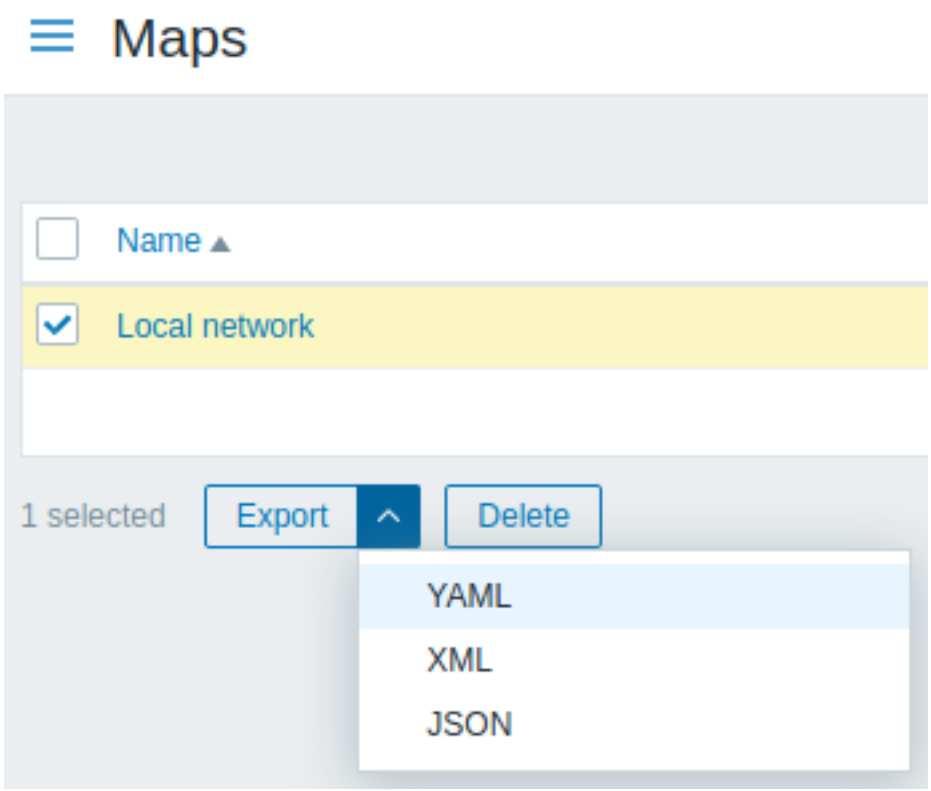
- 関連するすべての画像
- マップ構造（すべてのマップ設定、設定を含むすべての要素、マップリンク、およびマップリンクのステータスインジケーター）

Warning:
エクスポートされたマップに関連する可能性のあるホストグループ、ホスト、トリガー、他のマップ、またはその他の要素はエクスポートされません。そのため、マップが参照する要素の少なくとも 1 つが欠落している場合、インポートは失敗します。

エクスポート

ネットワークマップをエクスポートするには、次の手順を実行します。

1. *Monitoring* > *Maps* に移動します。
2. エクスポートするネットワークマップのチェックボックスをオンにします。
3. 一覧の下にある *Export* をクリックして、デフォルトの YAML 形式でエクスポートするか、矢印をクリックして別のエクスポート形式を選択します。



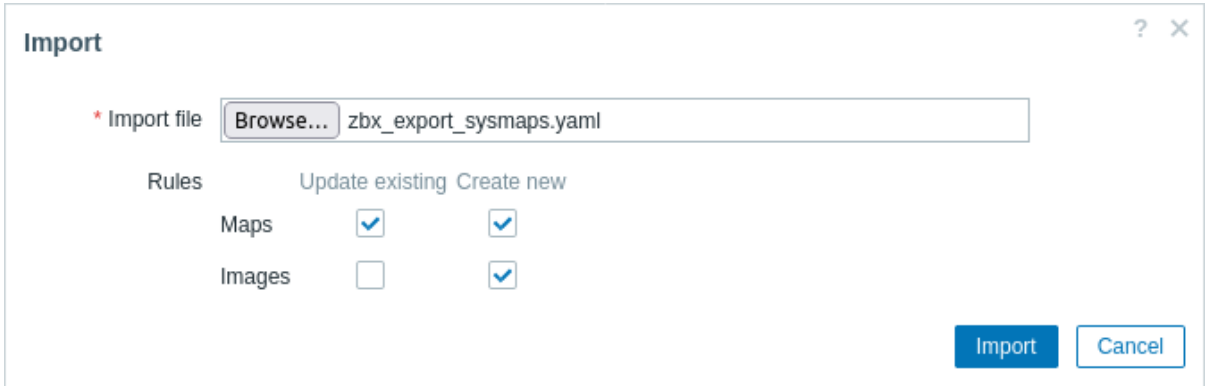
選択した形式に応じて、マップは既定の名前でローカルファイルにエクスポートされます。

- `zabbix_export_maps.yaml` - YAML でのエクスポート
- `zabbix_export_maps.xml` - XML でのエクスポート
- `zabbix_export_maps.json` - JSON でのエクスポート

インポート

ネットワークマップをインポートするには、次の手順を実行します。

1. 監視 > マップに移動します。
2. 右上隅にある インポートをクリックします。
3. インポートファイルを選択します。
4. 適用するインポートルールのチェックボックスにチェックを入れます（以下を参照）。
5. 設定フォームの右下隅にある インポートをクリックします。



インポートルール:

ルール	説明
既存のものを更新	既存のマップをインポートファイルのデータで更新します。選択しない場合、更新されません。
新規作成	インポートファイルのデータを使用して新しいマップを作成します。選択しない場合、作成されません。

マップに関する両方のオプションのチェックを外し、画像に関する該当するオプションにチェックを入れると、画像のみがインポートされます。画像のインポートは、特権管理者ユーザーのみが利用できます。

インポート中にネットワークマップへの変更を確認することはできません。インポートの成功または失敗を示すメッセージが Web インターフェースに表示されます。

Warning:
既存の画像を置き換えると、その画像を使用しているすべてのマップに影響します。

エクスポート形式

YAML にエクスポート:

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
      label_string_trigger: ''
      label_string_map: ''
      label_string_image: ''
      expand_macros: '1'
      background: {  }
      iconmap: {  }
      urls: {  }
      selements:
        - elementtype: '0'
          elements:
            - host: 'Zabbix server'
          label: |
            {HOST.NAME}
            {HOST.CONN}
          label_location: '0'
          x: '111'
          'y': '61'
          elementsubtype: '0'
          areatype: '0'
          width: '200'
          height: '200'
          viewtype: '0'
          use_iconmap: '0'
          selementid: '1'
          icon_off:
            name: Zabbix_server_3D_(128)
          icon_on: {  }
```

```
        icon_disabled: {  }
        icon_maintenance: {  }
        urls: {  }
        evaltype: '0'
    shapes:
      - type: '0'
        x: '0'
        'y': '0'
        width: '680'
        height: '15'
        text: '{MAP.NAME}'
        font: '9'
        font_size: '11'
        font_color: '000000'
        text_halign: '0'
        text_valign: '0'
        border_type: '0'
        border_width: '0'
        border_color: '000000'
        background_color: ''
        zindex: '0'
    lines: {  }
    links: {  }
```

エクスポートされた要素

エクスポートされた要素については、以下の表で説明します。

Element		Type	Description
images			画像のルート要素。
	name	string	一意の画像名。
	imagetype	integer	画像タイプ。 指定可能な値: 1 - 画像; 2 - 背景。
	encodedImage	string	Base64 でエンコードされた画像。
maps			maps のルート要素。

Maps

Element	Type	Description
name	string	一意のマップ名。
width	integer	マップの幅（ピクセル）。
height	integer	マップの高さ（ピクセル）。
label_type	integer	マップ要素のラベルタイプ。 可能な値: 0 - ラベル; 1 - ホスト IP アドレス; 2 - 要素名; 3 - ステータスのみ; 4 - なし。
label_location	integer	マップ要素ラベルの既定の表示位置。 可能な値: 0 - 下; 1 - 左; 2 - 右; 3 - 上。
highlight	integer	アクティブなトリガーとホストの状態に対するアイコンの強調表示を有効にします。 可能な値: 0 - いいえ; 1 - はい。

Element	Type	Description
expandproblem	integer	単一の障害がある要素について、障害トリガーを表示します。 可能な値: 0 - いいえ; 1 - はい。
markelements	integer	最近ステータスが変化したマップ要素を強調表示します。 可能な値: 0 - いいえ; 1 - はい。
show_unack	integer	障害の表示。 可能な値: 0 - すべての障害の件数; 1 - 未確認の障害の件数; 2 - 確認済みと未確認の障害の件数を個別に表示。
severity_min	integer	既定でマップに表示する最小のトリガー深刻度。 可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。
show_suppressed	integer	ホストメンテナンスのために抑制される（表示されない）障害を表示します。 可能な値: 0 - いいえ; 1 - はい。
grid_size	integer	マップグリッドのセルサイズ（ピクセル）。 `grid_show` が 0 に設定されている場合にサポートされます。 可能な値: 20, 40, 50, 75, 100。
grid_show	integer	マップ設定でグリッドを表示します。 可能な値: 0 - はい; 1 - いいえ。
grid_align	integer	マップ設定でアイコンを自動的に整列します。 可能な値: 0 - はい; 1 - いいえ。
label_format	integer	高度なラベル設定を使用します。 可能な値: 0 - いいえ; 1 - はい。
label_type_host	integer	ラベルをホストラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 可能な値: 0 - ラベル; 1 - ホスト IP アドレス; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_hostgroup	integer	ラベルをホストグループラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 可能な値: 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。

Element	Type	Description
label_type_trigger	integer	ラベルをトリガーラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 可能な値: 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_map	integer	ラベルをマップラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 可能な値: 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_image	integer	ラベルを画像ラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 可能な値: 0 - ラベル; 2 - 要素名; 4 - なし; 5 - カスタムラベル。
label_string_host	string	ホスト要素のカスタムラベル。 `label_type_host` が 5 に設定されている場合にサポートされます。
label_string_hostgroup	string	ホストグループ要素のカスタムラベル。 `label_type_hostgroup` が 5 に設定されている場合にサポートされます。
label_string_trigger	string	トリガー要素のカスタムラベル。 `label_type_trigger` が 5 に設定されている場合にサポートされます。
label_string_map	string	マップ要素のカスタムラベル。 `label_type_map` が 5 に設定されている場合にサポートされます。
label_string_image	string	画像要素のカスタムラベル。 `label_type_image` が 5 に設定されている場合にサポートされます。
expand_macros	integer	マップ設定のラベル内のマクロを展開します。 可能な値: 0 - いいえ; 1 - はい。
background		背景画像のルート要素（ある場合）。 `imagetype` が 2 に設定されている場合にサポートされます。
name	string	背景画像名。
iconmap		アイコンマッピングのルート要素（ある場合）。
name	string	アイコンマッピング名。
urls		マップまたは各マップ要素で使用される URL のルート要素。
name	string	リンク名。
url	string	リンク URL。
elementtype	integer	リンクが属するマップアイテムのタイプ。 可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - 画像。
selements		マップ selements のルート要素。
shapes		マップ図形のルート要素。
type	integer	図形タイプ。 可能な値: 0 - 長方形; 1 - 楕円。
x	integer	図形の X 座標（ピクセル）。
y	integer	図形の Y 座標（ピクセル）。
width	integer	図形の幅。

Element		Type	Description
	height	integer	図形の高さ。
	text	string	図形内のテキスト。
	font	integer	テキストのフォントスタイル。 可能な値: 0 - Georgia, serif; 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif; 2 - ”Times New Roman”, Times, serif; 3 - Arial, Helvetica, sans-serif; 4 - ”Arial Black”, Gadget, sans-serif; 5 - ”Comic Sans MS”, cursive, sans-serif; 6 - Impact, Charcoal, sans-serif; 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif; 8 - Tahoma, Geneva, sans-serif; 9 - ”Trebuchet MS”, Helvetica, sans-serif; 10 - Verdana, Geneva, sans-serif; 11 - ”Courier New”, Courier, monospace; 12 - ”Lucida Console”, Monaco, monospace.
	font_size	integer	フォントサイズ（ピクセル）。
	font_color	string	16 進コードで表されるフォント色。
	text_halign	integer	テキストの水平方向の配置。 可能な値: 0 - 中央; 1 - 左; 2 - 右。
	text_valign	integer	テキストの垂直方向の配置。 可能な値: 0 - 中央; 1 - 上; 2 - 下。
	border_type	integer	図形の枠線タイプ。 可能な値: 0 - なし; 1 - 太線; 2 - 点線; 3 - 破線。
	border_width	integer	枠線の幅（ピクセル）。
	border_color	string	16 進コードで表される枠線色。
	background_color	string	16 進コードで表される背景（塗りつぶし）色。
	zindex	integer	すべての図形と線の順序付けに使用する値（z-index）。
lines			マップ線のルート要素。
	x1	integer	線の点 1 の X 座標（ピクセル）。
	y1	integer	線の点 1 の Y 座標（ピクセル）。
	x2	integer	線の点 2 の X 座標（ピクセル）。
	y2	integer	線の点 2 の Y 座標（ピクセル）。
	line_type	integer	線のタイプ。 可能な値: 0 - なし; 1 - 太線; 2 - 点線; 3 - 破線。
	line_width	integer	線の幅（ピクセル）。
	line_color	string	16 進コードで表される線の色。
	zindex	integer	すべての図形と線の順序付けに使用する値（z-index）。
links			マップ要素間のリンクのルート要素。
	drawtype	integer	リンクのスタイル。 可能な値: 0 - 線; 2 - 太線; 3 - 点線; 4 - 破線。
	color	string	リンクの色（6 文字の 16 進数）。

Element	Type	Description
label	string	リンクラベル。
selementid1	id	接続する一方の要素の ID。
selementid2	id	接続するもう一方の要素の ID。
linktriggers		リンク状態インジケーター のルート要素。

Note:
See also: Map object (refer to the relevant property with a matching name).

マップ要素

Element	Type	Description
elementtype	integer	マップ要素のタイプ。 可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - イメージ。
elements		マップ上に表示される Zabbix エンティティ（ホスト、ホストグループ、マップなど）のルート要素。 すべてのエンティティには、それぞれ対応する要素（host など。たとえばExport format を参照）があります。
label	string	アイコンのラベル。
label_location	integer	ラベルの位置。 可能な値: -1 - マップのデフォルトを使用; 0 - 下; 1 - 左; 2 - 右; 3 - 上。
x	integer	X 軸上の位置。
y	integer	Y 軸上の位置。
elementsubtype	integer	要素のサブタイプ。 elementtype が 3 に設定されている場合にサポートされます。 可能な値: 0 - 単一のホストグループ; 1 - すべてのホストグループ。
areatype	integer	領域サイズ。 elementtype が 1 に設定されている場合にサポートされます。 可能な値: 0 - マップ全体と同じ; 1 - カスタムサイズ。
width	integer	領域の幅。 areatype が 1 に設定されている場合にサポートされます。
height	integer	領域の高さ。 areatype が 1 に設定されている場合にサポートされます。
viewtype	integer	領域の配置アルゴリズム。 elementsubtype が 1 に設定されている場合にサポートされます。 可能な値: 0 - 領域内に均等に配置。
use_iconmap	integer	この要素にアイコンマッピングを使用します。マップレベルでアイコンマッピングが有効な場合にのみ関連します。 可能な値: 0 - いいえ; 1 - はい。
selementid	id	一意の要素レコード ID。
icon_off		要素が OK 状態のときに使用するイメージのルート要素。
name	string	一意のイメージ名。
icon_on		要素が Problem 状態のときに使用するイメージのルート要素。

Element	Type	Description
name	string	一意のイメージ名。
icon_disabled		要素が無効のときに使用するイメージのルート要素。
name	string	一意のイメージ名。
icon_maintenance		要素がメンテナンス中のときに使用するイメージのルート要素。
name	string	一意のイメージ名。
urls		マップまたは各マップ要素で使用される URL のルート要素。
name	string	リンク名。
url	string	リンク URL。
evaltype	integer	タグの評価タイプ。
tags		障害タグのルート要素（ホストおよびホストグループ要素用）。 タグが指定されている場合、これらのタグを持つ障害のみがマップに表示されます。
tag	string	タグ名。
value	string	タグ値。
operator	integer	演算子。

Note:
こちらも参照: Map element object（対応する名前の関連プロパティを参照してください）。

マップリンクの状態インジケーター

Element	Type	Description
drawtype	integer	トリガーが「障害」状態のときのリンクスタイル。 指定可能な値: 0 - 線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	トリガーが「障害」状態のときのリンク色（6 文字の 16 進数）。
trigger		リンクの状態を示すために使用されるトリガーのルート要素。
description	string	トリガー名。
expression	string	トリガー式。
recovery_expression	string	トリガーの復旧式。

Note:
関連情報: Map link trigger object（対応する名前の関連プロパティを参照してください）。

メディアタイプ

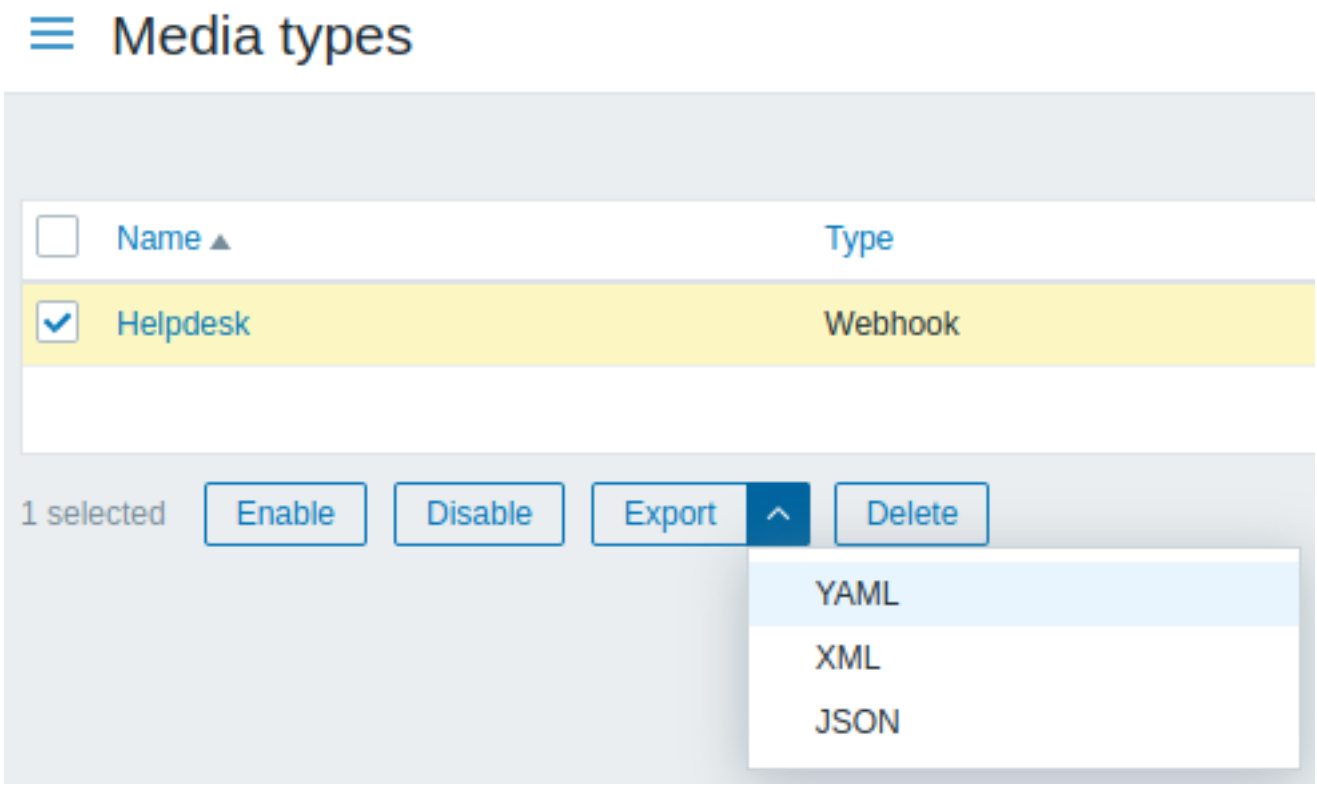
概要

メディアタイプは、関連するすべてのオブジェクトおよびオブジェクト間の関係とともにエクスポートされます。

エクスポート

メディアタイプをエクスポートするには、次の手順を実行します。

1. アラート > メディアタイプに移動します。
2. エクスポートするメディアタイプのチェックボックスをオンにします。
3. 一覧の下にある エクスポートをクリックして既定の YAML 形式でエクスポートするか、矢印をクリックして別のエクスポート形式を選択します。



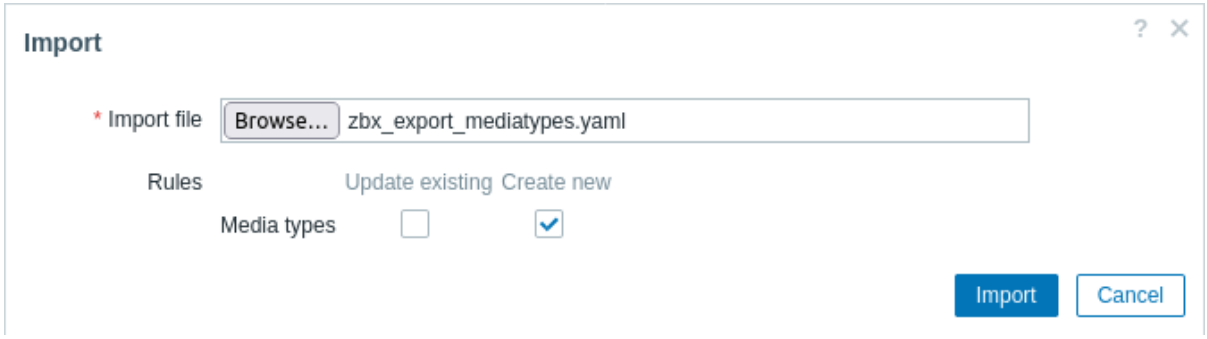
選択した形式に応じて、メディアタイプは既定の名前を持つローカルファイルにエクスポートされます。

- `zabbix_export_mediatypes.yaml` - YAML でのエクスポート;
- `zabbix_export_mediatypes.xml` - XML でのエクスポート;
- `zabbix_export_mediatypes.json` - JSON でのエクスポート。

インポート

メディアタイプをインポートするには、次の手順を実行します。

1. *Alerts* > *Media types* に移動します。
2. 右上隅にある *Import* をクリックします。
3. インポートファイルを選択します。
4. 適用するインポートルールのチェックボックスにチェックを入れます（以下を参照）。
5. 設定フォームの右下隅にある *Import* をクリックします。



インポートルール:

ルール	説明
Update existing	インポートファイルのデータを使用して既存のメディアタイプを更新するには、このチェックボックスにチェックを入れます。
Create new	インポートファイルのデータを使用して新しいメディアタイプを作成するには、このチェックボックスにチェックを入れます。インポートファイルにすでに存在するメディアタイプ（例: *Email*）が含まれている場合、新しいメディアタイプは作成されず、既存のメディアタイプも更新されません。

インポート中にメディアタイプへの変更内容を確認することはできません。インポートの成功または失敗を示すメッセージが Web インターフェースに表示されます。

エクスポート形式

YAML にエクスポート:

```
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
```

```
            {name: 'average', color: '#FFA059'},
            {name: 'high', color: '#E97659'},
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 1
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };

    if (priority == 2) {
        if (isNaN(params.retry) || params.retry < 30) {
            throw 'field "retry" should be a number with value of at least 30 if "priority" is set
```

```
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    設定手順については、こちらを参照してください: https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/temp

    token パラメータに Pushover アプリケーションキーを設定してください。
    Pushover メディアを Zabbix ユーザーに割り当てる際は、send to フィールドに user キーを追加してください。
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: '障害: {EVENT.NAME}'
      message: |
        障害発生時刻: {EVENT.TIME} {EVENT.DATE}
        障害名: {EVENT.NAME}
        ホスト: {HOST.NAME}
        深刻度: {EVENT.SEVERITY}
        運用データ: {EVENT.OPDATA}
        元の障害 ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
```

```
      operation_mode: RECOVERY
      subject: '{EVENT.DURATION} で解決: {EVENT.NAME}'
      message: |
        障害は {EVENT.RECOVERY.TIME} {EVENT.RECOVERY.DATE} に解決されました
        障害名: {EVENT.NAME}
        障害継続時間: {EVENT.DURATION}
        ホスト: {HOST.NAME}
        深刻度: {EVENT.SEVERITY}
        元の障害 ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: '更新された障害 {EVENT.AGE}: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} が {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME} に障害を {EVENT.UPDATE.ACTION} しました
        {EVENT.UPDATE.MESSAGE}

        現在の障害状態は {EVENT.STATUS}、経過時間は {EVENT.AGE}、確認済み: {EVENT.ACK.STATUS}。
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'ディスカバリ: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        ディスカバリルール: {DISCOVERY.RULE.NAME}

        デバイス IP: {DISCOVERY.DEVICE.IPADDRESS}
        デバイス DNS: {DISCOVERY.DEVICE.DNS}
        デバイス状態: {DISCOVERY.DEVICE.STATUS}
        デバイス稼働時間: {DISCOVERY.DEVICE.UPTIME}

        デバイスサービス名: {DISCOVERY.SERVICE.NAME}
        デバイスサービスポート: {DISCOVERY.SERVICE.PORT}
        デバイスサービス状態: {DISCOVERY.SERVICE.STATUS}
        デバイスサービス稼働時間: {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: '自動登録: {HOST.HOST}'
      message: |
        ホスト名: {HOST.HOST}
        ホスト IP: {HOST.IP}
        エージェントポート: {HOST.PORT}
```

エクスポートされた要素

エクスポートされた要素については、以下の表で説明します。

Element	Type	Description
name	string	(必須) メディアタイプ名。
type	string	(必須) メディアタイプで使用されるトランスポート。 可能な値:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	メディアタイプが有効かどうか。 可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)
max_sessions	integer	並列で処理できるアラートの最大数。 SMS の場合の可能な値:[1] 1 (デフォルト)。 その他のメディアタイプの場合の可能な値:[1] 0-100 (0 は無制限)。
attempts	integer	アラート送信の最大試行回数。 可能な値:[1] 1-10 (デフォルト: 3)。
attempt_interval	string	再試行の間隔（秒、またはtime suffix を使用）。 可能な値:[1] 0-60s (デフォルト: 10s)。
description	string	メディアタイプの説明。
message_templates		メディアタイプのメッセージテンプレートのルート要素。
event_source	string	(必須) イベントソース。 可能な値:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).

Element	Type	Description
operation_mode	string	動作モード。 可能な値:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	メッセージ件名。
message	string	メッセージ本文。

Note:
関連項目: Media type object（該当する名前のプロパティを参照してください）。

Email

以下の追加要素は、*Email* メディアタイプに対してのみエクスポートされます。

Element	Type	Description
provider	string	Email プロバイダー。
smtp_server	string	SMTP サーバー。
smtp_port	integer	接続する SMTP サーバーのポート。 Default: 25.
smtp_helo	string	SMTP helo。
smtp_email	string	通知の送信元となるメールアドレス。
smtp_security	string	使用する SMTP 接続のセキュリティレベル。 Possible values:[1] NONE (0, default), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	SMTP の SSL ホスト検証。 Possible values:[1] NO (0, default), YES (1).
smtp_verify_peer	string	SMTP の SSL ピア検証。 Possible values:[1] NO (0, default), YES (1).
smtp_authentication	string	使用する SMTP 認証方式。 Possible values:[1] NONE (0, default), PASSWORD (1), OAUTH (2).
username	string	ユーザー名。
password	string	認証パスワード。
redirection_url	string	OAuth 認可を戻すための Zabbix Web インターフェース URL。
client_id	string	OAuth 認可サーバーに登録されたクライアント識別子。
authorization_url	string	アクセストークンとリフレッシュトークンを取得するための、パラメーター付き OAuth URL。
token_url	string	認可トークンをアクセストークンとリフレッシュトークンに交換するための OAuth URL。
message_format	string	メッセージ形式。 Possible values:[1] TEXT (0), HTML (1, default).

Note:
関連情報: Media type object（該当する名前のプロパティを参照してください）。

SMS

以下の追加要素は、*SMS* メディアタイプに対してのみエクスポートされます。

Element	Type	Description
gsm_modem	string	(required) GSM モデムのシリアルデバイス名。

Note:
関連項目: Media type object（対応する名前の関連プロパティを参照してください）。

スクリプト

以下の追加要素は、*Script* メディアタイプに対してのみエクスポートされます。

Element	Type	Description
script name	string	(required) スクリプト名。

Element		Type	Description
parameters			スクリプトパラメータのルート要素。
	sortorder	string	(required) スクリプトにコマンドライン引数として渡されるスクリプトパラメータの順序。
	value	string	スクリプトパラメータの値。

Note:
関連情報: Media type object（対応する名前の関連プロパティを参照してください）。

webhook

以下の追加要素は、*webhook* メディアタイプに対してのみエクスポートされます。

Element	Type	Description
script	string	スクリプト。
timeout	string	Javascript スクリプトの HTTP リクエストのタイムアウト間隔。 可能な値:[1] 1-60s (デフォルト: 30s)。
process_tags	string	返されたタグを処理するかどうか。 可能な値:[1] NO (0, デフォルト), YES (1)。
show_event_menu	string	`event_menu_url` および `event_menu_name` フィールドで {EVENT.TAGS.*} マクロが正常に解決された場合に、イベントメニューに項目が存在することを示します。 可能な値:[1] NO (0, デフォルト), YES (1)。
event_menu_url	string	イベントメニュー項目の URL。{EVENT.TAGS.*} マクロをサポートします。
event_menu_name	string	イベントメニュー項目の名前。{EVENT.TAGS.*} マクロをサポートします。
parameters		webhook メディアタイプのパラメーターのルート要素。
name	string	(必須) webhook パラメーター名。
value	string	webhook パラメーター値。

Note:
参照: Media type object (対応する名前のプロパティを参照してください)。

脚注

[1] 角括弧内の API の整数値、たとえば ENABLED (0) は、参照用としてのみ記載されています。詳細については、表の項目内または各セクションの末尾にあるリンク先の API オブジェクトページを参照してください。

ダッシュボード

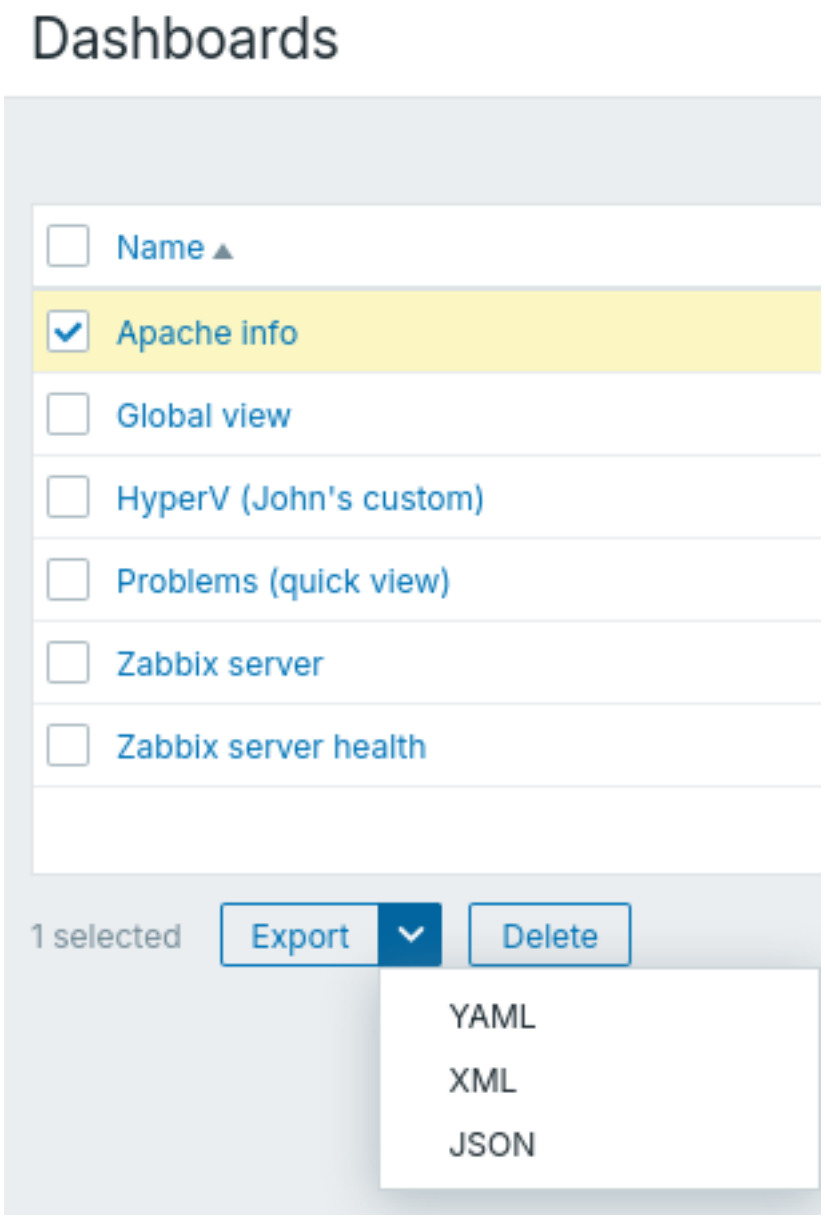
概要

ダッシュボードは、関連するすべてのオブジェクトおよびオブジェクト間の関係を含めてエクスポートされます。

エクスポート

ダッシュボードをエクスポートするには、次の手順を実行します。

1. ダッシュボード > すべてのダッシュボードに移動します。
2. エクスポートするダッシュボードのチェックボックスを選択します。
3. 一覧の下にある エクスポートをクリックして、デフォルトの YAML 形式でエクスポートするか、矢印をクリックして別のエクスポート形式を選択します。



選択した形式に応じて、ダッシュボードは次の既定名のローカルファイルとしてエクスポートされます。

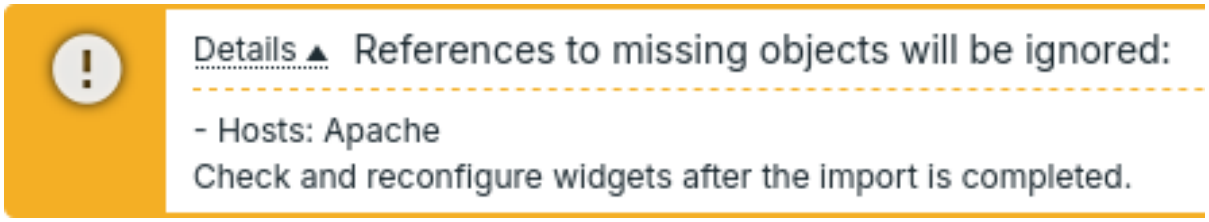
- `zbx_export_dashboards.yaml` - YAML でのエクスポート;
- `zbx_export_dashboards.xml` - XML でのエクスポート;
- `zbx_export_dashboards.json` - JSON でのエクスポート。

インポート

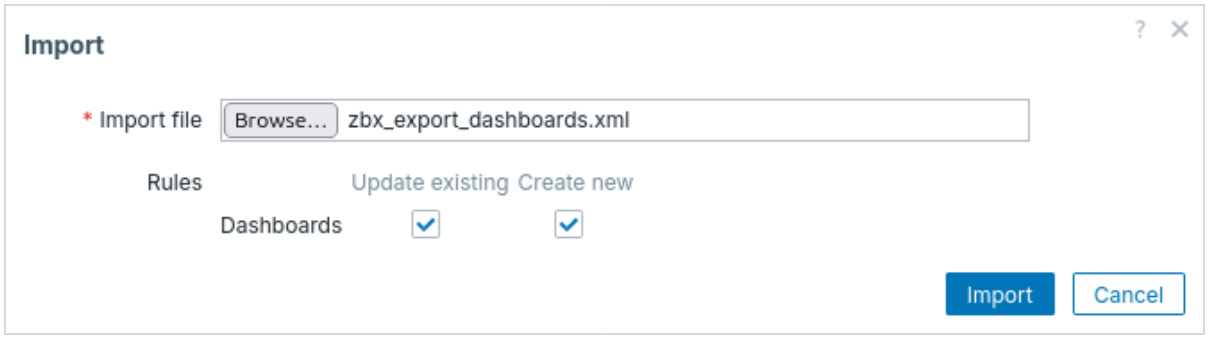
ダッシュボードをインポートするには、次の手順を実行します。

1. ダッシュボード > すべてのダッシュボードに移動します。
2. 右上隅の インポートをクリックします。ダッシュボードの作成と編集の権限がないユーザーには無効になります。
3. インポートファイルを選択します。
4. 適用するインポートルールのチェックボックスにチェックを入れます（以下を参照）。
5. 設定フォームの右下隅にある インポートをクリックします。

インポートファイルが、対象システムに存在しないオブジェクトを参照している場合、アラートメッセージが表示されます。これは、特定のホスト、アイテム、その他のオブジェクトを参照するウィジェットでは想定される動作です。このような場合、Zabbix ではダッシュボードをインポートできますが、不足している参照はクリアされます。また、データが必須の場合はウィジェットを完全には設定されていない状態としてマークし、フィールドがフィルタリングのみに使用される場合はフィルタリング条件を削除します。



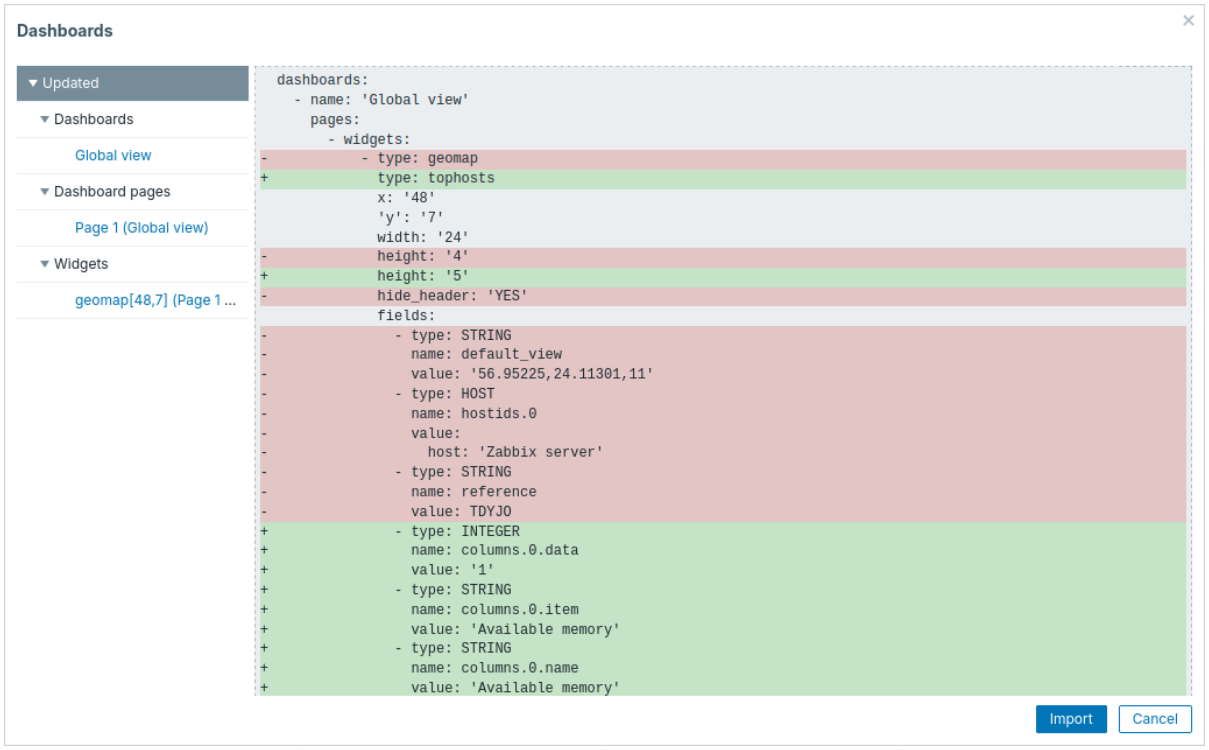
インポートボタンを再度クリックすると、ダッシュボードが追加および更新されます。



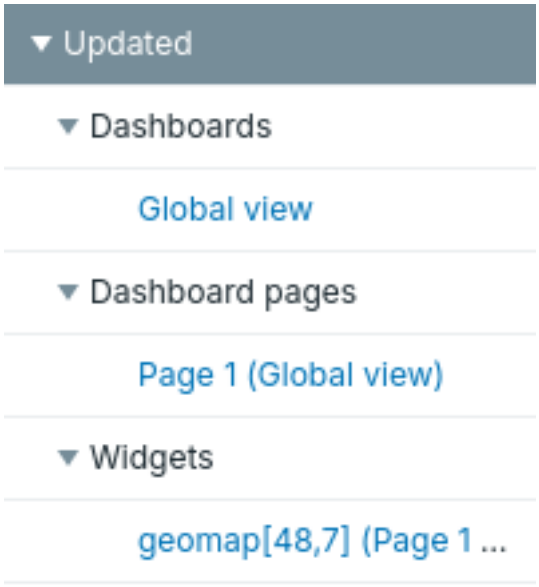
インポートルール:

ルール	説明
既存の更新	既存の要素をインポートファイルのデータで更新します。それ以外の場合、既存の要素は更新されません。
新規作成	インポートファイルのデータを使用して新しい要素を作成します。それ以外の場合、新しい要素は作成されません。

次の画面では、インポートされるダッシュボードの内容を確認できます。新しいダッシュボードの場合、すべての要素が緑色で表示されます。既存のダッシュボードを更新する場合、新しいダッシュボード要素は緑色で強調表示され、削除されたダッシュボード要素は赤色で強調表示されます。変更されていない要素は灰色の背景で表示されます。



左側のメニューを使用して、変更の一覧を移動できます。更新済みセクションには、既存のダッシュボード要素に加えられたすべての変更が表示されます。追加済みセクションには、新しいダッシュボード要素が表示されます。各セクションの要素は要素タイプごとにグループ化されています。灰色の矢印をクリックすると、要素グループを展開または折りたたむことができます。



ダッシュボードの変更内容を確認し、インポートをクリックしてダッシュボードをインポートします。インポートの成功または失敗を示すメッセージが Web インターフェースに表示されます。

エクスポート形式

YAML にエクスポート:

```yaml
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
              name: 'CPU 使用率の上位ホスト'
              width: '25'
              height: '4'
              fields:
                - type: INTEGER
                  name: column
                  value: '1'
                - type: INTEGER
                  name: columns.0.data
                  value: '2'
                - type: STRING
                  name: columns.0.item
                  value: 'System name'
                - type: STRING
                  name: columns.0.name
                  value: 'ホスト名'
                - type: STRING
                  name: columns.1.base_color
                  value: 4CAF50
                - type: INTEGER
                  name: columns.1.data
                  value: '1'
                - type: INTEGER
                  name: columns.1.display
                  value: '3'
                - type: STRING
                  name: columns.1.item
                  value: 'CPU utilization'
```

```
      - type: STRING
        name: columns.1.max
        value: '100'
      - type: STRING
        name: columns.1.min
        value: '0'
      - type: STRING
        name: columns.1.name
        value: Utilization
      - type: INTEGER
        name: columns.2.data
        value: '1'
      - type: STRING
        name: columns.2.item
        value: 'Load average (1m avg)'
      - type: STRING
        name: columns.2.name
        value: '1m avg'
      - type: INTEGER
        name: columns.3.data
        value: '1'
      - type: STRING
        name: columns.3.item
        value: 'Load average (5m avg)'
      - type: STRING
        name: columns.3.name
        value: '5m avg'
      - type: INTEGER
        name: columns.4.data
        value: '1'
      - type: STRING
        name: columns.4.item
        value: 'Load average (15m avg)'
      - type: STRING
        name: columns.4.name
        value: '15m avg'
      - type: INTEGER
        name: columns.5.data
        value: '1'
      - type: INTEGER
        name: columns.5.decimal_places
        value: '0'
      - type: STRING
        name: columns.5.item
        value: 'Number of processes'
      - type: STRING
        name: columns.5.name
        value: Processes
      - type: STRING
        name: columnsthresholds.1.color.0
        value: FFFF00
      - type: STRING
        name: columnsthresholds.1.color.1
        value: FF8000
      - type: STRING
        name: columnsthresholds.1.color.2
        value: FF4000
      - type: STRING
        name: columnsthresholds.1.threshold.0
        value: '50'
      - type: STRING
        name: columnsthresholds.1.threshold.1
```

```
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
        - type: hostavail
          'y': '4'
          width: '25'
          fields:
            - type: INTEGER
              name: interface_type.0
              value: '1'
        - type: problems
          name: '現在の障害'
          'y': '6'
          width: '48'
          height: '5'
          fields:
            - type: STRING
              name: reference
              value: BRUQE
            - type: INTEGER
              name: show
              value: '3'
            - type: INTEGER
              name: show_tags
              value: '3'
        - type: systeminfo
          x: '25'
          width: '23'
          height: '4'
        - type: problemsbysv
          name: '深刻度別の障害'
          x: '25'
          'y': '4'
          width: '23'
          fields:
            - type: STRING
              name: reference
              value: LBSVB
            - type: INTEGER
              name: show_type
              value: '1'
        - type: gauge
          name: 'Zabbix server'
          x: '48'
          width: '12'
          height: '4'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: angle
              value: '270'
            - type: INTEGER
              name: desc_size
              value: '8'
            - type: ITEM
              name: itemid.0
              value:
                host: 'Zabbix server'
```

```
      key: vm.memory.utilization
  - type: STRING
    name: max
    value: '100'
  - type: STRING
    name: min
    value: '0'
  - type: INTEGER
    name: scale_size
    value: '11'
  - type: INTEGER
    name: show.0
    value: '1'
  - type: INTEGER
    name: show.1
    value: '2'
  - type: INTEGER
    name: show.2
    value: '3'
  - type: INTEGER
    name: show.3
    value: '4'
  - type: INTEGER
    name: show.4
    value: '5'
  - type: STRING
    name: thresholds.0.color
    value: 42A5F5
  - type: STRING
    name: thresholds.0.threshold
    value: '0'
  - type: STRING
    name: thresholds.1.color
    value: 9CCC65
  - type: STRING
    name: thresholds.1.threshold
    value: '20'
  - type: STRING
    name: thresholds.2.color
    value: FFEE58
  - type: STRING
    name: thresholds.2.threshold
    value: '40'
  - type: STRING
    name: thresholds.3.color
    value: FFA726
  - type: STRING
    name: thresholds.3.threshold
    value: '60'
  - type: STRING
    name: thresholds.4.color
    value: FF7043
  - type: STRING
    name: thresholds.4.threshold
    value: '80'
  - type: INTEGER
    name: th_arc_size
    value: '9'
  - type: INTEGER
    name: th_show_arc
    value: '1'
  - type: INTEGER
```

```
            name: th_show_labels
            value: '1'
          - type: INTEGER
            name: units_bold
            value: '1'
          - type: INTEGER
            name: units_size
            value: '24'
          - type: INTEGER
            name: value_arc_size
            value: '30'
          - type: INTEGER
            name: value_bold
            value: '1'
          - type: INTEGER
            name: value_size
            value: '24'
      - type: svggraph
        x: '48'
        'y': '4'
        width: '24'
        height: '3'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: axisx
            value: '0'
          - type: STRING
            name: ds.0.color
            value: AB47BC
          - type: INTEGER
            name: ds.0.fill
            value: '0'
          - type: STRING
            name: ds.0.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.0.items.0
            value: 'Available memory'
          - type: INTEGER
            name: ds.0.transparency
            value: '2'
          - type: INTEGER
            name: ds.0.type
            value: '2'
          - type: INTEGER
            name: ds.0.width
            value: '2'
          - type: INTEGER
            name: ds.1.axisy
            value: '1'
          - type: STRING
            name: ds.1.color
            value: 42A5F5
          - type: INTEGER
            name: ds.1.fill
            value: '0'
          - type: STRING
            name: ds.1.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.1.items.0
```

```
            value: 'CPU utilization'
          - type: INTEGER
            name: ds.1.transparency
            value: '0'
          - type: INTEGER
            name: ds.1.type
            value: '2'
          - type: INTEGER
            name: ds.1.width
            value: '2'
          - type: STRING
            name: reference
            value: TOYGS
      - type: geomap
        x: '48'
        'y': '7'
        width: '24'
        height: '4'
        hide_header: 'YES'
        fields:
          - type: STRING
            name: default_view
            value: '56.95225,24.11301,11'
          - type: HOST
            name: hostids.0
            value:
              host: 'Zabbix server'
          - type: STRING
            name: reference
            value: TDYJO
      - type: clock
        name: 'ローカル時刻'
        x: '60'
        width: '12'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: clock_type
            value: '1'
          - type: INTEGER
            name: show.0
            value: '1'
          - type: INTEGER
            name: show.1
            value: '2'
          - type: INTEGER
            name: show.2
            value: '3'
          - type: INTEGER
            name: time_sec
            value: '0'
          - type: INTEGER
            name: tzone_format
            value: '1'
      - type: item
        name: 'サーバー性能'
        x: '60'
        'y': '2'
        width: '12'
        hide_header: 'YES'
        fields:
          - type: INTEGER
```

```
      name: aggregate_function
      value: '2'
    - type: INTEGER
      name: decimal_size
      value: '32'
    - type: STRING
      name: description
      value: 'Values per second'
    - type: INTEGER
      name: desc_h_pos
      value: '0'
    - type: INTEGER
      name: desc_v_pos
      value: '0'
    - type: ITEM
      name: itemid.0
      value:
        host: 'Zabbix server'
        key: 'zabbix[wcache,values]'
    - type: INTEGER
      name: show.0
      value: '1'
    - type: INTEGER
      name: show.1
      value: '2'
    - type: INTEGER
      name: show.2
      value: '5'
    - type: INTEGER
      name: sparkline.fill
      value: '2'
    - type: INTEGER
      name: sparkline.width
      value: '2'
    - type: STRING
      name: thresholds.0.color
      value: DCEDC8
    - type: STRING
      name: thresholds.0.threshold
      value: '20'
    - type: STRING
      name: thresholds.1.color
      value: FFF9C4
    - type: STRING
      name: thresholds.1.threshold
      value: '40'
    - type: STRING
      name: thresholds.2.color
      value: FFCCBC
    - type: STRING
      name: thresholds.2.threshold
      value: '60'
    - type: INTEGER
      name: units_bold
      value: '0'
    - type: INTEGER
      name: units_show
      value: '0'
    - type: INTEGER
      name: value_h_pos
      value: '0'
    - type: INTEGER
```

```
                name: value_size
                value: '32'
```

エクスポートされた要素

エクスポートされた要素については、以下の表で説明します。

Element	Type	Description
name	string	(必須) ダッシュボード名。
pages		ダッシュボードページのルート要素
widgets		ダッシュボードページのルート要素

Note:
関連情報: Dashboard object（対応する名前の関連プロパティを参照してください）。

ダッシュボードウィジェット

Element	Type	Description
type	string	(必須) ダッシュボードウィジェットの種類。 指定可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリーステータス; favgraphs - お気に入りグラフ; favmaps - お気に入りマップ; gauge - ゲージ; geomap - ジオマップ; graph - グラフ (クラシック); graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホストの可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemhistory - アイテム履歴; itemnavigator - アイテムナビゲーター; item - アイテム値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; problemhosts - 障害ホスト; problems - 障害; problemsbysv - 深刻度別障害; scatterplot - 散布図; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - 上位ホスト; topitems - 上位アイテム; toptriggers - 上位トリガー; trigover - トリガー概要; url - URL; web - Web 監視。
x	integer	ダッシュボードの左側からの水平方向の位置。 指定可能な値:[1] 0-71.
y	integer	ダッシュボードの上部からの垂直方向の位置。 指定可能な値:[1] 0-63.
name	string	カスタムウィジェット名。
width	integer	ウィジェットの幅。 指定可能な値:[1] 1-72.

Element	Type	Description
height	integer	ウィジェットの高さ。 指定可能な値:[1] 1-64.
hide_header	string	ウィジェットのヘッダーを非表示にします。 指定可能な値:[1] NO (0, デフォルト), YES (1).
fields		ダッシュボードウィジェットのフィールドのルート要素。

Note:
こちらも参照: Dashboard widget (対応する名前の関連プロパティを参照してください)。

ダッシュボードウィジェットフィールド

Element	Type	Description
type	string	(必須) ウィジェットフィールドのタイプ。 可能な値:[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(必須) ウィジェットフィールド名。
value	mixed	(必須) フィールドタイプに応じたウィジェットフィールドの値。
host	string	アイテムのホスト。
key	string	アイテムキー。

Note:
こちらも参照してください: Dashboard widget field (該当する名前のプロパティを参照してください)。

脚注

[1] 角括弧内の API の整数値、たとえば ENABLED (0) は、参照用としてのみ記載されています。詳細については、表の項目にあるリンク先の API オブジェクトページ、または各セクションの末尾を参照してください。

正規表現

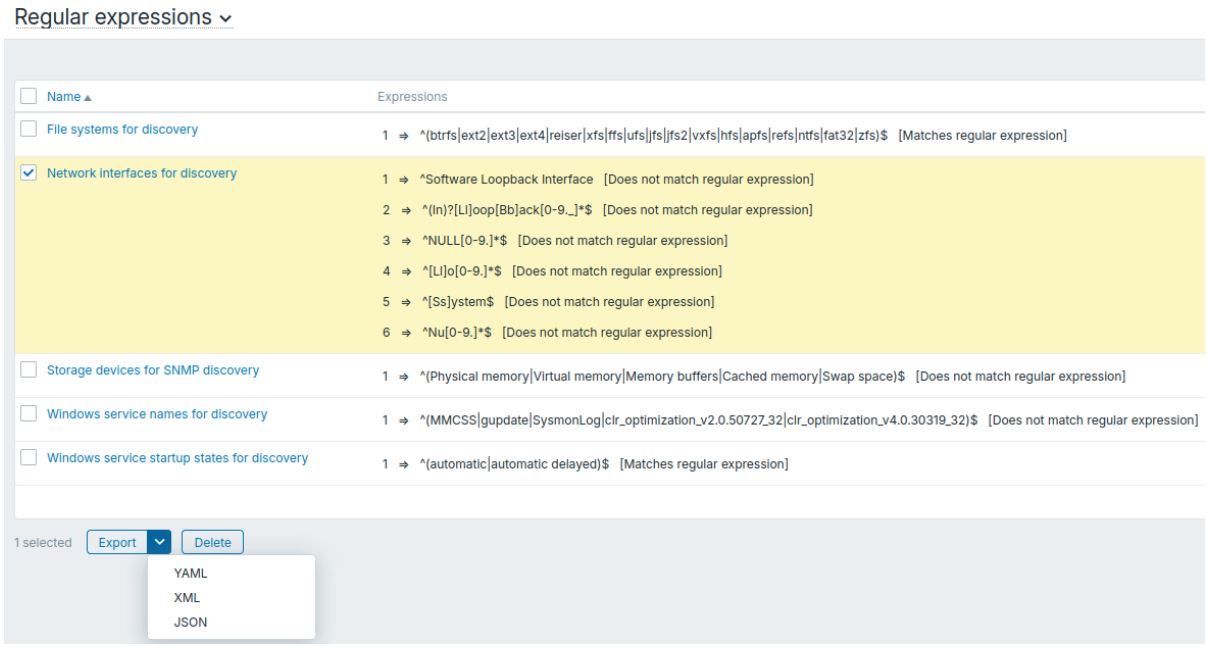
概要

正規表現は、関連するすべてのオブジェクトおよびオブジェクト間の関係とともにエクスポートされます。

エクスポート

正規表現をエクスポートするには、次の手順を実行します。

1. *Administration* > *General* > *Regular expressions* に移動します。
2. エクスポートする正規表現のチェックボックスを選択します。
3. 一覧の下にある *Export* をクリックすると、デフォルトの YAML 形式でエクスポートされます。別のエクスポート形式を選択するには、矢印をクリックします。



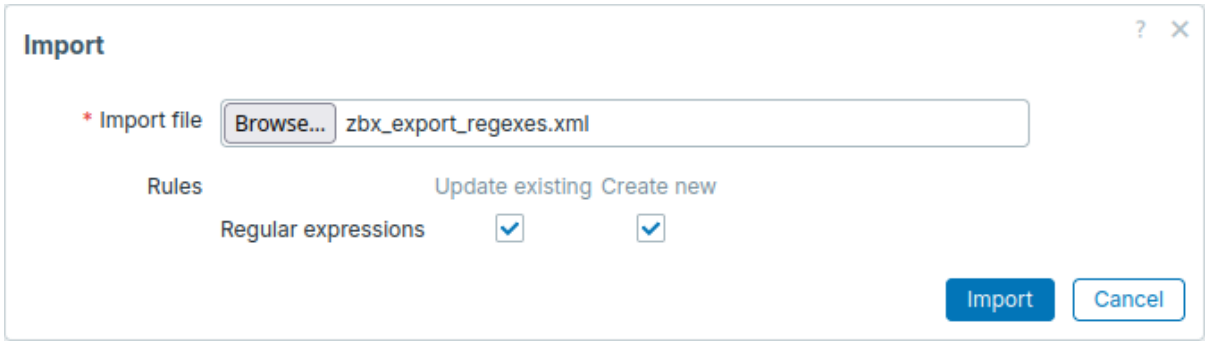
選択した形式に応じて、正規表現は既定の名前を持つローカルファイルにエクスポートされます。

- `zbx_export_regexes.yaml` - YAML でのエクスポート;
- `zbx_export_regexes.xml` - XML でのエクスポート;
- `zbx_export_regexes.json` - JSON でのエクスポート。

インポート

正規表現をインポートするには、次の手順を実行します。

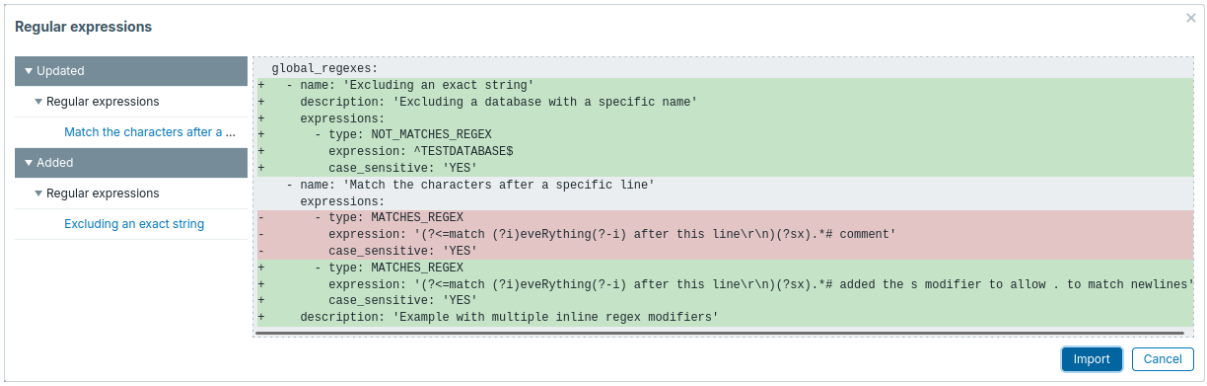
1. 管理 > 一般設定 > 正規表現に移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. 適用するインポートルールのチェックボックスにチェックを入れます（以下を参照）。
5. 設定フォームの右下隅にある インポートをクリックします。



インポートルール:

ルール	説明
既存の更新	インポートファイルのデータを使用して既存の正規表現を更新し、同じ名前の既存の正規表現を完全に上書きします。選択しない場合、既存の正規表現は更新されません。
新規作成	インポートファイルのデータを使用して新しい正規表現を作成します。選択しない場合、新しい正規表現は作成されません。

次の画面では、インポートされる正規表現の内容を確認できます。新しい正規表現の場合、すべてのデータが緑色で表示されます。既存の正規表現を更新する場合、新しい正規表現のデータは緑色で強調表示され、削除されたデータは赤色で強調表示されます。変更されていないデータは灰色の背景で表示されます。



左側のメニューを使用して、変更の一覧を移動できます。更新済みセクションには、既存の正規表現に加えられたすべての変更が表示されます。追加済みセクションには、新しい正規表現が一覧表示されます。矢印をクリックすると、正規表現のグループを展開または折りたたむことができます。

正規表現の変更内容を確認し、インポートをクリックして正規表現をインポートします。インポートの成功または失敗を示すメッセージが Web インターフェースに表示されます。

エクスポート形式

YAML にエクスポート:

```
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'ディスカバリ用のファイルシステム'
      description: '監視対象ホスト上のファイルシステムを検出して識別するために使用されるグローバル正規表現'
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'SNMP ディスカバリ用のストレージデバイス'
      expressions:
        - type: NOT_MATCHES_REGEX
          expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
          case_sensitive: 'YES'
```

エクスポートされた要素

正規表現のエクスポートされた要素:

Element	Type	Description
global_regexes	object	正規表現のルート要素。
name	string	(必須) 正規表現の名前。
description	text	正規表現の説明。
expressions	object	正規表現のルート要素。
type	string	正規表現のタイプ。
expression	string	正規表現。
exp_delimiter	string	正規表現の区切り文字。タイプが *Contains any substring from list* の場合に存在します。
case_sensitive	string	正規表現の大文字小文字の区別オプション。

Note:
関連情報: Regular expression object (対応する名前の関連プロパティを参照してください)。

正規表現

概要　Perl Compatible Regular Expressions (PCRE, PCRE2) は Zabbix でサポートされています。

Zabbix で正規表現を使用する方法は 2 つあります。

- 正規表現を手動で入力する

- Zabbix で作成したグローバル正規表現を使用する

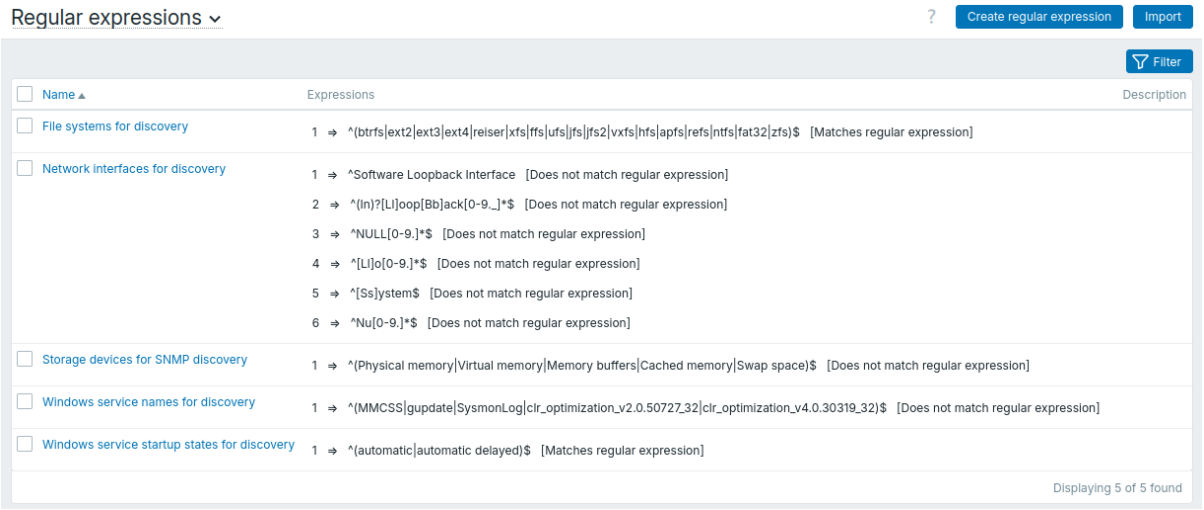
正規表現　supported places では、正規表現を手動で入力できます。
 式は @ で始めてはいけません。この記号は Zabbix でグローバル正規表現を参照するために使用されるためです。

> **Attention:**
> 正規表現を使用すると、スタックメモリが不足する可能性があります。詳細については、PCRE PERFORMANCE を参照してください。

複数行マッチングでは、^ と $ のアンカーは、それぞれ文字列全体の先頭と末尾ではなく、各行の先頭と末尾に一致することに注意してください。

さまざまなコンテキストでの正しいエスケープの例も参照してください。

グローバル正規表現　Zabbix では、複雑な正規表現を作成およびテストするための高度なエディターが Web インターフェースに用意されています。
 アクセスするには、*Administration* > *General* > *Regular expressions* に移動します。
 Zabbix には、デフォルトデータセットとして多数の事前定義されたグローバル正規表現が含まれています。

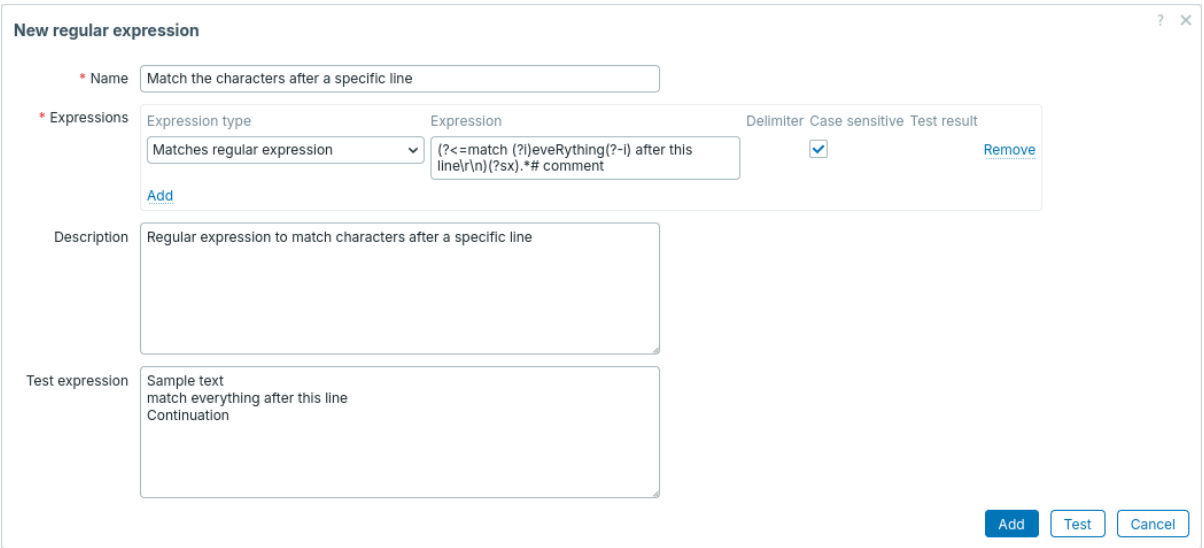


正規表現の一覧は、名前と説明でフィルタリングして、必要な式を見つけることができます。

設定

グローバル正規表現を作成するには、次の手順に従います。

1. 正規表現を作成をクリックします。
2. フォームに正規表現のパラメータを入力します。



すべての必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ		説明
名前		正規表現の名前を設定します。Unicode 文字を使用できます。
式		式セクションの追加をクリックして、新しいサブ式を追加します。
	式のタイプ	式のタイプを選択します。 文字列を含む - 部分文字列に一致します。 リスト内のいずれかの部分文字列を含む - 区切り文字で区切られたリスト内のいずれかの部分文字列に一致します。区切り文字には、コンマ ,、ドット .、スラッシュ / を使用できます。 文字列を含まない - 部分文字列以外のすべての文字列に一致します。 正規表現に一致する - 正規表現に一致します。 正規表現に一致しない - 正規表現に一致しません。
	式	部分文字列または正規表現を入力します。 最大長: 2048 文字。
	区切り文字	正規表現内のテキスト文字列を区切るためのコンマ ,、ドット .、またはスラッシュ /。このパラメータは、リスト内のいずれかの部分文字列を含む式のタイプが選択されている場合にのみ有効です。
	大文字と小文字を区別	正規表現で英字の大文字と小文字を区別するかどうかを指定するチェックボックスです。
説明		正規表現の説明を入力します。
式のテスト		正規表現のテストに使用する値です。

式内のスラッシュ / は、区切り文字ではなく文字どおりに扱われます。これにより、スラッシュを含む式をエラーなしで保存できます。

Attention:
Zabbix のカスタム正規表現名には、コンマ、スペースなどを含めることができます。参照時に誤って解釈される可能性がある場合（たとえば、アイテムキーのパラメータにコンマが含まれる場合）は、次のように参照全体を引用符で囲むことができます: *"@My custom regexp for purpose1, purpose2"*。
 その他の場所（たとえば、LLD ルールのプロパティ）では、正規表現名を引用符で囲まないでください。

正規表現を作成すると、名前の前に @ を付けて参照することで、Web インターフェースの複数の場所で使用できます。たとえば、`@mycustomregexp` のように指定します。

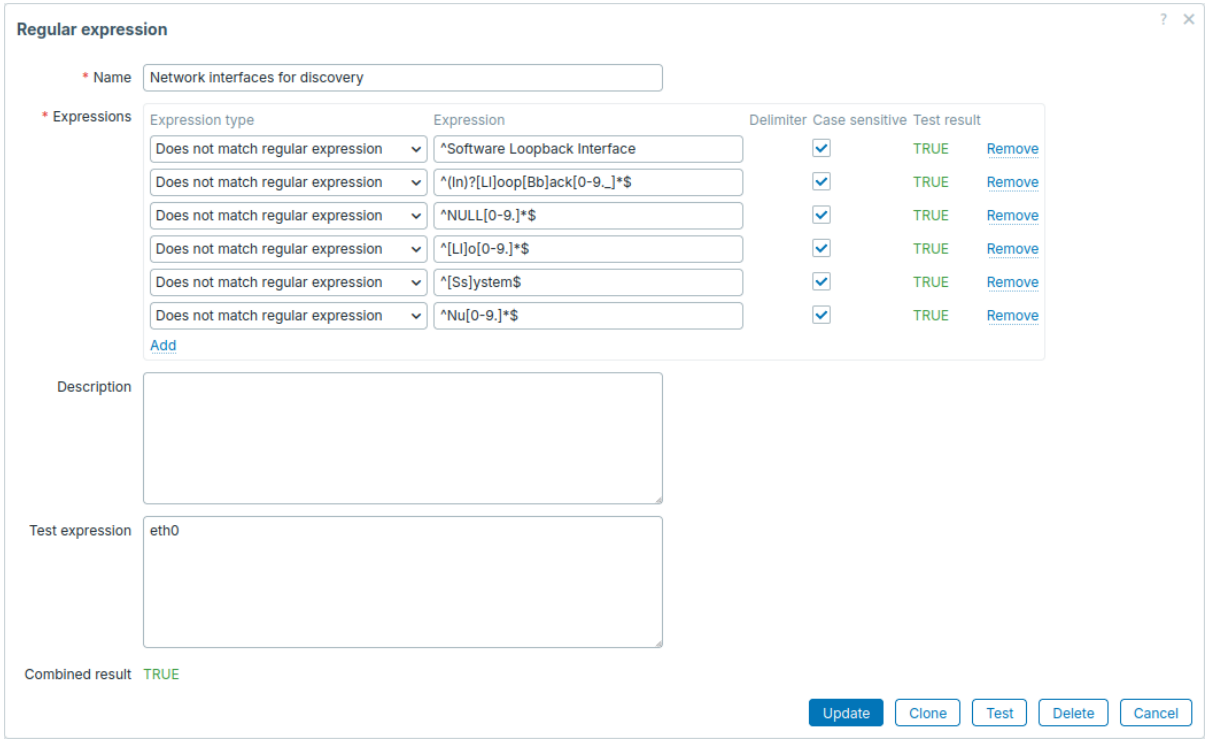
フォームのボタン

フォーム下部のボタンを使用して、さまざまな操作を実行できます。

ボタン	説明
Add	正規表現を追加します。このボタンは、新しい正規表現に対してのみ使用できます。
Update	正規表現のプロパティを更新します。
Clone	現在の正規表現のプロパティを基に、別の正規表現を作成します。
Test	指定した値に対して、正規表現とそのサブ表現をテストします。
Delete	正規表現を削除します。
Cancel	正規表現のプロパティの編集をキャンセルします。

正規表現のテスト

Test expression フィールドに値を入力し、*Test* ボタンをクリックすると、正規表現とその部分式をテストできます。



Test result 列には、各部分式のステータスが表示されます。
Combined result フィールドには、正規表現全体のステータスが表示されます。
複数の部分式が定義されている場合、Zabbix は論理 AND 演算子を使用して *Combined result* を計算します。つまり、少なくとも 1 つの結果が `False` の場合、*Combined result* も `False` になります。

デフォルトのグローバル正規表現

デフォルトのデータセットには、次の事前定義されたグローバル正規表現が含まれます。

Name	Expression	Matches
File systems for discovery	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	"btrfs" または"ext2" または"ext3" または"ext4" または"reiser" または"xfs" または"ffs" または"ufs" または"jfs" または"jfs2" または"vxfs" または"hfs" または"apfs" または"refs" または"ntfs" または"fat32" または"zfs"。
Network interfaces for discovery	`^Software Loopback Interface`	"Software Loopback Interface" で始まる文字列。
	`^lo$`	"lo"。
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	"In" で始まる場合があり、その後に"L" または"l"、続いて"oop"、続いて"B" または"b"、続いて"ack"、さらに任意個の数字、ドット、またはアンダースコアが続く文字列。
	`^NULL[0-9.]*$`	"NULL" で始まり、その後に任意個の数字またはドットが続く文字列。
	`^[Ll]o[0-9.]*$`	"Lo" または"lo" で始まり、その後に任意個の数字またはドットが続く文字列。
	`^[Ss]ystem$`	"System" または"system"。
	`^Nu[0-9.]*$`	"Nu" で始まり、その後に任意個の数字またはドットが続く文字列。
Storage devices for SNMP discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	"Physical memory" または"Virtual memory" または"Memory buffers" または"Cached memory" または"Swap space"。
Windows service names for discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	"MMCSS" または"gupdate" または"SysmonLog" または"clr_optimization_v2.0.50727_32" または"clr_optimization_v4.0.30319_32" に一致する文字列。各ドットは改行以外の任意の文字に置き換えることができます。

Name	Expression	Matches
Windows service startup states for discovery	`^(automatic|automatic delayed)$`	"automatic" または"automatic delayed"。

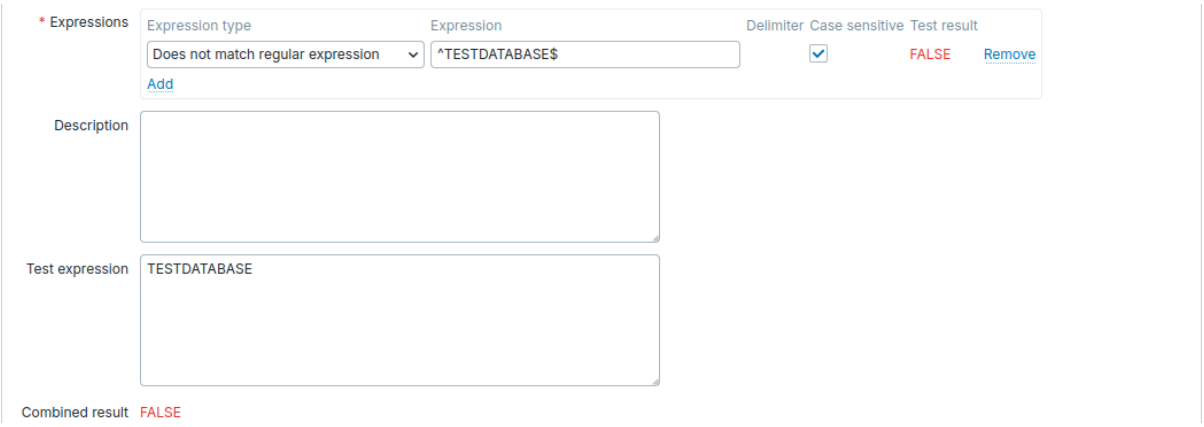
正規表現のインポート/エクスポート

正規表現 のエクスポート/インポート設定ページで説明されているように、グローバル正規表現をインポートまたはエクスポートできます。

例　完全一致する文字列を除外する例

特定の名前を持つデータベースを除き、低レベルディスカバリでデータベースを検出するには、次の式を使用します。

```
^TESTDATABASE$
```

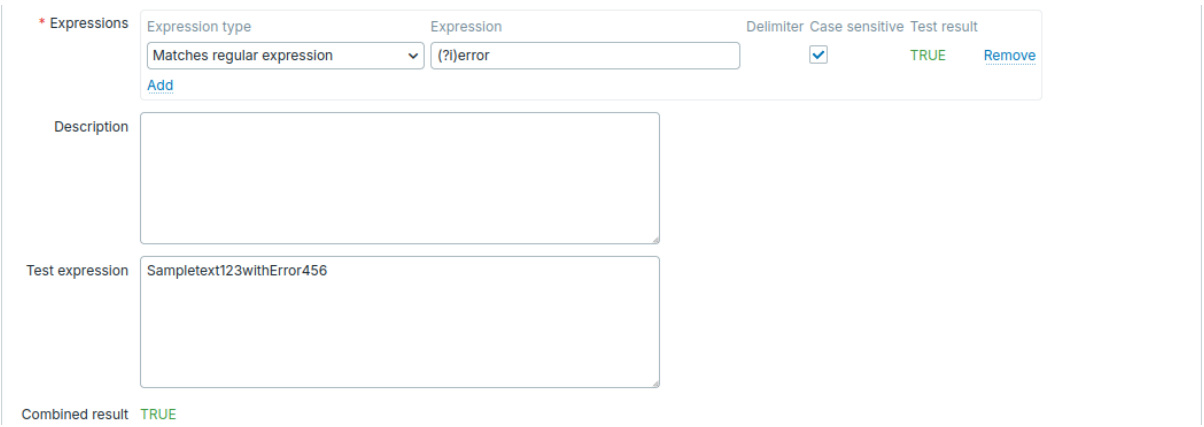


Test result と *Combined result* の両方に `FALSE` が表示されます。名前が"TESTDATABASE" に一致するため、*Does not match regular expression* 条件は満たされません。

インライン正規表現修飾子を使用した大文字小文字を区別しない一致の例

次の正規表現を使用します。これは、文字列"error" に対して大文字小文字を区別しない一致を行うためのインライン修飾子（`?i`）を含んでいます。

```
(?i)error
```



Test result と *Combined result* の両方に `TRUE` が表示されます。これは、*Case sensitive* が選択されていても、インライン修飾子（`?i`）により一致が大文字小文字を区別しなくなるため、文字列"error" が一致するからです。

複数のインライン正規表現修飾子を使用する例

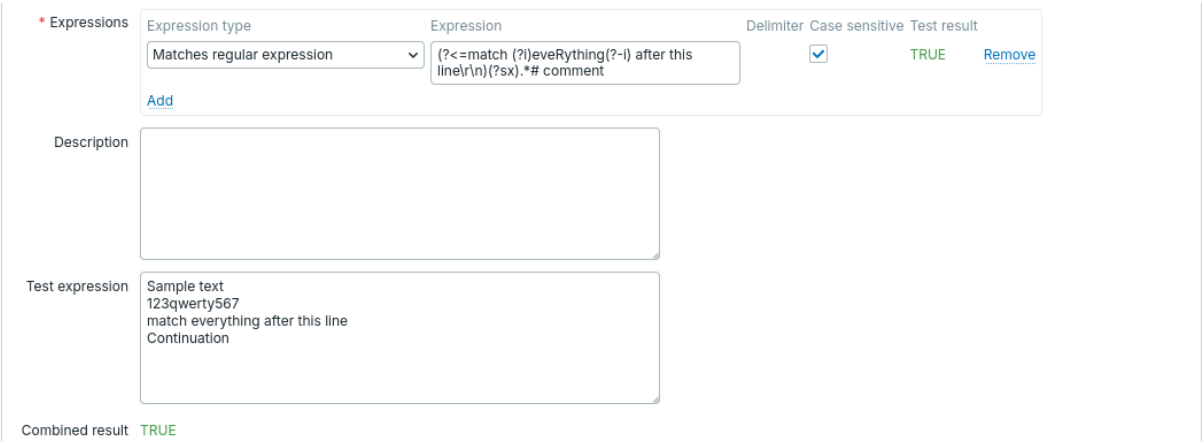
複数のインライン修飾子を含む次の正規表現を使用して、特定の行の後ろにある文字を一致させます。

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

ここで、各修飾子は次の意味を持ちます。

- （`?i`）- 大文字小文字を区別しない一致を有効にする;
- （`?-i`）- 大文字小文字を区別しない一致を無効にする;
- （`?s`）- dotall モードを有効にし、`.` が改行文字にも一致するようにする;

- （?x）- 拡張モードを有効にし、正規表現内の空白とコメントを無視できるようにする。



Test result と *Combined result* の両方で TRUE が表示されます。つまり、特定の行の後ろにある文字が一致しています。

Attention:
g 修飾子はインラインで指定できません。使用可能な修飾子の一覧は pcresyntax man page で確認できます。PCRE 構文の詳細については、PCRE HTML documentation を参照してください。

場所別の正規表現サポート

Location		Regular expression	Global regular expression	Multiline matching	Comments
エージェントアイテム					
	eventlog[]	Yes	Yes	Yes	`regexp`, `severity`, `source`, `eventid` parameters
	eventlog.count[]				`regexp`, `severity`, `source`, `eventid` parameters
	log[]				`regexp` parameter
	log.count[]				
	logrt[]		Yes/No		`regexp` parameter supports both, `file_regexp` parameter supports non-global expressions only
	logrt.count[]				
	proc.cpu.util[]		No	No	`cmdline` parameter
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	sensor[]				Linux 2.4 の `device` および `sensor` パラメータ
	system.hw.macaddr[]				`interface` parameter
	system.sw.packages[]				`regexp` parameter
	system.sw.packages.get[]				`regexp` parameter
	vfs.dir.count[]				`regex_incl`, `regex_excl`, `regex_excl_dir` parameters
	vfs.dir.get[]				`regex_incl`, `regex_excl`, `regex_excl_dir` parameters
	vfs.dir.size[]				`regex_incl`, `regex_excl`, `regex_excl_dir` parameters
	vfs.file.regexp[]			Yes	`regexp` parameter
	vfs.file.regmatch[]				
	web.page.regexp[]				
SNMP トラップ					

Location		Regular expression	Global regular expression	Multiline matching	Comments
	snmptrap[]	Yes	Yes	No	`regexp` parameter
アイテム値の前処理		Yes	No	No	`pattern` parameter
トリガー/計算アイテム用関数					
	count()	Yes	Yes	Yes	`operator` パラメータが *regexp* または *iregexp* の場合は `pattern` パラメータ
	countunique()	Yes	Yes		
	find()	Yes	Yes		
	logeventid()	Yes	Yes	No	`pattern` parameter
	logsource()				
低レベルディスカバリ					
	Filters	Yes	Yes	No	*Regular expression* field
	Overrides	Yes	No		In *matches*, *does not match* options for *Operation* conditions
アクション条件		Yes	No	No	In *matches*, *does not match* options for *Host name* and *Host metadata* autoregistration conditions
スクリプト		Yes	Yes	No	*Input validation rule* field
Web 監視		Yes	No	Yes	**regex:** プレフィックス付きの *Variables* *Required string* field
ユーザーマクロのコンテキスト		Yes	No	No	**regex:** プレフィックス付きのマクロコンテキスト
マクロ関数					
	regsub() iregsub()	Yes	No	No	`pattern` parameter
マップ内のリンクインジケーター		Yes	No	No	*Pattern* field (for text items)
アイコンマッピング		Yes	Yes	No	*Expression* field
値のマッピング		Yes	No	No	マッピングタイプが `regexp` の場合の *Value* field
エージェント設定パラメータ					
	AllowKeyRegexp DenyKeyRegexp	Yes	No	No	Pattern

イベントと解決

概要　Zabbix では、いくつかの種類のイベントが生成されます。

- トリガーイベント - トリガーのステータスが変化したとき（*OK*→ 障害 →*OK*）
- サービスイベント - サービスのステータスが変化したとき（*OK*→ 障害 →*OK*）
- ディスカバリーイベント - ホストまたはサービスが検出されたとき
- 自動登録イベント - アクティブエージェントがサーバーによって自動登録されたとき
- 内部イベント - アイテム/低レベルディスカバリルールがサポート対象外になったとき、またはトリガーが不明な状態になったとき

イベントにはタイムスタンプが付与され、通知メールの送信、チケットの作成、リモートコマンドの実行などのアクションの基礎として使用できます。

Web インターフェースでイベントの詳細を表示するには、監視 > 障害に移動します。
そこでイベントの日時をクリックするとイベント詳細ページが開き、メンテナンスのために抑制されたか、ユーザーによって手動で抑制されたかを含む、イベントに関する追加情報を確認できます。

障害の解決　トリガーによって作成された障害イベントは、障害の原因となった条件がもはや存在しない場合、自動的に解決されます。

障害イベントの管理に柔軟性と利便性を追加するため、次のオプションがあります。

- イベント相関
- イベントの確認と抑制
- 障害の手動クローズ
- メンテナンス期間 - 障害が抑制される期間を定義します
- タグ付け - 共通の特徴で障害にタグを付けます

トリガーイベント　トリガーの状態変化は、イベントの最も頻繁で最も重要な発生源です。トリガーの状態が変化するたびに、イベントが生成されます。イベントには、トリガーの状態変化の詳細、つまり、いつ発生したか、そして新しい状態が何であるかが含まれます。

トリガーによって作成されるイベントには、**Problem** と **OK** の 2 種類があります。

障害イベント

障害イベントは次の場合に作成されます。

- トリガーが OK 状態にあるときに、トリガー式が TRUE と評価された場合。
- トリガーで複数の障害イベント生成が有効になっている場合、トリガー式が TRUE と評価されるたび。

OK イベント

OK イベントは、関連する障害イベントを終了させ、次の 3 つのコンポーネントによって生成される場合があります。

- トリガー - 「OK event generation」および「OK event closes」の設定に基づく;
- イベント相関
- タスクマネージャー - イベントが手動でクローズされた場合

トリガーには、OK イベントの生成方法を制御する「OK イベント生成」設定があります。

- *Expression* - トリガーが障害状態にあるとき、その式が FALSE と評価されると OK イベントが生成されます。これは最も簡単な設定で、デフォルトで有効です。
- *Recovery expression* - トリガーが障害状態にあるとき、その式が FALSE と評価され、かつ復旧式が TRUE と評価されると OK イベントが生成されます。これは、トリガーの復旧条件が障害条件と異なる場合に使用できます。
- *None* - OK イベントは生成されません。これは、複数の障害イベント生成と組み合わせて、何かが発生したときに通知だけを送る用途に使用できます。

さらに、トリガーには、どの障害イベントをクローズするかを制御する「OK イベントでクローズ」設定があります。

- *All problems* - OK イベントは、トリガーによって作成されたすべての未クローズの障害をクローズします
- *All problems if tag values match* - OK イベントは、トリガーによって作成され、少なくとも 1 つの一致するタグ値を持つ未クローズの障害をクローズします。タグは「Tag for matching」トリガー設定で定義されます。クローズする障害イベントがない場合、OK イベントは生成されません。これは一般に、トリガーレベルのイベント相関と呼ばれます。

イベント相関

イベント相関（グローバルイベント相関とも呼ばれます）は、カスタムのイベントクローズ（OK イベントの生成）ルールを設定する方法です。

このルールでは、新しい障害イベントと既存の障害イベントをどのように組み合わせるかを定義し、対応する OK イベントを生成して、新しいイベントまたは一致したイベントをクローズできるようにします。

ただし、イベント相関は慎重に設定する必要があります。設定によっては、イベント処理のパフォーマンスに悪影響を及ぼしたり、意図した数を超えるイベントをクローズしたりする可能性があります（最悪の場合、すべての障害イベントがクローズされる可能性もあります）。設定に関するヒントを以下に示します。

1. 相関の範囲を常に狭めるため、制御イベント（古いイベントと組み合わせるイベント）に一意のタグを設定し、'new event tag' 相関条件を使用する
2. 'close old event' 操作を使用する場合は、古いイベントに基づく条件を必ず追加する。追加しないと、既存のすべての障害がクローズされる可能性がある
3. 異なる相関設定で使用される一般的なタグ名の使用を避ける

タスクマネージャー

トリガーで「手動クローズを許可」設定が有効になっている場合、そのトリガーによって生成された障害イベントを手動でクローズできます。これは、障害の更新時に Web インターフェースで行います。イベントは直接クローズされるのではなく、代わりに「イベントをクローズ」タスクが作成され、まもなくタスクマネージャーによって処理されます。タスクマネージャーは対応する OK イベントを生成し、障害イベントはクローズされます。

サービスイベント　サービスイベントは、これらのイベントに対するサービスアクションが有効になっている場合にのみ生成されます。この場合、サービス状態が変化するたびに新しいイベントが作成されます。

- 障害イベント - サービス状態が OK から 障害に変更されたとき
- OK イベント - サービス状態が 障害から OK に変更されたとき

イベントには、サービス状態の変更に関する詳細、つまり、いつ発生したか、および新しい状態が何であるかが含まれます。

検出イベント　Zabbix は、ネットワーク検出ルールで定義された IP 範囲を定期的にスキャンします。チェックの頻度は、ルールごとに個別に設定できます。ホストまたはサービスが検出されると、検出イベント（または複数のイベント）が生成されます。

Zabbix は次のイベントを生成します。

イベント	生成されるタイミング
Service Up	Zabbix がアクティブなサービスを検出するたび。
Service Down	Zabbix がサービスを検出できないたび。
Host Up	その IP に対して少なくとも 1 つのサービスが UP の場合。
Host Down	すべてのサービスが応答しない場合。
Service Discovered	サービスがダウン後に復旧した場合、または初めて検出された場合。
Service Lost	サービスが UP の後に失われた場合。
Host Discovered	ホストがダウン後に復旧した場合、または初めて検出された場合。
Host Lost	ホストが UP の後に失われた場合。

アクティブエージェントの自動登録イベント　アクティブエージェントの自動登録により、Zabbix にイベントが作成されます。

設定されている場合、以前は未知だったアクティブエージェントがチェックを要求したとき、またはホストメタデータが変更されたときに、アクティブエージェントの自動登録イベントが作成されます。サーバーは、受信したエージェントの IP アドレスとポートを使用して、新しい自動登録ホストを追加します。

詳細については、アクティブエージェントの自動登録 のページを参照してください。

内部イベント　内部イベントは、次の場合に発生します。

- アイテムの状態が'normal' から'unsupported' に変わる
- アイテムの状態が'unsupported' から'normal' に変わる
- ローレベルディスカバリルールの状態が'normal' から'unsupported' に変わる
- ローレベルディスカバリルールの状態が'unsupported' から'normal' に変わる
- トリガーの状態が'normal' から'unknown' に変わる
- トリガーの状態が'unknown' から'normal' に変わる

内部イベントを導入する目的は、内部イベントが発生したときにユーザーへ通知できるようにすることです。たとえば、アイテムが unsupported になってデータ収集を停止した場合などです。

内部イベントは、これらのイベントに対する内部アクションが有効な場合にのみ作成されます。内部イベントの生成を停止するには (たとえば、アイテムが unsupported になる場合)、*Alerts* > *Actions* > *Internal actions* で内部イベント用のすべてのアクションを無効にします。

Note:
内部アクションが無効な場合でも、オブジェクトが'unsupported' 状態にあると、そのオブジェクトの復旧イベントは引き続き作成されます。
内部アクションが有効な場合、オブジェクトが'unsupported' 状態にあると、そのオブジェクトの復旧イベントは、オブジェクトに対して'problem event' が作成されていなくても作成されます。

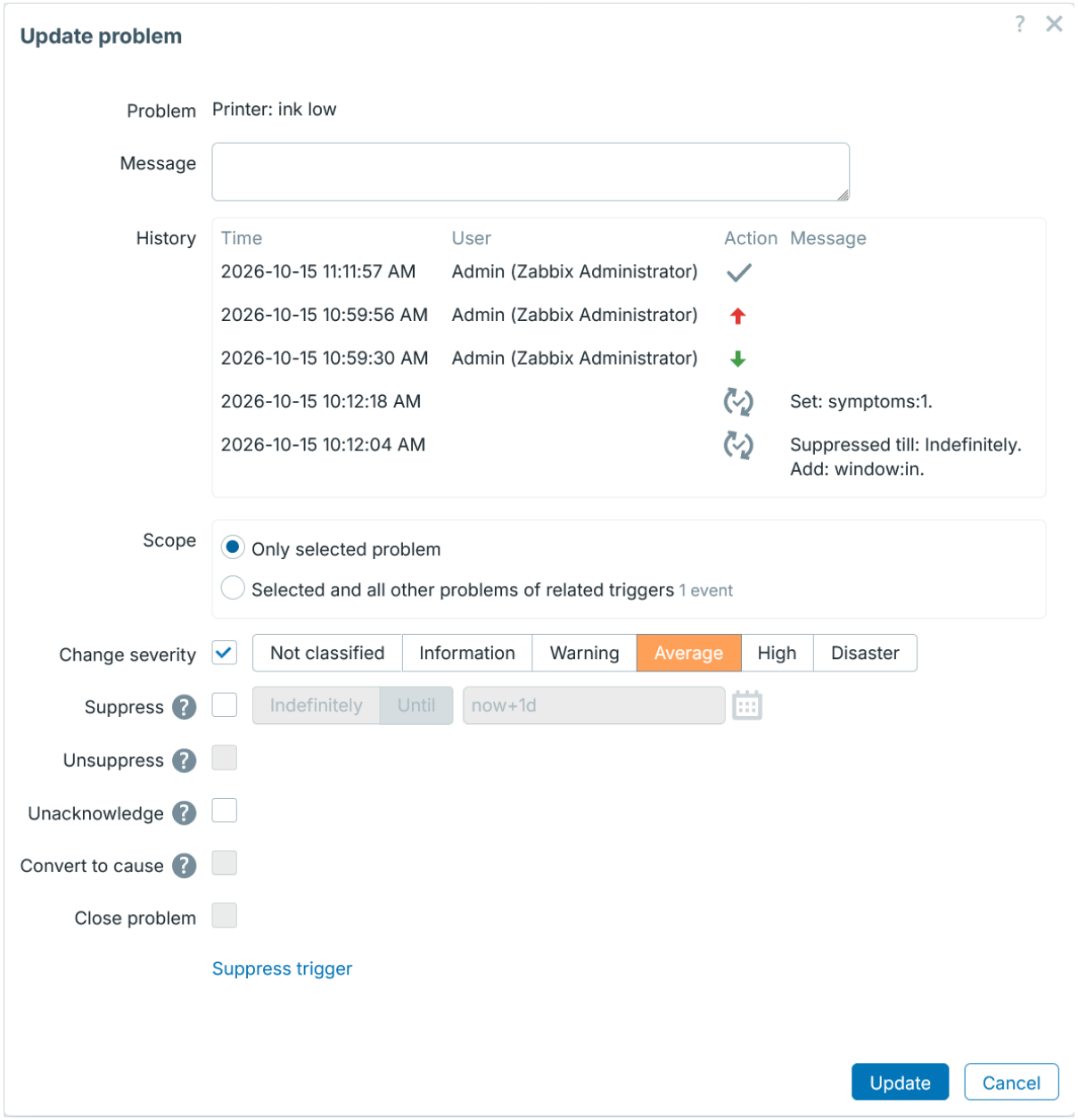
こちらも参照: unsupported アイテムに関する通知の受信

障害の確認と抑制

Zabbix の障害イベントは、ユーザーによって確認または抑制できます。

障害の更新　障害の更新ポップアップでは、次の操作を実行できます。

- 障害にコメントする。
- これまでのコメントとアクションを表示する。
- 障害の深刻度を変更する。
- 障害を抑制または抑制解除する。
- 障害を承認または承認解除する。
- 症状障害を原因障害に変更する。
- 障害を手動でクローズする。
- トリガーのアドホックメンテナンスを作成する。



必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
障害	障害が 1 つだけ選択されている場合は、障害名が表示されます。 複数の障害が選択されている場合は、*N* 個の障害が選択されていますと表示されます。
メッセージ	障害へのコメントとして入力するテキストを入力します（最大 2048 文字）。
履歴	障害に関する過去のアクティビティとコメントが、時刻およびユーザーの詳細とともに一覧表示されます。 ユーザーアクションを示すアイコンの意味については、イベントの詳細ページを参照してください。 更新対象として障害が 1 つだけ選択されている場合にのみ、履歴が表示されます。
範囲	深刻度の変更、障害の承認、手動クローズなどのアクションの範囲を定義します。 選択した障害のみ - このイベントだけに影響します。 選択した障害および関連するトリガーのその他すべての障害 - 障害の承認またはクローズの場合、このイベントと、これまでに承認またはクローズされていないその他すべての障害に影響します。 範囲にすでに承認またはクローズされている障害が含まれている場合、それらの障害が繰り返し承認またはクローズされることはありません。一方、メッセージおよび深刻度の変更操作の回数には制限がありません。
深刻度を変更	チェックボックスを選択し、深刻度ボタンをクリックして障害の深刻度を更新します。 深刻度を変更するチェックボックスは、選択した障害のうち少なくとも 1 つに読み書き権限がある場合に使用できます。更新をクリックすると、読み書き可能な障害のみが更新されます。 選択したトリガーのいずれにも読み書き権限がない場合、チェックボックスは無効になります。

パラメータ	説明
抑制	チェックボックスを選択して障害を抑制します。 無期限 - 無期限に抑制します。 指定時刻まで - 指定した時刻まで抑制します。絶対形式と相対形式のどちらの時刻形式も使用できます。例: `now+1d` - 現在から 1 日間（デフォルト） `now/w` - 現在の週の終わりまで `2022-05-28 12:00:00` - 絶対日時まで 単純な期間（例: 1d、1w）はサポートされていません。 このオプションを使用できるかどうかは、「障害の抑制」ユーザーロール設定によって決まります。 関連項目: 障害の抑制
抑制解除	チェックボックスを選択して障害の抑制を解除します。このチェックボックスは、選択した障害のうち少なくとも 1 つが抑制されている場合にのみ有効です。 このオプションを使用できるかどうかは、「障害の抑制」ユーザーロール設定によって決まります。
承認	チェックボックスを選択して障害を承認します。 このチェックボックスは、選択した障害の中に未承認の障害が少なくとも 1 つある場合に使用できます。 すでに承認されている障害に別の承認を追加することはできません（ただし、別のコメントを追加することはできます）。
承認解除	チェックボックスを選択して障害の承認を解除します。 このチェックボックスは、選択した障害の中に承認済みの障害が少なくとも 1 つある場合に使用できます。
原因に変換	チェックボックスを選択して、症状障害を原因障害に変換します。
障害をクローズ	チェックボックスを選択して、選択した障害を手動でクローズします。 障害をクローズするチェックボックスは、選択した障害のうち少なくとも 1 つについて、トリガー設定で 手動クローズを許可オプションが選択されている場合に使用できます。更新をクリックすると、クローズが許可されている障害のみがクローズされます。 手動でクローズできる障害がない場合、チェックボックスは無効になります。 すでにクローズされている障害が繰り返しクローズされることはありません。
トリガーを抑制	クリックして、トリガーのアドホックメンテナンスを開始します。 メンテナンス期間の設定フォームが開きます。フォームでは、現在から開始する 1 時間の 1 回限りのメンテナンス期間（ユーザーの設定およびプロファイルで設定可能）があらかじめ選択されています。該当するトリガーもあらかじめ選択されます。メンテナンス名には、`Ad-hoc: <event name>` の形式であらかじめ入力されます。

障害の承認　ユーザーが障害イベントについて通知を受け取った場合、Zabbix Web インターフェースに移動し、以下に示すいずれかの方法でその障害の障害更新ポップアップウィンドウを開いて、障害を承認できます。承認時には、その障害に対するコメントを入力でき、「対応中です」など、あるいはそのほか自由な内容を記入できます。

このようにすると、別のシステムユーザーが同じ障害を見つけた場合でも、すでに承認されているかどうかと、これまでのコメントをすぐに確認できます。

このようにして、複数のシステムユーザーが関与する障害解決のワークフローを、調整された形で進めることができます。

承認ステータスは、アクション操作 を定義する際にも使用されます。たとえば、イベントがしばらく承認されていない場合にのみ、上位管理者へ通知を送信するように定義できます。

イベントを承認してコメントを付けるには、ユーザーは対応するトリガーに対して少なくとも読み取り権限を持っている必要があります。障害の深刻度を変更したり障害をクローズしたりするには、ユーザーは対応するトリガーに対して読み書き権限を持っている必要があります。

障害を承認できる障害更新ポップアップウィンドウには、複数の方法でアクセスできます。

- 監視 > 障害で障害を選択し、一覧の下にある 一括更新をクリックします
- 次の場所にある障害の 更新列で 更新をクリックします:
  - ダッシュボード (障害および 深刻度別の障害ウィジェット)
  - 監視 > 障害
  - 監視 > 障害 > イベント詳細
- 次の場所で未解決の障害セルをクリックします:
  - ダッシュボード (トリガー概要ウィジェット)

ポップアップメニューには 更新オプションがあり、これを選択すると障害更新ウィンドウに移動します。

承認情報に基づいて、ダッシュボードまたはマップに表示する障害数を設定できます。設定するには、マップの設定と深刻度別の障害ダッシュボードウィジェットの両方で使用できる障害の表示オプションを選択します。

すべての障害数、未承認の障害数を合計から分けて表示するか、未承認の障害数のみを表示できます。

障害の更新情報（承認など）に基づいて、更新操作（メッセージの送信またはリモートコマンドの実行）を設定できます。

障害の抑制　障害の抑制を使用すると、後で対応できる障害を一時的に非表示にできます。これは、障害リストを整理し、最も緊急性の高い問題を優先して表示する場合に便利です。例えば、週末に問題が発生したものの、すぐに対応するほど緊急ではない場合、月曜日の朝まで「スヌーズ」できます。

障害の抑制では、単一の障害を非表示にできます。これは、メンテナンス中のホストに関連するすべての障害が非表示になる、ホストメンテナンスによる抑制とは異なります。

抑制された障害に対するトリガーアクションのオペレーションは、ホストメンテナンスの場合と同様に一時停止されます。

障害は、障害の更新 ウィンドウから抑制できます。ここでは、抑制はコメントの追加、深刻度の変更、確認などと並ぶ障害更新オプションの 1 つです。

また、同じ障害の更新ウィンドウから抑制を解除することもできます。

表示

抑制されると、障害は非表示になる前に、*Info* 列で点滅する

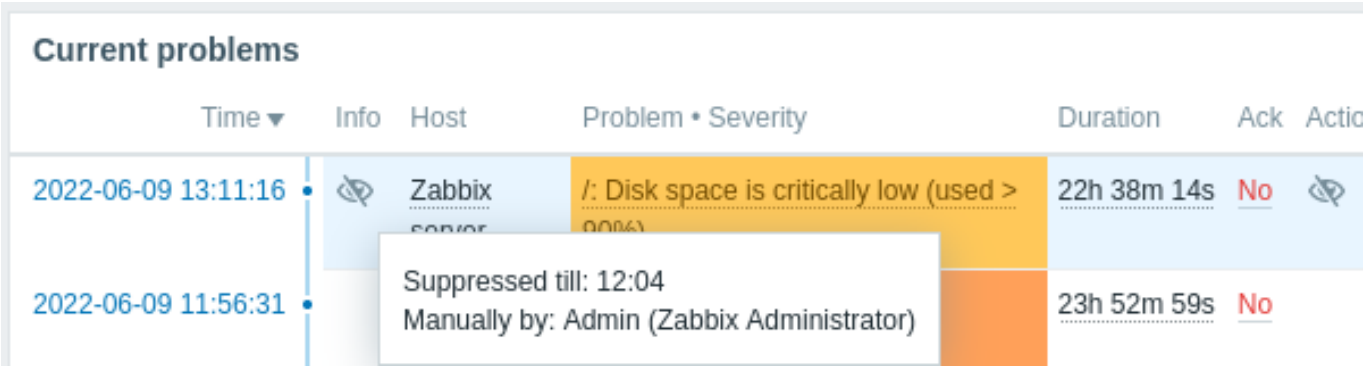
 抑制アイコンで示されます。

ホストメンテナンスが原因で抑制されている場合、*Info* 列の抑制アイコンはオレンジ色で表示されます。
メンテナンスと手動抑制の両方が適用される場合は、アイコンの色は手動抑制が優先されます。

抑制タスクが待機リストにある間、抑制アイコンは点滅します。
タスクマネージャーが障害を抑制すると、アイコンの点滅は停止します。
抑制アイコンが長時間点滅し続ける場合は、サーバーに問題があることを示している可能性があります。たとえば、サーバーが停止していてタスクマネージャーがタスクを完了できない場合です。
この同じロジックは抑制解除にも適用されます。
タスクが送信されてからサーバーがまだ完了していない短い間は、抑制解除アイコンが点滅します。

抑制された障害は、障害フィルター/ウィジェットの設定に応じて、非表示にも表示にもできます。

障害一覧に表示される場合、抑制された障害には抑制アイコンが付与され、マウスオーバーすると抑制の詳細が表示されます。



抑制の詳細は、*Actions* 列の抑制アイコンにマウスを合わせたときに表示されるポップアップにも表示されます。

障害の手動クローズ

概要　一般的に、障害イベントはトリガーの状態が `Problem` から `OK` に変わると自動的に解決されますが、トリガー式によって障害が解決されたかどうかを判断するのが難しい場合があります。
そのような場合は、障害を手動で解決する必要があります。

たとえば、*syslog* が、最適なパフォーマンスのために一部のカーネルパラメータを調整する必要があると報告することがあります。
この場合、問題は Linux 管理者に通知され、管理者が修正したうえで、障害を手動でクローズします。

障害を手動でクローズできるのは、*Allow manual close* オプションが有効なトリガーのみです。

障害が「手動でクローズ」されると、Zabbix は Zabbix サーバー向けに新しい内部タスクを生成します。
その後、*task manager* プロセスがこのタスクを実行し、OK イベントを生成することで、障害イベントをクローズします。

生成された *OK* イベントには、そのイベントに対して解決されたイベントタグの完全なセットが含まれます（テンプレート、ホスト、トリガーから継承されたタグを含む）。
これらのタグは通知や、{EVENT.RECOVERY.TAGS} および{EVENT.RECOVERY.TAGSJSON} などのマクロで利用できます。

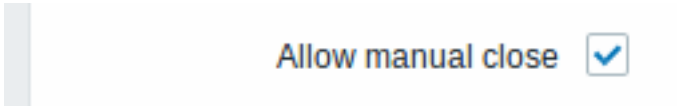
手動でクローズされた障害は、基になるトリガーが今後二度と *Problem* 状態にならないことを意味するわけではありません。
トリガー式は再評価され、次のような場合に障害が発生する可能性があります。

- トリガー式に含まれる任意のアイテムに新しいデータが到着したとき（スロットリング前処理ステップによって破棄された値は受信済みとは見なされず、トリガー式の再評価は行われません）；
- 式でdate and time および/またはnodata() 関数が使用されているとき。

設定　障害を手動でクローズするには、2 つの手順が必要です。

トリガーの設定

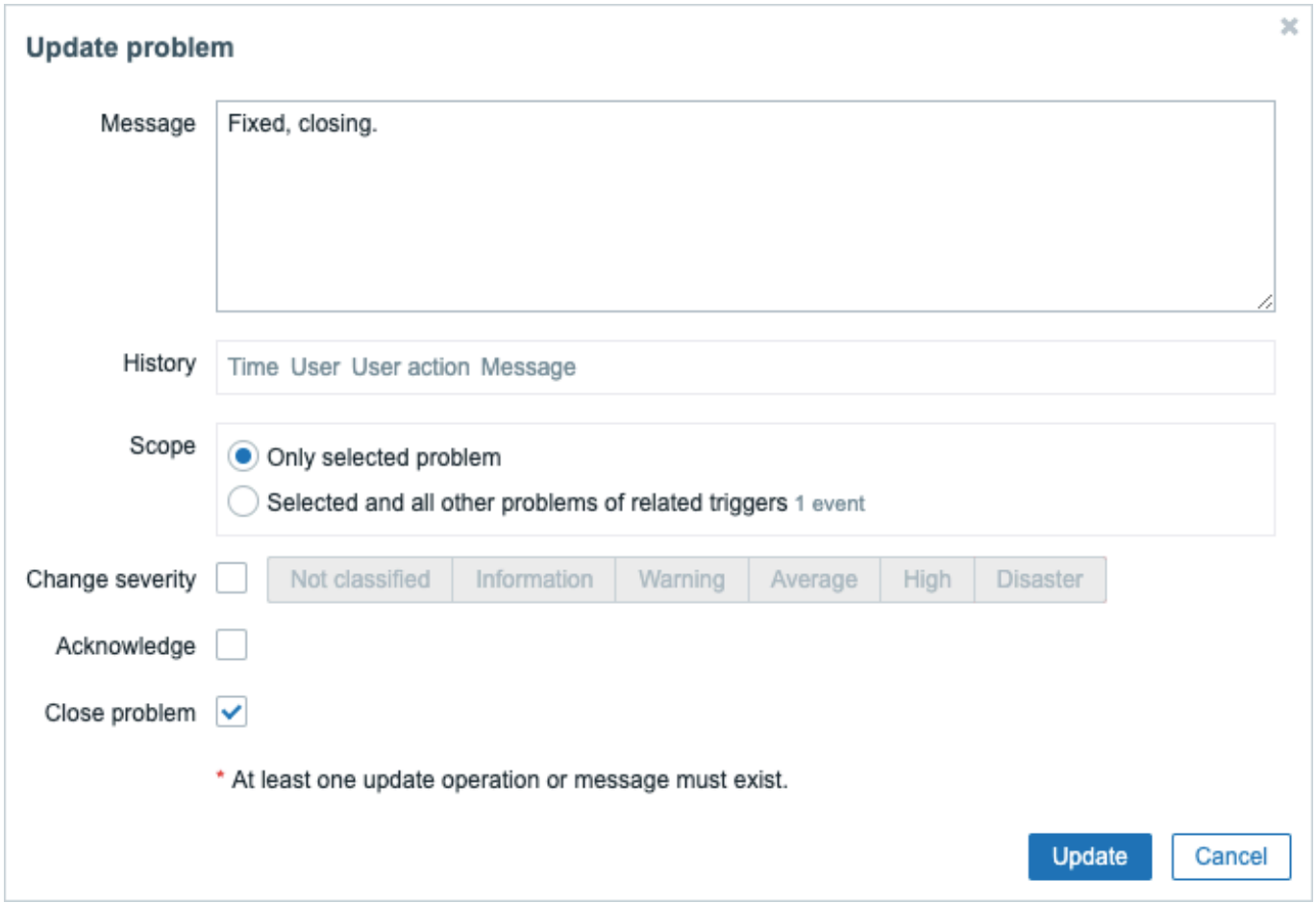
トリガーの設定で、手動クローズを許可オプションを有効にします。



障害更新ウィンドウ

Manual close フラグが設定されたトリガーで障害が発生した場合、その障害の障害更新 ポップアップウィンドウを開き、障害を手動でクローズできます。

障害をクローズするには、フォームで *Close problem* オプションを選択し、*Update* をクリックします。



必須入力フィールドには赤いアスタリスクが付いています。

このリクエストは Zabbix サーバーによって処理されます。通常、障害のクローズには数秒かかります。その処理中、*Monitoring* > *Problems* では障害のステータスとして *CLOSING* が表示されます。

検証　障害が手動でクローズされたことは、次の方法で確認できます。

- 監視 > 障害から表示できるイベント詳細で確認する。
- 通知メッセージ内で {EVENT.UPDATE.HISTORY} マクロを使用すると、この情報を取得できます。

メンテナンス期間

概要

メンテナンスは、あらかじめ定義した期間中に障害を抑制するために使用します。

次のエンティティに対してメンテナンス期間を定義できます。

- ホストグループ
- ホスト
- トリガー
- イベント名
- タグ

特定のタグまたはイベント名に対してメンテナンスを定義した場合、メンテナンスは対応するトリガーに対してのみ有効になります。ホストまたはホストグループのその他のトリガーはメンテナンス中になりません。

メンテナンスには、データ収集ありと データ収集なしの 2 種類があります。

データ収集ありのメンテナンス中は、トリガーは通常どおり処理され、必要に応じてイベントが作成されます。ただし、アクション設定で 抑制された障害に対する操作を一時停止オプションが選択されている場合、メンテナンス中のエンティティに対する障害のエスカレーションは一時停止されます。この場合、通知の送信やリモートコマンドの実行を含むエスカレーションステップは、メンテナンス期間中は無視されます。障害の復旧および更新操作はメンテナンス中も抑制されず、抑制されるのはエスカレーションのみであることに注意してください。メンテナンス中に開始した障害については、復旧通知は送信されません。

たとえば、障害の開始から 0 分、30 分、60 分後にエスカレーションステップがスケジュールされていて、実際の障害発生から 10 分後から 40 分後まで 30 分間のメンテナンスが実施される場合、障害が引き続き存在していれば、ステップ 2 と 3 は 30 分遅れて、60 分後と 90 分後に実行されます。同様に、メンテナンス中に障害が発生した場合、エスカレーションはメンテナンス終了後に開始されます。

メンテナンス中も通常どおり（遅延なく）障害通知を受信するには、アクション設定で 抑制された障害に対する操作を一時停止オプションの選択を解除する必要があります。

> **Note:**
> （トリガーの式で使用されている）少なくとも 1 つのホストがメンテナンスモードでない場合、Zabbix は障害通知を送信します。

メンテナンス中も Zabbix サーバーを稼働させておく必要があります。メンテナンス期間に変更がある場合、メンテナンスは 1 分ごと、または設定キャッシュが再読み込みされた時点で再計算されます。

タイマープロセスは、毎分 0 秒にホストのステータスをメンテナンスに変更する必要があるか、またはメンテナンスから戻す必要があるかを確認します。さらに、設定の更新後にメンテナンス期間に変更があるかどうかに基づいて、メンテナンスを開始または停止する必要があるかをタイマープロセスが毎秒確認します。したがって、メンテナンス期間の開始または停止の速度は、設定の更新間隔（デフォルトでは 10 秒）によって異なります。メンテナンス期間の変更には、有効期間の開始/有効期間の終了設定は含まれないことに注意してください。また、既存のアクティブなメンテナンス期間にエンティティ（ホストグループ、ホスト、トリガーなど）が追加された場合、変更は次の分の開始時にタイマープロセスによってのみ有効になります。

ホストがメンテナンスに入ると、Zabbix サーバーのタイマープロセスは、抑制が必要かどうかを確認するため、すべての未解決の障害を読み取ります。未解決の障害が多数ある場合、パフォーマンスに影響する可能性があります。また、メンテナンスが設定されていない場合でも、Zabbix サーバーの起動時にすべての未解決の障害が読み取られます。

メンテナンスの種類（データ収集なしのメンテナンスを含む）にかかわらず、Zabbix サーバー（またはプロキシ）は常にデータを収集します。データ収集なしが設定されている場合、データは後でサーバーによって無視されます。

データ収集なしのメンテナンスが終了した場合、`nodata()` 関数を使用するトリガーは、確認対象の期間中に次のチェックが行われるまで発火しません。

ホストがメンテナンス中にログアイテムが追加され、メンテナンスが終了した場合、メンテナンス終了後の新しいログファイルエントリのみが収集されます。

データ収集なしのメンテナンス中のホストにタイムスタンプ付きの値が送信された場合（Zabbix senderを使用する場合など）、この値は破棄されます。ただし、期限切れのメンテナンス期間に対してタイムスタンプ付きの値を送信することは可能で、その値は受け入れられます。

メンテナンス期間またはメンテナンスオプション（ホストグループ、ホスト、トリガーなど）をユーザーが変更した場合、変更は設定キャッシュの同期後にのみ有効になります。

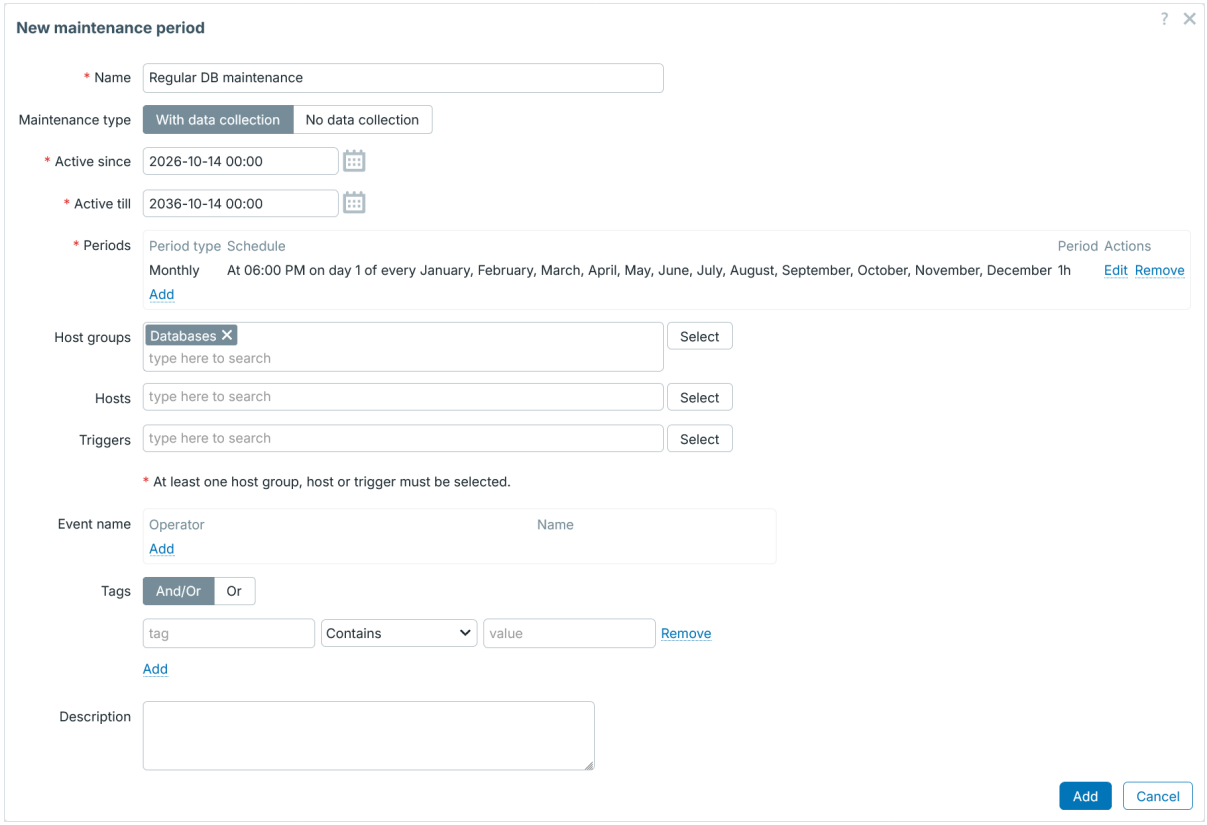
設定

メンテナンス期間を直ちに開始するには、次の手順を実行します。

1. 監視 > 障害に移動します。
2. 障害名をクリックします。
3. イベントメニュー（ホスト、トリガー、イベント名、タグ）からメンテナンスオプションを選択します。
4. フォームでメンテナンスパラメータを変更します。

計画されたメンテナンス期間を設定するには、次の手順を実行します。

1. データ収集 > メンテナンスに移動します。
2. メンテナンス期間の作成（または既存のメンテナンス期間の名前）をクリックします。
3. フォームにメンテナンスパラメータを入力します。



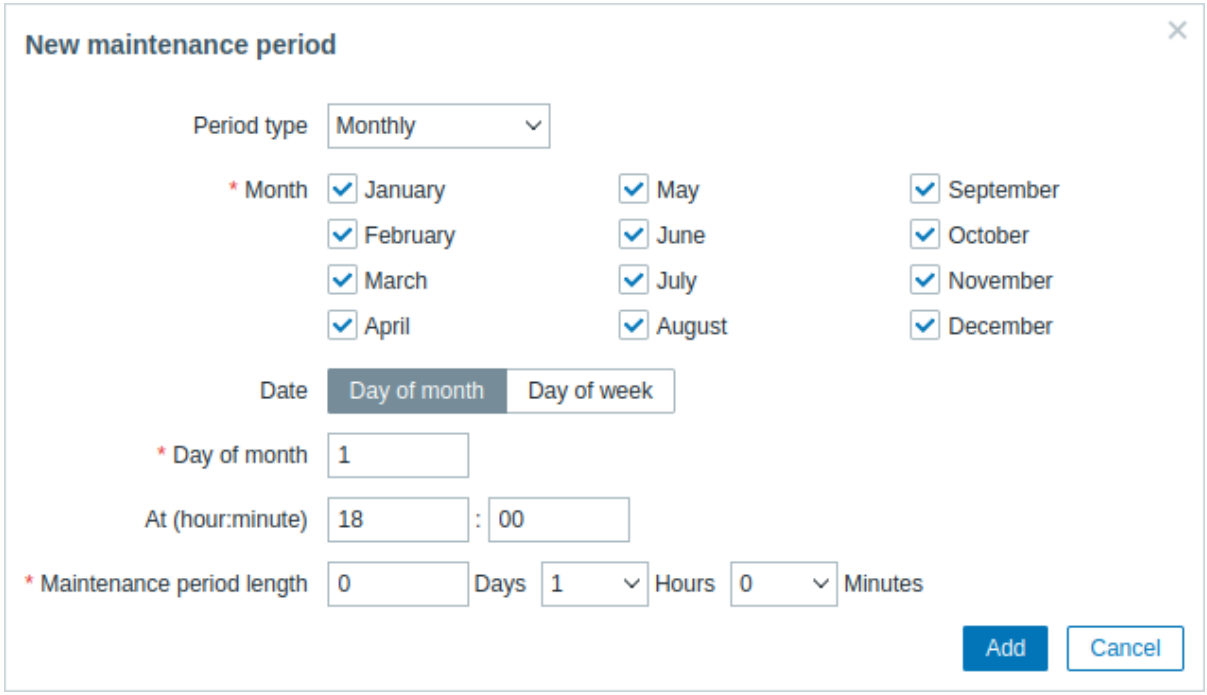
必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
名前	メンテナンス期間の名前。
メンテナンスタイプ	メンテナンスには、次の 2 種類を設定できます。 データ収集あり - メンテナンス中もサーバーによってデータが収集され、トリガーが処理されます。 データ収集なし - データは収集される場合がありますが、メンテナンス中はデータベースに保存されず、トリガー（nodata() 関数を含む）は発生しません。 各タイプが可用性レポートに与える影響については、メンテナンス期間の影響を参照してください。
開始日時	メンテナンス期間の実行が有効になる日時。 注: この時刻だけを設定しても、メンテナンス期間は有効になりません。メンテナンス期間は期間で設定する必要があります（以下を参照）。
終了日時	メンテナンス期間の実行が有効でなくなる日時。
期間	このブロックでは、メンテナンスを実施する正確な日付と時間を定義できます。Addをクリックすると、柔軟なメンテナンス期間フォームを含むポップアップウィンドウが開き、メンテナンススケジュールを定義できます。詳しくは、メンテナンス期間を参照してください。
ホストグループ	メンテナンスを有効にするホストグループを選択します。指定したホストグループのすべてのホストでメンテナンスが有効になります。このフィールドはオートコンプリートに対応しているため、入力を開始すると、利用可能なすべてのホストグループがドロップダウンに表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。そのため、ネストされたグループのホストでもメンテナンスが有効になります。
ホスト	メンテナンスを有効にするホストを選択します。 このフィールドはオートコンプリートに対応しているため、入力を開始すると、利用可能なすべてのホストがドロップダウンに表示されます。

パラメータ	説明
トリガー	メンテナンスを有効にするトリガーを選択します。 このフィールドはオートコンプリートに対応しているため、入力を開始すると、利用可能なすべてのトリガーがドロップダウンに表示されます。
イベント名	メンテナンスを有効にするイベント名を指定します。 複数の条件を設定できます。 各条件では、次の 2 つの演算子を使用できます。 含む - タグの値に入力した文字列が含まれる場合、指定したタグ名を含めます（部分文字列の一致、大文字と小文字を区別しない）。 含まない - タグの値に入力した文字列が含まれる場合、指定したタグ名を除外します（部分文字列の一致、大文字と小文字を区別しない）。
タグ	メンテナンスを有効にするタグを指定します。 複数の条件を設定できます。タグ名の照合では、常に大文字と小文字が区別されます。 各条件では、次の複数の演算子を使用できます。 含む - タグの値に入力した文字列が含まれる場合、指定したタグ名を含めます（部分文字列の一致、大文字と小文字を区別する）。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別する）。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別する）。 含まない - タグの値に入力した文字列が含まれる場合、指定したタグ名を除外します（部分文字列の一致、大文字と小文字を区別しない）。 条件には、次の 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は、Or 条件でグループ化されます。 **Or** - 1 つの条件を満たせば十分です。 タグを指定できるのは、メンテナンスタイプでデータ収集ありを選択した場合のみです。
説明	メンテナンス期間の説明。

メンテナンス期間

メンテナンス期間ウィンドウは、定期メンテナンスまたは 1 回限りのメンテナンスの時間をスケジュールするために使用します。フォームは動的で、選択した *Period type* に応じて利用可能なフィールドが変わります。



Period type	Description
One time only	1 回限りのメンテナンス期間を設定します: *Date* - メンテナンス期間の日時; *Maintenance period length* - メンテナンスを有効にする期間。

Period type	Description
Daily	日次のメンテナンス期間を設定します: *Every day(s)* - メンテナンスの頻度 (1 - *(default)* 毎日、2 - 2 日ごと、など); *At (hour:minute)* - メンテナンスを開始する時刻; *Maintenance period length* - メンテナンスを有効にする期間。 *Every day(s)* パラメータが”1” より大きい場合、開始日は *Active since* の時刻が属する日になります。例: - *Active since* が”2021-01-01 12:00”、*Every day(s)* が”2”、*At (hour:minute)* が”23:00” の場合、最初のメンテナンス期間は 1 月 1 日 23:00 に開始し、2 回目のメンテナンス期間は 1 月 3 日 23:00 に開始します; - *Active since* が”2021-01-01 12:00”、*Every day(s)* が”2”、*At (hour:minute)* が”01:00” の場合、最初のメンテナンス期間は 1 月 3 日 01:00 に開始し、2 回目のメンテナンス期間は 1 月 5 日 01:00 に開始します。
Weekly	週次のメンテナンス期間を設定します: *Every week(s)* - メンテナンスの頻度 (1 - *(default)* 毎週、2 - 2 週ごと、など); *Day of week* - メンテナンスを実施する曜日; *At (hour:minute)* - メンテナンスを開始する時刻; *Maintenance period length* - メンテナンスを有効にする期間。 *Every week(s)* パラメータが”1” より大きい場合、開始週は *Active since* の時刻が属する週になります。例については、上記の *Daily* の説明を参照してください。
Monthly	月次のメンテナンス期間を設定します: *Month* - 定期メンテナンスを実施する月をすべて選択します; *Date:* ***Day of month*** - メンテナンスを毎月同じ日付に実施する場合 (たとえば、毎月 1 日) はこのオプションを選択し、表示される *Day of month* フィールドで必要な日を選択します; *Date:* ***Day of week*** - メンテナンスを特定の日にのみ実施する場合 (たとえば、毎月第 1 月曜日) はこのオプションを選択し、ドロップダウンで必要な月内の週 (first, second, third, fourth, or last) を選択してから、メンテナンス日をチェックボックスで選択します; *At (hour:minute)* - メンテナンスを開始する時刻; *Maintenance period length* - メンテナンスを有効にする期間。

Attention:
メンテナンス期間を作成する際には、それを作成するユーザーのtime zone が使用されます。ただし、繰り返しメンテナンス期間（*Daily*、*Weekly*、*Monthly*）をスケジュールする場合は、Zabbix サーバーのタイムゾーンが使用されます。繰り返しメンテナンス期間の動作を予測可能にするには、Zabbix のすべての部分で共通のタイムゾーンを使用する必要があります。

完了したら、*Add* をクリックしてメンテナンス期間を *Periods* セクションに追加します。

夏時間（DST）の変更は、メンテナンスの継続時間に影響しないことに注意してください。たとえば、通常は 01:00 に開始して 03:00 に終了する 2 時間のメンテナンスが設定されているとします。

- 1 時間のメンテナンス後（02:00）に DST の変更が発生し、現在時刻が 02:00 から 03:00 に変わった場合、メンテナンスはさらに 1 時間継続します（04:00 まで）。
- 2 時間のメンテナンス後（03:00）に DST の変更が発生し、現在時刻が 03:00 から 02:00 に変わった場合、2 時間が経過しているため、メンテナンスは停止します。
- メンテナンス期間が DST の変更によって飛ばされる 1 時間の間に開始する場合、メンテナンスは開始されません。

メンテナンス期間が”1 day” に設定されている場合（Zabbix は日数を時間で計算するため、メンテナンスの実際の期間は 24 時間です）、00:00 に開始して翌日の 00:00 に終了するときは、次のようになります。

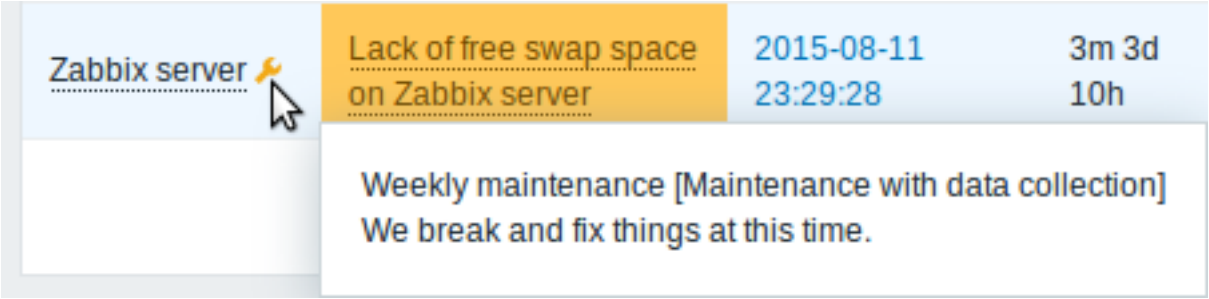
- 現在時刻が 1 時間進む場合、メンテナンスは翌日の 01:00 に停止します。
- 現在時刻が 1 時間戻る場合、メンテナンスはその日の 23:00 に停止します。

表示

メンテナンス中のホストの表示

ホスト名の横にオレンジ色のレンチアイコン が表示されている場合、そのホストがメンテナンス中であることを示します。これは次の場所で確認できます。

- ダッシュボード
- 監視 > 障害
- インベントリ > ホスト > ホストインベントリ詳細
- データ収集 > ホスト（ステータス列を参照）



アイコンにマウスポインターを合わせると、メンテナンスの詳細が表示されます。

さらに、メンテナンス中のホストは 監視 > マップでオレンジ色の背景で表示されます。

抑制された障害の表示

通常、メンテナンス中のホストの障害は抑制され、つまり Web インターフェースには表示されません。

抑制された障害は、手動で抑制されたために非表示になっている場合もあります。

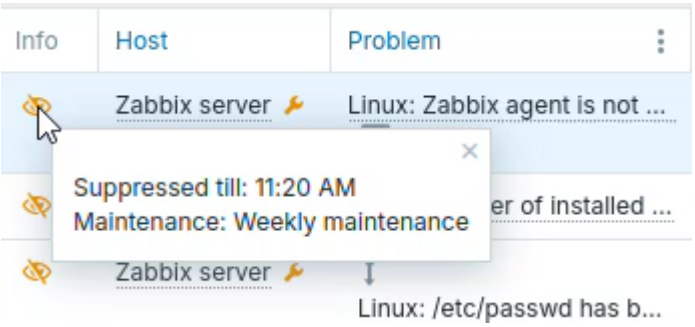
次の場所で 抑制された障害を表示オプションを選択すると、抑制された障害を表示できます。

- ダッシュボード（障害ホスト、障害、深刻度別の障害、トリガー概要ウィジェットの設定内）
- 監視 > 障害（フィルター内）
- 監視 > マップ（マップ設定内）
- グローバル通知（ユーザープロファイル設定内）

抑制された障害が表示されると、次のアイコンが表示されます: 。抑制の理由がホストのメンテナンスである場合は、次のアイコンが表示されます。

-  - ホストのメンテナンスにより障害が抑制されています。
-  - ホストのメンテナンス後に障害の抑制が解除されています。

アイコンにカーソルを合わせると、詳細が表示されます。イベントの詳細では、そのイベントがメンテナンスによって抑制されたのか、手動で抑制されたのかを確認できます。メンテナンスと手動抑制の両方が適用される場合、アイコンの色は手動抑制が優先されます。



メンテナンス中のキューの計算

Zabbix Web インターフェース（Administration > Queue）に表示されるキューは、Zabbix サーバーによって計算されます。
これらには *No data collection* メンテナンス中のアイテムは含まれず、値が遅延している場合でも、これらのアイテムのキュー長は常に 0 です。
With data collection メンテナンス中の遅延したアイテムは、引き続きキューにカウントされます。

Zabbix プロキシ は、Zabbix サーバーとプロキシ間でメンテナンス設定の同期が行われないため、メンテナンス期間を認識しません。Zabbix プロキシ上で計算される内部チェック（たとえば、`zabbix[queue,,]` や`zabbix[stats,,,queue,,]`）は、Zabbix サーバー上のメンテナンス状態に関係なく、遅延したアイテムを報告します。

その結果、*No data collection* メンテナンス中の同じアイテムについて、Zabbix Web インターフェースと Zabbix プロキシ上の内部チェックで、異なるキュー長が報告される場合があります。

タグ付け

タグには複数の用途があり、特に障害 を識別するために使用されます。
エンティティにタグが付けられると、そのタグ付きエンティティに関連する新しいイベントは、そのタグを継承します。
たとえば、次のようになります。

- タグ付きテンプレートの場合 - そのテンプレートのトリガーによって作成された任意のホスト障害は、テンプレートのタグを継承します。
- タグ付きホストの場合 - 任意のホスト障害はホストのタグを継承します。
- タグ付きアイテム/Web シナリオの場合 - 任意のアイテム/Web シナリオ障害は、アイテム/Web シナリオのタグを継承します。
- タグ付きトリガーの場合 - そのトリガーによって作成された任意の障害は、トリガーのタグを継承します。

障害イベントは、テンプレート、ホスト、アイテム/Web シナリオ、トリガーというエンティティの全チェーンからすべてのタグを継承します。
同一の `tag:value` の組み合わせは（解決済みマクロの後で）1 つにまとめられ、重複が回避されます。

手動クローズ によって生成された復旧イベントにも、テンプレート、ホスト、アイテム/Web シナリオ、トリガーから継承された解決済みイベントタグが含まれます。
これらのタグは通知内および`{EVENT.RECOVERY.TAGS}` や`{EVENT.RECOVERY.TAGSJSON}` などのマクロから利用できます。

カスタムイベントタグを使用すると、より柔軟に設定できます。
たとえば、次のような用途があります。

- イベント相関 をイベントタグに基づいて設定できます。
- アクション条件 をイベントタグに基づいて設定できます。
- アイテム障害をイベントタグに基づいてグループ化できます。
- 障害タグを使用して、障害をサービス にマッピングできます。

エンティティには、同じタグ名で異なるタグ値を付けることができます (たとえば、`component:memory` と `component:storage`)。同様に、エンティティは値のないタグと、同じタグの値付きタグを持つことができます（たとえば、`database` と `database:postgresql`）。
このようなタグは重複とは見なされません。

タグは、タグ名とタグ値で構成されます。
エンティティにタグを付ける場合、名前だけを使用することも、値と組み合わせることもできます（たとえば、`mysql`、`jira`、`target:mysql`、`service:jira` など）。

タグは、さまざまなエンティティに対して定義できます。

- テンプレート
- ホスト
- アイテム
- Web シナリオ
- トリガー
- サービス
- テンプレートのアイテムとトリガー
- ホスト、アイテム、トリガーのプロトタイプ

Templates、*Hosts*、*Items*、*Triggers*、*Web scenarios* の一覧とそれらのプロトタイプでは、*Tags* 列に通常のタグと継承されたタグの両方が表示されます。
継承されたタグにマウスオーバーするかクリックすると、JavaScript のツールチップに”Inherited tag” と表示されます。
継承されたタグが通常のタグとしても存在する場合、表示される一覧に応じて異なるツールチップテキストが表示されます（たとえば、テンプレート 一覧では”Inherited and template tag”、ホスト 一覧では”Inherited and host tag” など）。
Web インターフェースの他の場所では、通常のタグと継承されたタグはまとめて表示され（重複は削除されます）、アイコンや追加のツールチップテキストは表示されません。

> **Note:**
> タグの定義に関する 一般的な推奨事項 と、テンプレート、アイテム、トリガー、および 低レベルディスカバリルール に関する具体的なガイダンスについては、Zabbix 公式ガイドラインを参照してください。

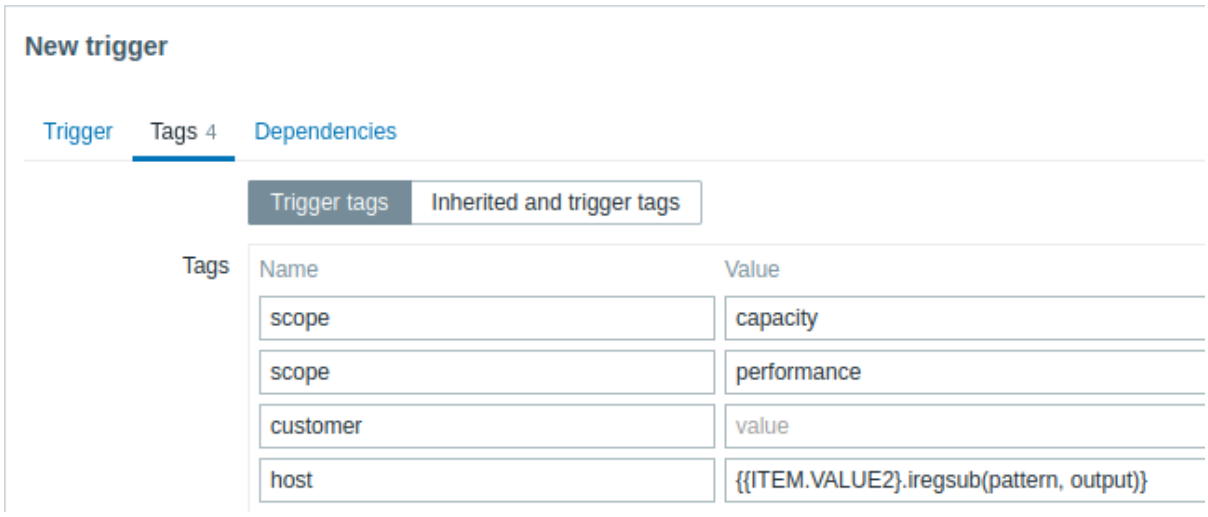
ユースケース　タグ付けの一般的なユースケースは次のとおりです。

1. トリガーイベントをマークする:
   - トリガータグを定義します（例: `scope:performance`）。
   - このトリガーによって作成された障害には、トリガータグが付与されます。
2. テンプレート継承の障害をマークする:
   - テンプレートタグを定義します（例: `target:mysql`）。
   - このテンプレートのトリガーによって作成された障害には、テンプレートタグが付与されます。
3. ホストの障害をマークする:
   - ホストタグを定義します（例: `service:jira`）。

- このホストのトリガーによって作成された障害には、ホストタグが付与されます。

4. 関連するアイテムをフィルタリングする:
   - アイテムタグを定義します（例: `component:cpu`）。
   - *Monitoring* > *Latest data* では、`component:cpu` タグでアイテムをフィルタリングできます。
5. アイテム値から抽出した情報をタグ値として使用する:
   - マクロをタグ値として持つタグを定義します（例: `tag-name:{{ITEM.VALUE<N>}.regsub()}`）。
   - *Monitoring* > *Problems* では、アイテム値から抽出されたデータにタグ値が解決され、障害に反映されます。
6. ログファイル内の障害を識別し、個別にクローズする:
   - アイテム値からマクロを使用して値を抽出するログ監視アイテム のトリガーに、トリガータグを定義します（例: `service:{{ITEM.VALUE<N>}.regsub()}`）。
   - トリガー設定 で、イベント相関 を設定します:
     - *PROBLEM event generation mode* を”Multiple” に設定する;
     - *OK event closes* を”All problems if tag values match” に設定する;
     - 照合用のタグを設定する。
   - ログアイテムのトリガーによって作成された障害にはトリガータグが付与され、個別にクローズされます。
7. 通知をフィルタリングする:
   - トリガータグを定義します（例: trigger1 には `scope:security`、trigger2 には `scope:availability`）。
   - アクション条件 でタグフィルタリングを使用し、タグデータに一致するイベントのみ通知を受け取ります。
8. 通知内の障害を識別する:
   - トリガータグを定義します。
   - 障害通知で{EVENT.TAGS} マクロを使用します。
   - 障害通知にはトリガータグが含まれるため、どのアプリケーション/サービスに属する通知かを識別しやすくなります。
9. テンプレートタグを使用して設定作業を簡素化する:
   - テンプレートトリガータグを定義します。
   - このテンプレートトリガーから作成されたトリガーには、そのタグが付与されます。
10. ローレベルディスカバリ（LLD）からのタグを持つトリガーを作成する:
    - タグ名または値に LLD マクロを含むトリガープロトタイプタグを定義します（例: `scope:{#FSNAME}`）。
    - このトリガープロトタイプから作成されたトリガーには、そのタグが付与されます。
11. サービスタグを使用してサービスを照合する:
    - サービスタグ を定義します。
    - 一致するタグを持つサービスに対してサービスアクション を設定します。
    - さらに、SLA 計算のためにサービスタグを使用してサービスをSLA にリンクします。
12. サービス障害タグを使用してサービスを障害にリンクする:
    - サービス設定 で障害タグ を定義します（例: `target:mysql`）。
    - 一致するタグを持つ障害は自動的にサービスに関連付けられ、設定されたサービス状態計算ルールに基づいてサービス状態が変化します。
13. ホストがメンテナンスモードのときに障害を抑制する:
    - メンテナンス期間設定 でタグを定義します。
    - 定義されたタグを持つ障害は抑制されます。
14. ユーザーグループにアクセス権を付与する:
    - ユーザーグループ設定 でタグを定義します。
    - ユーザーグループ内のユーザーは、定義されたタグを持つ障害のみを表示できます。

設定　タグは専用のタブで定義できます。たとえば、トリガーの設定 で指定できます。



マクロのサポート　タグ内の組み込みマクロ とユーザーマクロ は、障害発生時に解決されます。
障害が発生するまでは、これらのマクロは Zabbix Web インターフェースで未解決のまま表示されます。

ローレベルディスカバリマクロ は、ディスカバリ処理中に解決されます。

次のマクロは、トリガーのタグ名および値で使用できます。

- {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}, および {HOST.ID} の組み込みマクロ
- {INVENTORY.*} の組み込みマクロ（トリガー式で、1 つまたは複数のホストのホストインベントリ値を参照するために使用）
- ユーザーマクロおよびコンテキスト付きユーザーマクロ（コンテキストにはローレベルディスカバリマクロを含めることができます）
- ローレベルディスカバリマクロ（トリガープロトタイプのタグでのみ）

次のマクロは、テンプレート、ホスト、およびアイテム/web シナリオのタグ名と値で使用できます。

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} および {HOST.ID} の組み込みマクロ
- {INVENTORY.*} の組み込みマクロ
- ユーザーマクロ
- ローレベルディスカバリマクロ（ホストおよびアイテムプロトタイプのタグでのみ）

次のマクロは、トリガーに基づく通知で使用できます。

- {EVENT.TAGS} および {EVENT.RECOVERY.TAGS} の組み込みマクロ（これらのマクロは、イベントタグまたは復旧イベントタグのカンマ区切りリストに展開されます）
- {EVENT.TAGSJSON} および {EVENT.RECOVERY.TAGSJSON} の組み込みマクロ（これらのマクロは、イベントタグオブジェクトまたは復旧イベントタグオブジェクトを含む JSON 配列に展開されます）

トリガータグにおける部分文字列の抽出

部分文字列の抽出は、マクロfunction を使用してタグ名またはタグ値を設定する際にサポートされています。
この関数は、サポートされている マクロによって取得された値に対して正規表現を適用します。
例:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

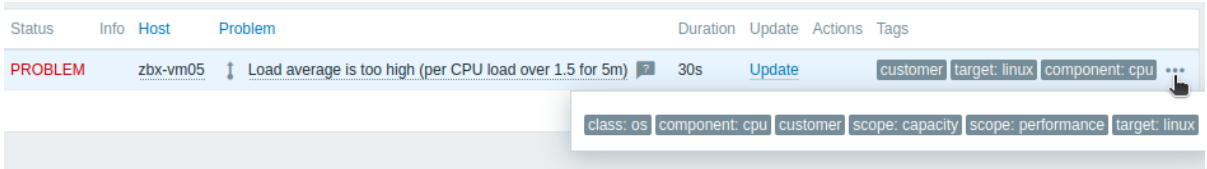
マクロの解決後にタグ名または値が 255 文字を超える場合は、255 文字に切り詰められます。

関連項目: イベントタグ付けのためのlow-level discovery macros でのマクロ関数の使用。

イベントタグの表示　タグは、定義されている場合、次の場所で新しいイベントとともに表示できます。

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Event details*
- *Dashboards* > *Problems* widget

表示されるタグの順序と数は、*Monitoring* > *Problems* または *Problems* ダッシュボードウィジェットの *Tag display priority* および *Show tags* フィルタリングオプションによって決まります。最大 3 つのタグまで表示できます。タグがそれ以上ある場合は、3 つのドットにカーソルを合わせると、ポップアップウィンドウにすべてのタグが表示されます。



イベント相関

概要

イベント相関を使用すると、障害イベントとその解決を非常に正確かつ柔軟に関連付けることができます。

イベント相関は、次のレベルで定義できます。

- トリガーレベル - 1 つのトリガーを使用して、個別の障害とその解決を関連付けることができます。
- グローバル - グローバル相関ルールを使用して、異なるトリガーまたはポーリング方法による障害とその解決を関連付けることができます。

トリガーベースのイベント相関

概要

トリガーベースのイベント相関を使用すると、1 つのトリガーによって報告された個別の障害を相関させることができます。

通常、1 つのトリガーによって作成されたすべての障害イベントは、OK イベントによってクローズできますが、より詳細なアプローチが必要になる場合があります。たとえば、ログファイルを監視する場合、ログファイル内の特定の障害を検出し、それらをすべて一括でクローズするのではなく、個別にクローズしたいことがあります。

これは、*PROBLEM event generation mode* パラメータが *Multiple* に設定されているトリガーの場合に該当します。このようなトリガーは通常、ログ監視やトラップ処理などに使用されます。

Zabbix では、タグ付けに基づいて障害イベントを関連付けることができます。タグを使用して値を抽出し、障害イベントを識別する情報を作成します。これを利用すると、一致するタグに基づいて障害を個別にクローズすることもできます。

つまり、同じトリガーで、イベントタグによって識別される個別のイベントを作成できます。そのため、障害イベントを 1 つずつ識別し、イベントタグによる識別情報に基づいて個別にクローズできます。

動作の仕組み

ログ監視では、次のような行に遭遇することがあります。

```
Line1: Service 1 stopped
Line2: Service 2 stopped
Line3: Service 1 was restarted
Line4: Service 2 was restarted
```

イベント相関の考え方は、Line1 の障害イベントを Line3 の復旧と、Line2 の障害イベントを Line4 の復旧と対応付け、それらの障害を 1 つずつクローズできるようにすることです。

```
Line1: Service 1 stopped
Line3: Service 1 was restarted #Line 1 の障害がクローズ

Line2: Service 2 stopped
Line4: Service 2 was restarted #Line 2 の障害がクローズ
```

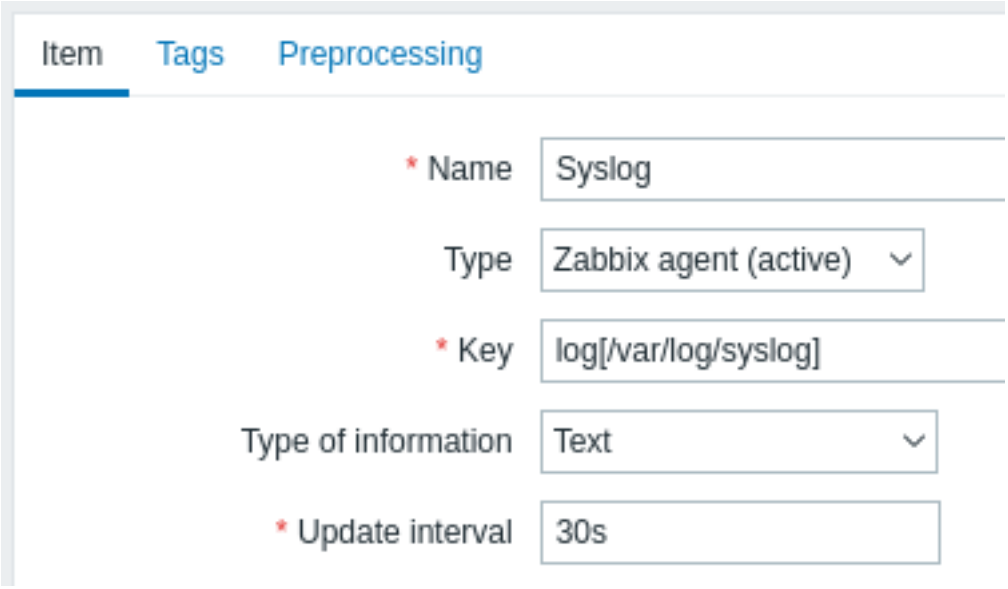
これを行うには、これらの関連するイベントに、たとえば”Service 1” や”Service 2” のようなタグを付ける必要があります。これは、ログ行に正規表現を適用してタグ値を抽出することで実現できます。すると、イベントが作成されたときに、それぞれ”Service 1” と”Service 2” のタグが付与され、障害を復旧と対応付けられるようになります。

設定

アイテム

まず、ログファイルを監視するアイテムを設定するとよいでしょう。たとえば、次のようにします。

```
log[/var/log/syslog]
```



アイテムを設定したら、設定変更が反映されるまで 1 分ほど待ち、その後最新データ に移動して、アイテムがデータの収集を開始していることを確認してください。

トリガー

アイテムが動作している状態で、トリガーを設定する必要があります。ログファイル内のどのエントリに注意を払うべきかを判断することが重要です。たとえば、次のトリガー式は'Stopping' のような文字列を検索し、潜在的な問題を示します。

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
"Stopping" という文字列を含む各行が障害と見なされるようにするには、トリガー設定の *Problem event generation mode* も'Multiple' に設定してください。

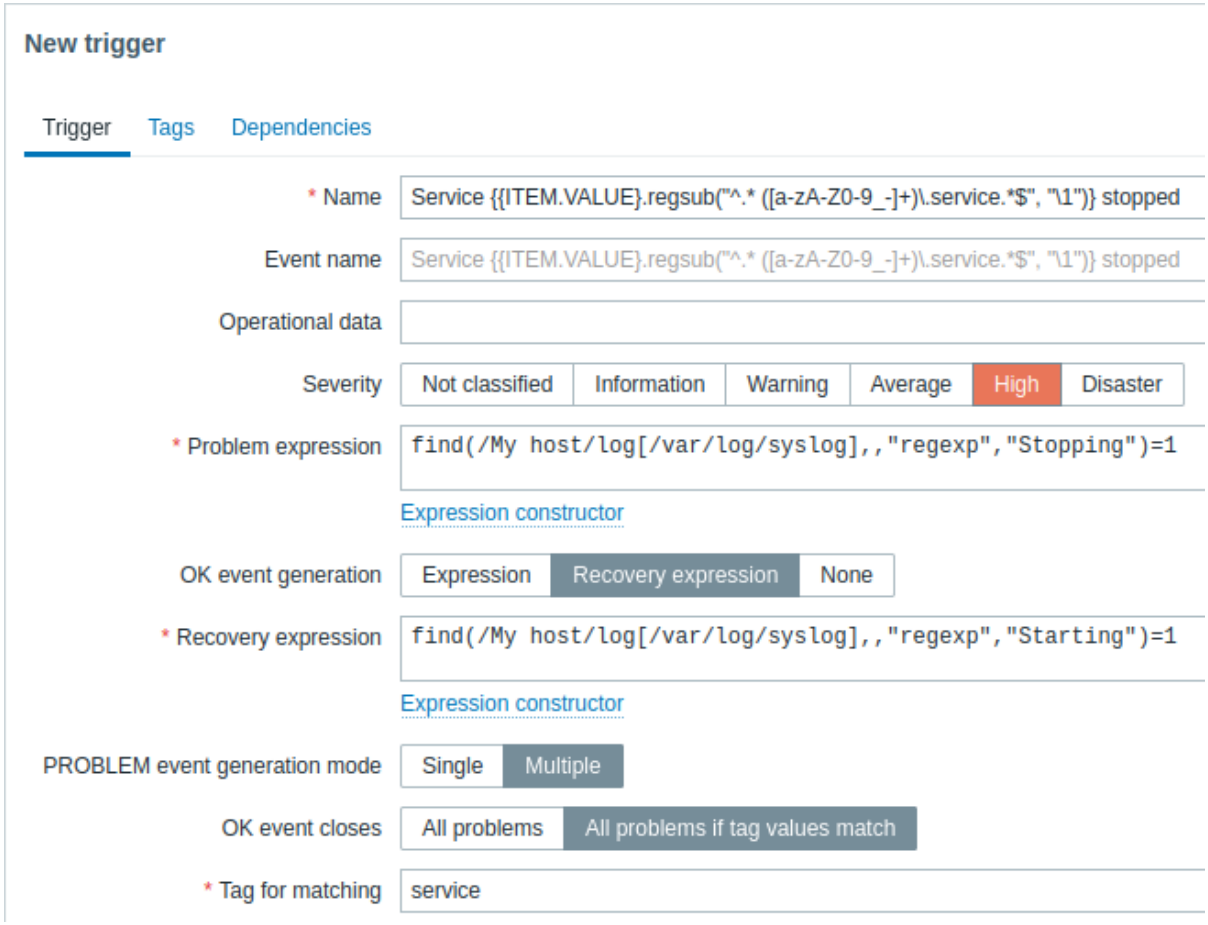
次に、復旧式を定義します。次の復旧式は、"Starting" という文字列を含むログ行が見つかった場合に、すべての障害を解決します。

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

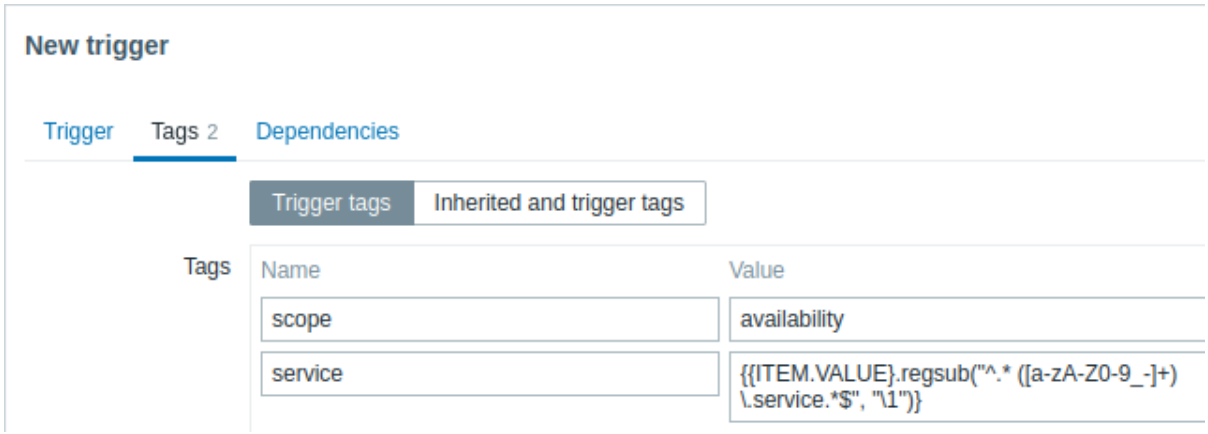
これを望まない場合は、対応する根本原因の障害がすべての障害ではなく、確実にクローズされるようにすることが重要です。そこでタグ付けが役立ちます。

トリガー設定でタグを指定することで、障害と復旧を一致させることができます。次の設定を行う必要があります。

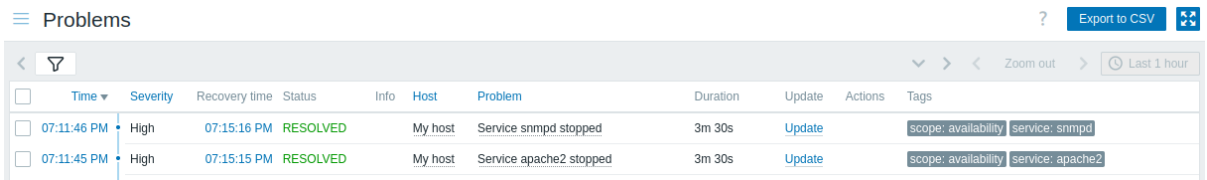
- *Problem event generation mode*: Multiple
- *OK event closes*: タグ値が一致する場合はすべての障害
- イベント照合用のタグ名を入力する



- ログ行からタグ値を抽出するようにタグを設定する



正しく設定できると、*Monitoring* > *Problems* で、アプリケーションごとにタグ付けされ、対応する復旧と照合された障害イベントを確認できるようになります。



Warning:
誤設定が起こり得るため、関連のない無関係な障害に対しても同様のイベントタグが作成される可能性がある場合は、以下のケースを確認してください。

- インデックス付きマクロは常にトリガー設定の *Expression* フィールドを参照し、*Recovery expression* は参照しません。たとえば、復旧イベントでは、**{ITEM.VALUE1}** は復旧時点における *problem expression* 内の最初のアイテムの最新値に解決されます。復旧式が別のアイテムに基づいており、復旧までの間に *problem expression* のアイテム値が変化すると、イベントのタグが異なり、相関されません。

- 2 つのアプリケーションが同じログファイルにエラーおよび復旧メッセージを書き込む場合、メッセージ形式が異なるときは {ITEM.VALUE} マクロからサービス A とサービス B の名前を抽出するために、タグ値に別々の正規表現を使って、同じトリガー内で 2 つの *service* タグを異なるタグ値で使用したいと考えるかもしれません。しかし、正規表現に一致しない場合は、これは想定どおりに動作しない可能性があります。一致しない正規表現は空のタグ値を生成し、障害イベントと OK イベントの両方で単一の空のタグ値があれば、それらは照合されてしまいます。そのため、サービス A の復旧メッセージが誤ってサービス B のエラーメッセージをクローズしてしまう可能性があります。
- 実際のタグとタグ値は、トリガーが発火したときにのみ表示されます。使用した正規表現が無効な場合は、黙って *UNKNOWN* 文字列に置き換えられます。最初の *UNKNOWN* タグ値を持つ障害イベントを見逃すと、同じ *UNKNOWN* タグ値を持つ後続の OK イベントが表示され、それらが本来クローズすべきでない障害イベントをクローズしてしまう可能性があります。
- ユーザーがタグ値としてマクロ関数なしで {ITEM.VALUE} マクロを使用する場合、255 文字の制限が適用されます。ログメッセージが長く、最初の 255 文字が特定性のない内容である場合、無関係な障害に対しても同様のイベントタグが生成される可能性があります。

グローバルイベント相関

概要

グローバルイベント相関を使用すると、Zabbix で監視しているすべてのメトリクスを対象に、イベントの相関関係を作成できます。

まったく異なるトリガーによって作成されたイベントを相関させ、それらすべてに同じ操作を適用できます。インテリジェントな相関ルールを作成することで、何千件もの繰り返し通知をなくし、障害の根本原因に集中できるようになります。

グローバルイベント相関は、1 つのトリガーに基づく障害と復旧のロジックから解放される強力な仕組みです。これまでは、1 つの障害イベントは 1 つのトリガーによって作成され、その障害の復旧も同じトリガーに依存していました。あるトリガーによって作成された障害を、別のトリガーで復旧させることはできませんでした。しかし、イベントタグに基づくイベント相関を使用すれば、それが可能になります。

例えば、ログトリガーでアプリケーションの障害を報告し、ポーリングトリガーでアプリケーションが稼働中であることを報告できます。イベントタグを利用して、ログトリガーに *status:down*、ポーリングトリガーに *status:up* というタグを付けることができます。その後、グローバル相関ルールでこれらのトリガーを関連付け、この相関に対して、古いイベントをクローズするなどの適切な操作を割り当てます。

別の用途として、グローバルイベント相関を使用して類似したトリガーを識別し、それらに同じ操作を適用できます。ネットワークポートの障害ごとに、障害レポートを 1 件だけ受け取れるとしたらどうでしょうか。すべてを報告する必要はありません。これもグローバルイベント相関で実現できます。

グローバルイベント相関は、相関ルールで設定します。相関ルールでは、新しい障害イベントを既存の障害イベントとどのように関連付けるか、また一致した場合に何を実行するか（新しいイベントをクローズする、対応する OK イベントを生成して一致した古いイベントをクローズする）を定義します。グローバル相関によって障害がクローズされた場合、そのことは 監視 > 障害の 情報列に表示されます。

グローバル相関ルールを設定できるのは、スーパー管理者レベルのユーザーのみです。

> **Attention:**
> イベント相関は、イベント処理のパフォーマンスに悪影響を及ぼしたり、設定を誤ると意図した数を超えるイベントをクローズしたりする可能性があるため、十分に注意して設定する必要があります（最悪の場合、すべての障害イベントがクローズされる可能性があります）。

グローバル相関を安全に設定するには、次の重要なヒントに従ってください。

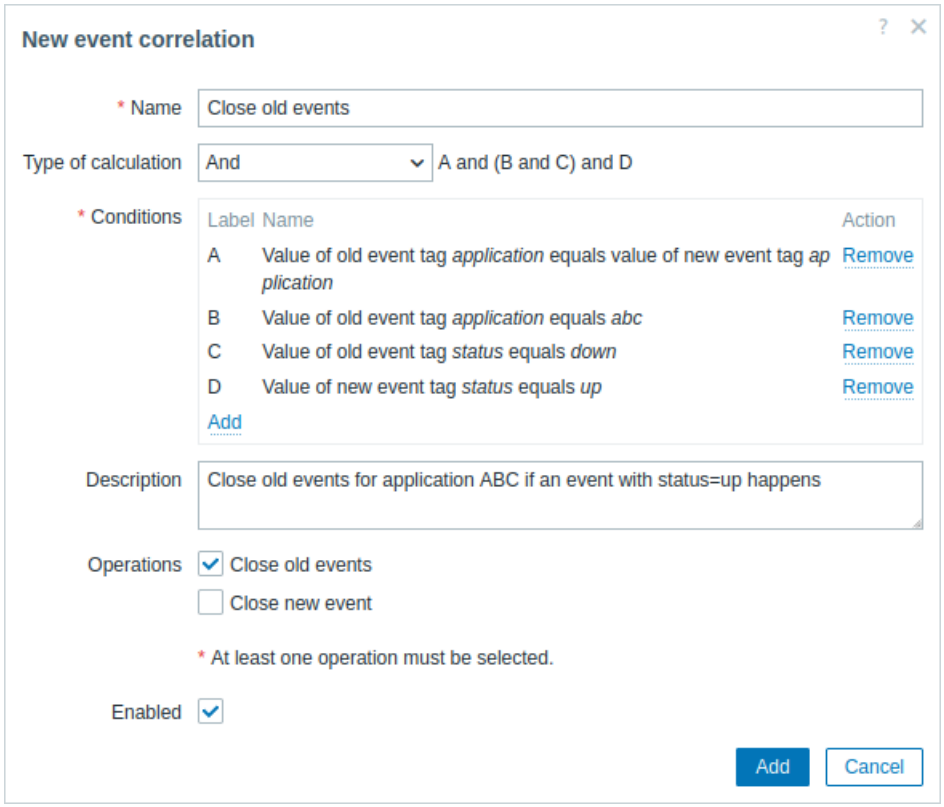
- 相関の範囲を絞り込む。新しいイベントに対して、古いイベントと関連付けるための一意のタグを必ず設定し、新しいイベントのタグ名相関条件を使用します。
- 古いイベントをクローズする場合は、明示的な古いイベント条件を追加する。古いイベントをクローズを選択する場合は、少なくとも 1 つの 古いイベント条件（例えば、古いイベントのタグ名、古いイベントのタグ値、または イベントタグのペア）を必ず追加してください。そうしないと、ルールが無関係な既存の障害イベントに一致してクローズする可能性があります（最悪の場合、すべての障害が対象になります）。実行時の値（host:port、セッション ID など）を照合する場合は、イベントタグのペアを使用し、可能な場合はホストまたはホストグループによってさらに一致範囲を絞り込むことを推奨します。
- 異なる相関設定で使用される可能性のある、一般的なタグ名の使用を避ける。
- 本当に必要な数に相関ルールを制限する。

関連項目: 既知の問題。

設定

イベント相関ルールをグローバルに設定するには:

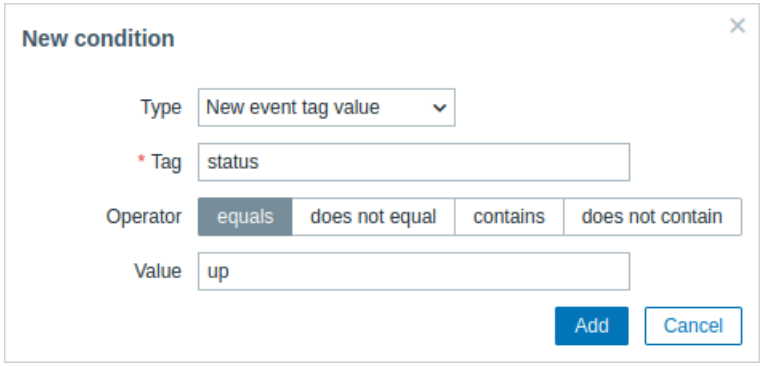
1. *Data collection* > *Event correlation* に移動します。
2. 右側の *Create event correlation* をクリックします（または、既存のルールを編集するには相関名をクリックします）。
3. フォームに相関ルールのパラメーターを入力します。



必須入力フィールドには赤いアスタリスクが付いています。

Parameter	Description
Name	一意の相関ルール名。
Type of calculation	条件の計算方法として、次のオプションを使用できます: **And** - すべての条件を満たす必要があります **Or** - 1 つの条件を満たせば十分です **And/Or** - 異なる条件タイプでは AND、同じ条件タイプでは OR **Custom expression** - アクション条件を評価するためのユーザー定義の計算式です。すべての条件（大文字の A、B、C、... で表されます）を含める必要があり、スペース、タブ、括弧 ( )、**and**（大文字小文字を区別）、**or**（大文字小文字を区別）、**not**（大文字小文字を区別）を含めることができます。
Conditions	条件の一覧です。条件の設定方法の詳細は以下を参照してください。
Description	相関ルールの説明。
Operations	イベントが相関されたときに実行する操作のチェックボックスをオンにします。使用できる操作は次のとおりです: **Close old events** - 新しいイベントが発生したときに古いイベントを閉じます。*Close old events* 操作を使用する場合は、必ず古いイベントに基づく条件を追加してください。追加しないと、既存の障害がすべて閉じられる可能性があります。 **Close new event** - 発生した新しいイベントを閉じます。 警告**!** *Close old events*/*Close new event* を使用する場合は、古い/新しいイベント条件を空欄のままにしないでください。*Close old events* 操作を、古いイベントに一致する条件を追加せずに選択すると、Zabbix は既存のすべての古いイベントに一致し、それらを閉じてしまう可能性があります。*Close old events* を使用する場合は、必ず明示的な古いイベント条件（たとえば、*Old event tag name* または *Event tag pair*）を追加してください。たとえば、*New event condition* のみと *Close old events* 操作を使用するルールでは、（欠落している）古いイベント条件に一致するすべての古いイベントが一致対象となり、結果として古い障害が閉じられます。
Enabled	このチェックボックスをオンにすると、相関ルールが有効になります。

新しい条件の詳細を設定するには、Conditions ブロックの Add をクリックします。ポップアップウィンドウが開き、条件の詳細を編集できます。



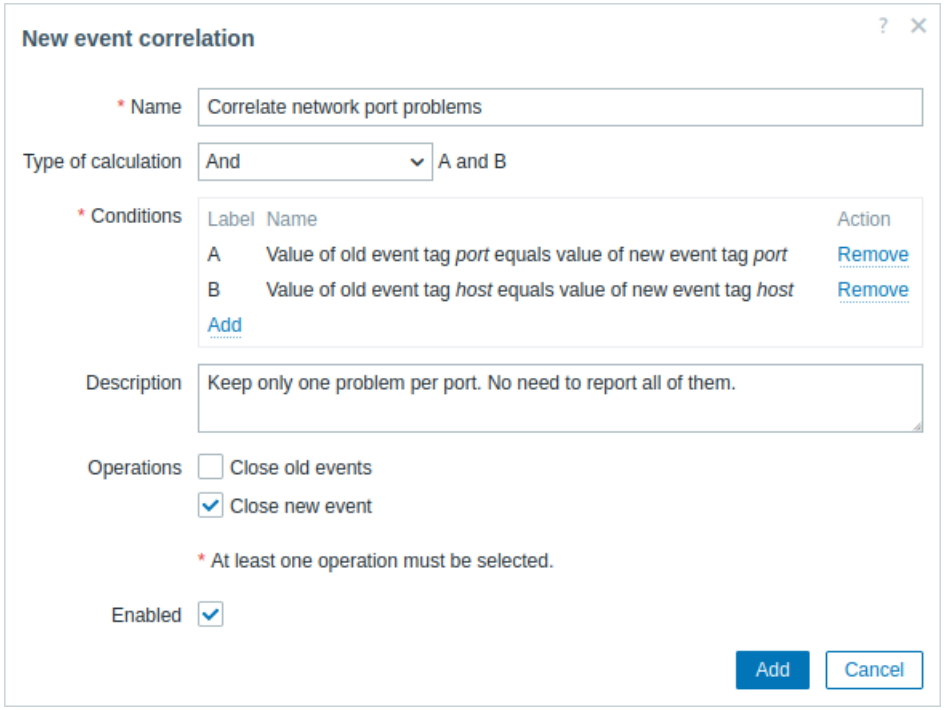
Parameter	Description
New condition	イベントを相関させる条件を選択します。 注意旧イベント条件が指定されていない場合、すべての旧イベントが一致して閉じられる可能性があります。同様に、新イベント条件が指定されていない場合、すべての新イベントが一致して閉じられる可能性があります。 使用できる条件は次のとおりです: **Old event tag name** - 一致させる古いイベントのタグ名を指定します。 **New event tag name** - 一致させる新しいイベントのタグ名を指定します。 **New event host group** - 一致させる新しいイベントのホストグループを指定します。 **Event tag pair** - 一致させる新しいイベントのタグ名と古いイベントのタグ名を指定します。この場合、両方のイベントのタグの値が一致すれば一致とみなされます。タグの名前は一致している必要はありません。 このオプションは、設定時点では不明な場合がある実行時の値を一致させるのに便利です（例 も参照してください）。 **Old event tag value** - 次の演算子を使用して、一致させる古いイベントのタグ名と値を指定します: *equals* - 古いイベントのタグ値と等しい *does not equal* - 古いイベントのタグ値と等しくない *contains* - 古いイベントのタグ値に文字列を含む *does not contain* - 古いイベントのタグ値に文字列を含まない **New event tag value** - 次の演算子を使用して、一致させる新しいイベントのタグ名と値を指定します: *equals* - 新しいイベントのタグ値と等しい *does not equal* - 新しいイベントのタグ値と等しくない *contains* - 新しいイベントのタグ値に文字列を含む *does not contain* - 新しいイベントのタグ値に文字列を含まない

Warning:
設定ミスが起こり得るため、無関係な障害に対して類似したイベントタグが作成される可能性がある場合は、以下のケースを確認してください。

- 実際のタグとタグ値は、トリガーが発火したときにのみ表示されます。使用した正規表現が無効な場合、それは静かに *UNKNOWN* 文字列に置き換えられます。最初の障害イベントで *UNKNOWN* タグ値を見逃すと、その後に同じ *UNKNOWN* タグ値を持つ OK イベントが発生し、本来閉じるべきではない障害イベントを閉じてしまう可能性があります。
- ユーザーがタグ値としてマクロ関数なしで {ITEM.VALUE} マクロを使用すると、255 文字の制限が適用されます。ログメッセージが長く、最初の 255 文字に特定性がない場合、無関係な障害に対しても類似したイベントタグが生成されることがあります。

例

同じネットワークポートからの繰り返しの障害イベントを停止します。



このグローバルな相関ルールは、トリガーに *host* と *port* のタグ値が存在し、元のイベントと新しいイベントでそれらが同じ場合に、障害を相関付けます。

この操作により、同じネットワークポート上の新しい障害イベントがクローズされ、元の障害のみがオープンのまま保持されます。

ディスカバリ

概要　ディスカバリは、手動で設定することなくホストを追加し、監視を構成するための自動的な方法を提供します。Zabbix は、さまざまなシナリオに対応するために複数のディスカバリ方法をサポートしています。

ディスカバリの方法

- ネットワークディスカバリ - ネットワーク上で利用可能なホストとサービスを自動的に検出します。
- アクティブエージェントの自動登録 - アクティブエージェントがサーバーに自動的に自身を登録できるようにします。
- ローレベルディスカバリ - プロトタイプとネストされたディスカバリルールを使用して、ファイルシステムやネットワークインターフェースなどの検出されたエンティティに対するアイテム、トリガー、グラフ、ホストを自動的に作成します。

ネットワークディスカバリ

概要

Zabbix は、効果的で非常に柔軟な自動ネットワーク検出機能を提供します。

ネットワーク検出を適切に設定すると、次のことが可能になります。

- Zabbix の導入を迅速化する。
- 管理を簡素化する。
- 過度な管理負荷をかけずに、変化の激しい環境で Zabbix を使用する。

Zabbix のネットワーク検出は、次の情報に基づいています。

- IP 範囲
- 外部サービスの可用性 (FTP、SSH、WEB、POP3、IMAP、TCP など)
- Zabbix エージェントから受信した情報 (暗号化されていないモードのみサポート)
- SNMP エージェントから受信した情報

これはネットワークトポロジーの検出は提供しません。

ネットワーク検出は基本的に、検出とアクションの 2 つのフェーズで構成されます。

ディスカバリ

Zabbix は、network discovery rules で定義された IP 範囲を定期的にスキャンします。
チェックの頻度は、各ルールごとに個別に設定できます。

各ルールには、その IP 範囲に対して実行するよう定義された一連のサービスチェックがあります。

ディスカバリルールは discovery manager によって処理されます。
discovery manager は、各ルールごとにタスク（ネットワークチェック）の一覧を持つジョブを作成します。
ネットワークチェックは、利用可能な discovery worker によって並列に実行されます（数は Web インターフェースで各ルールごとに設定可能です）。
例外は SNMPv3 チェックで、これは単一の worker によって処理されます。
同じ IP とポートを持つチェックのみが順次スケジュールされます。これは、一部のデバイスが同一ポートでの並列接続を受け付けないためです。

ネットワークチェックのキューサイズは、2000000、またはおよそ 4 GB のメモリに制限されています。
キューがいっぱいになると、ディスカバリルールはスキップされ、警告メッセージがログに出力されます。
キュー内のディスカバリチェック数を監視するには、内部アイテム `zabbix[discovery_queue]` を使用できます。

ディスカバリチェックは、他のチェックとは独立して処理されます。
いずれかのチェックでサービスが見つからない場合（または失敗した場合）でも、他のチェックは引き続き処理されます。

Note:
ディスカバリルールが実行中に変更された場合、現在のディスカバリ実行は中止されます。

ネットワークディスカバリモジュールによって実行される、サービスおよびホスト（IP）の各チェックは、ディスカバリイベントを生成します。

Event	Check of service result
Service Discovered	サービスが'down' の状態から'up' になった場合、または初めて検出された場合。
Service Up	サービスがすでに'up' の状態で、'up' である場合。
Service Lost	サービスが'up' の状態から'down' になった場合。
Service Down	サービスがすでに'down' の状態で、'down' である場合。
Host Discovered	ホストのすべてのサービスが'down' の状態だった後に、そのホストの少なくとも 1 つのサービスが'up' になる場合、または未登録のホストに属するサービスが検出された場合。
Host Up	少なくとも 1 つのサービスがすでに'up' である状態で、ホストの少なくとも 1 つのサービスが'up' になる場合。
Host Lost	少なくとも 1 つが'up' だった後に、ホストのすべてのサービスが'down' になる場合。
Host Down	ホストのすべてのサービスが、すでに'down' の状態で'down' である場合。

アクション

ディスカバリーイベントは、以下のような関連するアクションの基礎となります。

- 通知の送信
- ホストの追加/削除
- ホストの有効化/無効化
- ホストをグループに追加
- ホストをグループから削除
- ホストにタグを追加
- ホストからタグを削除
- テンプレートをホストにリンク/テンプレートをホストからリンク解除
- リモートスクリプトの実行

これらのアクションは、デバイスタイプ、IP、ステータス、稼働/停止時間などに応じて設定できます。ネットワークディスカバリーベースのイベントに対するアクションの設定の詳細については、アクションのオペレーションおよび条件のページを参照してください。

ネットワークディスカバリーアクションはイベントベースであるため、検出されたホストがオンラインの場合もオフラインの場合もトリガーされます。*Add host* のようなアクションが *Service Lost/Service Down* イベントでトリガーされるのを避けるために、アクション条件として *Discovery status: up* を追加することを強く推奨します。そうしないと、検出されたホストが手動で削除された場合でも *Service Lost/Service Down* イベントが生成され、次のディスカバリーサイクルで再作成されてしまいます。

Note:
検出されたホストにテンプレートをリンクする場合、リンク可能なテンプレートのいずれかに、ホストまたは他のリンク可能なテンプレートに既に存在する一意のエンティティ（例：アイテムキー）と同じ一意のエンティティ（例：アイテムキー）がある場合、テンプレートのリンクはまとめて失敗します。

ホストの作成

ホストを追加操作が選択されている場合、ホストが追加されます。また、ホストを追加操作がなくても、ホストに対するアクションを実行する操作を選択している場合は、ホストが追加されます。そのような操作は次のとおりです。

- ホストを有効にする。
- ホストを無効にする。
- ホストをホストグループに追加する。
- テンプレートをホストにリンクする。

作成されたホストは *Discovered hosts* グループに追加されます（デフォルト。*Administration* > *General* > *Other* で設定可能）。ホストを別のグループに追加したい場合は、ホストグループから削除操作（”Discovered hosts” を指定）と、ホストグループに追加操作（別のホストグループを指定）を両方追加してください。ホストはホストグループに属している必要があるためです。

検出されたデバイスの IP アドレスは、検出元（Zabbix サーバー、Zabbix プロキシ、またはプロキシグループ）およびインターフェース種別とともに、システム内でホストを検索するための条件として使用されます。同じ IP アドレス、インターフェース種別、および検出元を持つホストがすでに存在する場合、そのホストが操作の対象になります。検出元が異なる場合、検出されたエンティティは別のホストとして扱われ、新しいホストが作成されることがあります。

検出されたホストの IP アドレスが変更された場合、またはインターフェースが削除された場合、次回の検出時に新しいホストが作成されます。

ホスト名の付け方

ホストを追加する際、ホスト名はリバース DNS ルックアップの結果、またはリバースルックアップが失敗した場合は IP アドレスとなります。ルックアップは、ディスカバリを実行している Zabbix サーバーまたは Zabbix プロキシから実行されます。プロキシでルックアップに失敗した場合、サーバーで再試行されることはありません。そのような名前のホストがすでに存在する場合、次のホストには名前の末尾に **_2** が付加され、さらに **_3** と続きます。

DNS/IP ルックアップを上書きし、ホスト名にアイテム値を使用することも可能です。例えば:

- ディスカバリ用の Zabbix エージェントアイテムを使用して、Zabbix エージェントが稼働している複数のサーバーをディスカバリし、このアイテムが返す文字列値に基づいて自動的に適切な名前を割り当てることができます
- ディスカバリ用の SNMP エージェントアイテムを使用して、複数の SNMP ネットワークデバイスをディスカバリし、このアイテムが返す文字列値に基づいて自動的に適切な名前を割り当てることができます

ホスト名がアイテム値で設定されている場合、以降のディスカバリチェックでは更新されません。アイテム値でホスト名を設定できない場合は、デフォルト値 (DNS 名) が使用されます。

ディスカバリされた IP アドレスで既にホストが存在し、ディスカバリ元 (Zabbix サーバー、プロキシまたはプロキシグループ) が変更されていない場合、新しいホストは作成されません。ディスカバリ元が異なる場合、ディスカバリされたエンティティは別物として扱われ、新しいホストが作成される場合があります。ただし、ディスカバリアクションに操作 (テンプレートのリンク、ホストグループへの追加など) が含まれている場合は、IP アドレス、インターフェイスタイプ、ディスカバリ元で一致する既存のホストに対して実行されます。

ホストの削除

ネットワークディスカバリールールによって検出されたホストは、検出されたエンティティがルールの IP 範囲に存在しなくなった場合、監視データ > ディスカバリから自動的に削除されます。ホストは即座に削除されます。

ホスト追加時のインターフェース作成

ネットワークディスカバリの結果としてホストが追加されると、以下のルールに従ってインターフェースが作成されます。

- 検出されたサービス - たとえば、SNMP チェックが成功した場合は、SNMP インターフェースが作成されます。
- ホストが Zabbix エージェントと SNMP の両方のリクエストに応答した場合、両方の種類のインターフェースが作成されます。
- 一意性の条件が Zabbix エージェントまたは SNMP から返されたデータである場合、ホストに対して最初に見つかったインターフェースがデフォルトのものとして作成されます。デフォルトのインターフェースが停止しない限り、他の IP アドレスは無視されます。デフォルトのインターフェースが停止している場合は、他の IP アドレスのうち最も小さいものがデフォルトインターフェースの代替として使用されます。アクションの条件（Host IP など）は、インターフェースの追加には影響しません。注この動作は、すべてのインターフェースが同じディスカバリルールによって検出される場合に機能します。別のディスカバリルールが同じホストの別のインターフェースを検出した場合、追加のホストが追加されます。
- ホストがエージェントチェックのみに応答した場合、エージェントインターフェースのみを持つホストとして作成されます。後で SNMP に応答するようになった場合は、追加の SNMP インターフェースが追加されます。
- 3 つの個別のホストが最初に”IP” 一意性条件で検出されて作成され、その後ディスカバリルールが変更されて、ホスト A、B、C の一意性条件の結果が同一になった場合、B と C は最初のホスト A の追加インターフェースとして作成されます。個別のホスト B と C はそのまま残ります。*Monitoring* > *Discovery* では、追加されたインターフェースは”Discovered device” 列に黒字でインデント付きで表示されますが、”Monitored host” 列には最初に作成されたホスト A のみが表示されます。”Uptime/Downtime” は、追加インターフェースと見なされる IP については測定されません。

プロキシ設定の変更

異なるプロキシによって検出されたホストは、常に別々のホストとして扱われるわけではありません。
検出と一意性チェックは、プロキシグループの構造に依存します。プロキシが検出ルールを実行してホストを作成すると、そのホストはプロキシ自身ではなく、プロキシの親プロキシグループに追加されます。
Zabbix が検出中に IP の一意性を評価する際は、親プロキシグループによって監視されているホストを確認します。
そのグループ内の個々のプロキシによって監視されているホスト（検出を実行したプロキシを含む）は、一意性チェックでは無視されます。そのため、複数のプロキシが重複するサブネットを監視している場合、重複したホストが作成される可能性があります。

この動作により、異なるサブネットで使用される重複する IP 範囲全体で検出を動作させることができますが、すでに監視されているサブネットに割り当てられたプロキシを変更するのはより複雑です。これは、重複を避けるために、プロキシの変更を検出されたホストと親プロキシグループのメンバーシップの両方に対して一貫して適用する必要があるためです。

たとえば、検出ルールでプロキシを置き換える手順は次のとおりです。

1. 検出ルールを無効にします。
2. プロキシ設定を同期します。
3. 検出ルール内のプロキシを置き換えます。
4. このルールで検出されたすべてのホストのプロキシを置き換えます（重複を避けるため、親プロキシグループ内のホストと、そのグループ内の個々のプロキシによって監視されているホストの両方が更新されていることを確認します）。
5. 検出ルールを有効にします。

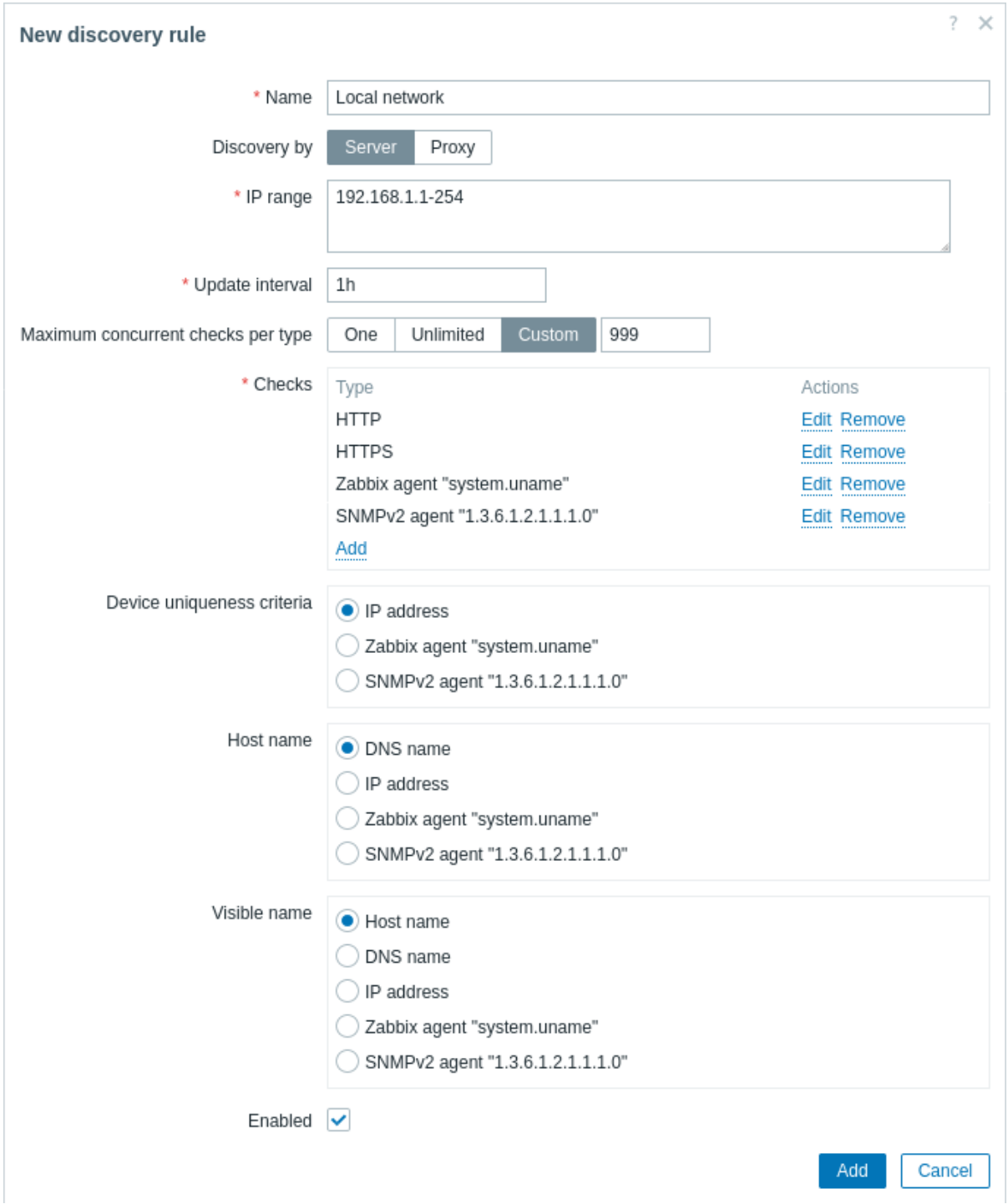
ネットワークディスカバリルールの設定

概要

Zabbix がホストとサービスを検出するために使用するネットワークディスカバリルールを設定するには、次の手順を実行します。

1. *Data collection* > *Discovery* に移動します。
2. *Create discovery rule* をクリックします（または、既存のルールを編集するにはルール名をクリックします）。
3. ディスカバリルールの属性を編集します。

ルール属性



必須の入力フィールドには赤いアスタリスクが付いています。

Parameter	Description
Name	ルールの一意な名前です。たとえば、「Local network」。
Discovery by	検出は次のいずれかによって実行されます: **Server** - Zabbix サーバーによる **Proxy** - Zabbix プロキシによる（proxy name フィールドで選択）
IP range	検出対象の IP アドレス範囲です。次の形式を指定できます: 単一 IP: 192.168.1.33 IP アドレスの範囲: 192.168.1-10.1-255. 範囲は、対象となるアドレスの総数（64K 未満）によって制限されます。 IP マスク: 192.168.4.0/24 サポートされる IP マスク: /16 - /30 for IPv4 addresses /112 - /128 for IPv6 addresses リスト: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 このフィールドは、スペース、タブ、および複数行に対応しています。

Parameter	Description
Update interval	このパラメーターは、Zabbix がルールをどのくらいの頻度で実行するかを定義します。 間隔は、前回の検出インスタンスの実行が終了した後に計測されるため、重複は発生しません。 Time suffixes がサポートされています。例: 30s, 1m, 2h, 1d。 User macros がサポートされています。 *Note* ユーザーマクロを使用し、その値が変更された場合（例: 1w → 1h）、次回のチェックは以前の値に従って実行されます（例の値ではかなり先になります）。
Maximum concurrent checks per type	検出チェックの並列処理のために、サービスチェックごとの検出スレッド（worker）の最大数を設定します: **One** - 1 スレッド **Unlimited** - 無制限のスレッド数（ただしStartDiscoverers パラメーターで指定された数を超えない） **Custom** - 任意のスレッド数を設定（0-999）。 SNMPv1 および SNMPv2 の非同期サービスチェックは、利用可能な worker によって並行して処理されます。SNMPv3 の非同期サービスチェックを持つすべての検出ルールは、libsnmp 実装の特性により 1 つの worker によって処理されます。そのため、worker 数を増やしても検出速度は向上しません。
Checks	Zabbix は検出にこのチェック一覧を使用します。新しいチェックをポップアップウィンドウで設定するには、Add をクリックします。 サポートされるチェック: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix エージェント, SNMPv1 エージェント, SNMPv2 エージェント, SNMPv3 エージェント, ICMP ping。 プロトコルベースの検出では、SNMP を除き、各ホストをテストするために **net.tcp.service[]** 機能を使用します。SNMP は SNMP OID を問い合わせます。Zabbix エージェントは、暗号化されていないモードでアイテムを問い合わせることでテストされます。詳細はagent items を参照してください。 「Ports」パラメーターには、次のいずれかを指定できます: 単一ポート: 22 ポート範囲: 22-45 リスト: 22-45,55,60-70 Zabbix 7.0 以降、LDAP チェックを除くすべてのサービスチェックは非同期で実行されます。 Zabbix 7.0 以降、HTTP/HTTPS のチェックは libcurl 経由で行われます。Zabbix サーバー/プロキシが libcurl なしでコンパイルされている場合、HTTP チェックは以前のバージョンと同様に動作します（つまり TCP チェックとして動作します）が、HTTPS チェックは動作しません。
Device uniqueness criteria	一意性の基準は次のいずれかです: **IP address** - 複数の単一 IP デバイスは処理しません。同じ IP を持つデバイスがすでに存在する場合、それはすでに検出済みと見なされ、新しいホストは追加されません。 **<discovery check>** - Zabbix エージェントまたは SNMP エージェントのチェックのいずれか。 検出時に使用される一意性の基準は、アクション実行時のシステム内でのホスト識別とは同じではないことに注意してください。検出時の一意性の基準は、2 つ以上の検出されたデバイスが同一かどうか（または異なるかどうか）を定義します。一方、Zabbix におけるホスト識別の基準は IP アドレスのみです（Host creation を参照）。
Host name	作成されるホストの技術名を次のいずれかで設定します: **DNS name** - DNS 名（デフォルト） **IP address** - IP アドレス **<discovery check>** - 検出チェックで受信した文字列値（例: Zabbix エージェント、SNMP エージェントのチェック） 関連項目: Host naming。
Visible name	作成されるホストの表示名を次のいずれかで設定します: **Host name** - 技術名（デフォルト） **DNS name** - DNS 名 **IP address** - IP アドレス **<discovery check>** - 検出チェックで受信した文字列値（例: Zabbix エージェント、SNMP エージェントのチェック） 関連項目: Host naming。
Enabled	チェックボックスをオンにすると、ルールは有効になり、Zabbix サーバーによって実行されます。 オフの場合、ルールは有効ではなく、実行されません。

ファイルディスクリプタ上限の超過

同時チェック数が多い場合、discovery manager のファイルディスクリプタ上限を使い切る可能性があります。

検出に必要なファイルディスクリプタ数は、discovery worker の数 * 1000 に相当します。
デフォルトではdiscovery worker は 5 個ですが、システムのソフトリミットはおおよそ 1024 です。

この上限に近づくと、Zabbix は各 worker ごとのデフォルトの「種類ごとの同時チェック数」を減らし、ログファイルに警告を書き込みます。
ただし、ユーザーが *Maximum concurrent checks per type* に Zabbix が計算した値より大きい値を設定している場合、Zabbix は 1 つの worker に対してユーザー定義の値を使用します。

実際に使用するシナリオ

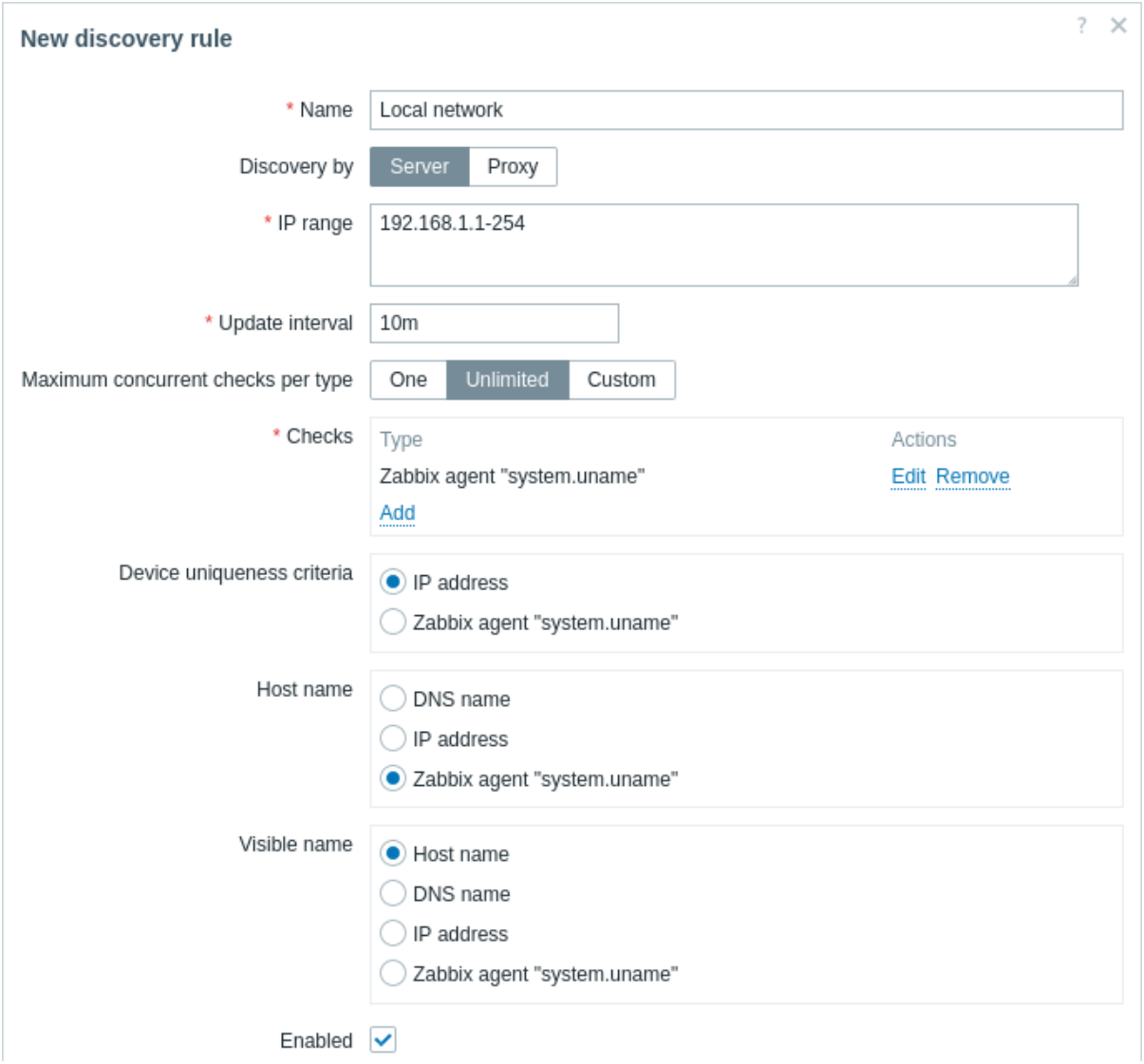
この例では、ローカルネットワークの 192.168.1.1～192.168.1.254 の範囲の IP アドレスをチェックするネットワークディスカバリを設定します。

このシナリオでは、以下のことをおこなう設定をします：

- Zabbix エージェントが動作しているホストを検出
- 検出は 10 分間隔で実行
- アップタイムが 1 時間以上のホストは監視対象に追加
- ダウンタイムが 24 時間以上のホストは監視対象から削除
- Linux ホストはグループ「Linux servers」に追加
- Windows ホストはグループ「Windows servers」に追加
- Linux ホストではテンプレート *Linux* を使用
- Windows ホストではテンプレート *Windows* を使用

ステップ 1

IP 範囲のネットワークディスカバリールールを定義します。



Zabbix は、192.168.1.1-192.168.1.254 の IP 範囲で Zabbix エージェントに接続し、**system.uname** キーの値を取得することでホストを検出しようとします。エージェントから受信した値は、ホストの名前付けや、異なるオペレーティングシステムごとに異なるアクションを適用するために使用できます。たとえば、Windows サーバーにはテンプレート *Windows* を、Linux サーバーにはテン

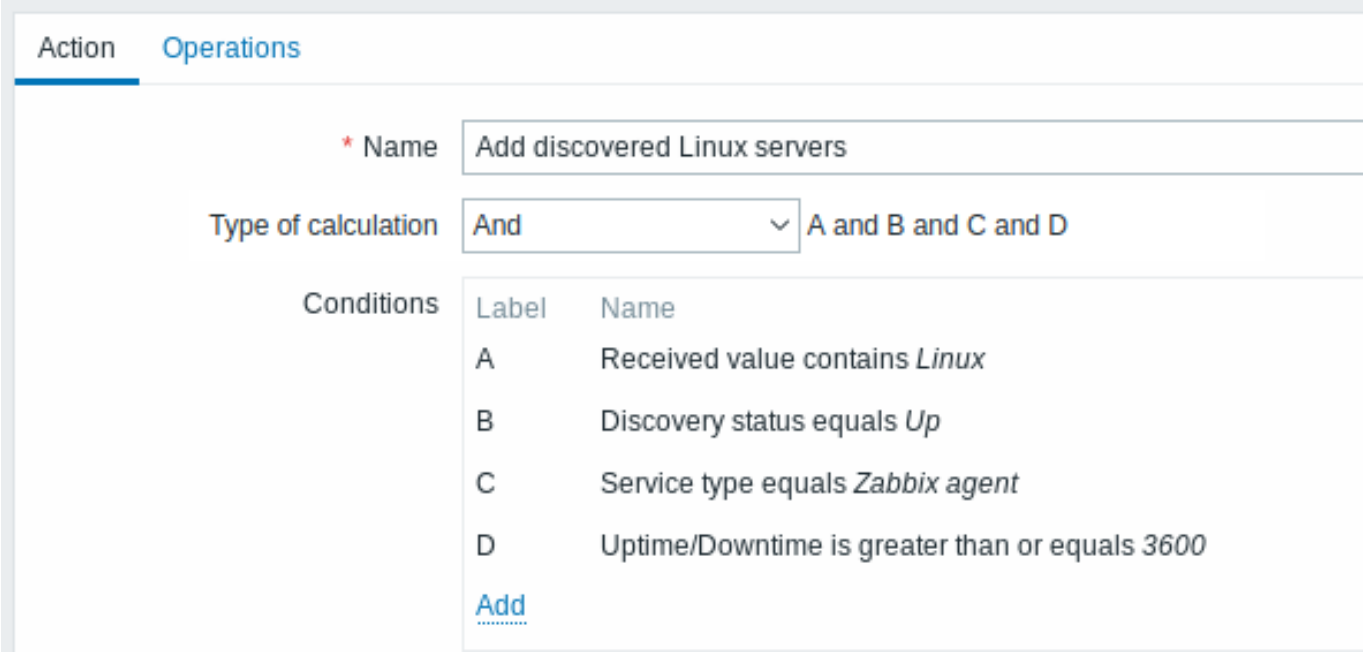
プレート *Linux* をリンクします。

このルールは 10 分ごとに実行されます。

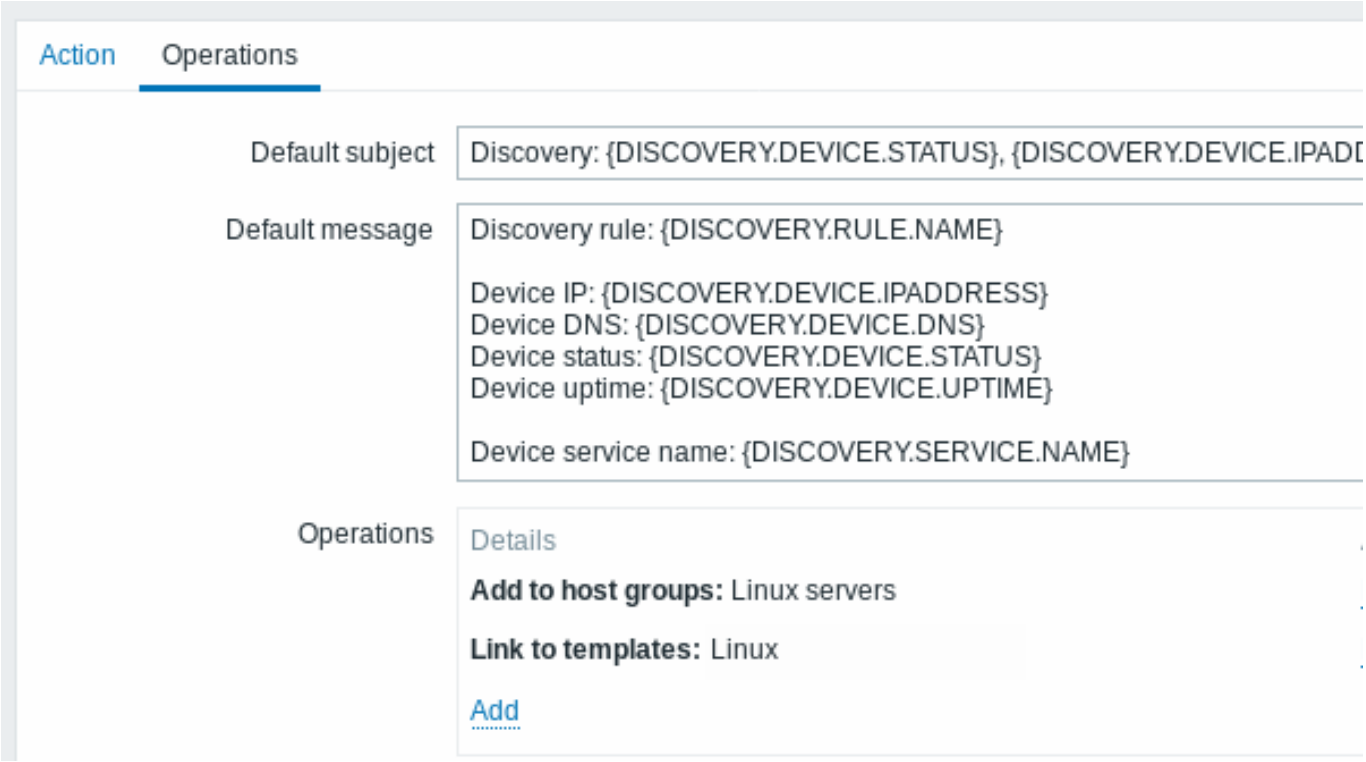
このルールが追加されると、Zabbix は自動的にディスカバリーを開始し、さらなる処理のためにディスカバリーベースのイベントを生成します。

ステップ 2

新たに検出された Linux サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。



このアクションは以下のコンディションが成立する場合有効になります。- サービス”Zabbix agent” が” 動作” している。- system.uname （ルールを定義する際に使用した Zabbix エージェントのキー）の値に”Linux” という文字列が含まれる。- アップタイムが 1 時間（3600 秒）を超えている。

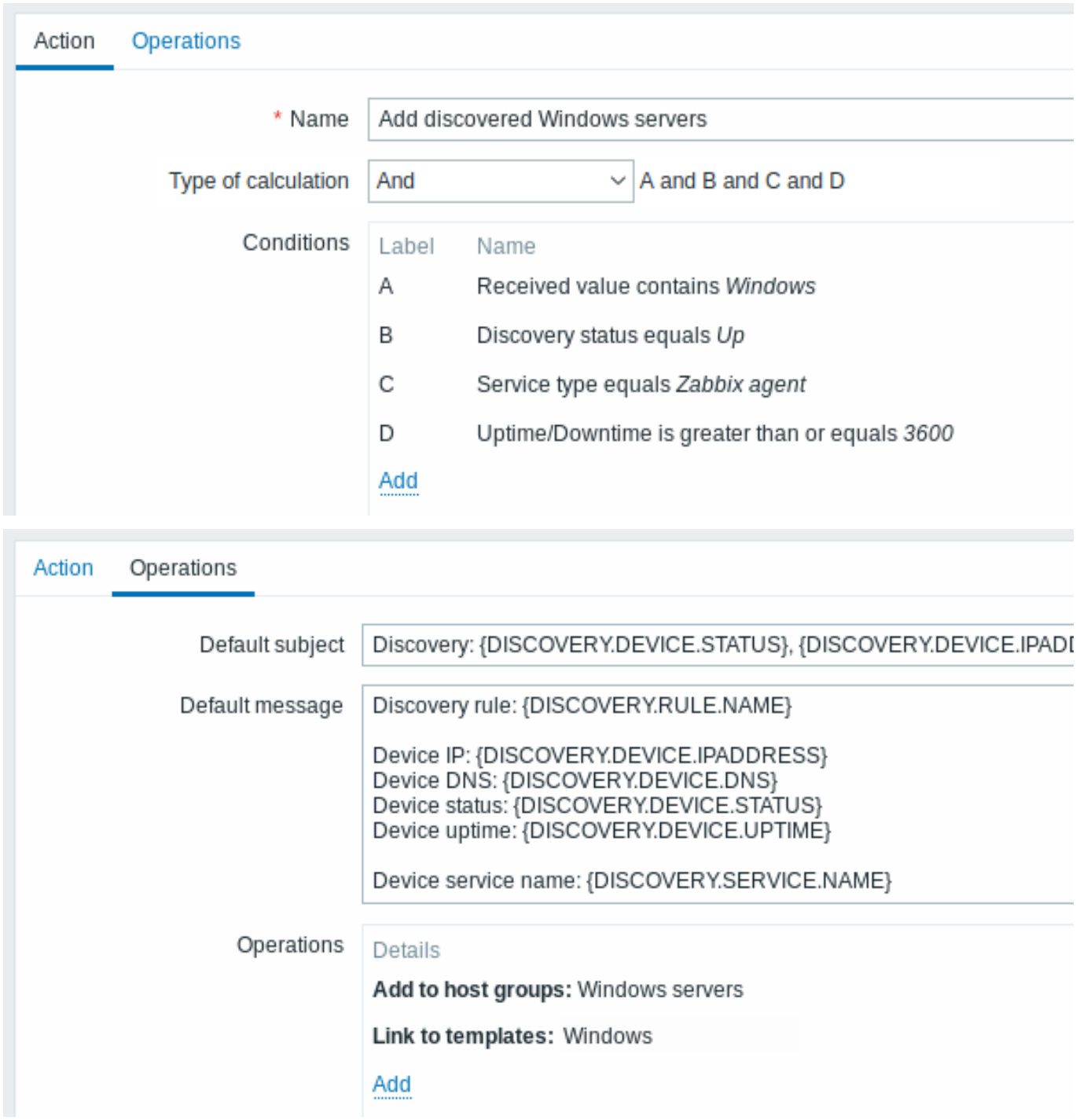


アクションによって以下のオペレーションが実行されます。

- 新たに検出したホストをグループ”Linux servers” に追加する。（まだ追加されていないホストの場合はホストの追加も実行する）
- ホストをテンプレート”Linux” にリンクする。”Linux” テンプレートに含まれるアイテムとトリガーを使用してホストの監視が自動的に開始される。

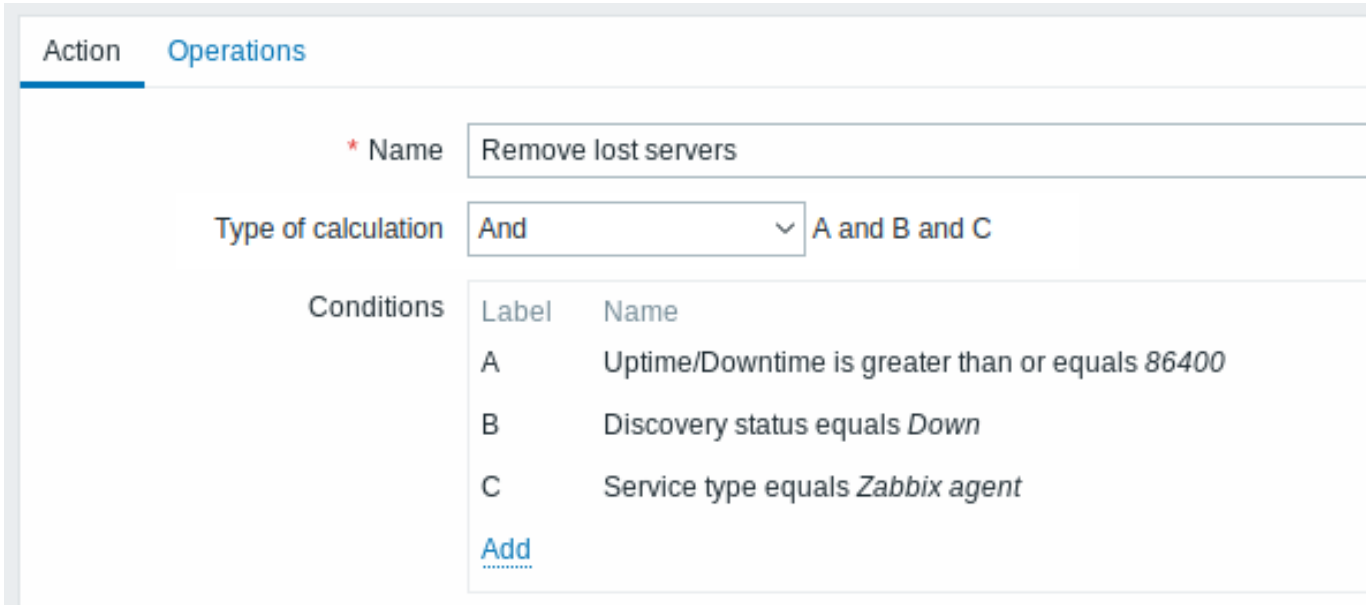
ステップ 3

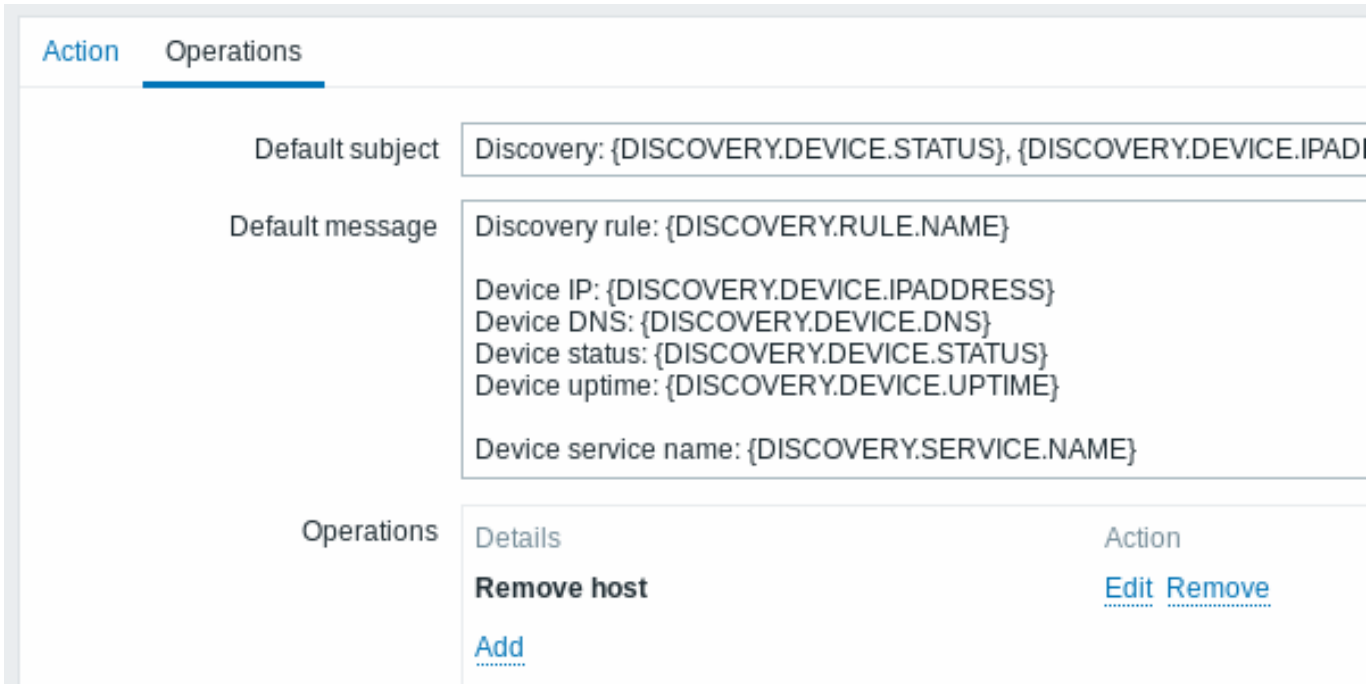
新たに検出された Windows サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。



ステップ 4

動作していないサーバーを削除するための検出アクションを定義します。





”Zabbix agent” サービスが 24 時間（86400 秒）以上 down している場合、サーバーは削除されます。A server will be removed if ”Zabbix agent” service is ’down’ for more than 24 hours (86400 seconds).

アクティブエージェントの自動登録

概要

アクティブ Zabbix エージェントの自動登録を許可することができ、その後サーバーはそれらの監視を開始できます。これにより、サーバー上で手動設定を行わずに、新しいホストを監視対象として追加できます。

自動登録は、これまで未知だったアクティブエージェントがチェックを要求したときに発生します。

この機能は、新しいクラウドノードを自動監視する場合に非常に便利です。Zabbix Cloud に新しいノードが追加されると、すぐにそのホストのパフォーマンスおよび可用性データの収集が自動的に開始されます。

アクティブエージェントの自動登録は、パッシブチェックを使用する追加済みホストの監視もサポートします。アクティブエージェントがチェックを要求する際、設定ファイルに`ListenIP` または`ListenPort` の設定パラメータが定義されていれば、それらもサーバーに送信されます。複数の IP アドレスが指定されている場合は、最初の 1 つがサーバーに送信されます。

サーバーは、新しく自動登録されたホストを追加する際に、受信した IP アドレスとポートを使用してエージェントを設定します。IP アドレスの値が受信されなかった場合は、受信接続に使用されたものが使われます。ポートの値が受信されなかった場合は、10050 が使用されます。

デフォルトのエージェントインターフェースとしてDNS 名 を使用してホストを自動登録するよう指定することもできます。

自動登録は次の場合に再実行されます。

- ホストのメタデータ 情報が変更された場合:
  - HostMetadata が変更され、エージェントが再起動した場合
  - HostMetadataItem が返す値が変更された場合
- メタデータがない手動作成ホストの場合
- ホストが手動で別の Zabbix プロキシによる監視に変更された場合
- 同じホストに対する自動登録が新しい Zabbix プロキシから来た場合

Zabbix サーバーおよび Zabbix プロキシにおけるアクティブエージェントの自動登録ハートビートは 120 秒です。そのため、検出されたホストが削除された場合、自動登録は 120 秒後に再実行されます。

設定

サーバーを指定する

エージェント設定ファイル（zabbix_agentd.conf）で、Zabbix サーバーが指定されていることを確認します。

```
ServerActive=10.0.0.1
```

zabbix_agentd.conf で *Hostname* を明示的に定義しない限り、サーバーはエージェントが配置されているシステムのホスト名を使用してホスト名を設定します。
Linux のシステムホスト名は、hostname コマンドを実行して取得できます。

Zabbix エージェントの設定で *Hostname* をカンマ区切りのホスト一覧として定義すると、一覧に含まれるすべてのホスト名に対してホストが作成されます。

設定ファイルを変更した後は、エージェントを再起動してください。

アクティブエージェントの自動登録用アクション

サーバーはエージェントから自動登録リクエストを受信すると、アクションを実行します。エージェントの自動登録には、イベントソースが「自動登録」のアクションを設定する必要があります。

> **Note:**
> アクティブエージェントを自動登録するために、ネットワークディスカバリを設定する必要はありません。

Zabbix の Web インターフェースで、次の手順を実行します。

1. *Alerts* > *Actions* に移動します。
2. *Autoregistration actions* を選択します。
3. *Create action* をクリックします。
4. *Action* タブで、アクション名を入力します。
5. 必要に応じて、条件を指定します。
   ホスト名またはホストメタデータの条件では、部分文字列一致または正規表現一致を使用できます。
   Host metadata 条件タイプを使用する場合は、ホストメタデータの使用を参照してください。
6. *Operations* タブで、*Add host*、*Add to host group*（例: *Discovered hosts*）、*Link templates* など、関連するオペレーションを追加します。

> **Note:**
> 自動登録されるホストがアクティブ監視のみをサポートする可能性が高い場合（Zabbix サーバーからファイアウォールで隔離されているホストなど）は、リンク先として *Template_Linux-active* のような専用テンプレートを作成するとよいでしょう。

作成されたホストは、デフォルトで *Discovered hosts* グループに追加されます（*Administration* > *General* > *Other* で設定可能）。ホストを別のグループに追加する場合は、*Remove from host group* オペレーション（「Discovered hosts」を指定）と、*Add to host group* オペレーション（別のホストグループを指定）の両方を追加してください。ホストは必ずホストグループに所属する必要があるためです。

セキュアな自動登録

暗号化接続で PSK ベースの認証を設定することにより、安全な自動登録が可能です。

暗号化レベルは、管理 > 一般 > 自動登録でグローバルに設定します。
暗号化なし、PSK 認証による TLS 暗号化、またはその両方（暗号化なしで登録できるホストと、暗号化接続で登録するホストを混在させる場合）を選択できます。

PSK による認証は、ホストを追加する前に Zabbix サーバーによって検証されます。
認証に成功するとホストが追加され、ホストとの接続は、グローバルの自動登録設定と同じアイデンティティ/事前共有鍵を使用する「PSK」のみに設定されます。

Attention:
プロキシを使用する環境で自動登録のセキュリティを確保するには、Zabbix サーバーとプロキシ間の暗号化を有効にしてください。

DNS をデフォルトインターフェースとして使用する

HostInterface および HostInterfaceItem 設定パラメータを使用すると、自動登録時のホストインターフェースにカスタム値を指定できます。

具体的には、ホストを IP アドレスではなく、DNS 名をデフォルトのエージェントインターフェースとして自動登録する場合に便利です。この場合、DNS 名を HostInterface または HostInterfaceItem パラメータの値として指定または返す必要があります。これらのパラメータのいずれかの値が、たとえば IP アドレスから DNS 名、またはその逆に変更されると、自動登録されたホストのデフォルトインターフェースがそれに応じて更新されます。この更新は新しいホストを作成するのではなく、既存のホストに適用されます。新しい値を送信するには、エージェントを再起動して自動登録プロセスを再開始する必要があります。

Note:
HostInterface または HostInterfaceItem パラメータが設定されていない場合は、listen_dns パラメータが使用されます。この値は、エージェントの IP アドレスに対して逆引き DNS を実行することで決定されます。逆引き DNS が正しく設定されていない場合、または無効な名前を返す場合、インターフェース値が無効になるため、自動登録が正しく実行されないか、失敗する可能性があります。

ホストメタデータの使用

エージェントがサーバーに自動登録リクエストを送信するとき、ホスト名も送信します。場合によっては（Amazon クラウドノードなど）、Zabbix サーバーが検出されたホストを区別するために、ホスト名だけでは不十分なことがあります。ホストメタデータを使用すると、エージェントからサーバーにその他の情報を任意で送信できます。

ホストメタデータは、エージェント設定ファイル（`zabbix_agentd.conf`）で設定します。設定ファイルでホストメタデータを指定する方法は 2 つあります。

```
HostMetadata
HostMetadataItem
```

上記のリンク先で、各オプションの説明を参照してください。

HostMetadataItem パラメータは、最大 65535 個の UTF-8 コードポイントを返すことができます。これより長い値は切り詰められます。

MySQL では、返される値にマルチバイト文字が含まれている場合、実際の最大文字数が少なくなることに注意してください。たとえば、3 バイト文字のみを含む値は合計 21844 文字まで、4 バイト文字のみを含む値は 16383 文字までに制限されます。

Attention:
アクティブなエージェントがアクティブチェックを更新するためのリクエストをサーバーに送信するたびに、自動登録が試行されます。リクエスト間の間隔は、エージェントのRefreshActiveChecksパラメータで指定します。最初のリクエストは、エージェントの再起動直後に送信されます。

例

HostMetadata を使用した OS による自動登録

Zabbix サーバーによってホストを自動登録したいとします。ネットワーク上にはアクティブな Zabbix エージェント（上記の「設定」セクションを参照）が存在します。ネットワーク上には Windows ホストと Linux ホストがあり、Zabbix の Web インターフェースには「Linux by Zabbix agent」および「Windows by Zabbix agent」テンプレートが用意されています。したがって、ホスト登録時には、登録されるホストに適切な Linux/Windows テンプレートを適用したいと考えています。デフォルトでは、自動登録時にサーバーに送信されるのはホスト名のみであり、これだけでは十分でない場合があります。適切なテンプレートがホストに適用されるようにするには、ホストメタデータを使用する必要があります。

Web インターフェースの設定

最初に Web インターフェースを設定します。2 つのアクションを作成します。最初のアクション:

- 名前: Linux ホストの自動登録
- 条件: ホストのメタデータに *Linux* を含む
- 操作: テンプレートのリンク: Linux by Zabbix agent

Note:
この場合、「ホストの追加」操作は省略できます。テンプレートをホストにリンクするには、最初にホストを追加する必要があるため、サーバーが自動的にそれを行います。

2 つ目のアクション:

- 名前: Windows ホストの自動登録
- 条件: ホストのメタデータに *Windows* を含む
- 操作: テンプレートのリンク: Windows by Zabbix agent

エージェントの設定

次に、エージェントを設定する必要があります。エージェントの設定ファイルに次の行を追加します。

```
HostMetadataItem=system.uname
```

このようにして、エージェントが動作しているホストに応じて、ホストのメタデータに「Linux」または「Windows」が含まれることを確認できます。この場合のホストメタデータの例:

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-OPXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

設定ファイルを変更した後は、エージェントを再起動するのを忘れないでください。

HostMetadata を使用した自動登録とテンプレートの制御

ステップ ***1 - HostMetadata*** で登録を保護する

ホストメタデータを使用して、不要なホストの登録に対する基本的な保護を行います。

Web インターフェースの設定

Web インターフェースでアクションを作成し、推測しにくいシークレットコードを使用して不要なホストを拒否します:

- 名前: Autoregistration action Linux
- 条件:
  - 計算のタイプ: AND
  - 条件 (A): ホストのメタデータに//Linux//を含む
  - 条件 (B): ホストのメタデータに//21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//を含む
- オペレーション:
  - ユーザーにメッセージを送信: Admin すべてのメディア経由
  - ホストグループに追加: Linux servers
  - テンプレートをリンク: Linux by Zabbix agent

この方法だけでは、データが平文で送信されるため、強力な保護は提供されませんのでご注意ください。変更を即座に反映させるには、設定キャッシュのリロードが必要です。

エージェントの設定

エージェントの設定ファイルに次の行を追加します。

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

ここで”Linux” はプラットフォーム名であり、残りの文字列は推測しにくい秘密のテキストです。

設定ファイルを変更した後は、エージェントを再起動することを忘れないでください。

ステップ ***2 -*** 登録済みホストにテンプレートを追加する

すでに登録されているホストに追加のテンプレートを追加することができます。この場合、MySQL by Zabbix agent テンプレートは、HostMetadata に MySQL トークンが含まれているホストのみにリンクされます。

Web インターフェースの設定

Web インターフェースでアクションを更新します:

- 名前: Autoregistration action Linux
- 条件:
  - 計算タイプ: AND
  - 条件 (A): ホストのメタデータに Linux を含む
  - 条件 (B): ホストのメタデータに 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae を含む
  - 条件 (C): ホストのメタデータに MySQL を含む
- オペレーション:
  - ユーザーにメッセージを送信: Admin (すべてのメディア)

- ホストグループに追加: Linux servers
- テンプレートをリンク: Linux by Zabbix agent
- テンプレートをリンク: MySQL by Zabbix Agent

エージェントの設定

エージェントの設定ファイルの次の行を更新します。

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

設定ファイルを変更した後は、エージェントを再起動することを忘れないでください。

ローレベルディスカバリ

概要　Low-level discovery (LLD) は、ホスト上のエンティティ（例: ファイルシステム、ネットワークインターフェース）を自動的に検出し、各エンティティごとに手動設定なしで対応するアイテム、トリガー、グラフを作成します。

LLD を使用するには、エンティティを記述する JSON データを収集する **discovery rule** を作成し、各エンティティに対する **prototypes**（item、trigger、graph）を用意します。

rule によって返される JSON はオブジェクトの配列である必要があり、各オブジェクトはキーと値のペアを使用して 1 つの検出済みエンティティを表します。たとえば、`net.if.discovery` を持つ discovery rule は次のような値を返す場合があります。

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

その後、対応するアイテム、トリガー、グラフのセットが prototypes から作成されます。インターフェースごとに 1 つずつ、`lo` と `eth0` 用に作成されます。

オブジェクトに `{#MACRO}` 形式のキーが含まれている場合、それらのマクロは prototypes で直接使用されます。含まれていない場合、または追加の/カスタムのマクロが必要な場合は、マクロを手動で定義 し、JSONPath を使用して JSON フィールドにマッピングできます。

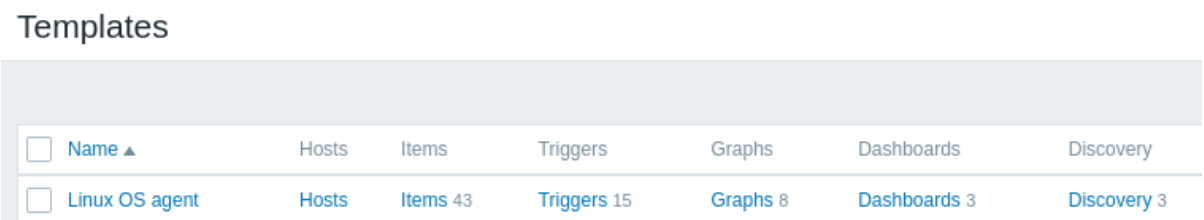
LLD はホスト も作成でき（たとえば、ハイパーバイザー上で検出された仮想マシン 向け）、複数レベルの検出のためにネストされた discovery rules もサポートします。

もはや検出されなくなったエンティティは、自動的に無効化または削除 できます。

低レベルディスカバリの設定　ここでは、ファイルシステムのディスカバリを例に、低レベルディスカバリについて説明します。

ディスカバリを設定するには、次の手順を実行します。

1. *Data collection* > *Templates* または *Hosts* に移動します。
2. 対応するテンプレート/ホストの行にある *Discovery* をクリックします。



3. 画面右上の *Create discovery rule* をクリックします。
4. 必要な詳細を入力して、ディスカバリルールのフォームを完成させます。

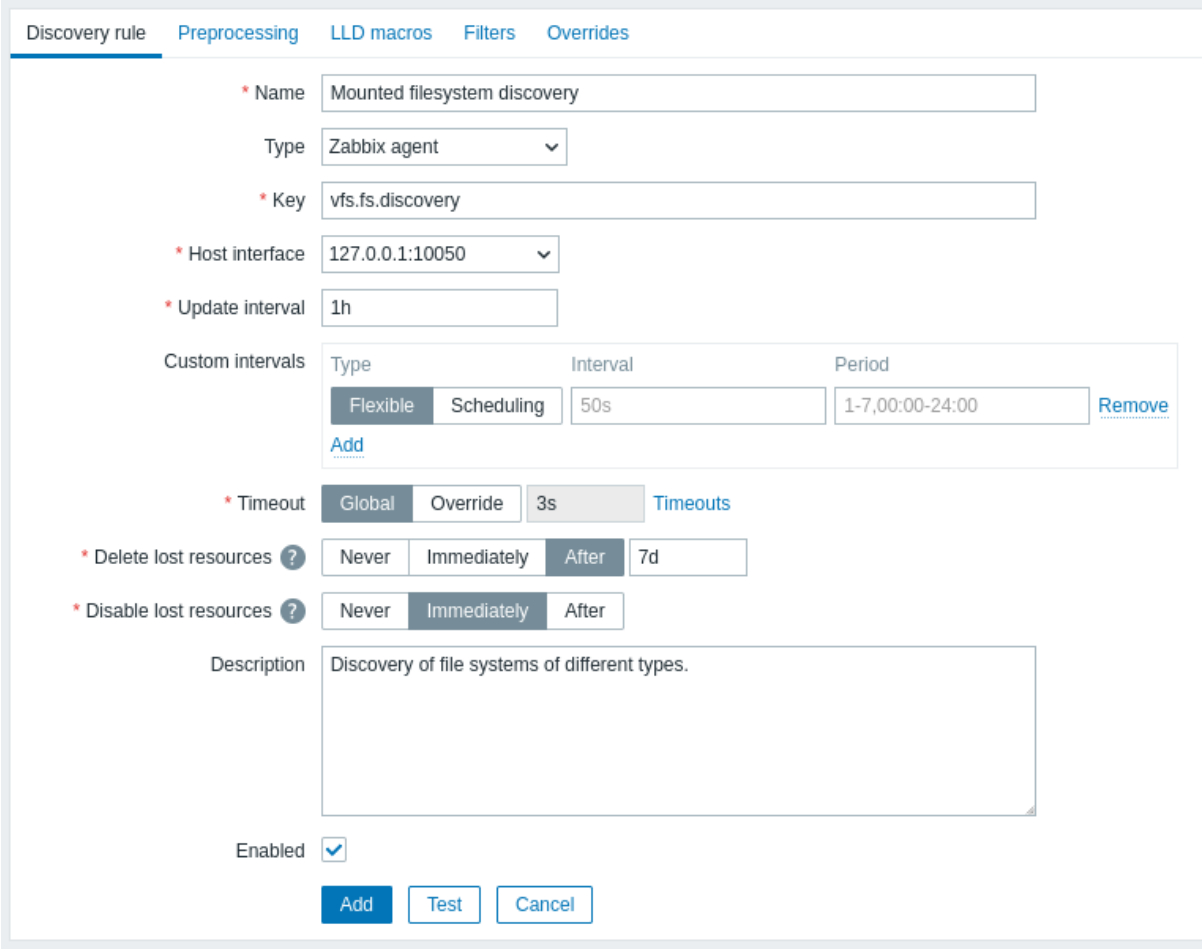
ディスカバリールール

ディスカバリールールフォームには 5 つのタブがあり、左から右に、ディスカバリー中のデータフローを表しています。

- ディスカバリールール - 最も重要な項目として、ディスカバリーデータを取得する組み込みアイテムまたはカスタムスクリプトを指定します。
- 前処理 - ディスカバリーされたデータに前処理を適用します。
- *LLD* マクロ - ディスカバリーされたアイテムやトリガーなどで使用するマクロ値を抽出できます。
- フィルター - ディスカバリーされた値をフィルタリングできます。

- オーバーライド - 特定のディスカバリーされたオブジェクトに適用する際に、アイテム、トリガー、グラフ、またはホストのプロトタイプを変更できます。

ディスカバリールールタブには、ディスカバリーに使用するアイテムキーと、いくつかの一般的なディスカバリールール属性が含まれています。



必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
名前	ディスカバリールールの名前。
タイプ	ディスカバリーを実行するチェックのタイプ。 この例では、*Zabbix* エージェントアイテムタイプを使用しています。 ディスカバリールールは、通常のアイテムに依存する依存アイテムにすることもできます。別のディスカバリールールに依存することはできません。依存アイテムの場合は、該当するタイプ（依存アイテム）を選択し、「マスターアイテム」フィールドでマスターアイテムを指定します。マスターアイテムは存在している必要があります。
キー	ディスカバリーアイテムのキーを入力します（最大 2048 文字）。 例えば、組み込みの「vfs.fs.discovery」アイテムキーを使用すると、コンピューター上に存在するファイルシステムの一覧、そのタイプ、マウントオプションを含む JSON 文字列を返すことができます。 ファイルシステムのディスカバリーには、「vfs.fs.get」エージェントキーによるディスカバリー結果を使用する方法もあることに注意してください（例を参照）。

パラメータ	説明
更新間隔	このフィールドでは、Zabbix がディスカバリーを実行する頻度を指定します。ファイルシステムのディスカバリーを設定したばかりの段階では短い間隔を設定するとよいでしょう。ただし、正常に動作することを確認した後は、ファイルシステムは通常それほど頻繁に変更されないため、30 分以上に設定できます。 時間のサフィックスがサポートされています（例: 30s、1m、2h、1d）。 ユーザーマクロがサポートされています。 注: カスタム間隔に 0 以外の値が存在する場合にのみ、更新間隔を「0」に設定できます。「0」に設定し、0 以外の値を持つカスタム間隔（フレキシブルまたはスケジュール）が存在する場合、アイテムはカスタム間隔の期間中にポーリングされます。 新しいディスカバリールールは、スケジューリングまたはフレキシブル更新間隔が設定され、更新間隔が 0 に設定されている場合を除き、作成から 60 秒以内にチェックされます。 既存のディスカバリールールでは、今すぐ実行ボタンを押すことで、ディスカバリーを直ちに実行できます。
カスタム間隔	アイテムをチェックするためのカスタムルールを作成できます。 フレキシブル - 更新間隔の例外（異なる頻度の間隔）を作成します。 スケジューリング - カスタムポーリングスケジュールを作成します。 詳細については、カスタム間隔を参照してください。
タイムアウト	ディスカバリーチェックのタイムアウトを設定します。タイムアウトオプションを選択します。 グローバル - プロキシまたはグローバルのタイムアウトを使用します（グレー表示されたタイムアウトフィールドに表示されます）。 上書き - カスタムタイムアウトを使用します（タイムアウトフィールドで設定します。許可される範囲: 1～600 秒）。時間のサフィックス（例: 30s、1m）とユーザーマクロがサポートされています。 タイムアウトリンクをクリックすると、プロキシのタイムアウト、またはプロキシを使用していない場合はグローバルのタイムアウトを設定できます。タイムアウトリンクは、管理者（スーパー）タイプのユーザーで、管理 > 一般設定または管理 > プロキシの Web インターフェースセクションへの権限を持つユーザーにのみ表示されます。
失われたリソースの削除	ディスカバリーのステータスが「検出されなくなった」に変わった後、ディスカバリーされたエンティティを削除するまでの期間を指定します。 なし - 削除しません。 即時 - 直ちに削除します。 次の期間の後 - 指定した期間の後に削除します。値は失われたリソースの無効化の値より大きくする必要があります。 時間のサフィックスがサポートされています（例: 2h、1d）。 ユーザーマクロがサポートされています。 注: 「即時」の使用は推奨されません。フィルターを誤って編集しただけで、すべての履歴データとともにエンティティが削除される可能性があるためです。 手動で無効にしたリソースは、ローレベルディスカバリーによって削除されないことに注意してください。
失われたリソースの無効化	ディスカバリーのステータスが「検出されなくなった」に変わった後、ディスカバリーされたエンティティを無効にするまでの期間を指定します。 なし - 無効にしません。 即時 - 直ちに無効にします。 次の期間の後 - 指定した期間の後に無効にします。値はディスカバリールールの更新間隔より大きくする必要があります。 自動的に無効化されたリソースは、ローレベルディスカバリーによって再検出されると再び有効になります。手動で無効にしたリソースは、再検出されても再び有効にはなりません。 失われたリソースの削除が「即時」に設定されている場合、このフィールドは表示されません。 時間のサフィックスがサポートされています（例: 2h、1d）。 ユーザーマクロがサポートされています。
説明	説明を入力します。
有効	チェックすると、ルールが処理されます。

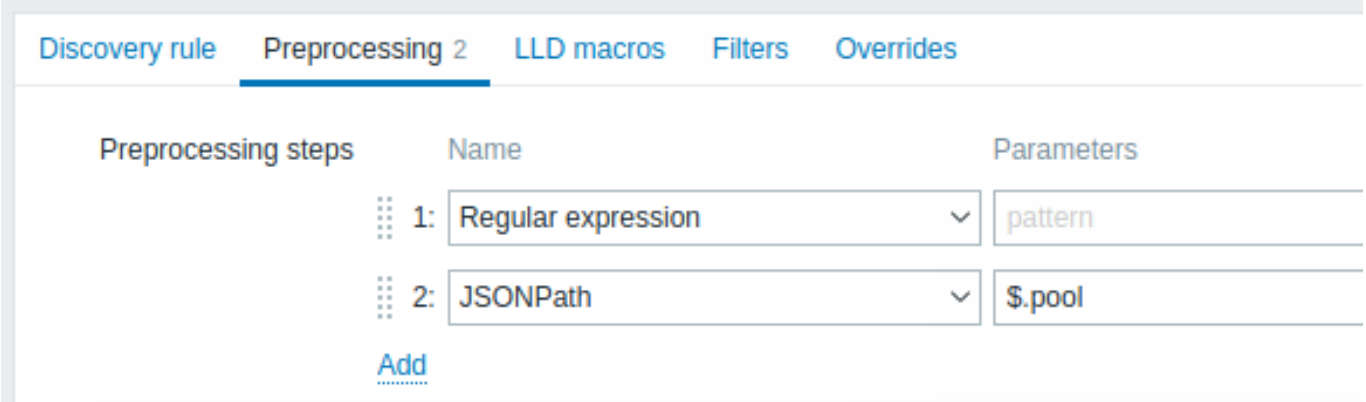
Note:
ディスカバリールールの履歴は保存されません。

前処理

前処理タブでは、ディスカバリの結果に適用する変換ルールを定義できます。このステップでは、1 つまたは複数の変換を設定できます。変換は、定義された順序で実行されます。すべての前処理は Zabbix サーバーによって実行されます。

関連項目:

- 前処理の詳細
- 前処理のテスト



タイプ	変換	説明
テキスト	正規表現	受信した値を <pattern> 正規表現に照合し、抽出した <output> で値を置換します。正規表現では、\N シーケンスを使用して最大 10 個のキャプチャグループを抽出できます。 パラメータ: **pattern** - 正規表現 **output** - 出力フォーマット用テンプレート。\N（N=1...9）エスケープシーケンスは、N 番目に一致したグループに置換されます。\0 エスケープシーケンスは、一致したテキストに置換されます。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	置換	検索文字列を検索し、別の文字列（または何も指定しない）に置換します。検索文字列のすべての出現箇所が置換されます。 パラメータ: **search string** - 検索および置換する文字列。大文字と小文字を区別します（必須） **replacement** - 検索文字列を置換する文字列。置換文字列を空にすることもでき、その場合、検索文字列が見つかると削除できます。 エスケープシーケンスを使用して、改行、キャリッジリターン、タブ、スペースを検索または置換できます（”\n \r \t \s”）。バックスラッシュは”\\” としてエスケープでき、エスケープシーケンスは”\\n” のようにエスケープできます。改行、キャリッジリターン、タブのエスケープは、ローレベルディスカバリ中に自動的に行われます。
構造化データ	*JSONPath*	JSONPath 機能を使用して、JSON データから値またはフラグメントを抽出します。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。

タイプ		
	XML XPath	XPath 機能を使用して、XML データから値またはフラグメントを抽出します。 このオプションを使用するには、Zabbix サーバーが libxml サポート付きでコンパイルされている必要があります。 例: `number(/document/item/value)` は、 `<document><item><value>10</value></item></document>` から 10 を抽出します。 `number(/document/item/@attribute)` は、`<document><item attribute="10"></item></document>` から 10 を抽出します。 `/document/item` は、`<document><item><value>10</value></item></document>` から `<item><value>10</value></item>` を抽出します。 名前空間はサポートされていないことに注意してください。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	CSV から JSON	CSV ファイルのデータを JSON 形式に変換します。 詳細については、CSV から JSON への前処理を参照してください。
	XML から JSON	XML 形式のデータを JSON に変換します。 詳細については、シリアライズルールを参照してください。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
SNMP		
	SNMP walk 値	指定した OID/MIB 名で値を抽出し、フォーマットオプションを適用します: 変更なし - Hex-STRING をエスケープされていない 16 進文字列として返します（注: 表示ヒントは引き続き適用されます）。 **Hex-STRING** から **UTF-8** - Hex-STRING を UTF-8 文字列に変換します。 **Hex-STRING** から **MAC** - Hex-STRING を MAC アドレス文字列に変換します（' 'は':'に置換されます）。 **BITS** から整数 - 16 進文字のシーケンス（例: "1A 2B 3C 4D"）として表現されたビット文字列の最初の 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、後続のバイトは無視されます。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	SNMP walk から JSON	SNMP 値を JSON に変換します。JSON 内のフィールド名と、対応する SNMP OID パスを指定します。フィールド値には、指定した SNMP OID パスの値が設定されます。 この前処理ステップは、SNMP OID ディスカバリに使用できます。 *SNMP walk* 値ステップと同様の値フォーマットオプションを使用できます。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	SNMP get 値	SNMP get 値にフォーマットオプションを適用します: **Hex-STRING** から **UTF-8** - Hex-STRING を UTF-8 文字列に変換します。 **Hex-STRING** から **MAC** - Hex-STRING を MAC アドレス文字列に変換します（' 'は':'に置換されます）。 **BITS** から整数 - 16 進文字のシーケンス（例: "1A 2B 3C 4D"）として表現されたビット文字列の最初の 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、後続のバイトは無視されます。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
カスタムスクリプト		

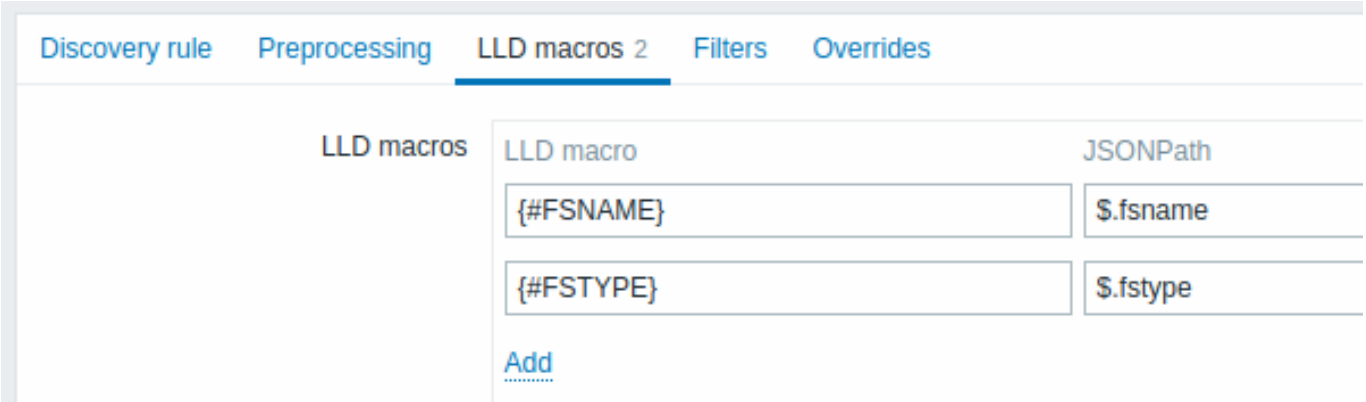
タイプ		
	JavaScript	パラメータフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターに、JavaScript コードを入力します。 使用できる JavaScript の長さは、使用するデータベースによって異なることに注意してください。 詳細については、Javascript 前処理を参照してください。
検証		
	正規表現に一致しない	値が一致してはならない正規表現を指定します。 例: `Error:(.*?)\.` 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	JSON 内のエラーを確認	JSONPath にあるアプリケーションレベルのエラーメッセージを確認します。確認に成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、前処理ステップの情報を追加せず、そのままユーザーに報告されることに注意してください。 例: `$.errors`。`{"errors":"e1"}` のような JSON を受信した場合、次の前処理ステップは実行されません。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	XML 内のエラーを確認	Xpath にあるアプリケーションレベルのエラーメッセージを確認します。確認に成功し、メッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、前処理ステップの情報を追加せず、そのままユーザーに報告されることに注意してください。 無効な XML の解析に失敗した場合、エラーは報告されません。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
	正規表現に一致	値が一致しなければならない正規表現を指定します。 失敗時のカスタム処理チェックボックスを選択すると、値を破棄する、指定した値を設定する、指定したエラーメッセージを設定する、のいずれかのカスタムエラー処理オプションを指定できます。
スロットリング		
	ハートビート付きで変更されていない値を破棄	定義した期間（秒単位）内に変更されていない値を破棄します。秒数の指定には正の整数値を使用できます（最小値は 1 秒）。このフィールドでは時間のサフィックスを使用できます（例: 30s、1m、2h、1d）。このフィールドではユーザーマクロおよびローレベルディスカバリマクロを使用できます。 ディスカバリ用アイテムには、1 つのスロットリングオプションのみ指定できます。 例: `1m`。同一のテキストが 60 秒以内にこのルールへ 2 回渡された場合、その値は破棄されます。 注: アイテムのプロトタイプを変更しても、スロットリングはリセットされません。スロットリングがリセットされるのは、前処理ステップを変更した場合のみです。
Prometheus		
	Prometheus から *JSON*	必要な Prometheus メトリクスを JSON に変換します。 詳細については、Prometheus チェックを参照してください。

ディスカバリルールがテンプレートを介してホストに適用されている場合、このタブの内容は読み取り専用になることに注意してください。

カスタムマクロ

LLD マクロタブでは、返される JSON にすでに `{#MACRO}` 形式のキーが含まれていない場合に、カスタムLLD マクロ を指定できます。

たとえば、vfs.fs.get エージェントアイテムキーは、事前定義された LLD マクロを含まない JSON filesystem data を返します。
 ここでは、マクロを手動で定義し、JSONPath 式を使用して JSON 内の値にマッピングします。



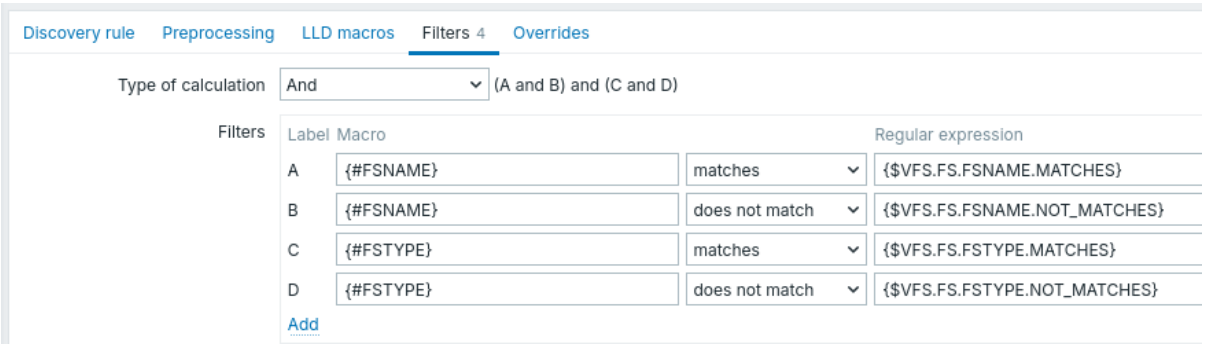
Parameter	Description
LLD macro	{#MACRO} 形式の LLD マクロ名。
JSONPath	JSON 配列内の各検出対象エンティティから抽出する値を定義する式。 たとえば、$.foo は [{"foo":"bar"}, {"foo":"baz"}] から "bar" と "baz" を抽出します。 JSONPath はドット表記とブラケット表記の両方をサポートしています。特殊文字または Unicode を含むキーにはブラケット表記を使用する必要があります。たとえば、 $['unicode + special chars #1']['unicode + special chars #2'] のように指定します。

検出ルールは、ルート ($.) に JSON 配列を返す必要があります。
 後方互換性のため、単一の "data" 配列を含む JSON オブジェクトも受け入れられます。この場合、"data" 配列は自動的に展開されます。

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

フィルター

フィルターを使用すると、条件に一致するエンティティに対してのみ、実際のアイテム、トリガー、グラフを生成できます。フィルタータブには、ディスカバリ値をフィルタリングできるディスカバリルールのフィルター定義が含まれています。



パラメーター	説明
計算のタイプ	フィルターの計算には、次のオプションを使用できます。 **And** - すべてのフィルターを通過する必要があります。 **Or** - 1 つのフィルターを通過すれば十分です。 **And/Or** - 異なるマクロ名には *And* を、同じマクロ名には *Or* を使用します。 カスタム計算式 - フィルターのカスタム計算を定義できます。計算式には、リスト内のすべてのフィルターを含める必要があります。255 文字に制限されています。

パラメーター	説明
フィルター	次のフィルター条件演算子を使用できます。一致する、一致しない、存在する、存在しない。 一致するおよび一致しない演算子では、Perl 互換正規表現（PCRE）が必要です。 正規表現フィールドには、正規表現を入力するか、グローバル正規表現を参照できます。 存在するおよび存在しない演算子を使用すると、応答内に指定した LLD マクロが存在するかどうかに基づいてエンティティをフィルタリングできます。 フィルターに含まれるマクロが応答にない場合、そのマクロに対して「存在しない」条件が指定されていない限り、検出されたエンティティは無視されます。 マクロが存在しないことによって計算式の結果に影響がある場合は、警告が表示されます。たとえば、次の場合です。 {#A} matches 1 and {#B} matches 2 - 警告が表示されます {#A} matches 1 or {#B} matches 2 - 警告は表示されません。

たとえば、C:、D:、E: のファイルシステムだけを対象にする場合は、マクロフィールドに {#FSNAME} を、正規表現フィールドに ^C|^D|^E を入力します。
 {#FSTYPE} マクロを使用してファイルシステムのタイプ（例：^ext|^reiserfs）でフィルタリングすることもできます。また、ドライブのタイプについては、{#FSDRIVETYPE} マクロ（例：fixed）を使用してフィルタリングできます（Windows エージェントでのみサポートされています）。

正規表現をテストするには、たとえば grep -E を使用します。

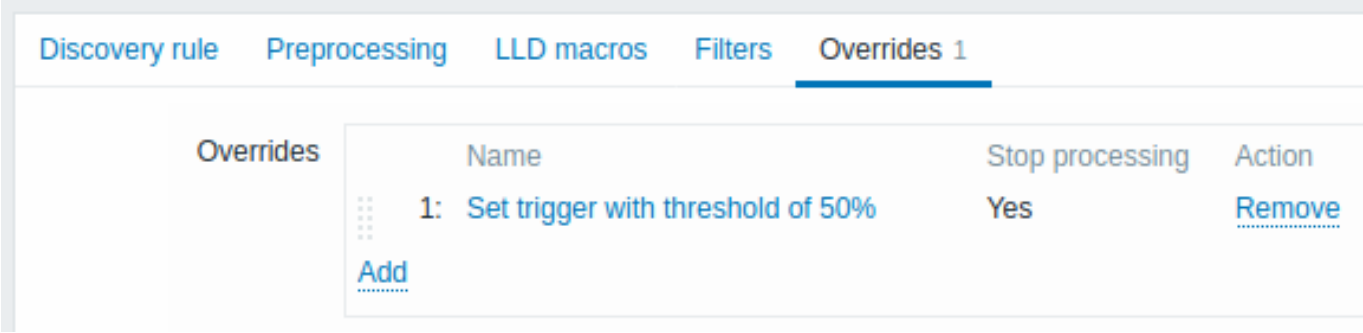
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
LLD ルールで使用する正規表現（たとえば、誤った「ディスカバリ対象のファイルシステム」正規表現）に誤りやタイプミスがあると、多数のホストに対して、数千件の設定要素、履歴値、イベントが削除される可能性があります。

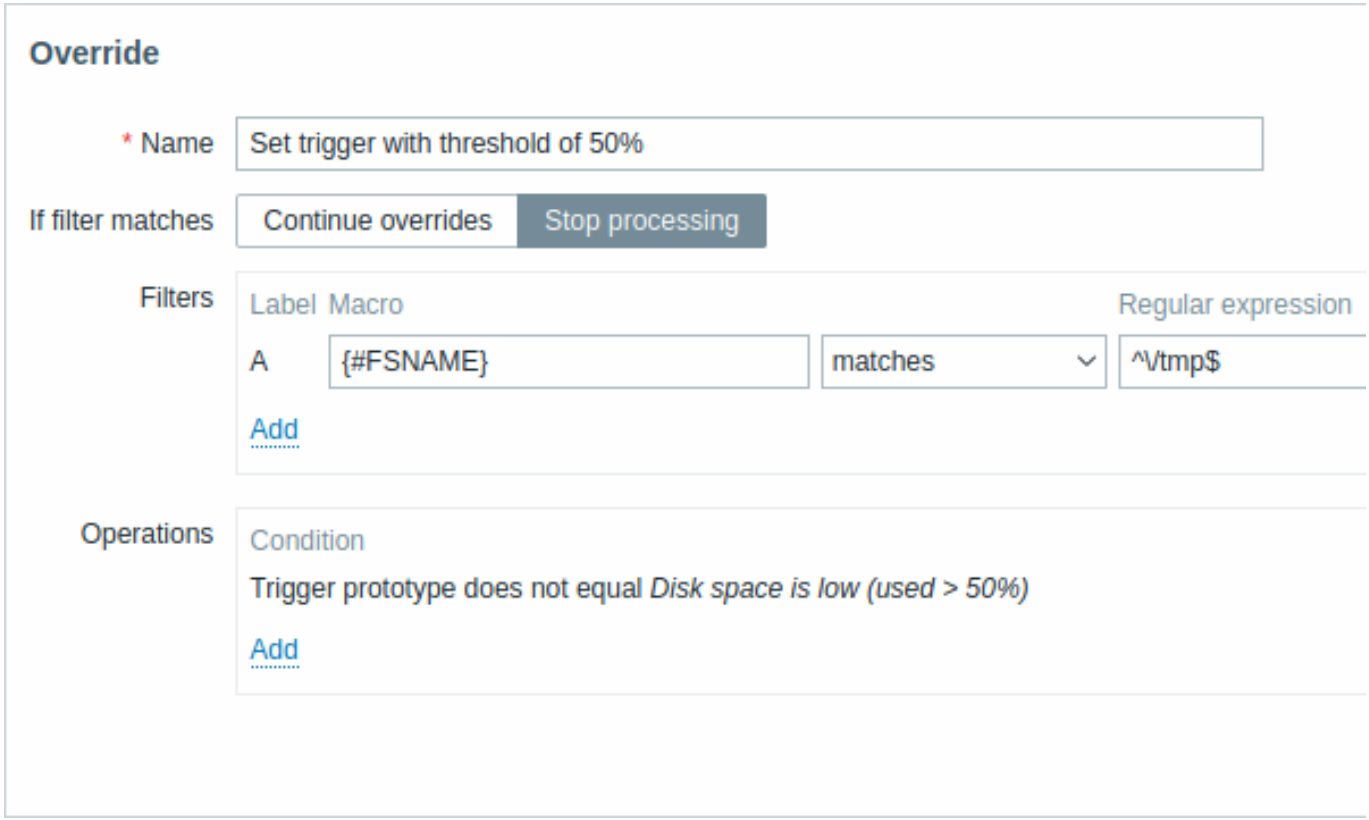
Attention:
大文字と小文字のみが異なるファイルシステム名を正しく検出するには、MySQL の Zabbix データベースを大文字と小文字を区別する設定で作成する必要があります。

オーバーライド

オーバーライドタブでは、指定した条件に一致する検出済みオブジェクトについて、アイテム、トリガー、グラフ、ホスト、ディスカバリのプロトタイプの一覧または属性を変更するルールを設定できます。



オーバーライド（存在する場合）は、ドラッグアンドドロップで並べ替え可能な一覧に表示され、定義された順序で実行されます。新しいオーバーライドの詳細を設定するには、オーバーライドブロックの Add をクリックします。既存のオーバーライドを編集するには、オーバーライド名をクリックします。ポップアップウィンドウが開き、オーバーライドルールの詳細を編集できます。

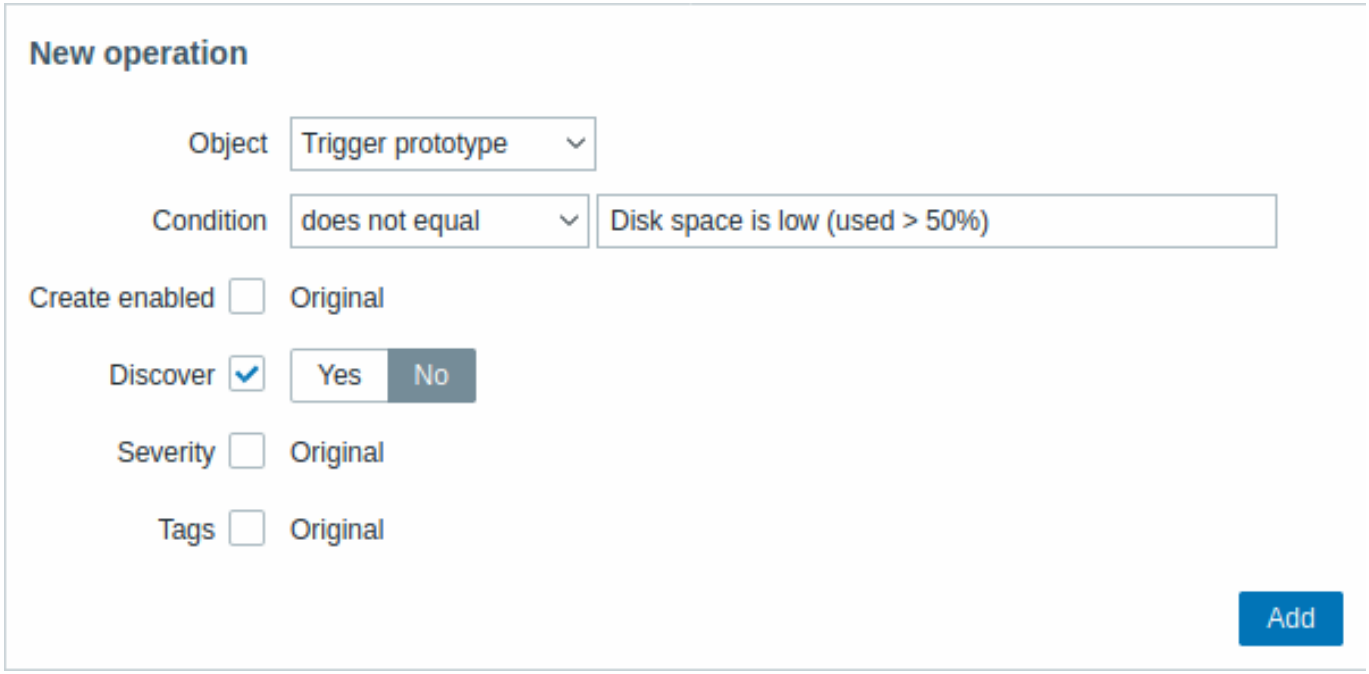


必須パラメータには赤いアスタリスクが付いています。

パラメータ	説明
名前	一意の（LLD ルールごとの）オーバーライド名。
フィルターが一致した場合	フィルター条件を満たしたときに、次のオーバーライドを処理するかどうかを定義します。 オーバーライドを続行 - 後続のオーバーライドが処理されます。 処理を停止 - 先行するオーバーライド（存在する場合）の操作とこのオーバーライドが実行され、条件に一致した LLD 行について後続のオーバーライドは無視されます。
フィルター	オーバーライドを適用する検出済みエンティティを決定します。オーバーライドフィルターはディスカバリルールのフィルターの後に処理され、同じ機能を持ちます。
操作	オーバーライド操作は、次の詳細とともに表示されます。 条件 - オブジェクトタイプと、オブジェクト名が満たすべき条件。例：トリガープロトタイプが *Disk space is low (used > 50%)* と等しくない。 アクション - 操作を編集および削除するリンクが表示されます。

操作の設定

新しい操作の詳細を設定するには、操作ブロックの Add をクリックします。既存の操作を編集するには、操作の横にある Edit をクリックします。操作の詳細を編集できるポップアップウィンドウが開きます。



パラメータ		説明
オブジェクト条件		5 種類のオブジェクトを使用できます。 アイテムプロトタイプ トリガープロトタイプ グラフプロトタイプ ホストプロトタイプ ディスカバリプロトタイプ 操作を適用するエンティティをフィルタリングできます。
	演算子	サポートされる演算子： 等しい - このプロトタイプに適用 等しくない - このプロトタイプを除くすべてのプロトタイプに適用 含む - プロトタイプ名にこの文字列が含まれる場合に適用 含まない - プロトタイプ名にこの文字列が含まれない場合に適用 一致する - プロトタイプ名が正規表現に一致する場合に適用 一致しない - プロトタイプ名が正規表現に一致しない場合に適用
	パターン：オブジェクトアイテムプロトタイプ	検索する正規表現または文字列。

パラメータ		説明
	有効として作成	チェックボックスを選択すると、元のアイテムプロトタイプ設定を上書きするボタンが表示されます。 はい - アイテムは有効な状態で追加されます。 いいえ - アイテムは検出済みエンティティに追加されますが、無効な状態になります。
	検出	チェックボックスを選択すると、元のアイテムプロトタイプ設定を上書きするボタンが表示されます。 はい - アイテムが追加されます。 いいえ - アイテムは追加されません。
	更新間隔	チェックボックスを選択すると、アイテムに異なる間隔を設定できる 2 つのオプションが表示されます。 遅延 - アイテムの更新間隔。ユーザーマクロ、LLD マクロ、および時間のサフィックス (例：30s、1m、2h、1d) がサポートされます。カスタム間隔を使用する場合は 0 に設定してください。 カスタム間隔 - Add をクリックして、柔軟な間隔またはスケジュール間隔を指定します。詳細はカスタム間隔を参照してください。
	ヒストリ	チェックボックスを選択すると、アイテムの異なるヒストリ保存期間を設定できるボタンが表示されます。 保存しない - 選択すると、ヒストリは保存されません。 保存期間 - 選択すると、保存期間を指定する入力フィールドが右側に表示されます。ユーザーマクロおよびLLD マクロがサポートされます。
	トレンド	チェックボックスを選択すると、アイテムの異なるトレンド保存期間を設定できるボタンが表示されます。 保存しない - 選択すると、トレンドは保存されません。 保存期間 - 選択すると、保存期間を指定する入力フィールドが右側に表示されます。ユーザーマクロおよびLLD マクロがサポートされます。
	タグ	チェックボックスを選択すると、タグと値のペアを指定できる新しいブロックが表示されます。ユーザーマクロおよびLLD マクロがサポートされます。 これらのタグは、タグ名が一致する場合でも、アイテムプロトタイプで指定されたタグに追加されます。
オブジェクト：トリガープロトタイプ		
	有効として作成	チェックボックスを選択すると、元のトリガープロトタイプ設定を上書きするボタンが表示されます。 はい - トリガーは有効な状態で追加されます。 いいえ - トリガーは検出済みエンティティに追加されますが、無効な状態になります。
	検出	チェックボックスを選択すると、元のトリガープロトタイプ設定を上書きするボタンが表示されます。 はい - トリガーが追加されます。 いいえ - トリガーは追加されません。
	深刻度	チェックボックスを選択すると、トリガーの深刻度ボタンが表示され、トリガーの深刻度を変更できます。

パラメータ		説明
	タグ	チェックボックスを選択すると、タグと値のペアを指定できる新しいブロックが表示されます。ユーザーマクロおよびLLD マクロがサポートされます。 これらのタグは、タグ名が一致する場合でも、トリガープロトタイプで指定されたタグに追加されます。
オブジェクト：グラフプロトタイプ		
	検出	チェックボックスを選択すると、元のグラフプロトタイプ設定を上書きするボタンが表示されます。 はい - グラフが追加されます。 いいえ - グラフは追加されません。
オブジェクト：ホストプロトタイプ		
	有効として作成	チェックボックスを選択すると、元のホストプロトタイプ設定を上書きするボタンが表示されます。 はい - ホストは有効な状態で作成されます。 いいえ - ホストは無効な状態で作成されます。
	検出	チェックボックスを選択すると、元のホストプロトタイプ設定を上書きするボタンが表示されます。 はい - ホストが検出されます。 いいえ - ホストは検出されません。
	テンプレートのリンク	チェックボックスを選択すると、テンプレートを指定する入力フィールドが表示されます。 テンプレート名の入力を開始するか、フィールドの横にある選択をクリックし、ポップアップウィンドウの一覧からテンプレートを選択します。 このオーバーライドのテンプレートは、ホストプロトタイプにすでにリンクされているすべてのテンプレートに追加されます。
	ホストインベントリ	チェックボックスを選択すると、ホストプロトタイプの異なるインベントリモードを選択できるボタンが表示されます。 無効 - ホストインベントリにデータを入力しません 手動 - 詳細を手動で入力します 自動 - 収集したメトリクスに基づいてホストインベントリデータを自動入力します。

パラメータ		説明
オブジェクト：ディスカバリプロトタイプ		
	有効として作成	チェックボックスを選択すると、元のディスカバリプロトタイプ設定を上書きするボタンが表示されます。 はい - ディスカバリルールは有効な状態で追加されます。 いいえ - ディスカバリルールは無効な状態で追加されます。
	検出	チェックボックスを選択すると、元のディスカバリプロトタイプ設定を上書きするボタンが表示されます。 はい - ディスカバリルールが追加されます。 いいえ - ディスカバリルールは追加されません。
	更新間隔	チェックボックスを選択すると、ディスカバリルールに異なる間隔を設定できる 2 つのオプションが表示されます。 遅延 - ルールの更新間隔。ユーザーマクロ、LLD マクロ、および時間のサフィックス（例：30s、1m、2h、1d）がサポートされます。カスタム間隔を使用する場合は 0 に設定してください。 カスタム間隔 - Add をクリックして、柔軟な間隔またはスケジュール間隔を指定します。詳細はカスタム間隔を参照してください。

フォームのボタン

フォーム下部のボタンを使用すると、いくつかの操作を実行できます。

Add	ディスカバリルールを追加します。このボタンは、新しいディスカバリルールの場合のみ使用できます。
Update	ディスカバリルールのプロパティを更新します。このボタンは、既存のディスカバリルールの場合のみ使用できます。
Clone	現在のディスカバリルールのプロパティに基づいて、別のディスカバリルールを作成します。
Execute now	ディスカバリルールに基づいて、直ちにディスカバリを実行します。ディスカバリルールはすでに存在している必要があります。詳しくはこちらを参照してください。 注直ちにディスカバリを実行する場合、設定キャッシュは更新されないため、結果にはディスカバリルール設定に対するごく最近の変更が反映されません。
Test	ディスカバリルールの設定をテストします。このボタンを使用すると、変更を恒久的に適用せずに、接続性やパラメータの正確性などの設定を確認できます。
Delete	ディスカバリルールを削除します。
Cancel	ディスカバリルールのプロパティの編集をキャンセルします。

検出されたエンティティ　以下のスクリーンショットは、ホストの設定で検出されたアイテム、トリガー、グラフがどのように表示されるかを示しています。検出されたエンティティには、それが由来するディスカバリールールへのオレンジ色のリンクが接頭辞として付けられます。

Items

All hosts / Zabbix server Enabled ZBX SNMP IPMI JMX Items 140 Triggers 77 Graphs 14 Discovery rules 6 Web scenarios

	Name ▲	Triggers	Key	Interval
	Mounted filesystem discovery: Get filesystems: FS [/]: Get data		vfs.fs.dependent[/,data]	
	Mounted filesystem discovery: FS [/]: Get data: FS [/]: Inodes: Free, in %	Triggers 2	vfs.fs.dependent.inode[/,pfree]	
	Mounted filesystem discovery: FS [/]: Get data: FS [/]: Option: Read-only	Triggers 1	vfs.fs.dependent[/,readonly]	
	Mounted filesystem discovery: FS [/]: Get data: FS [/]: Space: Available		vfs.fs.dependent.size[/,free]	
	Mounted filesystem discovery: FS [/]: Get data: FS [/]: Space: Total		vfs.fs.dependent.size[/,total]	
	Mounted filesystem discovery: FS [/]: Get data: FS [/]: Space: Used		vfs.fs.dependent.size[/,used]	
	Mounted filesystem discovery: FS [/]: Get data: FS [/]: Space: Used, in %	Triggers 2	vfs.fs.dependent.size[/,pused]	

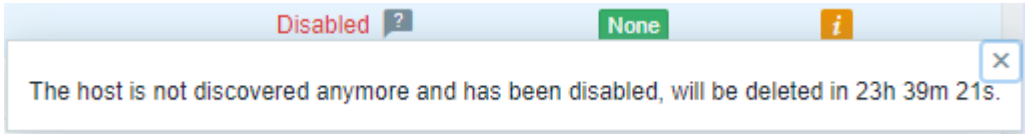
すでに同じ一意性基準を持つエンティティ（例えば、同じキーのアイテムや同じ名前のグラフ）が存在する場合、検出されたエンティティは作成されません。この場合、フロントエンドには、低レベルディスカバリールールが特定のエンティティを作成できなかったというエラーメッセージが表示されます。ただし、いくつかのエンティティが作成できずスキップされた場合でも、ディスカバリールール自体は未サポートにはなりません。ディスカバリールールは他のエンティティの作成/更新を続行します。

検出されたエンティティ（ホスト、ファイルシステム、インターフェースなど）が検出されなくなった場合（またはフィルターを通過しなくなった場合）、それに基づいて作成されたエンティティは自動的に無効化され、最終的に削除されることがあります。

失われたリソースは、失われたリソースを無効化パラメータの値に基づいて自動的に無効化される場合があります。これは失われたホスト、アイテム、トリガーに影響します。

失われたリソースは、失われたリソースを削除パラメータの値に基づいて自動的に削除される場合があります。これは失われたホスト、ホストグループ、アイテム、トリガー、グラフに影響します。

検出されたエンティティが「これ以上検出されない」状態になると、エンティティリストにライフタイムインジケーターが表示されます。マウスポインターをその上に移動すると、ステータスの詳細を示すメッセージが表示されます。



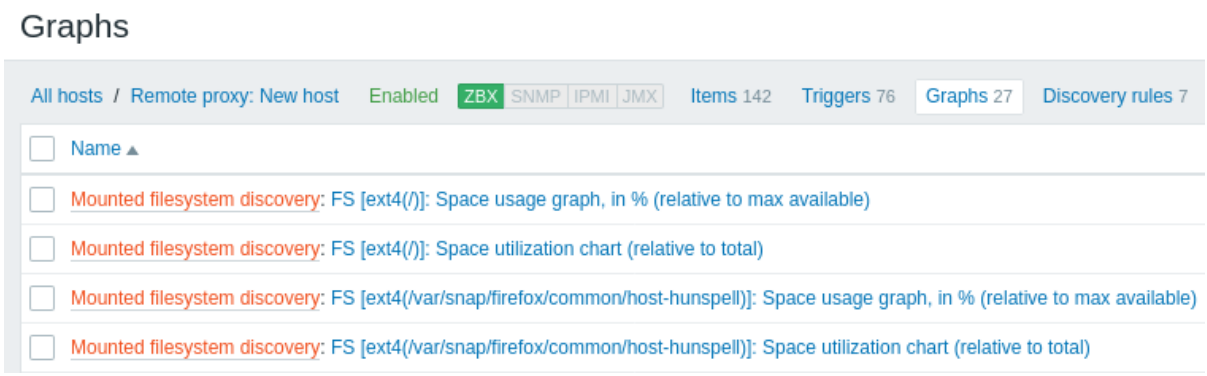
エンティティが削除対象としてマークされたが、期待された時刻に削除されなかった場合（ディスカバリールールやアイテムホストが無効化された場合）、次回ディスカバリールールが処理される際に削除されます。

他のエンティティを含むエンティティが削除対象としてマークされている場合、ディスカバリールールレベルで変更されても更新されません。例えば、LLD ベースのトリガーが削除対象のアイテムを含んでいる場合、それらは更新されません。

Triggers

All hosts / Zabbix server Enabled ZBX SNMP IPMI JMX Items 140 Triggers 77 Graphs 14 Discovery rules 6

	Severity	Value	Name ▲	Operational data
	Warning	OK	Mounted filesystem discovery: Linux: FS [/]: Running out of free inodes **Depends on:** Zabbix server: Linux: FS [/]: Running out of free inodes	Free inodes: {ITEM.LASTVALUE1}
	Warning	OK	Mounted filesystem discovery: Linux: FS [/]: Space is low **Depends on:** Zabbix server: Linux: FS [/]: Space is critically low	Space used: {{ITEM.LASTVALUE1}.fmtnum(1)}%



その他のディスカバリタイプ　他の種類の標準ディスカバリについての詳細やハウツーは、以下のセクションで確認できます。

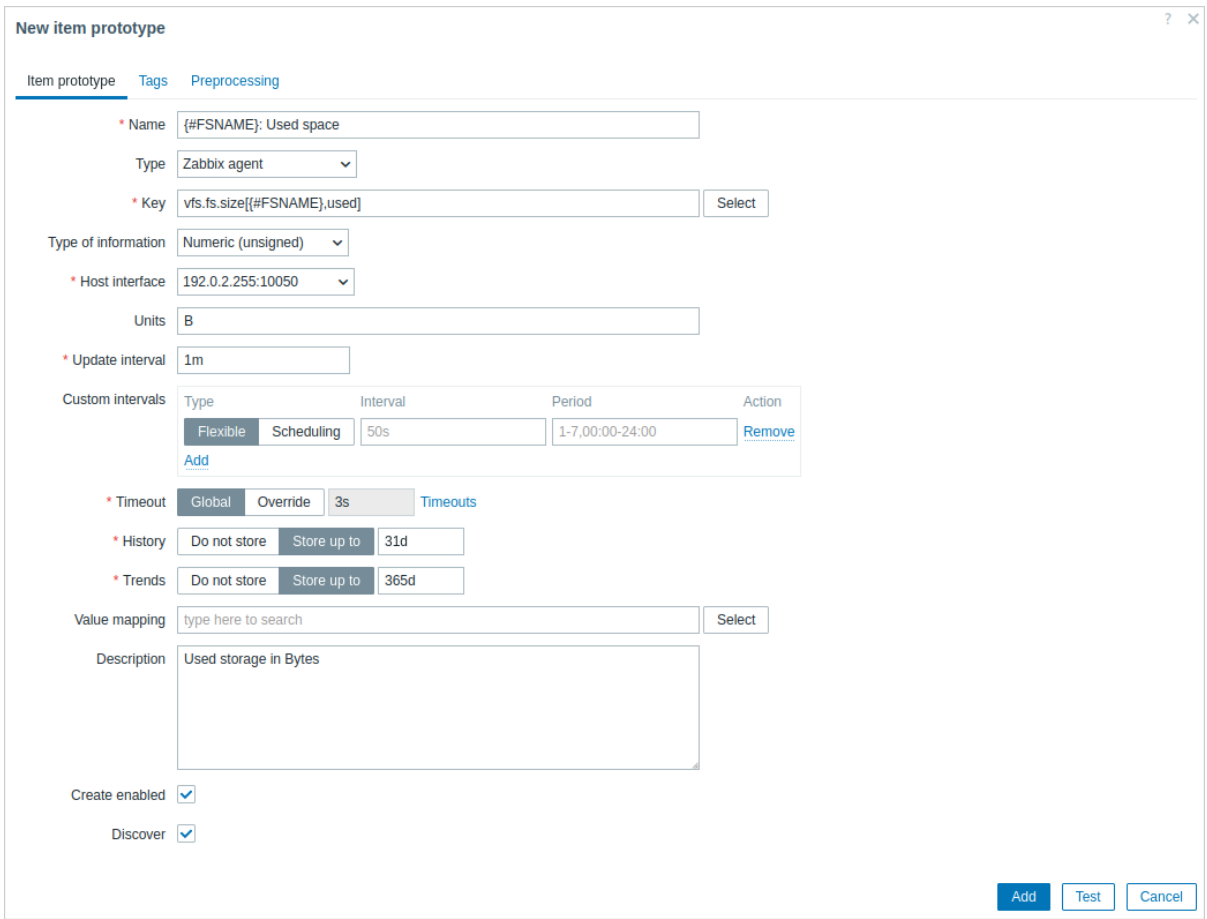
- ネットワークインターフェースのディスカバリ
- CPU および CPU コアのディスカバリ
- SNMP OIDのディスカバリ
- JMX オブジェクトのディスカバリ
- ODBC SQL クエリを使用したディスカバリ
- Windows サービスのディスカバリ
- Zabbix でのホストインターフェースのディスカバリ

ディスカバリアイテムの JSON 形式の詳細や、独自のファイルシステムディスカバリを Perl スクリプトとして実装する例については、カスタム LLD ルールの作成を参照してください。

アイテムのプロトタイプ

ルールを作成したら、そのルールのアイテムに移動し、アイテムのプロトタイプを作成をクリックしてアイテムのプロトタイプを作成します。

ファイルシステム名が必要な箇所で、`{#FSNAME}` マクロが使用されていることに注目してください。ディスカバリが正しく処理されるように、アイテムキーではローレベルディスカバリマクロを必ず使用する必要があります。ディスカバリルールが処理されると、このマクロは検出されたファイルシステムに置き換えられます。



Note:
アイテムキーのパラメータを使用しない場合は、アイテムキーの角括弧 [...] 内に LLD マクロを配置します（例: v_[{#MACRO}]）。

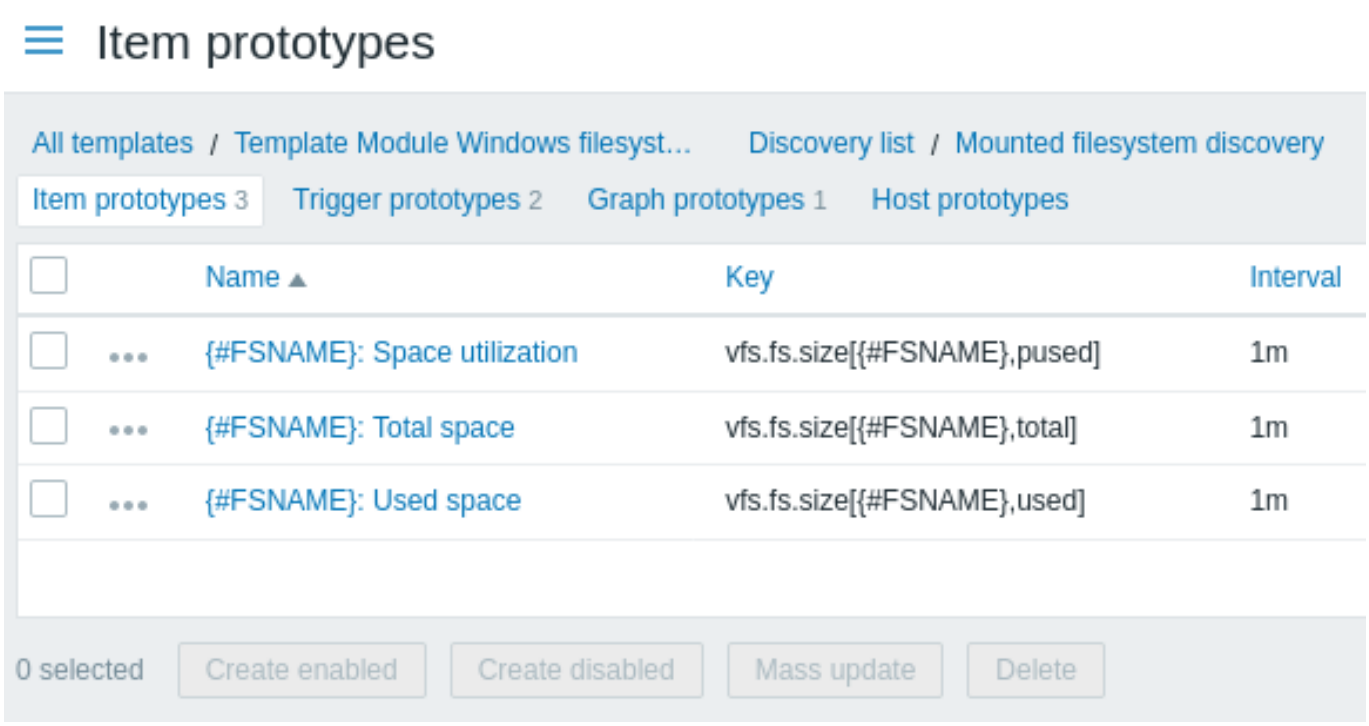
ローレベルディスカバリのマクロおよびユーザーマクロは、アイテムのプロトタイプの設定とアイテム値の前処理パラメータでサポートされています。更新間隔で使用する場合は、1 つのマクロでフィールド全体を埋める必要があります。1 つのフィールド内で複数のマクロを使用したり、マクロとテキストを混在させたりすることはできません。

Note:
正規表現および XPath の前処理パラメータで安全に使用できるように、ローレベルディスカバリマクロはコンテキストに応じてエスケープされます。

アイテムのプロトタイプ固有の属性:

パラメータ	説明
有効な状態で作成	チェックすると、アイテムは有効な状態で追加されます。 チェックしない場合、アイテムは検出されたエンティティに追加されますが、無効な状態になります。
ディスカバリ	チェックすると（デフォルト）、アイテムは検出されたエンティティに追加されます。 チェックしない場合、ディスカバリルールでこの設定が上書きされない限り、アイテムは検出されたエンティティに追加されません。

関心のある各ファイルシステムメトリクスについて、複数のアイテムのプロトタイプを作成できます。



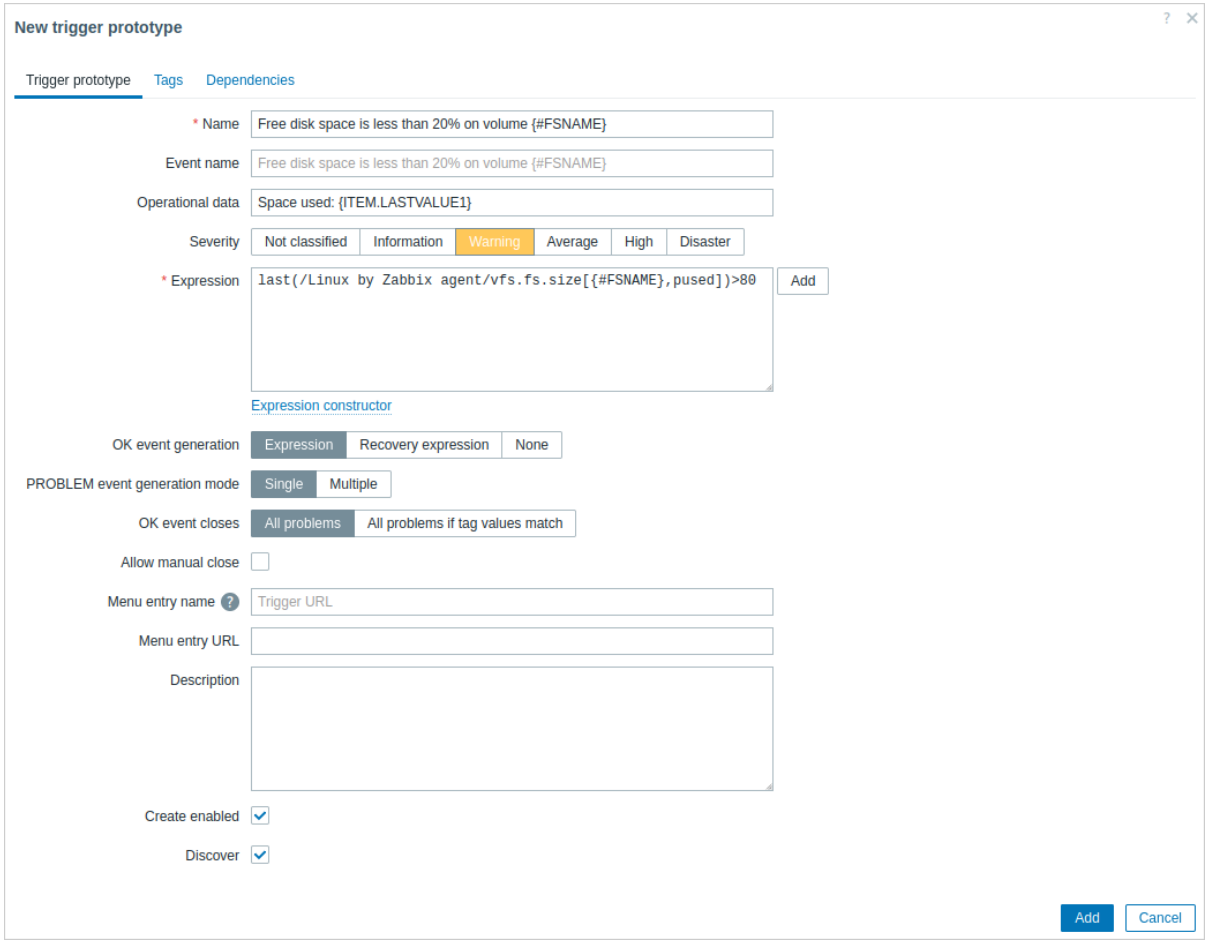
3 点アイコンをクリックすると、特定のアイテムのプロトタイプに関する次のオプションを含むメニューが開きます。

- トリガーのプロトタイプを作成 - このアイテムのプロトタイプに基づいてトリガーのプロトタイプを作成します
- トリガーのプロトタイプ - クリックすると、このアイテムのプロトタイプに設定済みのトリガーのプロトタイプへのリンク一覧が表示されます
- 依存アイテムを作成 - このアイテムのプロトタイプの依存アイテムを作成します

複数のアイテムのプロトタイプのプロパティを一度に更新する場合は、一括更新オプションを使用できます。

トリガープロトタイプ

トリガープロトタイプは、アイテムプロトタイプと同様の方法で作成します:

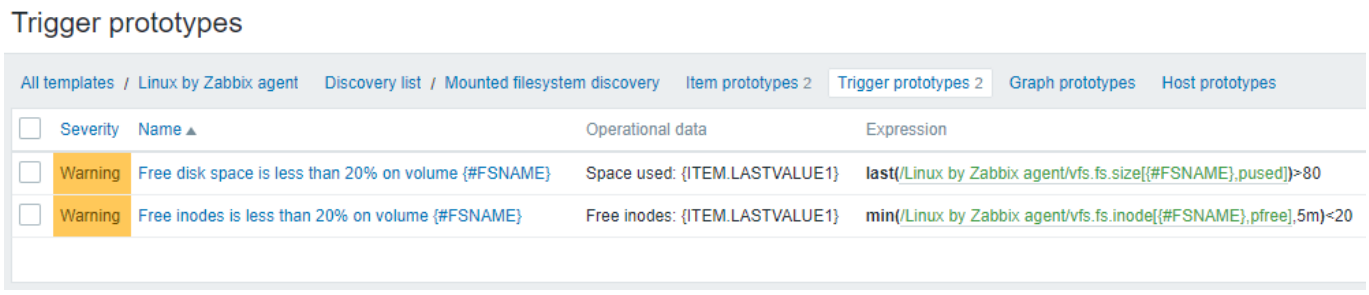


トリガープロトタイプに固有の属性:

Parameter	Description
Create enabled	チェックすると、トリガーは有効な状態で追加されます。 チェックしない場合、トリガーは検出されたエンティティに追加されますが、無効な状態になります。
Discover	チェックすると（デフォルト）、トリガーは検出されたエンティティに追加されます。 チェックしない場合、この設定が検出ルールで上書きされていない限り、トリガーは検出されたエンティティに追加されません。

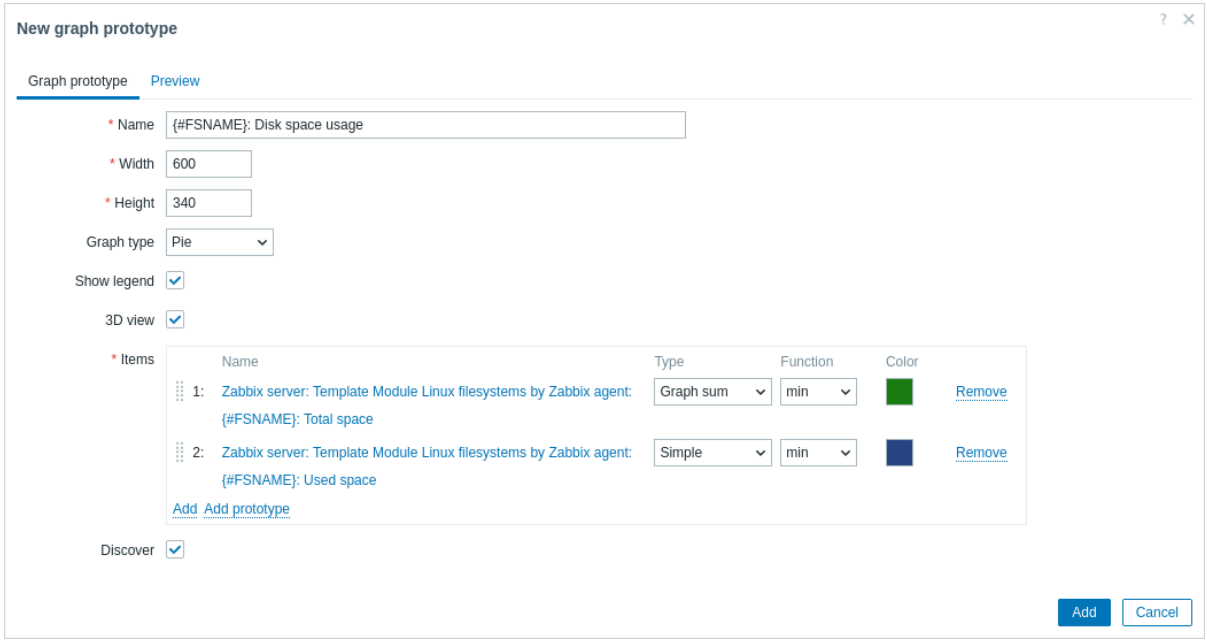
プロトタイプから実際のトリガーを作成する際、式で比較に使用する定数（例では'20'）を柔軟に変更できる必要がある場合があります。このような柔軟性を実現する方法として、コンテキスト付きユーザーマクロ が役立つことがあります。

トリガープロトタイプ間で依存関係を定義できます。これを行うには、*Dependencies* タブに移動します。トリガープロトタイプは、同じ低レベルディスカバリ（LLD）ルール内の別のトリガープロトタイプ、または通常のトリガーに依存できます。トリガープロトタイプは、別の LLD ルールのトリガープロトタイプや、トリガープロトタイプから作成されたトリガーには依存できません。ホストのトリガープロトタイプは、テンプレートのトリガーに依存できません。



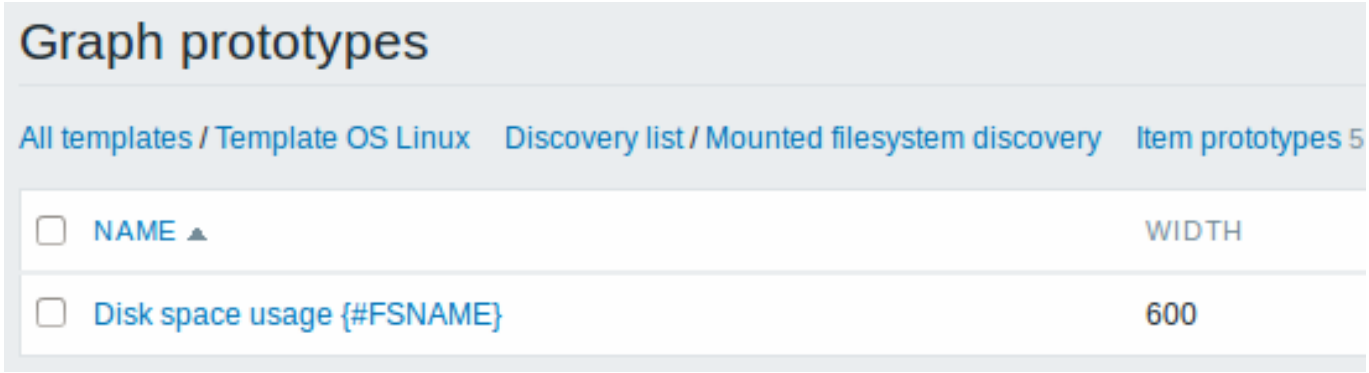
グラフプロトタイプ

グラフプロトタイプも作成できます。

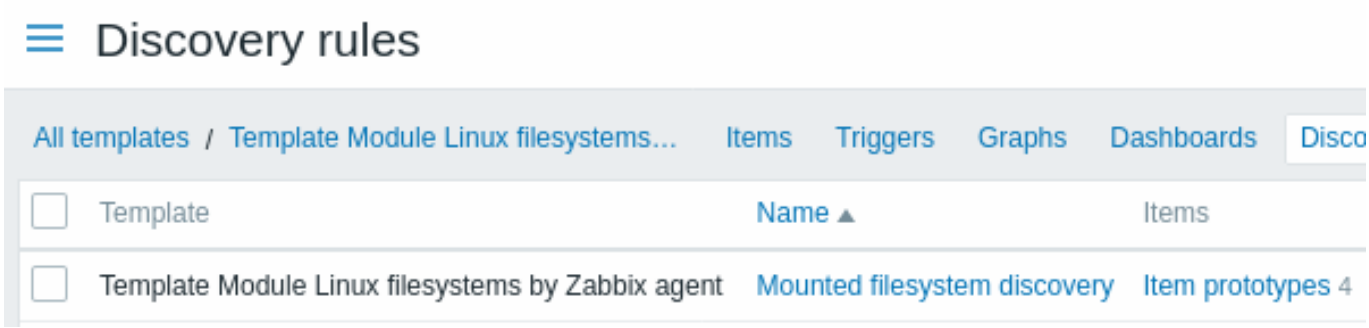


グラフプロトタイプに固有の属性:

Parameter	Description
Discover	チェックされている場合（デフォルト）、グラフは発見されたエンティティに追加されます。 チェックされていない場合、この設定が発見ルールで上書きされない限り、グラフは発見されたエンティティに追加されません。



最後に、以下のような発見ルールを作成しました。これには 5 つのアイテムプロトタイプ、2 つのトリガープロトタイプ、および 1 つのグラフプロトタイプがあります。



ホストプロトタイプ

ホストプロトタイプは、ローレベルディスカバリルールを通じてホストを作成するための設計図です。これらのプロトタイプは、ホストとして検出される前は、テンプレートからリンクされたものを除き、アイテムやトリガーを持つことはできません。

設定

ホストプロトタイプはlow-level discovery rules で設定します。

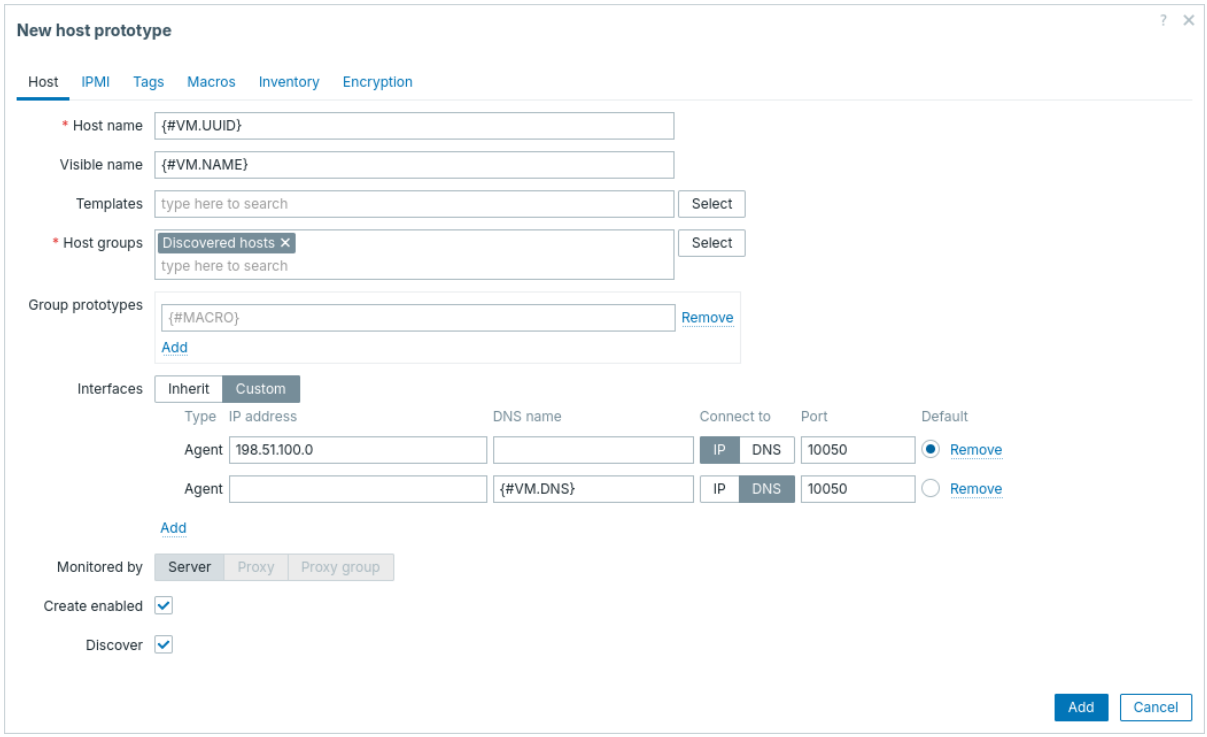
ホストプロトタイプを作成するには、次の手順を実行します。

1. *Data collection* > *Hosts* に移動します。

2. 必要なホストの *Discovery* をクリックして、そのホストに設定されている low-level discovery rules の一覧に移動します。

3. 必要な discovery rule の *Host prototypes* をクリックします。

4. 右上の *Create host prototype* ボタンをクリックし、設定フォームを完了します。



必須入力フィールドには赤いアスタリスクが付いています。

Parameter	Description
Host name	このパラメーターには、作成されるホストの名前を一意にするため、少なくとも 1 つのlow-level discovery macro を含める必要があります。
Visible name	Low-level discovery macros をサポートしています。
Group prototypes	low-level discovery macros を使用してホストグループプロトタイプを指定できます。 指定されたグループプロトタイプに基づいて、host groups が発見、作成され、作成されたホストにリンクされます。ほかの low-level discovery rules によってすでに作成されている発見済みグループも、作成されたホストにリンクされます。ただし、手動で 作成されたホストグループに一致する発見済みホストグループは、作成されたホストにリンクされません。
Interfaces	発見されたホストが discovery rule に属するホストから IP を継承するか（デフォルト）、またはcustom interfaces を取得するかを設定します。 Low-level discovery macros とuser macros をサポートしています。
Monitored by	このフィールドは、host configuration の対応するフィールドと自動的に同期されます。
Create enabled	発見されたホストのステータスを設定します。チェックしない場合、ホストは無効として作成されます。
Discover	ホストプロトタイプからホストを作成するかどうかを設定します。チェックしない場合、low-level discovery rule でこの設定がoverridden されていない限り、ホストプロトタイプからホストは作成されません。

Note:
タグ値およびホストプロトタイプの user macro 値でもlow-level discovery macros をサポートしています。
 Value maps はホストプロトタイプではサポートされていません。

ホストプロトタイプの設定例については、*Virtual machine monitoring* を参照してください。

ホストインターフェース

カスタムインターフェースを追加するには、インターフェースセレクターを「継承」から「カスタム」に切り替えます。Add をクリックし、インターフェースの種類（Zabbix エージェント、SNMP、JMX、IPMI）を選択します。

Note:
カスタムが選択されているがインターフェースが設定されていない場合、ホストはインターフェースなしで作成されます。
 継承が選択されていて、ホストプロトタイプがテンプレートに属している場合、検出されたすべてのホストは、テンプレートがリンクされているホストからホストインターフェースを継承します。

複数のカスタムインターフェースが指定されている場合、デフォルト列でプライマリインターフェースを設定できます。

カスタムホストインターフェースの設定例については、*VMware 監視セットアップ例*を参照してください。

Warning:
ホストインターフェースに正しいデータが含まれている場合のみ、ホストが作成されます。

検出されたホスト

ホストリストでは、検出されたホストの前に、それらを作成したディスカバリールールの名前が付けられます。

検出されたホストは、ほとんどのパラメータをホストプロトタイプから読み取り専用として継承します。次の検出されたホストのパラメータのみが設定可能です:

- テンプレート - 新しいテンプレートのリンクや手動で追加したテンプレートのリンク解除。ホストプロトタイプから継承されたテンプレートはリンク解除できません。
- 説明 - ホストの説明を追加または編集します。
- ステータス - ホストの有効化または無効化。
- タグ - 新しいタグの追加や手動で追加したタグの削除。ホストプロトタイプから継承されたタグは削除できません。タグは重複（同じ名前と値のタグ）できないことに注意してください。ホストプロトタイプに新しいタグが追加され、それが検出されたホストの手動で追加されたタグと一致する場合、ディスカバリー時に手動で追加されたタグは継承されたタグに置き換えられます。
- マクロ - 新しいホストマクロの追加や手動で追加したもの、ホストプロトタイプから継承したものの削除、マクロ値やタイプの変更。

Note:
検出されたホストは、ディスカバリールールが設定されているホストからユーザーマクロも継承します。これらの継承されたマクロは削除でき、値やタイプも変更できます。

検出されたホストが他のホストを検出することもできます。たとえば、ホストプロトタイプ（ハイパーバイザーの検出に使用）がリンクされているテンプレートに、独自のホストプロトタイプ（仮想マシンの検出に使用）を含む低レベルディスカバリールールが含まれている場合、Zabbix はハイパーバイザーとその仮想マシンの両方を検出します。これらの VM 上のコンテナも追加で検出するには、検出された VM に低レベルディスカバリールールを作成するか、ホストプロトタイプを含むテンプレートをリンクするか、またはホストプロトタイプにあらかじめテンプレートをリンクしておくことができます。

検出されたホストは手動で削除できます。ただし、ディスカバリーが有効になっている場合は再度検出されることに注意してください。

検出されなくなったホストは、次のいずれかになります:

- 自動的に無効化（ディスカバリールールの失われたリソースの無効化値に基づく）
- 自動的に削除（ディスカバリールールの失われたリソースの削除値に基づく）

ディスカバリプロトタイプ

概要

ディスカバリプロトタイプは、「親」ディスカバリルール内にネストされたローレベルディスカバリルールであり、それぞれ独自のアイテム、トリガーなどを持つオブジェクトを複数のレベルでディスカバリできます。例えば、データベースサーバー上のすべてのデータベースインスタンスをディスカバリし、次に各インスタンスの表領域をディスカバリし、さらに各表領域のテーブルをディスカバリできます。

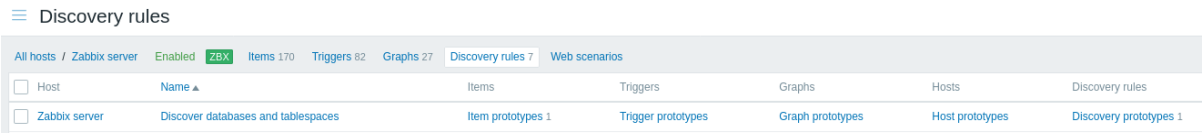
ディスカバリプロトタイプには、独自のアイテム、トリガー、グラフ、ホスト、およびディスカバリプロトタイプがあります。*Nested* タイプを指定すると、ネストされたディスカバリプロトタイプは親ルールと同じ JSON 値を使用します。

ディスカバリプロトタイプのネストレベルに制限はありません。

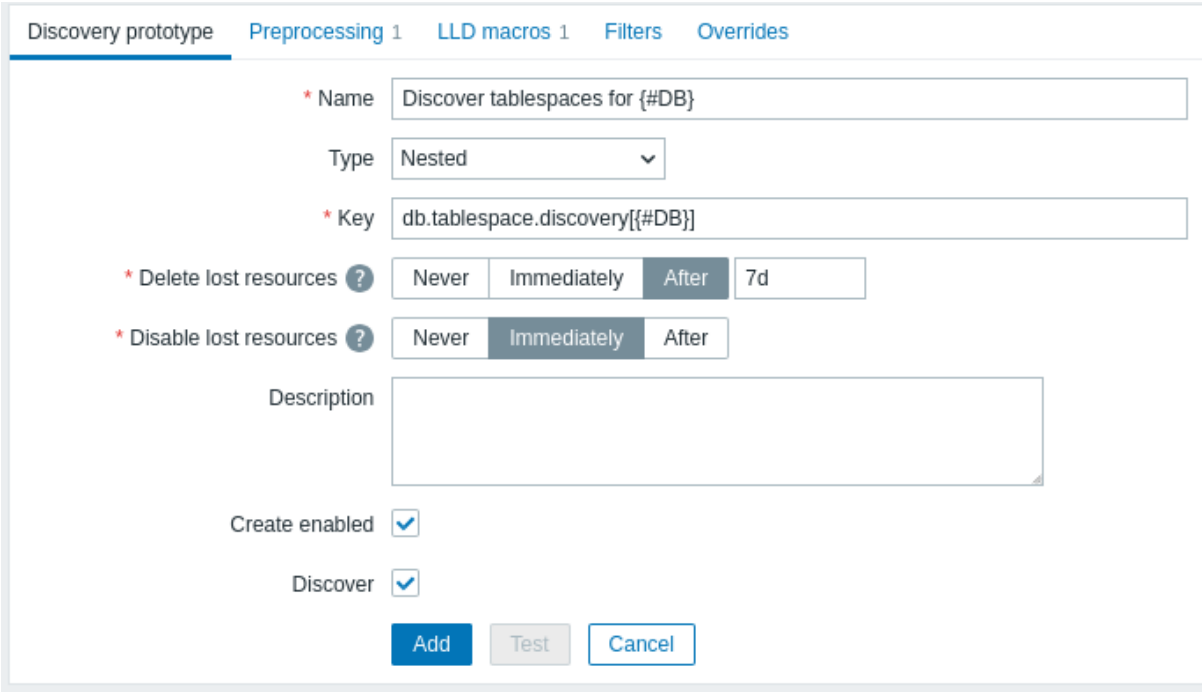
設定

ディスカバリプロトタイプを作成するには、次の手順を実行します。

- 既存のディスカバリルールの行で *Discovery prototypes* をクリックします



- *Create discovery prototype* をクリックします



このフォームの設定項目は、通常のlow-level discovery と共通です。

開いたディスカバリプロトタイプフォームで *Type* に”Nested” を選択すると、ディスカバリルール（ディスカバリプロトタイプから生成される）は、親のディスカバリルールと同じ JSON 値に含まれる JSON オブジェクトを基に生成されます。たとえば、元の JSON が [<object A>, <object B>] で、ネストされたディスカバリルールプロトタイプが 1 つある場合、object A と object B のデータをそれぞれ基に 2 つのディスカバリルールが生成されます。

この場合、ディスカバリプロトタイプは親ルールと同時に有効化されます。そのため、ネストされたルールでは前処理を使用して、親がすでに取得した同じデータの別の「スライス」を処理できます。

親の LLD ルールの LLD マクロは、ネストされたディスカバリルールでも利用できます。

検出されたホストでのネストされた LLD ルール

ネストされた低レベルディスカバリールールは、ホストプロトタイプに割り当てられたホストテンプレートで使用できます。検出されたホストにネストされたディスカバリールールが存在する場合、ホストの検出に使用された JSON オブジェクトは、このホストのすべてのネスト型 LLD ルールにも送信されます。詳細は例を参照してください。

ホストを作成したディスカバリールールの LLD マクロは、ネストされたディスカバリールールで利用できます。

例

次の複数レベルの JSON の例を受信することに基づいて、ディスカバリプロトタイプの考えられる適用例を示します。

```
[
  {
    "database": "db1",
    "created_at": "2024-02-01T12:30:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "10GB" },
      { "name": "ts2", "max_size": "20GB" },
      { "name": "ts3", "max_size": "15GB" }
    ]
  },
  {
    "database": "db2",
    "created_at": "2023-11-15T08:45:00Z",
    "encoding": "UTF16",
```

```
    "tablespaces": [
      { "name": "ts1", "max_size": "5GB" },
      { "name": "ts2", "max_size": "25GB" },
      { "name": "ts3", "max_size": "30GB" }
    ]
  },
  {
    "database": "db3",
    "created_at": "2024-01-05T15:10:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "12GB" },
      { "name": "ts2", "max_size": "18GB" },
      { "name": "ts3", "max_size": "22GB" }
    ]
  }
]
```

ケース 1

データベースサーバー上のデータベースインスタンスをディスカバリし、その後、各インスタンスの表領域をディスカバリします。

1. データベースサーバーのディスカバリに関連するホストが少なくとも 1 つ必要です。
2. このホストに対して、*Discover databases and tablespaces* という名前のLLD ルールを作成します。
3. このルールの *LLD* マクロタブに切り替え、マクロ `{#DB}=$.database` を追加します。
4. このルールに対して、*Active connections to {#DB}* という名前のアイテムのプロトタイプを追加します（タイプ: エージェント、キー: `db.connections[{#DB}]`）。
5. 各データベースに関連するアイテムがディスカバリされます。

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. このルールに対して、*Discover tablespaces for {#DB}* という名前のディスカバリのプロトタイプを作成します（タイプ: ネスト、キー: `db.tablespace.discovery[{#DB}]`）。
7. このディスカバリのプロトタイプの 前処理タブに切り替え、ステップ `JSONPath=$.tablespaces` を追加します。
8. このディスカバリのプロトタイプの *LLD* マクロタブに切り替え、マクロ `{#TSNAME}=$.name` を追加します。
9. このディスカバリのプロトタイプに対して、*Size of tablespace {#TSNAME} for {#DB}* という名前のアイテムのプロトタイプを作成します（タイプ: エージェント、キー: `db.ts.size[{#DB}, {#TSNAME}]`）。
10. 各データベースの各表領域に関連するアイテムがディスカバリされます。

```
Size of tablespace ts1 for db1, Size of tablespace ts2 for db1, Size of tablespace ts3 for db1,
Size of tablespace ts1 for db2, Size of tablespace ts2 for db2, Size of tablespace ts3 for db2,
Size of tablespace ts1 for db3, Size of tablespace ts2 for db3, Size of tablespace ts3 for db3.
```

キーは `db.ts.size[db1,ts1]`、`db.ts.size[db1,ts2]`、... `db.ts.size[db3,ts3]` です。

ケース 2

データベースサーバー上のデータベースインスタンスを検出済みホストとして表現して検出し、その後、各インスタンスの表領域を検出します。

1. データベースサーバーの検出に関連するホスト（ルートホスト）が少なくとも 1 つ必要です。
2. 各データベースの表領域を検出するためのテンプレートを作成します。
3. このテンプレートに、*{#DB}* へのアクティブ接続という名前のアイテムを作成します（タイプ: エージェント、キー: `db.connections[{#DB}]`）。
4. このテンプレートに、表領域を検出という名前のLLD ルール を作成します（タイプ: ネスト）。
5. このルールの *Preprocessing* タブに切り替え、ステップ `JSONPath=$.tablespaces` を追加します。
6. このルールの *LLD Macros* タブに切り替え、マクロ `{#TSNAME}=$.name` を追加します。
7. このルールのアイテムプロトタイプとして、*{#DB}* の表領域 *{#TSNAME}* のサイズという名前のアイテムを作成します（タイプ: エージェント、キー: `db.ts.size[{#DB}, {#TSNAME}]`）。

8. ルートホストに戻り、このホストに データベースと表領域を検出という名前のLLD ルール を作成します。
9. このルールの *LLD Macros* タブに切り替え、マクロ `{#DB}=$.database` を追加します。
10. このルールに、データベース *{#DB}* 用ホストという名前のホストプロトタイプを追加します。
11. このホストプロトタイプの *Macros* タブに切り替え、マクロ `{$DB}={#DB}` を追加します（手順 3 のアイテム名とキー用）。
12. 手順 2 のテンプレートをこのホストプロトタイプにリンクします。
13. 検出されたホストには、各データベースとその表領域に関連する検出済みアイテムが含まれます。

ホスト	アイテム
データベース *db1* 用ホスト	db1 へのアクティブ接続
	db1 の表領域 ts1 のサイズ
	db1 の表領域 ts2 のサイズ
	db1 の表領域 ts3 のサイズ
データベース *db2* 用ホスト	db2 へのアクティブ接続
	db2 の表領域 ts1 のサイズ
	db2 の表領域 ts2 のサイズ
	db2 の表領域 ts3 のサイズ
データベース *db3* 用ホスト	db3 へのアクティブ接続
	db3 の表領域 ts1 のサイズ
	db3 の表領域 ts2 のサイズ
	db3 の表領域 ts3 のサイズ

ローレベルディスカバリに関する注意事項

ユーザーマクロコンテキストでの LLD マクロの使用

LLD マクロはユーザーマクロコンテキスト内で使用できます。たとえば、トリガープロトタイプで使用できます。

同じアイテムに対する複数の LLD ルール

同じディスカバリ・アイテムを使用して、複数のローレベルディスカバリルールを定義できます。

これを行うには、Alias エージェントパラメータを定義します。これにより、異なるディスカバリルールで変更したディスカバリ・アイテムキーを使用できます。例えば、`vfs.fs.discovery[foo]`、`vfs.fs.discovery[bar]` などです。

返り値のデータ制限

Zabbix サーバーが直接受信する場合、ローレベルディスカバリルールの JSON データには制限がありません。これは、返り値がデータベースに保存されることなく処理されるためです。

カスタムローレベルディスカバリルールにも制限はありません。ただし、カスタムローレベルディスカバリルールのデータがユーザーパラメータを使用して取得される場合は、ユーザーパラメータの返り値の制限が適用されます。

データが Zabbix プロキシを経由する必要がある場合、このデータをデータベースに保存する必要があります。この場合、データベースの制限が適用されます。

ディスカバリルールの例

このセクションでは、ファイルシステム、ネットワークインターフェース、サービス、センサー、SNMP オブジェクト、Java アプリケーション、データベースなど、さまざまな種類のエンティティに対するディスカバリルールの設定例を紹介します。

各例では、使用するアイテムキーまたはクエリ、LLD マクロを含む JSON 出力、およびそれらの結果を使ってアイテム、トリガー、グラフのプロトタイプを作成する方法を含め、ディスカバリルールをどのように構成できるかを示します。

- ブロックデバイスのディスカバリ
- CPU および CPU コアのディスカバリ
- Zabbix におけるホストインターフェースのディスカバリ
- IPMI センサーのディスカバリ
- JMX オブジェクトのディスカバリ
- マウント済みファイルシステムのディスカバリ
- ネットワークインターフェースのディスカバリ
- SNMP OID のディスカバリ
- SNMP OID のディスカバリ（旧方式）
- systemd サービスのディスカバリ

- Windows パフォーマンスカウンターインスタンスのディスカバリ
- Windows サービスのディスカバリ
- WMI クエリを使用したディスカバリ
- ODBC SQL クエリを使用したディスカバリ
- Prometheus データを使用したディスカバリ

カスタム **LLD** ルール

概要

完全にカスタムの LLD ルールを作成し、任意の種類のエンティティをディスカバリすることも可能です。例えば、データベースサーバー上のデータベースをディスカバリできます。

これを行うには、検出されたオブジェクトと、必要に応じてそれらのいくつかのプロパティを指定した JSON 文字列を返すカスタムアイテムを作成する必要があります。
エンティティごとのマクロ数に制限はありません。組み込みのディスカバリルールでは 1 つまたは 2 つのマクロ（例えば、ファイルシステムのディスカバリでは 2 つ）が返されますが、さらに多くのマクロを返すことも可能です。

例

必要な JSON 文字列形式は、例を見るのが最もわかりやすいでしょう。
古い Zabbix 1.8 エージェント（`vfs.fs.discovery` キーをサポートしていないもの）を実行しているが、それでもファイルシステムをディスカバリする必要があるとします。
以下は、マウントされたファイルシステムを検出し、ファイルシステム名とタイプの両方を含む JSON を出力する Linux 用のシンプルな Perl スクリプトです。
これを使用する 1 つの方法は、キー `vfs.fs.discovery_perl` を持つ UserParameter として使用することです。

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"{#FSNAME}\":\"$fsname\",\n";
    print "\t\t\"{#FSTYPE}\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

Attention:
LLD マクロ名で使用できる記号は **0-9**、**A-Z**、**_**、**.** です。
名前に小文字は使用できません。

その出力例（わかりやすくするために整形したもの）を以下に示します。
カスタムディスカバリチェック用の JSON も同じ形式に従う必要があります。

```
[
    { "{#FSNAME}":"/",                      "{#FSTYPE}":"rootfs"   },
    { "{#FSNAME}":"/sys",                   "{#FSTYPE}":"sysfs"    },
    { "{#FSNAME}":"/proc",                  "{#FSTYPE}":"proc"     },
    { "{#FSNAME}":"/dev",                   "{#FSTYPE}":"devtmpfs" },
    { "{#FSNAME}":"/dev/pts",               "{#FSTYPE}":"devpts"   },
    { "{#FSNAME}":"/lib/init/rw",           "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/dev/shm",               "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/home",                  "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/tmp",                   "{#FSTYPE}":"ext3"     },
```

```
    { "{#FSNAME}":"/usr",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/var",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/sys/fs/fuse/connections",   "{#FSTYPE}":"fusectl"  }
]
```

前の例では、キーがプロトタイプで使用される LLD マクロ名と一致している必要があります。別の方法として、JSONPath を使用して LLD マクロ値を {#FSNAME} → $.fsname および {#FSTYPE} → $.fstype のように抽出することもでき、その場合は次のようなスクリプトが可能です。

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"fsname\":\"$fsname\",\n";
    print "\t\t\"fstype\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

その出力例（わかりやすくするために整形したもの）を以下に示します。

```
[
    { "fsname":"/",                         "fstype":"rootfs"   },
    { "fsname":"/sys",                      "fstype":"sysfs"    },
    { "fsname":"/proc",                     "fstype":"proc"     },
    { "fsname":"/dev",                      "fstype":"devtmpfs" },
    { "fsname":"/dev/pts",                  "fstype":"devpts"   },
    { "fsname":"/lib/init/rw",              "fstype":"tmpfs"    },
    { "fsname":"/dev/shm",                  "fstype":"tmpfs"    },
    { "fsname":"/home",                     "fstype":"ext3"     },
    { "fsname":"/tmp",                      "fstype":"ext3"     },
    { "fsname":"/usr",                      "fstype":"ext3"     },
    { "fsname":"/var",                      "fstype":"ext3"     },
    { "fsname":"/sys/fs/fuse/connections",  "fstype":"fusectl"  }
]
```

その後、ディスカバリルールの *Filter* フィールドで、マクロとして {#FSTYPE}、正規表現として `rootfs|ext3` を指定できます。

Note:
カスタム LLD ルールでは、FSNAME/FSTYPE のようなマクロ名を使用する必要はなく、任意の名前を自由に使用できます。
JSONPath を使用する場合、LLD 行はオブジェクトである配列要素になりますが、別の配列や値であることもあります。

なお、ユーザーパラメータを使用する場合、戻り値は 16 MB に制限されます。
詳細については、LLD 戻り値のデータ制限 を参照してください。

仮想マシンの監視

概要　Zabbix は、ローレベルディスカバリルールを使用して、VMware ハイパーバイザーおよび仮想マシンを自動的に検出し、事前定義されたホストプロトタイプに基づいて監視するホストを作成できます。

Zabbix には、VMware vCenter または ESXi ハイパーバイザーの監視用にすぐに使えるテンプレートも含まれています。

必要な VMware vCenter または vSphere の最小バージョンは 5.1 です。

データ収集　仮想マシンの監視は 2 つのステップで構成されます。

1. Zabbix の *vmware collector* プロセスが仮想マシンのデータを収集します。これらのプロセスは SOAP プロトコルを介して VMware の Web サービスから必要な情報を取得し、前処理を行い、Zabbix サーバーの共有メモリに保存します。
2. Zabbix の *poller* プロセスが、Zabbix のシンプルチェックであるVMware 監視アイテムキーを使用してデータを取得します。

Zabbix は収集したデータを VMware の構成データと VMware のパフォーマンスカウンターデータに分けます。両方のタイプのデータは *vmware collector* プロセスによって独立して収集されます。

VMware のパフォーマンスカウンター情報に基づいて、以下の統計情報が利用可能です。

- データストア
- ディスクデバイス
- CPU
- 電源
- ネットワークインターフェース
- カスタムパフォーマンスカウンターアイテム

VMware のパフォーマンスカウンターからデータを取得するアイテムの完全なリストについては、*VMware* 監視アイテムキーを参照してください。

VMware イベントの取得頻度はvmware.eventlogのポーリング間隔に依存しますが、5 秒未満にはできませんのでご注意ください。

設定　Zabbix サーバーがソース からコンパイルされる場合、仮想マシン監視を有効にするには `--with-libcurl --with-libxml2` の設定オプションを指定してコンパイルする必要があります。
Zabbix パッケージは、これらのオプションがすでに有効な状態でコンパイルされています。

仮想マシン監視のために、次の Zabbix サーバー設定ファイルパラメータを変更できます。

- StartVMwareCollectors

Note:
監視対象の VMware サービス数よりも多くのコレクターを有効にすることを推奨します。
そうしないと、VMware のパフォーマンスカウンター統計の取得が、VMware 設定データの取得によって遅延する可能性があります（大規模な環境では時間がかかります）。

 一般に、StartVMwareCollectors の値は 2 を下回らず、かつ監視対象の VMware サービス数の 2 倍を超えないようにする必要があります: サービス数 < StartVMwareCollectors < (サービス数 * 2)。
たとえば、1 つの VMware サービスを監視する場合は StartVMwareCollectors を 2 に設定し、3 つのサービスを監視する場合は StartVMwareCollectors を 5 に設定します。

 必要なコレクター数は、VMware 環境の規模や VMwareFrequency および VMwarePerfFrequency の設定パラメータにも依存することに注意してください。

- VMwareCacheSize
- VMwareFrequency
- VMwarePerfFrequency
- VMwareTimeout

Attention:
データストア容量メトリクスをサポートするには、VMware の vpxd.stats.maxQueryMetrics キーの値を少なくとも 64 に設定してください。
詳細については、*VMware Knowledge Base* 記事 を参照してください。

ディスカバリ

Zabbix は、低レベルディスカバリルール（たとえば、vmware.hv.discovery[{$VMWARE.URL}]）を使用して、VMware のハイパーバイザーと仮想マシンを自動的に検出できます。
さらに、Zabbix はホストプロトタイプを使用して、検出されたエンティティに対する実際のホストを自動生成できます。
詳細については、ホストプロトタイプ を参照してください。

設定例

VMware FQDN テンプレートを使用して VMware を監視するための Zabbix の基本的な設定方法については、*Zabbix で VMware を監視する* を参照してください。

ホスト、低レベルディスカバリルール、および VMware 監視用のホストプロトタイプを作成する方法のより詳細な例については、設定例 を参照してください。

詳細ログ　*vmware collector* プロセスによって収集されたデータは、デバッグレベル 5 を使用して詳細なデバッグ情報としてログに記録できます。デバッグレベルは、サーバーおよびプロキシの設定ファイルで設定するか、ランタイムコントロールオプション `-R log_level_increase="vmware collector,N"` を使用して設定できます。ここで、N はプロセス番号です。

たとえば、すべての *vmware collector* プロセスのデバッグレベルを 4 から 5 に引き上げるには、次のコマンドを実行します。

```
zabbix_server -R log_level_increase="vmware collector"
```

2 番目の *vmware collector* プロセスのデバッグレベルを 4 から 5 に引き上げるには、次のコマンドを実行します。

```
zabbix_server -R log_level_increase="vmware collector,2"
```

VMware collector データの詳細ログが不要になった場合は、`-R log_level_decrease` コマンドを実行して、デバッグレベルをデフォルト値（3）に戻すことを推奨します。

トラブルシューティング

- メトリクスが利用できない場合は、最近の VMware vSphere のバージョンでそれらが利用不可にされたり、デフォルトで無効化されたりしていないこと、またはパフォーマンスメトリクスのデータベースクエリに対して何らかの制限が設定されていないことを確認してください。詳細は ZBX-12094 を参照してください。
- `config.vpxd.stats.maxQueryMetrics` が無効であるか、許可される最大文字数を超えているというエラーが表示される場合は、vCenter Server の設定に `config.vpxd.stats.maxQueryMetrics` パラメータを追加してください。このパラメータの値は、VMware の `web.xml` ファイルにある `maxQuerysize` の値と同じである必要があります。詳細は *VMware Knowledge Base の記事* を参照してください。
- Zabbix のインストールでメモリ使用量が多すぎる疑いがある場合は、tcmalloc を使用した過剰なメモリ使用量のプロファイリング を参照してください。

VMware 監視アイテムキー

概要　このページでは、VMware 環境の監視に使用できるシンプルチェックの詳細を説明します。メトリクスは監視対象ごとにグループ化されています。

サポートされるアイテムキー　アイテムキーは、パラメータや追加情報を含めずに一覧表示されています。アイテムキーをクリックすると、詳細を表示できます。

アイテムキー	説明	アイテムグループ
vmware.eventlog	VMware イベントログ。	General service
vmware.fullname	VMware サービスの完全名。	
vmware.version	VMware サービスのバージョン。	
vmware.cl.perfcounter	VMware クラスタのパフォーマンスカウンターメトリクス。	Cluster
vmware.cluster.alarms.get	VMware クラスタのアラームデータ。	
vmware.cluster.discovery	VMware クラスタの検出。	
vmware.cluster.property	VMware クラスタのプロパティ。	
vmware.cluster.status	VMware クラスタのステータス。	
vmware.cluster.tags.get	VMware クラスタのタグ配列。	
vmware.datastore.alarms.get	VMware データストアのアラームデータ。	Datastore
vmware.datastore.discovery	VMware データストアの検出。	
vmware.datastore.hv.list	データストアのハイパーバイザー一覧。	
vmware.datastore.perfcounter	VMware データストアのパフォーマンスカウンター値。	
vmware.datastore.property	VMware データストアのプロパティ。	
vmware.datastore.read	データストアからの読み取り操作にかかる時間。	
vmware.datastore.size	VMware データストアの容量。バイト単位、または総容量に対する割合で表示されます。	
vmware.datastore.tags.get	VMware データストアのタグ配列。	
vmware.datastore.write	データストアへの書き込み操作にかかる時間。	
vmware.dc.alarms.get	VMware データセンターのアラームデータ。	Datacenter
vmware.dc.discovery	VMware データセンターの検出。	
vmware.dc.tags.get	VMware データセンターのタグ配列。	
vmware.dvswitch.discovery	VMware vSphere Distributed Switch の検出。	vSphere Distributed Switch

アイテムキー	説明	アイテムグループ
vmware.dvswitch.fetchportstate	VMware vSphere Distributed Switch のポートデータ。	
vmware.hv.alarms.get	VMware ハイパーバイザーのアラームデータ。	Hypervisor
vmware.hv.cluster.name	VMware ハイパーバイザーのクラスタ名。	
vmware.hv.connectionstate	VMware ハイパーバイザーの接続状態。	
vmware.hv.cpu.usage	VMware ハイパーバイザーのプロセッサ使用率（Hz）。	
vmware.hv.cpu.usage.perf	間隔中の VMware ハイパーバイザーのプロセッサ使用率（パーセント）。	
vmware.hv.cpu.utilization	間隔中の VMware ハイパーバイザーのプロセッサ使用率（パーセント）。電源管理または HT に依存します。	
vmware.hv.datacenter.name	VMware ハイパーバイザーのデータセンター名。	
vmware.hv.datastore.discovery	VMware ハイパーバイザーのデータストアの検出。	
vmware.hv.datastore.list	VMware ハイパーバイザーのデータストア一覧。	
vmware.hv.datastore.multipath	利用可能なデータストアパスの数。	
vmware.hv.datastore.read	データストアからの読み取り操作にかかる平均時間。	
vmware.hv.datastore.size	VMware データストアの容量。バイト単位、または総容量に対する割合で表示されます。	
vmware.hv.datastore.write	データストアへの書き込み操作にかかる平均時間。	
vmware.hv.discovery	VMware ハイパーバイザーの検出。	
vmware.hv.diskinfo.get	VMware ハイパーバイザーのディスクデータ。	
vmware.hv.fullname	バージョン情報を含む完全な製品名。	
vmware.hv.hw.cpu.freq	VMware ハイパーバイザーのプロセッサ周波数。	
vmware.hv.hw.cpu.model	VMware ハイパーバイザーのプロセッサモデル。	
vmware.hv.hw.cpu.num	VMware ハイパーバイザー上のプロセッサコア数。	
vmware.hv.hw.cpu.threads	VMware ハイパーバイザー上のプロセッサスレッド数。	
vmware.hv.hw.memory	VMware ハイパーバイザーの総メモリサイズ。	
vmware.hv.hw.model	VMware ハイパーバイザーのモデル。	
vmware.hv.hw.sensors.get	VMware ハイパーバイザーのハードウェアセンサー値。	
vmware.hv.hw.serialnumber	VMware ハイパーバイザーのシリアル番号。	
vmware.hv.hw.uuid	VMware ハイパーバイザーの BIOS UUID。	
vmware.hv.hw.vendor	VMware ハイパーバイザーのベンダー名。	
vmware.hv.maintenance	VMware ハイパーバイザーのメンテナンス状態。	
vmware.hv.memory.size.ballooned	VMware ハイパーバイザーのバルーンメモリサイズ。	
vmware.hv.memory.used	VMware ハイパーバイザーの使用済みメモリサイズ。	
vmware.hv.net.if.discovery	VMware ハイパーバイザーのネットワークインターフェースの検出。	
vmware.hv.network.in	VMware ハイパーバイザーのネットワーク入力統計。	
vmware.hv.network.linkspeed	VMware ハイパーバイザーのネットワークインターフェース速度。	
vmware.hv.network.out	VMware ハイパーバイザーのネットワーク出力統計。	
vmware.hv.perfcounter	VMware ハイパーバイザーのパフォーマンスカウンター値。	
vmware.hv.property	VMware ハイパーバイザーのプロパティ。	
vmware.hv.power	VMware ハイパーバイザーの電力使用量。	
vmware.hv.sensor.health.state	VMware ハイパーバイザーの正常性状態の集約センサー。	
vmware.hv.sensors.get	VMware ハイパーバイザーの HW ベンダー状態センサー。	
vmware.hv.status	VMware ハイパーバイザーのステータス。	
vmware.hv.tags.get	VMware ハイパーバイザーのタグ配列。	
vmware.hv.uptime	VMware ハイパーバイザーの稼働時間。	
vmware.hv.version	VMware ハイパーバイザーのバージョン。	
vmware.hv.vm.num	VMware ハイパーバイザー上の仮想マシン数。	
vmware.rp.cpu.usage	VMware Resource Pool における、間隔中の CPU 使用率（hertz）。	Resource pool
vmware.rp.memory	VMware リソースプールのメモリメトリクス。	
vmware.alarms.get	VMware 仮想センターのアラームデータ。	Virtual center
vmware.vm.alarms.get	VMware 仮想マシンのアラームデータ。	Virtual machine
vmware.vm.attribute	VMware 仮想マシンのカスタム属性値。	
vmware.vm.cluster.name	VMware 仮想マシン名。	
vmware.vm.consolidationneeded	VMware 仮想マシンのディスクは統合が必要です。	
vmware.vm.cpu.latency	物理 CPU へのアクセス競合により、仮想マシンが実行できなかった時間の割合。	
vmware.vm.cpu.num	VMware 仮想マシン上のプロセッサ数。	
vmware.vm.cpu.readiness	仮想マシンは実行可能な状態だったが、物理 CPU 上で実行するようスケジュールされなかった時間の割合。	

アイテムキー	説明	アイテムグループ
vmware.vm.cpu.ready	仮想マシンは実行可能な状態だったが、物理 CPU 上で実行するようスケジュールされなかった時間。	
vmware.vm.cpu.swapwait	スワップイン待ちに費やされた CPU 時間の割合。	
vmware.vm.cpu.usage	VMware 仮想マシンのプロセッサ使用率（Hz）。	
vmware.vm.cpu.usage.perf	間隔中の VMware 仮想マシンのプロセッサ使用率（パーセント）。	
vmware.vm.datacenter.name	VMware 仮想マシンのデータセンター名。	
vmware.vm.discovery	VMware 仮想マシンの検出。	
vmware.vm.guest.memory.size.swapped	スワップ領域に退避されたゲスト物理メモリの量。	
vmware.vm.guest.osuptime	前回のオペレーティングシステム起動からの経過時間。	
vmware.vm.hv.maintenance	VMware 仮想マシンのハイパーバイザーのメンテナンス状態。	
vmware.vm.hv.name	VMware 仮想マシンのハイパーバイザー名。	
vmware.vm.memory.size	VMware 仮想マシンの総メモリサイズ。	
vmware.vm.memory.size.ballooned	VMware 仮想マシンのバルーンメモリサイズ。	
vmware.vm.memory.size.compressed	VMware 仮想マシンの圧縮メモリサイズ。	
vmware.vm.memory.size.consumed	ゲスト物理メモリページをバックアップするために消費されたホスト物理メモリ量。	
vmware.vm.memory.size.private	VMware 仮想マシンのプライベートメモリサイズ。	
vmware.vm.memory.size.shared	VMware 仮想マシンの共有メモリサイズ。	
vmware.vm.memory.size.swapped	VMware 仮想マシンのスワップ済みメモリサイズ。	
vmware.vm.memory.size.usage.guest	VMware 仮想マシンのゲストメモリ使用量。	
vmware.vm.memory.size.usage.host	VMware 仮想マシンのホストメモリ使用量。	
vmware.vm.memory.usage	消費されたホスト物理メモリの割合。	
vmware.vm.net.if.discovery	VMware 仮想マシンのネットワークインターフェースの検出。	
vmware.vm.net.if.in	VMware 仮想マシンのネットワークインターフェース入力統計。	
vmware.vm.net.if.out	VMware 仮想マシンのネットワークインターフェース出力統計。	
vmware.vm.net.if.usage	間隔中の VMware 仮想マシンのネットワーク使用率。	
vmware.vm.perfcounter	VMware 仮想マシンのパフォーマンスカウンター値。	
vmware.vm.powerstate	VMware 仮想マシンの電源状態。	
vmware.vm.property	VMware 仮想マシンのプロパティ。	
vmware.vm.snapshot.get	VMware 仮想マシンのスナップショット状態。	
vmware.vm.state	VMware 仮想マシンの状態。	
vmware.vm.storage.committed	VMware 仮想マシンのコミット済みストレージ容量。	
vmware.vm.storage.readoio	収集間隔中の仮想ディスクへの未完了読み取り要求の平均数。	
vmware.vm.storage.totalreadlatency	仮想ディスクからの読み取りにかかる平均時間。	
vmware.vm.storage.totalwritelatency	仮想ディスクへの書き込みにかかる平均時間。	
vmware.vm.storage.uncommitted	VMware 仮想マシンの未コミットストレージ容量。	
vmware.vm.storage.unshared	VMware 仮想マシンの非共有ストレージ容量。	
vmware.vm.storage.writeoio	収集間隔中の仮想ディスクへの未完了書き込み要求の平均数。	
vmware.vm.tags.get	VMware 仮想マシンのタグ配列。	
vmware.vm.tools	VMware 仮想マシンのゲストツールの状態またはバージョン。	
vmware.vm.uptime	VMware 仮想マシンの稼働時間。	
vmware.vm.vfs.dev.discovery	VMware 仮想マシンのディスクデバイスの検出。	
vmware.vm.vfs.dev.read	VMware 仮想マシンのディスクデバイス読み取り統計。	
vmware.vm.vfs.dev.write	VMware 仮想マシンのディスクデバイス書き込み統計。	
vmware.vm.vfs.fs.discovery	VMware 仮想マシンのファイルシステムの検出。	
vmware.vm.vfs.fs.size	VMware 仮想マシンのファイルシステム統計。	

アイテムキーの詳細　山括弧のないパラメータは必須です。山括弧 **< >** で囲まれたパラメータはオプションです。

vmware.eventlog[url,<mode>,<severity>]

 VMware イベントログです。
 戻り値: ログ。

パラメータ:

- **url** - VMware サービスの URL。
- **mode** - *all* (デフォルト) または *skip* - 古いデータの処理をスキップします。
- **severity** - 重大度でフィルタリングします: *error*、*warning*、*info*、または *user*。複数の重大度をカンマ区切りのリストで指定する場合、このパラメータは引用符で囲む必要があります（例: `"error,warning,info,user"`）。デフォルトでは無効です。

コメント:

- URL ごとに `vmware.eventlog` アイテムキーは 1 つだけである必要があります。
- VMware イベントログレコードのフィルタリングの例も参照してください。

vmware.fullnameurl

 VMware サービスのフルネーム。
 戻り値：文字列。

パラメータ:

- **url** - VMware サービスの URL。

vmware.versionurl

 VMware サービスのバージョン。
 戻り値：*String*。

パラメータ:

- **url** - VMware サービスの URL。

vmware.cl.perfcounter[url,id,path,<instance>]

 VMware クラスタのパフォーマンスカウンタのメトリクス。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware クラスタの ID。id は `vmware.cluster.discovery[]` から {#CLUSTER.ID} として取得できます。
- **path** - パフォーマンスカウンタのパス[1]
- **instance** - パフォーマンスカウンタのインスタンス

vmware.cluster.alarms.get[url,id]

 VMware クラスタのアラームデータ。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **id** - VMware クラスタ ID。

vmware.cluster.discoveryurl

 VMware クラスタのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.cluster.property[url,id,prop]

 VMware クラスタのプロパティ。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **id** - VMware クラスタの ID
- **prop** - VMware SDKで定義されている VM オブジェクトのプロパティ名であるプロパティパス

例:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 VMware クラスタのステータス。
 戻り値: *0* - グレー; *1* - グリーン; *2* - イエロー; *3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **name** - VMware クラスタ名

vmware.cluster.tags.get[url,id]

 VMware クラスタタグの配列。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **id** - VMware クラスタ ID。

このアイテムは vSphere 6.5 以降で動作します。

vmware.datastore.alarms.get[url,uuid]

 VMware データストアのアラームデータ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware データストアのグローバル一意識別子。

vmware.datastore.discovery[url,<filter_uuid>]

 VMware データストアのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **filter_uuid** - 単一の HV または VM の一意の識別子 (デフォルト: 空)。指定した場合、リンクされたデータストアのみがディスカバリされます。

vmware.datastore.hv.list[url,datastore]

 データストアのハイパーバイザーのリスト。
 戻り値: *String*。

パラメーター:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前

出力例:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 VMware データストアのパフォーマンスカウンタ値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware データストアのグローバル一意識別子
- **path** - パフォーマンスカウンタのパス[1]
- **instance** - パフォーマンスカウンタのインスタンス。集計値の場合は空のインスタンスを使用します (デフォルト)。`instance` は `vmware.datastore.discovery[]` から {#DATASTORE.EXTENT} 配列の一部として取得できます。

vmware.datastore.property[url,uuid,prop]

 VMware データストアのプロパティ。
 戻り値：文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware データストアのグローバル一意識別子
- **prop** - プロパティパス

vmware.datastore.read[url,datastore,<mode>]

 データストアからの読み取り操作にかかった時間（ミリ秒）。
 戻り値：整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前
- **mode** - *latency*（平均値、デフォルト）または *maxlatency*（最大値）

vmware.datastore.size[url,datastore,<mode>]

 VMware データストアの容量（バイト単位または全体に対するパーセンテージ）。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。

パラメータ:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前
- **mode** - 指定可能な値: *total*（デフォルト）、*free*、*pfree*（空き容量のパーセンテージ）、*uncommitted*

vmware.datastore.tags.get[url,uuid]

 VMware データストアのタグ配列。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware データストアのグローバル一意識別子。

このアイテムは vSphere 6.5 以降で動作します。

vmware.datastore.write[url,datastore,<mode>]

 データストアへの書き込み操作にかかる時間 (ミリ秒)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **datastore** - データストアの UUID または名前
- **mode** - *latency* (平均値、デフォルト) または *maxlatency* (最大値)

vmware.dc.alarms.get[url,id]

 VMware データセンターのアラームデータ。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **id** - VMware データセンター ID。

vmware.dc.discoveryurl

 VMware データセンターのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.dc.tags.get[url,id]

 VMware データセンターのタグ配列。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **id** - VMware データセンターの ID。

このアイテムは vSphere 6.5 以降で動作します。

vmware.dvswitch.discoveryurl

 VMware vSphere Distributed Switch のディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 VMware vSphere Distributed Switch のポートデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware DVSwitch のグローバル一意識別子。
- **filter** - ポートを選択するための、カンマ区切りの条件を含む単一の文字列;
- **mode** - *state* (「config」XML ノードを除くすべての XML、デフォルト) または *full*。

filter パラメータは、VMware データオブジェクト DistributedVirtualSwitchPortCriteria で利用可能な criteria をサポートします。

例:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 VMware ハイパーバイザーのアラームデータ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービス URL;

- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.cluster.name[url,uuid]

 VMware ハイパーバイザークラスタ名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.connectionstate[url,uuid]

 VMware ハイパーバイザーの接続状態。
 戻り値: *String*: *connected*、*disconnected*、または *notResponding*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cpu.usage[url,uuid]

 VMware ハイパーバイザーのプロセッサ使用率 (Hz)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cpu.usage.perf[url,uuid]

 インターバル中の VMware ハイパーバイザーのプロセッサ使用率 (%)。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.cpu.utilization[url,uuid]

 インターバル中の VMware ハイパーバイザーのプロセッサ使用率 (%)。電源管理または HT に依存します。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.datacenter.name[url,uuid]

 VMware ハイパーバイザのデータセンター名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザのグローバル一意識別子

vmware.hv.datastore.discovery[url,uuid]

 VMware ハイパーバイザーデータストアのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.datastore.list[url,uuid]

 VMware ハイパーバイザーのデータストアのリスト。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

出力例:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 利用可能なデータストアパスの数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **partitionid** - `vmware.hv.datastore.discovery` からの物理デバイスの内部 ID

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 データストアからの読み取り操作の平均時間 (ミリ秒)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **mode** - *latency* (デフォルト)

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 VMware データストアの容量 (バイト単位または全体に対するパーセンテージ)。
 返り値: *Integer* - バイト単位; *Float* - パーセンテージ。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **mode** - 指定可能な値: *total* (デフォルト), *free*, *pfree* (空き容量のパーセンテージ), *uncommitted*

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 データストアへの書き込み操作にかかる平均時間 (ミリ秒)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **datastore** - データストアの UUID または名前
- **mode** - *latency* (デフォルト)

vmware.hv.discoveryurl

 VMware ハイパーバイザーのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.hv.diskinfo.get[url,uuid]

 VMware ハイパーバイザーのディスクデータ。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.fullname[url,uuid]

 バージョン情報を含む完全な製品名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.freq[url,uuid]

 VMware ハイパーバイザーのプロセッサ周波数 (Hz)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL

- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.model[url,uuid]

 VMware ハイパーバイザーのプロセッサモデル。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.num[url,uuid]

 VMware ハイパーバイザーのプロセッサコア数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.cpu.threads[url,uuid]

 VMware ハイパーバイザー上のプロセッサスレッド数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.memory[url,uuid]

 VMware ハイパーバイザーのメモリ総容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.model[url,uuid]

 VMware ハイパーバイザーモデル。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.hw.sensors.get[url,uuid]

 VMware ハイパーバイザーのハードウェアセンサー値。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.serialnumber[url,uuid]

 VMware ハイパーバイザーのシリアル番号。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere API 6.7 以降で動作します。

vmware.hv.hw.uuid[url,uuid]

 VMware ハイパーバイザーの BIOS UUID。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere API 6.7 以降で動作します。

vmware.hv.hw.vendor[url,uuid]

 VMware ハイパーバイザーのベンダー名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは vSphere API 6.7 以降で動作します。

vmware.hv.maintenance[url,uuid]

 VMware ハイパーバイザーのメンテナンスステータスです。
 戻り値: *0* - メンテナンス中ではない; *1* - メンテナンス中。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.memory.size.ballooned[url,uuid]

 VMware ハイパーバイザーのバルーンメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.memory.used[url,uuid]

 VMware ハイパーバイザーの使用メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.net.if.discovery[url,uuid]

 VMware ハイパーバイザーのネットワークインターフェースのディスカバリ。
 戻り値: *JSON* オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.network.in[url,uuid,<mode>]

 VMware ハイパーバイザーのネットワーク入力統計情報 (1 秒あたりのバイト数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **mode** - *bps* (デフォルト), *packets*, *dropped*, *errors*, *broadcast*

vmware.hv.network.linkspeed[url,uuid,ifname]

 VMware ハイパーバイザーのネットワークインターフェースの速度。
 戻り値：整数。ネットワークインターフェースがダウンしている場合は *0*、それ以外の場合はインターフェースの速度値を返します。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **ifname** - インターフェース名

vmware.hv.network.out[url,uuid,<mode>]

 VMware ハイパーバイザーのネットワーク出力統計情報 (1 秒あたりのバイト数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **mode** - *bps* (デフォルト), *packets*, *dropped*, *errors*, *broadcast*

vmware.hv.perfcounter[url,uuid,path,<instance>]

 VMware ハイパーバイザのパフォーマンスカウンタ値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザのグローバル一意識別子
- **path** - パフォーマンスカウンタのパス [1]
- **instance** - パフォーマンスカウンタのインスタンス。集計値の場合は空のインスタンスを使用します (デフォルト)。

vmware.hv.property[url,uuid,prop]

 VMware ハイパーバイザーのプロパティ。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **prop** - プロパティパス

vmware.hv.power[url,uuid,<max>]

 VMware ハイパーバイザーの消費電力 (W)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子
- **max** - 許容される最大消費電力

vmware.hv.sensor.health.state[url,uuid]

 VMware ハイパーバイザーのヘルス状態ロールアップセンサー。
 戻り値: 整数: *0* - グレー; *1* - グリーン; *2* - イエロー; *3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

VMware は *VMware Rollup Health State* センサーを非推奨にしたため、このアイテムは VMware vSphere 6.5 以降では動作しない場合があります。

vmware.hv.sensors.get[url,uuid]

 VMware ハイパーバイザーの HW ベンダー状態センサー。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.status[url,uuid]

 VMware ハイパーバイザーのステータス。
 戻り値: 整数: *0* - グレー; *1* - グリーン; *2* - イエロー; *3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムはホストシステムの全体的なステータスプロパティを使用します。

vmware.hv.tags.get[url,uuid]

 VMware ハイパーバイザーのタグ配列。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは vSphere 6.5 以降で動作します。

vmware.hv.uptime[url,uuid]

 VMware ハイパーバイザーの稼働時間 (秒)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムはホストシステムの全体的なステータスプロパティを使用します。

vmware.hv.version[url,uuid]

 VMware ハイパーバイザーのバージョン。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.hv.vm.num[url,uuid]

 VMware ハイパーバイザー上の仮想マシンの数。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

vmware.rp.cpu.usage[url,rpid]

 VMware リソースプールの間隔中の CPU 使用率 (ヘルツ単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **rpid** - VMware リソースプール ID

vmware.rp.memory[url,rpid,<mode>]

 VMware リソースプールのメモリメトリック。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **rpid** - VMware リソースプール ID
- **mode** - 可能な値:
consumed (デフォルト) - ゲスト物理メモリページのバックアップに消費されたホスト物理メモリの量
 ballooned - ゲスト内のバルーンドライバによって仮想マシンから回収されたゲスト物理メモリの量
overhead - 仮想マシンを実行するために ESXi データ構造によって消費されたホスト物理メモリの量

vmware.alarms.geturl

 VMware 仮想センターのアラームデータ。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL。

vmware.vm.alarms.get[url,uuid]

 VMware 仮想マシンのアラームデータ。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.attribute[url,uuid,name]

 VMware 仮想マシンのカスタム属性値。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **name** - カスタム属性名

vmware.vm.cluster.name[url,uuid]

 VMware 仮想マシン名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.consolidationneeded[url,uuid]

 VMware 仮想マシンディスクの統合が必要です。
 戻り値: *String*: *true* - 統合が必要; *false* - 統合が不要。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.latency[url,uuid]

 仮想マシンが物理 CPU へのアクセスを競合しているために実行できない時間の割合（パーセンテージ）。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.num[url,uuid]

 VMware 仮想マシンのプロセッサ数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.readiness[url,uuid,<instance>]

 仮想マシンが実行する準備ができていたが、物理 CPU でスケジュールできなかった時間の割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - CPU インスタンス

vmware.vm.cpu.ready[url,uuid]

 仮想マシンが実行の準備ができていたが、物理 CPU でスケジュールできなかった時間（ミリ秒単位）。CPU レディタイムは、ホスト上の仮想マシンの数とその CPU 負荷（%）に依存します。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 スワップインを待っている CPU 時間の割合。
 戻り値：*Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - CPU インスタンス

vmware.vm.cpu.usage[url,uuid]

 VMware 仮想マシンのプロセッサ使用率 (Hz)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.cpu.usage.perf[url,uuid]

 インターバル中の VMware 仮想マシンのプロセッサ使用率（パーセント）。
 戻り値：*Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.datacenter.name[url,uuid]

 VMware 仮想マシンのデータセンター名。
 戻り値: *String*。

パラメーター:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.discoveryurl

 VMware 仮想マシンのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.vm.guest.memory.size.swapped[url,uuid]

 スワップ領域にスワップアウトされたゲスト物理メモリの量 (KB)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.guest.osuptime[url,uuid]

 最後のオペレーティングシステムの起動から経過した合計時間（秒単位）。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.hv.maintenance[url,uuid]

 VMware 仮想マシンのハイパーバイザーのメンテナンス状態。
 戻り値: *0* - メンテナンス中でない; *1* - メンテナンス中。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.hv.name[url,uuid]

 VMware 仮想マシンのハイパーバイザー名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size[url,uuid]

 VMware 仮想マシンの合計メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.ballooned[url,uuid]

 VMware 仮想マシンのバルーンメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.compressed[url,uuid]

 VMware 仮想マシンの圧縮メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.consumed[url,uuid]

 ゲスト物理メモリページのバックアップに消費されたホスト物理メモリの量 (KB)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL

- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.private[url,uuid]

 VMware 仮想マシンのプライベートメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.shared[url,uuid]

 VMware 仮想マシンの共有メモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.swapped[url,uuid]

 VMware 仮想マシンのスワップメモリサイズ (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.usage.guest[url,uuid]

 VMware 仮想マシンのゲストメモリ使用量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.size.usage.host[url,uuid]

 VMware 仮想マシンのホストメモリ使用量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.memory.usage[url,uuid]

 消費されたホスト物理メモリの割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.net.if.discovery[url,uuid]

 VMware 仮想マシンのネットワークインターフェースのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 VMware 仮想マシンのネットワークインターフェースの入力統計情報 (1 秒あたりのバイト数/パケット数)。
 戻り値: 整数[2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ネットワークインターフェースのインスタンス
- **mode** - *bps* (デフォルト) または *pps* - 1 秒あたりのバイト数またはパケット数

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 VMware 仮想マシンのネットワークインターフェースの出力統計情報 (1 秒あたりのバイト数/パケット数)。
 戻り値: 整数[2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ネットワークインターフェースのインスタンス
- **mode** - *bps* (デフォルト) または *pps* - 1 秒あたりのバイト数またはパケット数

vmware.vm.net.if.usage[url,uuid,<instance>]

 インターバル中の VMware 仮想マシンのネットワーク使用率（送信レートと受信レートの合計）(KBps)。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ネットワークインターフェースのインスタンス

vmware.vm.perfcounter[url,uuid,path,<instance>]

 VMware 仮想マシンのパフォーマンスカウンター値。
 戻り値: 整数[2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **path** - パフォーマンスカウンターのパス[1]
- **instance** - パフォーマンスカウンターのインスタンス。集計値の場合は空のインスタンスを使用します (デフォルト)。

vmware.vm.powerstate[url,uuid]

 VMware 仮想マシンの電源状態。
 戻り値: *0* - poweredOff; *1* - poweredOn; *2* - suspended。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.property[url,uuid,prop]

 VMware 仮想マシンのプロパティ。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **prop** - プロパティパス

vmware.vm.snapshot.get[url,uuid]

 VMware 仮想マシンのスナップショット状態。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.state[url,uuid]

 VMware 仮想マシンの状態。
 戻り値：*String*：*notRunning*、*resetting*、*running*、*shuttingDown*、*standby*、または *unknown*。

パラメータ：

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.committed[url,uuid]

 VMware 仮想マシンのコミット済みストレージ容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.readoio[url,uuid,instance]

 収集間隔中に仮想ディスクへの未処理の読み取り要求の平均数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 仮想ディスクからの読み取りにかかる平均時間（ミリ秒）。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 仮想ディスクへの書き込みにかかる平均時間（ミリ秒）。
 戻り値：整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスインスタンス

vmware.vm.storage.uncommitted[url,uuid]

 VMware 仮想マシンの未コミットストレージ容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.unshared[url,uuid]

 VMware 仮想マシンの未共有ストレージ容量 (バイト単位)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.storage.writeoio[url,uuid,instance]

 収集間隔中に仮想ディスクへの未処理の書き込み要求の平均数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスインスタンス

vmware.vm.tags.get[url,uuid]

 VMware 仮想マシンのタグ配列。
 戻り値: JSON object。

パラメーター:

- **url** - VMware サービス URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

このアイテムは vSphere 6.5 以降で動作します。

vmware.vm.tools[url,uuid,<mode>]

 VMware 仮想マシンのゲストツールの状態。
 戻り値: *String*。
 status モードの場合: *guestToolsExecutingScripts* - VMware Tools が起動中; *guestToolsNotRunning* - VMware Tools が実行されていない; *guestToolsRunning* - VMware Tools が実行されている。
 version モードの場合: バージョン。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

- **mode** - *version*、*status* (デフォルト)

vmware.vm.uptime[url,uuid]

 VMware 仮想マシンの稼働時間 (秒)。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.vfs.dev.discovery[url,uuid]

 VMware 仮想マシンのディスクデバイスのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 VMware 仮想マシンのディスクデバイスの読み取り統計情報 (1 秒あたりのバイト数/操作数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス
- **mode** - *bps* (デフォルト) または *ops* - 1 秒あたりのバイト数または操作数

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 VMware 仮想マシンのディスクデバイスの書き込み統計情報 (1 秒あたりのバイト数/操作数)。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **instance** - ディスクデバイスのインスタンス
- **mode** - *bps* (デフォルト) または *ops* - 1 秒あたりのバイト数または操作数

vmware.vm.vfs.fs.discovery[url,uuid]

 VMware 仮想マシンのファイルシステムのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子

このアイテムを動作させるには、ゲスト仮想マシンに VMware Tools がインストールされている必要があります。

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 VMware 仮想マシンのファイルシステム統計情報 (バイト/パーセンテージ)。
 戻り値: バイトの場合は整数、パーセンテージの場合は浮動小数点数。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **fsname** - ファイルシステム名
- **mode** - *total*、*free*、*used*、*pfree*、または *pused*

このアイテムを動作させるには、ゲスト仮想マシンに VMware Tools がインストールされている必要があります。

脚注

[1] VMware 用のカスタムパフォーマンスカウンター名の作成 を参照してください。

[2] これらのアイテムの値は VMware のパフォーマンスカウンターから取得され、VMwarePerfFrequency パラメーター は Zabbix VMware キャッシュ内のデータを更新するために使用されます。

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write
- vmware.hv.network.in

- vmware.hv.network.out
- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

詳細

Zabbix で VMware 環境を監視するための詳細な設定方法は、仮想マシン監視 を参照してください。

仮想マシンのディスカバリキーのフィールド

以下の表は、仮想マシン関連のディスカバリキーから返されるフィールドを示しています。

Item key		
Description	**Field**	**Retrieved content**
vmware.cluster.discovery		
クラスタのディスカバリを実行します。	{#CLUSTER.ID}	クラスタ識別子。
	{#CLUSTER.NAME}	クラスタ名。
	”resource_pool”	リソースプールのデータを含む配列。リソースグループ ID、tags 配列、リソースプールパス、仮想マシン数を含みます。 配列構造: [{ "rpid":"resource group id", "tags":[{}], "rpath":"resource group path", "vm_count":0 }]
	”tags”	"tags" 配列の構造については、”tags” フィールドを参照してください。 タグ名、説明、カテゴリを含むタグの配列。 配列構造: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.datastore.discovery		
データストアのディスカバリを実行します。	{#DATASTORE}	データストア名。
	{#DATASTORE.EXTENT}	データストアの extent パーティション ID とインスタンス名を含む配列。 配列構造: [{ "partitionid":1, "instance":"name" }]
	{#DATASTORE.TYPE}	データストアのタイプ。 値の例: VMFS, NFS, vsan など。
	{#DATASTORE.UUID}	データストア識別子。

Item key		
	”tags”	タグ名、説明、カテゴリを含むタグの配列。 配列構造: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dc.discovery		
データセンターのディスカバリを実行します。	{#DATACENTER}	データセンター名。
	{#DATACENTERID}	データセンター識別子。
	”tags”	タグ名、説明、カテゴリを含むタグの配列。 配列構造: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dvswitch.discovery		
vSphere 分散スイッチのディスカバリを実行します。	{#DVS.NAME}	スイッチ名。
	{#DVS.UUID}	スイッチ識別子。
vmware.hv.discovery		
ハイパーバイザーのディスカバリを実行します。	{#HV.UUID}	一意のハイパーバイザー識別子。
	{#HV.ID}	ハイパーバイザー識別子（HostSystem managed object name）。
	{#HV.NAME}	ハイパーバイザー名。
	{#HV.NETNAME}	ハイパーバイザーのネットワークホスト名。
	{#HV.IP}	ハイパーバイザーの IP アドレス。空の場合があります。 複数のネットワークインターフェースを持つ HA 構成では、インターフェースの選択優先順位は次のとおりです。 - vCenter IP と同じ IP サブネットを共有する IP を優先します。 - デフォルトゲートウェイを持つ IP サブネットの IP を優先します。 - ID が最も小さいインターフェースの IP を優先します。
	{#CLUSTER.NAME}	クラスタ名。空の場合があります。
	{#DATACENTER.NAME}	データセンター名。
	{#PARENT.NAME}	ハイパーバイザーを格納するコンテナの名前。
	{#PARENT.TYPE}	ハイパーバイザーが格納されるコンテナのタイプ。値は `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware` で、”VMware” は不明なコンテナタイプを表します。
	”resource_pool”	リソースプールのデータを含む配列。リソースグループ ID、tags 配列、リソースプールパス、仮想マシン数を含みます。 配列構造: [{ "rpid":"resource group id", "tags":[{}], "rpath":"resource group path", "vm_count":0 }] "tags" 配列の構造については、”tags” フィールドを参照してください。

Item key		
	”tags”	タグ名、説明、カテゴリを含むタグの配列。 配列構造: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.hv.datastore.discovery		
ハイパーバイザーのデータストアのディスカバリを実行します。複数のハイパーバイザーが同じデータストアを使用できることに注意してください。	{#DATASTORE}	データストア名。
	{#DATASTORE.TYPE}	データストアのタイプ。 値の例: VMFS, NFS, vsan など。
	{#DATASTORE.UUID}	データストア識別子。
	{#MULTIPATH.COUNT}	登録済みのデータストアパス数。
	{#MULTIPATH.PARTITION.COUNT}	利用可能なディスクパーティション数。
	”datastore_extent”	データストア extent のインスタンス名とパーティション ID を含む配列。 配列構造: [{ "partitionid":1, "instance":"name" }]
	”tags”	タグ名、説明、カテゴリを含むタグの配列。 配列構造: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.hv.net.if.discovery		
ハイパーバイザーのネットワークインターフェースのディスカバリを実行します。	{#IFNAME}	インターフェース名。
	{#IFDRIVER}	インターフェースドライバー。
	{#IFDUPLEX}	インターフェースのデュプレックス設定。
	{#IFSPEED}	インターフェース速度。
	{#IFMAC}	インターフェースの MAC アドレス。
vmware.vm.discovery		
仮想マシンのディスカバリを実行します。	{#VM.UUID}	一意の仮想マシン識別子。
	{#VM.ID}	仮想マシン識別子（VirtualMachine managed object name）。
	{#VM.NAME}	仮想マシン名。
	{#HV.NAME}	ハイパーバイザー名。
	{#HV.UUID}	一意のハイパーバイザー識別子。
	{#HV.ID}	ハイパーバイザー識別子（HostSystem managed object name）。
	{#CLUSTER.NAME}	クラスタ名。空の場合があります。
	{#DATACENTER.NAME}	データセンター名。
	{#DATASTORE.NAME}	データストア名。
	{#DATASTORE.UUID}	データストア識別子。
	{#VM.IP}	仮想マシンの IP アドレス。空の場合があります。
	{#VM.DNS}	仮想マシンの DNS 名。空の場合があります。
	{#VM.GUESTFAMILY}	ゲスト仮想マシン OS のファミリー。空の場合があります。

Item key		
	{#VM.GUESTFULLNAME}	ゲスト仮想マシン OS の完全名。空の場合があります。
	{#VM.FOLDER}	仮想マシンの親フォルダの連鎖。ネストされたグループの値として使用できます。フォルダ名は”/” で連結されます。空の場合があります。
	{#VM.TOOLS.STATUS}	VMware 仮想マシンツールの状態。
	{#VM.POWERSTATE}	VMware 仮想マシンの電源状態（poweredOff, poweredOn, suspended）。
	{#VM.RPOOL.ID}	リソースプール識別子。
	{#VM.RPOOL.PATH}	”root” 名の”Resources” を除いた完全なリソースプールパス。フォルダ名は”/” で連結されます。
	{#VM.SNAPSHOT.COUNT}	VM スナップショット数。
	”tags”	タグ名、説明、カテゴリを含むタグの配列。 配列構造: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
	”vm_customattribute”	仮想マシンのカスタム属性の配列（定義されている場合）。 配列構造: [{ "name":"custom field name", "value":"custom field value" }]
	”net_if”	仮想マシンのネットワークインターフェースの配列。 配列構造: [{ "ifname": "interface name", "ifdesc": "interface description", "ifmac": "00:00:00:00:00:00", "ifconnected": true, "iftype": "interface type", "ifbackingdevice": "interface backing device", "ifdvswitch_uuid": "interface switch uuid", "ifdvswitch_portgroup": "interface switch port group", "ifdvswitch_port": "interface switch port", "ifip": ["interface ip addresses"] }] 返されるデータの説明については、”vmware.vm.net.if.discovery” アイテムキーを参照してください。
vmware.vm.net.if.discovery		
仮想マシンのネットワークインターフェースのディスカバリを実行します。	{#IFNAME}	ネットワークインターフェース名。
	{#IFDESC}	インターフェースの説明。
	{#IFMAC}	インターフェースの MAC アドレス。
	{#IFCONNECTED}	インターフェースの接続状態（false - 切断、true - 接続）。
	{#IFTYPE}	インターフェースタイプ。
	{#IFBACKINGDEVICE}	バックエンドデバイス名。
	{#IFDVSWITCH.UUID}	vSphere Distributed Switch の一意の識別子。
	{#IFDVSWITCH.PORTGROUP}	分散ポートグループ。
	{#IFDVSWITCH.PORT}	vSphere Distributed Switch のポート。
	”ifip”	インターフェースアドレスの配列。
vmware.vm.vfs.dev.discovery		
仮想マシンのディスクデバイスのディスカバリを実行します。	{#DISKNAME}	ディスクデバイス名。

Item key			
vmware.vm.vfs.fs.discovery 仮想マシンのファイルシステムのディスカバリを実行します。	{#FSNAME}	ファイルシステム名。	

VMware アイテムの **JSON** 例

概要　このセクションでは、さまざまな VMware アイテムによって返される JSON オブジェクトに関する追加情報を提供します。

vmware.*.alarms.get　アイテム **vmware.alarms.get[]**、**vmware.cluster.alarms.get[]**、**vmware.datastore.alarms.get[]**、**vmware.dc.alarms.get[]**、**vmware.hv.alarms.get[]**、**vmware.vm.alarms.get[]** は、次の構造を持つ JSON オブジェクトを返します（値は例です）。

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get　**vmware.cluster.tags.get[]**、**vmware.datastore.tags.get[]**、**vmware.dc.tags.get[]**、**vmware.hv.tags.get[]**、**vmware.vm.tags.get[]** のアイテムは、以下の構造を持つ JSON オブジェクトを返します（値は例として示しています）：

```
{
    "tags": [
```

```
        {
            "name": "Windows",
        "description": "tag for cat OS type",
        "category": "OS type"
    },
    {
            "name": "SQL Server",
        "description": "tag for cat application name",
        "category": "application name"
        },
    {
            // other tags
        }
    ]
}
```

vmware.hv.diskinfo.get アイテム **vmware.hv.diskinfo.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例です）：

```
[
  {
    "instance": "mpx.vmhba32:C0:T0:L0",
    "hv_uuid": "8002299e-d7b9-8728-d224-76004bbb6100",
    "datastore_uuid": "",
    "operational_state": [
      "ok"
    ],
    "lun_type": "disk",
    "queue_depth": 1,
    "model": "USB DISK",
    "vendor": "SMI Corp",
    "revision": "1100",
    "serial_number": "CCYYMMDDHHmmSS9S62CK",
    "vsan": {}
  },
  {
    // 他のインスタンス
  }
]
```

vmware.dvswitch.fetchports.get アイテム **vmware.dvswitch.fetchports.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例として提供されています）：

```
{
    "FetchDVPortsResponse":
    {
        "returnval": [
            {
                "key": "0",
                "dvsUuid": "50 36 6a 24 25 c0 10 9e-05 4a f6 ea 4e 3d 09 88",
                "portgroupKey": "dvportgroup-2023",
                "proxyHost":
                {
                    "@type": "HostSystem",
                    "#text": "host-2021"
                },
                "connectee":
                {
                    "connectedEntity":
                    {
                        "@type": "HostSystem",
                        "#text": "host-2021"
                    },
```

```
                    "nicKey": "vmnic0",
                    "type": "pnic"
                },
                "conflict": "false",
                "state":
                {
                    "runtimeInfo":
                    {
                        "linkUp": "true",
                        "blocked": "false",
                        "vlanIds":
                        {
                            "start": "0",
                            "end": "4094"
                        },
                        "trunkingMode": "true",
                        "linkPeer": "vmnic0",
                        "macAddress": "00:00:00:00:00:00",
                        "statusDetail": null,
                        "vmDirectPathGen2Active": "false",
                        "vmDirectPathGen2InactiveReasonOther": "portNptIncompatibleConnectee"
                    },
                    "stats":
                    {
                        "packetsInMulticast": "2385470",
                        "packetsOutMulticast": "45",
                        "bytesInMulticast": "309250248",
                        "bytesOutMulticast": "5890",
                        "packetsInUnicast": "155601537",
                        "packetsOutUnicast": "113008658",
                        "bytesInUnicast": "121609489384",
                        "bytesOutUnicast": "47240279759",
                        "packetsInBroadcast": "1040420",
                        "packetsOutBroadcast": "7051",
                        "bytesInBroadcast": "77339771",
                        "bytesOutBroadcast": "430392",
                        "packetsInDropped": "0",
                        "packetsOutDropped": "0",
                        "packetsInException": "0",
                        "packetsOutException": "0"
                    }
                },
                "connectionCookie": "1702765133",
                "lastStatusChange": "2022-03-25T14:01:11Z",
                "hostLocalPort": "false"
            },
            {
                //other keys
            }
        ]
    }
}
```

vmware.hv.hw.sensors.get アイテム **vmware.hv.hw.sensors.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例として提供されています）：

```
{
    "val":
        {
            "@type": "HostHardwareStatusInfo",
            "storageStatusInfo": [
                {
                    "name": "Intel Corporation HD Graphics 630 #2",
```

```
                "status":
                        {
                            "label": "Unknown",
                        "summary": "Cannot report on the current status of the physical element",
                        "key": "Unknown"
                    }
            },
            {
                    "name": "Intel Corporation 200 Series/Z370 Chipset Family USB 3.0 xHCI Controller #20"
                "status":
                        {
                     "label": "Unknown",
                     "summary": "Cannot report on the current status of the physical element",
                     "key": "Unknown"
                 }
            },
            {
                    // other hv hw sensors
                }
        ]
        }
}
```

vmware.hv.sensors.get アイテム **vmware.hv.sensors.get[]** は、以下の構造を持つ JSON オブジェクトを返します（値は例です）：

```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10500",
                "unitModifier": "-2",
                "baseUnits": "Watts",
                "sensorType": "other"
            },
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "Power Supply 1 PS 1 Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                    "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10000",
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                    // other hv sensors
                }
        ]
        }
```

```
}
```

vmware.vm.snapshot.get スナップショットが存在する場合、アイテム **vmware.snapshot.get[]** は以下の構造の JSON オブジェクトを返します（値は例です）：

```
{
  "snapshot": [
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:16:39 AM",
      "description": "Descr 1",
      "createtime": "2022-04-01T06:16:51.761Z",
      "size": 5755795171,
      "uniquesize": 5755795171
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:18:21 AM",
      "description": "Descr 2",
      "createtime": "2022-04-01T06:18:29.164999Z",
      "size": 118650595,
      "uniquesize": 118650595
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:37:29 AM",
      "description": "Descr 3",
      "createtime": "2022-04-01T06:37:53.534999Z",
      "size": 62935016,
      "uniquesize": 62935016
    }
  ],
  "count": 3,
  "latestdate": "2022-04-01T06:37:53.534999Z",
  "latestage": 22729203,
  "oldestdate": "2022-04-01T06:16:51.761Z",
  "oldestage": 22730465,
  "size": 5937380782,
  "uniquesize": 5937380782
}
```

スナップショットが存在しない場合、アイテム **vmware.snapshot.get[]** は空の値を持つ JSON オブジェクトを返します:

```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```

VMware 監視設定の例

概要

次の例では、VMware 仮想マシンの監視のために Zabbix をセットアップする方法について説明します。これには以下が含まれます。

- VMware 環境を表すホストの作成
- VMware 環境内の仮想マシンを検出するローレベルディスカバリルールの作成
- ローレベルディスカバリルールによって検出された仮想マシンのために、Zabbix が実際のホストを生成するためのホストプロトタイプの作成

前提条件

Note:
この例では VMware の設定は扱いません。VMware はすでに設定済みであることを前提としています。

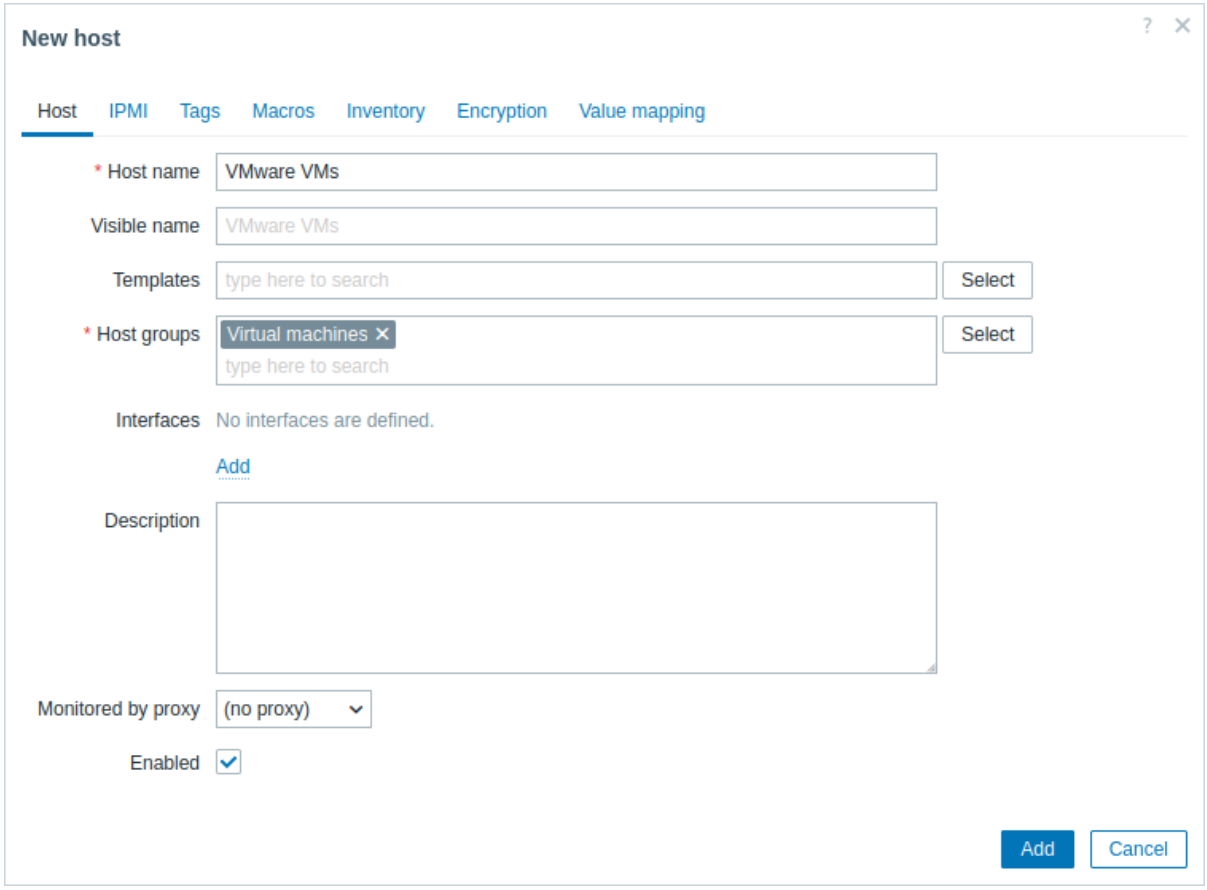
続行する前に、Zabbix サーバー設定ファイルで`StartVMwareCollectors` パラメータを 2 以上 に設定してください（デフォルト値は 0 です）。

ホストの作成

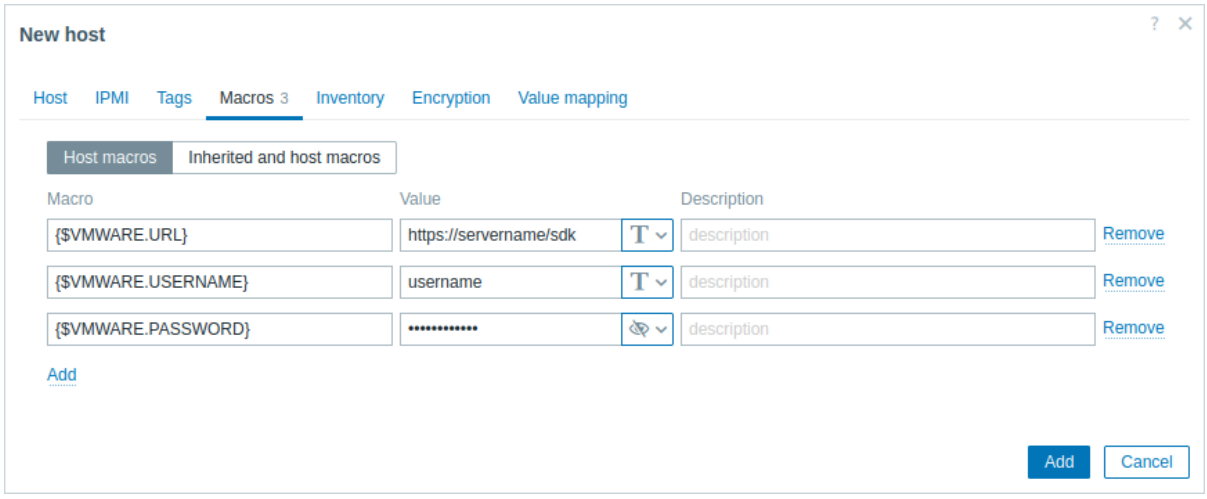
1. データ収集 > ホスト に移動します。

2. ホストを作成 します。

- ホスト名フィールドに、ホスト名を入力します（例: ”VMware VMs”）。
- ホストグループフィールドに、ホストグループを入力または選択します（例: ”Virtual machines”）。



- マクロタブで、次のホストマクロを設定します。
  - {$VMWARE.URL} - VMware サービス（ESXi ハイパーバイザー）SDK URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - VMware サービスのユーザー名
  - {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーパスワード



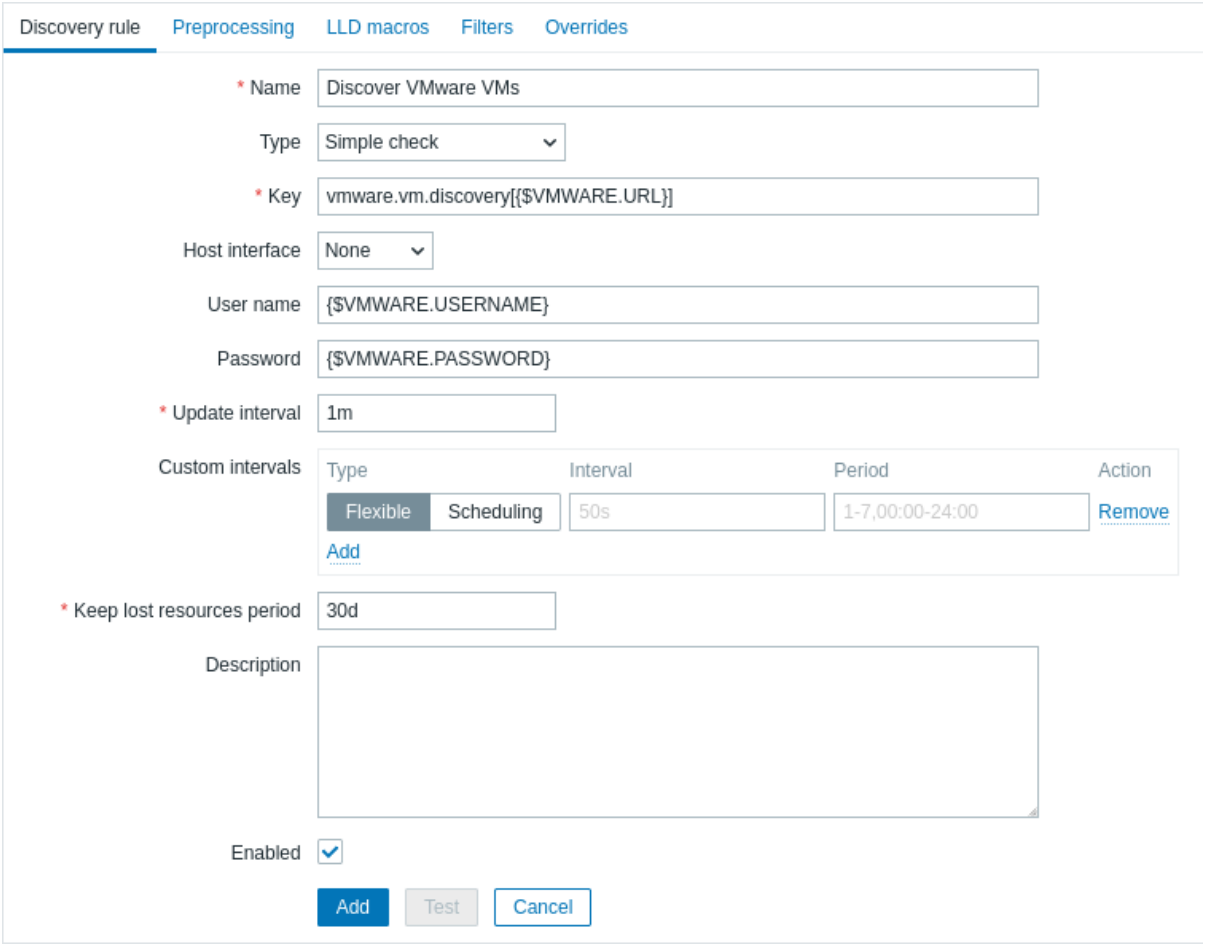
3. 追加ボタンをクリックしてホストを作成します。このホストは、VMware 環境を表します。

ローレベルディスカバリルールの作成

1. 作成したホストの *Discovery* をクリックして、そのホストのローレベルディスカバリルールの一覧に移動します。

2. 作成 でローレベルディスカバリルールを作成します。

- *Name* フィールドに、ローレベルディスカバリルール名を入力します（例: ”VMware VM を検出”）。
- *Type* フィールドで、”Simple check” を選択します。
- *Key* フィールドに、VMware 仮想マシンを検出するための組み込みアイテムキー`vmware.vm.discovery[{$VMWARE.URL}]`を入力します。
- *User name* フィールドと *Password* フィールドに、事前にホストで設定した対応するマクロを入力します。



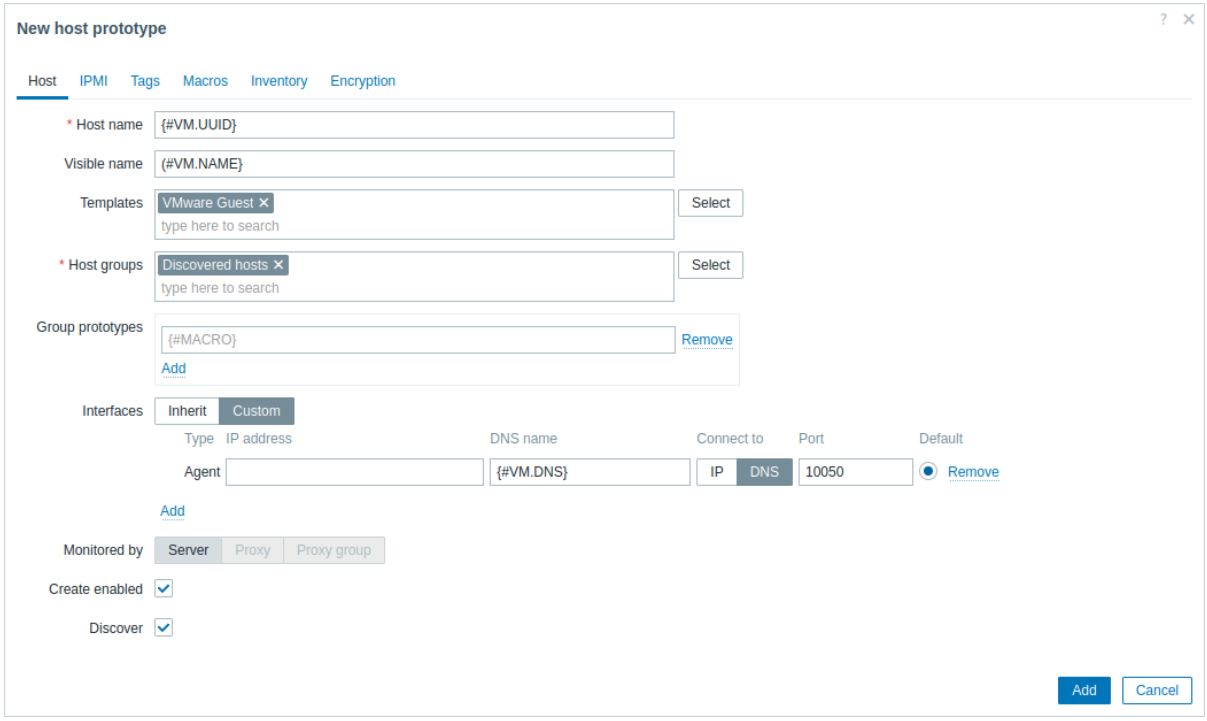
3. *Add* ボタンをクリックして、ローレベルディスカバリルールを作成します。
このディスカバリルールにより、VMware 環境内の仮想マシンが検出されます。

ホストプロトタイプの作成

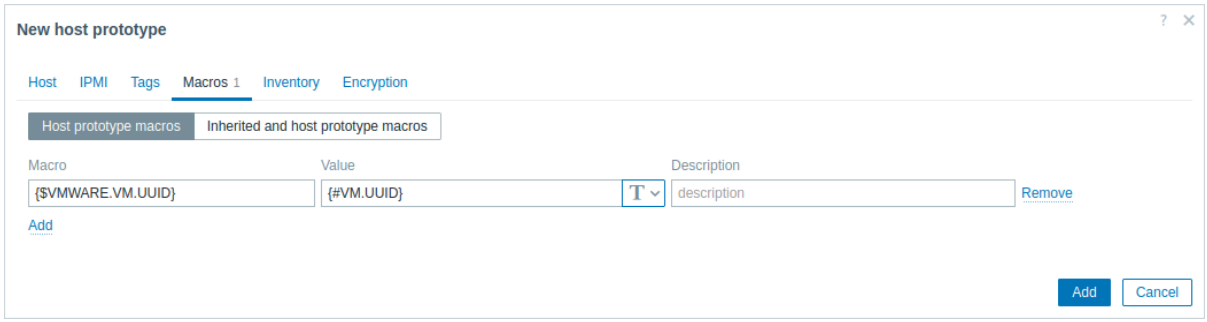
1. 低レベルディスカバリルールの一覧で、先ほど作成した低レベルディスカバリルールのホストプロトタイプをクリックします。

2. ホストプロトタイプを作成します。ホストプロトタイプは低レベルディスカバリルールによってホストを作成するための設計図であるため、ほとんどのフィールドには低レベルディスカバリマクロが含まれます。これにより、先ほど作成した低レベルディスカバリルールで取得した内容に基づいてホストが作成されます。

- ホスト名フィールドには、マクロ {#VM.UUID} を入力します。
- 表示名フィールドには、マクロ {#VM.NAME} を入力します。
- テンプレートフィールドには、「VMware Guest」テンプレートを入力または選択します。このテンプレートには、仮想マシンの電源状態、CPU 使用率、メモリ使用率、ネットワークデバイスなどを監視するためのVMware アイテムおよびディスカバリルールが含まれています。
- ホストグループフィールドには、ホストグループ (例:「Discovered hosts」) を入力または選択します。
- インターフェースフィールドには、カスタムホストインターフェースを追加します。その後、*DNS* 名フィールドにマクロ {#VM.DNS} を入力するか、*IP* アドレスフィールドにマクロ {#VM.IP} を入力します。または、VMware 環境の仮想マシンに複数のインターフェースがある場合は、高度なホストインターフェースの設定セクションに進んでください。カスタムホストインターフェースの設定は、*VMware Guest* テンプレートを正しく機能させるために必要です。



- マクロタブで、{$VMWARE.VM.UUID} マクロに値 {#VM.UUID} を設定します。これは、このマクロをアイテムパラメータのホストレベルのユーザーマクロとして使用する *VMware Guest* テンプレートを正しく機能させるために必要です (例: *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*)。

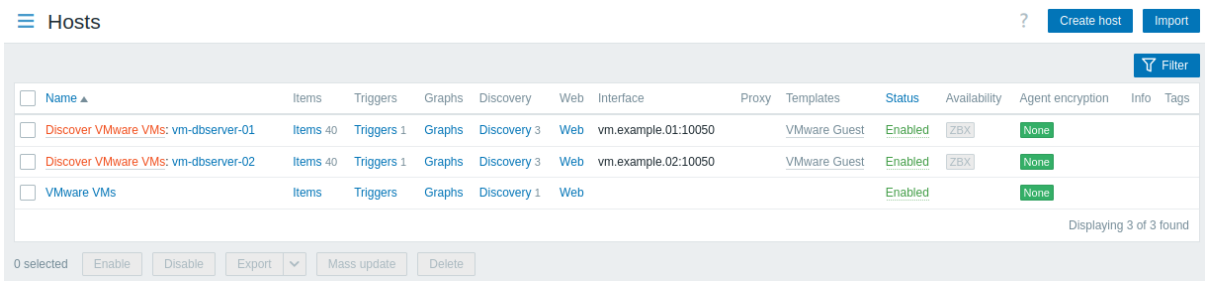


3. 追加ボタンをクリックしてホストプロトタイプを作成します。このホストプロトタイプは、先ほど作成した低レベルディスカバリルールで検出された仮想マシンのホストを作成するために使用されます。

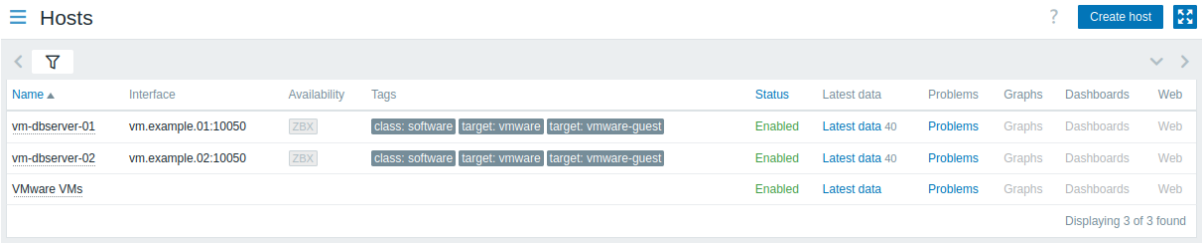
ホストとメトリクスを表示する

ホストプロトタイプが作成されると、低レベルディスカバリルールによって検出された VMware 仮想マシンのホストが作成され、Zabbix はそれらの監視を開始します。
必要に応じて、ホストのディスカバリと作成は手動で実行することもできます。

作成されたホストを表示するには、データ収集 > ホストメニューセクションに移動します。



収集されたメトリクスを表示するには、監視 > ホストメニューセクションに移動し、いずれかのホストの 最新データをクリックします。



高度なホストインターフェースの設定

「低レベルディスカバリルールの作成」セクションで設定した `vmware.vm.discovery[{$VMWARE.URL}]` アイテムキーは、"net_if" フィールドでネットワークインターフェースのデータを返します。

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

このデータを使用してカスタムホストインターフェースを設定できます。

1. 低レベルディスカバリルールを作成する際に、追加で低レベルディスカバリマクロを設定します。*LLD* マクロタブで、JSONPath値を持つカスタム LLD マクロを作成します。例:

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`

2. ホストプロトタイプを作成する際に、カスタムホストインターフェースを追加し、*DNS* 名または *IP* アドレスフィールドに LLD マクロを入力します。

Web 監視

概要　Zabbix を使用すると、Web サイトの可用性に関するいくつかの側面を確認できます。

> **Attention:**
> Web 監視を行うには、Zabbix サーバーを最初に cURL (libcurl) サポート付きで設定する必要があります。

Web 監視を有効にするには、Web シナリオを定義する必要があります。Web シナリオは、1 つ以上の HTTP リクエスト、つまり「ステップ」から構成されます。ステップは、あらかじめ定義された順序で Zabbix サーバーによって定期的に実行されます。ホストがプ

ロキシによって監視されている場合、ステップはプロキシによって実行されます。

Web シナリオは、アイテム、トリガーなどと同様にホスト/テンプレートに関連付けられます。つまり、Web シナリオはテンプレートレベルで作成し、1 回の操作で複数のホストに適用することもできます。

任意の Web シナリオでは、次の情報が収集されます。

- シナリオ全体の全ステップにおける 1 秒あたりの平均ダウンロード速度
- 失敗したステップ番号
- 最後のエラーメッセージ

任意の Web シナリオの各ステップでは、次の情報が収集されます。

- 1 秒あたりのダウンロード速度
- 応答時間
- 応答コード

詳細については、Web 監視アイテムを参照してください。

Web シナリオの実行によって収集されたデータはデータベースに保存されます。このデータは、グラフ、トリガー、および通知に自動的に使用されます。

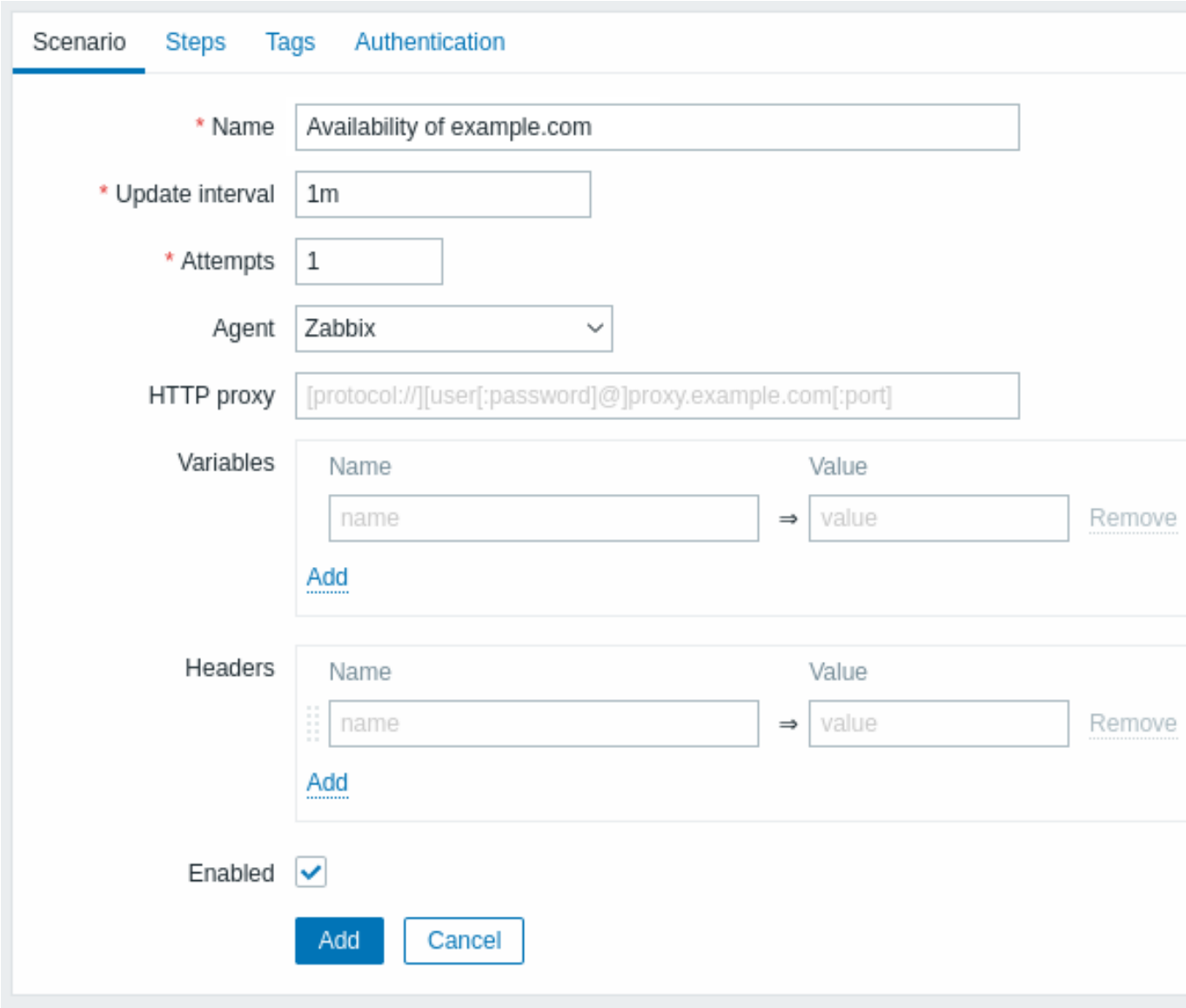
Zabbix は、取得した HTML ページにあらかじめ定義された文字列が含まれているかどうかも確認できます。また、ログインをシミュレートし、ページ上でマウスクリックを模擬した一連の操作を実行することもできます。

Zabbix の Web 監視は HTTP と HTTPS の両方をサポートしています。Web シナリオの実行時には、必要に応じてリダイレクトを追跡します（下記のオプション *Follow redirects* を参照）。リダイレクトの最大数は 10 に固定されています（cURL オプション CURLOPT_MAXREDIRS を使用）。すべての cookie は、1 つのシナリオの実行中に保持されます。

Web シナリオの設定　Web シナリオを設定するには、次の手順を実行します。

1. データ収集 > ホスト（またはテンプレート）に移動します。
2. ホスト/テンプレートの行にある *Web* をクリックします。
3. 右側の *Web* シナリオの作成をクリックします（既存のシナリオを編集する場合は、シナリオ名をクリックします）。
4. フォームにシナリオのパラメータを入力します。

シナリオタブでは、Web シナリオの一般的なパラメータを設定できます。



必須入力フィールドには赤いアスタリスクが付いています。

シナリオのパラメータ:

パラメータ	説明
名前	一意のシナリオ名。 ユーザーマクロがサポートされています。注：ユーザーマクロを使用した場合、これらのマクロはWeb 監視アイテム名内で未解決のままになります。
更新間隔	シナリオを実行する頻度。 時間のサフィックスがサポートされています（例: 30s、1m、2h、1d）。 ユーザーマクロがサポートされています。注：ユーザーマクロを使用し、その値を変更した場合（例: 5m → 30s）、次回のチェックは以前の値に従って実行されます（この例では、より後の時刻に実行されます）。 新しい Web シナリオは、作成後 60 秒以内にチェックされます。
試行回数	Web シナリオのステップを実行する試行回数。ネットワーク障害（タイムアウト、接続不可など）が発生した場合、Zabbix はステップの実行を数回繰り返すことができます。設定した値は、シナリオの各ステップに同じように適用されます。最大 10 回まで指定でき、デフォルト値は 1 です。 注：誤ったレスポンスコードや、必要な文字列との不一致を理由に、Zabbix がステップを再実行することはありません。
エージェント	クライアントエージェントを選択します。 Zabbix は、選択したブラウザーとして動作します。Web サイトがブラウザーごとに異なるコンテンツを返す場合に便利です。 このフィールドではユーザーマクロを使用できます。

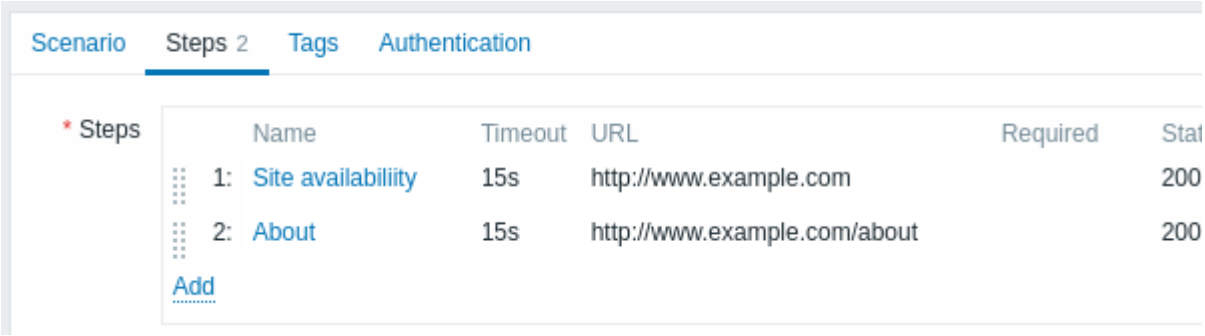
パラメータ	説明
HTTP プロキシ	`[protocol://][username[:password]@]proxy.example.com[:port]` の形式で、使用する HTTP プロキシを指定できます。 これにより、CURLOPT_PROXY cURL オプションが設定されます。 オプションの `protocol://`プレフィックスを使用して、代替プロキシプロトコルを指定できます (プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました)。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。 デフォルトでは、ポート 1080 が使用されます。 指定した場合、プロキシは http_proxy、HTTPS_PROXY などのプロキシ関連の環境変数を上書きします。指定しない場合、プロキシはプロキシ関連の環境変数を上書きしません。入力した値は「そのまま」渡され、妥当性のチェックは行われません。 SOCKS プロキシのアドレスも入力できます。誤ったプロトコルを指定すると接続に失敗し、アイテムはサポートされていない状態になります。 注：HTTP プロキシでサポートされるのは、単純な認証のみです。 このフィールドではユーザーマクロを使用できます。
変数	シナリオのステップで使用できる変数（URL、POST 変数）。 次の形式で指定します。 **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()}（マクロ関数を参照） 例: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} その後、ステップ内でマクロを {username}、{password}、{hostid} などとして参照できます。Zabbix はこれらを実際の値に自動的に置き換えます。`regex:` を含む変数では、正規表現の値を取得するために 1 つのステップが必要です。そのため、抽出された値は次のステップ以降でのみ適用できます。 値の部分が `regex:` で始まる場合、その後の部分は Web ページを検索する正規表現として扱われ、見つかった場合は一致した文字列が変数に保存されます。一致した値を抽出するには、少なくとも 1 つのサブグループが必要です。 ユーザーマクロおよび `{HOST.*}`マクロがサポートされています。 変数をクエリフィールドまたは POST 変数のフォームデータで使用する場合、変数は自動的に URL エンコードされます。ただし、raw POST または URL 内で直接使用する場合は、手動で URL エンコードする必要があります。
ヘッダー	HTTP ヘッダーは、リクエストの実行時に使用されます。デフォルトヘッダーとカスタムヘッダーを使用できます。 ヘッダーは、シナリオレベルのドロップダウンリストから選択したエージェントの種類に応じたデフォルト設定を使用して割り当てられ、ステップレベルでカスタム定義されていない限り、すべてのステップに適用されます。 ステップレベルでヘッダーを定義すると、シナリオレベルのドロップダウンリストから「**User-Agent**」を選択して割り当てられたデフォルトヘッダーを除き、それまでに定義されたすべてのヘッダーが自動的に破棄されることに注意してください。 ただし、「User-Agent」デフォルトヘッダーも、ステップレベルで指定することで上書きできます。 シナリオレベルでヘッダーを解除するには、ステップレベルでヘッダー名を指定し、値を指定せずに設定します。 ヘッダーは、HTTP プロトコルで表示される場合と同じ構文で一覧表示します。 CURLOPT_HTTPHEADER cURL オプションでサポートされる追加機能も使用できます。 例: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 ユーザーマクロおよび `{HOST.*}`マクロがサポートされています。
有効	このチェックボックスを選択するとシナリオが有効になり、選択しない場合は無効になります。

既存のシナリオを編集する場合、フォームには次の 2 つの追加ボタンが表示されます。

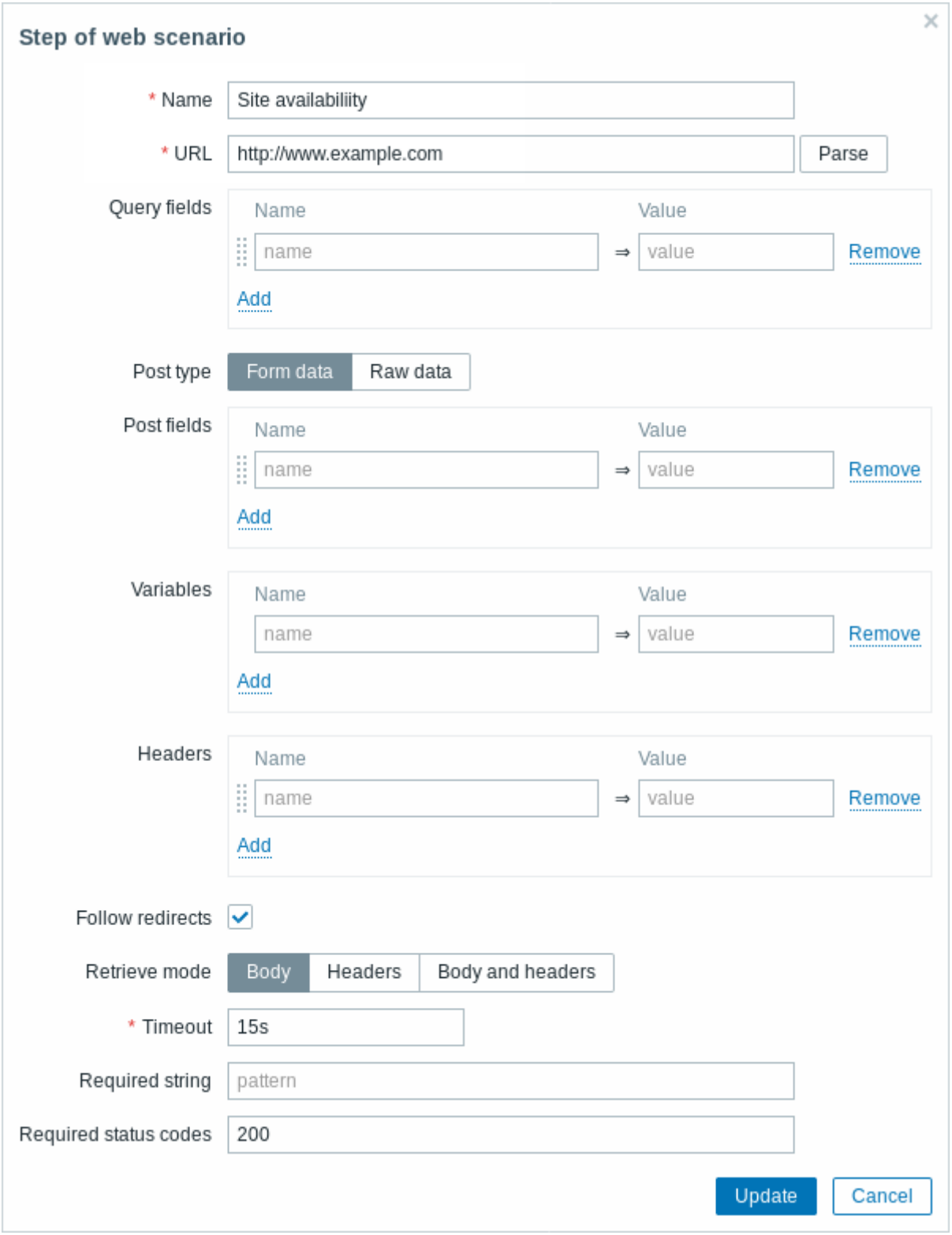
Clone	既存のシナリオのプロパティに基づいて、別のシナリオを作成します。
Clear history and trends	シナリオの履歴データとトレンドデータを削除します。データの削除後、サーバーはシナリオを直ち

Note:
HTTP プロキシフィールドを空のままにした場合、HTTP プロキシを使用する別の方法として、プロキシ関連の環境変数を設定できます。
HTTP チェックの場合は、Zabbix サーバーユーザーに対して `http_proxy` 環境変数を設定します。例: `http_proxy=http://proxy_ip:proxy_port`
HTTPS チェックの場合は、`HTTPS_PROXY` 環境変数を設定します。例: `HTTPS_PROXY=http://proxy_ip:proxy_port`。
詳細については、シェルコマンド `# man curl` を実行してください。

ステップタブでは、Web シナリオのステップを設定できます。Web シナリオのステップを追加するには、ステップブロックの追加をクリックします。



Note:
秘密情報を含むユーザーマクロを URL で使用しないでください。"******" に解決されます。



設定手順

ステップのパラメータ:

パラメータ	説明
名前	一意のステップ名。 ユーザーマクロがサポートされています。注：ユーザーマクロを使用した場合、これらのマクロはWeb 監視アイテム名で未解決のままになります。

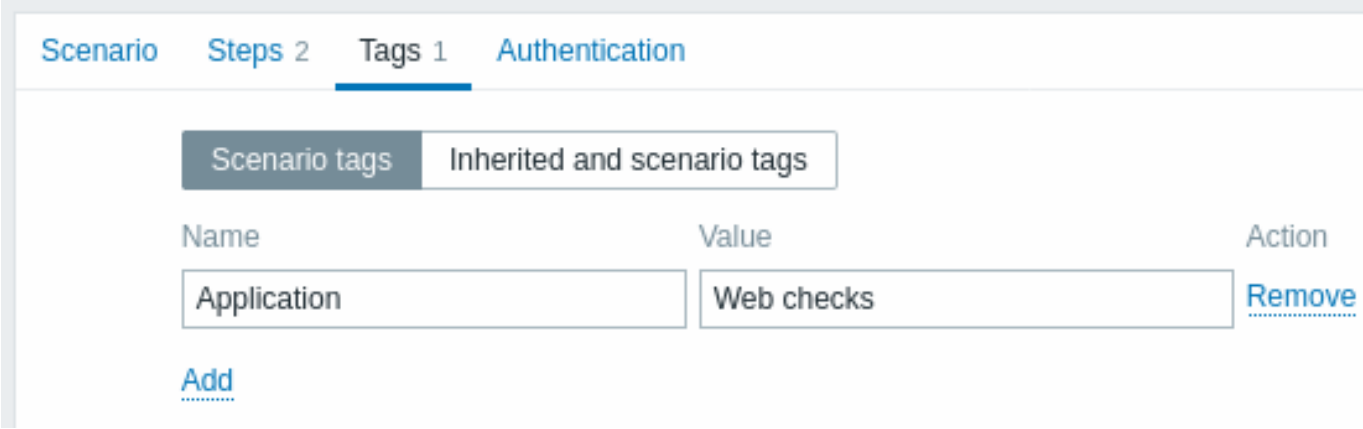
パラメータ	説明
URL	接続してデータを取得する URL。例: https://www.example.com http://www.example.com/download ドメイン名には Unicode 文字を指定できます。Web シナリオのステップ実行時に、自動的に ASCII の Punycode へ変換されます。 解析ボタンを使用すると、オプションのクエリフィールド（?key=value&limit=10 など）を URL から分離し、属性と値をクエリフィールドに移動して自動的に URL エンコードできます。 URL では、{macro} 構文を使用して変数を指定できます。変数は、{{macro}.urlencode()} 構文を使用して手動で URL エンコードできます。 ユーザーマクロおよび {HOST.*}マクロがサポートされています。 セキュリティ上の理由から、URL に実際のユーザー名やパスワードを含めないでください。認証情報が公開されるのを避けるため、代わりにユーザーマクロを使用してください。 最大 2048 文字です。
クエリフィールド	URL に対する HTTP GET 変数。 属性と値のペアとして指定します。 値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、または {HOST.*} マクロの値は、解決された後に自動的に URL エンコードされます。{{macro}.urlencode()} 構文を使用すると、二重に URL エンコードされます。 ユーザーマクロおよび {HOST.*}マクロがサポートされています。
Post	HTTP POST 変数。 フォームデータモードでは、属性と値のペアとして指定します。 値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、または {HOST.*} マクロの値は、解決された後に自動的に URL エンコードされます。 **Raw** データモードでは、属性と値が 1 行に表示され、**&** 記号で連結されます。 Raw 値は、{{macro}.urlencode()} または {{macro}.urldecode()} 構文を使用して手動で URL エンコードまたはデコードできます。 例: id=2345&userid={user} {user} が Web シナリオの変数として定義されている場合、ステップ実行時にその値へ置き換えられます。変数を URL エンコードする場合は、{user} を {{user}.urlencode()} に置き換えてください。 ユーザーマクロおよび {HOST.*}マクロがサポートされています。
変数	GET および POST 関数で使用できるステップレベルの変数。 属性と値のペアとして指定します。 ステップレベルの変数は、シナリオレベルの変数または前のステップの変数より優先されます。ただし、ステップレベルの変数の値は、現在のステップではなく、その次のステップにのみ影響します。 次の形式で指定します: **{macro}**=value **{macro}**=regex:<regular expression> 詳細については、シナリオレベルの変数の説明を参照してください。 変数は、クエリフィールドまたは POST 変数のフォームデータで使用する場合は自動的に URL エンコードされますが、Raw POST または URL で直接使用する場合は手動で URL エンコードする必要があります。
ヘッダー	リクエスト実行時に送信されるカスタム HTTP ヘッダー。 属性と値のペアとして指定します。 ステップレベルで定義されたヘッダーは、そのステップで使用されます。 ステップレベルでヘッダーを定義すると、シナリオレベルのドロップダウンリストから「**User-Agent**」を選択して割り当てたデフォルトヘッダーを除き、それまでに定義されたすべてのヘッダーが自動的に破棄されることに注意してください。 ただし、「User-Agent」のデフォルトヘッダーも、ステップレベルで指定することで上書きできます。たとえば、ヘッダー名を割り当てて値を設定しない場合、シナリオレベルのデフォルトヘッダーは解除されます。 ユーザーマクロおよび {HOST.*} マクロがサポートされています。 この設定により、cURL オプションのCURLOPT_HTTPHEADERが設定されます。
リダイレクトに従う	HTTP リダイレクトに従う場合はチェックボックスを選択します。 この設定により、cURL オプションのCURLOPT_FOLLOWLOCATIONが設定されます。
取得モード	取得モードを選択します: **Body** - HTTP レスポンスから本文のみを取得 **Headers** - HTTP レスポンスからヘッダーのみを取得 **Body and headers** - HTTP レスポンスから本文とヘッダーを取得

パラメータ	説明
タイムアウト	Zabbix は、設定された時間を超えて URL の処理を行いません（1 秒から最大 1 時間）。実際には、このパラメータは URL への接続確立にかかる最大時間と、HTTP リクエストの実行にかかる最大時間を定義します。そのため、Zabbix がステップに費やす時間はタイムアウトの **2** 倍を超えません。 時間のサフィックスがサポートされています（例: 30s、1m、1h）。ユーザーマクロがサポートされています。
必須文字列	必須の正規表現パターン。 取得したコンテンツ（HTML）が必須パターンに一致しない場合、ステップは失敗します。空の場合、必須文字列のチェックは実行されません。 例: Homepage of Zabbix Welcome.*admin 注: Zabbix の Web インターフェースで作成された正規表現をこのフィールドで参照することはサポートされていません。 ユーザーマクロおよび {HOST.*}マクロがサポートされています。
必須ステータスコード	想定される HTTP ステータスコードの一覧。Zabbix が一覧にないコードを取得した場合、ステップは失敗します。 空の場合、ステータスコードのチェックは実行されません。 例: 200,201,210-299 ユーザーマクロがサポートされています。

Note:
Web シナリオのステップに対する変更は、シナリオ全体を保存した場合にのみ保存されます。

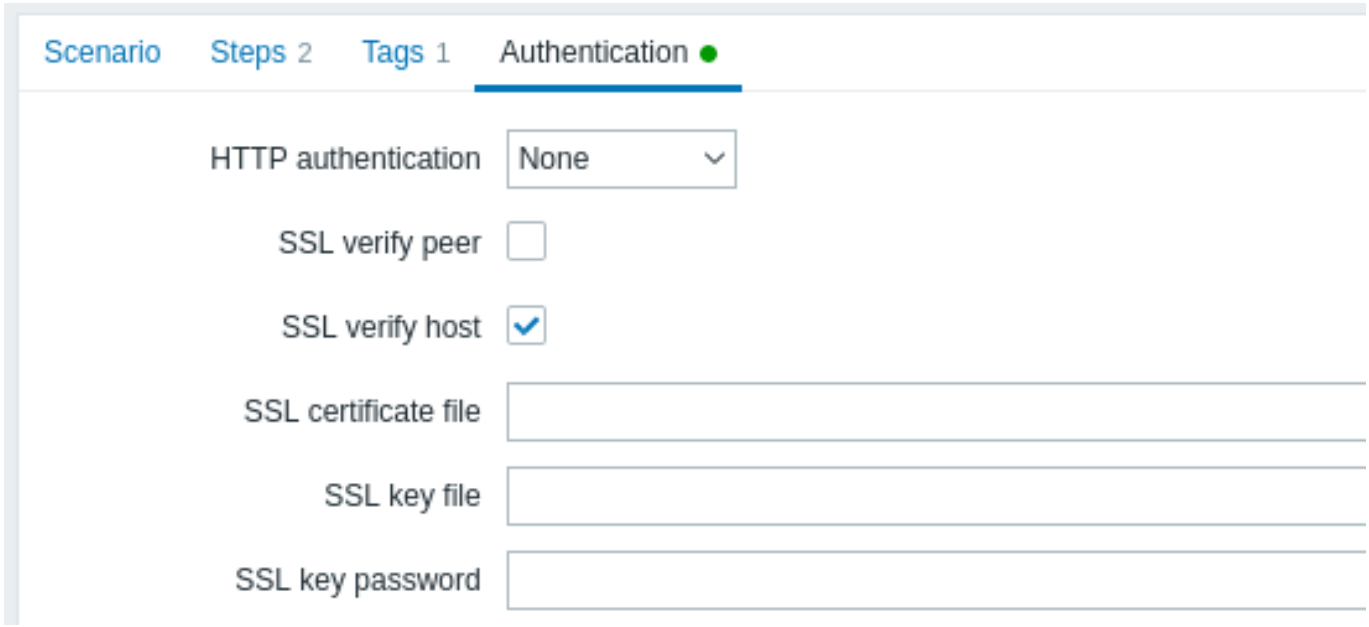
Web 監視のステップを設定する方法については、実際の例も参照してください。

タグの設定　タグタブでは、シナリオレベルのタグを定義できます。



タグ付けにより、Web シナリオとWeb 監視アイテムをフィルターできます。

認証の設定　認証タブでは、シナリオの認証オプションを設定できます。タブ名の横に緑色のドットが表示されている場合、何らかの HTTP 認証が有効になっています。



認証パラメータ:

パラメータ	説明
HTTP 認証	認証オプションを選択します: なし - 認証を使用しません; **Basic** - Basic 認証を使用します; **NTLM** - NTLM (Windows NT LAN Manager) 認証を使用します; **Kerberos** - Kerberos 認証を使用します（Zabbix での Kerberos の設定も参照）; **Digest** - Digest 認証を使用します。
ユーザー	ユーザー名を入力します（最大 255 文字）。 *HTTP* 認証が Basic、NTLM、Kerberos、または Digest に設定されている場合に使用できます。ユーザーマクロを使用できます。
パスワード	ユーザーパスワードを入力します（最大 255 文字）。 *HTTP* 認証が Basic、NTLM、Kerberos、または Digest に設定されている場合に使用できます。ユーザーマクロを使用できます。
SSL ピアの検証	Web サーバーの SSL 証明書を検証するには、チェックボックスをオンにします。 サーバー証明書は、システム全体の CA ロケーションにある信頼された認証局（CA）証明書に対して検証されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータSSLCALocationで上書きできます。 これにより、CURLOPT_SSL_VERIFYPEER cURL オプションが設定されます。
SSL ホストの検証	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスをオンにします。 これにより、CURLOPT_SSL_VERIFYHOST cURL オプションが設定されます。
SSL 証明書ファイル	クライアント認証に使用する SSL 証明書ファイルの名前です。証明書ファイルは PEM[1] 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は、*SSL* キーファイルフィールドを空のままにします。鍵が暗号化されている場合は、*SSL* キーパスワードフィールドにパスワードを指定します。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータSSLCertLocationで指定します。 このフィールドでは `HOST.*` マクロとユーザーマクロを使用できます。 これにより、CURLOPT_SSLCERT cURL オプションが設定されます。
SSL キーファイル	クライアント認証に使用する SSL 秘密鍵ファイルの名前です。秘密鍵ファイルは PEM[1] 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータSSLKeyLocationで指定します。 このフィールドでは `HOST.*` マクロとユーザーマクロを使用できます。 これにより、CURLOPT_SSLKEY cURL オプションが設定されます。
SSL キーパスワード	SSL 秘密鍵ファイルのパスワードです。 このフィールドではユーザーマクロを使用できます。 これにより、CURLOPT_KEYPASSWD cURL オプションが設定されます。

Attention:
[1] Zabbix は、PEM 形式の証明書と秘密鍵ファイルのみをサポートします。証明書と秘密鍵のデータが PKCS #12 形式のファイル（通常、拡張子は *.p12 または *.pfx）に含まれている場合は、次のコマンドを使用して PEM ファイルを生成できます:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Note:
Zabbix サーバーは、再起動せずに証明書の変更を反映します。

Note:
クライアント証明書と秘密鍵が 1 つのファイルに含まれている場合は、「SSL 証明書ファイル」フィールドにそのファイルを指定し、「SSL キーファイル」フィールドを空のままにします。証明書と鍵は、PEM 形式である必要があります。証明書と鍵の結合は簡単です:

```
cat client.crt client.key > client.pem
```

表示　ホストに設定された Web シナリオを表示するには、次の手順を実行します。

1. 監視 > ホストに移動します。
2. 一覧からホストを見つけ、最後の列にある *Web* ハイパーリンクをクリックします。
3. シナリオ名をクリックすると、詳細情報が表示されます。



Web シナリオの概要は、Web 監視ウィジェットを使用して ダッシュボードに表示することもできます。

Web シナリオの実行結果は、監視 > 最新データセクションで確認できます。

拡張監視　受信した HTML ページの内容をログに記録する必要がある場合があります。
これは、Web シナリオのステップが失敗した場合に特に有用です。
デバッグレベル 5（trace）は、その目的のために使用されます。

このレベルは、サーバー およびプロキシ の設定ファイルで、またはランタイム制御 オプション (-R log_level_increase="http poller,N"、ここで N はプロセス番号) を使用して設定できます。
以下の例では、デバッグレベル 4 がすでに設定されている場合に、拡張監視を開始する方法を示します。

```
# すべての http poller のログレベルを上げる:
zabbix_server -R log_level_increase="http poller"

# 2 番目の http poller のログレベルを上げる:
zabbix_server -R log_level_increase="http poller,2"
```

拡張 Web 監視が不要な場合は、-R log_level_decrease オプションを使用して停止できます。

Web 監視アイテム

概要

Web シナリオが作成されると、モニタリング用にいくつかの新しいアイテムが自動的に追加されます。

すべてのアイテムは、Web シナリオからタグを継承します

シナリオアイテム

シナリオが作成されると、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for scenario <Scenario>	このアイテムはシナリオ全体のダウンロード速度 (1 秒あたりのバイト数)、つまりすべてのステップの平均速度を収集します。 アイテムキー: web.test.in[Scenario,,bps] タイプ: *Numeric(float)*
Failed step of scenario <Scenario>	このアイテムはシナリオで失敗したステップの番号を表示します。すべてのステップが正常に実行されると、0 が返されます。 アイテムキー: web.test.fail[Scenario] タイプ: *Numeric(unsigned)*
Last error message of scenario <Scenario>	このアイテムはシナリオの最後のエラーメッセージテキストを返します。新しい値は、シナリオに失敗したステップがある場合にのみ保存されます。すべての手順に問題がない場合、新しい値は収集されません。 アイテムキー: web.test.error[Scenario] タイプ: *Character*

”Scenario” の代わりに実際のシナリオ名が使用されます。

Note:
シナリオ名にユーザーマクロが含まれている場合、これらのマクロは Web 監視アイテム名で未解決のままになります。

 シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

Note:
Web 監視アイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

これらのアイテムを使用して、トリガーを作成し、通知条件を定義できます。

例 1

「Web シナリオに失敗しました」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.fail[Scenario])<>0
```

必ず’Scenario’ を本来のシナリオ名に置き換えてください。

例 2

トリガー名に問題の説明を含む”Web scenario failed” トリガーを作成するには、次の名前でトリガーを定義します。

```
Web scenario "Scenario" failed: {ITEM.VALUE}
```

さらに次のようにトリガー式を定義します

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 3

「Web アプリケーションが遅い」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.in[Scenario,,bps])<10000
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

シナリオのステップアイテム

ステップが作成されるとすぐに、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for step <Step> of scenario <Scenario>	このアイテムはステップのダウンロード速度（バイト/秒）を取得します。 Item key: web.test.in[Scenario,Step,bps] Type: *Numeric(float)*
Response time for step <Step> of scenario <Scenario>	このアイテムはステップの応答時間を秒単位で取得します。リクエストの開始からすべての情報が転送されるまでがカウントされます。 Item key: web.test.time[Scenario,Step,resp] Type:*Numeric(float)*
Response code for step <Step> of scenario <Scenario>	このアイテムはステップの応答コードを取得します。 Item key:web.test.rspcode[Scenario,Step]Type: *Numeric(unsigned)*

"Scenario" と"Step" の代わりに、実際のシナリオとステップ名がそれぞれ使用されます。

Note:
Web モニタリングアイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

Note:
シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

これらのアイテムを使用してトリガーを作成し、通知条件を定義できます。たとえば「ZabbixGUI ログインが遅すぎる」トリガーを作成するには次のようにトリガー式を定義します。

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

実際のシナリオ

概要

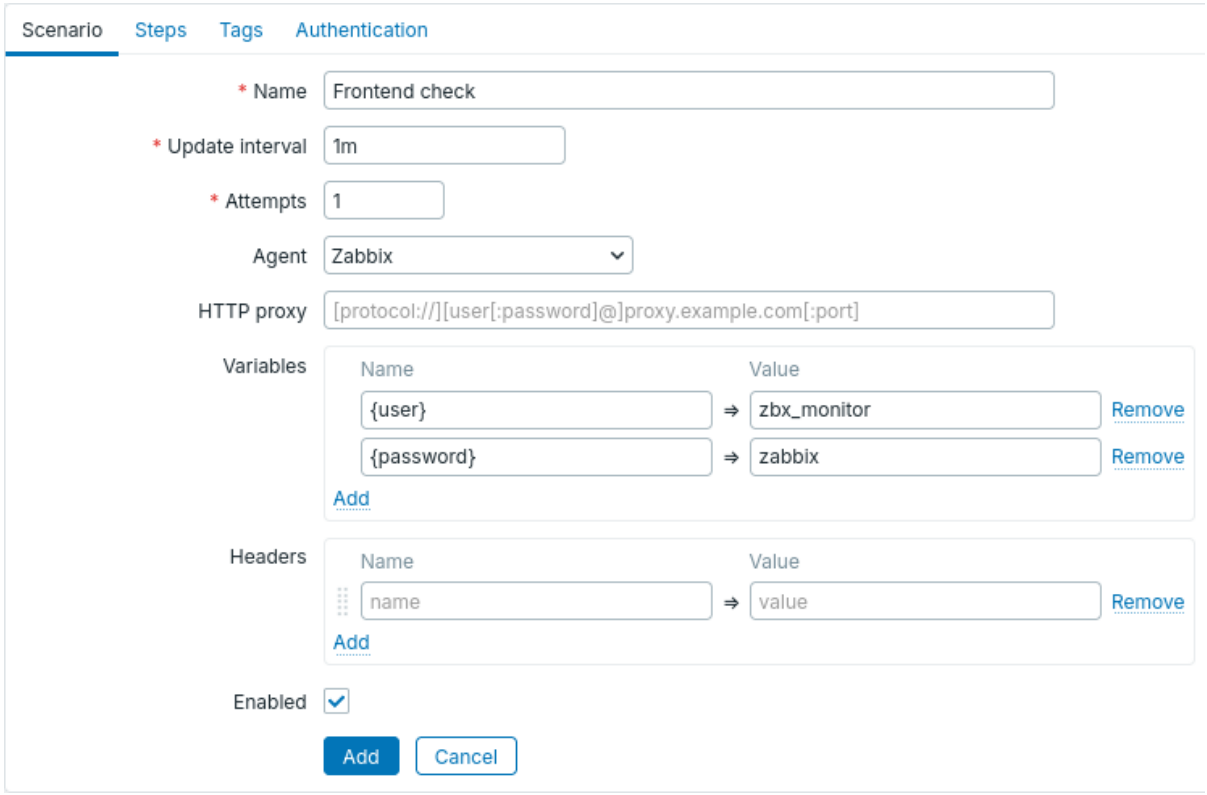
このセクションでは、Web 監視をどのように使用できるかを、実際の例に沿って手順ごとに説明します。

Zabbix の Web 監視を使用して、Zabbix Web インターフェースを監視します。目的は、利用可能であるか、正しい内容を提供しているか、そしてどれだけ速く動作するかを確認することです。これを実現するには、最初のページの到達可否の確認、ユーザー名とパスワードでのログイン、ログイン成功の検証、ログアウト、およびログアウトの確認を含む、いくつかの手順が必要です。

シナリオ

新しい web シナリオを追加する

1. データ収集 > ホストに移動します。
2. ホストを選択し、そのホストの行にある *Web* をクリックします。
3. *Web* シナリオの作成をクリックします。



4. 新しいシナリオのフォームで、シナリオタブに次のフィールドを入力します。

    - 名前: Frontend check.
    - 更新間隔: 1m.
    - 試行回数: 1.
    - エージェント: Zabbix.
 変数セクションで、次の名前と値のペアを追加します。
    - {user}: 既存の Zabbix ユーザー認証情報;
    - {password}: 既存の Zabbix ユーザー認証情報。

Note:
安全上の理由から、監視用途には最小権限のみを持つ別のユーザーを作成して使用することを推奨します。

必要に応じて、タグタブに切り替えて web シナリオのタグを追加します。web シナリオのタグを使用すると、関連するアイテムやトリガーをすばやく識別したり、収集したデータを検索したりできます。たとえば、このチュートリアルに適したタグは `component: web-scenario` および/または `target: frontend` です。

Web シナリオのステップを設定する

Steps タブに切り替え、シナリオのステップを定義します。
 各ステップを設定したら、*Add* ボタンをクリックします。

Attention:
Zabbix Web インターフェースはログイン時に JavaScript リダイレクトを使用するため、ログインは最初に実行する必要があり、ログイン後の機能はその後のステップでのみ確認できます。また、ログインステップでは **index.php** ファイルへの完全な URL を使用する必要があります。

すべてのステップに共通するフィールド

ステップ固有のフィールドに加えて、以下で説明する各ステップに次の共通フィールドを入力します。

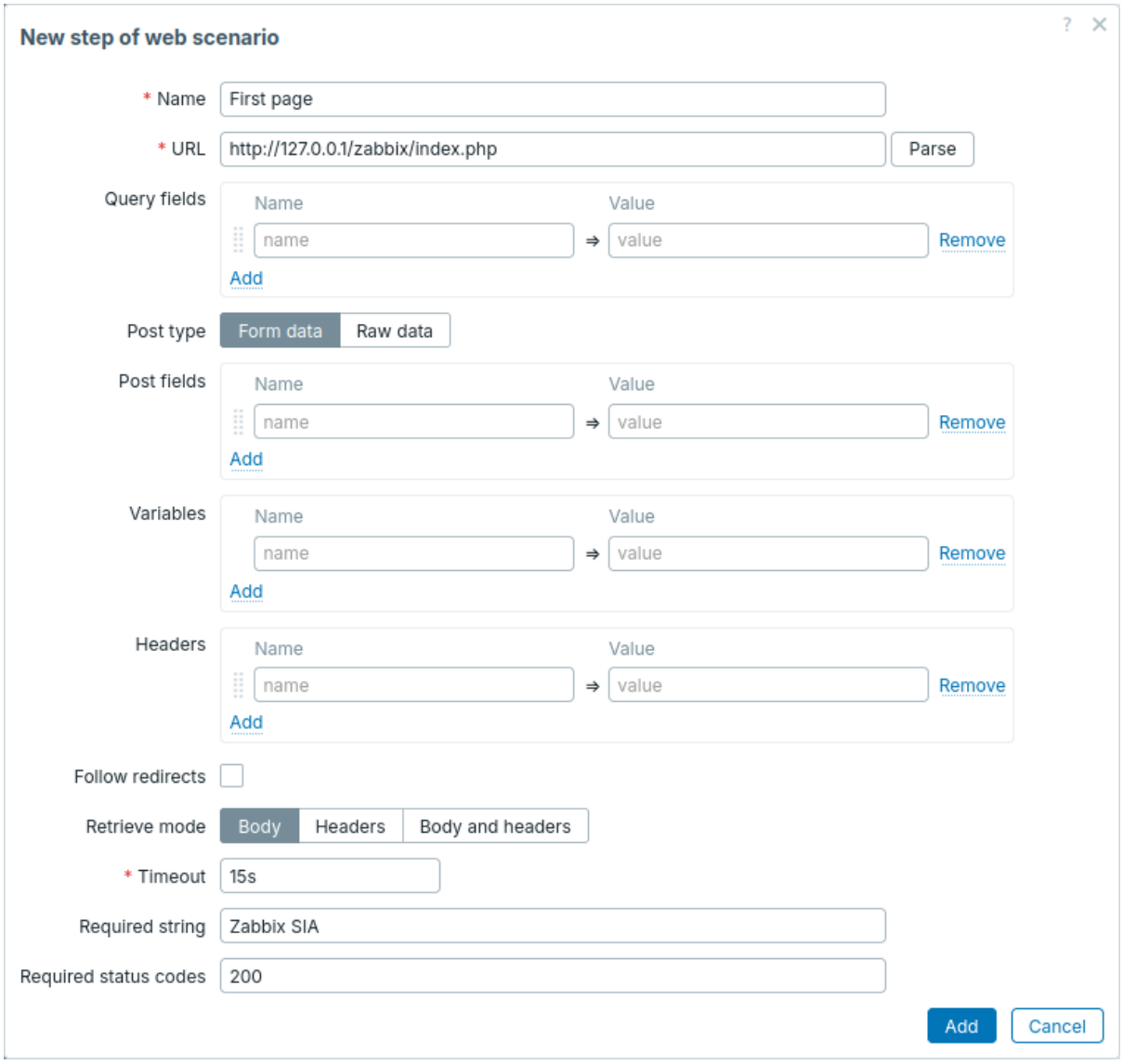
- *URL*: Zabbix の Web インターフェースの URL（Apache の場合は `http://<server_ip_or_name>/zabbix/index.php`、NGINX の場合は `http://<server_ip_or_name>/index.php`）。Web シナリオのステップ 3 では、以下で説明する別の URL を使用します。
- *Timeout*: 15s。
- *Required status codes*: 200。

Web シナリオのステップ 1

最初のページが正しく応答し、HTTP ステータスコード 200 を返し、*Zabbix SIA* というテキストを含んでいることを確認します。
 次のフィールドを入力します。

- *Name*: First page.

- *Timeout*: 15s.
- *Required string*: Zabbix SIA.
- すべてのステップに共通するフィールド。

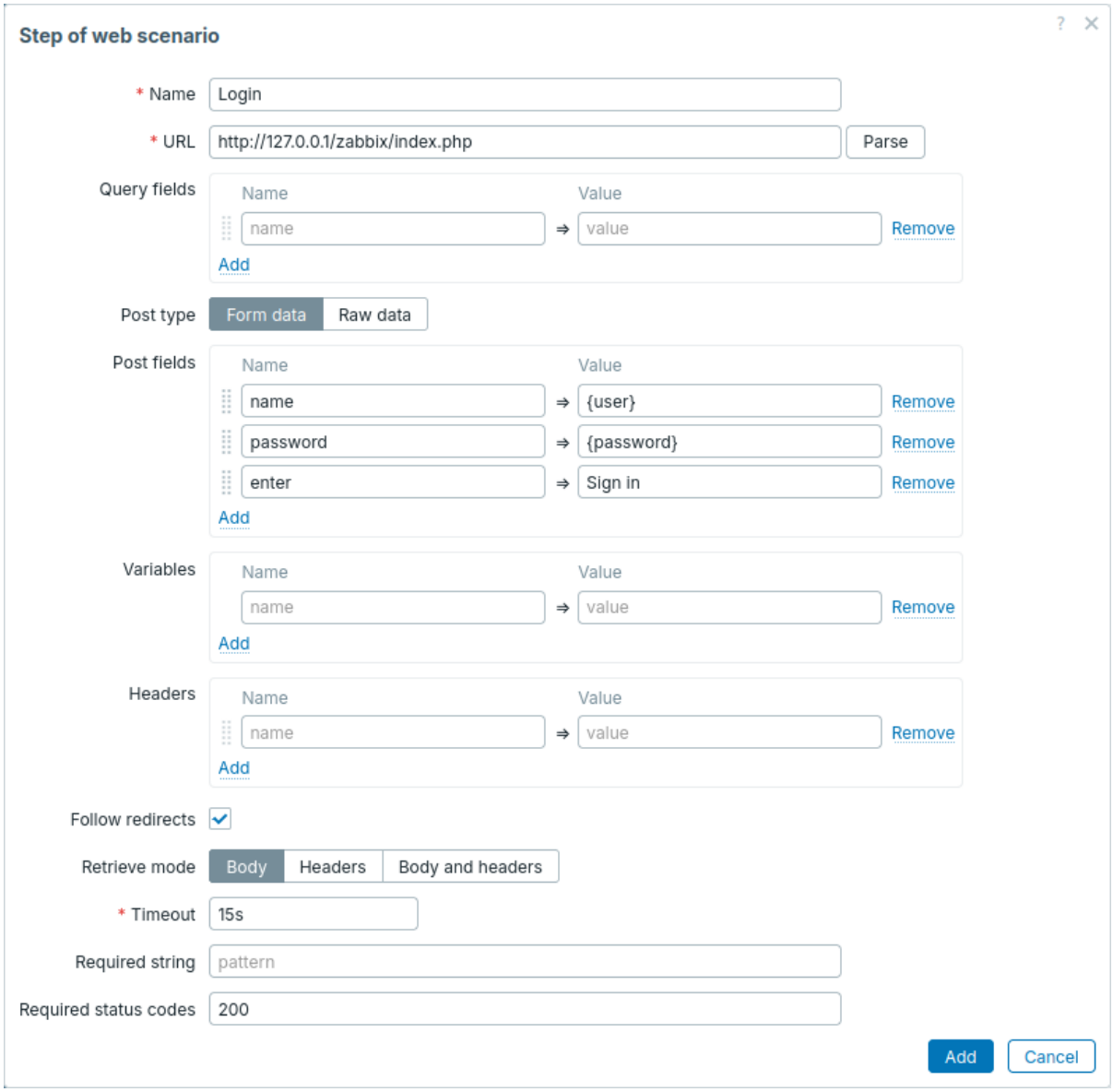


Web シナリオのステップ 2

Scenario タブで定義したマクロ（変数）*{user}* と *{password}* を使用して、Zabbix Web インターフェースにログインします。
次のフィールドを入力します。

- *Name*: Login.
- *Post fields* セクションで、次の名前と値のペアを追加します。
  - name: `{user}`;
  - password: `{password}`;
  - enter: Sign in.
- すべてのステップに共通のフィールド。

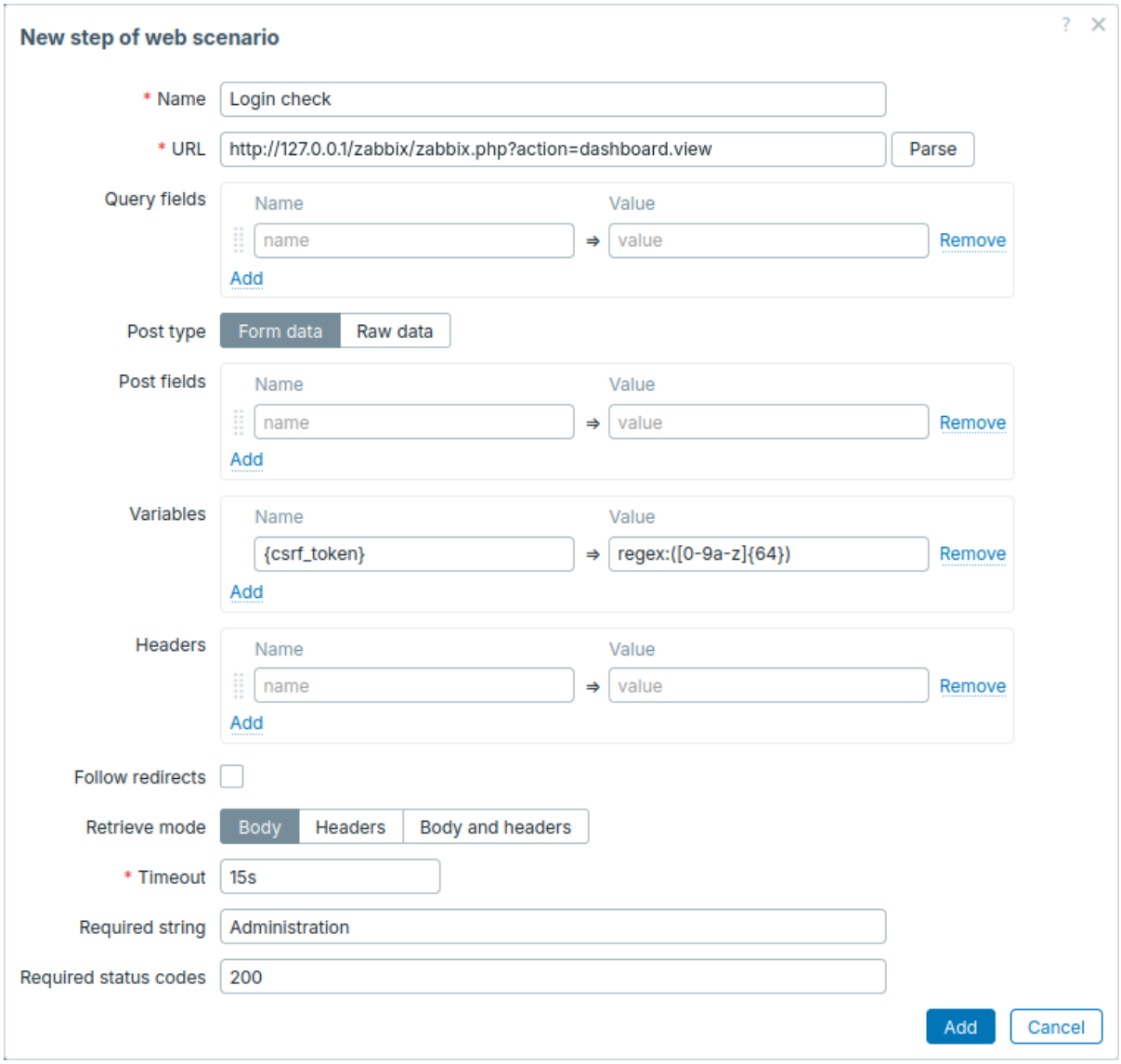
Follow redirects チェックボックスを選択します。



Web シナリオステップ 3

ログイン後、ログイン時にのみ表示される文字列、たとえば *Administration* が表示されることを確認して成功を検証します。
次のフィールドを入力します。

- *Name*: Login check.
- *URL*: ログイン後にユーザーがリダイレクトされる URL（例: http://127.0.0.1/zabbix/zabbix.php?action=dashboard.view）。
- *Variables* セクションで、名前と値のペア `{csrf_token}: regex:([0-9a-z]{64})` を追加します。この変数は、割り当てられた CSRF トークンの値を取得し、web シナリオステップ 4 で再利用するために使用します。
- *Required string*: Administration.
- すべてのステップに共通のフィールド。



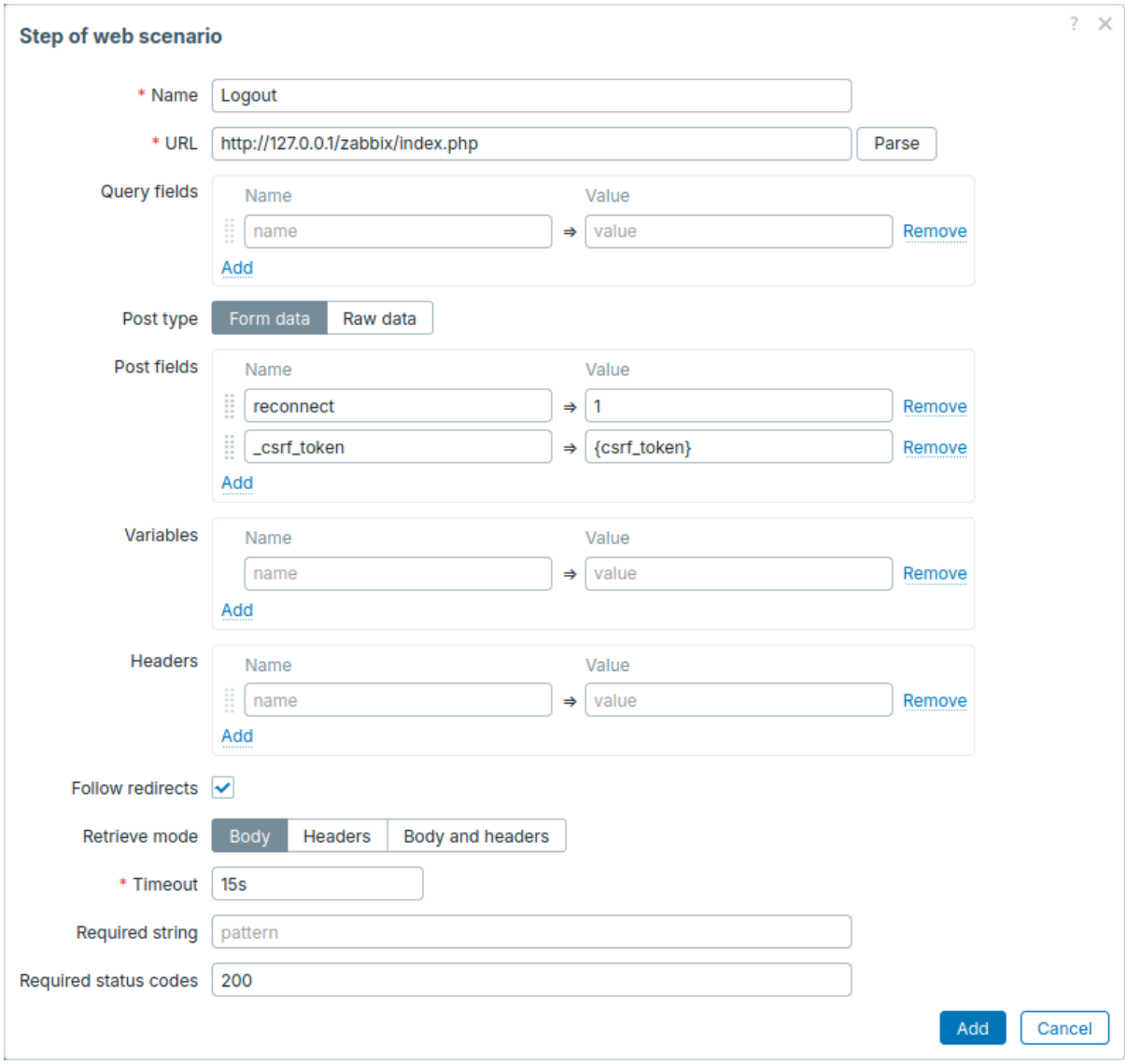
Web シナリオのステップ 4

Web インターフェースへのアクセスとログインを確認したら、ログアウトのステップを追加します。追加しない場合、Zabbix データベースに開いたセッションのレコードが大量に蓄積されます。
 次のフィールドに入力します。

- 名前: Logout。
- *Post* フィールドセクションで、次の名前と値のペアを追加します。
  - reconnect: 1;
  - _csrf_token: `{csrf_token}`。
- すべてのステップに共通するフィールド。

リダイレクトに従うチェックボックスを選択します。

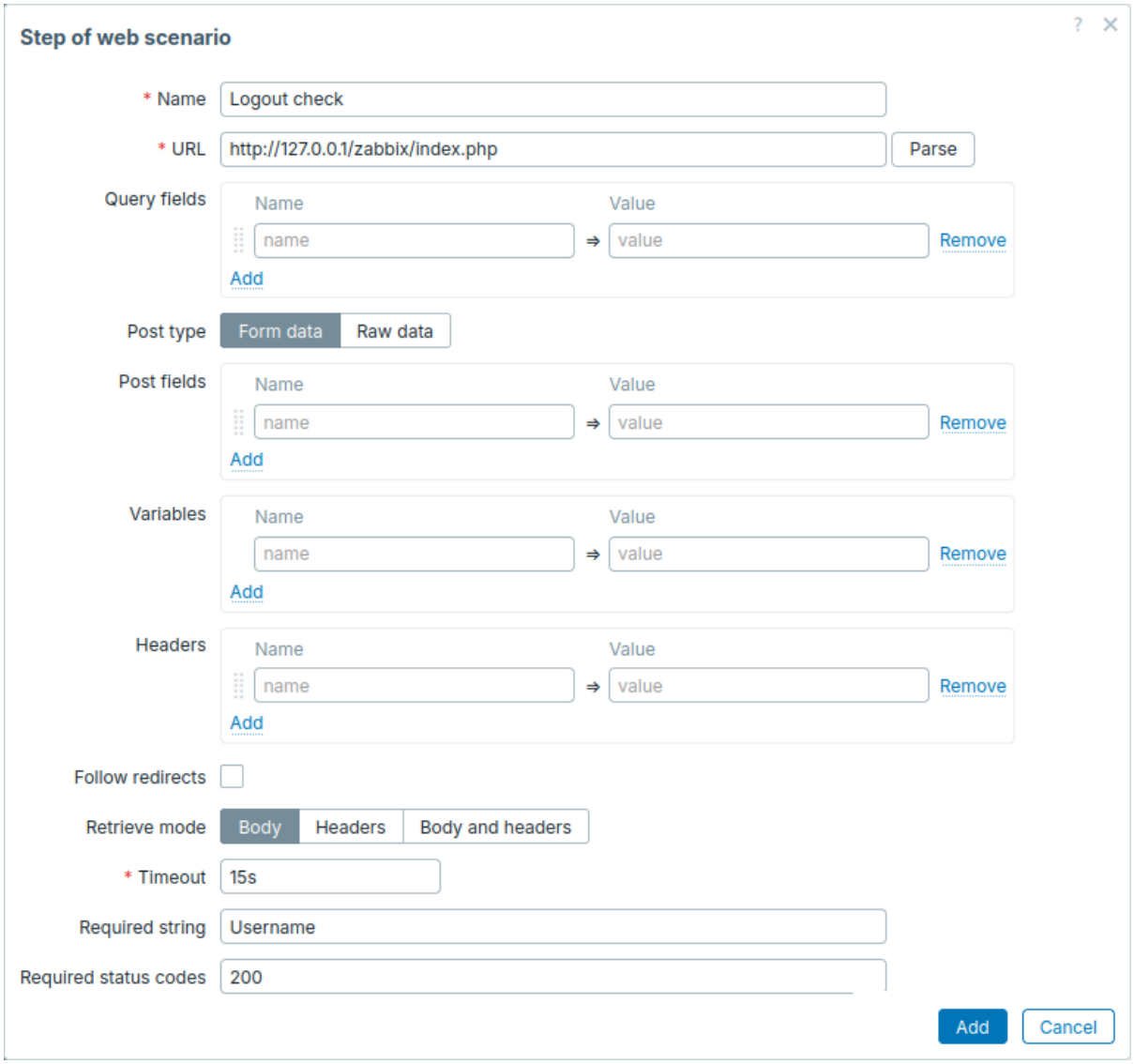
このステップでは、Web シナリオのステップ 3で取得した変数 `{csrf_token}` を使用します。



Web シナリオステップ 5

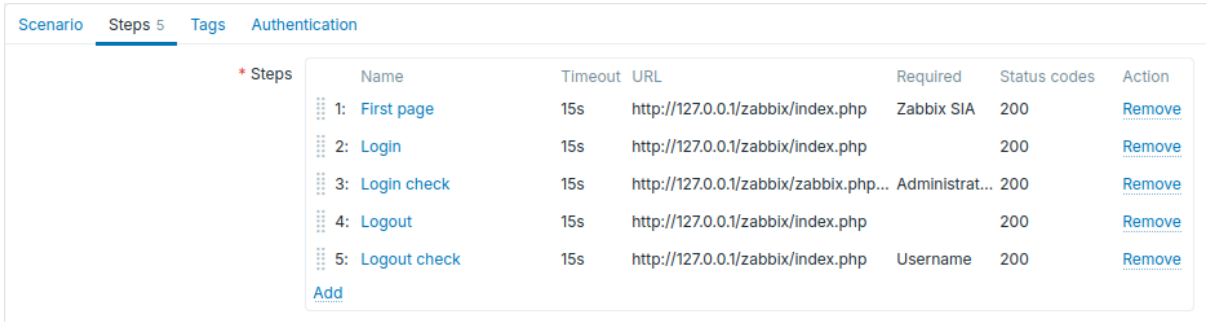
ログアウトを確認するには、*Username* 文字列を確認します。
 次のフィールドを入力します。

- *Name*: Logout check.
- *Required string*: Username.
- すべてのステップに共通のフィールド。



ステップの完全な設定

Web シナリオステップの完全な設定は次のようになります。

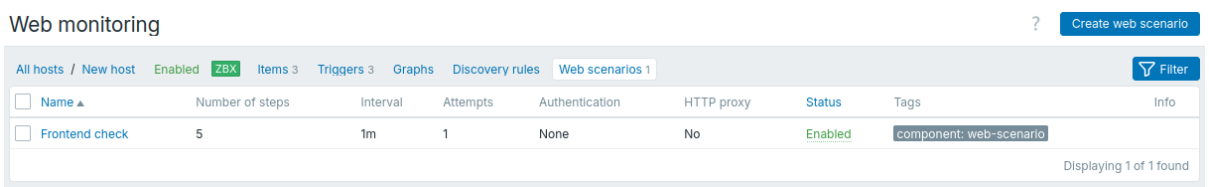


結果を確認する

作成した Web 監視シナリオを保存します。

Web シナリオが作成されると、監視用の新しいweb monitoring itemsが自動的に追加されます。

このシナリオはホストに追加されます。Web シナリオの情報を表示するには、*Monitoring* > *Hosts* に移動し、一覧からホストを見つけて、最後の列にある *Web* ハイパーリンクをクリックします。



シナリオ名をクリックすると、詳細な統計情報を表示できます。

サービス監視

概要　サービス監視は、IT インフラストラクチャ全体のサービスツリーを概観し、インフラストラクチャの弱点を特定し、さまざまな IT サービスの SLA を計算し、その他の情報をより高いレベルで確認するために使用できる、ビジネスレベルの監視です。サービス監視では、ディスク容量不足やプロセッサー負荷の高さなどの低レベルの詳細ではなく、サービス全体の可用性に重点を置きます。また、サービスが期待どおりに動作していない場合に、障害の根本原因を特定する機能も提供します。

サービス監視では、監視対象データを階層構造で表現できます。

非常に単純なサービス構造は、次のようになります。

```
Service
|
|-Workstations
| |
| |-Workstation1
| |
| |-Workstation2
|
|-Servers
```

構造内の各ノードにはステータス属性があります。ステータスは、選択したアルゴリズムに従って計算され、上位レベルに伝播されます。個々のノードのステータスは、マッピングされた障害のステータスによって影響を受けます。障害のマッピングは、タグ付けによって行います。

サービス状態の変更が検出された場合、Zabbix は通知を送信したり、Zabbix サーバー上でスクリプトを自動的に実行したりできます。子サービスのステータスに基づいて、親サービスを「障害状態」に移行するかどうかを柔軟なルールで定義できます。サービスの障害データは、SLA の計算や、柔軟な条件に基づく SLA レポートの送信に使用できます。

サービス監視は、次のセクションで構成される「サービス」メニューで設定します。

- サービス

「サービス」セクションでは、親サービスを追加し、続けて親サービスに子サービスを追加することで、監視対象インフラストラクチャの階層を構築できます。

サービスツリーの設定に加えて、このセクションではインフラストラクチャ全体の概要を確認でき、サービス状態の変更につながった障害をすばやく特定できます。

- SLA

このセクションでは、サービスレベル契約を定義し、特定のサービスに対するサービスレベル目標を設定できます。

- SLA レポート

このセクションでは、SLA レポートを表示できます。

サービスアクション

サービスのアクションも設定できます。

サービスアクションは任意であり、次の操作を実行できます。

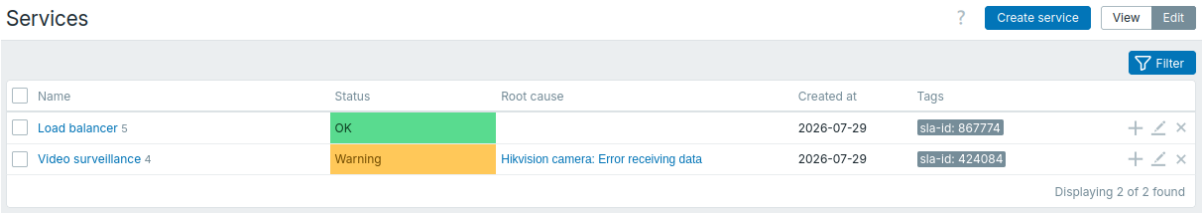
- サービスが停止したことを通知する
- サービス状態の変更時に Zabbix サーバー上でリモートコマンドを実行する
- サービスが再び稼働したときに復旧通知を送信する

関連項目:

- SLA 監視設定の例
- 6.0 未満の Zabbix バージョンからのサービスのアップグレードに関する注意事項

サービスツリー

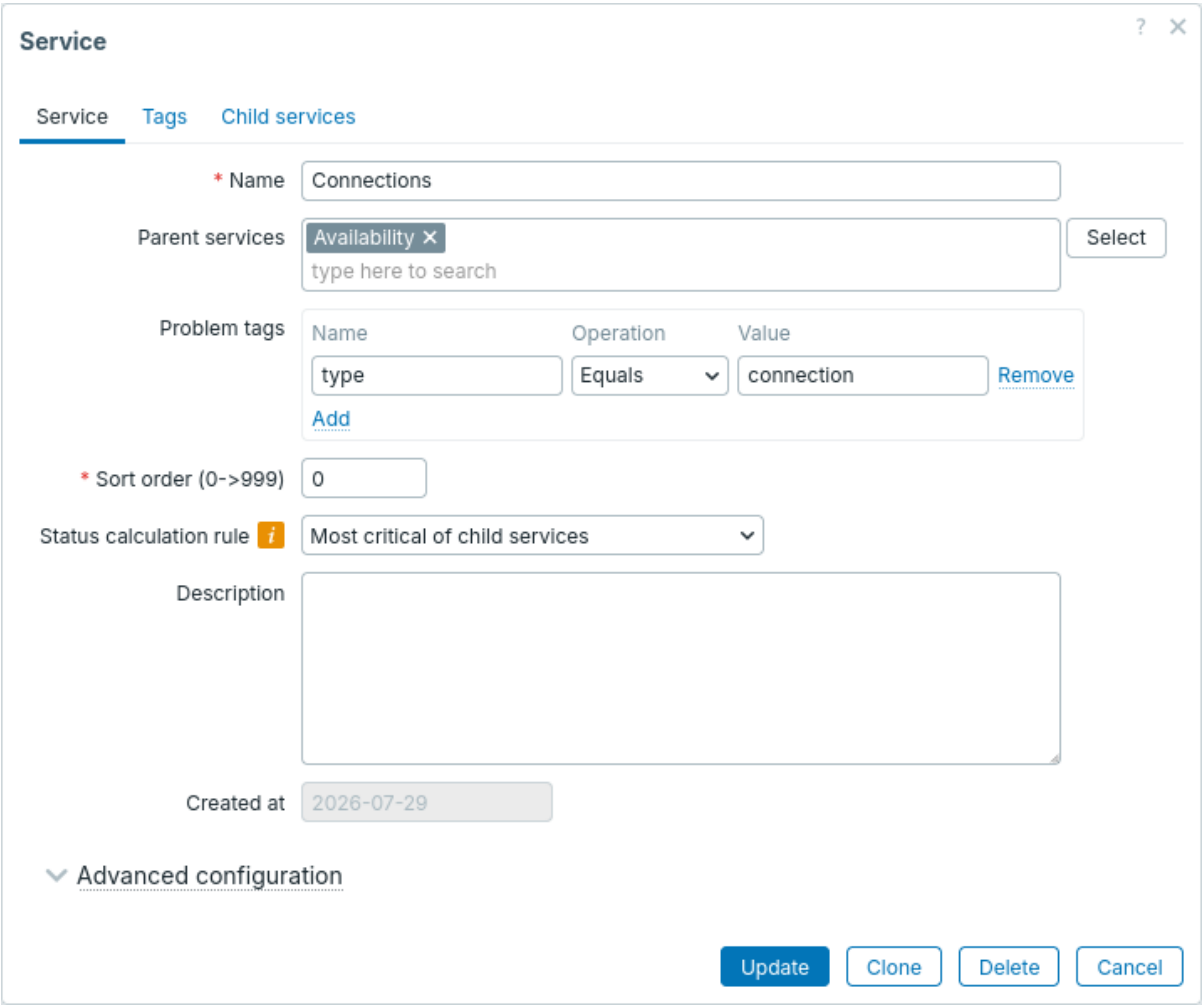
サービスツリーは、*Services* > *Services* メニューセクションで設定します。右上隅で、view から編集モードに切り替えます。



新しいサービスを設定するには、右上隅の *Create service* ボタンをクリックします。

子サービスをすばやく追加するには、代わりに親サービスの横にあるプラスアイコンをクリックします。これにより同じサービス設定フォームが開きますが、*Parent services* パラメータはあらかじめ入力された状態になります。

サービス設定　**Service** タブで、必要なサービスパラメータを指定します。

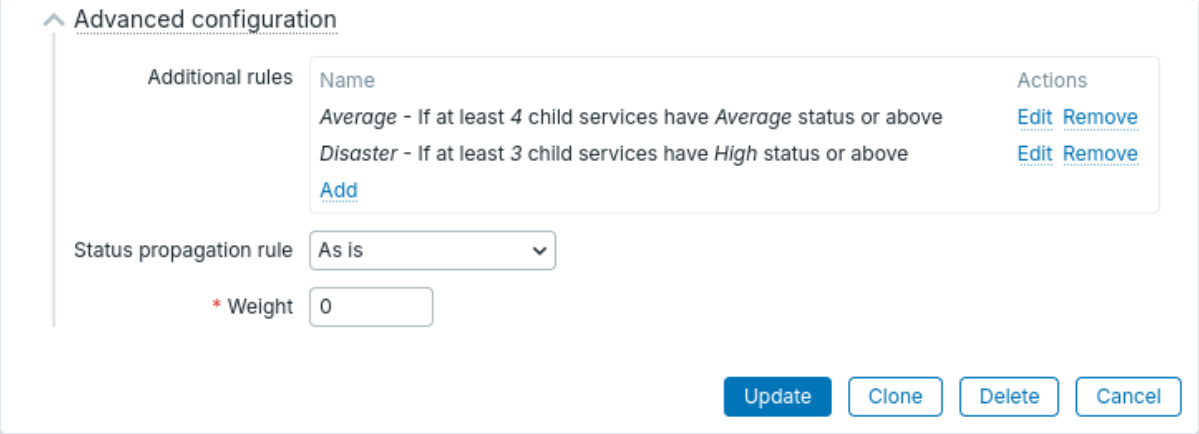


必須入力フィールドには赤いアスタリスクが表示されます。

Parameter	Description
Name	サービス名。
Parent services	サービスが属する親サービス。 最上位レベルのサービスを追加する場合は、このフィールドを空欄のままにします。 1 つのサービスに複数の親サービスを設定できます。この場合、サービスツリーでは各親サービスの下に表示されます。
Problem tags	障害データをサービスにマッピングするためのタグを指定します。 複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件で使用できる演算子は次の 2 つです。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別） **Contains** - 入力した文字列を含むタグ値を持つ指定したタグ名を含めます（部分一致、大文字と小文字を区別）
Sort order	表示順。数値が小さいものが先に表示されます。

Parameter	Description
Status calculation rule	サービスステータスの計算ルール: **Most critical if all children have problems** - すべての子サービスに障害がある場合、子サービス内で最も重大な障害を使用してサービスステータスの色を決定します **Most critical of child services** - 子サービス内で最も重大な障害を使用してサービスステータスの色を決定します **Set status to OK** - サービスステータスを計算しません 追加のステータス計算ルールは、advanced configuration オプションで設定できます。
Description	サービスの説明。
Created at	サービスが作成された時刻。既存のサービスを編集する場合に表示されます。
Advanced configuration	*Advanced configuration* 見出しをクリックすると、advanced configuration オプションが表示されます。

詳細設定



Parameter	Description
Additional rules	*Add* をクリックして、追加のステータス計算ルールを設定します。
Set status to	条件に一致した場合のサービスステータスを、*OK*（デフォルト）、*Not classified*、*Information*、*Warning*、*Average*、*High*、または *Disaster* のいずれかに設定します。
Condition	直接の子サービスに対する条件を選択します: **(Status)** ステータス以上の子サービスが少なくとも **(N)** 個ある場合 **(Status)** ステータス以上の子サービスが少なくとも **(N%)** ある場合 **(Status)** ステータス以下の子サービスが **(N)** 個未満の場合 **(Status)** ステータス以下の子サービスが **(N%)** 未満の場合 **(Status)** ステータス以上の子サービスの重みの合計が少なくとも **(W)** である場合 **(Status)** ステータス以上の子サービスの重みの合計が少なくとも **(N%)** である場合 **(Status)** ステータス以下の子サービスの重みの合計が **(W)** 未満の場合 **(Status)** ステータス以下の子サービスの重みの合計が **(N%)** 未満の場合 複数の条件が指定され、状況が複数の条件に一致する場合は、最も深刻なレベルが設定されます。
N (W)	条件における N または W の値 (1-100000)、または N% (1-100) を設定します。
Status	条件における *Status* の値を選択します: *OK*（デフォルト）、*Not classified*、*Information*、*Warning*、*Average*、*High*、または *Disaster*。
Status propagation rule	親サービスへサービスステータスを伝播するためのルール: **As is** - ステータスは変更せずに伝播されます **Increase by** - 伝播されるステータスを 1 から 5 段階まで上げることができます **Decrease by** - 伝播されるステータスを 1 から 5 段階まで下げることができます **Ignore this service** - ステータスは親サービスへまったく伝播されません **Fixed status** - ステータスは静的に伝播されます。つまり、常に同じステータスになります
Weight	サービスの重み（0（デフォルト）から 1000000 までの整数）。

Note:
追加のステータス計算ルールは、主の *Status calculation rule* パラメーターに従って計算されたレベルよりも深刻度を上げる場合にのみ使用できます。追加ルールによりステータスが Warning になる場合でも、*Status calculation rule* によりステータスが Disaster であれば、サービスのステータスは Disaster になります。

Tags タブには、サービスレベルのタグ が含まれています。サービスレベルのタグは、サービスを識別するために使用されます。この種類のタグは、障害をサービスにマッピングするためには使用されません（その場合は、最初のタブにある障害タグ を使用してください）。

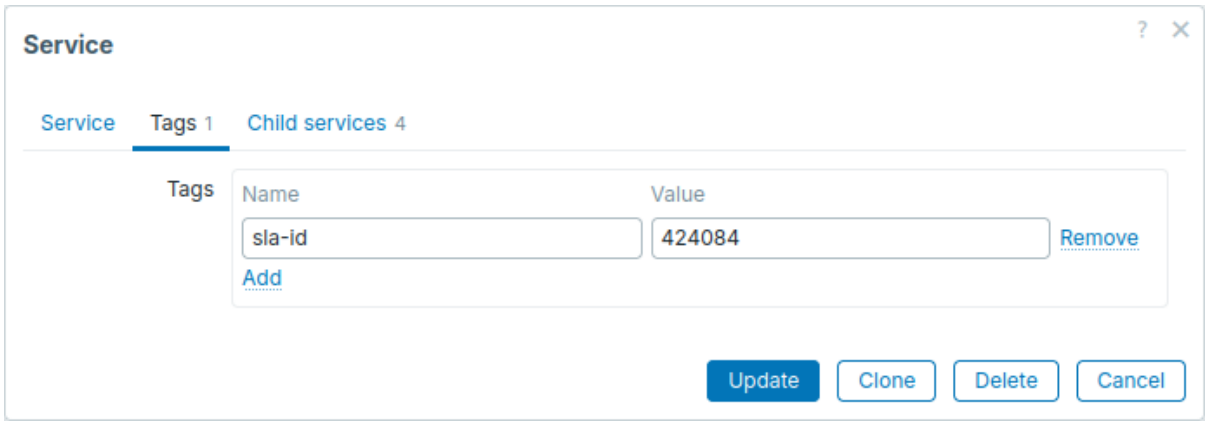
子サービスタブでは、依存するサービスを指定できます。既存のサービスの一覧からサービスを追加するには、追加をクリックします。新しい子サービスを追加する場合は、まずこのサービスを保存してから、作成したサービスの横にあるプラスアイコンをクリックしてください。

タグ　サービスには 2 種類のタグがあります。

- サービスタグ
- 障害タグ

サービスタグ

サービスタグは、サービスをservice actionsおよびSLAsに一致させるために使用されます。これらのタグは、サービス設定の *Tags* タブで指定します。SLA のマッピングには *OR* ロジックが使用されます。サービスに少なくとも 1 つ一致するタグがある場合、そのサービスは SLA にマッピングされます。サービスアクションでは、マッピングルールを設定でき、*AND*、*OR*、または *AND/OR* ロジックのいずれかを使用できます。

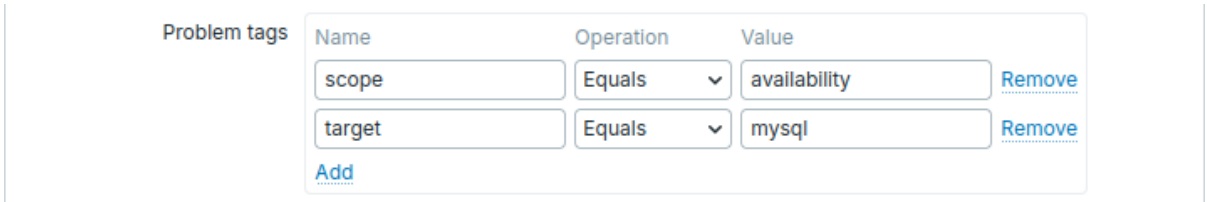


障害タグ

障害タグは、障害とサービスを一致させるために使用されます。これらのタグは、主要なサービス設定タブで指定します。

障害タグを定義でき、障害と直接関連付けられるのは、階層の最下位レベルにある子サービスのみです。障害タグが一致すると、サービスのステータスは障害と同じステータスに変わります。複数の障害がある場合、サービスは最も深刻な障害のステータスを持ちます。親サービスのステータスは、その後、ステータス計算ルールに従って子サービスのステータスに基づいて計算されます。

複数のタグが指定されている場合は、*AND* ロジックが使用されます。つまり、障害がサービスにマッピングされるには、サービス設定で指定されたすべてのタグを持っている必要があります。

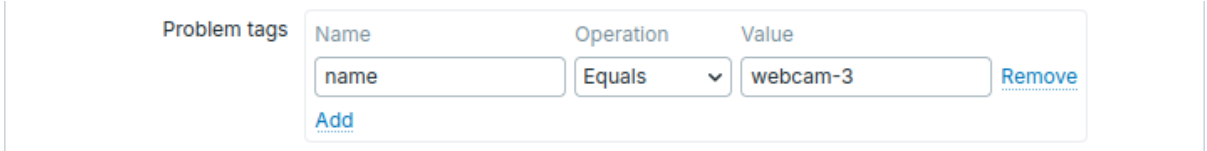


Note:
Zabbix の障害は、テンプレート、ホスト、アイテム、web シナリオ、およびトリガーのチェーン全体からタグを継承します。これらのタグはいずれも、障害をサービスに一致させるために使用できます。

例*:*

障害 *Web camera 3 is down* にはタグ `type:video-surveillance`、`floor:1`、`name:webcam-3` があり、ステータスは *Warning* です。

サービス **Web camera 3** には、指定された障害タグ `name:webcam-3` のみがあります。



この障害が検出されると、サービスのステータスは *OK* から *Warning* に変わります。

もしサービス **Web camera 3** に障害タグ `name:webcam-3` と `floor:2` があった場合、条件が部分的にしか満たされないため、障害が検出されてもステータスは変更されません。

既存のサービスの変更

> **Note:**
> 以下に説明するボタンは、*Services* セクションが編集モードのときにのみ表示されます。

既存のサービスを編集するには、サービスの横にある鉛筆アイコンをクリックします。

既存のサービスを複製するには、鉛筆アイコンをクリックして設定を開き、次に *Clone* ボタンをクリックします。サービスを複製すると、親リンクは保持されますが、子リンクは保持されません。

サービスを削除するには、その横にある `x` アイコンをクリックします。親サービスを削除しても、その子サービスは削除されず、サービスツリーで 1 つ上の階層に移動します（第 1 階層の子は、削除された親サービスと同じ階層になります）。

サービス一覧の下にある 2 つのボタンでは、一括編集のオプションを利用できます。

- *Mass update* - サービスプロパティを一括更新
- *Delete* - サービスを削除

これらのオプションを使用するには、対象のサービスの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

SLA

概要　サービスを作成すると、サービスのパフォーマンスがサービスレベル合意（SLA）に沿っているかどうかの監視を開始できます。

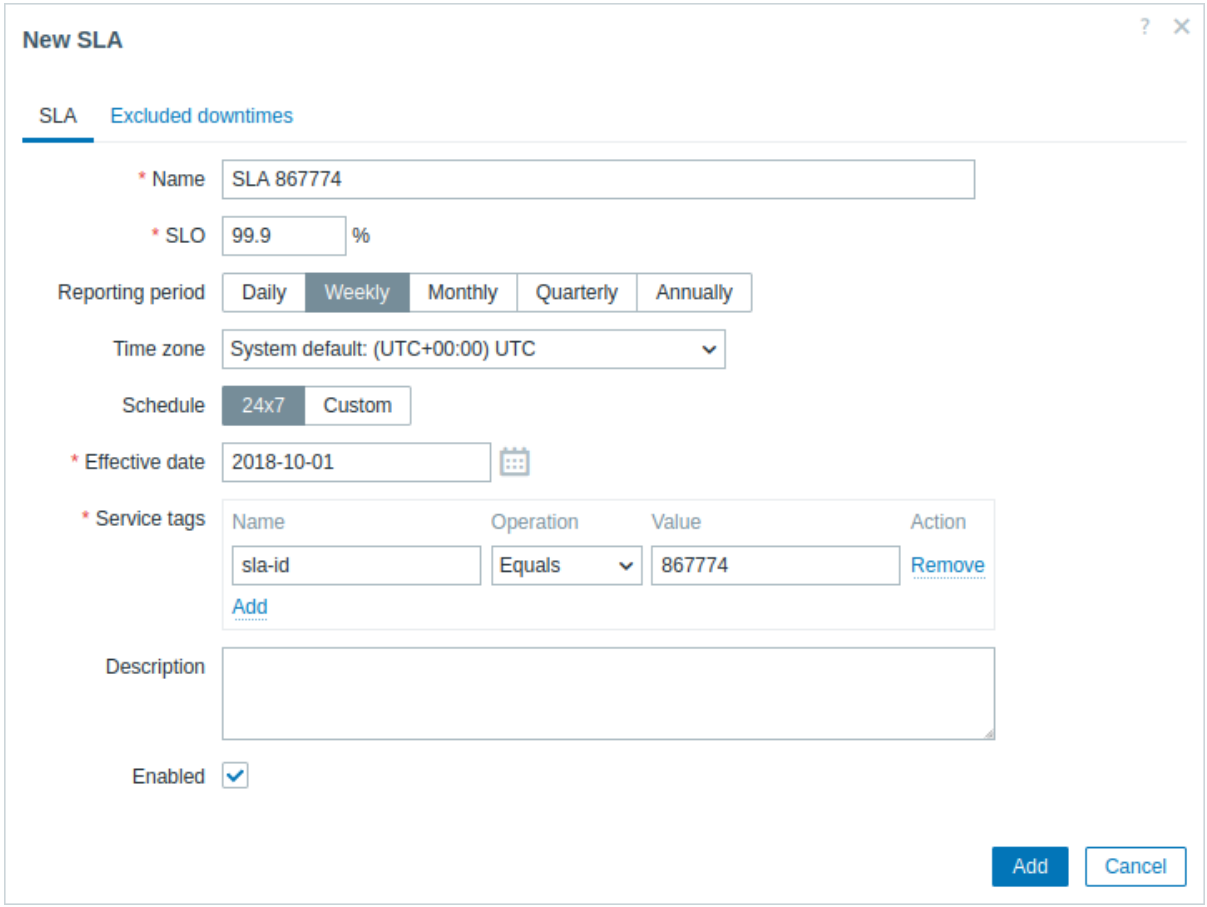
サービス > *SLA* メニューセクションでは、さまざまなサービスの SLA を設定できます。Zabbix の SLA では、サービスレベル目標（SLO）、想定される稼働スケジュール、および計画メンテナンス時間を定義します。

SLA とサービスは、サービスタグによって照合されます。同じ SLA を複数のサービスに適用できます。パフォーマンスは、照合された各サービスについて個別に測定されます。1 つのサービスに複数の SLA を割り当てることもできます。この場合、各 SLA のデータが個別に表示されます。

SLA レポートでは、Zabbix は実際のサービス可用性を測定するサービスレベル指標（SLI）データを提供します。サービスが SLA の目標を満たしているかどうかは、SLO（想定可用性、%）と SLI（実際の可用性、%）を比較して判定されます。

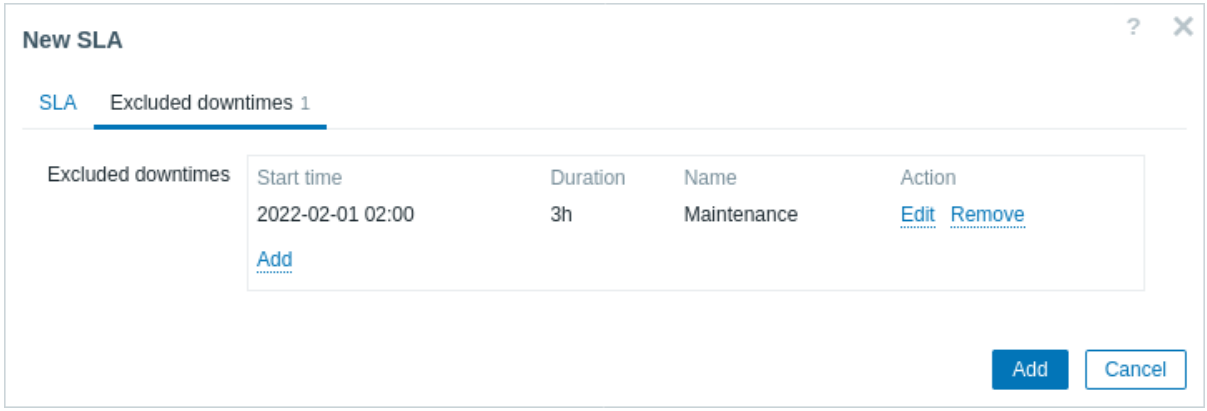
設定　新しい SLA を作成するには、*SLA* の作成ボタンをクリックします。

SLA タブでは、SLA の一般的なパラメータを指定できます。



パラメータ	説明
名前	SLA の名前を入力します。
SLO	サービスレベル目標（SLO）をパーセントで入力します。
レポート期間	SLA レポートで SLI データの計算に使用する SLI レポート期間を選択します。 日次 - 毎日、00:00:00 から 23:59:59 まで。 週次 - 毎週、日曜日の 00:00:00 から土曜日の 23:59:59 まで。 月次 - 毎月、1 日の 00:00:00 から末日の 23:59:59 まで。 四半期 - 各暦四半期：第 1 四半期（1 月～3 月）、第 2 四半期（4 月～6 月）、第 3 四半期（7 月～9 月）、第 4 四半期（10 月～12 月）。 年次 - 各暦年、1 月 1 日の 00:00:00 から 12 月 31 日の 23:59:59 まで。
タイムゾーン	SLA のタイムゾーンを選択します。
スケジュール	SLA のスケジュール（24 時間 365 日またはカスタム）を選択します。
有効日	SLA の計算を開始する日付を選択します。
サービスタグ	この SLA を適用するサービスを識別するためのサービスタグを追加します。 名前 - サービスタグ名。完全一致で、大文字と小文字が区別されます。 操作 - タグの値を完全一致（大文字と小文字を区別）させる場合は等しいを、タグの値の一部を一致(大文字と小文字を区別しない）させる場合は含むを選択します。 値 - 選択した操作に従って検索するサービスタグの値。 少なくとも 1 つのサービスタグが一致すると、そのサービスに SLA が適用されます。
説明	SLA の説明を追加します。
有効	チェックボックスをオンにして、SLA の計算を有効にします。

除外するダウンタイムタブでは、SLA の計算から除外するダウンタイムを指定できます。



追加をクリックして除外するダウンタイムを設定し、期間名、開始日、期間を入力します。

SLA レポート　サービスが SLA と比較してどのように動作しているかは、SLA レポートで確認できます。SLA レポートは次の場所から表示できます。

- *SLA* セクションで SLA レポートのハイパーリンクをクリックする。
- *Services* セクションで Info タブ内の SLA 名をクリックする。
- ダッシュボードのウィジェット *SLA report* で表示する。

SLA を設定すると、services セクションの *Info* タブにもサービスのパフォーマンスに関する情報が表示されます。

セットアップ例

概要　このセクションでは、Zabbix 高可用性クラスターをサービスとして監視するためのセットアップについて説明します。

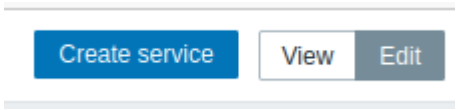
前提条件　サービス監視を設定する前に、以下のホストが設定されている必要があります。

- *HA ノード 1*：少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）`component:ha-node-1`
- *HA ノード 2*：少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）`component:ha-node-2`

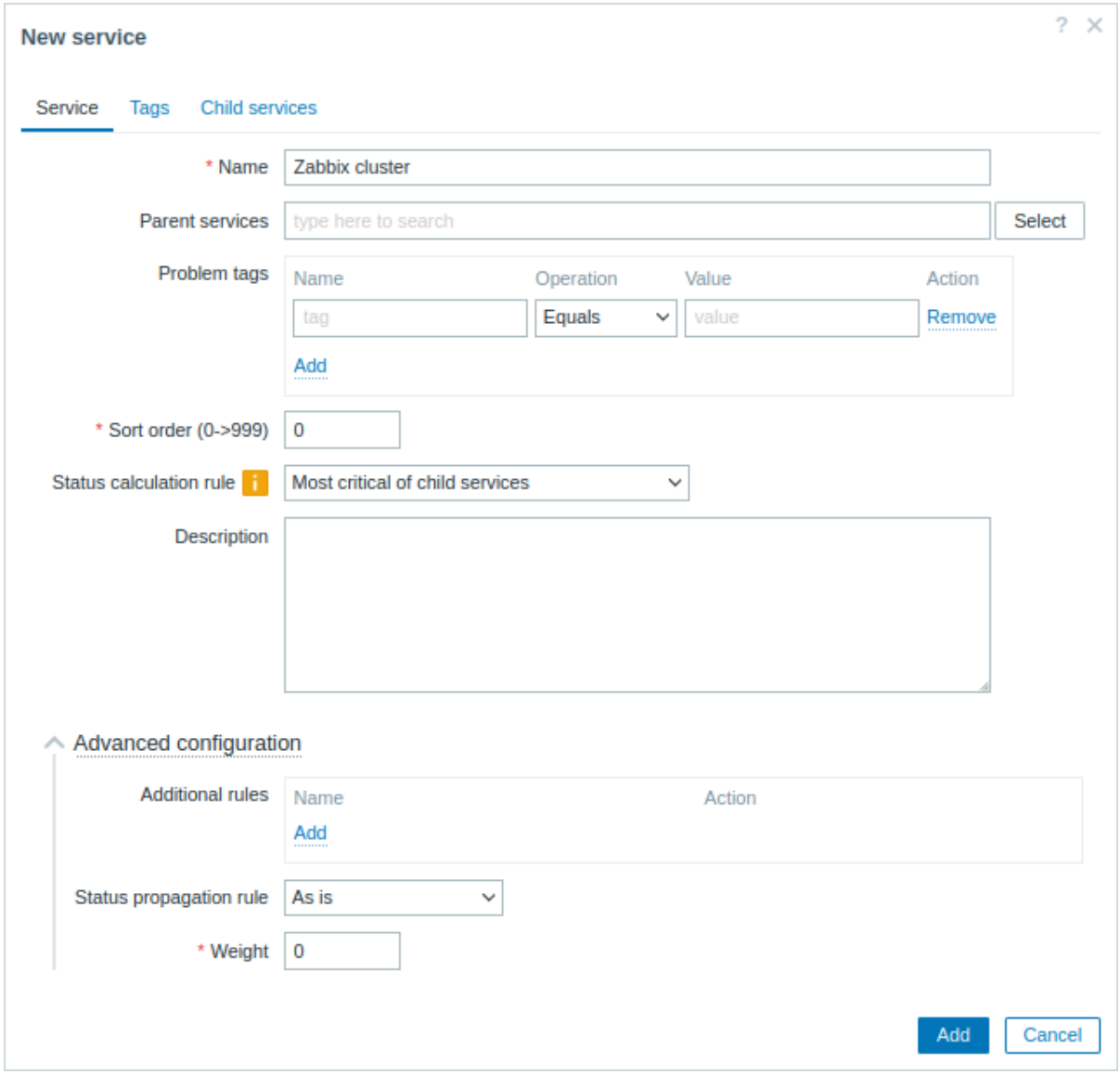
サービスツリー　次のステップは、サービスツリーを構築することです。この例では、インフラストラクチャは非常に基本的で、3 つのサービスで構成されています。*Zabbix cluster*（親）と、2 つの子サービス *Zabbix server node 1* および *Zabbix server node 2* です。

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

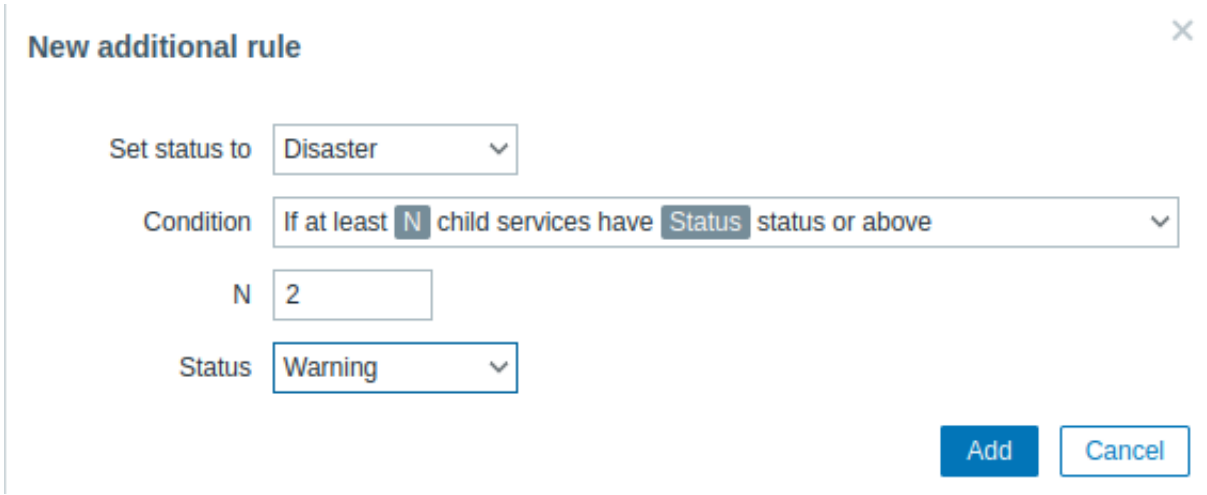
Services ページで、*Edit* モードを有効にして *Create service* をクリックします。



サービス設定ウィンドウで、名前に *Zabbix cluster* を入力し、*Advanced configuration* の見出しをクリックして詳細設定オプションを表示します。



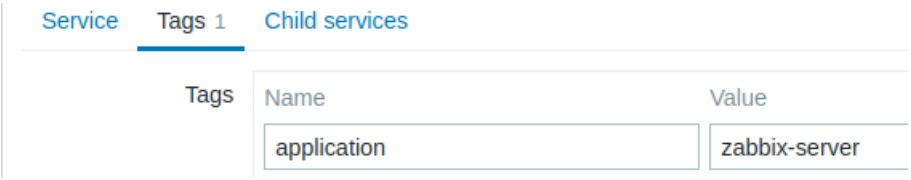
追加ルールを設定します。



Zabbix cluster には 2 つの子サービスがあり、それぞれ HA ノードに対応します。両方の HA ノードに少なくとも *Warning* 状態の問題がある場合、親サービスの状態は *Disaster* に設定される必要があります。これを実現するには、追加ルールを次のように設定します。

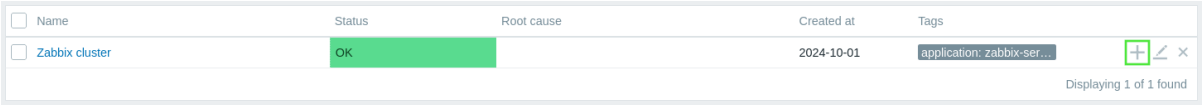
- 状態を次に設定: Disaster
- 条件: 少なくとも N 個の子サービスが Status 状態以上である場合
- N: 2
- Status: Warning

Tags タブに切り替え、タグ `application:zabbix-server` を追加します。このタグは、後でサービスアクションと SLA レポートで使用されます。



新しいサービスを保存します。

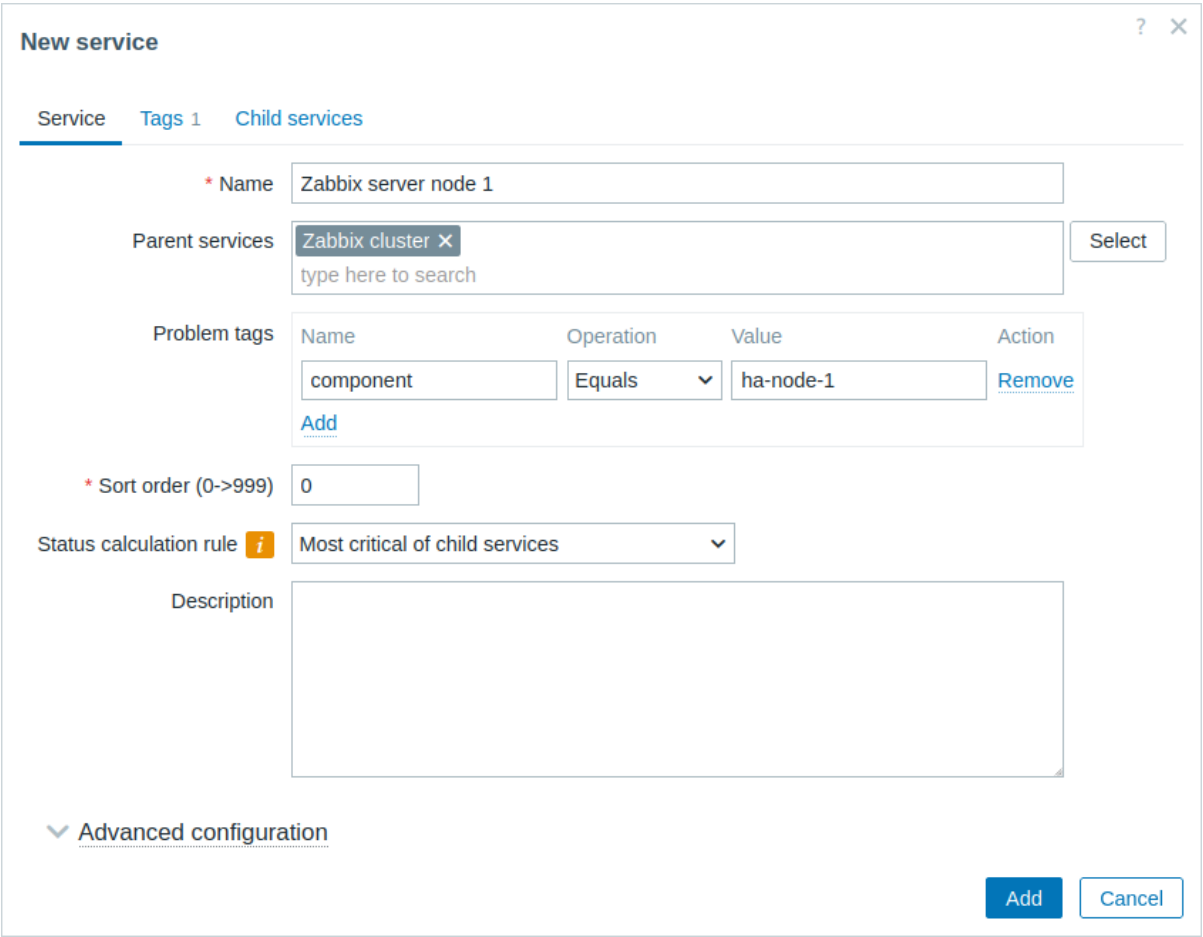
子サービスを追加するには、Zabbix クラスターサービスの横にあるプラスアイコンをクリックします（このアイコンは 編集モードでのみ表示されます）。



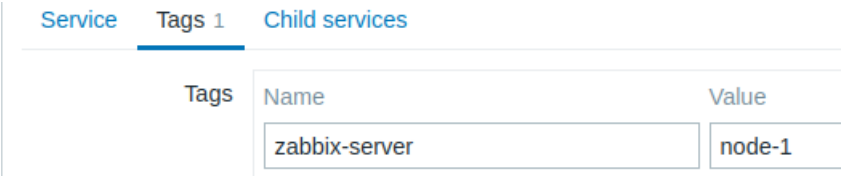
サービス設定ウィンドウで、名前に *Zabbix server node 1* を入力します。なお、親サービスのパラメータには、すでに *Zabbix cluster* が入力されています。

このサービスの可用性は、問題タグ `component:ha-node-1` が付いたホスト *HA node 1* の問題の影響を受けます。問題タグのパラメータに、次を入力します。

- 名前: component
- 操作: Equals
- 値: ha-node-1



Tags タブに切り替え、サービスタグ `zabbix-server:node-1` を追加します。このタグは、後でサービスアクションと SLA レポートで使用されます。



新しいサービスを保存します。

「Zabbix server node 2」という名前で Zabbix クラスタの別の子サービスを作成します。

問題タグを次のように設定します。

- 名前: component
- 演算子: Equals
- 値: ha-node-2

タグタブに切り替え、サービスタグ `zabbix-server:node-2` を追加します。

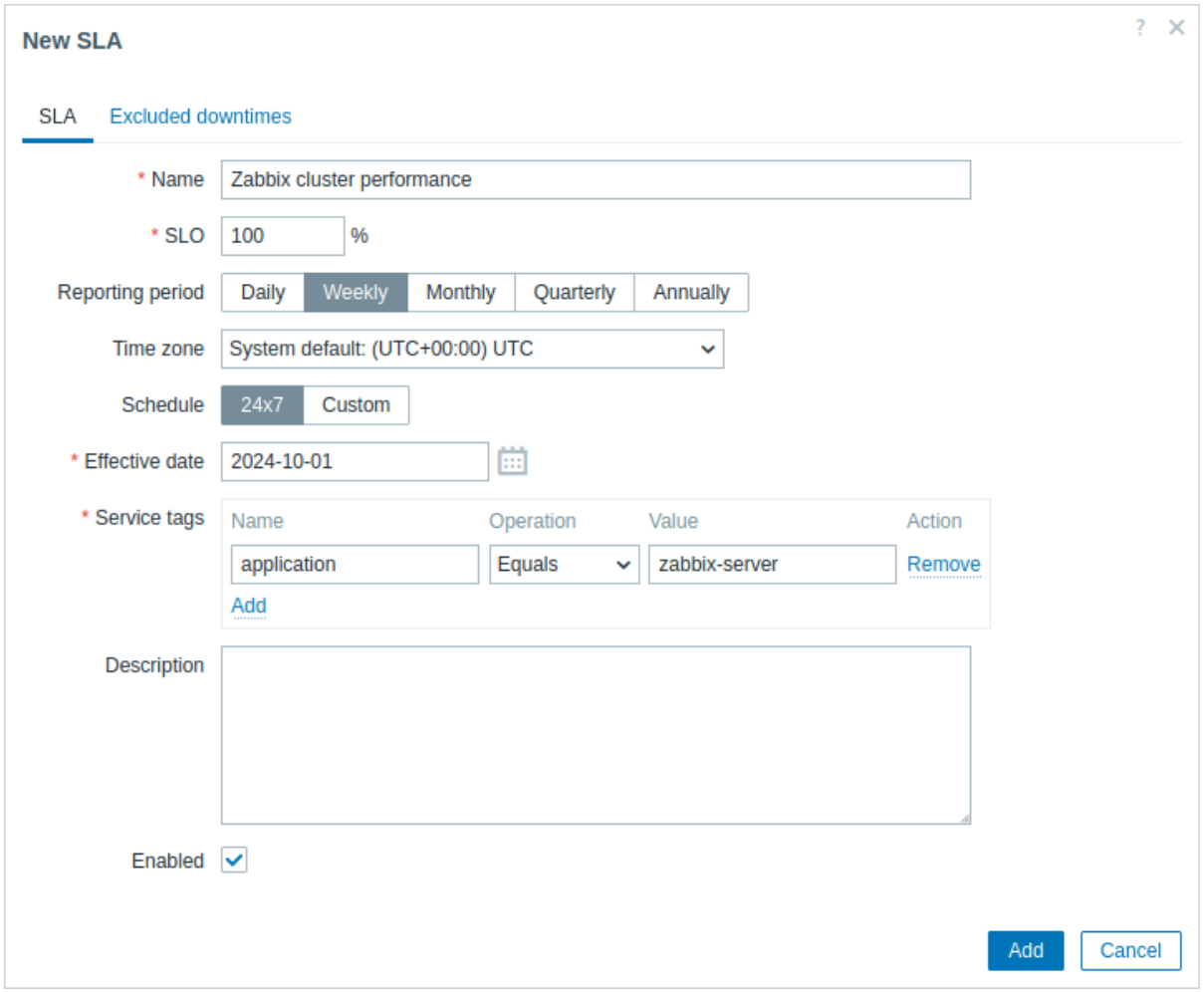
新しいサービスを保存します。

SLA　この例では、半年ごとに実施する 1 時間のメンテナンス期間を除き、Zabbix クラスターの期待されるパフォーマンスは 100% です。

まず、新しいサービスレベル合意を追加する必要があります。

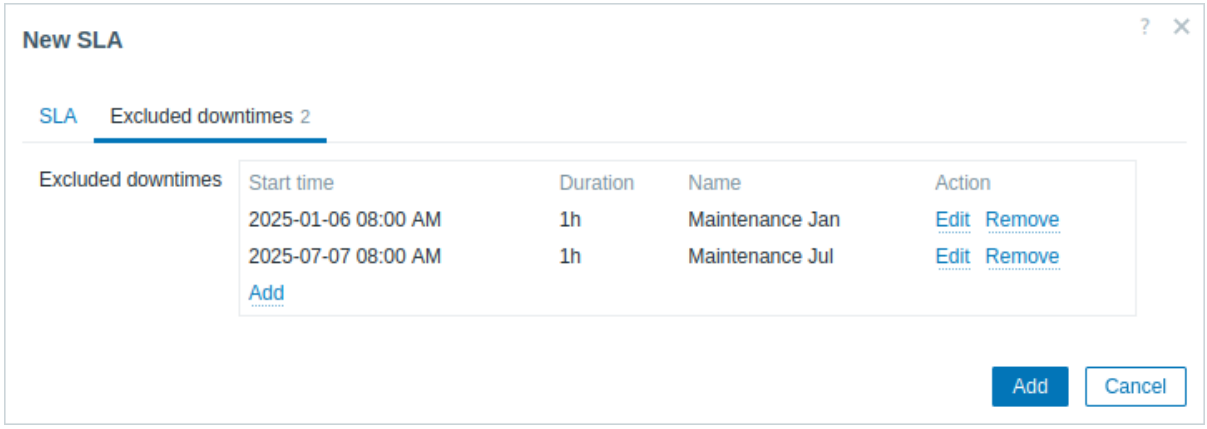
1. サービス > *SLA* に移動します。
2. *SLA* の作成をクリックします。
3. 名前に *Zabbix* クラスターのパフォーマンスと入力し、SLO を 100% に設定します。
   サービス Zabbix クラスターには、`application:zabbix-server` サービスタグが設定されています。
4. Zabbix クラスターのパフォーマンス測定にこの SLA を使用するには、サービスタグパラメータに次の値を指定します。

- 名前: application
- 操作: 等しい
- 値: zabbix-server



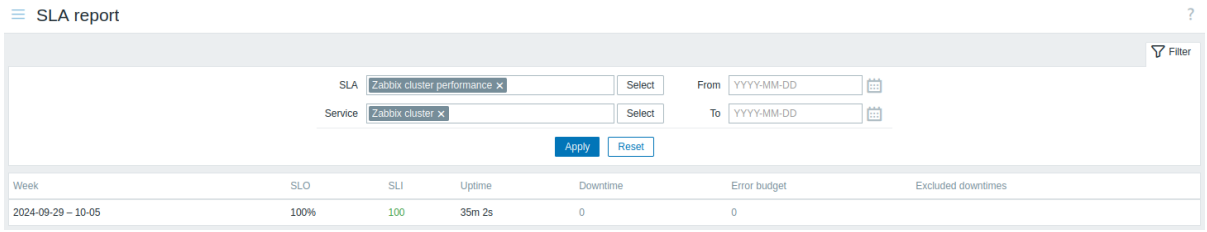
実際の環境では、レポート期間、タイムゾーン、開始日を更新したり、スケジュールを 24 時間 365 日からカスタム設定に変更したりすることもできます。
この例では、デフォルト設定で十分です。

5. 除外するダウンタイムタブに切り替え、スケジュールされたメンテナンス期間のダウンタイムを追加して、これらの期間を SLA の計算から除外します。
6. 除外するダウンタイムセクションで、追加リンクをクリックし、ダウンタイムの名前、予定開始時刻、期間を入力します。

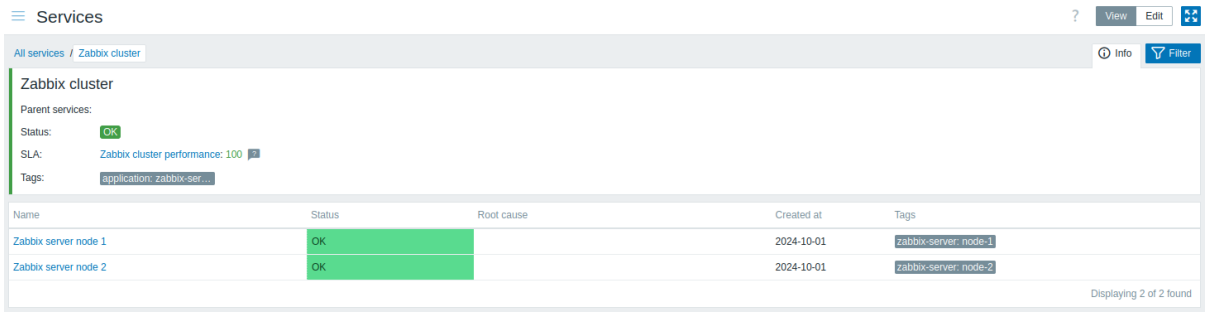


7. 追加をクリックして、新しい SLA を保存します。

SLA レポートセクションに切り替えると、Zabbix クラスターの SLA レポートを表示できます。



SLA 情報は、サービスセクションでも確認できます。



サービス監視のアップグレード

概要　Zabbix 6.0 では、サービス監視機能が大幅に再設計されました（変更点の一覧については、Zabbix 6.0.0 の新機能を参照してください）。

このページでは、以前の Zabbix バージョンで定義されたサービスと SLA が、Zabbix 6.0 以降へのアップグレード時にどのように変更されるかを説明します。

サービス　古いバージョンの Zabbix では、サービスにはソフト依存関係とハード依存関係の 2 種類がありました。アップグレード後は、すべての依存関係が同等になります。

サービス「Child service」が以前にハード依存関係を通じて「Parent service 1」に、さらにソフト依存関係を通じて「Parent service 2」にリンクされていた場合、アップグレード後の「Child service」には、親サービスとして「Parent service 1」と「Parent service 2」の 2 つが設定されます。

障害とサービスのトリガーベースのマッピングは、タグベースのマッピングに置き換えられました。Zabbix 6.0 以降では、サービス設定フォームに新しいパラメーター *Problem tags* が追加され、障害の一致条件として 1 つ以上のタグ名と値のペアを指定できます。サービスにリンクされているトリガーには、新しいタグ `ServiceLink : <trigger ID>:<trigger name>` が付与されます（タグ値は 32 文字に切り詰められます）。リンクされたサービスには、同じ値を持つ `ServiceLink` problem tag が付与されます。

状態計算ルール

「状態計算アルゴリズム」は、以下のルールに従ってアップグレードされます。

- 計算しない → ステータスを OK に設定
- 障害、少なくとも 1 つの子サービスに障害がある場合 → 子サービスの中で最も重大なもの

- 障害、すべての子サービスに障害がある場合 → すべての子サービスに障害がある場合は最も重大なもの

SLA　以前は、SLA ターゲットを各サービスごとに個別に定義する必要がありました。Zabbix 6.0 以降、SLA は独立したエンティティとなり、サービスのスケジュール、期待されるサービスレベル目標（SLO）、および計算から除外するダウンタイム期間の情報を含むようになりました。設定後、SLA はservice tags を通じて複数のサービスに割り当てることができます。

アップグレード時には、次の処理が行われます。

- 各サービスに対して定義されている同一の SLA はグループ化され、各グループごとに 1 つの SLA が作成されます。
- 影響を受ける各サービスには特別なタグ `SLA:<ID>` が付与され、同じタグが対応する SLA の *Service tags* パラメータにも指定されます。
- SLA レポートの新しいメトリクスであるサービス作成時刻は、既存のサービスについては 01/01/2000 00:00 に設定されます。

暗号化とセキュリティ

暗号化　Zabbix は、Transport Layer Security (TLS) プロトコル v.1.2 および 1.3（暗号ライブラリに依存）を使用して、Zabbix コンポーネント間の暗号化通信をサポートします。証明書ベースの暗号化と事前共有鍵ベースの暗号化がサポートされています。

暗号化は、次の接続に対して設定できます。

- Zabbix サーバー、Zabbix プロキシ、Zabbix エージェント、Zabbix web service、zabbix_sender および zabbix_get ユーティリティ間
- Zabbix データベースへの接続Zabbix Web インターフェースおよびサーバー/プロキシから
- Zabbix Web インターフェースと Zabbix サーバー間

暗号化はオプションであり、個々のコンポーネントごとに設定できます。

- 一部のプロキシおよびエージェントは、サーバーとの通信に証明書ベースの暗号化を使用するよう設定でき、他のものは事前共有鍵ベースの暗号化を使用でき、さらに別のものは暗号化されていない通信のまま継続できます（従来どおり）。
- サーバー（プロキシ）は、ホストごとに異なる暗号化設定を使用できます。

Zabbix のデーモンプログラムは、暗号化された受信接続と暗号化されていない受信接続の両方に対して 1 つの待ち受けポートを使用します。暗号化を追加しても、ファイアウォールで新しいポートを開く必要はありません。

制限事項

- 秘密鍵は、起動時に Zabbix コンポーネントから読み取り可能なファイルに平文で保存されます。
- 事前共有鍵は Zabbix Web インターフェースで入力され、Zabbix データベースに平文で保存されます。
- 組み込みの暗号化は、Zabbix Web インターフェースを実行している Web サーバーとユーザーの Web ブラウザ間の通信を保護しません。
- 現在、暗号化された各接続は完全な TLS ハンドシェイクで開始され、セッションキャッシュやチケットは実装されていません。
- 暗号化を追加すると、ネットワーク遅延に応じてアイテムのチェックとアクションにかかる時間が増加します。
  - たとえば、パケット遅延が 100ms の場合、TCP 接続を開いて暗号化されていない要求を送信するのに約 200ms かかります。暗号化を使用すると、TLS 接続の確立に約 1000 ms が追加されます。
  - タイムアウトを増やす必要がある場合があります。そうしないと、エージェント上でリモートスクリプトを実行する一部のアイテムやアクションは、暗号化されていない接続では動作しても、暗号化された接続ではタイムアウトで失敗することがあります。
- ネットワークディスカバリ では暗号化はサポートされていません。ネットワークディスカバリによって実行される Zabbix エージェントのチェックは暗号化されず、Zabbix エージェントが暗号化されていない接続を拒否するように設定されている場合、これらのチェックは成功しません。

暗号化サポート付きで **Zabbix** をコンパイルする　暗号化をサポートするには、Zabbix を以下のいずれかのサポート対象暗号ライブラリとともにコンパイルおよびリンクする必要があります。

- GnuTLS - バージョン 3.1.18 以降
- OpenSSL - バージョン 1.0.1、1.0.2、1.1.0、1.1.1、3.0.x - 3.5.x
- LibreSSL - バージョン 2.7.4、2.8.2 でテスト済み:
  - LibreSSL 2.6.x はサポートされていません
  - LibreSSL は OpenSSL の互換代替としてサポートされています。新しい `tls_*()` LibreSSL 固有の API 関数は使用されません。LibreSSL でコンパイルされた Zabbix コンポーネントでは PSK を使用できず、証明書のみ使用できます。

> **Note:**
> Zabbix の Web インターフェースで SSL を設定する方法については、こちらのベストプラクティスを参照してください。

ライブラリは、`configure` スクリプトに対応するオプションを指定して選択します。

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]`（LibreSSL にも使用）

例えば、*OpenSSL* を使用してサーバーとエージェントのソースを設定するには、次のように指定します。

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

異なる Zabbix コンポーネントを、異なる暗号ライブラリでコンパイルすることもできます（例えば、サーバーを *OpenSSL*、エージェントを *GnuTLS* でコンパイルするなど）。

Attention:
事前共有鍵（PSK）を使用する予定がある場合は、PSK を使用する Zabbix コンポーネントで *GnuTLS* または *OpenSSL 1.1.0*（またはそれ以降）のライブラリを使用することを検討してください。*GnuTLS* および *OpenSSL 1.1.0* のライブラリは、Perfect Forward Secrecyに対応した PSK 暗号スイートをサポートしています。*OpenSSL* ライブラリの古いバージョン（1.0.1、1.0.2c）も PSK をサポートしていますが、使用可能な PSK 暗号スイートは Perfect Forward Secrecy を提供しません。

接続の暗号化管理　Zabbix の接続では、次の暗号化方式を使用できます。

- 暗号化なし（デフォルト）
- RSA 証明書ベースの暗号化
- PSK ベースの暗号化

Zabbix コンポーネント間の暗号化を指定するために使用される重要なパラメータは 2 つあります。

- TLSConnect - 送信接続で使用する暗号化方式を指定します（暗号化なし、PSK、または証明書）
- TLSAccept - 受信接続で許可する接続の種類を指定します（暗号化なし、PSK、または証明書）。1 つ以上の値を指定できます。

`TLSConnect` は、Zabbix プロキシ（アクティブモードでは、サーバーへの接続のみを指定）および Zabbix エージェント（アクティブチェック）の設定ファイルで使用されます。Zabbix の Web インターフェースでは、`TLSConnect` に相当する項目は、データ収集 > ホスト > <some host> > 暗号化タブの ホストへの接続フィールド、および 管理 > プロキシ > <some proxy> > 暗号化タブの プロキシへの接続フィールドです。設定された接続の暗号化方式に失敗した場合、他の暗号化方式は試行されません。

`TLSAccept` は、Zabbix プロキシ（パッシブモードでは、サーバーからの接続のみを指定）および Zabbix エージェント（パッシブチェック）の設定ファイルで使用されます。Zabbix の Web インターフェースでは、TLSAccept に相当する項目は、データ収集 > ホスト > <some host> > 暗号化タブの ホストからの接続フィールド、および 管理 > プロキシ > <some proxy> > 暗号化タブの プロキシからの接続フィールドです。

通常、受信接続には 1 種類の暗号化方式のみを設定します。ただし、最小限のダウンタイムとロールバックの可能性を確保しながら、暗号化方式を変更したい場合があります。たとえば、暗号化なしから証明書ベースの暗号化へ切り替える場合です。これを実現するには、次の手順を実行します。

- エージェントの設定ファイルで `TLSAccept=unencrypted,cert` を設定し、Zabbix エージェントを再起動します。
- 証明書を使用して `zabbix_get` でエージェントへの接続をテストします。正常に動作する場合は、Zabbix の Web インターフェースで データ収集 > ホスト > <some host> > 暗号化タブを開き、ホストへの接続を 証明書に設定して、そのエージェントの暗号化を再設定できます。
- サーバーの設定キャッシュが更新されると（そのホストがプロキシによって監視されている場合は、プロキシの設定も更新されると）、そのエージェントへの接続が暗号化されます。
- すべてが期待どおりに動作する場合は、エージェントの設定ファイルで `TLSAccept=cert` を設定し、Zabbix エージェントを再起動します。これで、エージェントは証明書ベースの暗号化接続のみを受け入れるようになります。暗号化されていない接続および PSK ベースの接続は拒否されます。

サーバーとプロキシでも同様に動作します。Zabbix の Web インターフェースでホスト設定の ホストからの接続を「証明書」に設定すると、エージェント（アクティブチェック）および zabbix_sender（トラッパーアイテム）からの証明書ベースの暗号化接続のみが受け入れられます。

通常、受信接続と送信接続には同じ暗号化方式を使用するか、どちらも暗号化しないように設定します。ただし、技術的には非対称に設定することも可能です。たとえば、受信接続には証明書ベースの暗号化を使用し、送信接続には PSK ベースの暗号化を使用できます。

各ホストの暗号化設定は、Zabbix の Web インターフェースの データ収集 > ホストにある エージェントの暗号化列に表示されます。
例：

例	ホストへの接続	ホストから許可される接続	ホストから拒否される
NONE	暗号化なし	暗号化なし	暗号化された接続、証
CERT NONE PSK CERT	暗号化、証明書ベース	暗号化、証明書ベース	暗号化なしおよび PSK
PSK NONE PSK CERT	暗号化、PSK ベース	暗号化、PSK ベース	暗号化なしおよび証明書

例	ホストへの接続	ホストから許可される接続	ホストから拒否される
PSK NONE PSK CERT	暗号化、PSK ベース	暗号化なしおよび PSK ベースの暗号化接続	証明書ベースの暗号化
CERT NONE PSK CERT	暗号化、証明書ベース	暗号化なし、PSK または証明書ベースの暗号化接続	-

Attention:
接続はデフォルトでは暗号化されません。各ホストおよびプロキシに対して、個別に暗号化を設定する必要があります。

zabbix_get と **zabbix_sender** の暗号化対応　暗号化を使用した場合の使い方については、zabbix_get とzabbix_sender の man ページを参照してください。

暗号スイート　暗号スイートは、デフォルトでは Zabbix の起動時に内部で設定されます。

また、GnuTLS と OpenSSL ではユーザーが設定した暗号スイートもサポートされています。ユーザーは設定 により、セキュリティポリシーに従って暗号スイートを構成できます。この機能の使用は任意です（組み込みのデフォルト暗号スイートは引き続き動作します）。

デフォルト設定でコンパイルされた暗号ライブラリでは、Zabbix の組み込みルールにより通常、次の暗号スイートになります（優先度の高い順から低い順）：

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

ユーザー設定の **ciphersuite**　組み込みの ciphersuite 選択基準は、ユーザー設定の ciphersuite によって上書きできます。

Attention:
ユーザー設定の ciphersuite は、TLS ciphersuite、そのセキュリティ、設定ミスの影響を理解しており、TLS のトラブルシューティングに慣れている上級ユーザー向けの機能です。

組み込みの ciphersuite 選択基準は、次のパラメーターを使用して上書きできます。

上書き範囲	パラメーター	値	説明
証明書の ciphersuite 選択	TLSCipherCert13	TLS 1.3 プロトコル向けの有効な OpenSSL 1.1.1 cipher strings（その値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 の証明書ベースの ciphersuite 選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherCert	TLS 1.2 向けの有効な OpenSSL cipher strings または有効な GnuTLS priority strings。その値は、それぞれ `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）の証明書ベースの ciphersuite 選択基準
PSK の ciphersuite 選択	TLSCipherPSK13	TLS 1.3 プロトコル向けの有効な OpenSSL 1.1.1 cipher strings（その値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 の PSK ベースの ciphersuite 選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherPSK	TLS 1.2 向けの有効な OpenSSL cipher strings または有効な GnuTLS priority strings。その値は、それぞれ `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）の PSK ベースの ciphersuite 選択基準
証明書と PSK の統合 ciphersuite リスト	TLSCipherAll13	TLS 1.3 プロトコル向けの有効な OpenSSL 1.1.1 cipher strings（その値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 の ciphersuite 選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherAll	TLS 1.2 向けの有効な OpenSSL cipher strings または有効な GnuTLS priority strings。その値は、それぞれ `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）の ciphersuite 選択基準

zabbix_get および zabbix_sender ユーティリティで ciphersuite 選択を上書きするには、コマンドラインパラメーターを使用します。

- `--tls-cipher13`
- `--tls-cipher`

新しいパラメーターは任意です。パラメーターが指定されていない場合は、内部のデフォルト値が使用されます。パラメーターが定義されている場合は、空にできません。

crypto ライブラリで TLSCipher* 値の設定に失敗した場合、サーバー、プロキシ、またはエージェントは起動せず、エラーがログに記録されます。

各パラメーターがいつ適用されるかを理解することが重要です。

外向き接続

最も単純なケースは外向き接続です。

- 証明書を使用する外向き接続の場合 - TLSCipherCert13 または TLSCipherCert を使用します
- PSK を使用する外向き接続の場合 - TLSCipherPSK13 または TLSCipherPSK を使用します
- zabbix_get および zabbix_sender ユーティリティの場合、コマンドラインパラメータ `--tls-cipher13` または `--tls-cipher` を使用できます（暗号化は `--tls-connect` パラメータで明確に指定されます）

受信接続

受信接続では、ルールがコンポーネントと設定ごとに異なるため、少し複雑になります。

Zabbix エージェントの場合:

Agent connection setup	Cipher configuration
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13

Agent connection setup	Cipher configuration
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Zabbix サーバーおよび プロキシの場合:

Connection setup	Cipher configuration
Outgoing connections using PSK	TLSCipherPSK, TLSCipherPSK13
Incoming connections using certificates	TLSCipherAll, TLSCipherAll13
Incoming connections using PSK if server has no certificate	TLSCipherPSK, TLSCipherPSK13
Incoming connections using PSK if server has certificate	TLSCipherAll, TLSCipherAll13

上の 2 つの表から、次のようなパターンが見て取れます。

- TLSCipherAll と TLSCipherAll13 は、証明書ベース および PSK ベースの暗号スイートを組み合わせた一覧を使用する場合にのみ指定できます。これが該当するのは次の 2 つの場合です。証明書が設定されたサーバー（プロキシ）(PSK 暗号スイートは、暗号ライブラリが PSK をサポートしている場合、プロキシでは常に設定されます)、および証明書ベースと PSK ベースの両方の受信接続を受け入れるように設定されたエージェントです。
- その他の場合は、TLSCipherCert* および/または TLSCipherPSK* で十分です。

以下の表は、`TLSCipher*` の組み込みデフォルト値を示しています。これらは、独自のカスタム値を設定する際の出発点として適しています。

Parameter	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Parameter	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] デフォルト値は、古い OpenSSL バージョン（1.0.1、1.0.2、1.1.0）、LibreSSL、または OpenSSL が PSK サポートなしでコンパイルされている場合には異なります。

ユーザー設定の暗号スイートの例

以下に、ユーザー設定の暗号スイートの例を示します。

- 暗号文字列のテストと PFS 暗号スイートのみを許可する
- AES128 から AES256 への切り替え

暗号スイートのテストと PFS 暗号スイートのみの許可

選択された暗号スイートを確認するには、設定ファイルで DebugLevel=4 を設定するか、zabbix_sender に `-vv` オプションを指定します。

目的の暗号スイートを取得するまで、`TLSCipher*` パラメータを試行錯誤する必要がある場合があります。`TLSCipher*` パラメータを調整するだけで Zabbix サーバー、プロキシ、またはエージェントを何度も再起動するのは不便です。より便利な方法として、zabbix_sender または `openssl` コマンドを使用できます。ここでは両方の方法を説明します。

1. zabbix_sender を使用する場合

まず、`zabbix_agentd.conf` ファイルと同じ構文で、例えば `/home/zabbix/test.conf` にテスト設定ファイルを作成します。

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

この例では、有効な CA 証明書、エージェント証明書、および PSK が必要です。証明書と PSK のファイルパスおよびファイル名は、環境に合わせて調整してください。

証明書を使用せず、PSK のみを使用する場合は、より簡単なテストファイルを作成できます。

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

zabbix_sender を実行すると、選択された暗号スイートを確認できます（OpenSSL 1.1.d でコンパイルした例）。

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

ここでは、デフォルトで選択される暗号スイートを確認できます。これらのデフォルト値は、古いバージョンの OpenSSL（1.0.1 以降）を使用しているシステム上で動作する Zabbix エージェントとの相互運用性を確保するために選択されています。

新しいシステムでは、少数の暗号スイートのみを許可してセキュリティを強化できます。例えば、PFS（Perfect Forward Secrecy、完全前方秘匿性）を備えた暗号スイートのみを許可できます。ここでは、PFS をサポートする暗号スイートのみを許可するように TLSCipher* パラメータを設定します。

> **Attention:**
> PSK を使用する場合、OpenSSL 1.0.1 および 1.0.2 を使用するシステムとは相互運用できなくなります。証明書ベースの暗号化は動作するはずです。

test.conf 設定ファイルに次の 2 行を追加します。

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

その後、もう一度テストします。

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

「certificate ciphersuites」と「PSK ciphersuites」のリストが変わり、以前より短くなっていることが分かります。想定どおり、TLS 1.3 の暗号スイートと TLS 1.2 の ECDHE-* 暗号スイートのみが含まれています。

2. TLSCipherAll および TLSCipherAll13 は zabbix_sender ではテストできません。これらは、上の例に示した「certificate and PSK ciphersuites」の値には影響しません。TLSCipherAll および TLSCipherAll13 を調整するには、エージェント、プロキシ、またはサーバーで試行する必要があります。

したがって、PFS 暗号スイートのみを許可するには、証明書が設定され、さらにエージェントに PSK も設定されている場合、次の最大 3 つのパラメータを zabbix_agentd.conf、zabbix_proxy.conf、zabbix_server.conf に追加する必要があります。

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

Zabbix 環境で PSK ベースの暗号化のみを使用し、証明書を使用しない場合は、次の 1 つだけを設定します。

```
TLSCipherPSK=kECDHEPSK+AES128
```

仕組みを理解したら、Zabbix の外部でも openssl コマンドを使用して暗号スイートの選択をテストできます。ここでは、3 つすべての TLSCipher* パラメータ値をテストします。

```
$ openssl ciphers EECDH+aRSA+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers EECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

より詳細な出力を得るには、-V オプションを指定して openssl ciphers を使用するとよいでしょう。

```
$ openssl ciphers -V EECDH+aRSA+AES128:kECDHEPSK+AES128
          0x13,0x02 - TLS_AES_256_GCM_SHA384  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(256) Mac=AEAD
          0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256
          0x13,0x01 - TLS_AES_128_GCM_SHA256  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(128) Mac=AEAD
          0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
          0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
          0xC0,0x13 - ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
          0xC0,0x37 - ECDHE-PSK-AES128-CBC-SHA256 TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA256
          0xC0,0x35 - ECDHE-PSK-AES128-CBC-SHA TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA1
```

同様に、GnuTLS の優先度文字列もテストできます。

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

AES128 から AES256 への切り替え

Zabbix では、データの組み込みデフォルトとして AES128 を使用します。ここでは、証明書を使用しており、OpenSSL 1.1.1 で AES256 に切り替えたいとします。

これは、zabbix_server.conf にそれぞれのパラメータを追加することで実現できます。

```
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
証明書関連の暗号スイートのみが使用されますが、TLSCipherPSK* パラメータも定義されています。これは、より広い相互運用性のために安全性の低い暗号を含むデフォルト値を避けるためです。PSK の暗号スイートは、サーバー/プロキシでは完全には無効化できません。

また、zabbix_agentd.conf では次のように設定します。

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

暗号化の問題のトラブルシューティング　一般的な推奨事項:

- 問題のケースで、どのコンポーネントが TLS クライアントとして動作し、どのコンポーネントが TLS サーバーとして動作するかを把握することから始めてください。
 Zabbix サーバー、プロキシ、エージェントは、相互のやり取りに応じて、いずれも TLS サーバーおよびクライアントとして動作できます。
 たとえば、Zabbix サーバーがパッシブチェックのためにエージェントへ接続する場合、TLS クライアントとして動作します。エージェントは TLS サーバーの役割を担います。
 Zabbix エージェントがプロキシからアクティブチェックの一覧を要求する場合、TLS クライアントとして動作します。プロキシは TLS サーバーの役割を担います。
 zabbix_get および zabbix_sender ユーティリティは、常に TLS クライアントとして動作します。
- Zabbix は相互認証を使用します。
 それぞれの側が相手を検証し、接続を拒否する場合があります。
 たとえば、Zabbix サーバーがエージェントへ接続する際、エージェントの証明書が無効であれば、接続を直ちに閉じることがあります。逆に、Zabbix エージェントがサーバーからの接続を受け入れる場合でも、サーバーがエージェントから信頼されていなければ接続を閉じることがあります。
- TLS クライアント側と TLS サーバー側の両方でログファイルを確認してください。
 接続を拒否した側には、拒否した正確な理由が記録されている場合があります。もう一方の側には、より一般的なエラー（例: "Connection closed by peer", "connection was non-properly terminated"）が表示されることがよくあります。
- 暗号化の設定ミスにより、実際の原因をまったく示さない紛らわしいエラーメッセージが出ることがあります。
 以下の節では、トラブルシューティングに役立つ可能性のあるメッセージと原因の候補を、網羅的ではありませんがまとめています。
 なお、異なる暗号ライブラリ（OpenSSL、GnuTLS）は、同じ問題状況でも異なるエラーメッセージを出すことがよくあります。
 また、エラーメッセージは、両側で使用されている暗号ライブラリの組み合わせによっても変わることがあります。

接続タイプまたは権限の問題のトラブルシューティング　サーバーは PSK でエージェントに接続するよう設定されていますが、エージェントは暗号化されていない接続のみを受け入れます

サーバーまたはプロキシのログで（*GnuTLS* 3.3.16 の場合）

```
エージェントからの値の取得に失敗しました: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 TLS接続が正しく終了されませんでした。
```

サーバーまたはプロキシのログで（*OpenSSL* 1.0.2c の場合）

```
エージェントからの値の取得に失敗しました: TCP接続は成功しましたが、[[127.0.0.1]:10050] へのTLSを確立できま
    相手側によって接続が閉じられました。許可された接続タイプとアクセス権を確認してください
```

一方が証明書で接続し、もう一方が PSK のみを受け入れる、またはその逆の場合

任意のログ（*GnuTLS* 使用時）:

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21 Could not negotiate a supported cipher suite.
```

任意のログ（*OpenSSL* 1.0.2c 使用時）:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

TLS サポート付きでコンパイルされた **Zabbix sender** を、**TLS** 非対応でコンパイルされた **Zabbix** サーバー/プロキシにデータ送信しようとしています　接続側のログ:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by pee
```

Windows:

```
...OpenSSL library (version OpenSSL 1.1.1a  20 Nov 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

受信側のログ:

```
...failed to accept an incoming connection: from 127.0.0.1: TLS のサポートはコンパイルされていません
```

一方は **PSK** で接続するが、もう一方は **LibreSSL** を使用しているか、暗号化サポートなしでコンパイルされている　LibreSSL は PSK をサポートしていません。

接続側のログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

受信側のログ:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Zabbix Web インターフェース:

```
Get value from エージェント failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_cor
```

一方は **PSK** で接続するが、もう一方は **PSK** サポートを無効にした **OpenSSL** を使用している　接続側のログ:

```
...TCP は成功しましたが、[[192.168.1.2]:10050] への TLS を確立できません: SSL_connect() が結果コード SSL_
```

受信側のログ:

```
...受信した接続を許可できませんでした: from 192.168.1.2: TLS handshake が結果コード 1 を設定しました: fil
```

証明書を使用した暗号化

概要　Zabbix は、公開の証明機関 (CA) または社内の証明機関 (CA) によって署名された、PEM 形式の RSA 証明書を使用できます。

証明書の検証は、事前に設定された CA 証明書に対して行われます。必要に応じて、証明書失効リスト (CRL) を使用できます。

各 Zabbix コンポーネントに設定できる証明書は 1 つだけです。

内部 CA のセットアップと運用、証明書要求の生成と署名、証明書の失効については、OpenSSL PKI Tutorial v2.0 などのチュートリアルを参照してください。

証明書拡張は慎重に検討し、テストしてください。詳細については、X.509 v3 証明書拡張の使用に関する制限 を参照してください。

証明書設定パラメータ　以下の設定パラメータは、Zabbix コンポーネントで証明書を設定するためにサポートされています。

Parameter	Mandatory	Description
TLSCAFile	yes	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルの完全なパス名。 複数のメンバーを持つ証明書チェーンを使用する場合は、下位レベルの CA 証明書を先に、その後に上位レベルの CA 証明書を配置してください。 複数の CA の証明書を 1 つのファイルに含めることができます。
TLSCRLFile	no	証明書失効リスト (CRL) を含むファイルの完全なパス名。
TLSCertFile	yes	証明書を含むファイルの完全なパス名。 複数のメンバーを持つ証明書チェーンを使用する場合は、サーバー、プロキシ、またはエージェントの証明書を先頭に置き、その後に下位レベルの CA 証明書、最後に上位レベルの CA 証明書を配置してください。
TLSKeyFile	yes	秘密鍵を含むファイルの完全なパス名。 このファイルは、適切なアクセス権を設定し、Zabbix ユーザー のみが読み取れるようにしてください。
TLSServerCertIssuer	no	許可されるサーバー証明書の発行者。
TLSServerCertSubject	no	許可されるサーバー証明書のサブジェクト。

設定例　必要な証明書を設定したら、Zabbix コンポーネントが証明書ベースの暗号化を使用するように設定します。

以下に、設定手順の詳細を示します。

- Zabbix サーバー
- Zabbix プロキシ
- Zabbix エージェント

Zabbix サーバー

1. CA 証明書ファイルを準備します。

ピア証明書を検証するには、Zabbix サーバーが最上位の自己署名ルート CA 証明書を含むファイルにアクセスできる必要があります。たとえば、2 つの独立したルート CA の証明書が必要な場合は、/home/zabbix/zabbix_ca_file.crt にファイルを作成し、証明書を格納します。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Zabbix サーバーの証明書または証明書チェーンをファイルに格納します。たとえば、/home/zabbix/zabbix_server.crt に格納します。最初の証明書は Zabbix サーバーの証明書で、その後に中間 CA 証明書を続けます。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
```

```
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJMOHw==
-----END CERTIFICATE-----
```

Note:
クライアント証明書とサーバー証明書の両方で、証明書の検証プロセスに影響を与えないよう、上記で説明した属性のみを使用してください。たとえば、*X509v3 Subject Alternative Name* または *Netscape Cert Type* 拡張が使用されている場合、OpenSSL で暗号化接続の確立に失敗することがあります。詳細については、X.509 v3 証明書拡張の使用に関する制限を参照してください。

3. Zabbix サーバーの秘密鍵をファイルに格納します。たとえば、/home/zabbix/zabbix_server.key に格納します。

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Zabbix サーバー設定ファイルの TLS 設定パラメータを編集します。

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix プロキシ

1. Zabbix サーバーのセクションで説明されているとおり、トップレベル CA 証明書、Zabbix プロキシの証明書または証明書チェーン、および秘密鍵を含むファイルを準備します。次に、Zabbix のプロキシ設定ファイルで、TLSCAFile、TLSCertFile、TLSKeyFile

パラメータを適切に編集します。

2. Zabbix のプロキシ設定ファイルで、追加の TLS パラメータを編集します。

- アクティブプロキシの場合: TLSConnect=cert
- パッシブプロキシの場合: TLSAccept=cert

> **Note:**
> プロキシのセキュリティを向上させるため、TLSServerCertIssuer および TLSServerCertSubject パラメータも設定できます。詳細については、許可する証明書の発行者とサブジェクトの制限を参照してください。

最終的なプロキシ設定ファイルの TLS パラメータは、次のようになります。

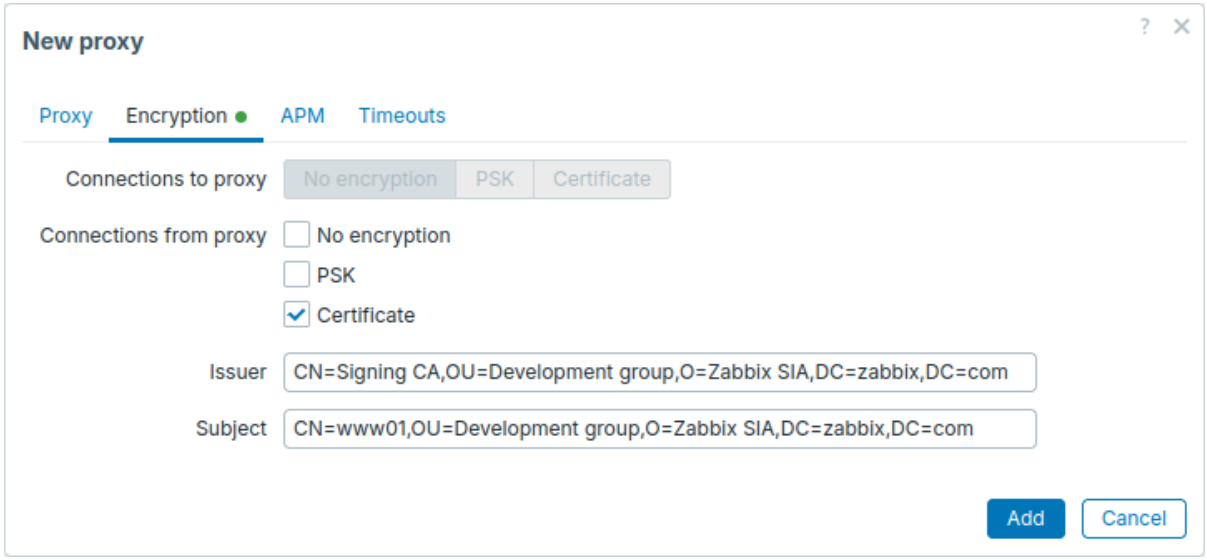
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Zabbix の Web インターフェースで、このプロキシの暗号化を設定します。

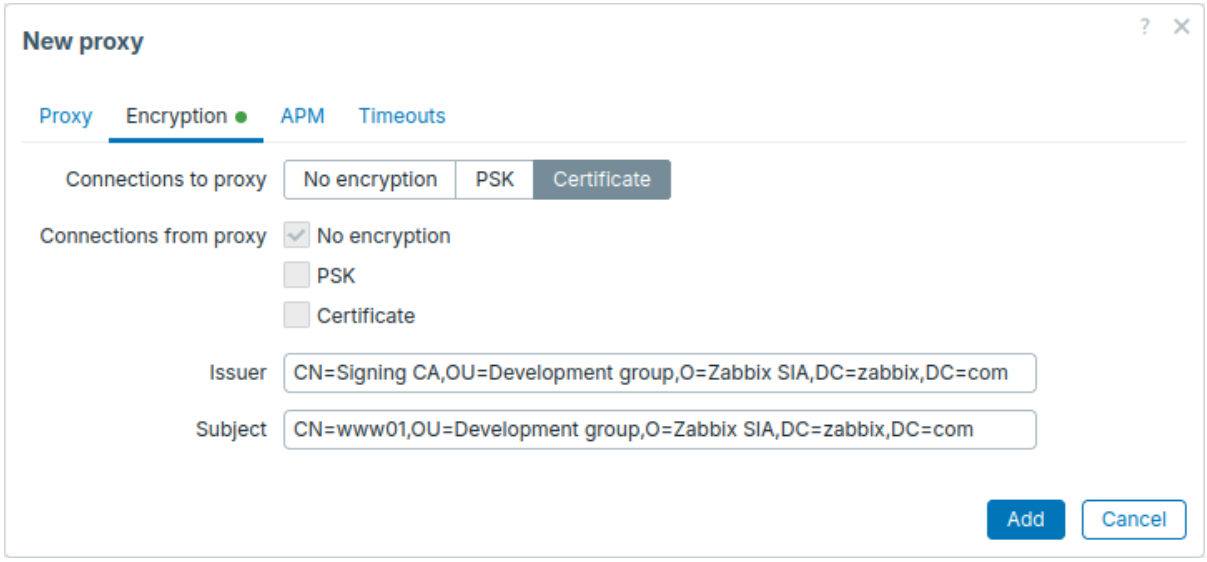
- 管理 > プロキシに移動します。
- プロキシを選択し、暗号化タブをクリックします。

以下の例では、発行者フィールドとサブジェクトフィールドに値が入力されています。これらのフィールドを使用する理由と方法の詳細については、許可する証明書の発行者とサブジェクトの制限を参照してください。

アクティブプロキシの場合:



パッシブプロキシの場合:



Zabbix エージェント

1. Zabbix サーバーのセクションで説明されているように、トップレベル CA 証明書、Zabbix エージェントの証明書/証明書チェーン、および秘密鍵を含むファイルを準備します。次に、Zabbix のエージェント設定ファイルで、TLSCAFile、TLSCertFile、TLSKeyFile パラメータを適切に編集します。

2. Zabbix のエージェント設定ファイルで、追加の TLS パラメータを編集します。

- アクティブエージェントの場合: TLSConnect=cert
- パッシブエージェントの場合: TLSAccept=cert

Note:
エージェントのセキュリティを向上させるには、TLSServerCertIssuer および TLSServerCertSubject パラメータを設定できます。詳細については、許可する証明書の発行者とサブジェクトの制限を参照してください。

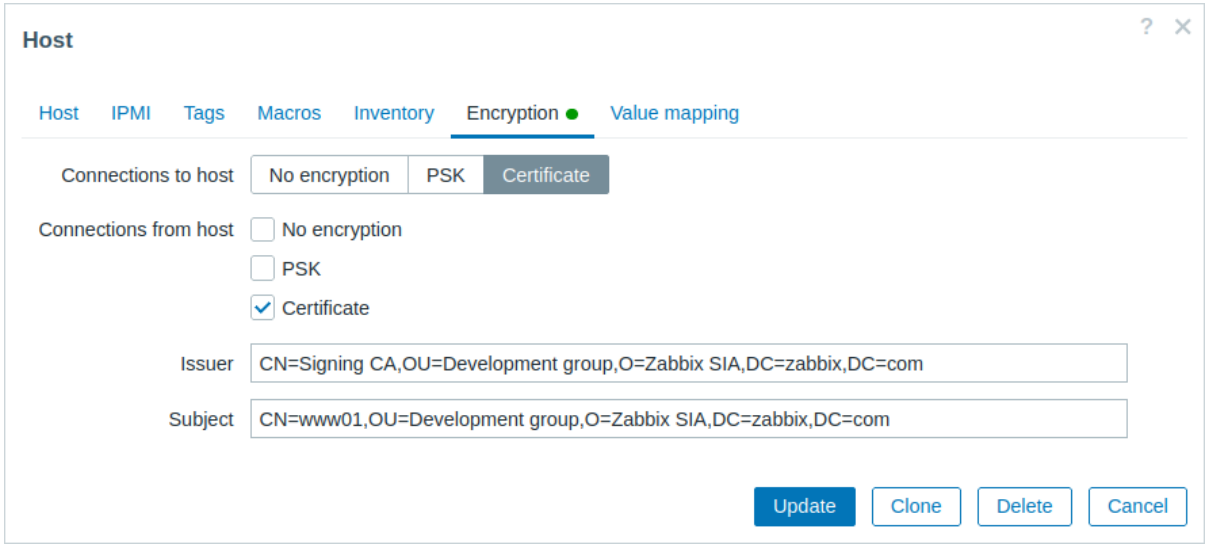
最終的なエージェント設定ファイルの TLS パラメータは、次のようになります。この例では、ホストがプロキシによって監視されることを前提としているため、証明書の Subject として指定されています。

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. このエージェントによって監視されるホストの暗号化を Zabbix Web インターフェースで設定します。

- データ収集 > ホストに移動します。
- ホストを選択し、暗号化タブをクリックします。

以下の例では、発行者フィールドとサブジェクトフィールドに値が入力されています。これらのフィールドを使用する理由と方法の詳細については、許可する証明書の発行者とサブジェクトの制限を参照してください。



Zabbix Web サービス

1. Zabbix サーバーのセクションで説明されているとおり、トップレベル CA 証明書、Zabbix Web サービスの証明書/証明書チェーン、および秘密鍵を含むファイルを準備します。次に、Zabbix Web サービス設定ファイルの `TLSCAFile`、`TLSCertFile`、`TLSKeyFile` パラメータを適切に編集します。

2. Zabbix Web サービス設定ファイルで、追加の TLS パラメータ `TLSAccept=cert` を編集します。

最終的な Web サービス設定ファイルの TLS パラメータは、次のようになります。

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Zabbix サーバー設定ファイルの `WebServiceURL` パラメータを編集して、TLS を設定した Zabbix Web サービスに接続するよう Zabbix サーバーを設定します。

```
WebServiceURL=https://example.com:443/report
```

許可する証明書の発行者とサブジェクトを制限する　2 つの Zabbix コンポーネント（サーバーとエージェントなど）が TLS 接続を確立すると、それぞれの証明書を検証します。ピア証明書が信頼された CA（`TLSCAFile` で事前設定された最上位証明書）によって署名され、有効期限内であり、その他のチェックにも合格している場合、コンポーネント間の通信を開始できます。この最も単純なケースでは、証明書の発行者とサブジェクトは検証されません。

ただし、これはリスクを伴います。有効な証明書を持つ誰もが、他の誰かになりすますことができるためです（例えば、ホスト証明書を使用してサーバーになりすますことができます）。証明書が専用の社内 CA によって署名され、なりすましのリスクが低い小規模な環境では許容できる場合がありますが、より大規模な環境やセキュリティ要件の高い環境では十分でない可能性があります。

最上位 CA が、Zabbix で受け入れるべきでない証明書を発行する場合、またはなりすましのリスクを低減したい場合は、発行者とサブジェクトを指定して、許可する証明書を制限できます。

例えば、Zabbix のプロキシ設定ファイルで、次のように指定できます。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

これらの設定により、アクティブプロキシは、証明書の発行者またはサブジェクトが異なる Zabbix サーバーとは通信しません。同様に、パッシブプロキシは、そのようなサーバーからのリクエストを受け入れません。

`Issuer` と `Subject` 文字列の照合ルール

`Issuer` と `Subject` 文字列の照合ルールは次のとおりです。

- `Issuer` と `Subject` の文字列は個別にチェックされます。どちらも省略可能です。
- 指定されていない文字列は、任意の文字列が受け入れられることを意味します。
- 文字列は そのまま比較され、完全一致する必要があります。
- UTF-8 文字がサポートされています。ただし、ワイルドカード (*) や正規表現はサポートされていません。
- 次の RFC 4514 の要件が実装されています。エスケープが必要な文字は、('\' バックスラッシュ、U+005C) を使用してエスケープします。
  - 文字列のどこでも: '"' (U+0022)、'+' (U+002B)、',' (U+002C)、';' (U+003B)、'<' (U+003C)、'>' (U+003E)、'\\' (U+005C);

    - 文字列の先頭: スペース (' ', U+0020) または番号記号 ('#', U+0023);
    - 文字列の末尾: スペース (' ', U+0020)。
- Null 文字 (U+0000) はサポートされていません。Null 文字が見つかった場合、照合は失敗します。
- RFC 4517 および RFC 4518 の標準はサポートされていません。

たとえば、Issuer と Subject の組織 (O) 文字列に末尾のスペースが含まれ、Subject の組織単位 (OU) 文字列に二重引用符が含まれている場合、これらの文字はエスケープする必要があります。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development head,O=\ Example SIA\ ,DC=example,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group \"5\",O=\ Example SIA\ ,DC=example,DC=com
```

フィールドの順序と書式

Zabbix は RFC 4514 の推奨に従っており、これらのフィールドは「逆順」で記述します。つまり、最下位レベルのフィールド (CN) から始め、中間レベルのフィールド (OU, O) を経て、最上位レベルのフィールド (DC) で終わります。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

一方、OpenSSL はデフォルトでは Issuer と Subject の文字列を上位レベルから下位レベルの順で表示します。次の例では、Issuer と Subject フィールドは最上位レベル (DC) から始まり、最下位レベル (CN) のフィールドで終わります。また、スペースやフィールド区切りの書式も使用するオプションによって異なるため、Zabbix が必要とする形式とは一致しません。

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Issuer と *Subject* の文字列を Zabbix 用に正しく整形するには、OpenSSL を次のオプション付きで実行します。

```
$ openssl x509 -noout -issuer -subject \
    -nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname\
    -in /home/zabbix/zabbix_proxy.crt
```

これにより、出力は逆順でカンマ区切りになり、Zabbix の設定ファイルおよび Web インターフェースで使用できる形式になります。

```
issuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

X.509 v3 証明書拡張の使用に関する制限　Zabbix で X.509 v3 証明書を実装する場合、特定の拡張は完全にはサポートされないか、動作が一貫しない可能性があります。

Subject Alternative Name 拡張

Zabbix は、IP アドレスやメールアドレスなどの代替 DNS 名を指定するために使用される *Subject Alternative Name* 拡張をサポートしていません。
Zabbix が検証できるのは、証明書の *Subject* フィールドの値のみです (許可される証明書の発行者と Subject の制限 を参照してください)。
証明書に subjectAltName フィールドが含まれている場合、証明書検証の結果は、Zabbix コンポーネントのビルドに使用された特定の暗号ツールキットによって異なる場合があります。
そのため、Zabbix はこれらの組み合わせに基づいて証明書を受け入れる場合もあれば、拒否する場合もあります。

Extended Key Usage 拡張

Zabbix は *Extended Key Usage* 拡張をサポートしています。
ただし、これを使用する場合は、通常、*clientAuth*（TLS WWW クライアント認証用）と *serverAuth*（TLS WWW サーバー認証用）の両方の属性を指定する必要があります。
例:

- パッシブチェックでは、Zabbix エージェントが TLS サーバーとして動作するため、エージェントの証明書に *serverAuth* 属性を含める必要があります。
- アクティブチェックでは、エージェントが TLS クライアントとして動作するため、エージェントの証明書に *clientAuth* 属性を含める必要があります。

GnuTLS はキー使用法違反に対して警告を出す場合がありますが、通常はこれらの警告があっても通信を継続できます。

Name Constraints 拡張

Name Constraints 拡張のサポートは、暗号ツールキットによって異なります。
選択したツールキットがこの拡張をサポートしていることを確認してください。
この拡張では、使用しているツールキットによっては、このセクションが critical としてマークされている場合に、Zabbix が CA 証明書を読み込めなくなることがあります。

証明書失効リスト **(CRL)** 証明書が侵害された場合、認証局 (CA) はその証明書を証明書失効リスト (CRL) に含めることで失効させることができます。
CRL は設定ファイルを通じて管理され、サーバー、プロキシ、およびエージェントの設定ファイルで TLSCRLFile パラメーターを使用して指定できます。
例:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

この場合、zabbix_crl_file.crt には複数の CA の CRL を含めることができ、次のようになります:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

CRL ファイルは Zabbix の起動時にのみ読み込まれます。
CRL を更新するには、Zabbix を再起動してください。

> **Attention:**
> Zabbix コンポーネントが OpenSSL でコンパイルされており、CRL を使用する場合は、証明書チェーン内の各最上位 CA および中間 CA に対応する CRL が TLSCRLFile に含まれていることを確認してください（空の CRL でも構いません）。

証明書の問題のトラブルシューティング CRL とともに使用される OpenSSL で、証明書チェーン内の一部の CA の CRL が TLSCRLFile に含まれていない場合

TLS サーバーログで *OpenSSL* ピアの場合:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

TLS サーバーログで *GnuTLS* ピアの場合:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL の有効期限切れ、またはサーバー稼働中に期限切れになる場合

OpenSSL、サーバーログ内:

- 期限切れ前:

```
プロキシ "proxy-openssl-1.0.1e" に接続できません: TCP は成功しましたが、[[127.0.0.1]:20004] への TLS を確
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- 期限切れ後:

```
プロキシ "proxy-openssl-1.0.1e" に接続できません: TCP は成功しましたが、[[127.0.0.1]:20004] への TLS を確
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

ここでのポイントは、有効な CRL がある場合、失効した証明書は”certificate revoked” として報告されることです。CRL の有効期限が切れると、エラーメッセージは”certificate expired” に変わりますが、これはかなり紛らわしいです。

GnuTLS、サーバーログ内:

- 期限切れ前後で同じ:

```
プロキシ "proxy-openssl-1.0.1e" に接続できません: TCP は成功しましたが、[[127.0.0.1]:20004] への TLS を確
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

自己署名証明書、未知の CA

OpenSSL のログ:

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
      line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
      TLS write fatal alert "unknown CA"'
```

これは、サーバー証明書が CA によって署名されているにもかかわらず、誤って Issuer と Subject の文字列が同じになっていた場合に確認されました。Issuer と Subject は最上位の CA 証明書では同じでもかまいませんが、サーバー証明書では同じにできません。(同様に、プロキシおよびエージェントの証明書にも当てはまります。)

証明書に同じ Issuer と Subject のエントリが含まれているかどうかを確認するには、次を実行します:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

ルート（最上位）証明書では、Issuer と Subject の値が同一であっても問題ありません。

事前共有鍵を使用した暗号化

概要　Zabbix の各事前共有鍵（PSK）は、実際には次の 2 つの組み合わせです。

- 秘密ではない PSK ID 文字列
- 秘密の PSK 文字列値

PSK ID 文字列は、空ではない UTF-8 文字列です。たとえば、「PSK ID 001 Zabbix agentd」のようになります。これは、この特定の PSK を Zabbix コンポーネントが参照するための一意の名前です。PSK ID 文字列には機密情報を含めないでください。PSK ID 文字列は暗号化されずにネットワーク経由で送信されます。

PSK 値は､推測が困難な 16 進数の文字列です｡たとえば､e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9 のようになります。

> **Warning:**
> ドキュメントに記載されている PSK の例を、本番環境では絶対に使用しないでください。代わりに、暗号学的に安全な一意の PSK を生成してください。

サイズ制限　Zabbix では PSK identity と PSK value にサイズ制限があります。一部のケースでは、暗号ライブラリの制限がさらに小さい場合があります。

Component	PSK identity max size	PSK value min size	PSK value max size
Zabbix	128 UTF-8 characters	128-bit (16-byte PSK, entered as 32 hexadecimal digits)	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)
GnuTLS	128 bytes (may include UTF-8 characters)	-	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)
OpenSSL 1.0.x, 1.1.0	127 bytes (may include UTF-8 characters)	-	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)
OpenSSL 1.1.1	127 bytes (may include UTF-8 characters)	-	512-bit (64-byte PSK, entered as 128 hexadecimal digits)
OpenSSL 1.1.1a and later	127 bytes (may include UTF-8 characters)	-	2048-bit (256-byte PSK, entered as 512 hexadecimal digits)

Attention:
Zabbix の Web インターフェースでは、使用する暗号ライブラリに関係なく、最大 128 文字の PSK identity 文字列と最大 2048 ビットの PSK を設定できます。
 一部の Zabbix コンポーネントがより小さい制限をサポートしている場合、それらのコンポーネントで許可される長さに収まるように PSK identity と PSK value を設定することは、ユーザーの責任です。
 長さの制限を超えると、Zabbix コンポーネント間の通信に失敗します。

Zabbix サーバーが PSK を使用してエージェントに接続する前に、サーバーはデータベース（実際には設定キャッシュ）で、そのエージェントに設定された PSK identity と PSK value を検索します。接続を受信すると、エージェントは自身の設定ファイルに設定された PSK identity と PSK value を使用します。双方で同じ PSK identity 文字列と PSK value が使用されている場合、接続が成功する可能性があります。

Attention:
各 PSK identity には、1 つの PSK value のみを対応付ける必要があります。同じ identity 文字列で異なる値を持つ PSK が 2 つ存在しないようにすることは、ユーザーの責任です。これを怠ると、この PSK identity 文字列を使用する PSK による Zabbix コンポーネント間の通信で、予期しないエラーや中断が発生する可能性があります。

PSK の生成　たとえば、以下のコマンドを使用して 256 ビット（32 バイト）の PSK を生成できます。

- *OpenSSL* を使用する場合:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- *GnuTLS* を使用する場合:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

上記の「psktool」は、PSK identity とそれに関連付けられた PSK を含むデータベースファイルを生成します。Zabbix が PSK ファイルに必要とするのは PSK のみであるため、ファイルから identity 文字列とコロン（「:」）を削除する必要があります。

サーバーとエージェント間の通信で **PSK** を設定する（例）　エージェントのホストで、上記で生成した PSK の値を、たとえば /home/zabbix/zabbix_agentd.psk ファイルに書き込みます。ファイルの 1 行目には PSK の値を記述する必要があります。

PSK ファイルのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにしてください。

Zabbix のエージェント設定ファイル（zabbix_agentd.conf）で TLS パラメータを編集します。たとえば、次のように設定します。

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

エージェントは、PSK を使用する接続のみでサーバーに接続し（アクティブチェック）、サーバーおよび zabbix_get からの接続を受け入れます。PSK ID は「PSK 001」になります。

エージェントを再起動します。これで、たとえば次のように zabbix_get を使用して接続をテストできます。

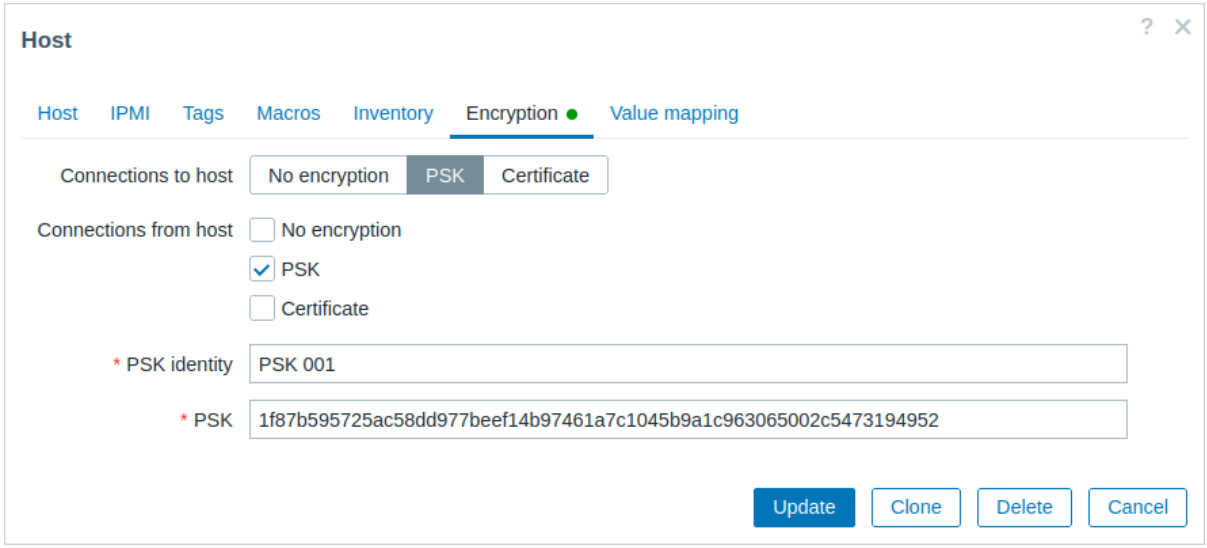
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(ダウンタイムを最小限に抑えるには、接続暗号化の管理で接続タイプを変更する方法を確認してください)。

Zabbix の Web インターフェースで、このエージェントの PSK 暗号化を設定します。

1. データ収集 > ホストに移動します。
2. ホストを選択し、暗号化タブをクリックします。
3. PSK 暗号化の詳細を設定します。

例：



すべての必須入力フィールドには赤いアスタリスクが付いています。

設定キャッシュがデータベースと同期されると、新しい接続で PSK が使用されます。エラーメッセージがないか、サーバーとエージェントのログファイルを確認してください。

サーバー - アクティブプロキシ間の通信で **PSK** を設定する(例) プロキシ上で、PSKの値をファイル(例：/home/zabbix/zabbix_proxy.psk)に書き込みます。ファイルの最初のテキスト文字列に PSK を記述する必要があります。例：

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

PSK ファイルのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

Zabbix のプロキシ設定ファイル（zabbix_proxy.conf）で TLS パラメータを編集します。例：

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

プロキシは PSK を使用してサーバーに接続します。PSK ID は「PSK 002」になります。

(ダウンタイムを最小限に抑えるには、接続暗号化の管理で接続タイプを変更する方法を参照してください)。

Zabbix の Web インターフェースで、このプロキシの PSK を設定します。管理 > プロキシに移動し、プロキシを選択して、暗号化タブに移動します。「プロキシからの接続」で PSK にチェックを入れます。「PSK ID」フィールドに「PSK 002」を貼り付け、「PSK」フィールドに「e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9」を貼り付けます。「更新」をクリックします。

プロキシを再起動します。プロキシは PSK ベースの暗号化接続を使用してサーバーに接続するようになります。サーバーとプロキシのログファイルでエラーメッセージを確認してください。

パッシブプロキシの場合も、手順はほぼ同じです。唯一の違いは、Zabbix のプロキシ設定ファイルで TLSAccept=psk を設定し、Zabbix の Web インターフェースでプロキシへの接続を PSK に設定することです。

PSK の障害のトラブルシューティング PSK に奇数個の 16 進数桁が含まれています

プロキシまたはエージェントが起動せず、プロキシまたはエージェントのログに次のメッセージが表示されます:

```
file "/home/zabbix/zabbix_proxy.psk" に無効な PSK があります
```

128 バイトを超える PSK identity 文字列が GnuTLS に渡される

TLS クライアント側のログ:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

TLS サーバー側のログ:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

OpenSSL 1.1.1 で使用される PSK 値が長すぎる

接続側のログ:

```
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK 1"
...zbx_psk_client_cb() requested PSK identity "PSK 1"
...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\ex
```

受信側のログ:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

この問題は通常、OpenSSL を 1.0.x または 1.1.0 から 1.1.1 にアップグレードした場合に発生し、PSK 値が 512 ビット（64 バイトの PSK、128 桁の 16 進数）より長いときに起こります。

こちらも参照: 値サイズの制限

データベースへの安全な接続

概要

このセクションでは、以下の間で安全な TLS 接続を確立するための Zabbix のセットアップ手順と設定例を示します。

Database	Zabbix components
MySQL	Zabbix Web インターフェース, Zabbix サーバー, Zabbix プロキシ
PostgreSQL	Zabbix Web インターフェース, Zabbix サーバー, Zabbix プロキシ

DBMS 内で接続の暗号化を設定する方法については、詳細を各ベンダーの公式ドキュメントで確認してください。

- MySQL: ソースおよびレプリカのレプリケーションデータベースサーバー。
- MySQL: グループレプリケーションなどのデータベースサーバー。
- PostgreSQL 暗号化オプション。

すべての例は、CentOS 8 を使用し、公式リポジトリから入手できる MySQL CE (8.0) および PgSQL (13) の GA リリースを基にしています。

要件

暗号化を設定するには、以下が必要です。

- 開発元がサポートするオペレーティングシステムで、OpenSSL >=1.1.X または代替手段が利用できること。

Note:
特に新規インストールの場合は、OS がサポート終了状態であることは避けることを推奨します。

- 開発元が提供する公式リポジトリからインストールされ、保守されているデータベースエンジン（RDBMS）。オペレーティングシステムには、暗号化サポートが実装されていない古いデータベースソフトウェアのバージョンが同梱されていることがよくあります。たとえば、RHEL 7 ベースのシステムや、暗号化サポートのない PostgreSQL 9.2、MariaDB 5.5 などです。

用語

このオプションを設定すると、Zabbix サーバー/プロキシおよび Web インターフェースからデータベースへの接続に TLS 接続を使用することが強制されます。

- `required` - アイデンティティ確認なしで、TLS を転送モードとして使用して接続します
- `verify_ca` - TLS を使用して接続し、証明書を検証します
- `verify_full` - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベースのアイデンティティ (CN) がその証明書と一致することを検証します

Zabbix 設定

Web インターフェースからデータベースへの接続

データベースへの安全な接続は、Web インターフェースのインストール時に設定できます。

- DB 接続の設定の手順で、*Database TLS encryption* チェックボックスを選択して、転送時の暗号化を有効にします。
- *TLS encryption* フィールドを選択すると表示される *Verify database certificate* チェックボックスを選択して、証明書を使用した暗号化を有効にします。

Note:
MySQL の場合、*Database host* に localhost が設定されていると、*Database TLS encryption* チェックボックスは無効になります。これは、ソケットファイル（Unix の場合）または共有メモリ（Windows の場合）を使用する接続を暗号化できないためです。
 PostgreSQL の場合、*Database host* フィールドの値がスラッシュで始まるか、フィールドが空の場合、*TLS encryption* チェックボックスは無効になります。

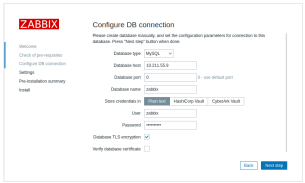
TLS 暗号化で証明書を使用するモード（両方のチェックボックスを選択した場合）では、次のパラメータを使用できます。

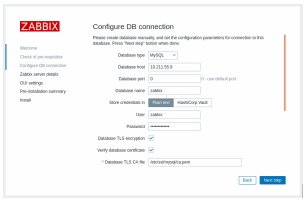
Parameter	Description
Database TLS CA file	有効な TLS 認証局（CA）ファイルへの絶対パスを指定します。
Database TLS key file	有効な TLS キーファイルへの絶対パスを指定します。
Database TLS certificate file	有効な TLS 証明書ファイルへの絶対パスを指定します。
Database host verification	ホスト検証を有効にするには、このチェックボックスを選択します。 PHP MySQL ライブラリでは、ピア証明書の検証手順をスキップできないため、MySQL では無効です。
Database TLS cipher list	有効な暗号のカスタムリストを指定します。暗号リストの形式は、OpenSSL 標準に準拠している必要があります。 MySQL でのみ使用できます。

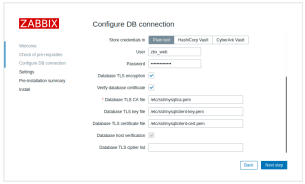
Attention:
TLS パラメータには、有効なファイルを指定する必要があります。存在しないファイルや無効なファイルを指定すると、認証エラーが発生します。
 証明書ファイルが書き込み可能な場合、Web インターフェースはSystem informationレポートに「TLS certificate files must be read-only.」という警告を生成します（PHP ユーザーが証明書の所有者である場合にのみ表示されます）。
パスワードで保護された証明書はサポートされていません。

使用例

Zabbix の Web インターフェースでは、GUI を使用して次のオプションを定義します: `required`、`verify_ca`、`verify_full`。インストールウィザードの *DB* 接続の設定ステップで、必要なオプションを指定します。
これらのオプションは、Web インターフェース設定ファイル（`zabbix.conf.php`）に次のようにマッピングされます。

GUI 設定	設定ファイル	説明	結果
	... // TLS 接続に使用。 $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	1. データベース *TLS* 暗号化をチェック 2. データベース証明書の検証をチェックしない	`required` モードを有効にします。

GUI 設定	設定ファイル	説明	結果
	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem'; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	1. データベース *TLS* 暗号化とデータベース証明書の検証をチェック 2. データベース *TLS CA* ファイルへのパスを指定	`verify_ca` モードを有効にします。

	...
	// 厳密に定義された Cipher リストを使用する TLS 接続に使用。
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '<cipher_list>';
	...
	
	または:
	
	...
	// Cipher リストを定義せず、MySQL サーバーによって選択される TLS 接続に使用
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '';
	...
		1. データベース *TLS* 暗号化とデータベース証明書の検証をチェック
		2. データベース *TLS* キーファイルへのパスを指定
		3. データベース *TLS CA* ファイルへのパスを指定
		4. データベース *TLS* 証明書ファイルへのパスを指定
		5. データベース *TLS Cipher* リストを指定（任意）
			MySQL の `verify_full` モードを有効にします。

GUI 設定	設定ファイル	説明	結果
	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = '<key_file_path>'; $DB['CERT_FILE'] = '<key_file_path>'; $DB['CA_FILE'] = '<key_file_path>'; $DB['VERIFY_HOST'] = true; $DB['CIPHER_LIST'] = ' '; ...	1. データベース *TLS* 暗号化とデータベース証明書の検証をチェック 2. データベース *TLS* キーファイルへのパスを指定 3. データベース *TLS CA* ファイルへのパスを指定 4. データベース *TLS* 証明書ファイルへのパスを指定 5. データベースホストの検証をチェック	PostgreSQL の `verify_full` モードを有効にします。

関連項目: MySQL の暗号化設定例、PostgreSQL の暗号化設定例。

Zabbix サーバー/プロキシの設定

データベースへの安全な接続は、Zabbix サーバーおよび/またはプロキシの設定ファイルで、それぞれのパラメータを使用して設定できます。

設定	結果
なし	暗号化なしでデータベースに接続します。
`DBTLSConnect=required` を設定	サーバー/プロキシはデータベースへの TLS 接続を確立します。暗号化されていない接続は許可されません。
1. `DBTLSConnect=verify_ca` を設定 2. `DBTLSCAFile` を設定 - TLS 認証局ファイルを指定	データベース証明書を検証した後、サーバー/プロキシはデータベースへの TLS 接続を確立します。
1. `DBTLSConnect=verify_full` を設定 2. `DBTLSCAFile` を設定 - TLS 認証局ファイルを指定	データベース証明書とデータベースのホスト ID を検証した後、サーバー/プロキシはデータベースへの TLS 接続を確立します。
1. `DBTLSCAFile` を設定 - TLS 認証局ファイルを指定 2. `DBTLSCertFile` を設定 - クライアント公開鍵証明書ファイルを指定 3. `DBTLSKeyFile` を設定 - クライアント秘密鍵ファイルを指定	サーバー/プロキシはデータベースへの接続時にクライアント証明書を提示します。
`DBTLSCipher` を設定 - TLS 1.2 までの TLS プロトコルを使用する接続でクライアントが許可する暗号化暗号のリスト	(MySQL) TLS 接続は、指定されたリスト内の暗号化暗号を使用して確立されます。 (PostgreSQL) このオプションを設定すると、エラーとして扱われます。
または `DBTLSCipher13` - TLS 1.3 プロトコルを使用する接続でクライアントが許可する暗号化暗号のリスト	

MySQL 暗号化設定

概要

このセクションでは、CentOS 8.2 と MySQL 8.4.0 向けの暗号化設定例をいくつか紹介します。データベースへの接続を暗号化するためのクイックスタートガイドとして利用できます。

Attention:
MySQL ホストが localhost に設定されている場合、暗号化オプションは利用できません。この場合、Zabbix Web インターフェースとデータベース間の接続は、Unix ではソケットファイル、Windows では共有メモリを使用するため、暗号化できません。

> **Note:**
> 暗号化の組み合わせは、このページに記載されているものに限定されません。利用可能な組み合わせは、これ以外にも多数あります。

前提条件

公式リポジトリから MySQL データベースをインストールします。

MySQL リポジトリの使用方法の詳細については、MySQL documentationを参照してください。

MySQL サーバーは、自己署名証明書を使用した安全な接続を受け入れる準備ができています。

暗号化された接続を使用しているユーザーを確認するには、次のクエリを実行します（Performance Schema は ON にしておく必要があります）。

```
SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, processli
FROM performance_schema.status_by_thread  AS sbt
JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
ORDER BY tls_version;
```

トランスポート層のみの暗号化

MySQL の設定

データベースの最新バージョンは、すぐに使用できる required 暗号化モードで構成されています。
初期セットアップと起動の後、サーバー側の証明書が作成されます。

主要コンポーネント用のユーザーとロールを作成します（<strong_password> は強力で一意のパスワードに置き換えてください）。

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
 'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
 REQUIRE SSL
 PASSWORD HISTORY 5;

mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';

mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_sr
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';

mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';

mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

X.509 プロトコルは ID の確認には使用されず、ユーザーは暗号化された接続のみを使用するように構成されていることに注意してください。
ユーザーの設定についての詳細は、MySQL のドキュメントを参照してください。

接続を確認するには、次のコマンドを実行します（ソケット接続は安全な接続のテストには使用できません）。

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

現在のステータスと使用可能な暗号スイートを確認します。

```
mysql> status
--------------
mysql Ver 8.4.0 for Linux on x86_64 (MySQL Community Server - GPL)

Connection id: 62
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384


mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
```

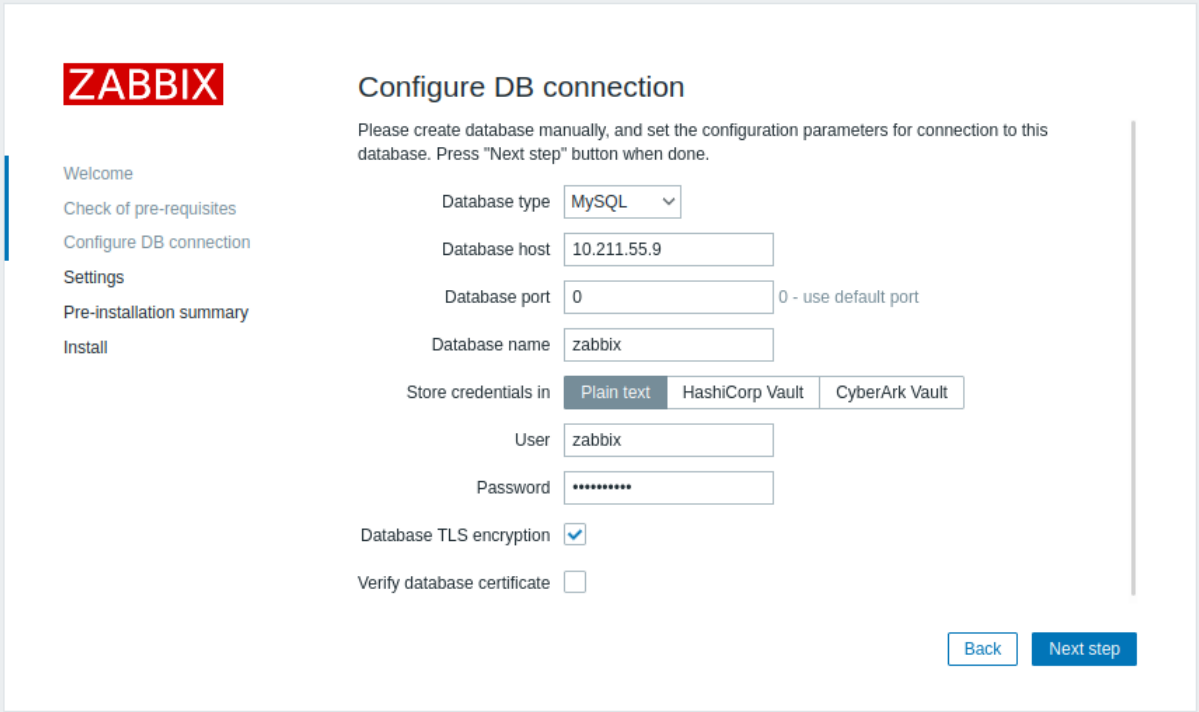
```
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で転送時のみの暗号化を有効にするには、次の操作を行います。

- *Database TLS encryption* を確認する
- *Verify database certificate* のチェックを外す



サーバー

サーバーとデータベース間の接続でトランスポートのみの暗号化を有効にするには、サーバー設定ファイルを編集します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

証明機関の検証を伴う暗号化

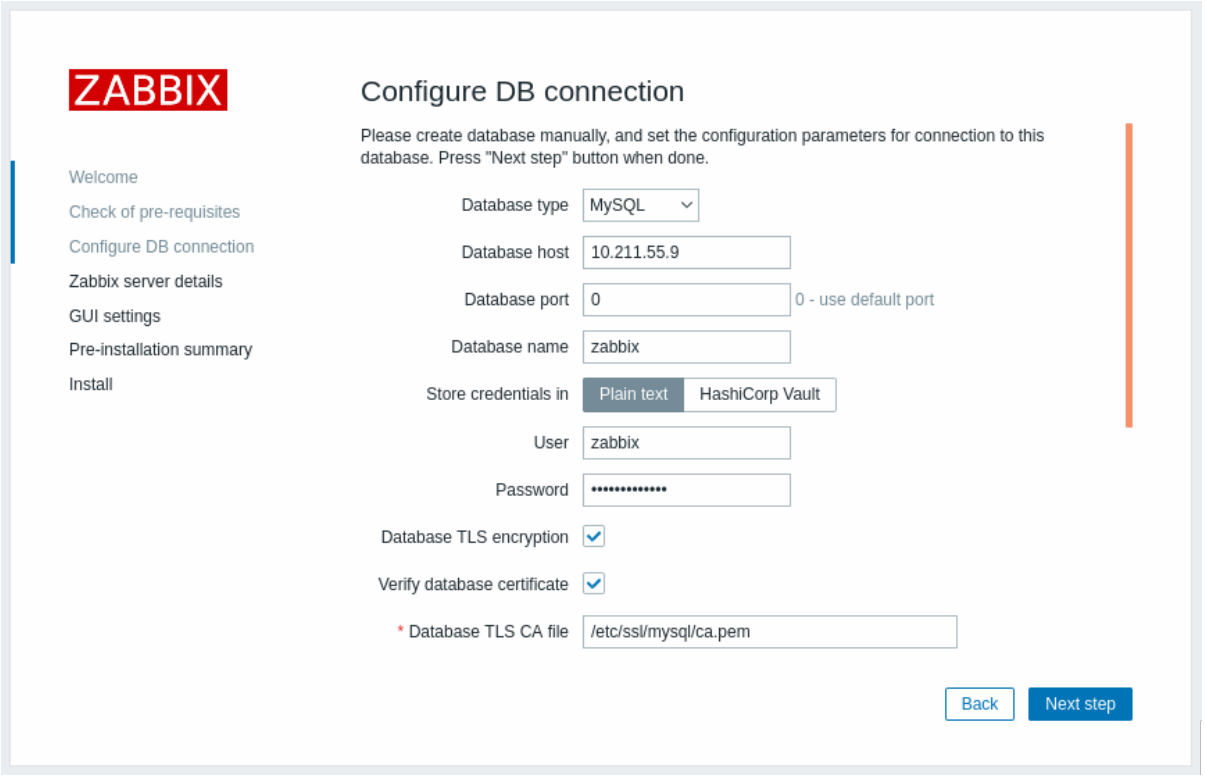
必要な MySQL CA を Zabbix Web インターフェースサーバーにコピーし、Web サーバーがこのファイルを読み取れるように適切な権限を設定します。

Note:
このモードは、古い MySQL ライブラリのため RHEL 7 では動作しません。

Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で、証明書検証を使用した暗号化を有効にするには、次の手順を実行します。

- *Database TLS encryption* と *Verify database certificate* にチェックを入れます
- Database TLS CA ファイルのパスを指定します



または、Web インターフェース設定ファイル（zabbix.conf.php）で設定できます。

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

コマンドラインツールを使用して、必要なユーザーで接続できるか確認し、トラブルシューティングを行います。

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

サーバー

Zabbix サーバーとデータベース間の接続で、証明書検証による暗号化を有効にするには、サーバー設定ファイルを編集します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

完全検証付きの暗号化

MySQL 設定

MySQL CE サーバーの設定オプション（*/etc/my.cnf.d/server-tls.cnf*）を次のように設定します。

```
[mysqld]
...
#### この例では、キーは MySQL CE の datadir ディレクトリに配置されています
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
...
```

MySQL CE サーバーおよびクライアント（Zabbix Web インターフェース）のキーは、MySQL CE のドキュメントに従って手動で作成する必要があります: Creating SSL and RSA certificates and keys using MySQL または Creating SSL certificates and keys using openssl

Attention:
MySQL サーバー証明書には、Common Name フィールドとして FQDN 名を設定してください。Zabbix Web インターフェースは、データベースとの通信に DNS 名、またはデータベースホストの IP アドレスを使用するためです。

MySQL ユーザーを作成します:

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH caching_sha2_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

そのユーザーでログインできるか確認します:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

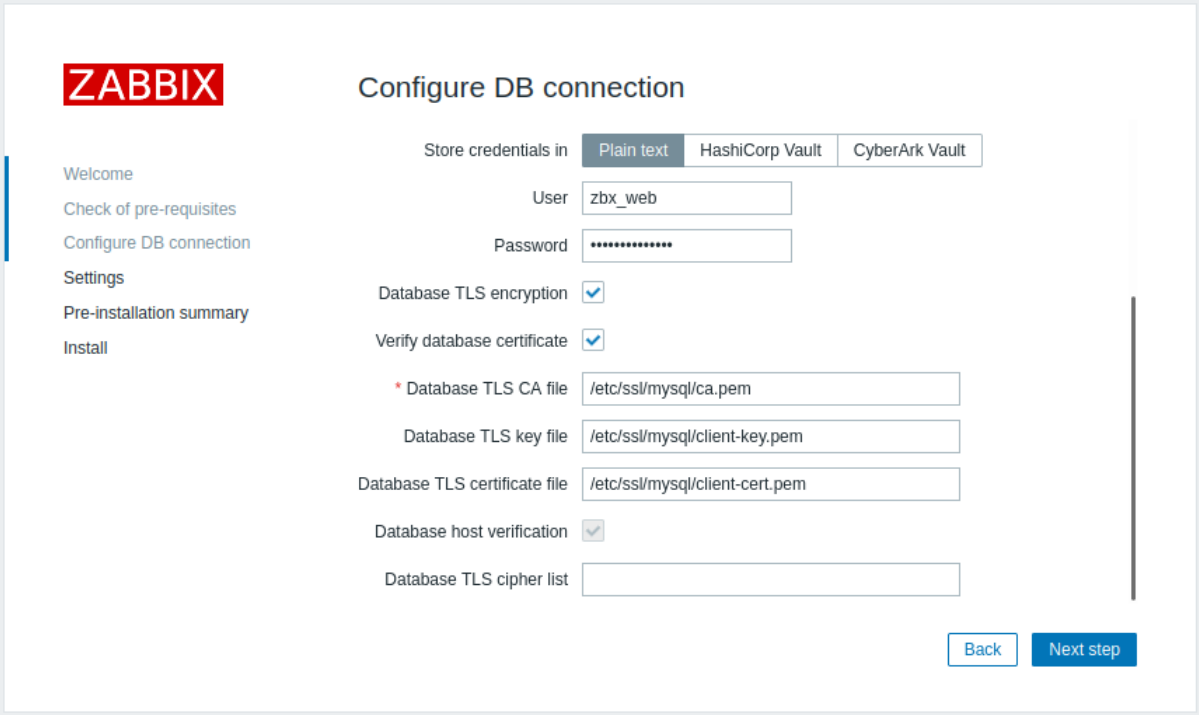
Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で、完全な検証を伴う暗号化を有効にするには、次の手順を実行します。

- データベース *TLS* 暗号化にチェックを入れ、データベース証明書の検証を有効にする
- データベース *TLS* キーファイルのパスを指定する
- データベース *TLS CA* ファイルのパスを指定する
- データベース *TLS* 証明書ファイルのパスを指定する

データベースホストの検証にはチェックが入っており、グレー表示されていることに注意してください。この手順は MySQL では省略できません。

Warning:
データベース *TLS* 暗号リストフィールドを空のままにすると、Web インターフェース（クライアント）とサーバーの両方で許可されている共通の暗号が有効になります。または、暗号設定の要件に従って、暗号を明示的に設定できます。



また、Web インターフェース設定ファイル（zabbix.conf.php）で設定することもできます。

```
...
// Used for TLS connection with strictly defined Cipher list.
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

```
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
...
// or
...
// Used for TLS connection without Cipher list defined - selected by MySQL server
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続で完全な検証を伴う暗号化を有効にするには、サーバー設定ファイルを編集します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

PostgreSQL 暗号化設定

概要

このセクションでは、CentOS 8.2 と PostgreSQL 13 に対するいくつかの暗号化設定例を示します。

> **Note:**
> Zabbix Web インターフェースと PostgreSQL 間の接続は、*Database host* フィールドの値がスラッシュで始まる場合、またはそのフィールドが空の場合、暗号化できません（GUI のパラメータは無効になります）。

前提条件

公式リポジトリを使用して PostgreSQL データベースをインストールします。

PostgreSQL は、デフォルトでは TLS 接続を受け入れるように設定されていません。`postgresql.conf` による証明書の準備と、`ph_hba.conf` を通じたユーザーアクセス制御について、PostgreSQL のドキュメントの手順に従ってください。

デフォルトでは、PostgreSQL のソケットは localhost にバインドされています。ネットワーク経由のリモート接続を許可するには、実際のネットワークインターフェースで待ち受けるようにしてください。

すべてのモード に対する PostgreSQL の設定は、次のようになります。

`/var/lib/pgsql/13/data/postgresql.conf`:

```
...
ssl = on
ssl_ca_file = 'root.crt'
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'
ssl_prefer_server_ciphers = on
ssl_min_protocol_version = 'TLSv1.3'
...
```

アクセス制御については、`/var/lib/pgsql/13/data/pg_hba.conf` を調整します。

```
...
### require
```

```
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

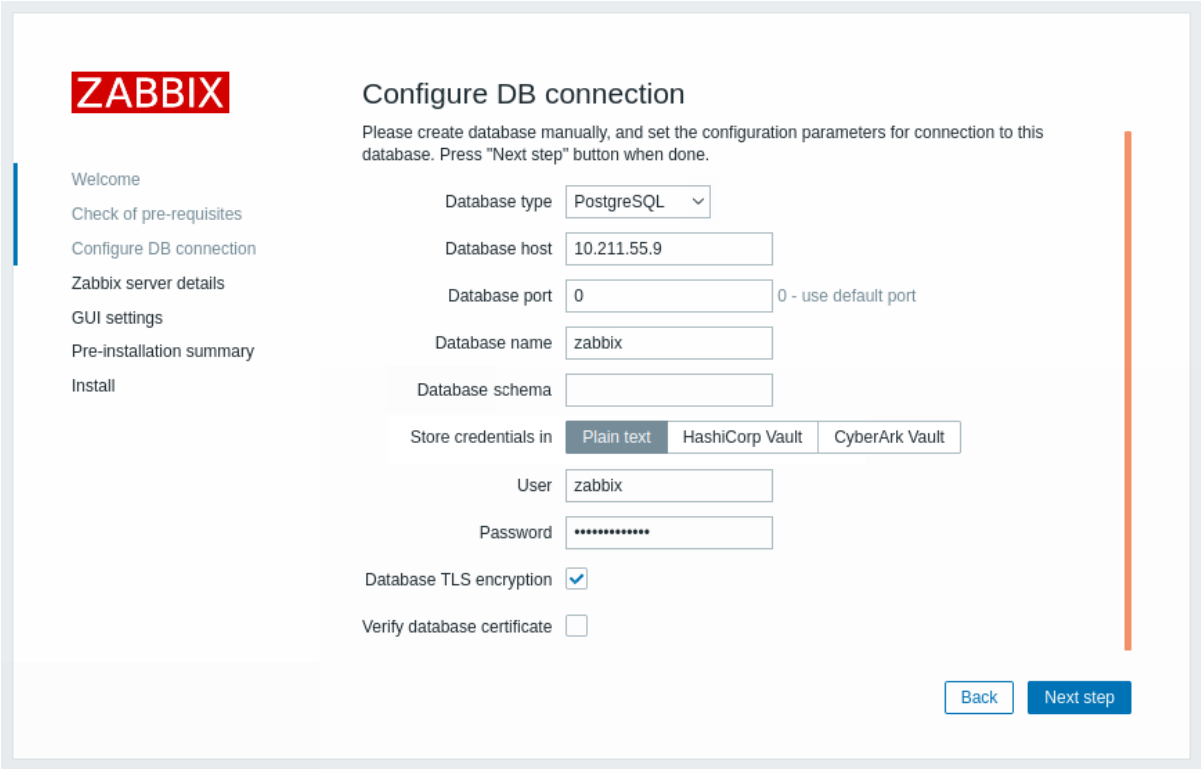
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

トランスポート層のみの暗号化

Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で、転送時のみの暗号化を有効にするには:

- *Database TLS encryption* を確認する
- *Verify database certificate* はオフのままにする



サーバー

サーバーとデータベース間の接続でトランスポートのみの暗号化を有効にするには、サーバー設定ファイルを編集します。

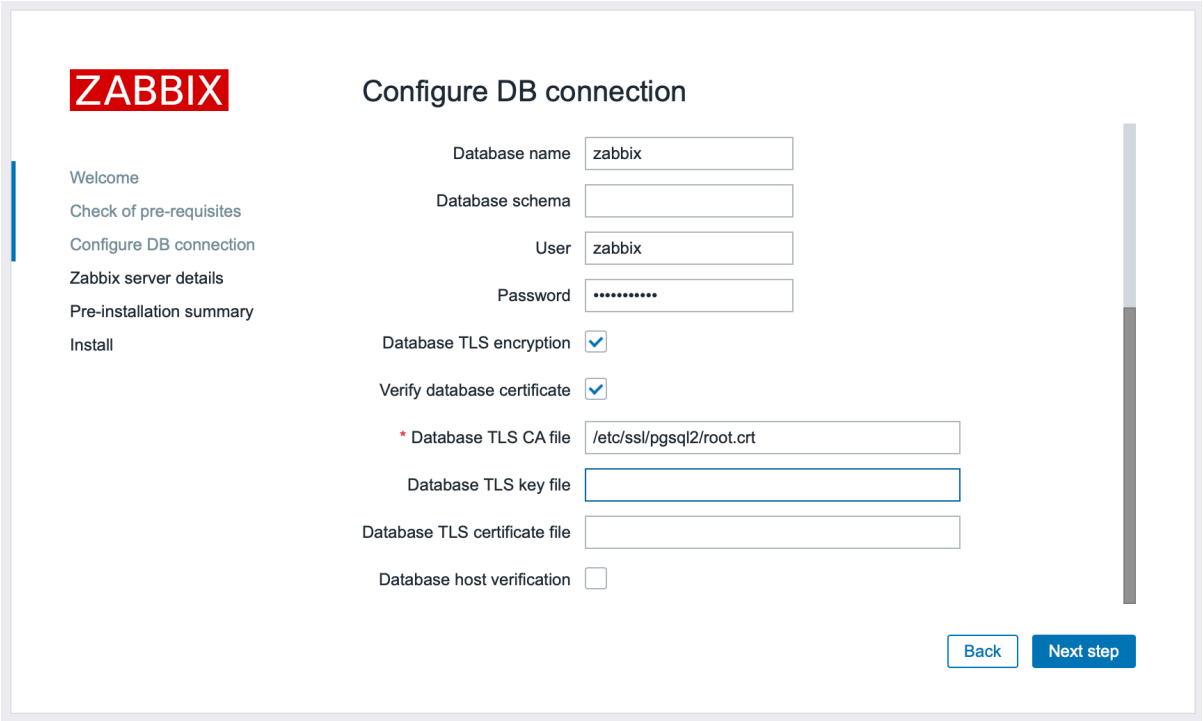
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

証明機関の検証による暗号化

Web インターフェース

認証局の検証を使用して、Zabbix Web インターフェースとデータベース間の接続を暗号化するには、次の手順を実行します。

- データベースの *TLS* 暗号化と データベース証明書の検証をオンにします
- データベース *TLS CA* ファイルのパスを指定します



または、Web インターフェース設定ファイル（zabbix.conf.php）で設定できます。

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続で、証明書検証を使用した暗号化を有効にするには、サーバー設定ファイルを編集します（<strong_password> を強力で一意のパスワードに置き換えてください）。

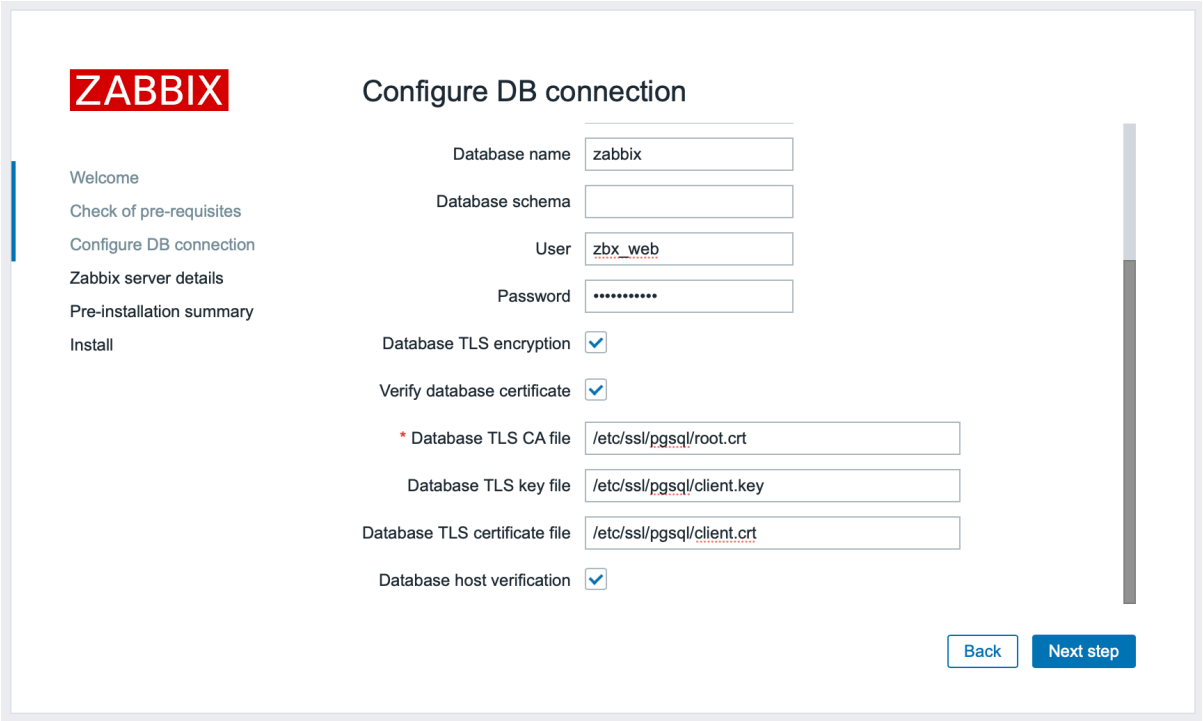
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

完全検証付きの暗号化

Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で、証明書による暗号化およびデータベースホストの識別情報の検証を有効にするには、次の手順を実行します。

- *Database TLS encryption* にチェックを入れ、*Verify database certificate* にチェックを入れる
- *Database TLS key file* のパスを指定する
- *Database TLS CA file* のパスを指定する
- *Database TLS certificate file* のパスを指定する
- *Database host verification* にチェックを入れる



または、Web インターフェース設定ファイル（zabbix.conf.php）で設定できます。

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続で、証明書およびデータベースホストの識別情報の検証による暗号化を有効にするには、サーバー設定ファイルを編集します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

Web インターフェースとサーバー間の安全な接続

概要

このセクションでは、Zabbix の Web インターフェースと Zabbix サーバー間の安全な TLS 接続を確立するためのセットアップ手順と設定例を説明します。

設定

デフォルトでは、Zabbix Web インターフェースと Zabbix サーバー間の通信は暗号化されません。セキュリティを強化するには、両方で TLS を有効にします。以下に、これを行う最も簡単な方法の例を示します。

1. 証明書とキーを生成します。

作業ディレクトリを作成します。

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

CA 証明書を作成します（MyZabbixCA の値は実際のコモンネームに合わせて変更してください）。

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Zabbix サーバー用の秘密鍵と証明書を生成します（zabbix-server.example.com の値は実際のコモンネームに合わせて変更してください）。

```
sudo openssl genrsa -out server.key 2048
sudo openssl req -new -key server.key -out server.csr -subj "/CN=zabbix-server.example.com/"
sudo openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out serv
```

Zabbix Web インターフェース用の秘密鍵と証明書を生成します（zabbix-frontend-node の値は実際のコモンネームに合わせて変更してください）。

```
sudo openssl genrsa -out frontend.key 2048
sudo openssl req -new -key frontend.key -out frontend.csr -subj "/CN=zabbix-frontend-node/"
sudo openssl x509 -req -days 365 -in frontend.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out fr
```

2. 適切な権限を設定します。

Zabbix サーバーの場合（所有者とグループは、ディストリビューションの Zabbix サーバーデーモンユーザーに合わせて変更してください）。

```
sudo chown root:zabbix /etc/zabbix/ssl/server.{crt,key} /etc/zabbix/ssl/ca.crt
sudo chmod 640 /etc/zabbix/ssl/server.key
sudo chmod 644 /etc/zabbix/ssl/server.crt /etc/zabbix/ssl/ca.crt
```

Web インターフェースの場合（所有者とグループは、ディストリビューションの Web サーバーユーザーに合わせて変更してください）。

```
sudo chown root:www-data /etc/zabbix/ssl/frontend.{crt,key}
sudo chmod 640 /etc/zabbix/ssl/frontend.key
sudo chmod 644 /etc/zabbix/ssl/frontend.crt
```

3. Zabbix サーバーを設定します。

サーバー設定ファイル（zabbix_server.conf）に次の設定を追加します。

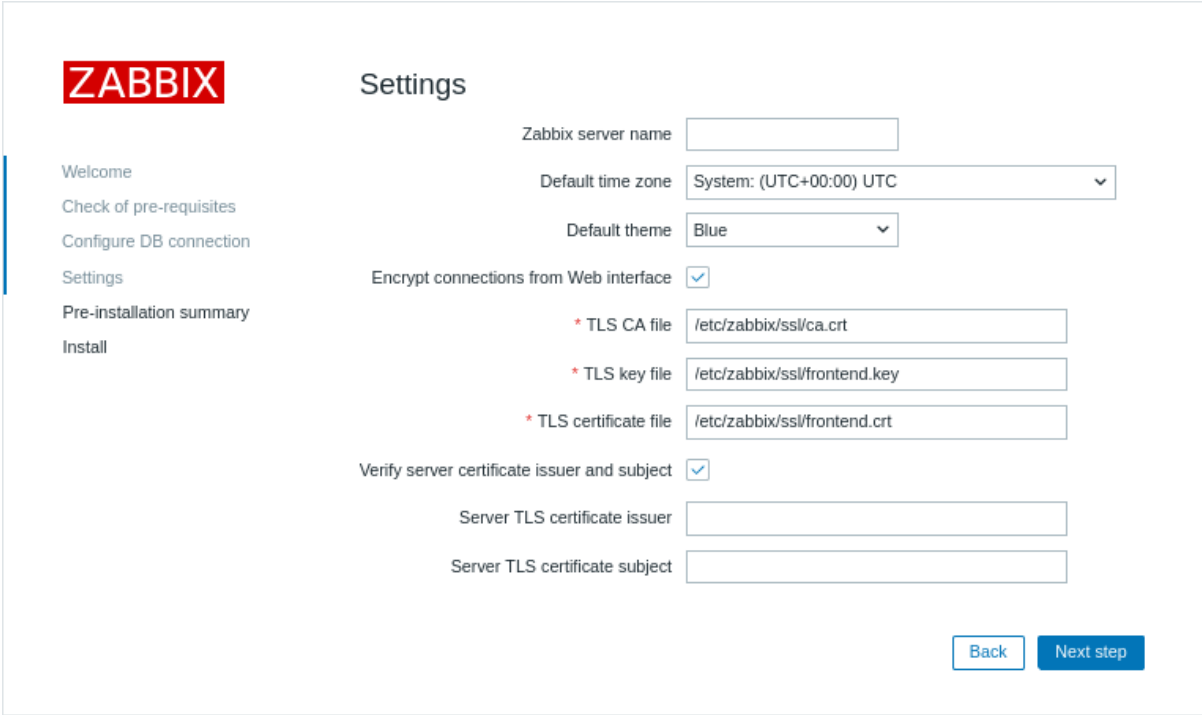
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
### オプション:
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

次に、サーバーを再起動します。

```
sudo systemctl restart zabbix-server
```

4. Zabbix Web インターフェースを設定します。

Web インターフェースのインストール中に、*Web* インターフェースからの接続を暗号化オプションを有効にします（必要に応じて、サーバー証明書の発行者とサブジェクトを検証オプションも有効にします）。次に、*TLS CA* ファイル、*TLS* キーファイル、*TLS* 証明書ファイルの各フィールド（必要に応じて、サーバー *TLS* 証明書の発行者および サーバー *TLS* 証明書のサブジェクトフィールドも）に値を入力します。



パラメーター	説明
TLS CA ファイル	サーバーの証明書の検証に使用する認証局（CA）証明書ファイルの完全なパスを指定します。
TLS キーファイル	クライアント証明書に対応するクライアント秘密鍵ファイルの完全なパスを指定します。
TLS 証明書ファイル	相互 TLS 認証が必要な場合は、クライアント証明書ファイルの完全なパスを指定します。
サーバー *TLS* 証明書の発行者	サーバーの証明書と照合する発行者の識別名（DN）を指定します。
サーバー *TLS* 証明書のサブジェクト	サーバーの証明書と照合するサブジェクトの識別名（DN）を指定します。

既存のインストールでは、Web インターフェース設定ファイルの次のパラメーターを編集します。

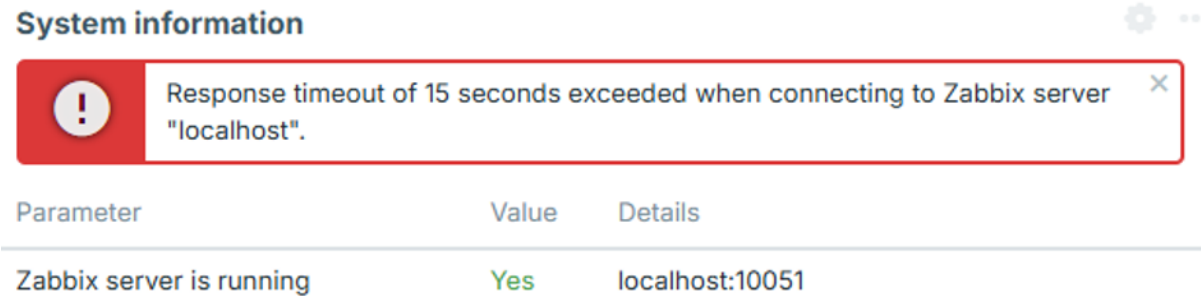
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// オプション:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

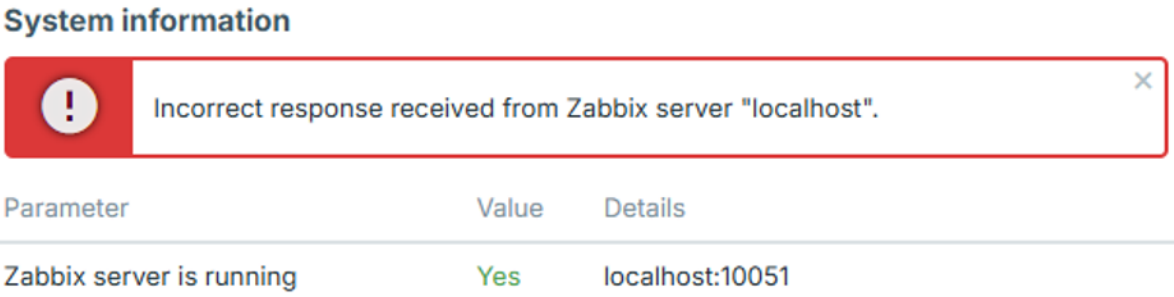
5. Zabbix Web インターフェースまたは Zabbix サーバーのログファイルにエラーメッセージがないことを確認して、暗号化を検証します。

```
tail -f /var/log/zabbix/zabbix_server.log
```

トラブルシューティング

TLS 設定が正しくない、または不完全な場合、Zabbix の Web インターフェースでエラーが発生したり、Zabbix サーバーが実行中であるにもかかわらず接続が正常に確立されていないことを示す場合があります。





正確な設定上の問題を特定するには、Zabbix サーバーのログレベルを 4 に設定し、Web インターフェースでエラーを再現してから、対応するエラーメッセージがないか Zabbix サーバーのログを確認します。

以下では、一般的なエラーと考えられる原因について説明します。

Web インターフェースの IP アドレスが許可されていない

Zabbix サーバーのログに、次のようなエラーが記録されます。

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

Web インターフェースの IP アドレスが承認されていないため、TCP ハンドシェイク後に接続が切断されます。

`zabbix_server.conf` の `FrontendAllowedIP` パラメータに Web インターフェースの IP アドレスを追加し、Zabbix サーバーを再起動します。

暗号化されていない接続は許可されていません

Zabbix サーバーのログに、次のようなエラーが記録されます。

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

Zabbix サーバーは、`TLSFrontendAccept=cert` などにより、TLS 接続のみを受け入れるように設定されていますが、Web インターフェースが暗号化なしで接続しようとしています。

Web インターフェースの暗号化を有効にし、設定の手順 4 で説明されているとおり、必要な TLS ファイルを `zabbix.conf.php` に設定してください。

Web インターフェースが TLS を試行しているが、サーバーは暗号化されていない接続を想定している

Zabbix サーバーのログに、次のようなエラーが記録されます。

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

正確なエラーメッセージは、OpenSSL のバージョンによって異なります。

Zabbix サーバーは、`TLSFrontendAccept=unencrypted` などにより、暗号化されていない接続のみを受け入れるように設定されています。一方、Web インターフェースは TLS 接続を確立しようとしています。

`zabbix_server.conf` で `TLSFrontendAccept=cert` を設定し、`TLSCertFile` と `TLSKeyFile` を定義して、Zabbix サーバーが TLS 接続を受け入れるように設定します。または、`zabbix.conf.php` で Web インターフェースの暗号化を無効にします。

Web インターフェースの証明書が信頼されていない

Zabbix サーバーのログに、次のようなエラーが記録されます。

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

Web インターフェースが証明書を提示しましたが、Zabbix サーバーはその証明書に署名した認証局（CA）を信頼していません。

同じ障害が、Web インターフェースのシステム情報に次のように表示される場合もあります。

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

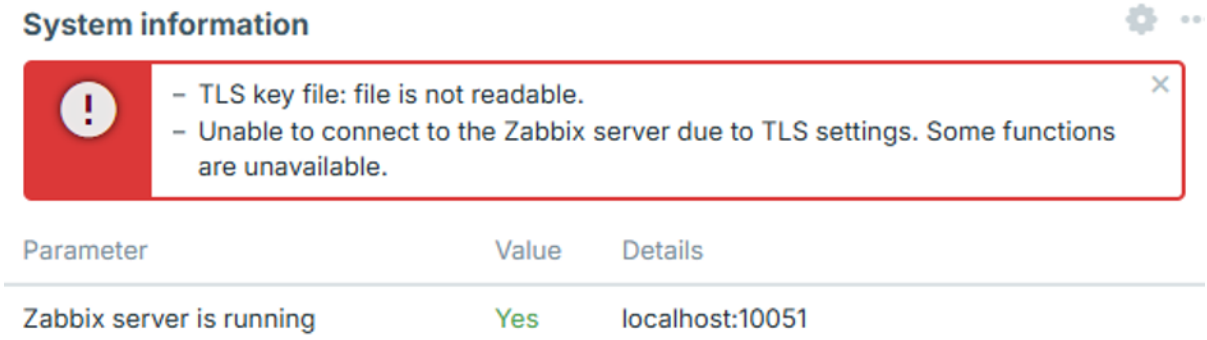
また、Web インターフェースに「Zabbix サーバー"localhost" から正しくない応答を受信しました」と表示される場合もあります。

Web インターフェースの証明書が、`zabbix_server.conf` の `TLSCAFile` で指定された CA によって署名されていることを確認してください。

TLS キーファイルを読み取れない

Web インターフェースに「TLS キーファイル: ファイルを読み取れません。」と表示される場合があります。

また、「TLS 設定が原因で Zabbix サーバーに接続できません。一部の機能は利用できません。」と表示される場合もあります。

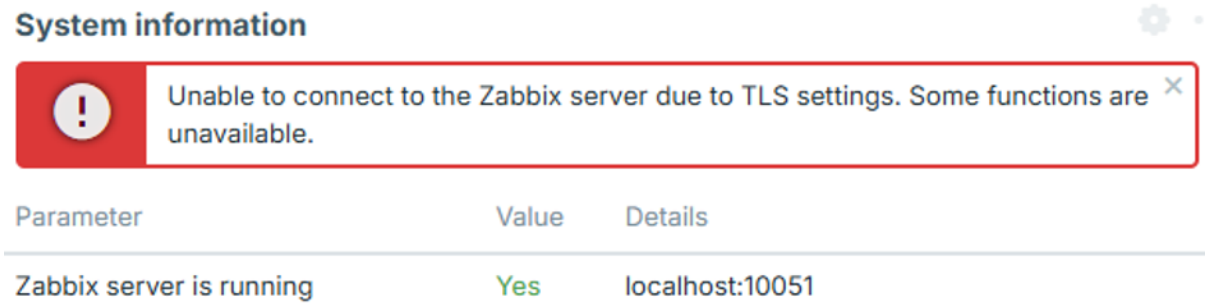


Web サーバーユーザーに、`zabbix.conf.php` で設定されている 1 つ以上の証明書、キー、または CA ファイルを読み取るための十分な権限がありません。これは、証明書ファイルが /root/ やユーザーのホームディレクトリなど、アクセスが制限されたディレクトリに保存されている場合に発生することがあります。

証明書ファイルを Web サーバーがアクセスできるディレクトリに移動し、設定の手順 2 で説明されているように適切な権限を設定してください。

TLS ファイルが無効または破損している

Web インターフェースに「TLS 設定が原因で Zabbix サーバーに接続できません。一部の機能は利用できません。」と表示される場合があります。



設定された証明書ファイルまたはキーファイルが存在し、読み取り可能であっても、その内容が空、破損、またはその他の理由で無効な場合に発生することがあります。この場合、Web インターフェースは Zabbix サーバーへの接続を確立せず、Zabbix サーバーのログにも対応する接続試行は記録されません。

証明書ファイルとキーファイルに、有効で適切にフォーマットされたデータが含まれていること、および秘密鍵が証明書に対応していることを確認してください。

アクセス制御

概要

このセクションでは、アクセス制御を安全に設定するためのベストプラクティスを説明します。

最小権限の原則

ユーザーアカウントは、常に可能な限り少ない権限で実行する必要があります。
これは、Zabbix Web インターフェースのユーザー、データベースユーザー、または Zabbix サーバー/プロキシ/エージェントプロセス用のユーザーには、想定された機能を実行するために必要な権限のみを付与すべきであることを意味します。

Attention:
'zabbix' ユーザーに追加の権限を与えると、設定ファイルにアクセスしたり、インフラストラクチャのセキュリティを侵害し得る操作を実行したりできるようになります。

ユーザーアカウントの権限を設定する際は、Zabbix のWeb インターフェースのユーザータイプ を考慮してください。
Admin ユーザータイプは *Super Admin* ユーザータイプより権限が少ないものの、設定の管理やカスタムスクリプトの実行は可能です。

Note:
一部の情報は、権限のないユーザーでも参照できます。
たとえば、*Alerts* → *Scripts* は *Super Admin* ユーザーのみが利用できますが、スクリプト自体は Zabbix API から取得することもできます。
この場合、スクリプトの権限を制限し、スクリプト内の機密情報（たとえばアクセス資格情報）を除外することで、グローバルスクリプトで利用可能な機密情報の漏えいを防ぐのに役立ちます。

Zabbix agent 用の安全なユーザー

デフォルトでは、Zabbix サーバー、プロキシ、およびエージェント（または agent 2）のプロセスは 1 つの zabbix ユーザーを共有します。
Zabbix エージェント/agent 2（サーバー/プロキシと同じマシンで動作している）が、サーバー/プロキシの設定内にある機密情報（たとえばデータベース認証情報）へアクセスするのを防ぐには、エージェントを別のユーザーで実行する必要があります。

Zabbix エージェントの場合:

1. 安全なグループとユーザー（例: zabbix-agent）を作成します。
2. エージェントの設定ファイルのUser パラメータにこのユーザーを設定します。
3. エージェントを再起動 して、新しいユーザーに権限を切り替えます。

Zabbix agent 2 では、agent 2 の設定ファイル が User パラメータをサポートしていないため、設定は service レベルで適用する必要があります。
例については、ZBX-26442 を参照してください。

SSL 設定への書き込みアクセスを取り消す（Windows）

Windows で Zabbix エージェントをコンパイルしており、OpenSSL が保護されていないディレクトリ（例: C:\zabbix、c:\openssl-64bit、C:\OpenSSL-Win64-111-static、または C:\dev\openssl）に配置されている場合は、このディレクトリに対する管理者以外のユーザーの書き込みアクセスを必ず取り消してください。
そうしないと、エージェントは権限のないユーザーによって変更可能なパスから SSL 設定を読み込むため、潜在的なセキュリティ脆弱性につながる可能性があります。

Zabbix コンポーネントのセキュリティ強化

Zabbix コンポーネントのセキュリティを強化するため、一部の機能を無効にできます。

- Zabbix サーバーでのグローバルスクリプト実行は、サーバー設定で EnableGlobalScripts=0 を設定すると無効にできます。
- Zabbix プロキシでのグローバルスクリプト実行は、デフォルトで無効になっています(プロキシ設定で EnableRemoteCommands=1 を設定すると有効にできます)。
- Zabbix エージェントでのグローバルスクリプト実行は、デフォルトで無効になっています（エージェント設定で、許可する各コマンドに対して AllowKey=system.run[<command>,*] または AllowKeyRegexp=^system\.run\[.*\]$ パラメータを追加すると有効にできます)。
- ユーザーのHTTP 認証設定へのアクセスは、デフォルトで有効になっています。Web インターフェース設定で $ZBX_FEATURE_FLAGS['http_auth_enabled'] = false を設定すると無効にできます。
- モジュール設定へのアクセスは、デフォルトで有効になっています。Web インターフェース設定で $ZBX_FEATURE_FLAGS['modules_c = false を設定すると無効にできます。
- 特定のメディアタイプの編集は、Web インターフェース設定で $ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...] を設定すると制限できます。指定可能な値は'email'、'script'、'sms'、'webhook'、'push'です。1 つ以上の値を指定できます。

zabbix.conf.php 設定ファイルが存在する場合、setup.php を実行して Web インターフェースを再インストールすることは制限されます。

Zabbix エージェントによる Windows での UNC パスへのアクセス

Windows 上の Zabbix エージェントは、vfs.file.*、vfs.dir.*、modbus.get、perf_counter* などのアイテムで、UNC パス（\\server\share\file.txt のような SMB 共有）をたどります。状況によっては、これがセキュリティリスクになる可能性があります。

Windows が UNC パスへのアクセスを要求されると、そのサーバーで認証を試みます。つまり、悪意のあるリクエストを Zabbix エージェントに送信することで、リクエスト元のサーバーに NTLM ハッシュが公開される可能性があります。ユーザーは、必要に応じてAllowKey、AllowKeyRegexp、DenyKey、DenyKeyRegexpの設定パラメータを構成することで、このリスクを軽減できます。

MySQL/MariaDB の保護

概要

このセクションでは、MySQL/MariaDB データベースを保護するためのベストプラクティスを説明します。

Note:
基本的なセットアップについては、デフォルトのMySQL/MariaDB データベース作成 の手順を参照してください。これには、Zabbix データベースに対する完全な権限を持つ'zabbix' ユーザーの作成が含まれます。このユーザーはデータベース所有者であり、アップグレード 時にデータベース構造を変更するために必要な権限も持っています。

セキュリティを向上させるには、最小限の権限を持つ追加のデータベースロールとユーザーを作成することを推奨します。これらのロールとユーザーは、最小権限の原則 に基づいて設定する必要があります。つまり、想定された機能を実行するために不可欠な権限のみを持つべきです。

ユーザーロールの作成

次のロールを、それぞれ対応する権限で作成します。

- **zbx_srv** - Zabbix サーバーとプロキシを実行するためのロール:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - Zabbix Web インターフェースと API を実行するためのロール:

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - テーブルバックアップ用のロール:

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

Note:
テーブルの復元とアップグレードは、データベース所有者が実行する必要があります。

- **zbx_part** - データベースのパーティショニング用に権限を絞ったロール。このロールは、特定のデータベーステーブルに対する権限を付与するため、データベース作成後にのみ作成できることに注意してください。

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- MariaDB の場合: 次の行はスキップします (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

ロールを作成した後、それらをユーザーに割り当てることができます。

ユーザーロールの割り当て

作成したユーザーロールを割り当てるには、ユーザーを作成し、該当するロールを割り当てます。<user>、<host>、<role>、<password> を適切な値に置き換えてください。<password> には、強力で一意のパスワードを設定してください。

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- MariaDB の場合: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

たとえば、Zabbix サーバーおよびプロキシの実行用ロールを作成して割り当てるには、次のようにします。

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
```

```
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

PostgreSQL/TimescaleDB の保護

概要

このセクションでは、PostgreSQL データベースを保護するためのベストプラクティスを説明します。

> **Note:**
> 基本的なセットアップについては、デフォルトのPostgreSQL database creation の手順を参照してください。これには、Zabbix データベースに対してフル権限を持つ'zabbix' ユーザーの作成が含まれます。このユーザーはデータベース所有者であり、アップグレード 時にデータベース構造を変更するために必要な権限も持っています。

セキュリティを向上させるには、安全なスキーマ使用パターンを構成し、さらに最小限の権限を持つ追加のデータベースロールとユーザーを作成することを推奨します。これらのロールとユーザーは、最小権限の原則 に基づいて設定する必要があります。つまり、意図した機能を実行するために不可欠な権限のみを持つべきです。

データベースのセットアップ

データベース所有者となるユーザーを作成し、Zabbix データベースを作成します。
データベース所有者は、データベース作成時に指定したユーザーです。

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

> **Note:**
> データベースのクリーンインストールまたはアップグレードは、データベース所有者が実行する必要があります。
> これは、データベースオブジェクトを削除したり、その定義を変更したりする権限はデータベース所有者に固有の特権であり、付与も取り消しもできないためです。

> **Attention:**
> このページの以下のコマンドは、PostgreSQL への接続が特に zabbix データベースに対して行われている状態で実行する必要があります。

zabbix スキーマを作成し、データベース所有者 (usr_owner) をこのスキーマの所有者として設定します。

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

安全なスキーマの使用パターンを設定します。

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Note: search_path should point to the "zabbix" schema:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

データベースのセットアップ後は、ユーザーのロール作成に進みます。

ユーザーロールの作成

次のロールを、それぞれ対応する権限で作成します。

- **zbx_srv** - Zabbix サーバーとプロキシを実行するためのロール:

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - Zabbix Web インターフェースと API を実行するためのロール:

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - テーブルのバックアップ用ロール:

```
CREATE ROLE zbx_bckp;
GRANT CONNECT ON DATABASE zabbix TO zbx_bckp;
GRANT USAGE ON SCHEMA zabbix TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT ON TABLES TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

> **Note:**
> テーブルの復元は、データベース所有者のみが実行できます。

ロールを作成した後、それらをユーザーに割り当てることができます。

ユーザーロールの割り当て

作成したユーザーロールを割り当てるには、ユーザーを作成し、該当するロールを割り当てます。<user>、<role>、<password> を適切な値に置き換えてください。<password> には、強力で一意のパスワードを指定してください。

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

たとえば、Zabbix サーバーおよびプロキシを実行するためのロールを作成して割り当てるには、次のようにします。

```
CREATE USER usr_srv WITH ENCRYPTED password 'strong-password';
GRANT zbx_srv TO usr_srv;
```

TimescaleDB を使用したデータベースのパーティショニング

データベースのパーティショニングは TimescaleDB によって実現されます。
TimescaleDB を利用するには、Zabbix サーバーにデータベース所有者権限が必要です。

PostgreSQL の zabbix スキーマがすでに zabbix データベース内に作成されている場合は、次のコマンドで TimescaleDB を有効にできます。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

暗号化

概要

このセクションには、暗号化を安全に設定するためのベストプラクティスが記載されています。

Zabbix Web インターフェースの SSL 設定

RHEL ベースのシステムでは、mod_ssl パッケージをインストールします:

```
dnf install mod_ssl
```

SSL キー用のディレクトリを作成します:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

SSL 証明書を作成します:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

プロンプトには適切な値を入力してください。最も重要なのは Common Name を求められる行です。サーバーに関連付けたいドメイン名を入力する必要があります。ドメイン名がない場合は、代わりに公開 IP アドレスを入力できます。

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Apache の SSL 設定ファイル（/etc/httpd/conf.d/ssl.conf）を編集します:

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

変更を適用するため、Apache サービスを再起動します:

```
systemctl restart httpd.service
```

Web サーバーの保護

概要

このセクションでは、Web サーバーを安全に設定するためのベストプラクティスを説明します。

Zabbix SSL へのルート URL リダイレクトの強制

RHEL ベースのシステムでは、Apache の設定 (`/etc/httpd/conf/httpd.conf`) に仮想ホストを追加し、ドキュメントルートを Zabbix SSL URL へ恒久的にリダイレクトするよう設定します。
`example.com` は実際のサーバー名に置き換えてください。

```
### 次の行を追加します:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

変更を適用するには、Apache サービスを再起動します:

```
systemctl restart httpd.service
```

Web サーバーで HTTP Strict Transport Security (HSTS) を有効にする

Zabbix Web インターフェースをプロトコルダウングレード攻撃から保護するため、Web サーバーで HSTS ポリシーを有効にすることを推奨します。

Apache の設定で Zabbix Web インターフェースに対して HSTS ポリシーを有効にするには、次の手順に従います。

1. 仮想ホストの設定ファイルを見つけます。

- `/etc/httpd/conf/httpd.conf`（RHEL ベースのシステム）
- `/etc/apache2/sites-available/000-default.conf`（Debian/Ubuntu）

2. 仮想ホストの設定ファイルに次のディレクティブを追加します。

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. 変更を適用するために Apache サービスを再起動します。

```
### On RHEL-based systems:
systemctl restart httpd.service

### On Debian/Ubuntu
systemctl restart apache2.service
```

Zabbix での Secure および SameSite セッション Cookie の強制

Zabbix を設定する際は、セキュリティを強化し、クロスサイトリクエストフォージェリ（CSRF）攻撃を防ぐために、セッション Cookie に対して secure 属性と SameSite 属性を強制することが重要です。ただし、`SameSite=Strict` を強制すると、次のような特定のシナリオで問題が発生する場合があります。

- 同一ドメインの iframe を埋め込んだ場合に、ダッシュボードの URL ウィジェットで「user not logged in」と表示される。
- HTTPS ではなく HTTP 経由でダッシュボードにアクセスするユーザーが、ログインの問題に直面する可能性がある。
- 特定の Zabbix メニューセクションやホストへの URL を共有できない。

これらの問題を軽減するため、ユーザーが SameSite ポリシーを調整できるようにする必要があります。

1. Secure cookies

secure フラグを設定すると、Cookie は HTTPS 経由でのみ送信され、暗号化されていない接続での漏えいを防げます。

Zabbix で secure cookies を有効にするには、Web サーバーの設定に次の設定を追加または変更します。

Apache の場合:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Nginx の場合:

```
proxy_cookie_path / "/; Secure";
```

Zabbix の Web インターフェースに HTTPS 経由でアクセスしていることを確認してください。そうでない場合、Secure フラグ付きの Cookie は送信されません。

2. SameSite 属性の設定

Web サーバーの設定でも SameSite 属性を強制できます。

Apache の場合:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Nginx（version 1.19.3+）の場合:

```
proxy_cookie_flags ~ samesite=Strict; # Replace ~ with 'zbx_session' for specificity
```

Web サーバーで Content Security Policy (CSP) を有効にする

Zabbix Web インターフェースをクロスサイトスクリプティング (XSS)、データインジェクション、および同様の攻撃から保護するため、Web サーバーで Content Security Policy を有効にすることを推奨します。
そのためには、Web サーバーが HTTP ヘッダー を返すように設定します。

> **Attention:**
> 以下の CSP ヘッダー設定は、デフォルトの Zabbix Web インターフェースのインストール、およびすべてのコンテンツがサイトのドメイン（サブドメインを除く）から提供される場合にのみ適用されます。
> たとえば、*URL* ウィジェットを設定してサイトのサブドメインや外部ドメインからコンテンツを表示する場合、*OpenStreetMap* から別の地図エンジンに切り替える場合、または外部 CSS やウィジェットを追加する場合は、別の CSP ヘッダー設定が必要になることがあります。
> Duo Universal Prompt の多要素認証 メソッドを使用している場合は、仮想ホストの設定ファイルにある CSP ディレクティブへ”duo.com” を追加してください。

Apache 設定で Zabbix Web インターフェースの CSP を有効にするには、次の手順に従います。

1. 仮想ホストの設定ファイルを見つけます:

- RHEL ベースのシステムでは `/etc/httpd/conf/httpd.conf`
- Debian/Ubuntu では `/etc/apache2/sites-available/000-default.conf`

2. 仮想ホストの設定ファイルに次のディレクティブを追加します:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. 変更を適用するために Apache サービスを再起動します:

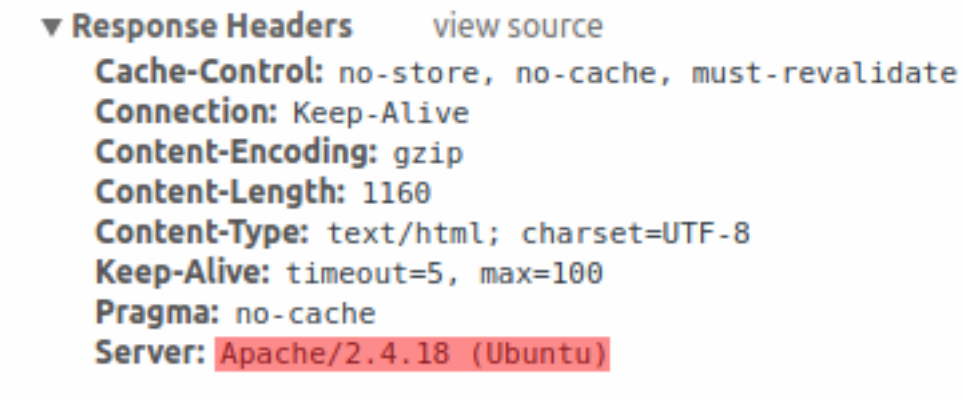
```
### RHEL ベースのシステムでは:
systemctl restart httpd.service

### Debian/Ubuntu では
systemctl restart apache2.service
```

Web サーバー情報の公開を無効にする

セキュリティを向上させるため、すべての Web サーバー署名を無効にすることを推奨します。

デフォルトでは、Web サーバーはソフトウェア署名を公開します。



Apache の設定ファイルに次のパラメータを追加すると、署名を無効にできます。

```
ServerSignature Off
ServerTokens Prod
```

PHP の署名（X-Powered-By HTTP ヘッダー）は、`php.ini` 設定ファイルを変更することで無効にできます（デフォルトでは署名は無効になっています）。

```
expose_php = Off
```

設定ファイルの変更を適用するには、Web サーバーを再起動する必要があります。

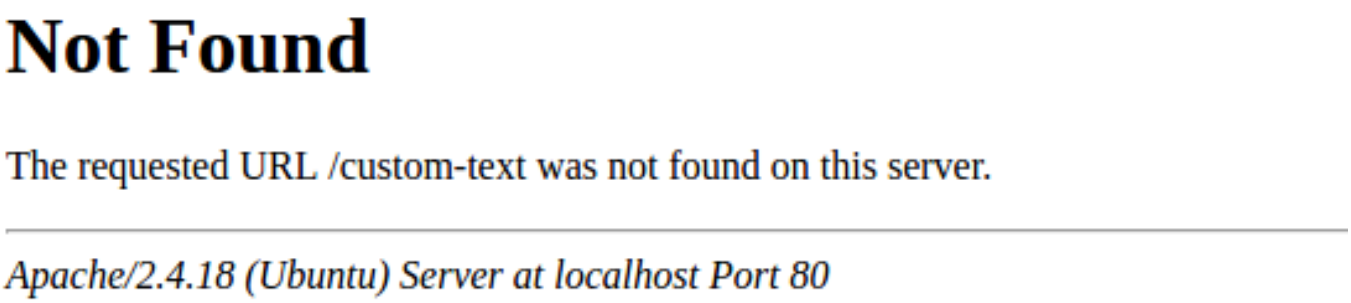
さらにセキュリティを強化するには、Apache で *mod_security* ツール（パッケージ *libapache2-mod-security2*）を使用できます。このツールを使用すると、Web サーバー署名からバージョンだけを削除するのではなく、サーバー署名自体を削除できます。*mod_security* のインストール後に「SecServerSignature」に任意の値を設定すると、サーバー署名を任意の値に変更できます。

ソフトウェア署名を削除または変更する方法については、ご使用の Web サーバーのドキュメントを参照してください。

Web サーバーのデフォルトエラーページを無効にする

情報漏えいを防ぐため、デフォルトエラーページを無効にすることを推奨します。

デフォルトでは、Web サーバーは組み込みのエラーページを使用します。



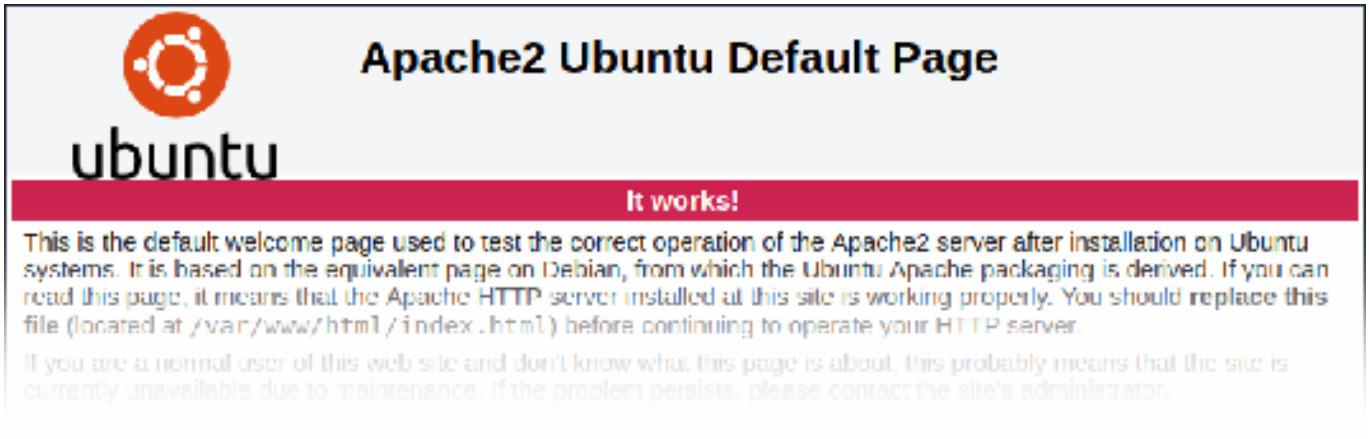
これらのデフォルトエラーページは、置き換えるか削除してください。たとえば、Apache Web サーバーでは、「ErrorDocument」ディレクティブを使用して、カスタムエラーページまたはエラーテキストを定義できます。

デフォルトエラーページを置き換える、または削除する方法については、ご使用の Web サーバーのドキュメントを参照してください。

Web サーバーのテストページの削除

情報漏えいを防ぐため、Web サーバーのテストページを削除することを推奨します。

デフォルトでは、Apache Web サーバーの Web ルートに `index.html` テストページが含まれています。



デフォルトのテストページを削除する方法については、ご使用の Web サーバーのドキュメントを参照してください。

X-Frame-Options HTTP レスポンスヘッダーの設定

デフォルトでは、Zabbix は *Use X-Frame-Options HTTP header* パラメータが SAMEORIGIN に設定された状態で構成されています。これは、コンテンツがページ自体と同じオリジンを持つフレーム内でのみ読み込めることを意味します。

外部 URL からコンテンツを取得する Zabbix Web インターフェースの要素（具体的には、URL dashboard widget）は、すべてのサンドボックス制限が有効なサンドボックス内で取得したコンテンツを表示します。

これらの設定により、Zabbix Web インターフェースのセキュリティが強化され、XSS およびクリックジャッキング攻撃に対する保護が提供されます。
Super admin ユーザーは、必要に応じて *Use iframe sandboxing* および *Use X-Frame-Options HTTP header* パラメータを変更できます。
デフォルト設定を変更する前に、リスクと利点を慎重に検討してください。
iframe サンドボックス化または X-Frame-Options HTTP ヘッダーを完全に無効にすることは推奨されません。

一般的なパスワード一覧ファイルを非表示にする

パスワードのブルートフォース攻撃の難易度を高めるために、ui/data/top_passwords.txt ファイルへのアクセスを制限することを推奨します。
このファイルには、最も一般的でコンテキスト固有のパスワード一覧が含まれており、そのようなパスワードの設定をユーザーが行えないようにします（パスワードポリシーで 推測されやすいパスワードを避けるパラメータが有効な場合）。

top_passwords.txt ファイルへのアクセスを制限するには、Web サーバーの設定を変更します。

Apache では、.htaccess ファイルを使用してファイルアクセスを制限できます。

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

NGINX では、location ディレクティブを使用してファイルアクセスを制限できます。

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

シークレットの保存

概要　Zabbix は、セキュアな vault から機密情報を取得するように設定できます。以下のシークレット管理サービスがサポートされています: HashiCorp Vault KV Secrets Engine - Version 2、CyberArk Vault CV12。

シークレットは、次の取得に使用できます:

- ユーザーマクロの値
- データベースアクセス資格情報

Zabbix は、シークレットが別の担当者によって管理されていることを前提として、vault 内のシークレットに対して読み取り専用アクセスを提供します。

特定の vault プロバイダーの設定については、次を参照してください。

- HashiCorp configuration
- CyberArk configuration

シークレット値のキャッシュ　デフォルトでは、Vault のシークレットマクロの値は、設定データが更新されるたびに Zabbix サーバーによって取得され、設定キャッシュに保存されます。Zabbix プロキシは、設定同期のたびに Zabbix サーバーから Vault のシークレットマクロの値を受信し、独自の設定キャッシュに保存します。

> **Attention:**
> Zabbix サーバーとプロキシ間の暗号化を有効にする必要があります。有効にしない場合、サーバーの警告メッセージがログに記録されます。

また、マクロの値を Zabbix サーバーと Zabbix プロキシがそれぞれ独立して取得するように設定することもできます。

Vault からキャッシュされたシークレット値の更新を手動で実行するには、`secrets_reload` コマンドラインオプションを使用します。

Zabbix Web インターフェースのデータベース認証情報のキャッシュはデフォルトで無効になっていますが、Web インターフェース設定ファイルで `$DB['VAULT_CACHE'] = true` オプションを設定することで有効にできます。認証情報は、ファイルシステムの一時ファイルディレクトリを使用してローカルキャッシュに保存されます。Web サーバーは、専用の一時フォルダーへの書き込みを許可する必要があります（たとえば、Apache では設定オプション `PrivateTmp=True` を設定する必要があります）。データキャッシュを更新または無効化する頻度を制御するには、`ZBX_DATA_CACHE_TTL`定数を使用します。

TLS 設定　Zabbix コンポーネントと vault 間の通信で TLS を設定するには、認証局（CA）によって署名された証明書をシステム全体のデフォルト CA ストアに追加します。
別の場所を使用する場合は、Zabbix のserver/proxy設定パラメーターで SSLCALocation ディレクトリを指定し、
そのディレクトリ内に証明書ファイルを配置してから、CLI commandを実行します:

```
c_rehash .
```

CyberArk の設定

このセクションでは、Zabbix が CyberArk Vault CV12 からシークレットを取得するように設定する方法を説明します。

Vault は、公式のCyberArk documentationに記載されているとおりにインストールおよび設定されている必要があります。

Zabbix での TLS 設定については、*Storage of secrets*を参照してください。

データベース認証情報

データベース認証情報を含むシークレットへのアクセスは、各 Zabbix コンポーネントで個別に設定します。

サーバーとプロキシ

Zabbix サーバーまたはプロキシのデータベース認証情報をボールトから取得するには、設定ファイルで次の設定パラメータを指定します。

- `Vault` - 使用するボールトプロバイダー。
- `VaultURL` - ボールトサーバーの HTTP(S) URL。
- `VaultDBPath` - データベース認証情報を含むボールトシークレットへのクエリ。認証情報は「Content」および「UserName」キーから取得されます（このオプションは、DBUser と DBPassword が指定されていない場合にのみ使用できます）。
- `VaultTLSCertFile`、`VaultTLSKeyFile` - SSL 証明書およびキーファイルの名前。これらのオプションの設定は必須ではありませんが、強く推奨されます。
- `VaultPrefix` - ボールトに応じた、ボールトパスまたはクエリのカスタムプレフィックス。指定しない場合は、最適なデフォルト値が使用されます。

> **Attention:**
> `Vault`、`VaultURL`、`VaultTLSCertFile`、`VaultTLSKeyFile`、`VaultPrefix` 設定パラメータは、Zabbix サーバー（および設定されている場合は Zabbix プロキシ）がシークレットボールトマクロを処理する際のボールト認証にも使用されます。Zabbix サーバーとプロキシは、VaultDBPath に含まれるデータベース認証情報を含むボールトシークレットマクロを開きません。

Zabbix サーバーと Zabbix プロキシは、起動時に `zabbix_server.conf` および `zabbix_proxy.conf` ファイルからボールト関連の設定パラメータを読み取ります。

例

1. zabbix_server.conf で、以下のパラメータを指定します:

```
Vault=CyberArk
VaultURL=https://127.0.0.1:1858
VaultDBPath=AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
VaultTLSCertFile=cert.pem
VaultTLSKeyFile=key.pem
VaultPrefix=/AIMWebService/api/Accounts?
```

2. Zabbix は、以下の API リクエストを vault に送信します:

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. vault の応答には、"Content" と "UserName" のキーが含まれます:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

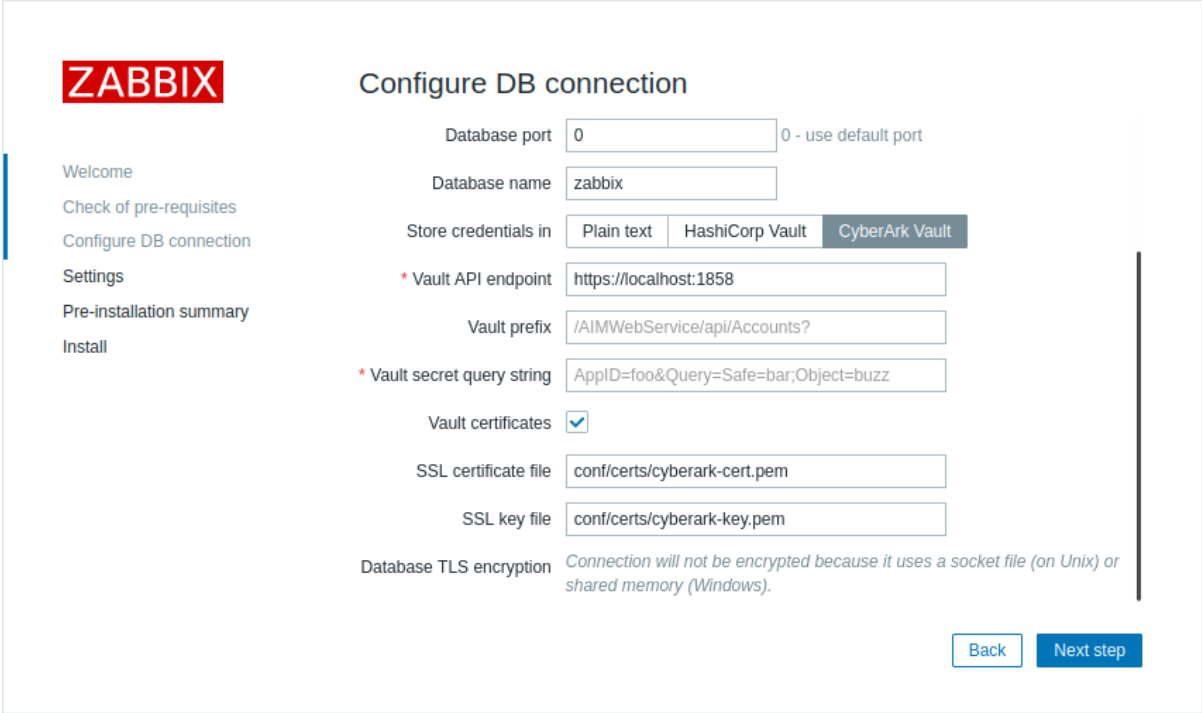
4. その結果、Zabbix はデータベース認証に以下の認証情報を使用します:

- ユーザー名: <username>
- パスワード: <password>

Web インターフェース

Zabbix Web インターフェースのデータベース認証情報を vault から取得するには、Web インターフェースのインストール時に以下のパラメータを指定します。

1. *Configure DB Connection* の手順で、*Store credentials in* パラメータに *CyberArk Vault* を設定します。



2. 次に、追加のパラメータを入力します。

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:1858	scheme://host:port 形式で vault への接続に使用する URL を指定します。

Parameter	Mandatory	Default value	Description
Vault prefix	no	/AIMWebService/api/Accounts?	Vault のパスまたはクエリに使用するカスタムプレフィックスを指定します。指定しない場合は、デフォルト値が使用されます。
Vault secret query string	yes		データベース認証情報の取得元を指定するクエリです。 例: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		チェックボックスを選択すると、クライアント認証を設定するための追加パラメータが表示されます。このパラメータは任意ですが、CyberArk Vault との通信では有効にすることを強く推奨します。
SSL certificate file	no	conf/certs/cyberark-cert.pem	SSL 証明書ファイルへのパスです。ファイルは PEM 形式である必要があります。 証明書ファイルに秘密鍵も含まれている場合は、SSL key file パラメータを空のままにします。
SSL key file	no	conf/certs/cyberark-key.pem	クライアント認証に使用する SSL 秘密鍵ファイルの名前です。ファイルは PEM 形式である必要があります。

ユーザーマクロの値

Vault secret のユーザーマクロ値を保存するために CyberArk Vault を使用するには、次の条件を満たしていることを確認してください。

- Zabbix サーバーが CyberArk Vault と連携するように設定されていること。
- *Administration > General > Other* の *Vault provider* パラメータが”CyberArk Vault” に設定されていること。



Note:
Zabbix サーバー（および設定されている場合は Zabbix プロキシ）には、vault から *Vault secret* マクロ値へアクセスする権限が必要です。Zabbix Web インターフェースにはそのようなアクセスは必要ありません。

マクロ値にはクエリ（`query:key` 形式）を含める必要があります。

Zabbix によるマクロ値の処理の詳細については、*Vault secret macros* を参照してください。

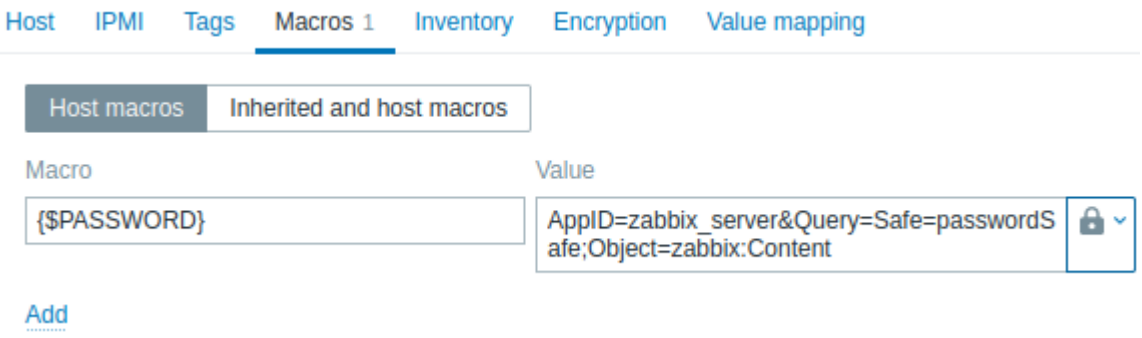
クエリ構文

コロン記号（`:`）は、クエリとキーを区切るために予約されています。

クエリ自体にスラッシュまたはコロンが含まれる場合、これらの記号は URL エンコードする必要があります（`/` は `%2F`、`:` は `%3A` としてエンコードされます）。

例

1. Zabbix で、タイプが *Vault secret* で、値が `AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content` のユーザーマクロ `{$PASSWORD}` を追加します。



2. Zabbix は、次の API リクエストを Vault に送信します。

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
```

```
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. Vault のレスポンスには、キー Content が含まれます。

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. その結果、Zabbix はマクロ {$PASSWORD} を値 <password> に解決します。

既存の設定の更新

CyberArk Vault からシークレットを取得する既存の設定を更新するには、次の手順を実行します。

1. データベース認証情報セクションの説明に従って、Zabbix サーバーまたはプロキシの設定ファイルのパラメータを更新します。
2. Zabbix のWeb インターフェース設定ファイルのパラメータを更新します。

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. 必要に応じて、ユーザーマクロの値セクションの説明に従ってユーザーマクロを設定します。

HashiCorp Vault からシークレットを取得する既存の設定を更新するには、*HashiCorp の設定*を参照してください。

HashiCorp 設定

概要

このセクションでは、HashiCorp Vault KV Secrets Engine - Version 2 からシークレットを取得するための Zabbix の設定方法について説明します。

Vault は、公式の HashiCorp documentation に記載されているとおりにデプロイおよび設定されている必要があります。

Zabbix での TLS の設定については、*Storage of secrets* を参照してください。

データベース認証情報の取得

データベース認証情報を含むシークレットを正常に取得するには、次の両方を設定する必要があります。

- Zabbix サーバー/プロキシ
- Zabbix Web インターフェース

サーバー/プロキシ

Zabbix サーバーまたはプロキシを設定するには、設定ファイルで次の設定パラメータを指定します。

- `Vault` - 使用する vault プロバイダー。
- `VaultToken` - vault 認証トークン（詳細は Zabbix サーバー/プロキシの設定ファイルを参照）。`VaultAppRoleID` とは併用できません。
- `VaultAppRoleID` および `VaultAppSecretID` - vault 認証のロール ID およびシークレット ID（詳細は Zabbix サーバー/プロキシの設定ファイルを参照）。`VaultToken` とは併用できません。
- `VaultURL` - vault サーバーの HTTP(S) URL。
- `VaultDBPath` - データベース認証情報を含む vault シークレットへのパス（このオプションは、DBUser と DBPassword が指定されていない場合にのみ使用できます）。Zabbix サーバーまたはプロキシは、キー”password” および”username” を使用して認証情報を取得します。
- `VaultPrefix` - vault に応じた、vault パスまたはクエリのカスタムプレフィックス。指定しない場合は、最適なデフォルト値が使用されます。プレフィックスは / で始まり、/ で終わる必要があります。

Attention:
Vault、VaultToken（または VaultAppRoleID/VaultAppSecretID）、VaultURL、および VaultPrefix の設定パラメータは、Zabbix サーバーが vault シークレットマクロを処理する際の vault 認証にも使用されます（設定されている場合は Zabbix プロキシでも使用されます）。Zabbix サーバーおよびプロキシは、VaultDBPath のデータベース認証情報を含む vault シークレットマクロを展開しません。

 プロキシごとに異なるトークンを使用することを強く推奨します。

Zabbix サーバーおよび Zabbix プロキシは、起動時に zabbix_server.conf および zabbix_proxy.conf から vault 関連の設定パラメータを読み取ります。さらに、Zabbix サーバーおよび Zabbix プロキシは起動時に VAULT_TOKEN 環境変数を一度読み取り、fork されたスクリプトから利用できないように unset します。VaultToken と VAULT_TOKEN の両方のパラメータに値が含まれている場合はエラーになります。

例

1. zabbix_server.conf で、以下のパラメータを指定します。

VaultToken を使用する場合:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

または、VaultAppRoleID を使用する場合:

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. 以下の CLI コマンドを実行して、vault に必要な secret を作成します。

```
#### まだ有効になっていない場合は "secret/" マウントポイントを有効化します。なお、"kv-v2" を使用する必要があります。
vault secrets enable -path=secret/ kv-v2

#### マウントポイント "secret/" とパス "zabbix/database" の下に、username と password のキーを持つ新しい secret を追
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### secret が正常に追加されたことを確認します。
vault kv get secret/zabbix/database

#### 最後に Curl でテストします。なお、"data" はマウントポイントの後ろに手動で追加し、"/v1" はマウントポイントの前に追加する
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. その結果、Zabbix サーバーはデータベース認証用に以下の認証情報を取得します。

- ユーザー名: zabbix
- パスワード: <password>

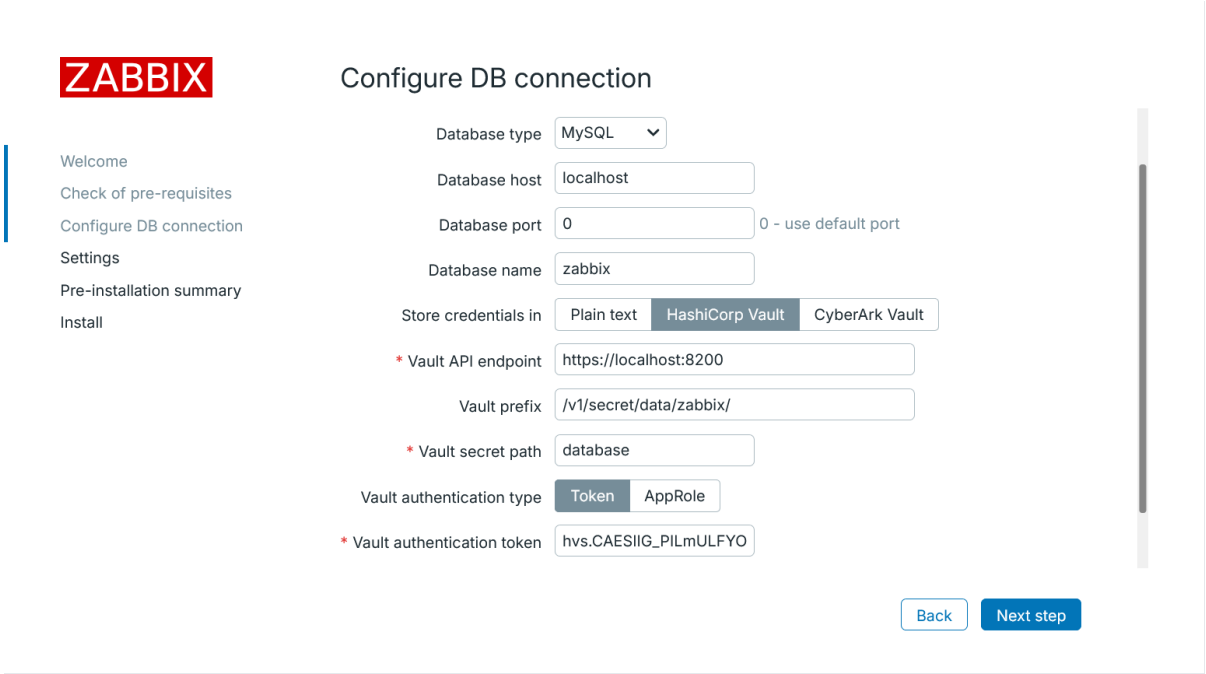
Web インターフェース

Zabbix Web インターフェースでは、インストール時、または Web インターフェースの設定ファイル（zabbix.conf.php）を更新することで、vault からデータベースの認証情報を取得するように設定できます。

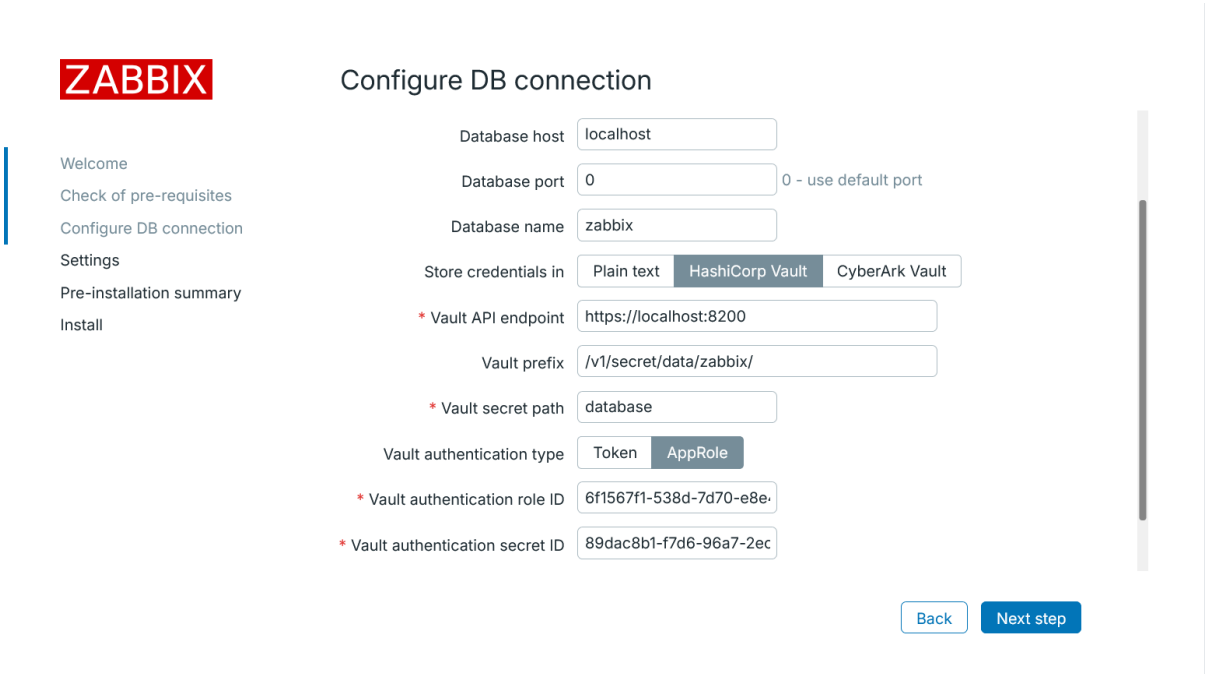
Attention:
前回の Web インターフェースのインストール以降に vault の認証情報が変更された場合は、既存の設定を更新してください。

Web インターフェースのインストール中に、*Configure DB Connection*の手順で接続パラメータを指定する必要があります。

トークン認証を使用した設定例:



AppRole 認証を使用した設定例:



パラメータ	必須	デフォルト値	説明
Store credentials in	いいえ		**HashiCorp Vault** を選択します。
Vault API endpoint	はい	https://localhost:8200	Vault への接続に使用する URL を `scheme://host:port` 形式で指定します。
Vault prefix	いいえ	/v1/secret/data/	vault のパスまたはクエリに使用するカスタムプレフィックスを指定します。指定しない場合はデフォルト値が使用されます。 例: `/v1/secret/data/zabbix/`
Vault secret path	はい		データベースの認証情報をキー「password」および「username」で取得するシークレットへのパスを指定します。 例: `database`
Vault authentication type	いいえ		vault の認証タイプを選択します。 **Token** - トークン認証を使用します。 **AppRole** - AppRole 認証方式を使用します。

パラメータ	必須	デフォルト値	説明
Vault authentication token	はい		シークレットパスへの読み取り専用アクセスに使用する認証トークンを指定します。 トークンおよび vault ポリシーの作成方法については、HashiCorp のドキュメントを参照してください。 このフィールドは、vault の認証タイプとして *Token* が選択されている場合にのみ使用できます。
Vault authentication role ID	はい		AppRole のロール ID を指定します。 AppRole の詳細については、HashiCorp のドキュメントを参照してください。 このフィールドは、vault の認証タイプとして *AppRole* が選択されている場合にのみ使用できます。
Vault authentication secret ID	はい		AppRole に対して発行されたシークレット ID を指定します。 AppRole の詳細については、HashiCorp のドキュメントを参照してください。 このフィールドは、vault の認証タイプとして *AppRole* が選択されている場合にのみ使用できます。

ユーザーマクロ値の取得

Vault secret ユーザーマクロ値を保存するために HashiCorp Vault を使用するには、次の条件を満たしていることを確認してください。

- Zabbix サーバー/プロキシが HashiCorp Vault と連携するように設定 されている。
- *Administration > General > Other* の *Vault provider* パラメータが”HashiCorp Vault”（デフォルト）に設定されている。



Note:
Zabbix サーバー（および設定 されている場合の Zabbix プロキシ）は、vault から *Vault secret* マクロ値へアクセスする必要があります。Zabbix Web インターフェースにはそのようなアクセスは必要ありません。

マクロ値には参照パスを `path:key` の形式で含める必要があります。たとえば、`macros:password` です。Zabbix サーバー/プロキシの設定時に指定した認証トークン（`VaultToken` パラメータ）は、このパスに対する読み取り専用アクセス権を提供している必要があります。

Zabbix によるマクロ値の処理の詳細については、*Vault secret macros* を参照してください。

パス構文

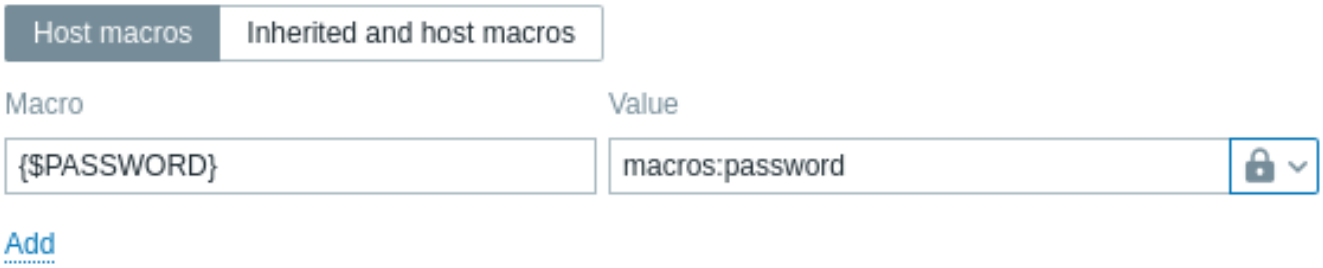
スラッシュ（”/”）とコロン（”:”）の記号は予約されています。

スラッシュは、マウントポイントとパスを区切るためにのみ使用できます（例: *secret/zabbix*。この場合、マウントポイントは”secret”、パスは”zabbix” です）。Vault マクロの場合、コロンはパス/クエリとキーを区切るためにのみ使用できます。

スラッシュとコロンの記号は、スラッシュで区切られる名前のマウントポイントを作成する必要がある場合（例: *foo/bar/zabbix*。この場合、マウントポイントは”foo/bar”、パスは”zabbix” であり、”foo%2Fbar/zabbix” としてエンコードできます）、またはマウントポイント名やパスにコロンを含める必要がある場合に、URL エンコードできます。

例

1. Zabbix で、タイプが”Vault secret” で値が `macros:password` のユーザーマクロ `{$PASSWORD}` を追加します。



2. 次の CLI コマンドを実行して、vault に必要な secret を作成します。

```
#### "secret/" マウントポイントがまだ有効でない場合は有効化します。なお、"kv-v2" を使用する必要があります。
vault secrets enable -path=secret/ kv-v2

#### マウントポイント "secret/" とパス "zabbix/macros" の下に、キー "password" を持つ新しい secret を追加します。
vault kv put -mount=secret zabbix/macros password=<password>

#### secret が正常に追加されたことを確認します。
vault kv get secret/zabbix/macros

#### 最後に Curl でテストします。なお、"data" はマウントポイントの後ろに手動で追加し、"/v1" はマウントポイントの前に追加する
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. その結果、Zabbix はマクロ {$PASSWORD} を次の値に解決します: <password>

既存の設定の更新

HashiCorp Vault からシークレットを取得するための既存の設定を更新するには、次の手順を実行します。

1. データベース認証情報セクションの説明に従って、Zabbix サーバーまたはプロキシの設定ファイルのパラメータを更新します。
2. Zabbix のフロントエンド設定ファイルのパラメータを更新します。

Vault トークンを使用する場合:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

または、Vault AppRole 方式を使用する場合:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. 必要に応じて、ユーザーマクロの値セクションの説明に従ってユーザーマクロを設定します。

CyberArk Vault からシークレットを取得するための既存の設定を更新するには、*CyberArk の設定*を参照してください。

トラブルシューティング

1. Zabbix サーバー/プロキシのログファイルで:

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

考えられる原因: HashiCorp Vault が（再）起動されましたが、unseal されていません（デフォルト設定では必要です。Auto Unseal が設定されている場合は不要です）。

2. Zabbix サーバー/プロキシのログファイルで:

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

または

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

考えられる原因: VaultAppSecretID が HashiCorp vault で期限切れになったか、使用回数の上限に達しています。

Google Chrome TLS 証明書の信頼

概要

このページでは、自己署名証明書またはプライベート認証局で保護された Zabbix Web インターフェース、または zabbix-web-service エンドポイントで Google Chrome を使用するための Zabbix のセットアップ手順と設定例を説明します。

これらの手順では、対象の Web サーバーがすでに HTTPS 用に設定されていることを前提としています。Zabbix Web インターフェースで TLS を設定する方法については、Web インターフェースへの安全な接続 を参照してください。

設定

Linux 上の Google Chrome は、信頼済み証明書用にユーザーごとの NSS 証明書データベースを使用します。
Chrome に自己署名証明書またはプライベート認証局を信頼させるには、Chrome を実行するユーザーアカウントの NSS データベースに必要な証明書を追加します。

1. 必要なパッケージをインストールします。

Debian/Ubuntu の場合:

```
sudo apt install ca-certificates libnss3-tools
```

RHEL ベースのシステムの場合:

```
sudo dnf install ca-certificates nss-tools
```

または:

```
sudo yum install ca-certificates nss-tools
```

2. 証明書ファイルを準備します。

必要な各証明書について、自己署名証明書用の PEM ベースの `.crt` ファイルを作成するか、ルート認証局証明書用の PEM ベースの `.crt` ファイルを個別に作成し、必要に応じて各中間証明書用の `.crt` ファイルも作成します。

3. NSS データベースのディレクトリを作成します。

Chrome を実行するユーザーのホームディレクトリを使用します。

例:

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

対象システム上の Chrome が別の NSS データベースの場所を使用している場合は、そのディレクトリを使用してください。

証明書ファイルが Google Chrome を実行するユーザーから読み取り可能であることを確認してください。
必要に応じて、そのユーザーがアクセスできる場所にコピーします。

4. 証明書をインポートします。

自己署名証明書を使用する場合:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix self-signed certificate" \
  -i /path/to/self-signed.crt
```

プライベート認証局を使用する場合は、ルート認証局証明書をデータベースにインポートします:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix root certificate authority" \
  -i /path/to/root-ca.crt
```

中間証明書を使用する場合は、各証明書を個別にインポートします:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix intermediate certificate" \
  -i /path/to/intermediate-ca.crt
```

5. 証明書データベースを確認します。

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Chrome をテストします。

Google Chrome は、起動したユーザーアカウントの証明書データベースを使用します。
ユーザーアカウントに書き込み可能なホームディレクトリがない場合は、テストを実行する前に書き込み可能なホームディレクトリと XDG ディレクトリを設定してください。

例 (google-chrome の代わりに Chromium を使用する場合は chromium に置き換えてください。):

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

設定が正しければ、このコマンドは要求したページの HTML を返します。

Chrome が証明書を信頼しない場合、このコマンドは NET::ERR_CERT_AUTHORITY_INVALID などのメッセージを含む証明書エラーページを返します。

トラブルシューティング

Error	Possible cause
NET::ERR_CERT_AUTHORITY_INVALID	証明書が、使用したユーザープロファイルにインポートされている、ルート認証局証明書が不足している、または中間証明書が不足している。
NET::ERR_CERT_COMMON_NAME_INVALID	証明書が、ブラウザーの URL で使用されているホスト名と一致していない。
NET::ERR_CERT_DATE_INVALID	証明書の有効期限が切れている、まだ有効になっていない、またはシステムクロックが正しくない。
Chrome still shows a certificate warning after import	Chrome が現在のユーザーに対して別の NSS データベースを使用しているか、証明書の信頼フラグが正しくない。

セキュリティのベストプラクティス

概要　このセクションでは、Zabbix を安全にセットアップするためのベストプラクティスについて説明します。

このセクションのプラクティスは、Zabbix の動作に必須ではありませんが、システムのセキュリティを向上させるために推奨されます。

Zabbix モバイルアプリと Zabbix ブリッジアダプターの安全性を維持するためのベストプラクティスについては、モバイルアクセスのセキュリティ保護を参照してください。

UTF-8 エンコーディング　UTF-8 は、Zabbix でサポートされている唯一のエンコーディングです。これは、セキュリティ上の欠陥なく動作することが確認されています。他のエンコーディングの一部を使用する場合、既知のセキュリティ上の問題があることに注意してください。

Windows インストーラのパス　Windows インストーラを使用する場合は、インストーラが提供する既定のパスを使用することを推奨します。
適切な権限なしにカスタムパスを使用すると、インストールのセキュリティが損なわれる可能性があります。

ユーザー定義のグローバルスクリプトにおけるマクロ　セキュリティを強化するため、ユーザー定義のグローバルスクリプトでは、通常のマクロではなくマクロ関数を使用することを推奨します。マクロは自動的にエスケープされないためです。

Zabbix セキュリティアドバイザリと **CVE** データベース　*Zabbix Security Advisories and CVE database* を参照してください。

HTML メールテンプレート　HTML メールに使用するメッセージテンプレートを作成または編集する場合は、すべてのマクロを必ず htmlencode() マクロ関数で囲んでください。例:

```
<b>Problem started</b> at {{EVENT.TIME}.htmlencode()} on {{EVENT.DATE}.htmlencode()}<br><b>Problem name:</
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Severity:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Operational
data:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>Original problem ID:</b> {{EVENT.ID}.htmlencode()}<br>{{TRIG
```

htmlencode() を使用すると、マクロの値に含まれる HTML 文字がエスケープされ、通知への HTML の挿入を防止できます（たとえば、攻撃者が通知に悪意のあるリンクやフィッシングリンクを挿入することを防ぎます）。

注: Zabbix が提供するデフォルトの HTML メールメッセージでは、すでにマクロに htmlencode() が適用されています。このガイダンスは、既存のテンプレートを編集する場合や新しいテンプレートを作成する場合に適用されます。テンプレートを使用して HTML メールを送信する前に、マクロがエンコードされていることを確認してください。

Web インターフェース

概要　どこからでも、またどのプラットフォームからでも Zabbix に簡単にアクセスできるように、Web ベースのインターフェースが提供されています。

> **Note:**
> 複数の Web インターフェースインスタンスを使用する場合は、すべての Web インターフェースでロケールと言語ライブラリ（LDAP、SAML など）が同じようにインストールおよび設定されていることを確認してください。

Web インターフェースのヘルプ　Zabbix の Web インターフェースのセクションおよびフォームの右上には、ヘルプリンク が用意されています。このリンクから、ドキュメント内の対応する箇所に直接移動できます。

ヘルプアイコン が付いたフォームフィールドでは、アイコンにカーソルを合わせると、対応するヒントが表示されます。

キオスクモード　キオスクモードボタン（右上にあります）が表示される Web インターフェースのセクションでは、ボタンをクリックすると、そのセクションがキオスクモードに切り替わります。キオスクモードでは、セクションのコンテンツのみが表示されます。

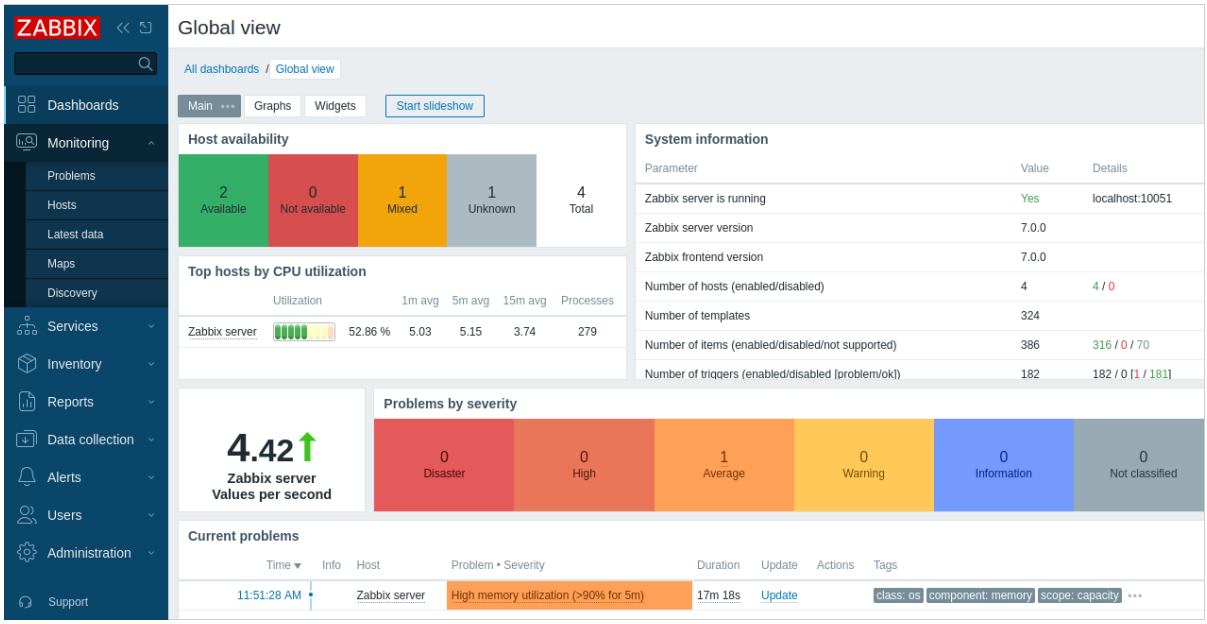
キオスクモードを終了するには、コンテンツ上にマウスカーソルを移動し、右上に表示される 通常表示ボタン をクリックします。

メニュー

概要

サイドバーのメニューから、さまざまな Zabbix フロントエンドセクションにアクセスできます。

デフォルトのテーマでは、メニューは紺色です。

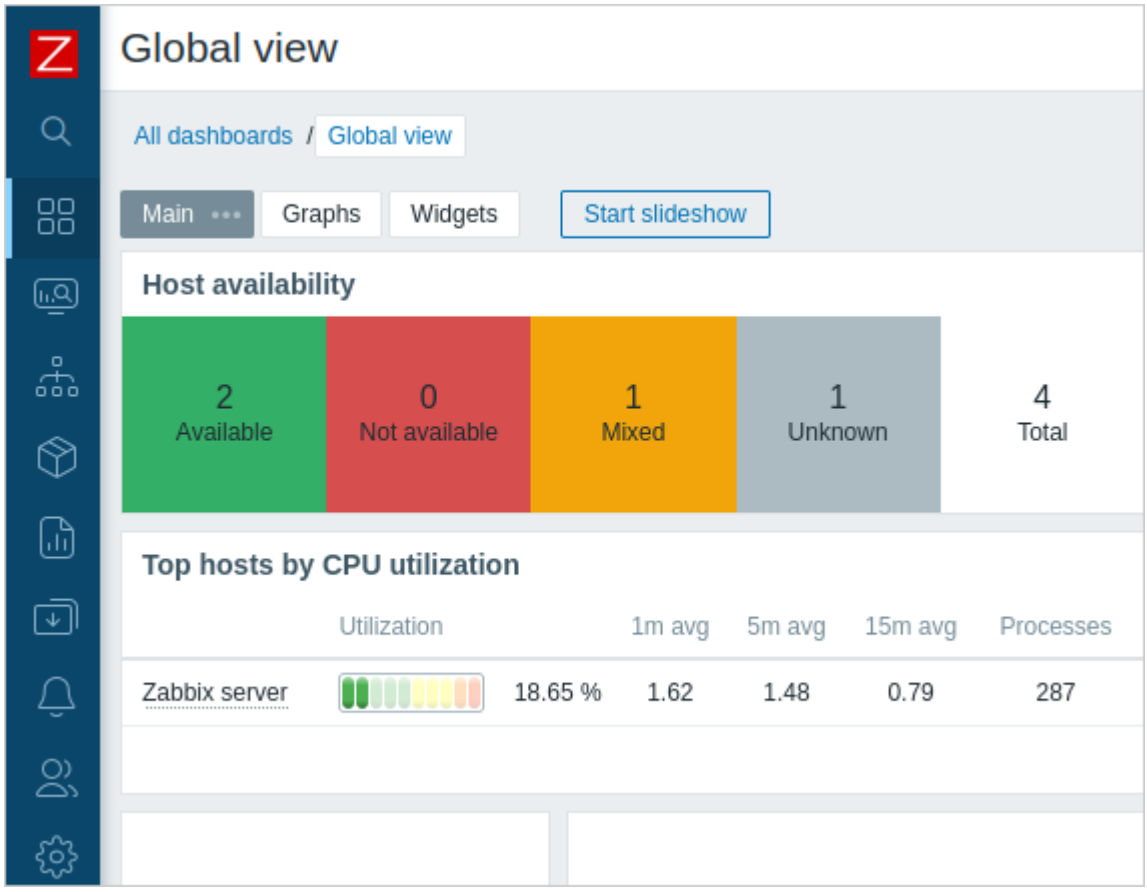


メニューの操作

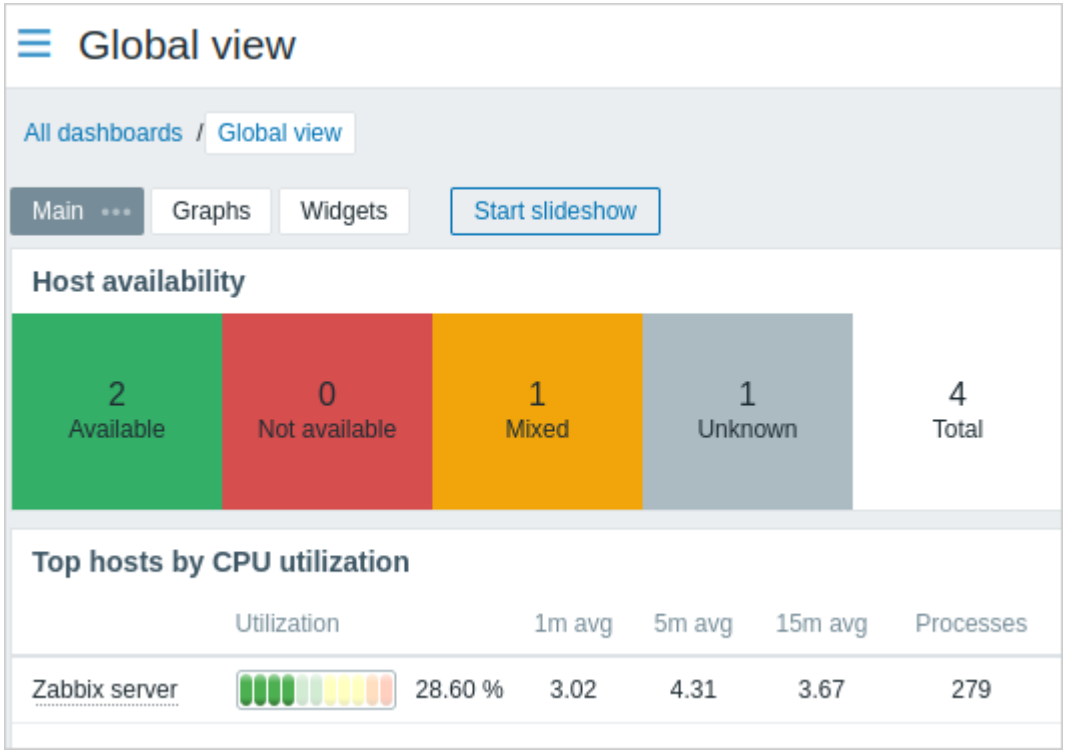
グローバル検索ボックスは Zabbix ロゴの下にあります。

メニューは折りたたむことも、完全に非表示にすることもできます。

- 折りたたむには、Zabbix ロゴの横にある  をクリックします。折りたたまれたメニューでは、アイコンのみが表示されます。



- 非表示にするには、Zabbix ロゴの横にある  をクリックします。非表示のメニューでは、すべてが隠れます。



折りたたまれたメニュー

メニューをアイコンのみの表示に折りたたむと、マウスカーソルをメニュー上に置いたときに、完全なメニューが再表示されます。完全なメニューはページのコンテンツ上に重なって表示されることに注意してください。ページのコンテンツを右に移動するには、展開ボタンをクリックする必要があります。マウスカーソルを再び完全なメニューの外に移動すると、メニューは 2 秒後に再度折りたたまれます。

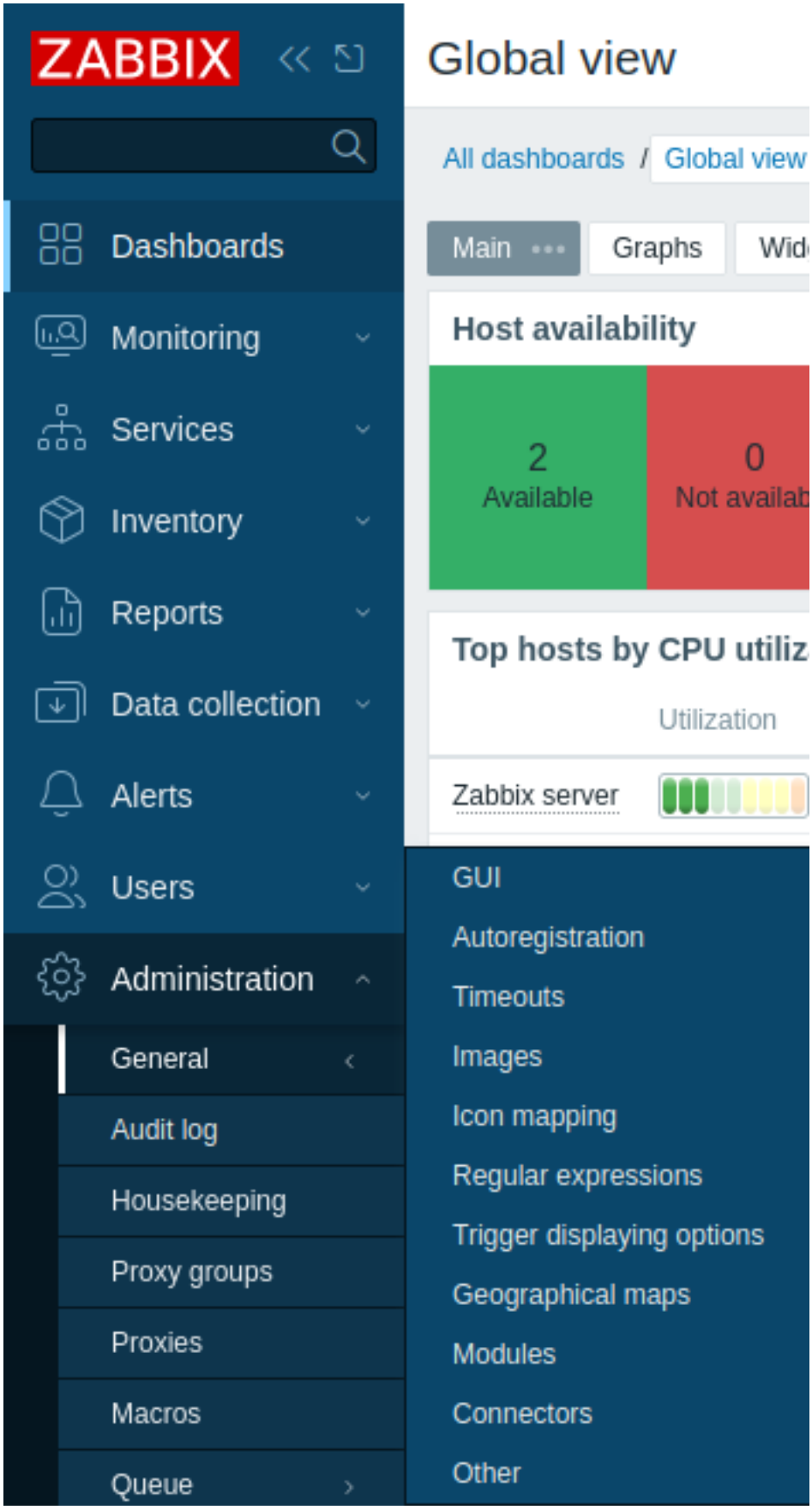
折りたたまれたメニューを完全に再表示するには、Tab キーを押すこともできます。Tab キーを繰り返し押すと、次のメニュー要素にフォーカスを移動できます。

非表示のメニュー

メニューが完全に非表示になっていても、バーガーアイコン をクリックするだけで、フルメニューをすぐに表示できます。
なお、メニューはページコンテンツの上に重なって再表示されます。
ページコンテンツを右に移動するには、サイドバーを表示ボタンをクリックしてメニューを再表示する必要があります。

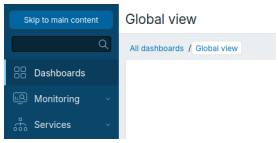
メニューの階層

メニューには最大 3 つの階層があります。



アクセシビリティ: ナビゲーションをスキップ

メインコンテンツへスキップリンクを使用すると、キーボードおよびスクリーンリーダーのユーザーは、繰り返し表示されるサイドバーのナビゲーションをスキップして、メインコンテンツへ直接移動できます。



通常の使用時には非表示のままで、ページの読み込み後に Tab キーが押されたときや、スクリーンリーダーを使用してアクセスしたときなど、フォーカスが当たった場合にのみ表示されます。

コンテキストメニュー

Zabbix では、メインメニューに加えて、ホスト、アイテム、およびevent のコンテキストメニューが用意されており、最新値、シンプルグラフ、設定フォーム、関連スクリプト、外部リンクなど、よく使用するエンティティへすばやくアクセスできます。

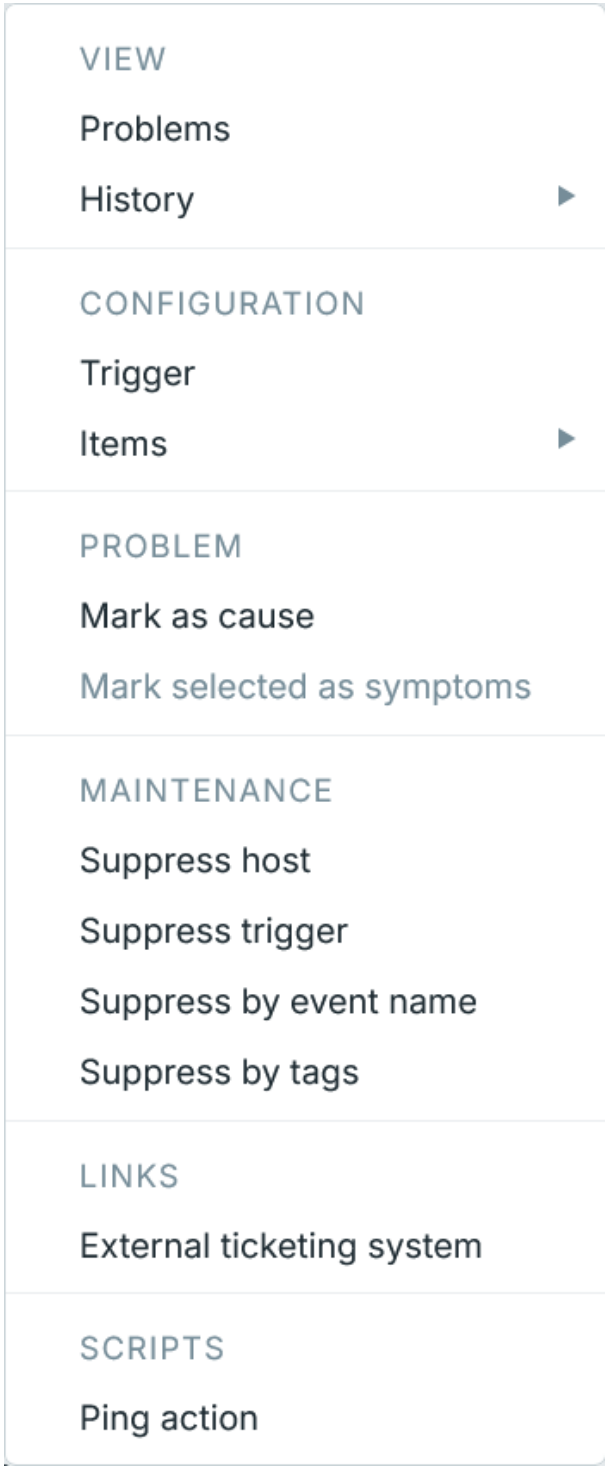
コンテキストメニューは、対応している場所でホスト、アイテム、または問題/トリガー名をクリックすると表示されます。

イベントメニュー

概要

イベントメニューには、イベントで頻繁に必要となるアクションまたは Web インターフェースのセクションへのショートカットが含まれています。

イベントメニューは、イベント名をクリックすると開くことができます。



内容

イベントのコンテキストメニューには、表示、アクション、設定、障害、メンテナンス、リンク、スクリプトのセクションがあります。設定されていないエンティティについては、リンクが無効になり、グレーで表示されます。スクリプトおよびリンクのセクションは、それぞれのエンティティが設定されている場合に表示されます。

表示セクションには、次のリンクが含まれます。

- 障害 - 基になるトリガーの未解決の障害一覧を開きます。
- 履歴 - 基になるアイテムの最新データグラフまたはアイテム履歴を表示します。トリガーが複数のアイテムを使用している場合は、それぞれのアイテムへのリンクが利用できます。

アクションセクションは、トリガーの概要ウィジェットでのみ利用できます。次のリンクが含まれます。

- 障害を更新 - 障害の更新画面を開きます。

設定セクションには、次の設定へのリンクが含まれます。

- 障害を発生させたトリガー;
- トリガーの式で使用されているアイテム。

Note:
設定セクションは、Admin ユーザーおよび Super admin ユーザーのみ利用できることに注意してください。

障害セクションには、次のオプションがあります。

- 原因としてマーク - 障害を原因としてマークします。
- 選択した障害を症状としてマーク - 選択した障害を、この障害の症状としてマークします。

メンテナンスセクションには、アドホックメンテナンスを開始するための次のオプションがあります。

- ホストを抑制 - ホストのメンテナンス期間を開始します。
- トリガーを抑制 - トリガーのメンテナンス期間を開始します。
- イベント名で抑制 - イベント名のメンテナンス期間を開始します。
- タグで抑制 - 対応するタグを持つイベントのメンテナンス期間を開始します。

該当するオプションを選択すると、メンテナンス期間の設定フォームが開きます。

現在から開始する 1 時間の 1 回限りのメンテナンス期間（ユーザーの設定およびプロファイルで設定可能）があらかじめ選択されます。該当するホスト、トリガー、イベント名、タグもあらかじめ選択されます。メンテナンス名には、`Ad-hoc: <event name>` の形式であらかじめ入力されます。

リンクセクションには、次のリンクが含まれます。

- 設定されたトリガー URLにアクセスするリンク;
- グローバルスクリプトで設定されたカスタムリンクにアクセスするリンク（スコープが「手動イベントアクション」で、タイプが「URL」のもの）;
- 障害に設定された外部チケットにアクセスするリンク（webhookの設定時にイベントメニューエントリを含めるオプションを参照）。

スクリプトセクションには、グローバルスクリプトを実行するためのリンクが含まれます（スコープは手動イベントアクション）。この機能は、外部システムの障害チケットを管理するために使用するスクリプトの実行に便利です。

サポートされる場所

イベントのコンテキストメニューは、さまざまな Web インターフェースのセクションで障害またはイベント名をクリックすることで表示できます。たとえば、次の場所です。

- ダッシュボードウィジェット、たとえば *Problems* ウィジェット、*Trigger overview* ウィジェットなど
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > イベント詳細
- *Reports* > *Top 100 triggers*（この場所ではグローバルスクリプトと外部チケットへのアクセスはサポートされていません）

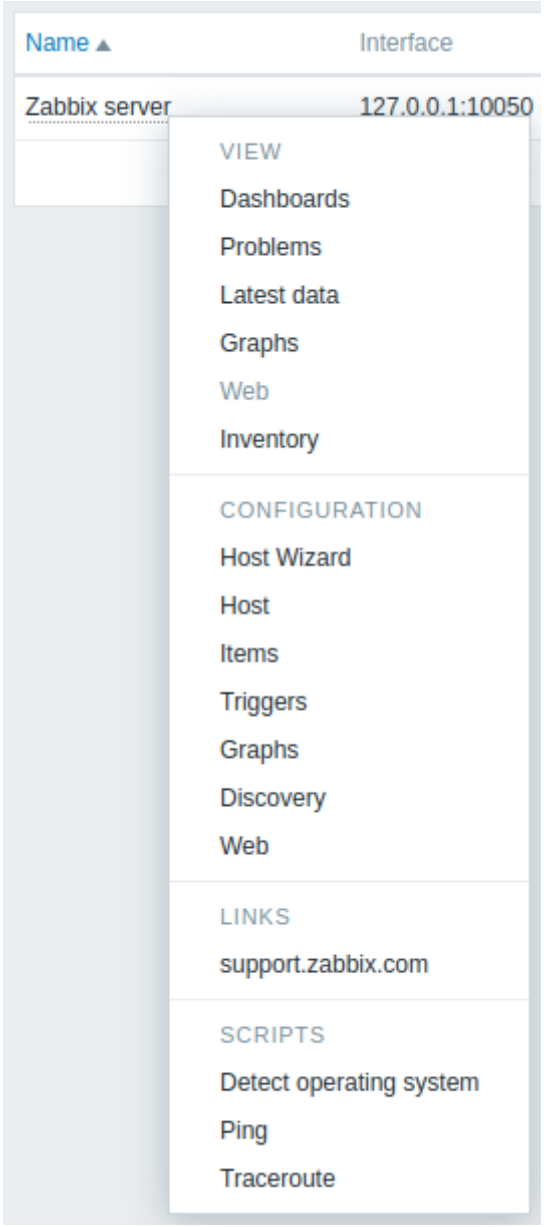
ホストメニュー

概要

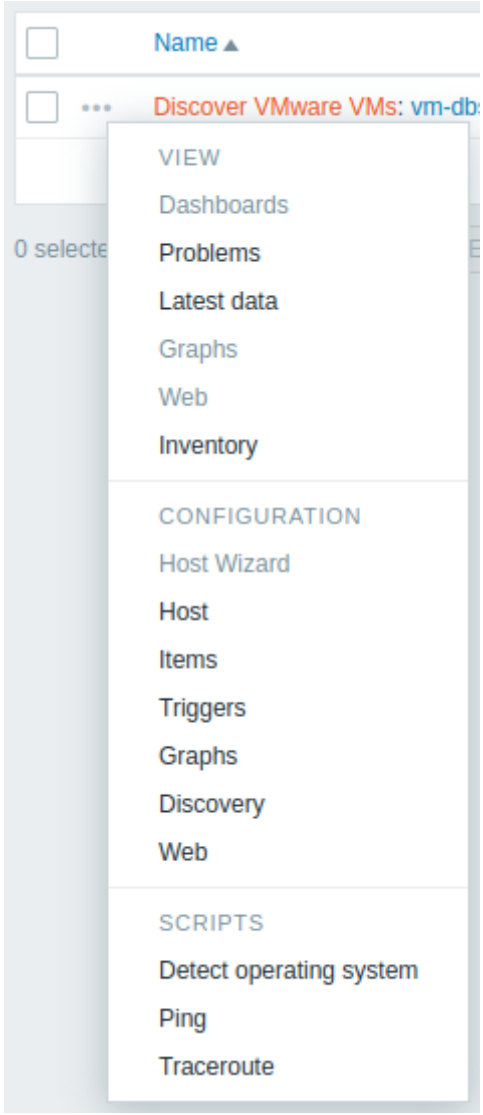
ホストメニューには、ホストに対して頻繁に必要となる操作や Web インターフェースのセクションへのショートカットが含まれています。

ホストメニューは、Web インターフェースのセクションに応じて、ホスト名または三点アイコンをクリックすると開くことができます。例:

Monitoring > *Latest data*



Data collection > *Hosts*



コンテンツ

ホストのコンテキストメニューには、表示、設定、リンク、スクリプトの 4 つのセクションがあります。設定されていないエンティティへのリンクは無効になり、グレーで表示されます。エンティティが設定されている場合、スクリプトおよびリンクセクションが表示されます。

表示セクションには、次のリンクが含まれます。

- ダッシュボード - ウィジェットとグラフを開きます。
- 障害 - 基になるトリガーの未解決の障害一覧が表示された障害セクションを開きます。
- 最新データ - 現在のホストの最新データ一覧が表示された最新データセクションを開きます。
- グラフ - 現在のホストのシンプルグラフを開きます。
- **Web** - 設定された Web シナリオへのリンクを開きます。
- インベントリ - 現在のホストのインベントリへのリンクを開きます。

設定セクションには、次のリンクが含まれます。

- ホストウィザード - 現在のホストのホストウィザードを開きます（ディスカバリされたホストでは無効）。
- ホスト - 現在のホストの設定フォームを開きます。
- アイテム - 現在のホストのアイテム一覧を開きます。
- トリガー - 現在のホストのトリガー一覧を開きます。
- グラフ - 現在のホストのシンプルグラフを開きます。
- ディスカバリ - 現在のホストのローレベルディスカバリルール一覧を開きます。

- **Web** - 現在のホストの Web シナリオ一覧を開きます。

> **Note:**
> 設定セクションは、管理者および特権管理者ユーザーのみ使用できます。

リンクセクションには、次のリンクが含まれます。

- 設定されたトリガー URLにアクセスします。
- グローバルスクリプトで設定されたカスタムリンクにアクセスします（スコープが手動ホストアクションで、タイプが「URL」のもの）。

スクリプトセクションでは、現在のホストに設定されたグローバルスクリプトを実行できます。これらのスクリプトをホストメニューで使用できるようにするには、スコープを手動ホストアクションとして定義する必要があります。

対応する場所

ホストメニューは、Web インターフェースのさまざまなセクションでホスト名をクリックすると表示されます。例:

- ダッシュボードウィジェット の *Problems*、*Top items*、*Trigger overview* など
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > イベント詳細
- *Monitoring* > *Hosts*
- *Monitoring* > *Hosts* > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

ホストメニューは、Data collection > Hosts で三点リーダーのアイコンをクリックしても表示されます。

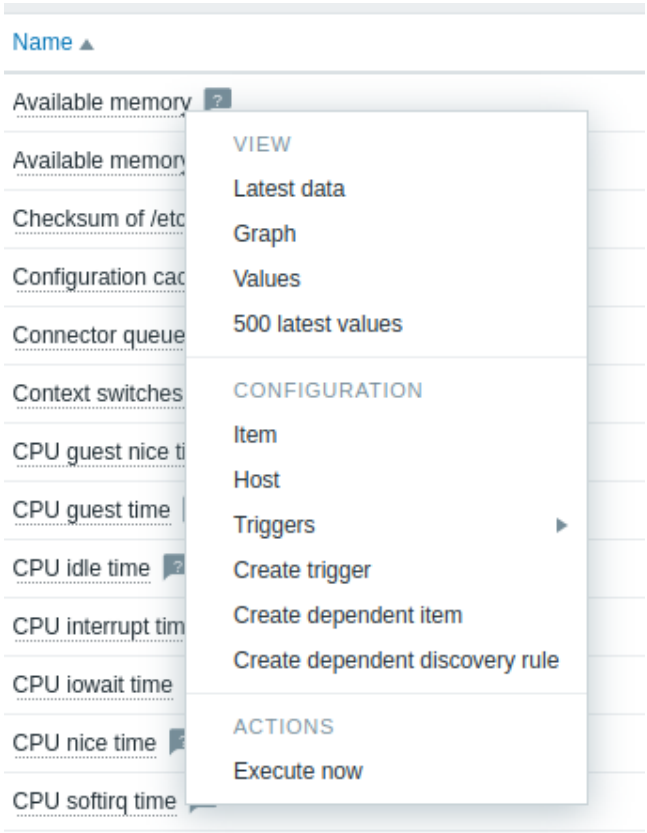
アイテムメニュー

概要

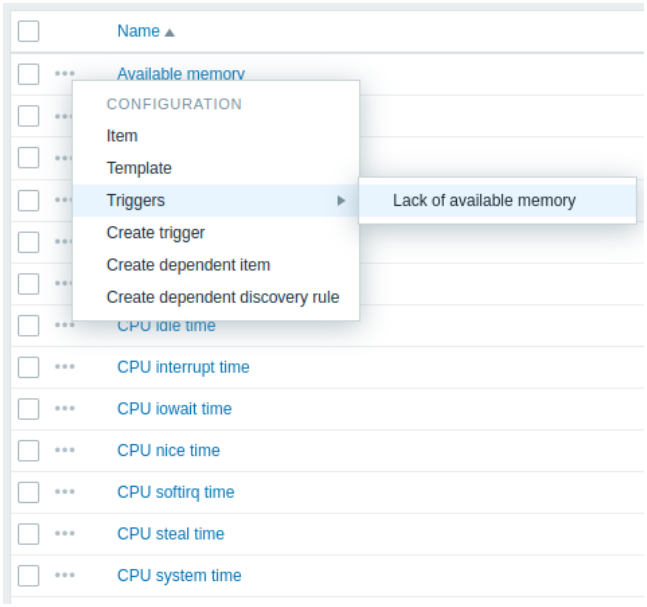
アイテムメニューには、アイテムに対して頻繁に必要となる操作や Web インターフェースのセクションへのショートカットが含まれています。

アイテムメニューは、Web インターフェースのセクションによって、アイテム名または三点アイコンをクリックすることで開くことができます。例:

Monitoring > *Latest data*



Data Collection > *Templates* > *Items*



内容

アイテムメニューには、表示、設定、アクションの 3 つのセクションがあります。

表示セクションには、次のオプションがあります。

- 最新データ - 現在のホストとアイテムでフィルターされた最新データ セクションを開きます。
- グラフ - 現在のアイテムのシンプルグラフ を開きます。
- 値 - 過去 60 分以内に現在のアイテムで受信したすべての値の一覧を開きます。
- 最新 **500** 個の値 - 現在のアイテムの最新 500 個の値の一覧を開きます。

設定セクション（管理者および 特権管理者タイプのユーザーのみ利用可能）には、次のオプションがあります。

- アイテム - 現在のアイテムのアイテム設定フォームを開きます。
- テンプレート - アイテムが所属するテンプレートのテンプレート設定フォームを開きます（アイテムメニューに データ収集 > テンプレート > アイテム からアクセスした場合のみ利用可能）。
- ホスト - アイテムが所属するホストのホスト設定フォームを開きます。
- トリガー - マウスオーバーすると、アイテムのトリガーがある場合にその一覧を開きます。トリガーをクリックすると、そのトリガー設定フォームが開きます。
- トリガーの作成 - このアイテム用のトリガーを作成するためのトリガー設定フォームを開きます。
- 依存アイテムの作成 - 現在のアイテムをマスターアイテムとして、依存アイテムを作成するためのアイテム設定フォームを開きます。
- 依存ディスカバリルールの作成 - 現在のアイテムをマスターアイテムとして、依存ディスカバリルールを作成するためのディスカバリルール設定フォームを開きます。

アクションセクションには、次のオプションがあります。

- 今すぐ実行 - 新しいアイテムの値を取得するためのチェックを実行します。

対応場所

アイテムメニューは、Web インターフェースのさまざまなセクションでアイテム名をクリックすると表示されます。例:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items*
- *Data collection* > *Hosts* > *Discovery rules* > *Item prototypes*

アイテムメニューは、ダッシュボードウィジェット*Top items* でアイテムの値をクリックしても表示されます。

Web インターフェースのセクション

メニュー構成　Zabbix の Web インターフェースメニューは、次の構成になっています。

- ダッシュボード
- 監視
  - 障害
  - ホスト
  - 最新データ
  - マップ
  - ディスカバリ
- APM
  - トレース
  - メトリクス
  - ログ
- サービス
  - サービス
  - SLA
  - SLA レポート
- インベントリ
  - 概要
  - ホスト
- レポート
  - システム情報
  - スケジュールレポート
  - 可用性レポート
  - トリガー上位 100
  - 監査ログ
  - アクションログ
  - 通知
- データ収集
  - テンプレートグループ
  - ホストグループ
  - テンプレート
  - ホスト
  - メンテナンス
  - イベント相関
  - ディスカバリ
- 通知
  - アクション
    - トリガーアクション
    - サービスアクション
    - ディスカバリアクション
    - 自動登録アクション
    - 内部アクション
  - メディアタイプ
  - スクリプト
- ユーザー
  - ユーザーグループ
  - ユーザーロール
  - ユーザー
  - API トークン
  - 認証
  - デバイス
- 管理
  - 一般設定
    - GUI
    - 自動登録
    - タイムアウト
    - イメージ
    - アイコンマッピング

        * 正規表現
        * トリガー表示オプション
        * 地理マップ
        * モジュール
        * コネクター
        * その他
    - 監査ログ
    - ハウスキーピング
    - プロキシグループ
    - プロキシ
    - マクロ
    - キュー
        * キューの概要
        * プロキシ別のキューの概要
        * キューの詳細

ダッシュボード

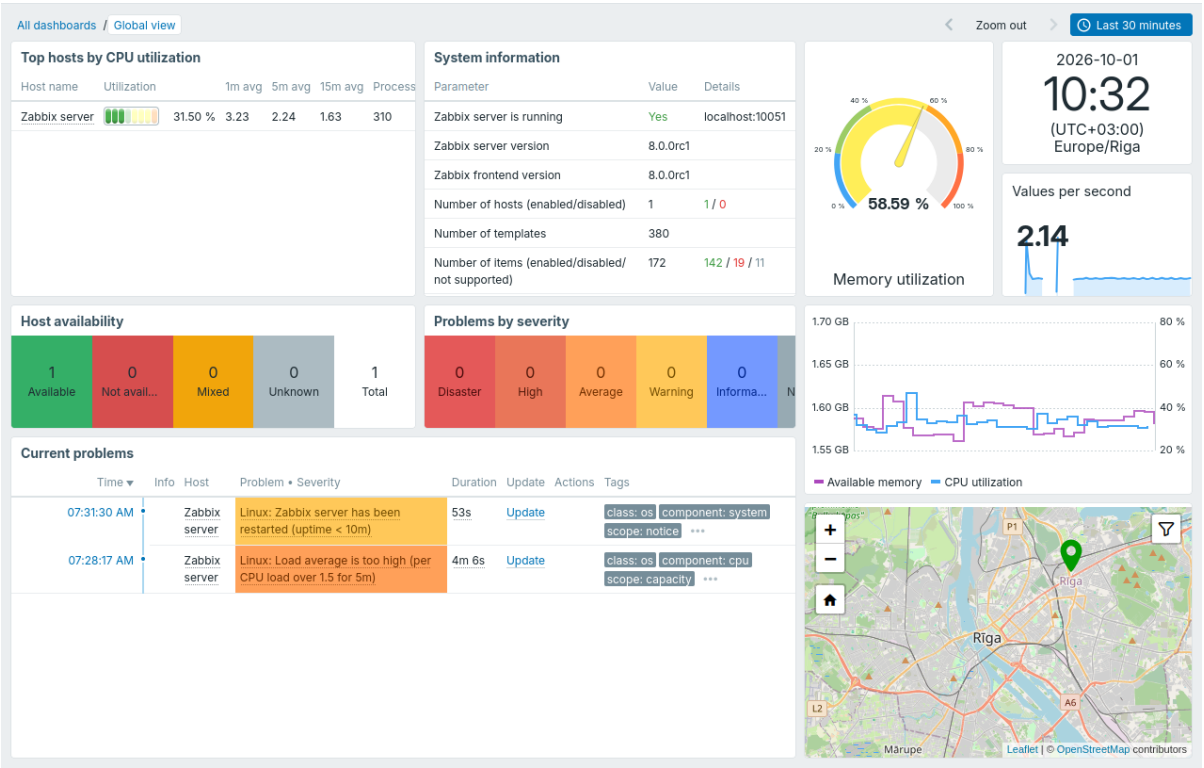
ダッシュボードでは、監視環境の概要を確認できます。要件に合わせて作成およびカスタマイズできます。

構成要素はウィジェットです。グラフ、マップ、最近の障害の一覧など、それぞれ特定の種類の情報を表示するように設計されています。

ウィジェットは、複数のダッシュボードページに整理できます。このようなページはスライドショーとして表示できます。また、タブをクリックして任意のページに移動することもできます。

複数のダッシュボードを作成できますが、一度に表示できるのは 1 つだけです。

Zabbix には、いくつかのデフォルトダッシュボードが用意されています。その 1 つが *Global view* ダッシュボードです。このダッシュボードには、Zabbix サーバーを含む、監視対象のホストの正常性とステータスが表示されます。

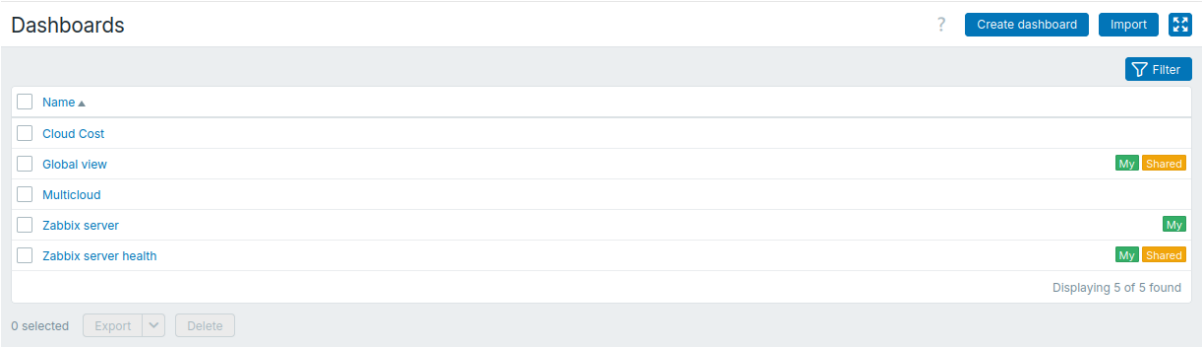


Dashboards repositoryには、さらに多くのダッシュボードがあります。これらのダッシュボードは、既存の Zabbix テンプレートで動作するように設計されており、ダウンロードして Zabbix 環境にインポートできます。前提条件を含む各ダッシュボードの詳細な説明は、ダッシュボードの `README` ファイルに記載されています。

ダッシュボード一覧

ダッシュボードの一覧を開くには、次のいずれかを行います。

- メニューでダッシュボードをクリックします。
- ダッシュボードの表示中にすべてのダッシュボードをクリックして、ダッシュボード一覧を開きます。



ダッシュボードメニューへのアクセスは、ユーザーロールと、*UI* 要素へのアクセスの下にあるダッシュボードの権限によって異なります。

ここから、次の操作を実行できます。- ダッシュボードを作成します。- ダッシュボードを削除します。チェックボックスを選択して削除ボタンをクリックすると、ダッシュボードを削除できます。- ダッシュボードをインポートします。- ダッシュボードをエクスポートします。

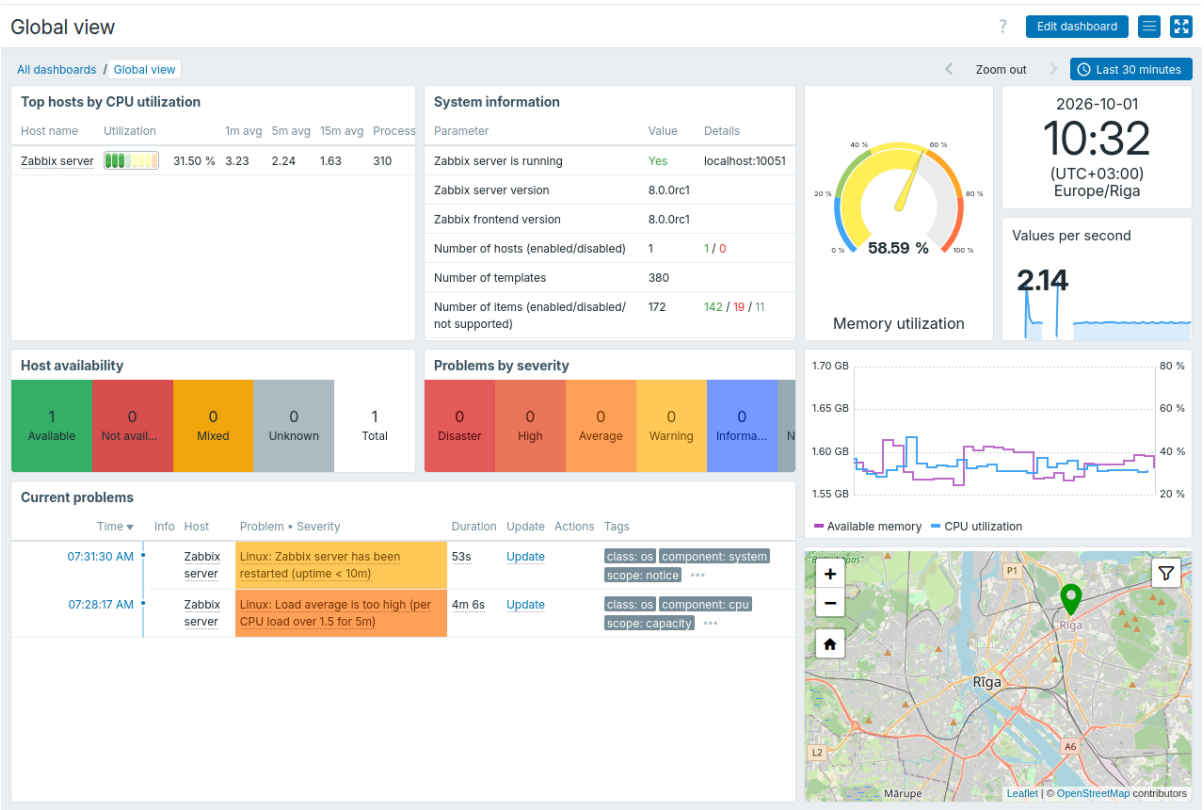
ダッシュボードには、所有権と共有設定を示すタグがあります。

- 自分のダッシュボード - ダッシュボードは自分が所有しており、自分だけが表示できます。
- 自分のダッシュボード + 共有 - ダッシュボードは自分が所有していますが、公開されているか、特定のユーザーまたはユーザーグループと共有されています。
- タグのないダッシュボードは、他のユーザーが所有していますが、公開されているか、自分と共有されています。

ダッシュボード名と所有者で一覧をフィルターできます。

ダッシュボードの表示

ダッシュボード一覧でダッシュボード名をクリックすると、ダッシュボードが表示されます。



Zabbix ユーザーインターフェースは、次回のセッションで最後に開いたダッシュボードを記憶します。

Note:
ダッシュボードに表示できるデータは、ホストへのアクセス権によって異なります。

グラフを表示するウィジェットでは、ウィジェットの上にある期間セレクターで時間範囲を設定します。現在の時間範囲がラベルとして表示されます。ラベルをクリックすると、セレクターを開閉できます。

ここから、次のようにダッシュボードを管理できます。

- ダッシュボードを編集します。
- ダッシュボードのアクションメニューから、次の操作を実行します。
  - 共有 - ダッシュボードの共有設定を編集します。
  - 新規作成 - 新しいダッシュボードを作成します。
  - 複製 - 現在のダッシュボードを複製して、新しいダッシュボードを作成します。保存する前に、ダッシュボードのプロパティを編集できます。
  - 削除 - ダッシュボードを削除します。
  - エクスポート - ダッシュボードをエクスポートします。
  - 新しいレポートを作成 - スケジュールレポートを作成します。スケジュールレポートを管理する権限がない場合、このオプションは無効になります。
  - 関連レポートを表示 - 現在のダッシュボードに関連する既存のレポート一覧を表示します。関連するレポートがない場合、またはスケジュールレポートを管理する権限がない場合、このオプションは無効になります。
- キオスクモードに切り替えます。
- ダッシュボードのスライドショーを開始または停止します。

ダッシュボードの表示モードでは、ウィジェットに対して次のオプションを使用できます。

- ウィジェットを編集します。
- ウィジェットのアクションメニュー から、次の操作を実行します。
  - コピー - ウィジェットをコピーします。
  - 画像をダウンロード - ウィジェットを PNG 画像としてダウンロードします。このオプションは、この機能をサポートするウィジェットで使用できます（各ウィジェットのページを参照してください）。
  - 更新間隔 - ウィジェットの内容を更新する頻度を選択します。

複数のページがあるダッシュボードでは、ダッシュボードの表示モードでページのアクションメニュー から、次の操作を実行できます。

- コピー - ページをコピーします。
- プロパティ - 名前やスライドショーの間隔など、ページのプロパティをカスタマイズします。

スライドショー

ダッシュボードのプロパティでスライドショーを開始チェックボックスを選択すると、ダッシュボードに複数のページがある場合、スライドショーが自動的に開始されます。それ以外の場合は、スライドショーを開始ボタンをクリックしてスライドショーを開始できます。



ページは、ダッシュボードおよび個々のページのプロパティで指定された間隔に従って切り替わります。個々のページにスライドショーの間隔プロパティが定義されている場合、そのページに指定された間隔が適用されます。それ以外の場合は、ダッシュボードのデフォルトのスライドショーの間隔が適用されます。

スライドショーを停止するには、スライドショーを停止をクリックします。

キオスクモードでは、次のボタンを使用します。

-  スライドショーを開始します。
-  スライドショーを停止します。
-  前のページに移動します。
-  次のページに移動します。

ダッシュボードの共有

ダッシュボードはパブリックまたはプライベートに設定できます。

パブリックダッシュボードはすべてのユーザーに表示されます。プライベートダッシュボードは、所有者と、共有対象として指定されたユーザーまたはユーザーグループにのみ表示されます。

ダッシュボードの共有設定を変更するには、ダッシュボードを表示している状態で、ダッシュボードのアクションメニューから共有をクリックします。

パラメーター	説明
タイプ	ダッシュボードのタイプを選択します。 プライベート - ダッシュボードは、選択したユーザーグループとユーザーにのみ表示されます。 パブリック - ダッシュボードはすべてのユーザーに表示されます。

パラメーター	説明
ユーザーグループ共有の一覧	ダッシュボードにアクセスできるユーザーグループを選択します。 読み取り専用または読み取り/書き込みアクセスを付与できます。
ユーザー共有の一覧	ダッシュボードにアクセスできるユーザーを選択します。 読み取り専用または読み取り/書き込みアクセスを付与できます。

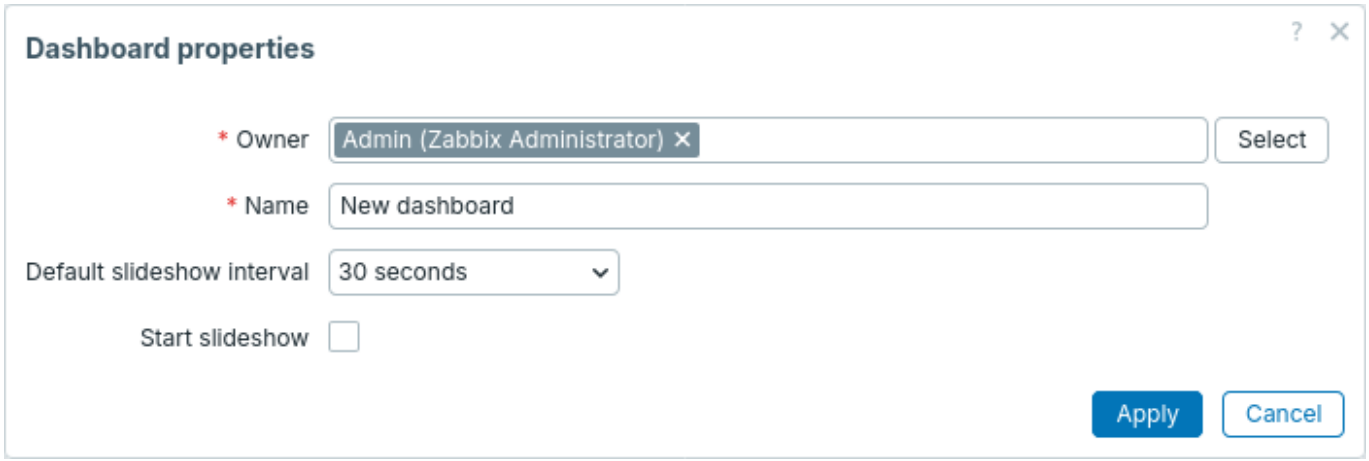
共有オプションによってユーザーが実行できるアクションがどのように制限されるかについては、ダッシュボードの権限を参照してください。

ダッシュボードの作成

ダッシュボードは、次の 2 つの方法で作成できます。

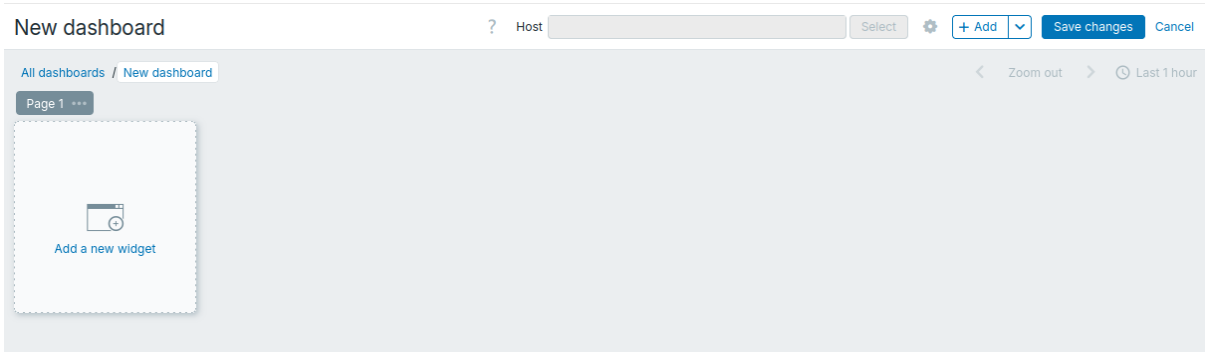
- ダッシュボード一覧で、ダッシュボードの作成ボタンをクリックします。
- ダッシュボードの表示モードで、ダッシュボードのアクションメニューから新規作成をクリックします。

次に、ダッシュボードのプロパティを指定します。



パラメーター	説明
所有者	ダッシュボードの所有者となるユーザーを選択します。
名前	ダッシュボード名を入力します。
デフォルトのスライドショー間隔	スライドショーで次のページに移動するまで、各ダッシュボードページを表示する時間を選択します。
スライドショーを開始	ダッシュボードに複数のページがある場合に、スライドショーを自動的に開始するには、このチェックボッ

次に、適用をクリックします。新しいダッシュボードが編集モードで開きます。



ここから、ダッシュボードのウィジェットとページを追加できます。

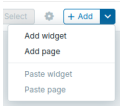
ダッシュボードの最小幅は 1200 ピクセルです。幅の狭いウィンドウでは、水平スクロールバーが表示されます。幅の広いダッシュボードは常にブラウザーウィンドウ全体に表示されます。ウィジェットをウィンドウの端より外側に配置することはできません。

変更を保存ボタンをクリックしてダッシュボードを保存するか、キャンセルをクリックして前の画面に戻ります。

ウィジェットの追加

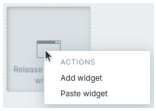
ダッシュボードの編集モードでは、ウィジェットを次の 2 つの方法で追加できます。

- 追加ボタンをクリックします。または、追加ボタンのドロップダウン矢印をクリックし、ウィジェットの追加を選択します。



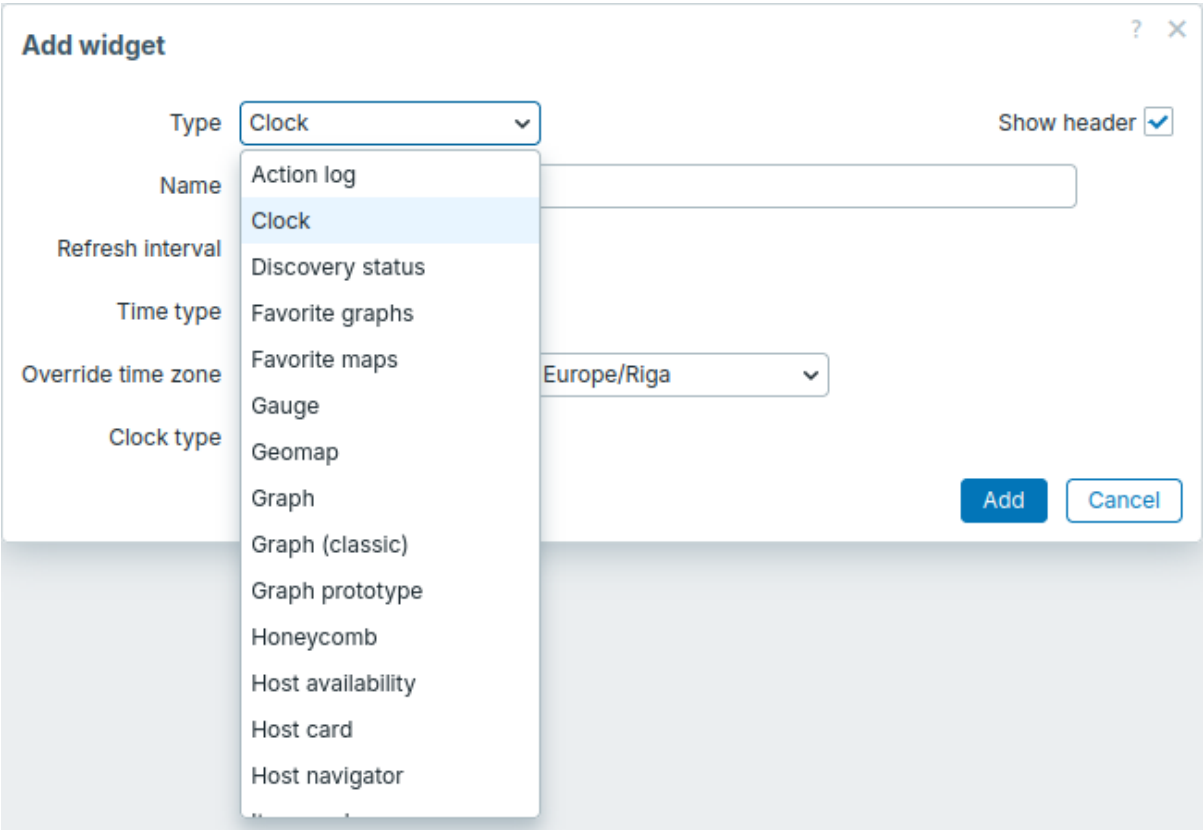
ウィジェットの設定フォームが開き、ウィジェットのプレースホルダーがデフォルトサイズで、ダッシュボード上の最初に利用可能な空き領域に追加されます。

- ダッシュボードの空いている領域にマウスポインターを移動します。次に、ウィジェットのプレースホルダーをクリックして、ウィジェットの設定フォームを開きます。または、プレースホルダーをクリックしてドラッグし、サイズを変更してからマウスボタンを離すと、ウィジェットの設定フォームが開きます。ウィジェットがクリップボードにコピーされている場合は、最初にウィジェットの追加またはウィジェットの貼り付けを選択するよう求められます。



次の手順を実行します。

1. ウィジェットの設定フォームで、ウィジェットのタイプを選択します。
2. ウィジェットのパラメータを設定します。
3. 追加をクリックして、ウィジェットをダッシュボードに追加します。



ウィジェットはデフォルトサイズで追加されるか、利用可能なスペースに合わせてサイズが変更されます。ウィジェットの編集セクションの説明に従って、ウィジェットのサイズ変更や位置変更を行えます。

また、ウィジェットをコピーして貼り付けることで、同じ設定の新しいウィジェットを作成できます。同じダッシュボード内だけでなく、異なるブラウザータブで開いたダッシュボード間でも利用できます。

ウィジェットのコピーと貼り付け

ウィジェットをコピーして貼り付けるには、次の手順を実行します。

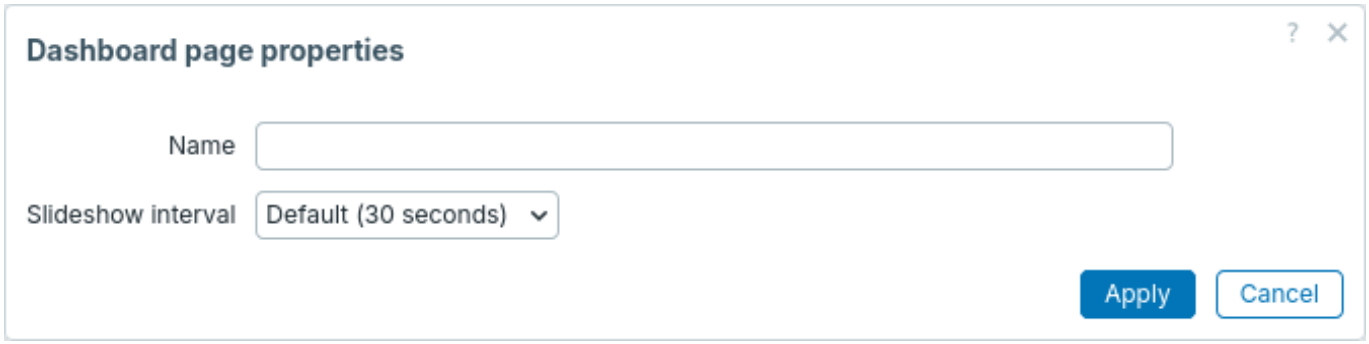
1. ウィジェットの アクションメニュー ⋮ で、コピーを選択します。
2. ダッシュボードの編集モードで、追加ボタンのドロップダウン矢印をクリックし、ウィジェットを貼り付けを選択します。
   または、ダッシュボード内の空いている領域をクリックし、ウィジェットを貼り付けを選択します。

既存のウィジェットの アクションメニューにある 貼り付けオプションを使用して、コピーしたウィジェットを既存のウィジェット上に貼り付けることもできます。

ページの追加

ダッシュボードに新しいページを追加するには、次の手順を実行します。

1. ダッシュボードの編集モードで、追加ボタンのドロップダウン矢印をクリックし、ページの追加を選択します。
2. ページのプロパティを入力します。
    - 名前 - ページの名前を入力します。名前を空のままにすると、ページ番号を表す「N」を使用して、ページ *N* という名前でページが追加されます。
    - スライドショーの間隔 - スライドショー が次のページに移動するまでページを表示する時間を選択します。



3. 適用をクリックします。

新しいページ用の新しいタブ（例: ページ *2*）が追加されます。



ページをコピーして貼り付けることで、同じプロパティを持つ新しいページを作成することもできます。同じダッシュボード内だけでなく、異なるタブで開いているダッシュボード間でも実行できます。

ページのコピー/貼り付け

ページをコピーして貼り付けるには、次の手順を実行します。

1. ページの アクションメニュー で、コピーを選択します。
2. ダッシュボードの編集モードで、追加ボタンのドロップダウン矢印をクリックし、ページの貼り付けを選択します。

ダッシュボードの編集

ダッシュボードを編集するには、右上隅にある ダッシュボードの編集ボタンをクリックします。または、ウィジェットの 編集ボタン をクリックします。ダッシュボードが編集モードで開きます。

ここから、次のようにダッシュボードを管理できます。

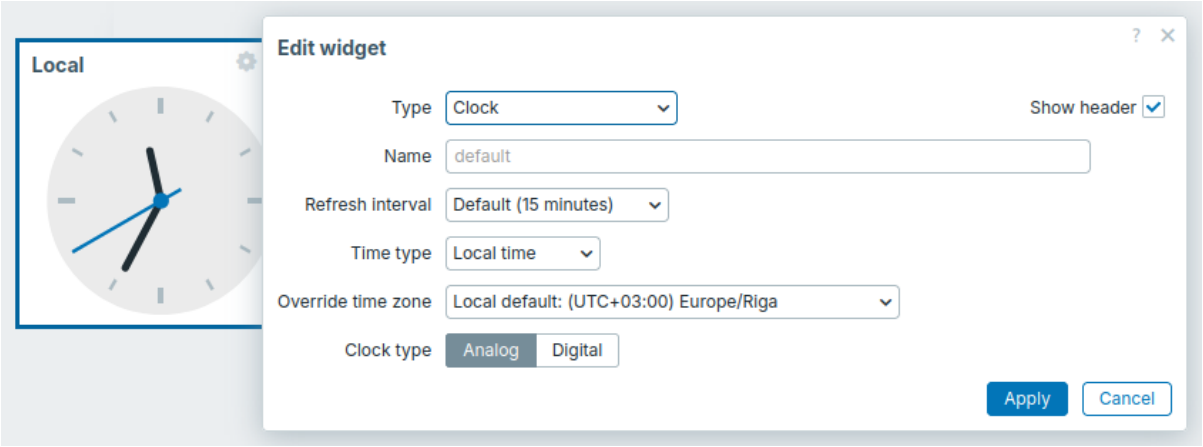
- 歯車ボタン をクリックして、ダッシュボードのプロパティを編集します。
- ダッシュボードウィジェットを編集します。
- 追加ボタンのドロップダウン矢印をクリックして、次のダッシュボード操作にアクセスします。
    - ウィジェットの追加 - ダッシュボードにウィジェットを追加します。
    - ページの追加 - ダッシュボードにページを追加します。
    - ウィジェットの貼り付け - コピーしたウィジェットを貼り付けます。ウィジェットがコピーされていない場合、このオプションはグレー表示されます。一度にコピーできるウィジェットは 1 つだけです。
    - ページの貼り付け - コピーしたページを貼り付けます。ページがコピーされていない場合、このオプションはグレー表示されます。一度にコピーできるページは 1 つだけです。

ダッシュボードの編集後、変更を保存ボタンをクリックしてダッシュボードを保存するか、キャンセルをクリックして前の画面に戻ります。

ウィジェットの編集

ダッシュボードの編集モードで、ウィジェットの 編集ボタン をクリックすると、ウィジェットを編集できます。または、ダッシュボードを表示しているときにウィジェットにカーソルを合わせ、編集ボタンをクリックします。

ウィジェットの編集中は強調表示され、変更内容がリアルタイムでプレビューされます。



タイトルバーをドラッグすると、ウィジェットとその設定フォームの位置を変更できます。また、サイズ変更ハンドルをドラッグすると、ウィジェットのサイズを変更できます。



ウィジェットの編集中は、ウィジェットの アクションメニュー ... から次の操作を実行できます。

- コピー - ウィジェットをコピーします。
- 貼り付け - コピーしたウィジェットをこのウィジェットに貼り付けます。ウィジェットがコピーされていない場合、このオプションはグレー表示されます。
- 削除 - ウィジェットを削除します。

必須のウィジェットパラメータが設定されていない場合、ウィジェットは ウィジェットが完全に設定されていません状態になります。

編集後、ウィジェット設定フォームで 適用をクリックし、ダッシュボードの右上にある 変更を保存をクリックして変更を適用します。

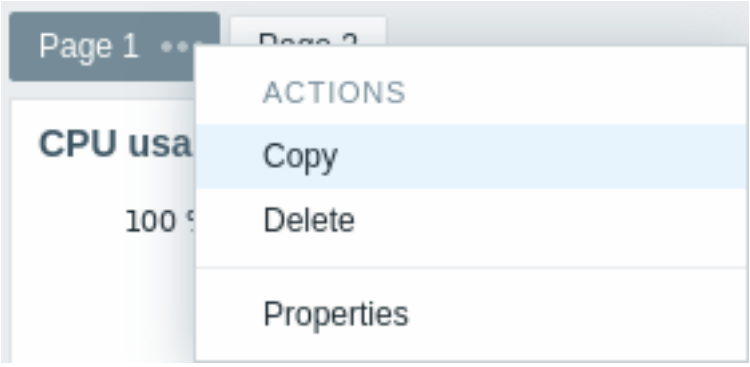
ページの編集

ページはダッシュボードの編集モードで編集できます。

ページタブをドラッグアンドドロップして、ページの順序を変更できます。順序を変更してもページ名は変更されません。ただし、自動的に割り当てられた名前（*Page N*）のページは、番号が振り直されます。

ページの編集中は、ページの *Actions* メニュー から次の操作を実行できます。

- *Copy* - ページをコピーします。
- *Delete* - ページを削除します。
- *Properties* - ページのプロパティをカスタマイズします。



ダッシュボードへの権限

一般ユーザーおよび *Admin* タイプのユーザーのダッシュボードへの権限は、割り当てられたユーザーロールでダッシュボードへのアクセスと ダッシュボードの作成と編集オプションが有効になっている場合、以下のように制限されます。

- ダッシュボードの共有設定にユーザーまたはユーザーグループが追加されていない場合でも、パブリックダッシュボードを表示および複製できます。
- 共有設定で指定されたとおり、少なくとも 読み取り権限がある場合、プライベートダッシュボードを表示および複製できます。
- 共有設定で指定されたとおり、読み書き権限がある場合にのみ、ダッシュボードを編集および削除できます。
- ダッシュボードの所有者を変更することはできません。

ダッシュボードウィジェット

概要

Zabbix は、監視データを可視化し要約するさまざまなダッシュボードウィジェットを提供しており、システムのパフォーマンスと可用性を明確に把握できます。

以下は、主な目的ごとにグループ化したすべてのウィジェットの概要です。

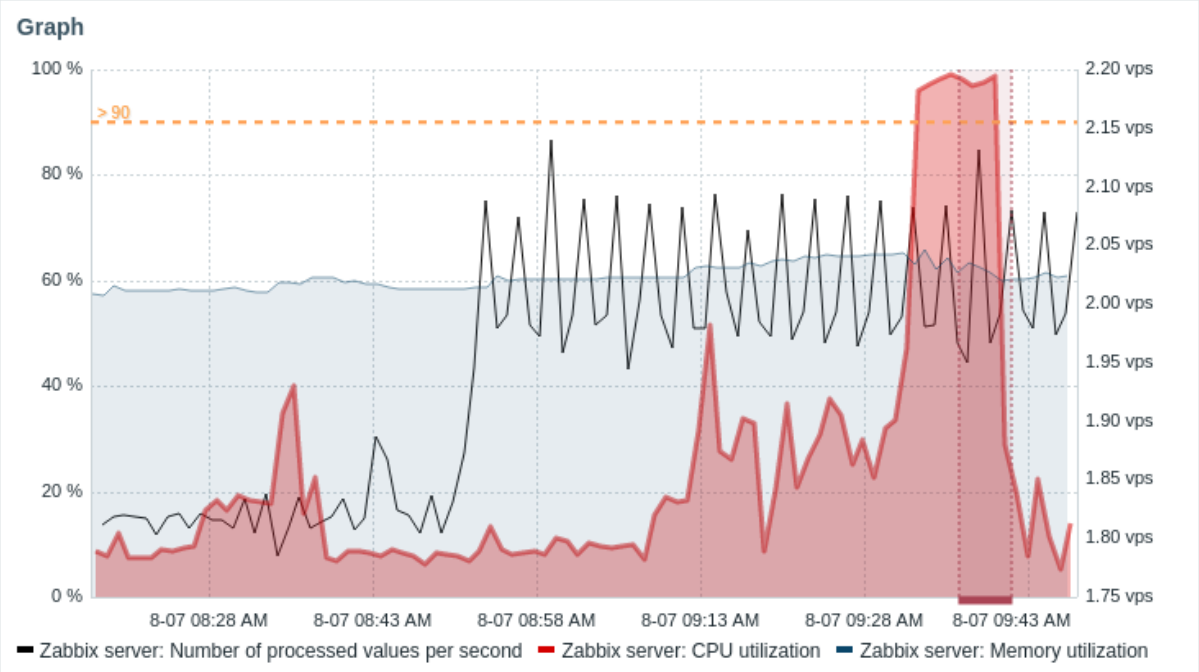
ウィジェットの設定の詳細については、ダッシュボードウィジェットのパラメータを参照してください。

データの可視化

これらのウィジェットは、さまざまな視覚的な形式でデータを表示し、さまざまな方法でデータや問題を監視するのに役立ちます。

グラフ

グラフウィジェットは、数値アイテムデータをベクターグラフとして表示します。これにより、メトリクスの追跡、問題の特定、時間やホスト間での値の比較が容易になります。



関連情報:

- グラフ（クラシック） - 数値アイテムデータを画像ベースのグラフとして表示する従来のグラフウィジェットです。
- グラフプロトタイプ - LLD ルールのグラフプロトタイプやアイテムプロトタイプに基づいて自動作成されたカスタムグラフのグリッドを表示します。
- お気に入りグラフ - 現在のユーザーがお気に入りとしてマークしたグラフへのショートカットのアルファベット順リストを表示します。

散布図

散布図ウィジェットは、X 軸と Y 軸に沿って個々のデータポイントをプロットすることで、2 つのメトリクス間の関係を表示します。これにより、データセット内のパターン、クラスター、相関関係、外れ値を明らかにするのに役立ちます。

円グラフ

円グラフウィジェットは、数値アイテムデータをベクターベースの円グラフまたはドーナツグラフとして表示します。
 このウィジェットは、アイテムやホストが全体のデータセットにどのように貢献しているかを視覚化するのに役立ちます。

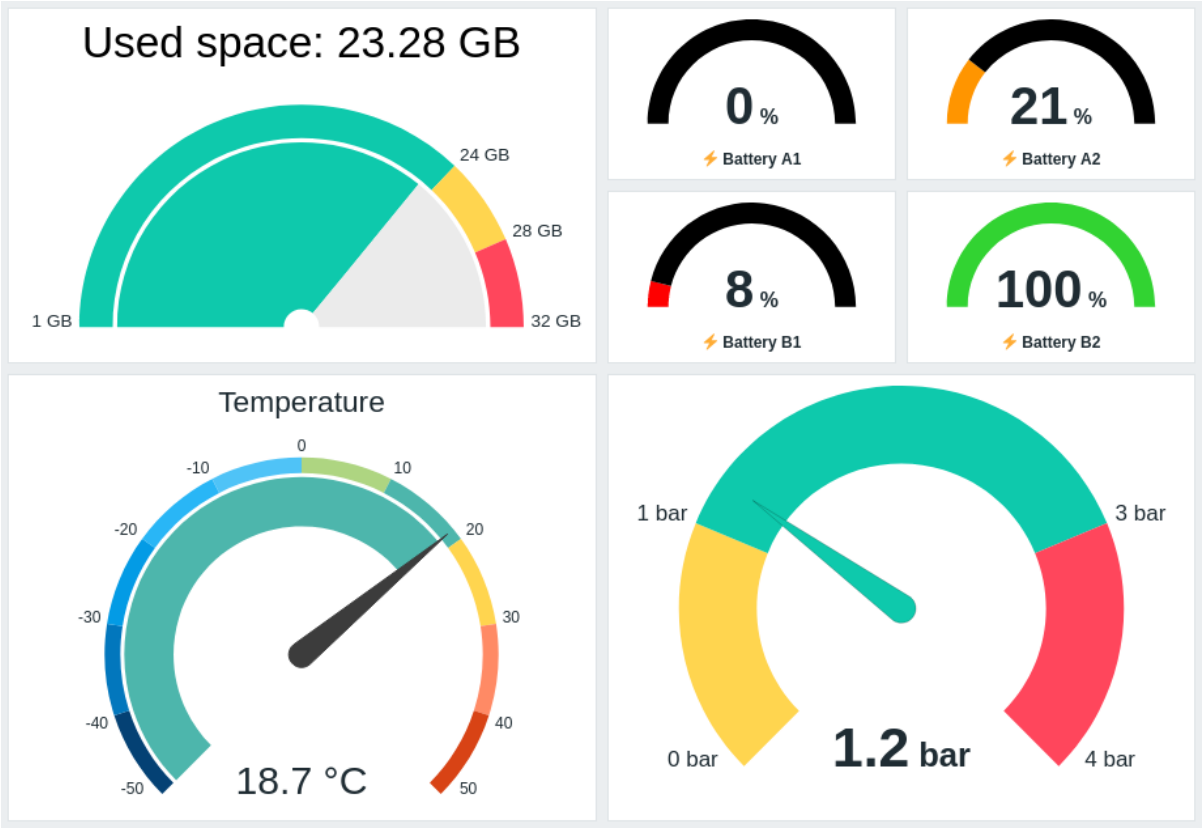
—

—

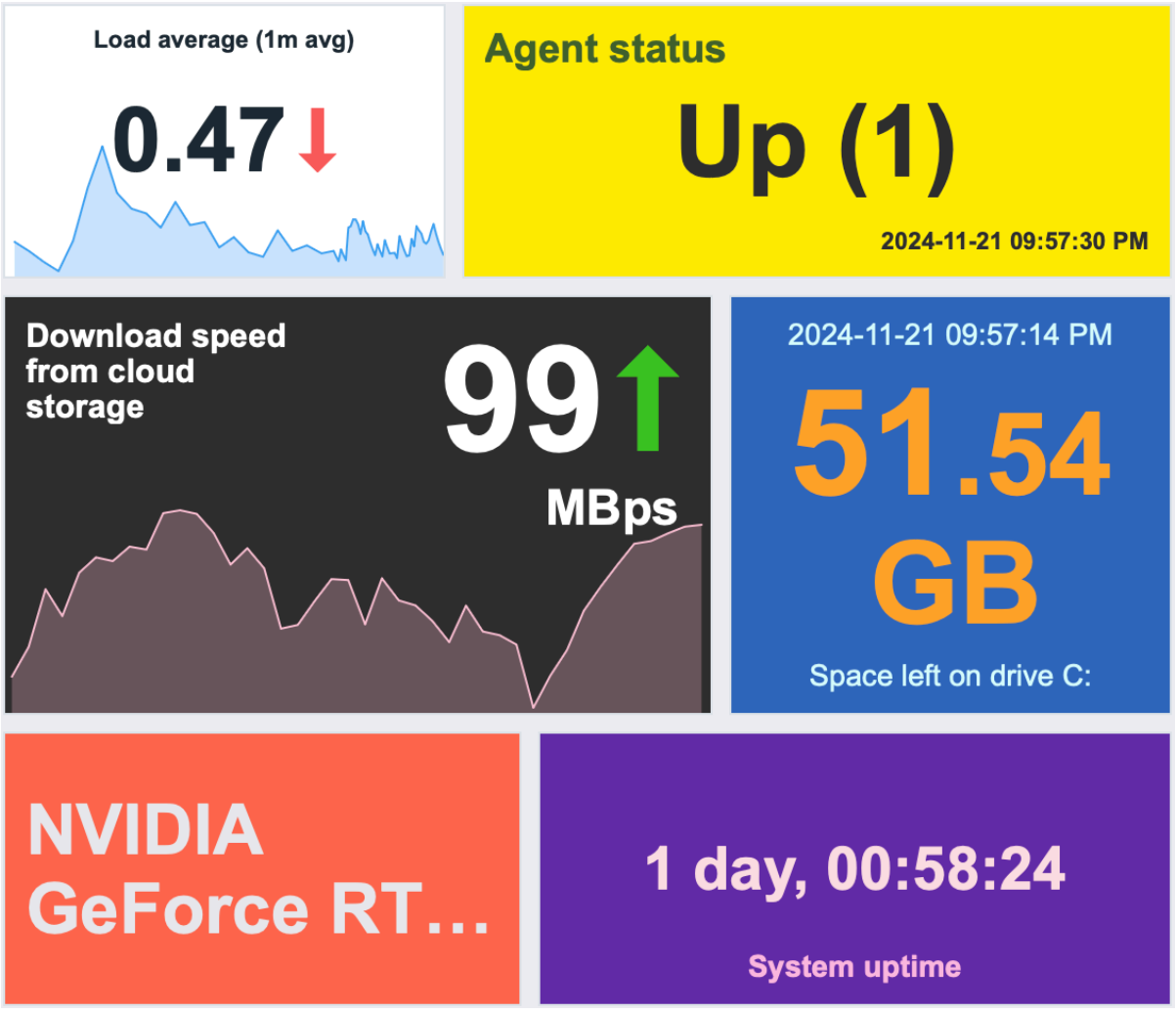
ゲージ/アイテム値

ゲージ/アイテム値ウィジェットは、単一のアイテムの値を表示します。これは、重要なメトリクスを監視したり、しきい値を可視化したり、データの急激な変化を検出したりするのに便利です。

ゲージウィジェット:



アイテム値ウィジェット:

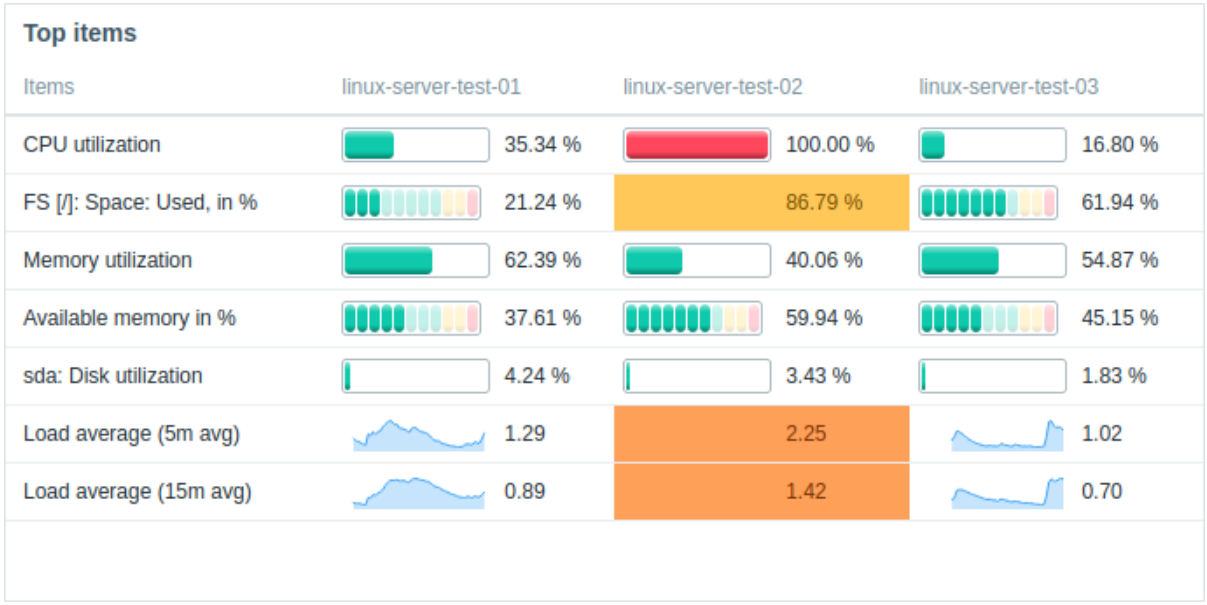


アイテム履歴

アイテム履歴ウィジェットは、すべてのアイテムタイプ（数値、テキストなど）の最新データを表形式で表示します。プログレスバーや画像（ブラウザーアイテムに便利）を表示したり、値をハイライト表示したり（ログファイル監視に便利）することができます。

トップアイテム

トップアイテムウィジェットは、選択したアイテムのトップ/ボトム値を表示し、そのパフォーマンスの概要を迅速に把握できます。

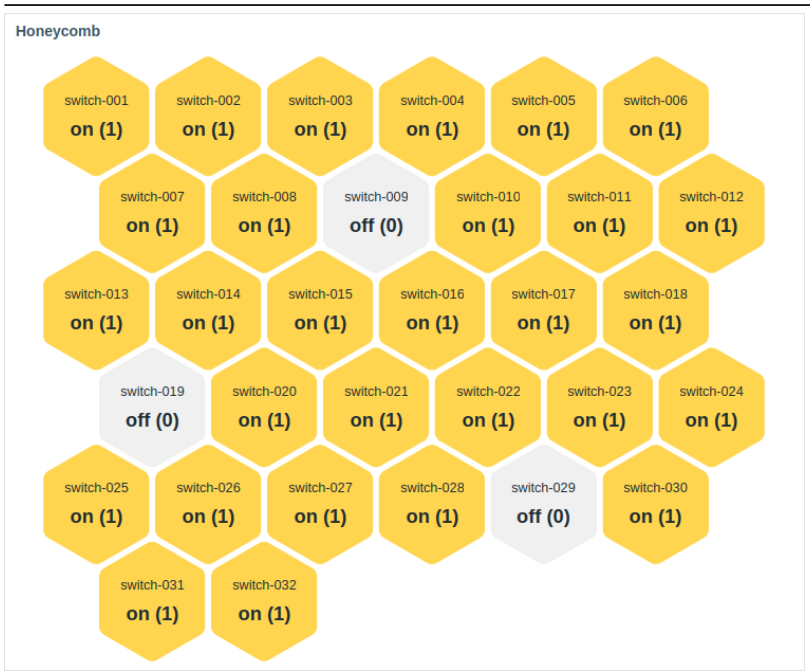


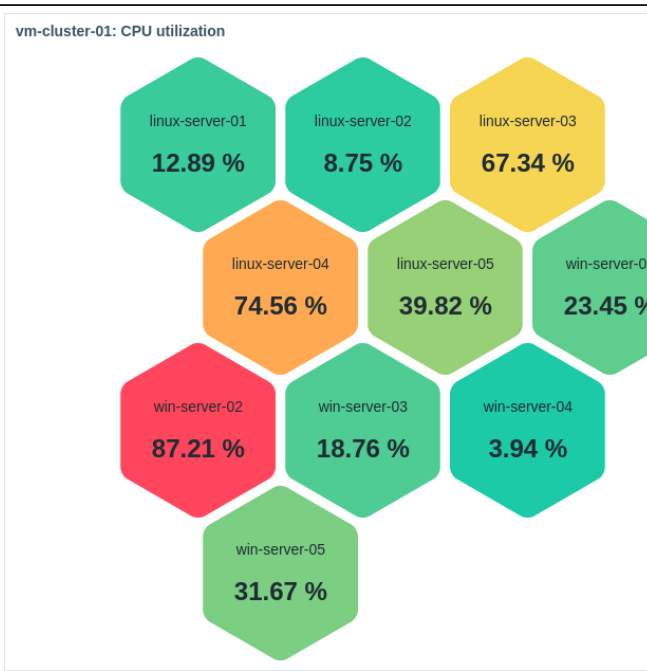
関連情報:

- トップホスト - 複数ホストの選択したアイテムのトップ/ボトム値を表示します。
- トップトリガー - 問題数が最も多いトリガーを表示します。

ハニカム

ハニカムウィジェットは、ホストまたはアイテムを六角形のセルとしてグリッドに表示します。このレイアウトにより、問題のあるホスト/アイテムの検出、問題のクラスターの特定、グループの比較などが容易になります。



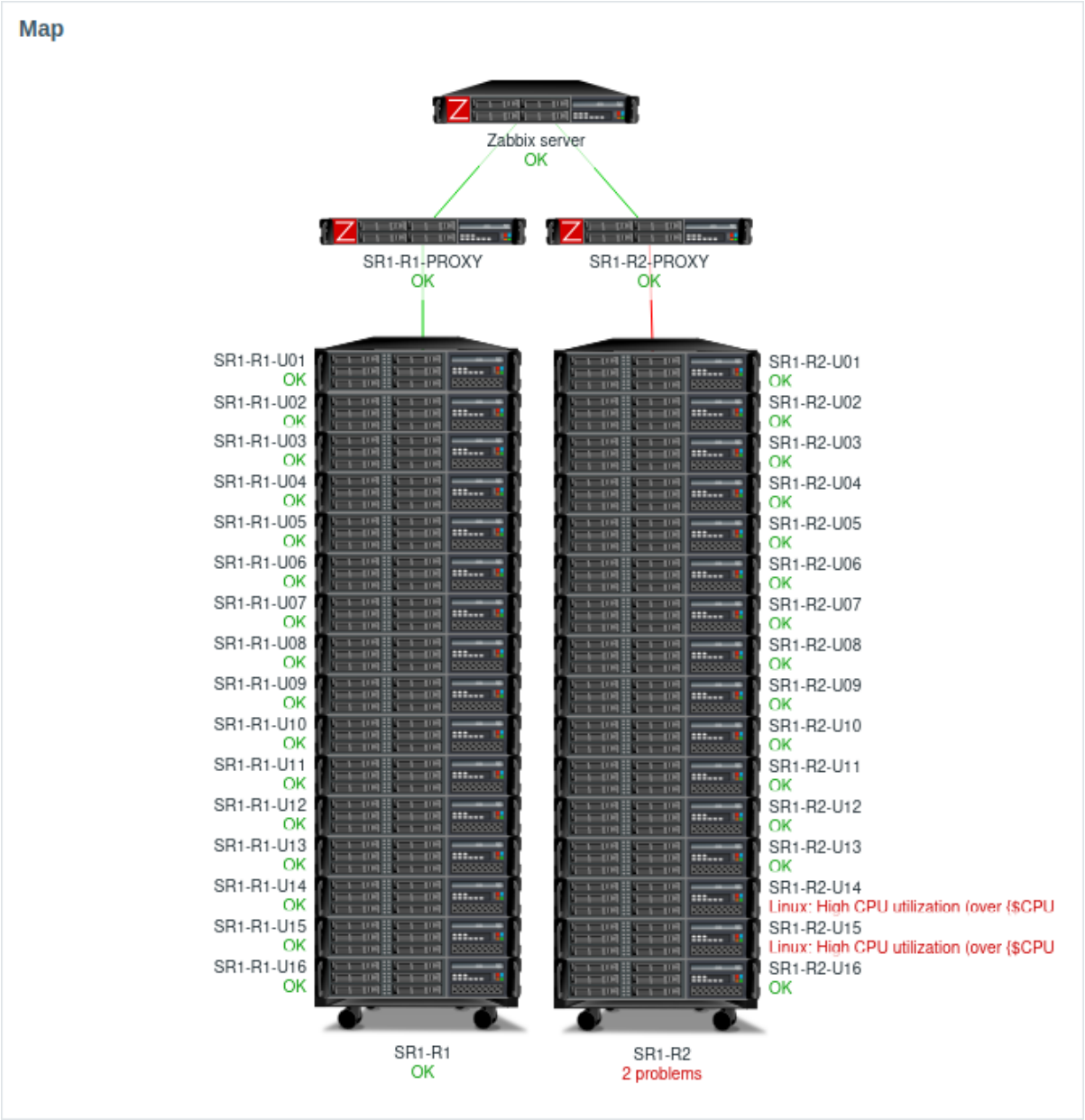


マップ

マップウィジェットは、ネットワークトポロジーやホストの位置を可視化し、全体および場所ごとの健全性を評価するのに役立ちます。

マップ

マップウィジェットは、ネットワークマップや類似のビジュアライゼーションを表示し、ネットワークの動的な概要を提供します。



関連情報:

- マップナビゲーションツリー - 既存のマップの階層と、各マップおよびマップグループの問題数を表示します。
- お気に入りマップ - 現在のユーザーがお気に入りとしてマークしたマップへのショートカットのリストをアルファベット順に表示します。

ジオマップ

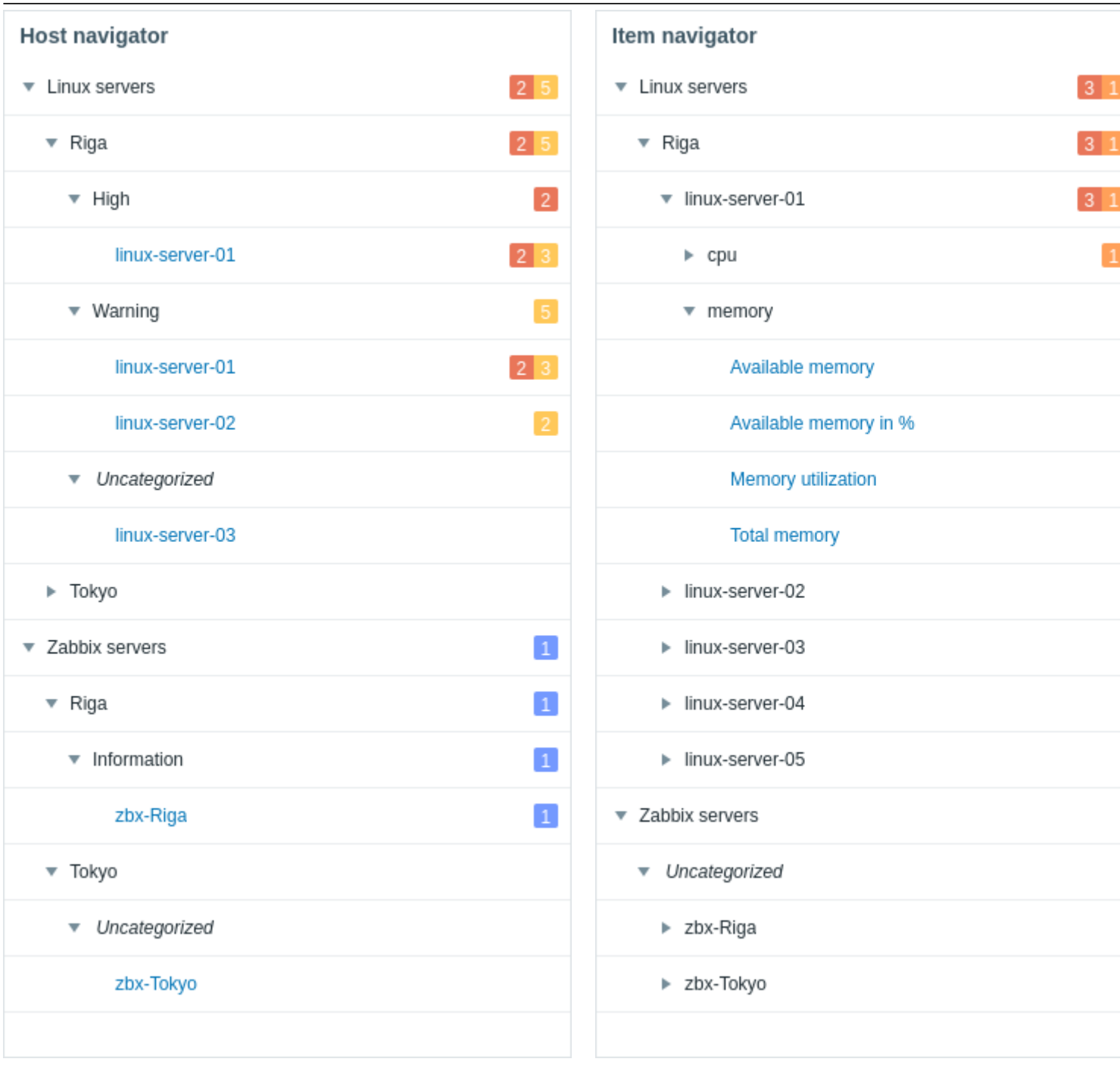
ジオマップウィジェットは、インタラクティブな Leaflet ベースのマップ上にホストをマーカーとして表示し、マーカーの色で各ホストの最も深刻な問題を示します。

ホスト/アイテムの詳細

これらのウィジェットは、ホストまたはアイテムのリストと詳細ビューを提供します。

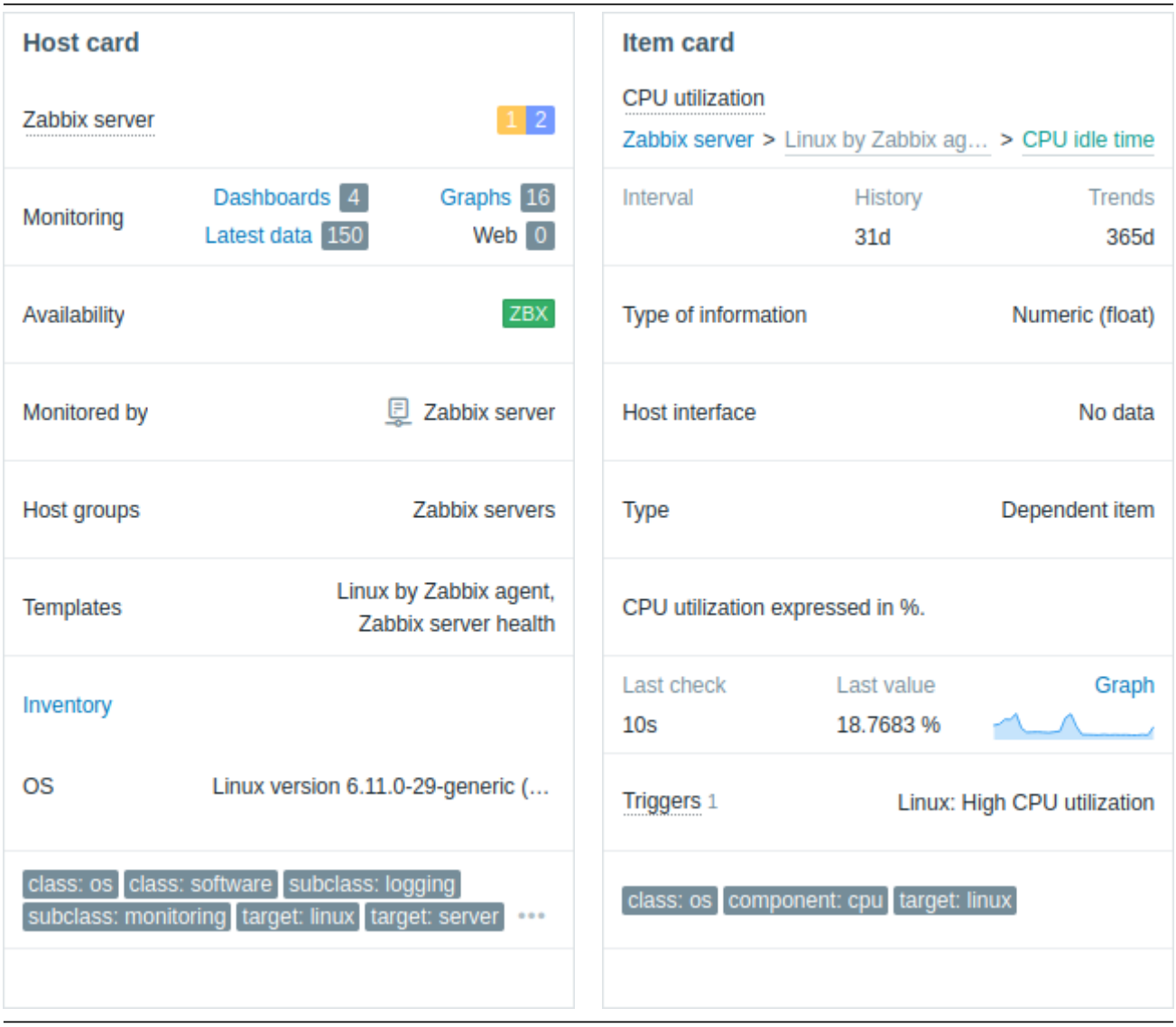
ホスト/アイテムナビゲーター

ホストナビゲーター/アイテムナビゲーターウィジェットは、さまざまなフィルタリングやグループ化オプションに基づいてホストやアイテムのリストを表示します。これらのウィジェットは、選択したホストやアイテムに基づいて他のウィジェットが表示する内容を制御するのに最も便利です。



ホスト/アイテムカード

ホストカード/アイテムカードウィジェットは、単一のホスト/アイテムの詳細を表示し、ホストまたはアイテムのステータスや設定（ホストの可用性、問題、インベントリ、アイテムの最新データ、エラー、説明など）をすばやく確認できます。



ステータスとレポート

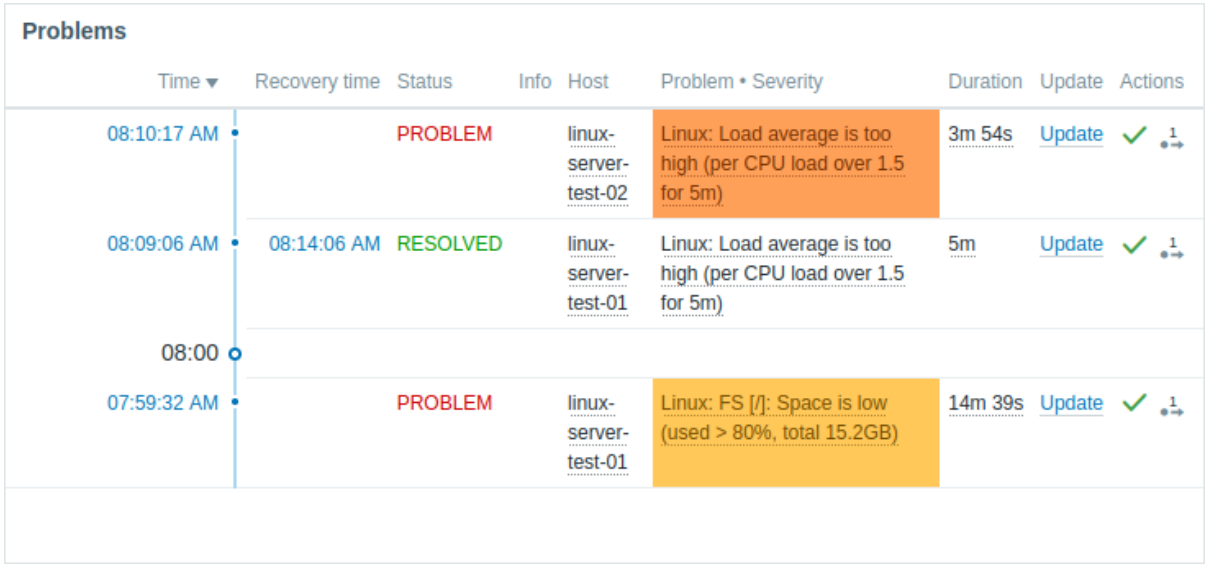
これらのウィジェットは、ステータスデータ（可用性、トリガー、問題など）やレポートを表示します。

ホストの可用性

ホストの可用性ウィジェットは、選択したホストグループ全体のホストの可用性を表示し、稼働中または停止中のホスト数を監視できます。

問題

問題ウィジェットは、現在の問題をさまざまなパラメータ（ホストグループ、ホスト、問題名など）でフィルタリングして表示し、注意が必要な内容を明確に把握できます。



関連情報:

- 深刻度別の問題 - 深刻度ごとにグループ化された原因問題の数を表示します。
- 問題ホスト - ホストグループごとの原因問題の数を表示し、各グループの最大の問題の深刻度を表示します。

トリガー概要

トリガー概要ウィジェットは、ホストトリガーの現在の状態を色付きブロックのテーブルとして表示します。異なるホストのトリガーの健全性とアクティビティの迅速な視覚的スナップショットを提供します。



アクションログ

アクションログウィジェットは、アクション内で実行された操作（通知、リモートコマンド）の詳細を表示します。詳細なログは、監査、トラブルシューティング、および各アクションの実行の監視に役立ちます。

SLA レポート

SLA レポートウィジェットは、SLA レポートを表示し、定義された目標に対するサービスのパフォーマンスを監視するのに役立ちます。

SLA report

Day	SLO	SLI	Uptime	Downtime	Error budget	Excluded downtimes
2025-09-04	99.9%	99.8437	5h 40m 50s	28s	-8s	
2025-09-03	99.9%	99.9479	23h 59m 15s	45s	41s	
2025-09-02	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-09-01	99.9%	99.9251	22h 58m 58s	1h 1m 2s	21s	2025-09-01 01:00 AM Maintenance: 1h
2025-08-31	99.9%	99.6863	23h 55m 29s	4m 31s	-3m 5s	
2025-08-30	99.9%	99.9190	23h 58m 50s	1m 10s	16s	
2025-08-29	99.9%	99.7095	23h 55m 49s	4m 11s	-2m 45s	
2025-08-28	99.9%	99.9421	23h 59m 10s	50s	36s	
2025-08-27	99.9%	99.7188	23h 55m 57s	3m 15s	-2m 37s	
2025-08-26	99.9%	99.7755	23h 56m 46s	3m 14s	-1m 48s	
2025-08-25	99.9%	99.9094	22h 58m 45s	1h 1m 15s	8s	2025-08-25 01:00 AM Maintenance: 1h

Web 監視

Web 監視ウィジェットは、アクティブな Web シナリオのステータスを表示します。

Web monitoring

Host group ▲	Ok	Failed	Unknown
Applications	1	1	1
Applications/External	1		
Applications/Internal			1
Applications/Test		1	

ディスカバリーステータス

ディスカバリーステータスウィジェットは、有効なネットワークディスカバリールールによって検出されたデバイスのステータスを表示します。ネットワークディスカバリーが正しく機能しているかどうか、および稼働中または停止中のデバイスの数を監視するのに役立ちます。

ダッシュボードのユーティリティとシステム情報

これらのウィジェットは、ダッシュボードを強化するための補足情報やツールを提供します。

システム情報

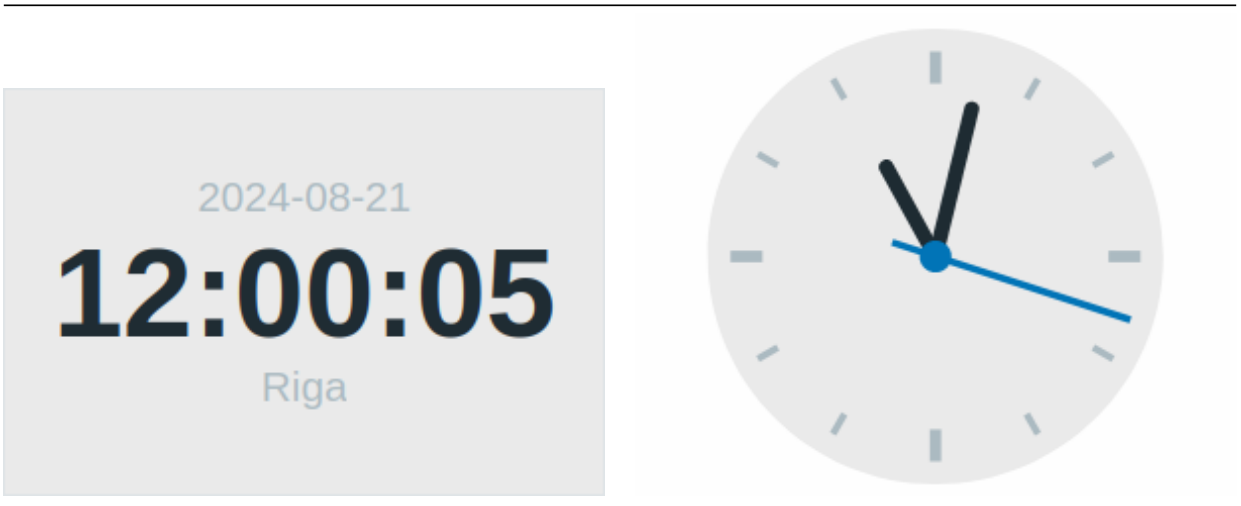
システム情報ウィジェットは、Zabbix サーバーおよびシステムデータの主要な概要、または高可用性ノードの詳細を表示します。

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	8.0.0	Up to date
Zabbix frontend version	8.0.0	Up to date
Software update last checked	2025-08-30	
Latest release	8.0.0	Release notes
Number of hosts (enabled/disabled)	2	2 / 0
Number of templates	312	
Number of items (enabled/disabled/not supported)	176	165 / 0 / 11
Number of triggers (enabled/disabled [problem/ok])	89	89 / 0 [4 / 85]
Number of users (online)	2	1
Required server performance, new values per second	2.68	
Global scripts on Zabbix server	Disabled	
High availability cluster	Disabled	

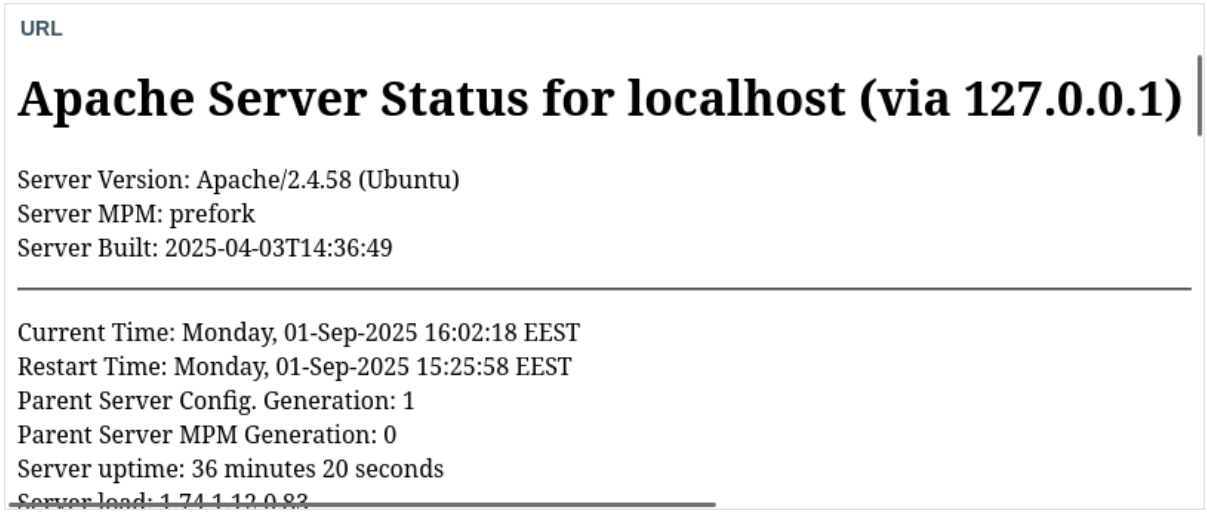
時計

時計ウィジェットは、ローカル、サーバー、またはホストの時刻をアナログまたはデジタル形式で表示します。



URL

URLウィジェットは、URL から取得したコンテンツを表示します。



ダッシュボードウィジェットのパラメーター

概要

このページでは、すべてのダッシュボードウィジェットに共通するパラメーターと、複数のウィジェットが互いに、またはダッシュボードと構成データを交換できる動的パラメーターについて説明します。

各ウィジェットの固有のパラメーターについては、サイドバーから該当するウィジェットのページに移動してください。

共通パラメータ

以下のパラメータは、すべてのウィジェットに共通です。

名前	ウィジェット名を入力します。
更新間隔	デフォルトの更新間隔を設定します。 ウィジェットのデフォルトの更新間隔は、ウィジェットの種類に応じて 更新なしから *15* 分までの範囲で設定されています。例： - *URL* ウィジェットでは 更新なし; - アクションログウィジェットでは *1* 分; - 時計ウィジェットでは *15* 分。 すべてのユーザーに対して、更新間隔をデフォルト値に設定できます。ダッシュボードを編集モードに切り替え、ウィジェットを編集ボタンをクリックして、ドロップダウンリストから目的の更新間隔を選択します。 各ユーザーは、自分のウィジェット更新間隔を設定することもできます。ダッシュボードの表示モードで、ウィジェット上の 3 つのドット ボタンをクリックし、ドロップダウンリストから目的の更新間隔を選択します。ユーザー固有の更新間隔はウィジェットの設定よりも優先され、ウィジェットの設定が変更された場合でも保持されることに注意してください。
ヘッダーを表示	チェックボックスを選択すると、ウィジェットのヘッダーが常に表示されます。 選択を解除すると、スペースを節約するためにヘッダーが非表示になり、ウィジェット上にマウスを移動した場合にのみ表示されます（表示モードと編集モードの両方）。また、ウィジェットを新しい場所にドラッグしている間は、ヘッダーが半透明で表示されます。

動的パラメータ

複数のウィジェットパラメータ（例：ホスト、ホストの上書き、期間）は動的データソースをサポートしています。静的な値を設定する代わりに、これらのパラメータを他のウィジェットやダッシュボード自体にリンクできます。ウィジェットは、ダッシュボードの他の部分からの選択やブロードキャストに基づいて自動的に更新されます。

ウィジェットの互換性

一部のウィジェットは他のウィジェットに設定データをブロードキャストでき、一部はデータをリッスンでき、また一部はその両方が可能です。例えば:

- アクションログウィジェットは、グラフ、グラフ (クラシック)、グラフプロトタイプウィジェットから期間データのみを取得できます。

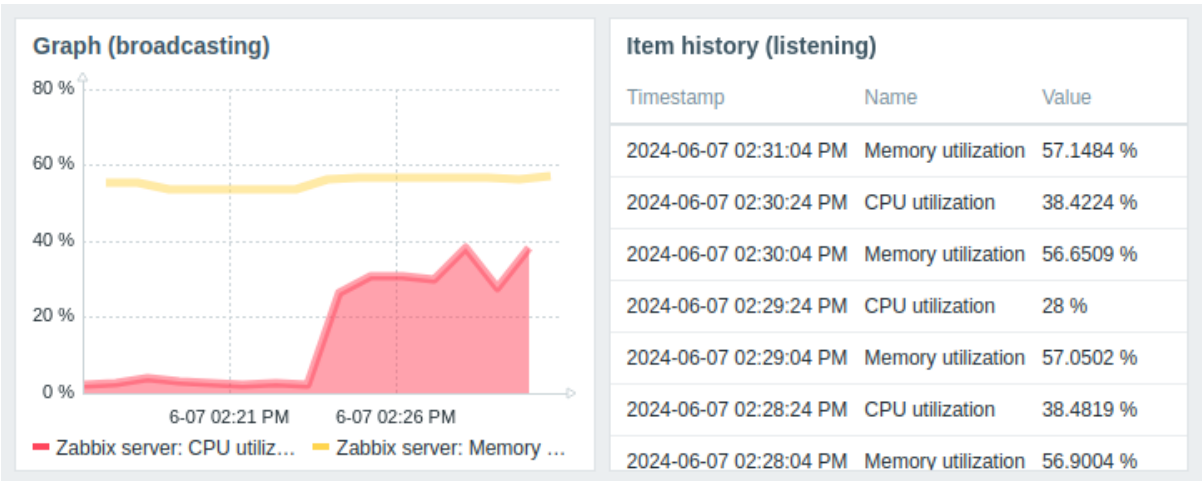
- ジオマップウィジェットは、ホストデータをリッスンするウィジェット (ハニカム、トップアイテムなど) にホストデータをブロードキャストでき、また、ホストグループやホストデータをブロードキャストするウィジェット (ハニカム、問題のあるホストなど) からデータをリッスンすることもできます。
- 時計ウィジェットは、データのブロードキャストやリッスンはできません。

以下の表は、各ウィジェットのブロードキャストおよびリッスンの機能を示しています。

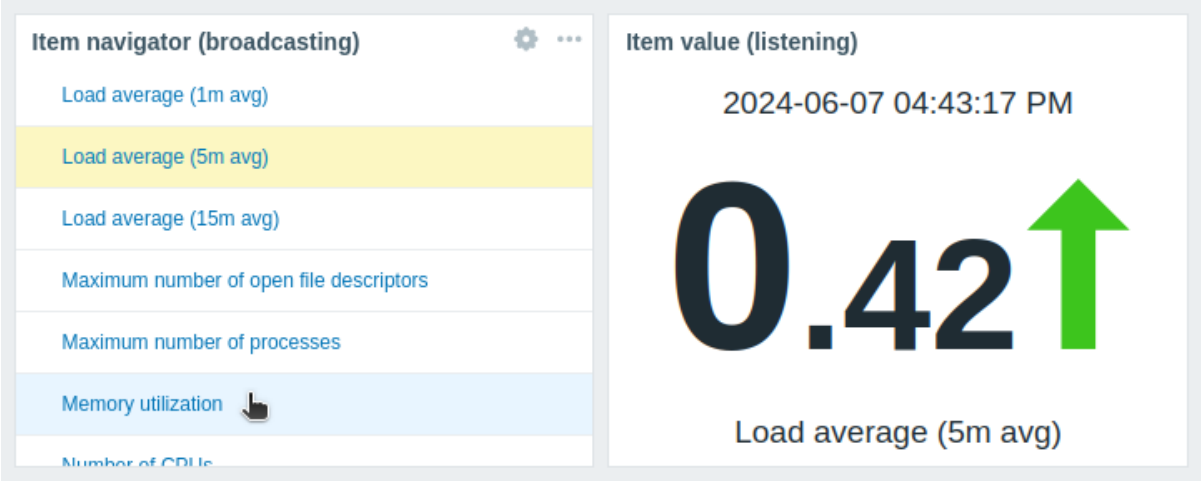
ウィジェット	ブロードキャスト	リッスン
アクションログ	-	期間
時計	-	-
ディスカバリの状態	-	-
お気に入りのグラフ	-	-
お気に入りのマップ	-	-
ゲージ	-	ホスト、アイテム
ジオマップ	ホスト	ホストグループ、ホスト
グラフ	期間、アイテム	ホスト、アイテム、期間
グラフ (クラシック)	期間	ホスト、アイテム、グラフ、期間
グラフプロトタイプ	期間	ホスト、期間
ハニカム	ホスト、アイテム	ホストグループ、ホスト
ホストの可用性	-	ホストグループ
ホストカード	-	ホスト
ホストナビゲータ	ホスト	ホストグループ
アイテムカード	-	ホスト、アイテム、期間
アイテム履歴	アイテム	ホスト、期間
アイテムナビゲータ	アイテム	ホストグループ、ホスト
アイテム値	-	ホスト、アイテム、期間
マップ	ホストグループ、ホスト	マップ
マップナビゲーションツリー	マップ	-
円グラフ	-	アイテム、期間
問題のあるホスト	ホストグループ	ホストグループ、ホスト
障害	イベント	ホストグループ、ホスト
深刻度別の障害	ホストグループ	ホストグループ、ホスト
散布図	期間、アイテム	ホスト、アイテム、期間
SLA レポート	-	-
システム情報	-	-
トップホスト	ホスト	ホストグループ、ホスト
トップアイテム	-	ホストグループ、ホスト
トップトリガー	-	期間
トリガーの概要	-	ホストグループ、ホスト
URL	-	ホスト
Web 監視	ホストグループ	ホストグループ、ホスト

ウィジェットは、他のウィジェットにデータをブロードキャストする方法が異なります。

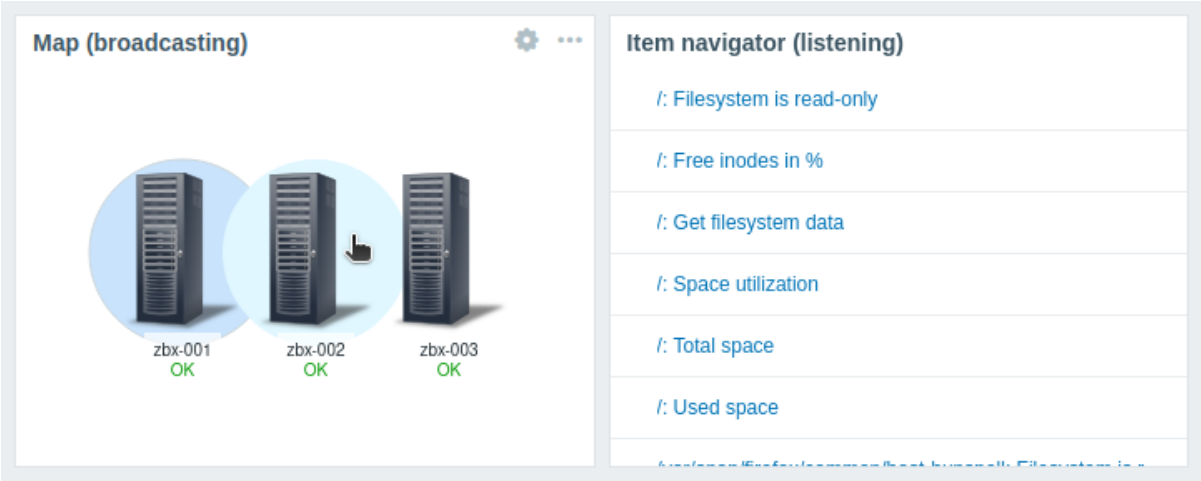
ブロードキャストをサポートするウィジェットは、作成時に自動的にブロードキャストを開始します。たとえば、グラフウィジェットは、リスニングウィジェットに対して期間データを即座にブロードキャストします。



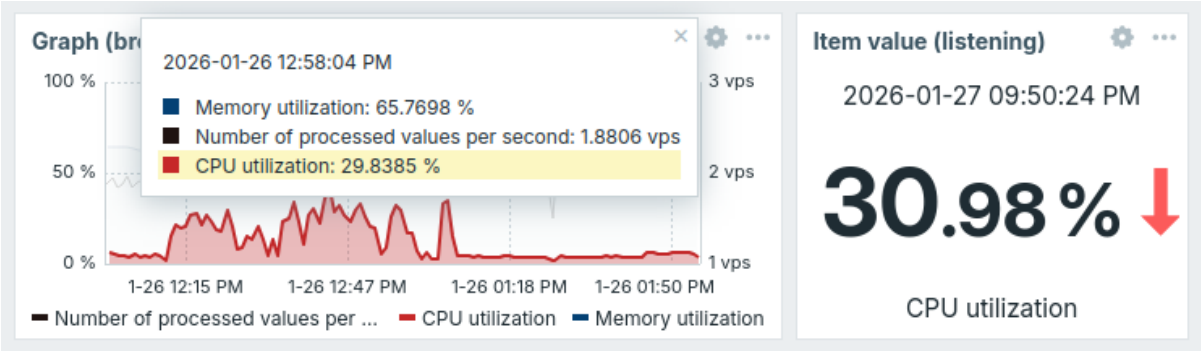
エンティティの選択をサポートするウィジェットは、最初に利用可能なエンティティのデータを自動的にブロードキャストします。たとえば、アイテムナビゲーターウィジェットは、アイテムリスト内の最初のアイテムのデータをブロードキャストします。別のアイテムを選択すると、そのアイテムのデータをブロードキャストします。マウスオーバー時にはアイテムが水色でハイライトされ、選択時には黄色でハイライトされます。



マップウィジェットも同様に動作し、ウィジェットの表示領域の左上隅に最も近い要素（ジオマップウィジェットの場合は中心に最も近い要素）のデータをブロードキャストします。別の要素を選択すると、その要素のデータをブロードキャストします。マウスオーバー時には要素が水色でハイライトされ、選択時には濃い青色でハイライトされます。

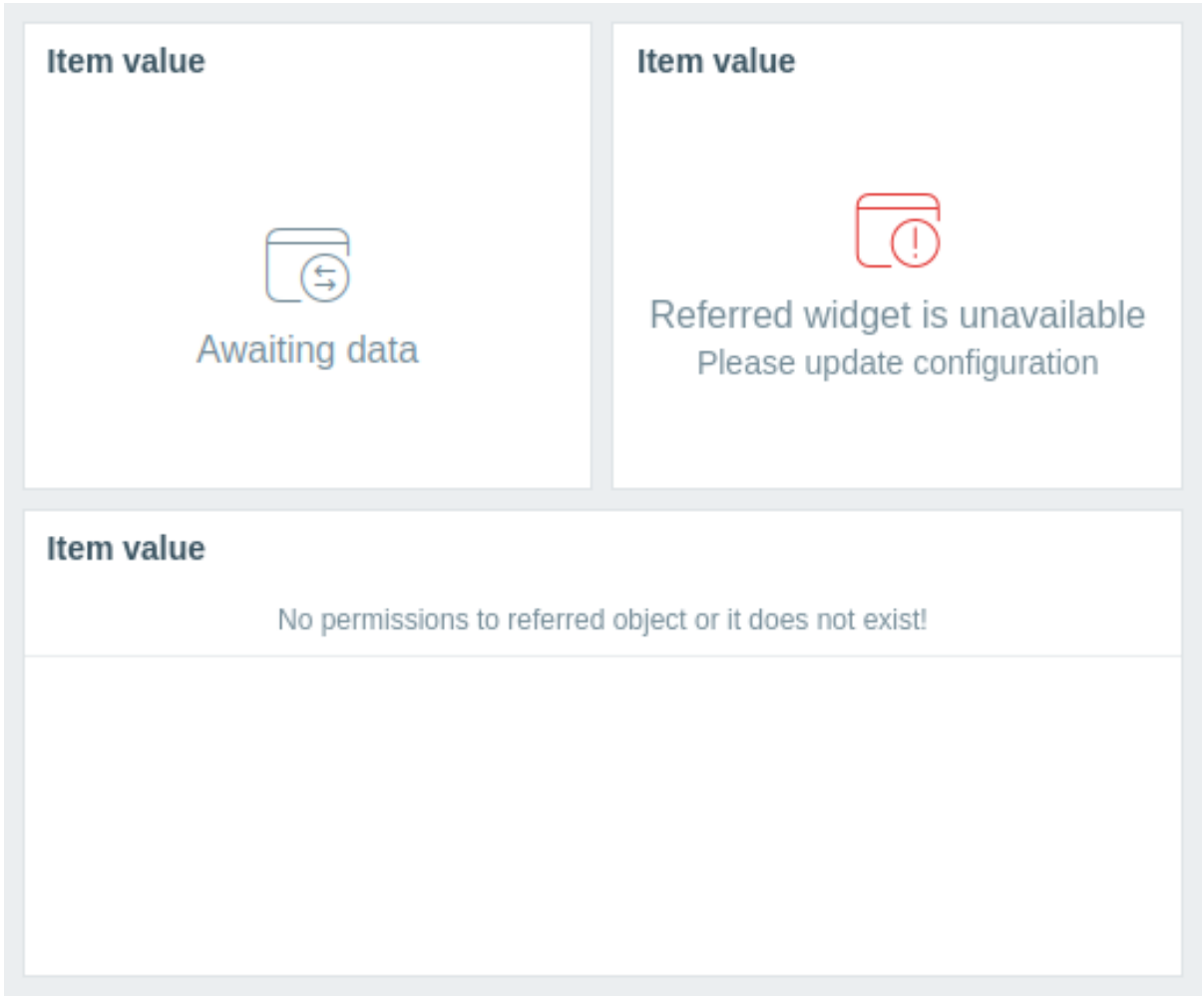


グラフや散布図などのウィジェットは、ツールチップ内でアイテムを選択したときにアイテムデータをブロードキャストできます。



ウィジェットは、他のウィジェットからのデータをリッスンする際にも特定の動作をします。

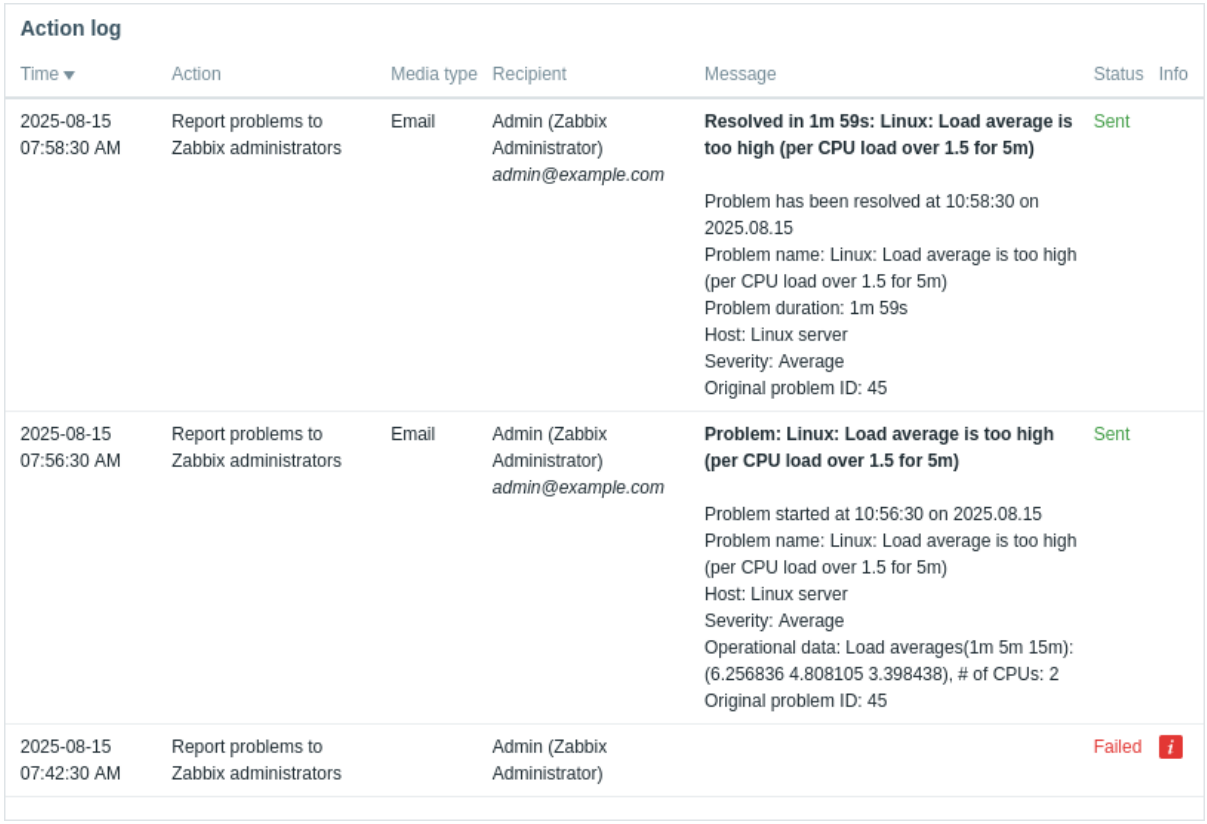
- データソースウィジェットがデータをブロードキャストしていない場合、リスニングウィジェットはデータ待機中の状態になります。
- データソースウィジェットが削除された場合、互換性のないウィジェットに置き換えられた場合、または別のダッシュボードページに移動された場合、リスニングウィジェットは参照ウィジェットが利用できませんの状態になります。
- データソース（ウィジェットまたはダッシュボード）で指定されたホストに、リスニングウィジェットで設定されたエンティティ（アイテム、グラフ、マップなど）が存在しない場合、またはユーザーにホストへのアクセス権限がない場合、リスニングウィジェットには次のメッセージが表示されます：”参照オブジェクトへの権限がないか、存在しません！”



アクションログ

概要

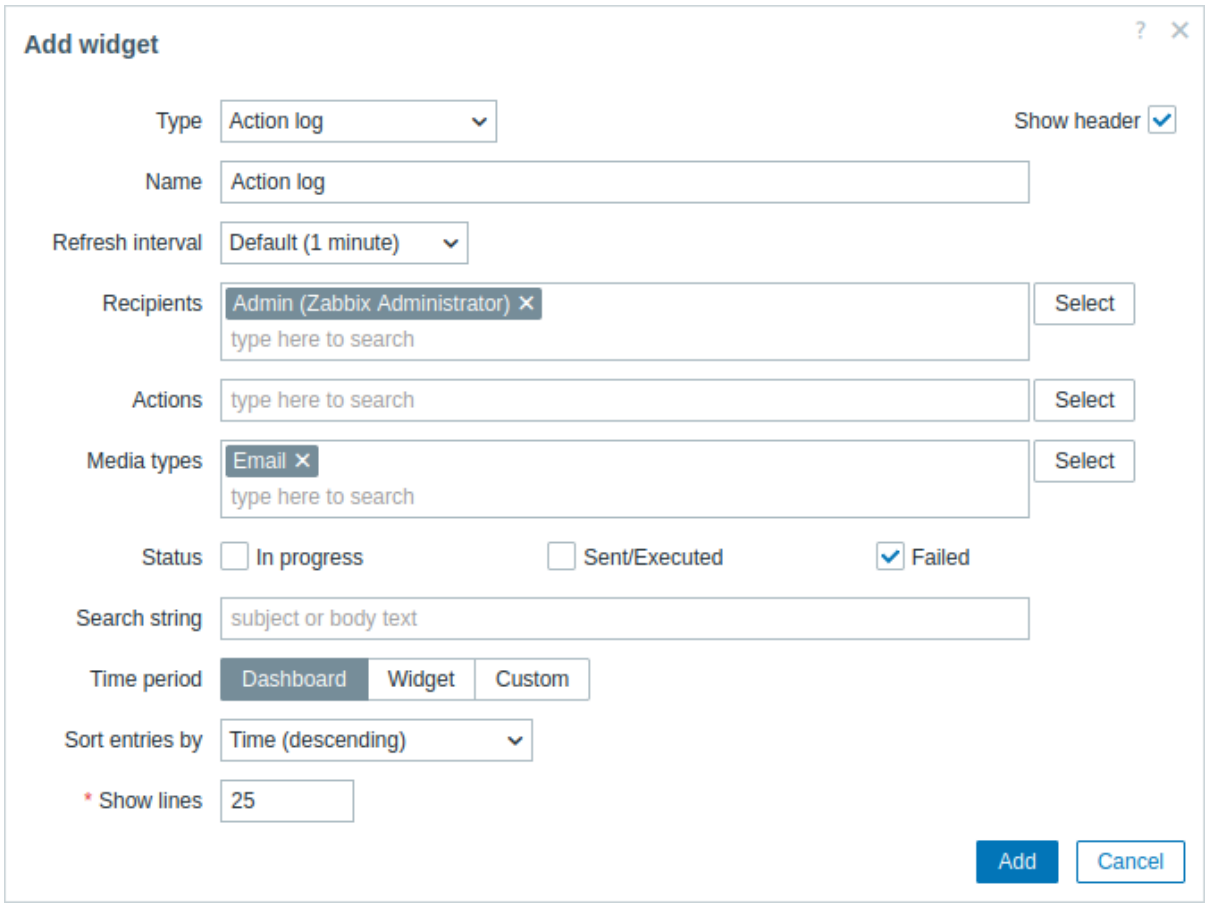
アクションログウィジェットは、アクション内で実行されたオペレーション（通知、リモートコマンド）の詳細を表示します。詳細なログは、各アクションの実行の監査、トラブルシューティング、監視に役立ちます。



レポート > アクションログと同じデータを表示し、最大 1000 件のレコードを表示できます。

設定

設定するには、タイプとして アクションログを選択します。



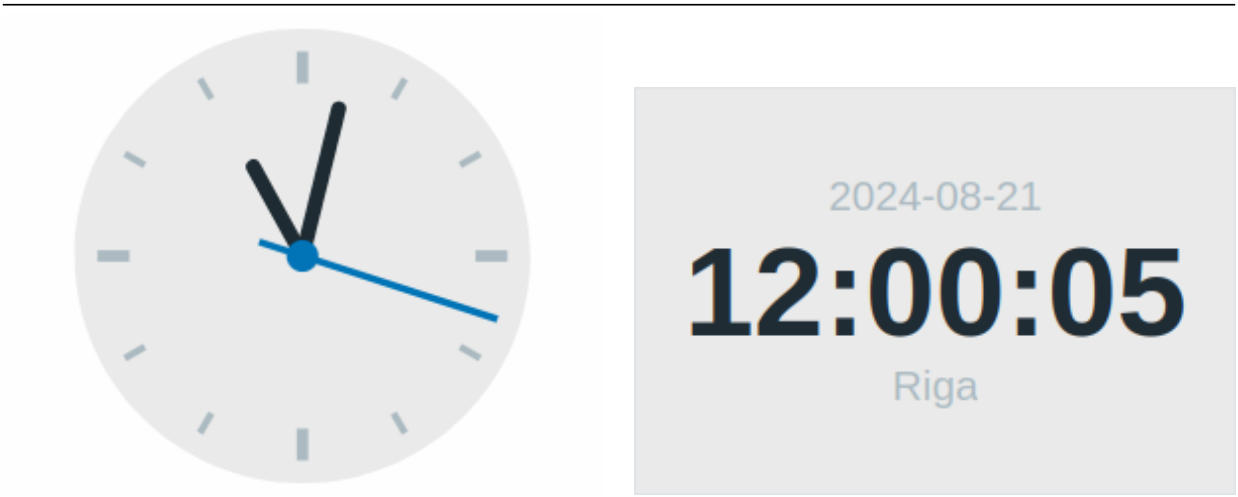
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

受信者	受信者でエントリをフィルタリングします。このフィールドはオートコンプリート対応で、受信者名の入力を始めると一致する受信者のドロップダウンが表示されます。受信者が選択されていない場合、すべての受信者のアクション操作の詳細が表示されます。
アクション	アクションでエントリをフィルタリングします。このフィールドはオートコンプリート対応で、アクション名の入力を始めると一致するアクションのドロップダウンが表示されます。アクションが選択されていない場合、すべてのアクションのアクション操作の詳細が表示されます。
メディアタイプ	メディアタイプでエントリをフィルタリングします。このフィールドはオートコンプリート対応で、メディアタイプ名の入力を始めると一致するメディアタイプのドロップダウンが表示されます。メディアタイプが選択されていない場合、すべてのメディアタイプのアクション操作の詳細が表示されます。
ステータス	それぞれのステータスでエントリをフィルタリングするにはチェックボックスをマークします: 進行中 - 進行中のアクション操作が表示されます; 送信済み/実行済み - 通知が送信された、または実行されたアクション操作が表示されます; 失敗 - 失敗したアクション操作が表示されます。
検索文字列	メッセージ/リモートコマンドの内容でエントリをフィルタリングします。ここに文字列を入力すると、その文字列を含むメッセージ/リモートコマンドのアクション操作のみが表示されます。マクロは展開されません。
期間	期間でエントリをフィルタリングします。期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクタを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - *From* および *To* パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できることに注意してください。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
エントリの並び順	エントリを以下で並び替えます: 時刻（降順または昇順）; タイプ（降順または昇順）; ステータス（降順または昇順）; 受信者（降順または昇順）。
表示行数	ウィジェットに表示するアクションログの行数を設定します。

クロック

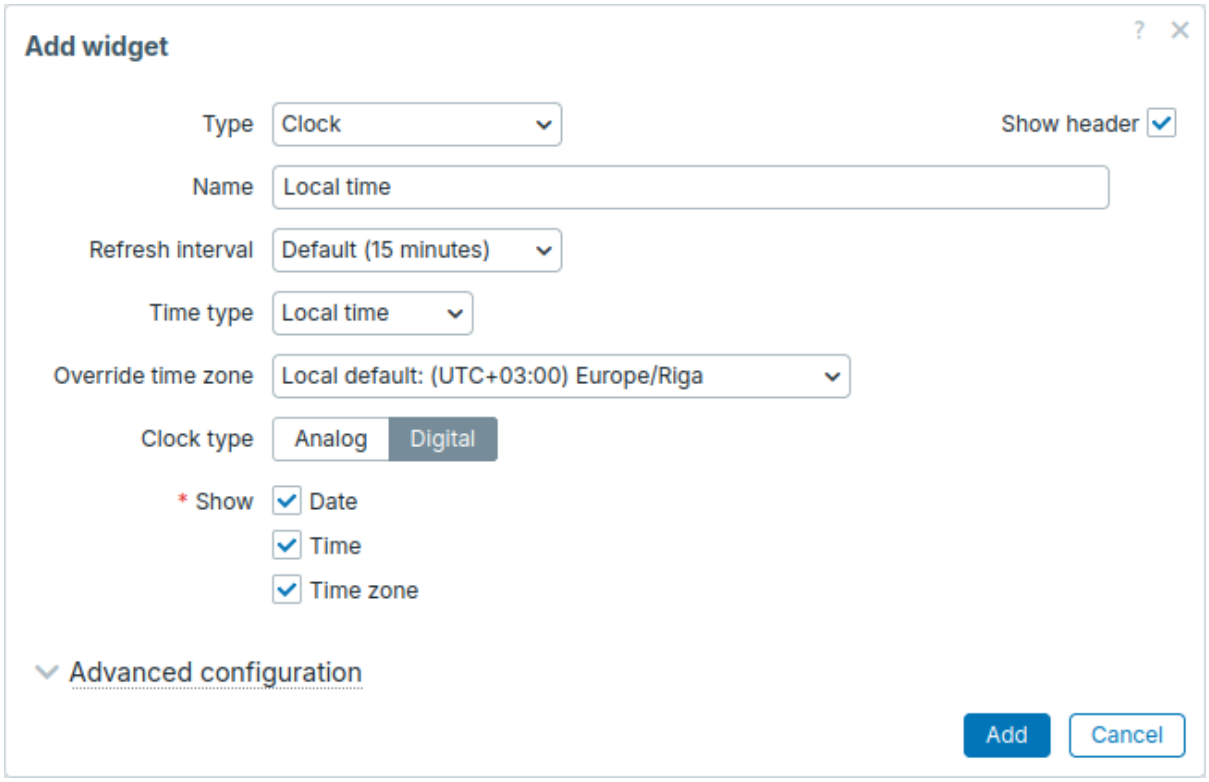
概要

時計ウィジェットは、アナログまたはデジタル形式でローカル、サーバー、またはホストの時刻を表示します。



設定

設定するには、タイプとして *Clock* を選択します。



すべてのウィジェットに共通するパラメータに加えて、次の固有のオプションを設定できます。

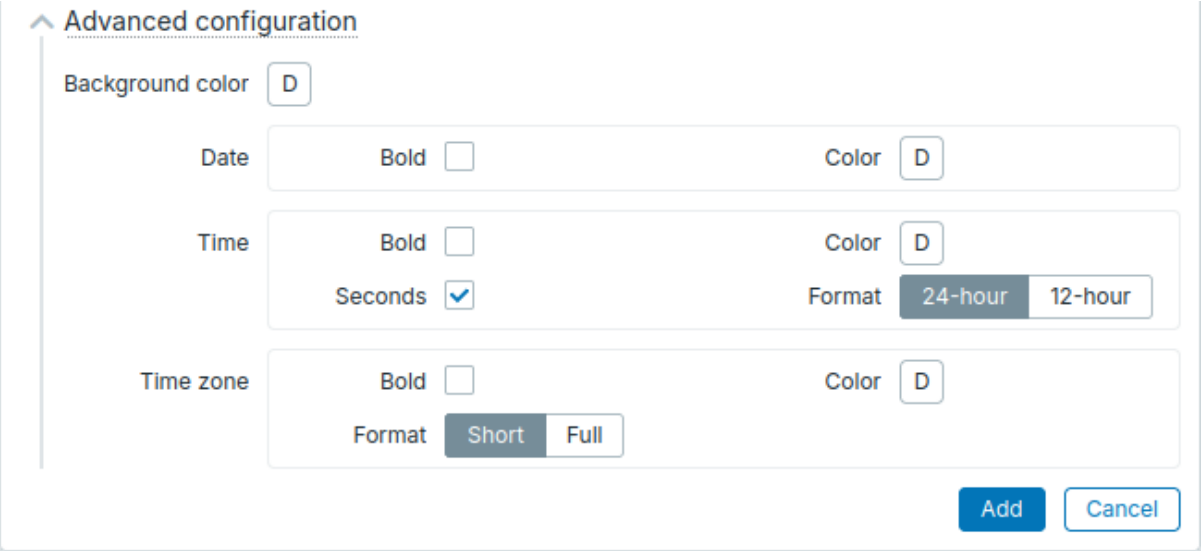
Time type	時刻のソースを選択します。 **Local time** - ブラウザーを実行しているデバイスのシステム時刻。 **Server time** - Zabbix ユーザーに対してグローバルまたは設定されたタイムゾーン。 **Host time** - ホストのアイテムから取得した時刻。
Item	ホスト時刻を表示するために、`system.localtime[local]`アイテムキーを持つアイテムを選択します。 このパラメータは、*Time type* が「Host time」に設定されている場合に使用できます。
Clock type	時計のタイプ（アナログまたはデジタル）を選択します。
Override time zone	タイムゾーンを選択します。 このパラメータは、*Local time* および *Server time* の時刻タイプで使用できます。
Show	デジタル時計に表示する情報単位（日時、時刻、タイムゾーン）を 1 つ以上選択します。 単位のサイズは、表示する単位、ウィジェットのサイズ、および表示倍率に応じて自動的に調整されます。 このパラメータは、*Clock type* が「Digital」に設定されている場合に使用できます。

Advanced configuration	*Advanced configuration* ヘッダーをクリックすると、デジタル時計の高度な設定オプションが表示されます。 このセクションは、*Clock type* が「Digital」に設定されている場合に使用できます。

高度な設定

高度な設定オプションは、折りたたみ可能な高度な設定セクションで使用できます。また、表示フィールドで選択されている要素に対してのみ使用できます（上記参照）。

さらに、高度な設定ではウィジェット全体の背景色を変更できます。

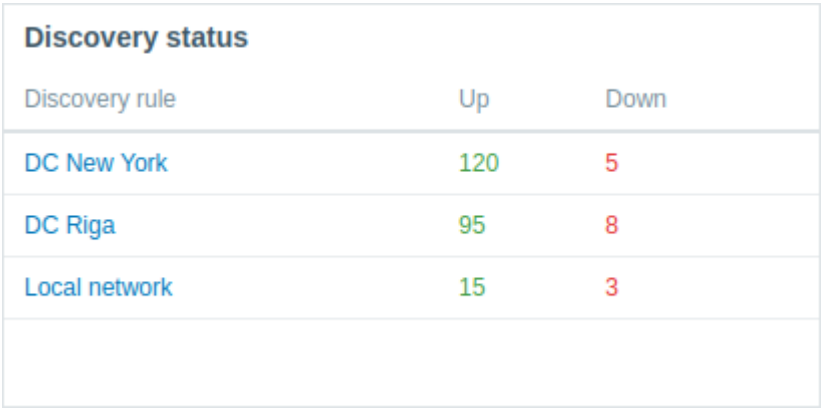


背景色		カラーピッカーから背景色を選択します。 D はデフォルトの色（Web インターフェースのテーマによって異なります）を示します。 デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
日付		このセクションは、表示パラメータで日付チェックボックスが選択されている場合に使用できます。
	太字	チェックボックスを選択すると、日付が太字で表示されます。
	色	カラーピッカーから日付の色を選択します。 D はデフォルトの色（Web インターフェースのテーマによって異なります）を示します。 デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
時刻		このセクションは、表示パラメータで時刻チェックボックスが選択されている場合に使用できます。
	太字	チェックボックスを選択すると、時刻が太字で表示されます。
	色	カラーピッカーから時刻の色を選択します。 D はデフォルトの色（Web インターフェースのテーマによって異なります）を示します。 デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
	秒	チェックボックスを選択すると、秒が表示されます。それ以外の場合は、時と分のみが表示されます。
	形式	24 時間形式または 12 時間形式で表示するよう選択します。
タイムゾーン		このセクションは、表示パラメータでタイムゾーンチェックボックスが選択されている場合に使用できます。
	太字	チェックボックスを選択すると、タイムゾーンが太字で表示されます。
	色	カラーピッカーからタイムゾーンの色を選択します。 D はデフォルトの色（Web インターフェースのテーマによって異なります）を示します。 デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
	形式	タイムゾーンを短い形式（例: `New York`）または完全な形式（例: `(UTC-04:00) America/New York`）で表示するよう選択します。

ディスカバリーステータス

概要

ディスカバリーステータスウィジェットは、有効なネットワークディスカバリールールによって検出されたデバイスのステータスを表示します。ネットワークディスカバリーが正しく動作しているかどうか、また、稼働中または停止中のデバイスの数を監視するのに役立ちます。



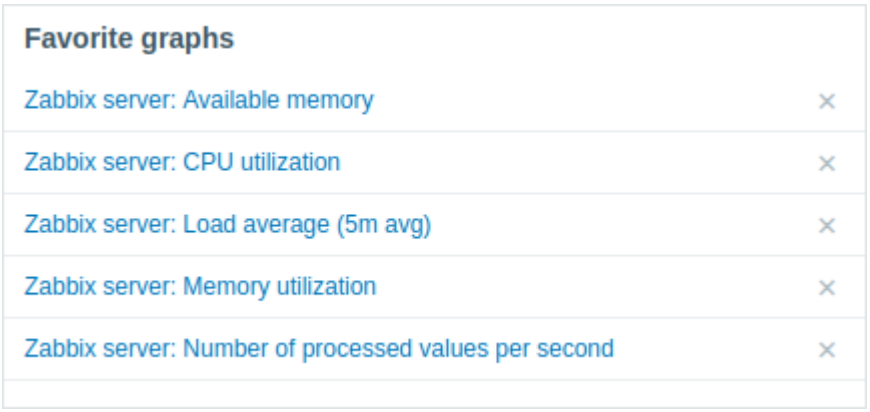
設定

このウィジェットのすべての設定パラメータは、すべてのウィジェットに共通です。

お気に入りグラフ

概要

お気に入りグラフウィジェットは、現在のユーザーがお気に入りとしてマークしたグラフへのショートカットのアルファベット順リストを表示します。



x をクリックすると、リストとお気に入りからグラフが削除されます。

グラフは、監視 > 最新データでアイテムグラフを表示しているときに、お気に入りに追加ボタンをクリックすることで、お気に入りとしてマークされます。



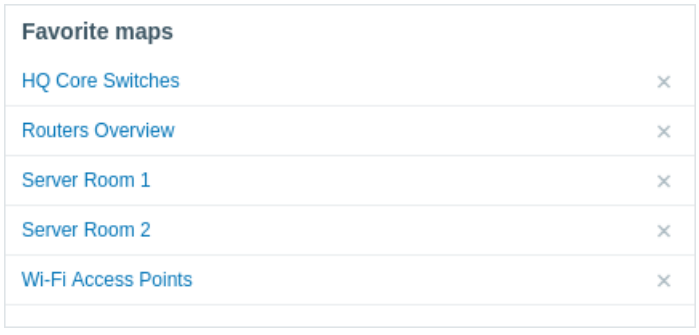
設定

このウィジェットのすべての設定パラメータは、すべてのウィジェットに共通です。

お気に入りのマップ

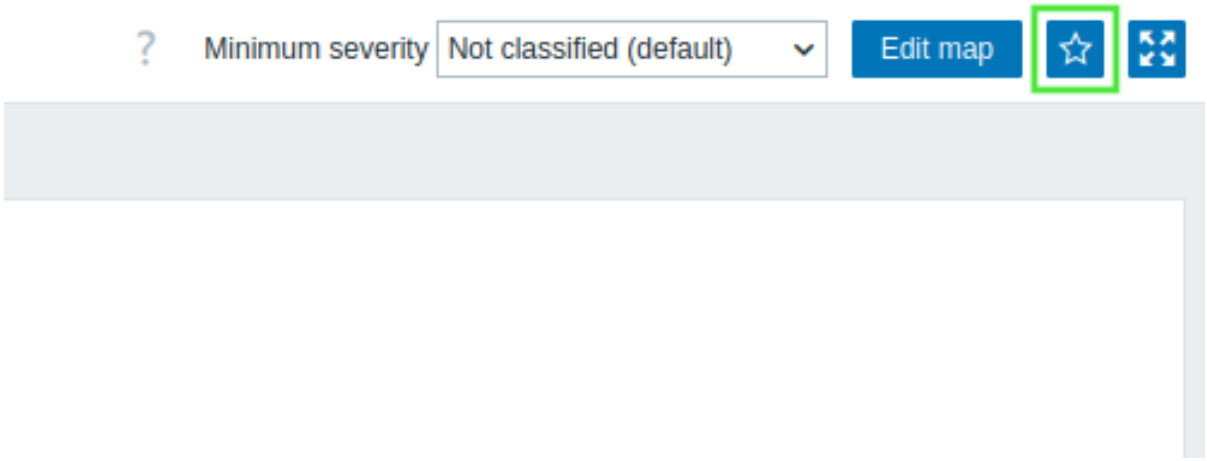
概要

お気に入りのマップウィジェットは、現在のユーザーがお気に入りに登録したマップへのショートカットのリストをアルファベット順に表示します。



x をクリックすると、お気に入りからマップが削除されます。

マップは、監視 > マップでマップを表示しているときに、お気に入りに追加ボタンをクリックすることでお気に入りに登録されます。



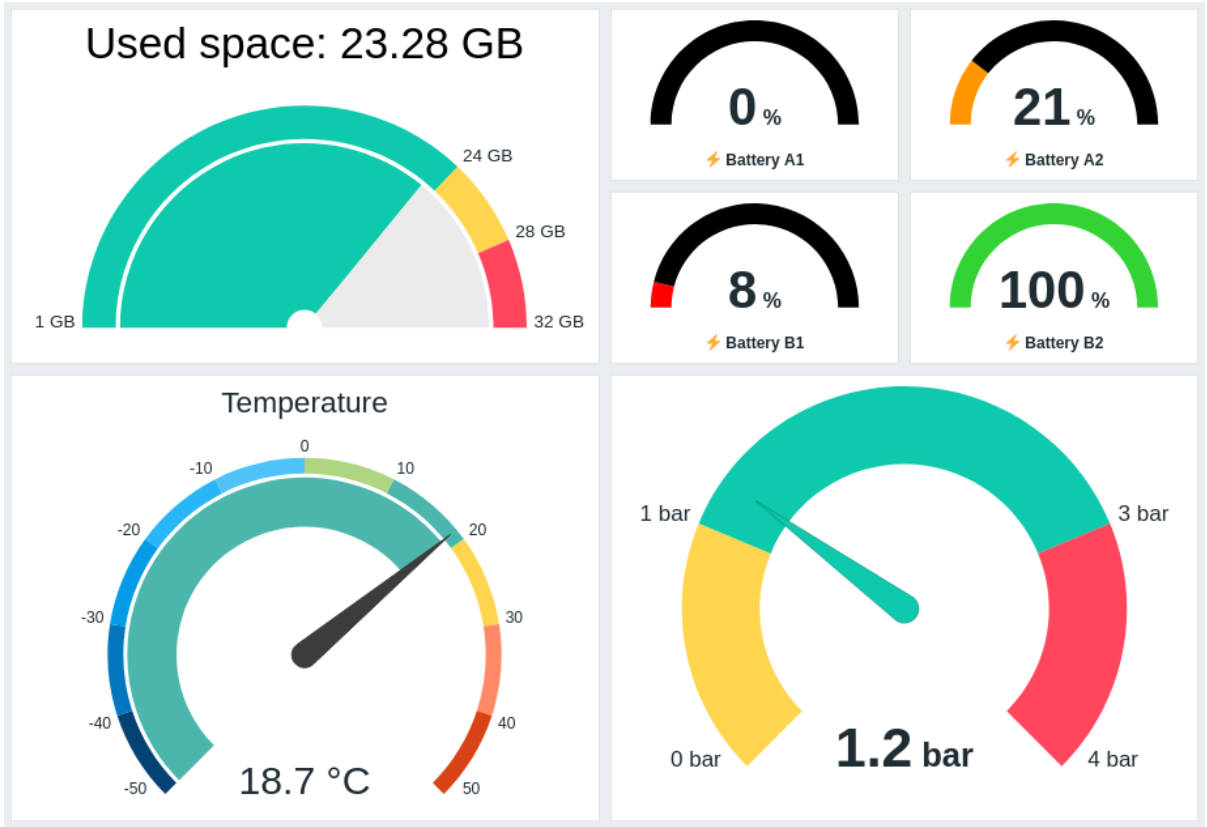
設定

このウィジェットのすべての設定パラメータは、すべてのウィジェットに共通です。

ゲージ

概要

ゲージウィジェットは、単一の数値アイテムの値をゲージとして表示します。重要なメトリクスを監視したり、しきい値を視覚化したり、急激な変化を検出したりする場合に便利です。



ウィジェットには、次の情報を表示するように設定できます。

- アイテムの説明（使用済み領域: {ITEM.LASTVALUE}、温度）
- アイテムの値と単位（21 %、18.7 °C、1.2 bar）
- ゲージの目盛り（1GB/24GB/28GB/32GB、-50/-40/-30 など）
- ゲージの円弧（ゲージ値の円弧とゲージしきい値の円弧）
- ゲージの針

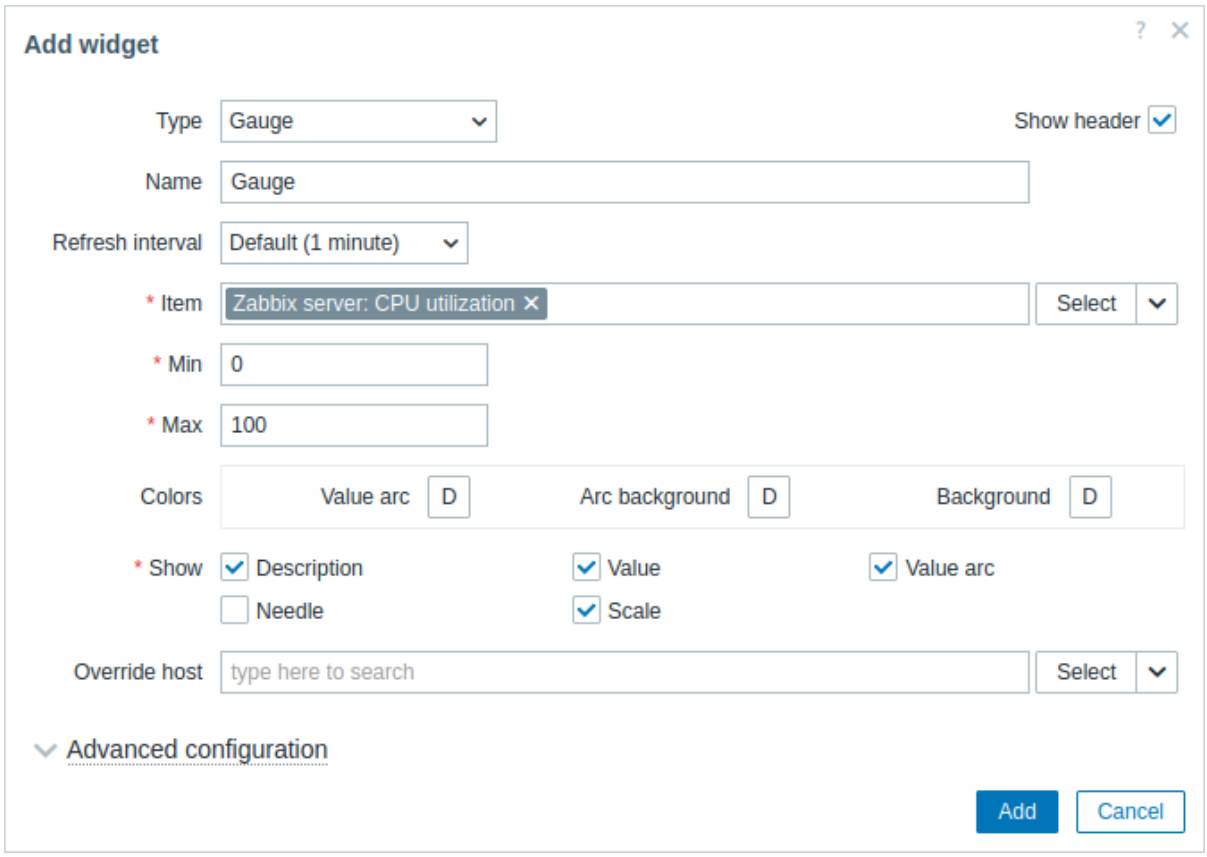
ウィジェットをクリックすると、そのアイテムのシンプルグラフが開きます。

ゲージウィジェットに表示される情報は、ウィジェット上にマウスカーソルを合わせたときに表示されるウィジェットのアクションメニュー ... から画像をダウンロードを選択すると、PNG 画像としてダウンロードできます。ブラウザーの設定に応じて、画像は自動

的に Downloads フォルダーに保存されるか、保存場所を選択できる保存ダイアログが開きます。

設定

設定するには、タイプとして *Gauge* を選択します。



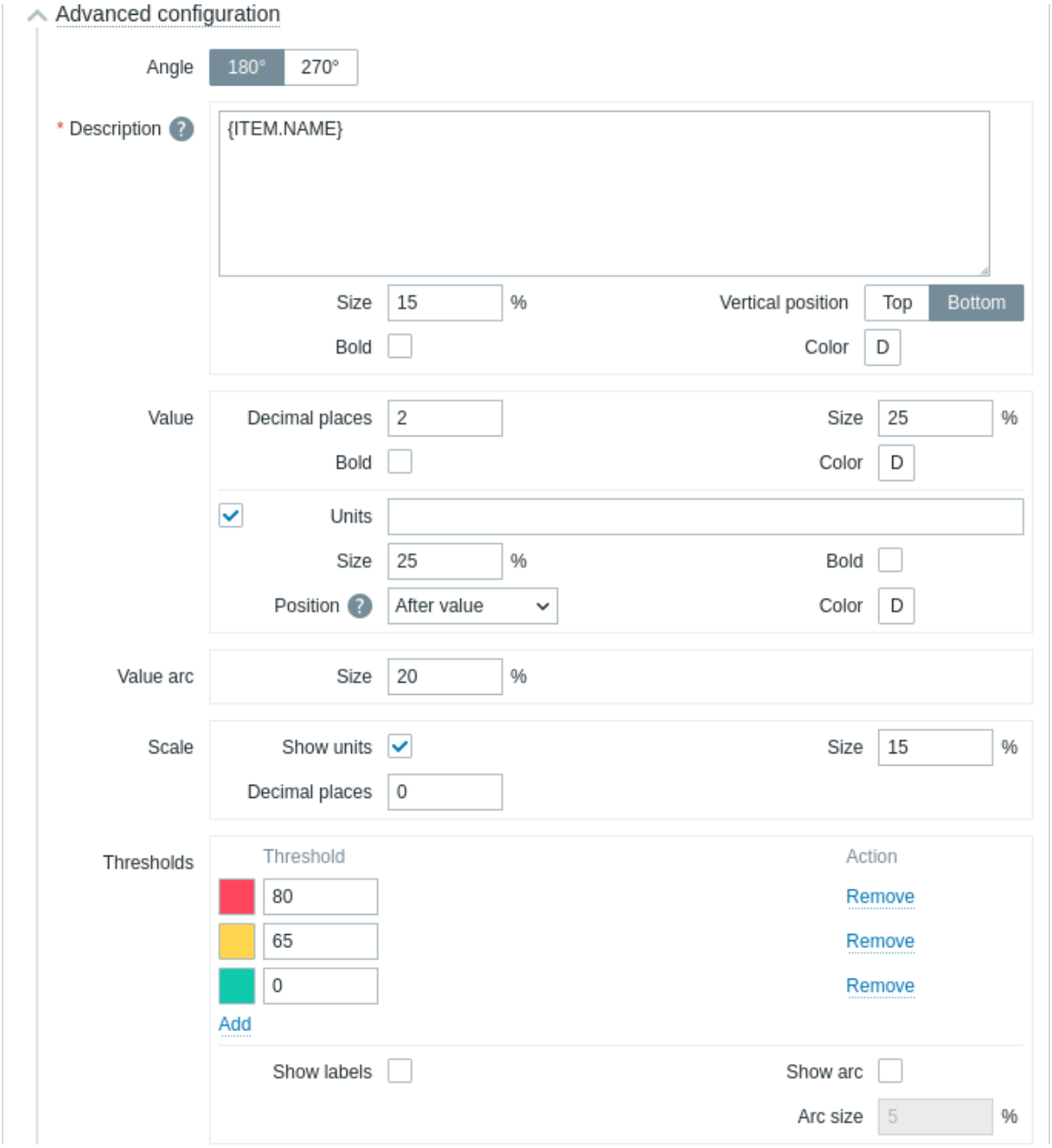
すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

Item	アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、アイテム名の入力を開始すると、一致するアイテムのドロップダウンが表示されます。 numeric データ（バイナリデータを除く）を返すアイテムのみ選択できることに注意してください。
Min	ゲージの最小値を入力します。 接尾辞（たとえば”1d”、”2w”、”4K”、”8G”）をサポートしています。値のマッピングをサポートしています。
Max	ゲージの最大値を入力します。 接尾辞（たとえば”1d”、”2w”、”4K”、”8G”）をサポートしています。値のマッピングをサポートしています。
Colors	カラーピッカーから色を選択します。 **Value arc** - ゲージの値アークの色を選択します。 **Arc background** - ゲージの値アークおよびゲージしきい値アークの背景色を選択します。 **Background** - ウィジェットの背景色を選択します。 ”D” はデフォルト色を表し、これは Web インターフェースのテーマに依存します。*Thresholds* が設定されている場合、**Value arc** のデフォルト色はしきい値の色に依存します。デフォルト色に戻すには、カラーピッカーの *Use default* ボタンをクリックします。
Show	対応するゲージ要素（説明、値、値アーク、針、目盛り（ゲージアークの開始位置と終了位置にあるゲージの最小値と最大値））を表示するには、チェックボックスをオンにします。非表示にするにはオフにします。少なくとも 1 つの要素を選択する必要があります。 ゲージの針と目盛りは、ゲージの値アークまたはゲージしきい値アーク（高度な設定オプションを参照）が表示されている場合に表示できます。また、ゲージの針が表示されている場合、値は針の下に配置されます。針が非表示の場合、値はゲージアークの下端に揃えて表示されます。

Override host	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードのhost selectorを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Advanced configuration	*Advanced configuration* ヘッダーをクリックすると、高度な設定オプションが表示されます。ここでは、*Show* フィールドで選択したゲージ要素を調整することもできます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用できます。



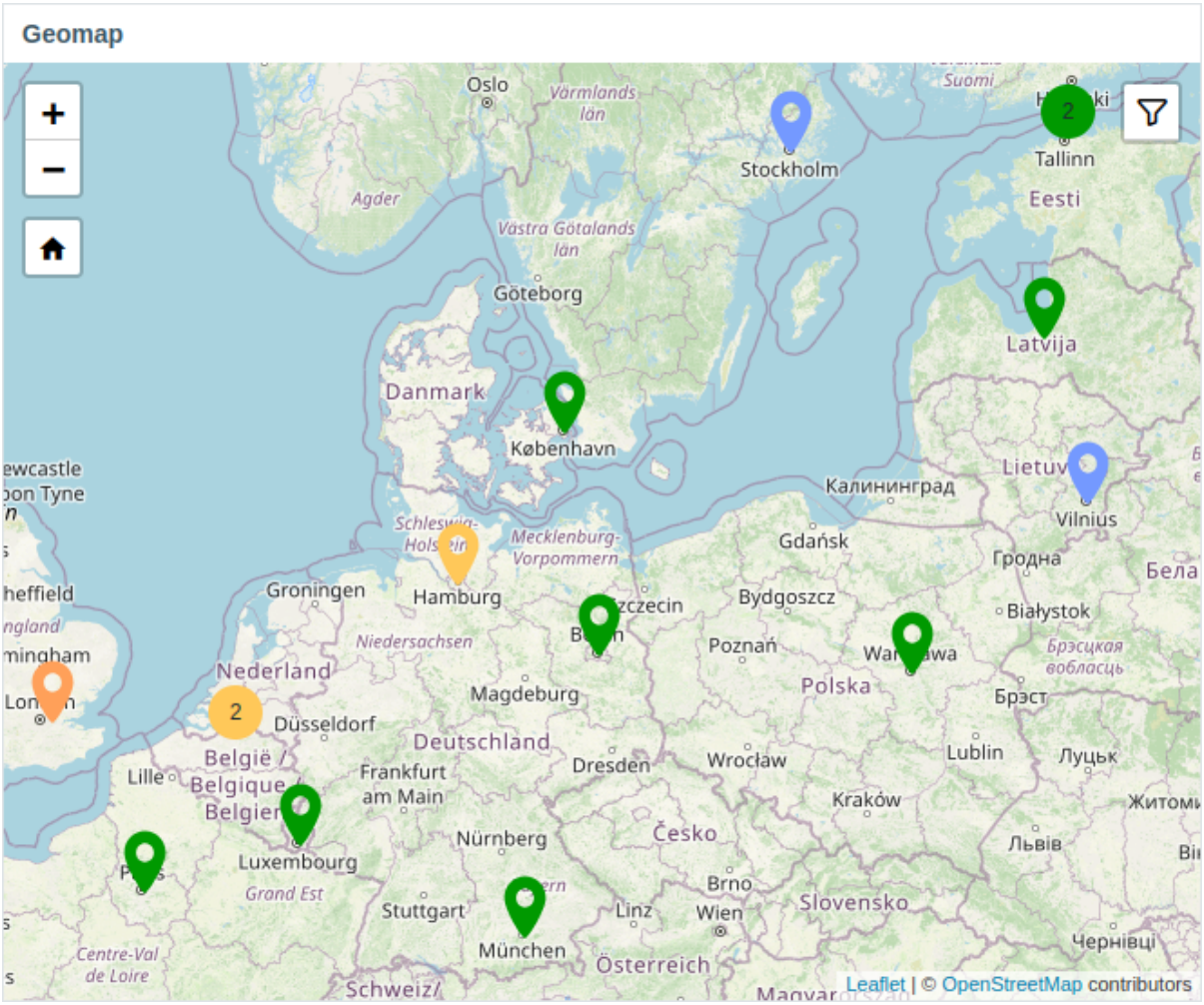
角度	ゲージの角度（180° または 270°）を選択します。
説明	
説明	アイテムの説明を入力します。この説明はデフォルトのアイテム名を上書きする場合があります。 複数行の説明がサポートされています。テキストとサポートされているマクロの組み合わせが可能です。 {HOST.*}、{ITEM.*}、{INVENTORY.*} およびユーザーマクロがサポートされています。
サイズ	アイテムの説明のフォントサイズの高さを入力します（パーセント、ウィジェット全体の高さに対する割合）。
垂直位置	アイテムの説明の垂直位置（ゲージアークに対して上または下）を選択します。

太字	チェックボックスをオンにすると、アイテムの説明が太字で表示されます。
色	カラーピッカーからアイテムの説明の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
値	
小数点以下の桁数	値とともに表示する小数点以下の桁数を入力します。 このオプションは、数値（浮動小数点）データを返すアイテムにのみ影響します。
サイズ	値のフォントサイズの高さを入力します（パーセント、ゲージアークの高さに対する割合）。
太字	チェックボックスをオンにすると、値が太字で表示されます。
色	カラーピッカーから値の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
単位	
単位	チェックボックスをオンにすると、アイテム値とともに単位が表示されます。 単位名を入力すると、アイテム設定で設定された単位を上書きします。
サイズ	アイテム単位のフォントサイズの高さを入力します（パーセント、ゲージアークの高さに対する割合）。
太字	チェックボックスをオンにすると、アイテム単位が太字で表示されます。
位置	アイテム単位の位置（値に対して上、下、前、後）を選択します。 このオプションは、次の時間関連の単位（unixtime、uptime、s）には無視されます。
色	カラーピッカーからアイテム単位の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
値アーク	
アークサイズ	ゲージ値アークのサイズの高さを入力します（パーセント、ゲージアーク半径に対する割合）。
針	
色	カラーピッカーからゲージ針の色を選択します。 ”D” はデフォルト色を表し、フロントエンドのテーマによって異なります。しきい値が設定されている場合、針のデフォルト色はしきい値の色に依存します。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
目盛り	
単位を表示	チェックボックスをオンにすると、ゲージの最小値と最大値に単位が表示されます。
サイズ	ゲージの最小値と最大値のフォントサイズの高さを入力します（パーセント、ゲージアークの高さに対する割合）。
小数点以下の桁数	ゲージの最小値と最大値とともに表示する小数点以下の桁数を入力します。 このオプションは、数値（浮動小数点）データを返すアイテムにのみ影響します。
しきい値	
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 しきい値リストは保存時に昇順でソートされます。 しきい値として設定された色は、数値アイテムに対してのみ正しく表示されることに注意してください。 サフィックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています。値のマッピングもサポートされています。
ラベルを表示	チェックボックスをオンにすると、しきい値がゲージ目盛りのラベルとして表示されます。
アークを表示	チェックボックスをオンにすると、ゲージしきい値アークが表示されます。
アークサイズ	ゲージしきい値アークのサイズの高さを入力します（パーセント、ゲージアーク半径に対する割合）。

ジオマップ

概要

Geomap ウィジェットは、インタラクティブな Leaflet ベースの地図上にホストをマーカーとして表示し、マーカーの色で各ホストの最も深刻な問題を示します。

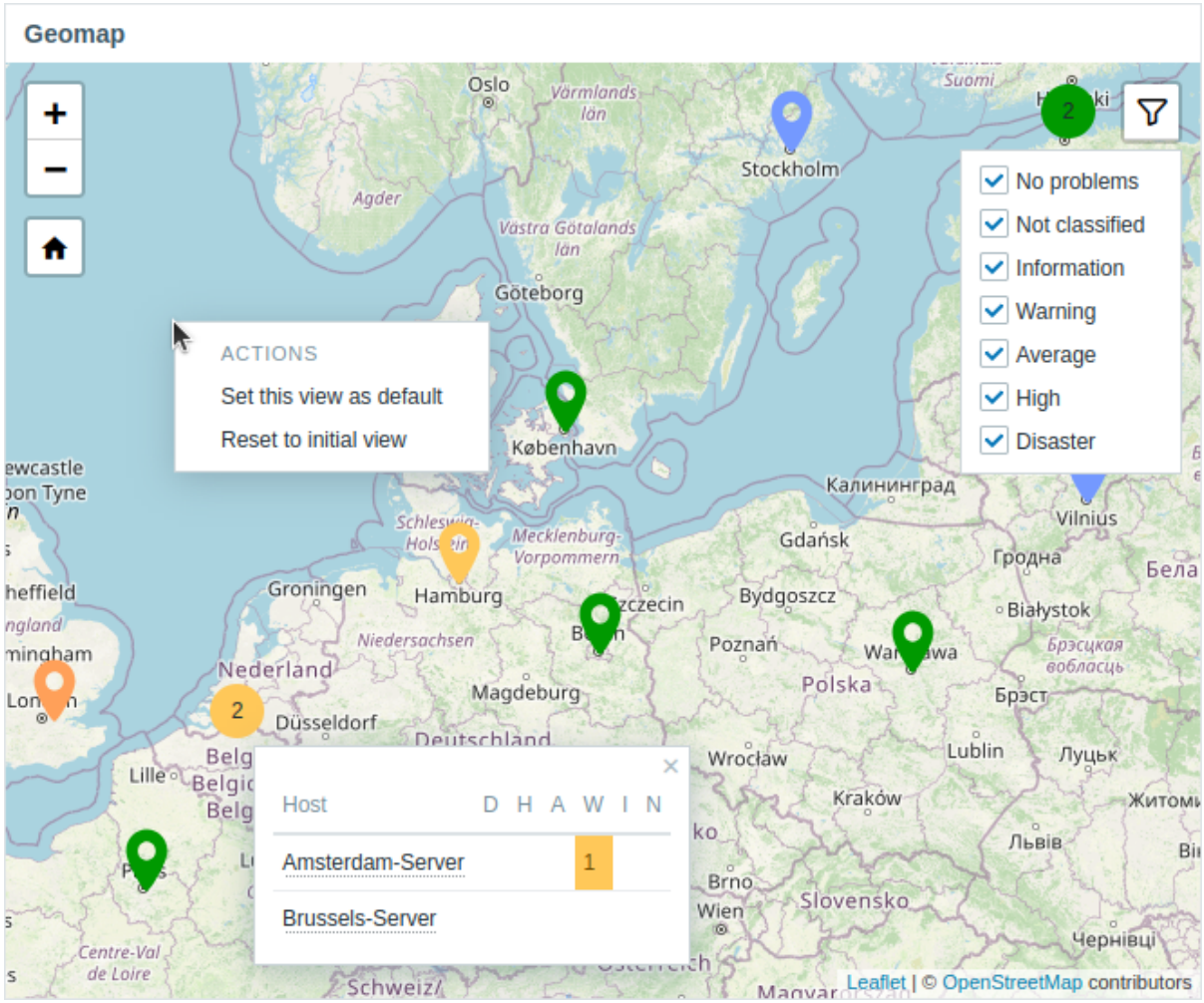


Note:
Zabbix は複数の定義済み地図タイルプロバイダーを提供しており、管理 > 一般 > 地理的地図でカスタムプロバイダーを追加できます。

デフォルトでは、ウィジェットはインベントリに有効な座標（緯度-90～90、経度-180～180）が設定されている有効なホストをすべて表示します。

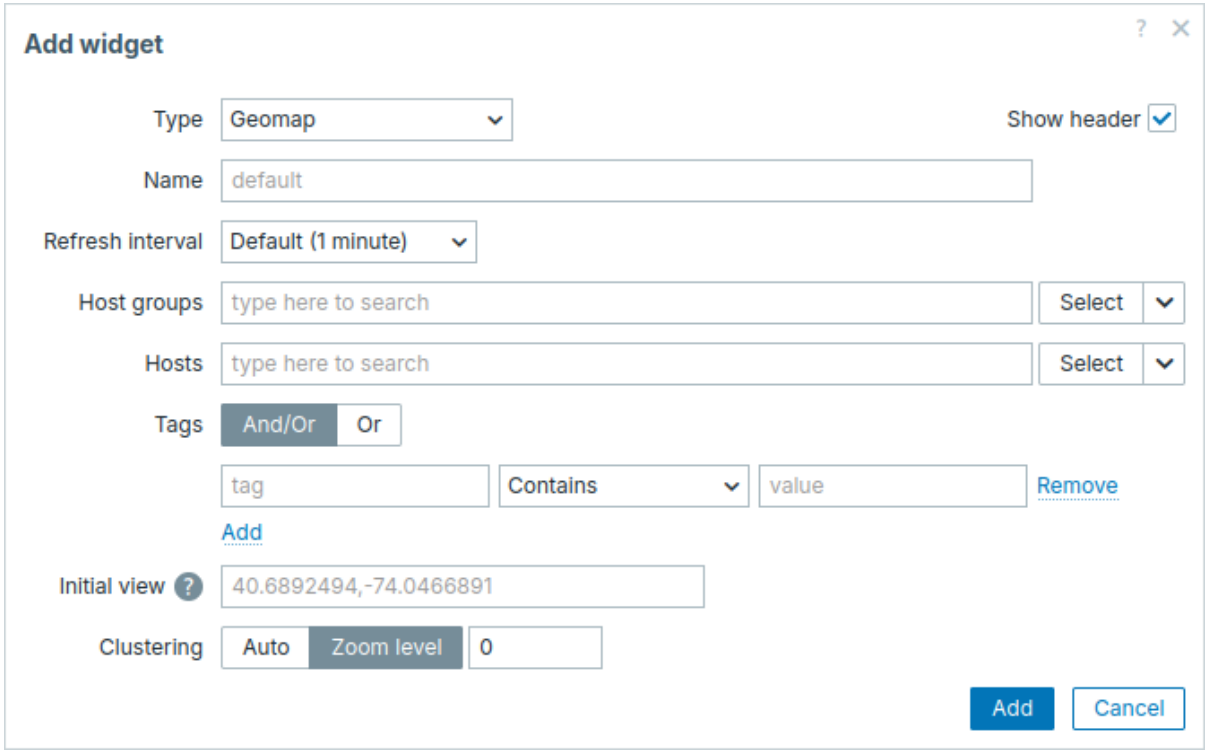
ウィジェットと対話するには、以下のオプションを使用します：

- マーカーをクリックするとホスト名と未解決の問題が表示され、ホスト名をクリックするとホストメニューが開きます。
- 右上のフィルターアイコンで問題の深刻度でホストをフィルタリングできます。
- 地図を右クリックしてデフォルトビューを設定またはリセットし、ホームアイコンでデフォルトビューに戻ります。



設定

ウィジェットを追加するには、タイプとして *Geomap* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

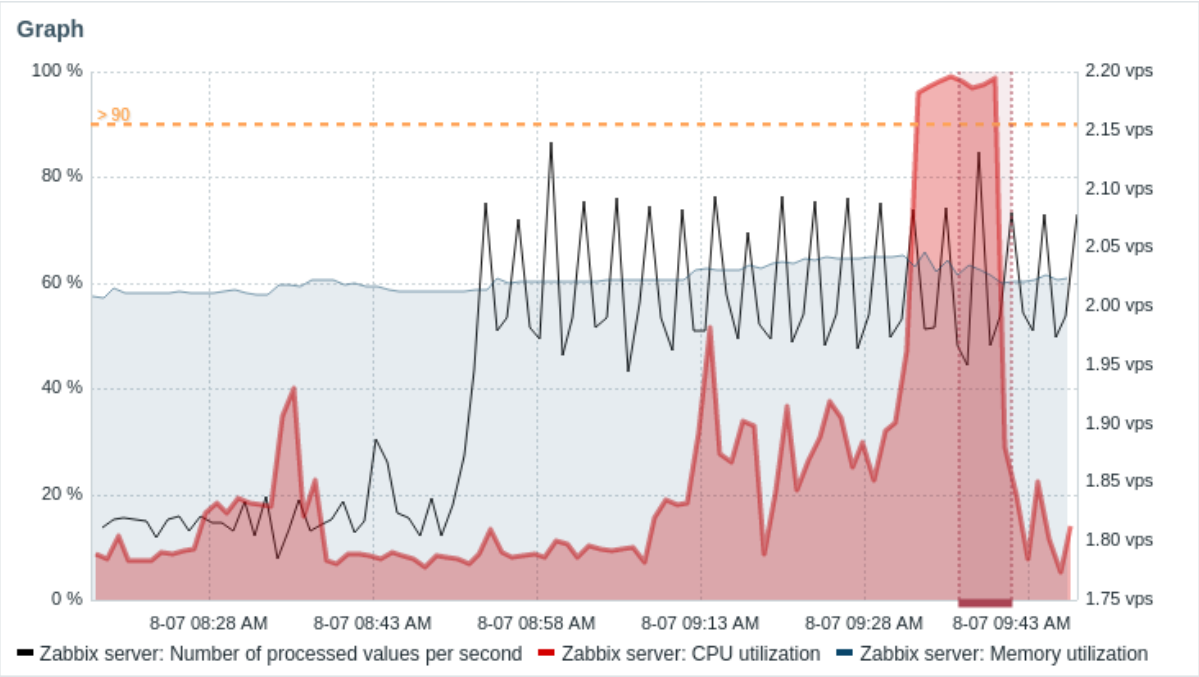
ホストグループ	マップに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 ホストグループフィールドとホストフィールドの両方で何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	マップに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストグループフィールドとホストフィールドの両方で何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
タグ	ウィジェットに表示されるホスト数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は、*Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
初期表示	ウィジェットの初回読み込み時に表示する中心座標と、必要に応じてズームレベルを、`<latitude>,<longitude>,<zoom>` 形式でカンマ区切りで指定します。 初期ズームが指定されている場合、Geomap ウィジェットは指定されたズームレベルで読み込まれます。指定されていない場合、初期ズームは、そのタイルプロバイダー固有のmax zoomの半分として計算されます。 デフォルト表示が設定されている場合、初期表示は無視されます（以下を参照）。 例: 40.6892494,-74.0466891 40.6892494,-122.0466891
クラスタリング	近接するホストマーカーを、件数付きの単一マーカーにどのようにまとめるかを指定します。 **Auto** - マーカーは自動的にまとめられます。 **Zoom level** - 指定したズームレベルに基づいてマーカーをまとめます（0 - 最大ズームアウト、30 - 最大ズームイン）。たとえば、このパラメータを 0 に設定すると、どのズームレベルでもマーカーはまとめられません。15 に設定すると、最大ズームアウトからズームレベル 15 までマーカーがまとめられます。30 に設定すると、すべてのズームレベルでマーカーがまとめられます。 最大ズームインレベルは、地図タイルプロバイダーによって決まることに注意してください。

グラフ

概要

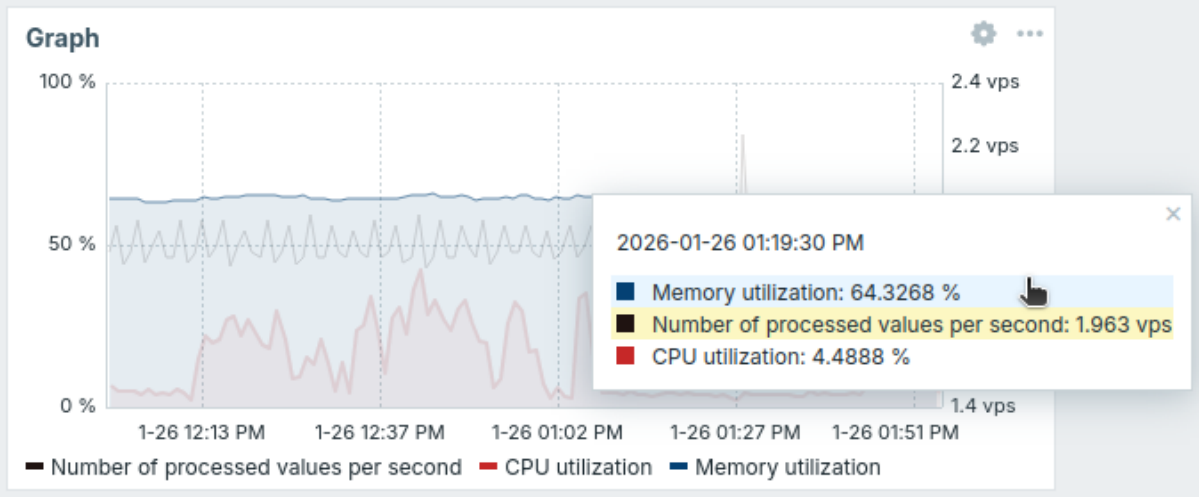
グラフウィジェットには、数値アイテムのデータがベクターベースのグラフとして表示されます。傾向の追跡、障害の発見、時間の

経過やホスト間での値の比較に役立ちます。

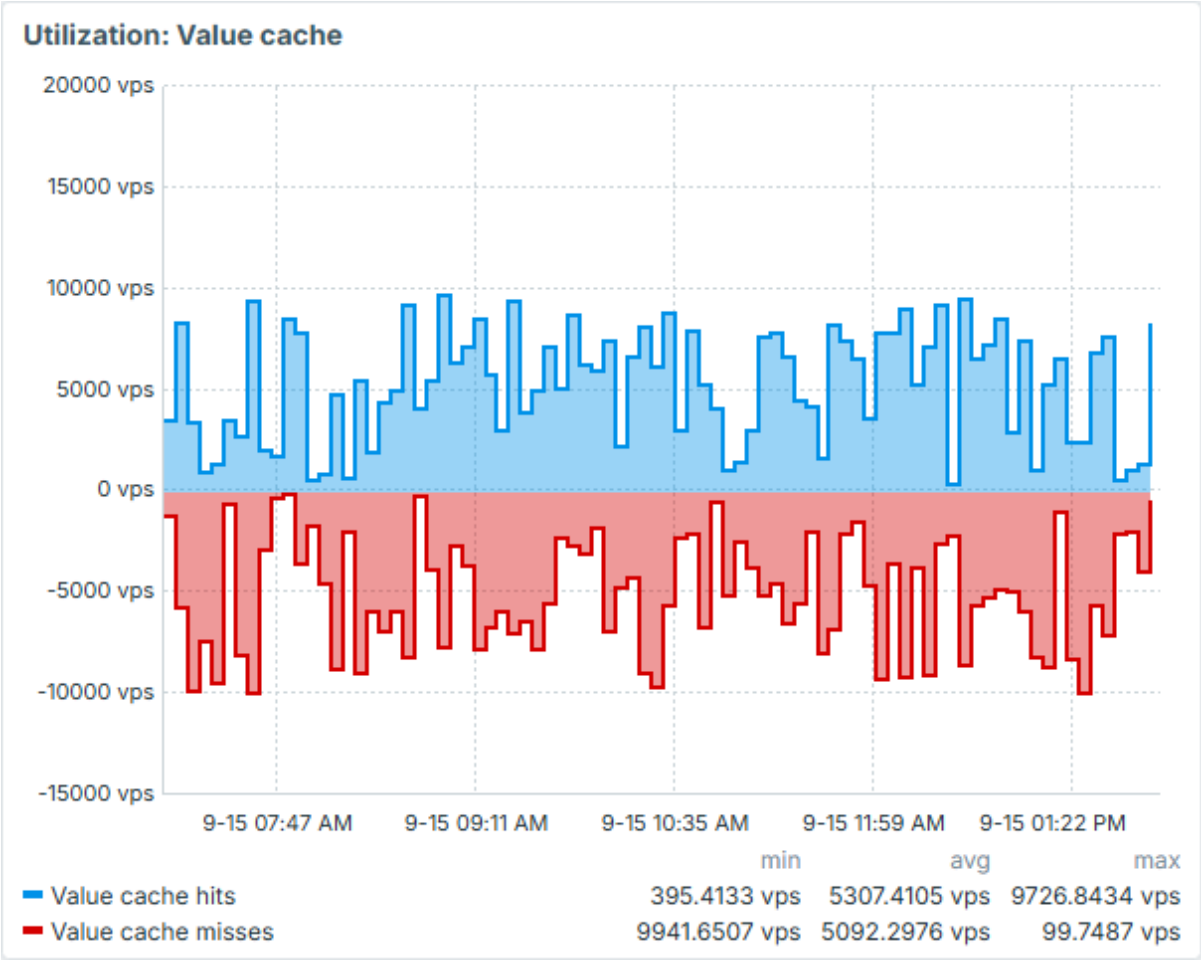


グラフにカーソルを合わせると、値の降順に並べ替えられたアイテムの一覧がツールチップに表示されます。

ツールチップ内のアイテムにカーソルを合わせると、そのグラフが強調表示されます（その他のグラフは薄く表示されます）。アイテムを選択すると、そのデータが他のウィジェットにブロードキャストされます。



グラフは、Y 軸の値を反転して表示することもできます。これは視覚的な表示にのみ影響し、元の値は変更しません。

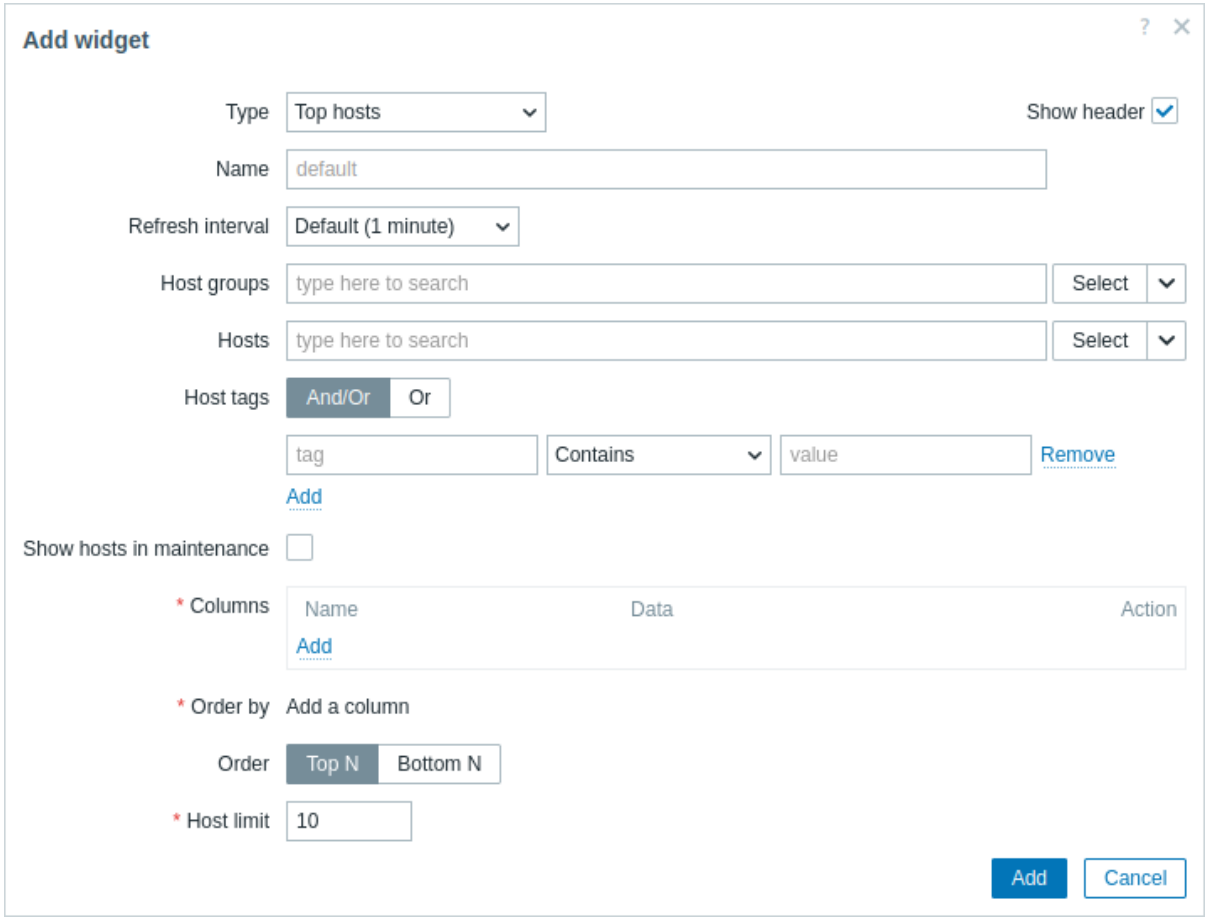


従来のグラフ（クラシック）ウィジェットも利用できます。

グラフウィジェットに表示される情報は、ウィジェットにカーソルを合わせると表示されるウィジェットのアクションメニュー ... から画像をダウンロードを選択することで、PNG 画像としてダウンロードできます。ブラウザーの設定に応じて、画像は自動的に Downloads フォルダーに保存されるか、保存先を選択できる保存ダイアログが開きます。

設定

設定するにはタイプとしてグラフを選択します。



データセット

データセットタブでは、データセットを追加してグラフに表示するデータを選択できます。追加できるデータセットには、次の 2 種類があります。

- アイテムパターン - 一致するアイテムのデータを表示します。単一のベースカラーを選択するか、パレットの行を選択して、一致した各アイテムに異なる色を割り当てることができます。
- アイテムリスト - 選択したアイテムのデータを表示します。ピッカーから各アイテムの色を個別に選択できます。

デフォルトでは、アイテムパターンデータセットが追加されます。

データセット	アイテムパターンデータセットの場合: ホストおよびアイテムのパターンを選択または入力します。これらのパターンに一致するアイテムのデータがグラフに表示されます。表示できるアイテムは最大 50 個です。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上の文字に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。 ワイルドカード記号は常に解釈されるため、たとえば *item2* や *item3* など、他にも一致するアイテムがある場合に、*item** という名前のアイテムだけを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストおよびアイテムのパターンを指定する必要があります。 詳細は、データセットの設定詳細を参照してください。 アイテムリストデータセットの場合: アイテムを追加ボタンをクリックして、グラフに表示するアイテムを選択します。 ウィジェットを追加ボタンをクリックして、アイテムのデータソース として互換性のあるウィジェットを選択することもできます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットを指定する必要があります。 詳細は、データセットの設定詳細を参照してください。 数値アイテムタイプのみ使用できます。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは使用できません。また、アイテムリストを指定するパラメータでは、テンプレートに設定されたアイテムのみ選択できます。

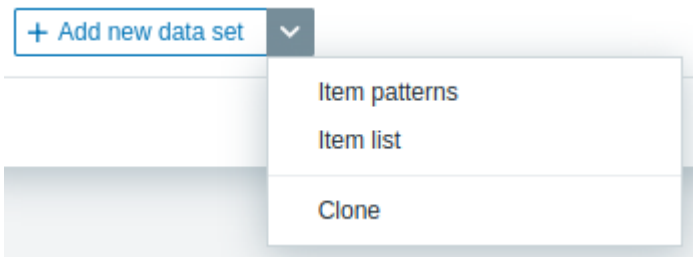
描画	メトリクスの描画タイプを選択します。 選択できる描画タイプは、線（デフォルト）、ポイント、階段、棒です。 線または階段グラフにデータポイントが 1 つしかない場合、描画タイプに関係なくポイントとして描画されます。ポイントサイズは線の幅から計算されますが、線の幅がそれより小さい場合でも 3 ピクセル未満にはなりません。
積み上げ	データを積み上げて表示する場合（塗りつぶされた領域を表示する場合）にチェックボックスをオンにします。 ポイント描画タイプを選択すると、このオプションは無効になります。
幅	線の幅を設定します。 線または 階段描画タイプを選択した場合に使用できます。
ポイントサイズ	ポイントのサイズを設定します。 ポイント描画タイプを選択した場合に使用できます。
透明度	透明度を設定します。
塗りつぶし	塗りつぶしレベルを設定します。 線または 階段描画タイプを選択した場合に使用できます。
欠損データ	欠損データの表示方法を選択します。 なし - 空白のままにします。 接続 - 2 つの境界値を接続します。 **0** として扱う - 欠損データを 0 として表示します。 最後の既知の値 - 欠損データを最後の既知の値と同じ値で表示します。ポイントおよび 棒描画タイプには適用されません。
ホストを上書き	ホストのデータソース として、互換性のあるウィジェットまたはダッシュボードのホストセレクター を選択します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータは使用できません。
Y 軸	Y 軸を表示するグラフの側を選択します。
値を反転	グラフに表示する前に、すべての Y 軸の値に-1 を乗算する場合にチェックボックスをオンにします。 これは視覚的な表示にのみ影響し、元のデータは変更されません。 このオプションはツールチップ、凡例、パーセンタイル線には影響しません。軸で シンプルトリガーオプションを選択した場合、トリガーのしきい値は反転して表示されます。
時間シフト	必要に応じて時間シフトを指定します。 このフィールドでは時間のサフィックスを使用できます。負の値も指定できます。
集計関数	選択した 集計間隔内で、各アイテムまたはデータセット全体に使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の数を表示します。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します。 使用しないを選択すると、すべての値が表示されます（集計なし）。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では、数値以外のデータが数値に変換されます。 詳細は、グラフでの集計を参照してください。
集計間隔	値を集計する間隔を指定します。 このフィールドでは時間のサフィックスを使用できます。サフィックスのない数値は秒として扱われます。 ウィジェットがトレンドに基づいて履歴データを表示するように設定されている場合（履歴データの選択が トレンドまたは 自動に設定されている場合）、1 時間の倍数となる集計間隔（例: 3600、60m、1h、3h など）を使用することを推奨します。トレンドには 1 時間単位で集計された値が保存されるため、1 時間の倍数ではない集計間隔（例: 100s、7min、15min、90min など）を使用すると、解釈が難しい結果になる場合があります。
集計対象	集計する対象を指定します。 各アイテム - データセット内の各アイテムを個別に集計して表示します。 データセット - データセット内のすべてのアイテムを集計し、1 つの値として表示します。

近似	グラフの縦方向の 1 ピクセルあたりに複数の値が存在する場合に表示する値を指定します。 **all** - 最小値、最大値、平均値を表示します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 この設定は、更新間隔が短く、長期間のグラフ（10 分ごとに収集した 1 年分の値など）を表示する場合に便利です。
データセットラベル	グラフの データセット設定およびグラフの 凡例（集計されたデータセットの場合）に表示するデータセットラベルを指定します。 指定した データセットラベルがあるデータセットも含め、すべてのデータセットに番号が付けられます。ラベルを指定しない場合、データセットには番号に基づいて自動的にラベルが付けられます（例: 「データセット #2」、「データセット #3」など）。データセットを並べ替えたりドラッグしたりすると、データセットの番号は再計算されます。 長すぎるデータセットラベルは、表示領域に収まるように短縮されます（例: 「Number of proc...」）。
アイテムタグ	ウィジェットに表示するアイテムをフィルターするタグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる、指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる、指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には次の 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

データセット設定の詳細

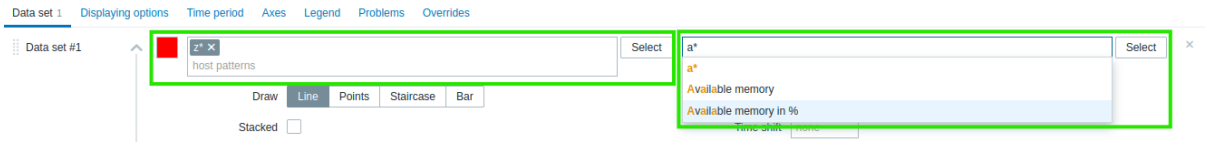
既存のデータセットが一覧に表示されます。次の操作を実行できます。

- 移動アイコンをクリックして、データセットを一覧内の新しい位置にドラッグします。
- 展開アイコンをクリックして、データセットの詳細を展開します。展開すると、このアイコンは 折りたたみアイコンに変わります。
- 色アイコンをクリックして、カラーピッカーを開きます。16 進数コードを入力するか、単色のスウォッチを選択できます。また、パレットタブに切り替えて、あらかじめ定義された色の行を選択することもできます。選択した色は、アイテム一覧データセットには直接適用され、アイテムパターンには生成される色合いの基準色として適用されます。Tab キーでダイアログのコントロール間を移動し、矢印キーでスウォッチまたはパレットの行を移動します。Enter キーで選択し、Esc キーでキャンセルします。
- 新しいデータセットを追加ボタンをクリックすると、ホストとアイテムパターンを選択できる空のデータセットが追加されます。新しいデータセットを追加ボタンの横にある下向きアイコンをクリックすると、ドロップダウンメニューが表示され、新しい アイテムパターンまたは アイテム一覧データセットを追加したり、現在開いているデータセットを 複製したりできます。すべてのデータセットが折りたたまれている場合、複製オプションは使用できません。

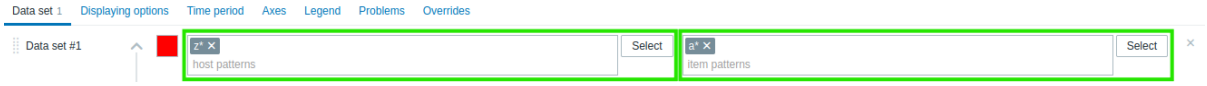


アイテムパターンデータセットには、ホストパターンとアイテムパターンのフィールドがあり、どちらもワイルドカード記号（*）を含む完全な名前またはパターンを認識します。この機能により、選択したパターンを含むすべてのホスト名とアイテム名を選択できます。アイテムパターンフィールドにアイテム名またはアイテムパターンを入力すると、選択したホスト名に属するアイテムのみがドロップダウンリストに表示されます。

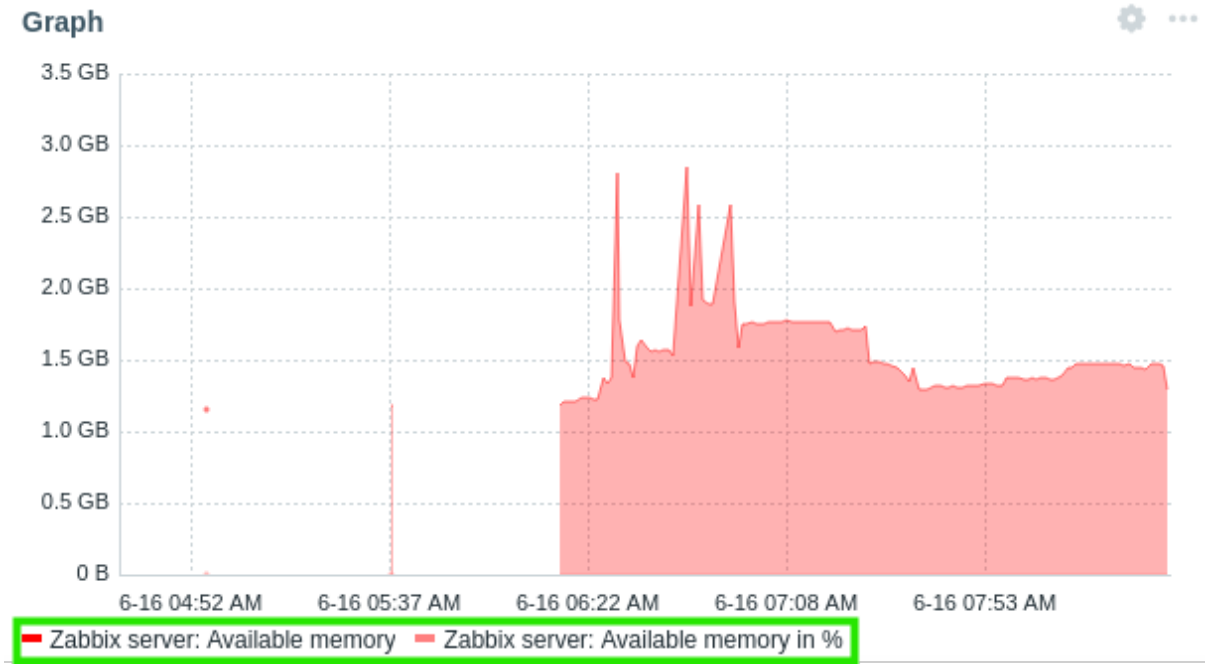
たとえば、ホストパターンフィールドにパターン **z*** を入力すると、ドロップダウンリストにはこのパターンを含むすべてのホスト名が表示されます：**z*, Zabbix server, Zabbix proxy**。*Enter* を押すと、このパターンが受け入れられ、**z*** として表示されます。同様に、アイテムパターンフィールドにパターン **a*** を入力すると、ドロップダウンリストにはこのパターンを含むすべてのアイテム名が表示されます：**a*, Available memory, Available memory in %**。



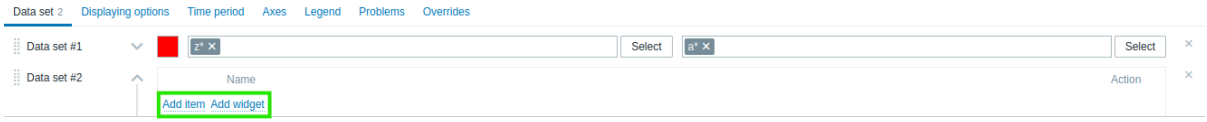
Enter を押すと、パターンが受け入れられ、**a*** として表示されます。



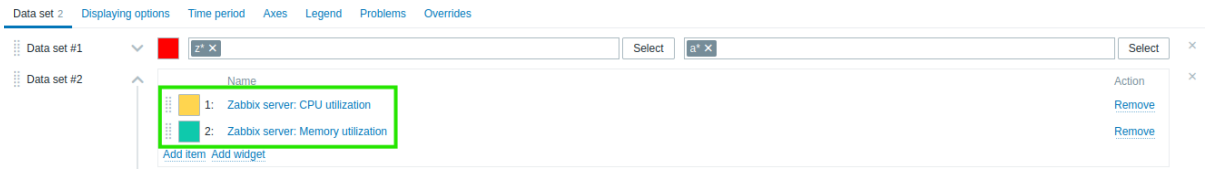
その後、グラフには選択したホスト名に属するすべてのアイテムが表示されます。



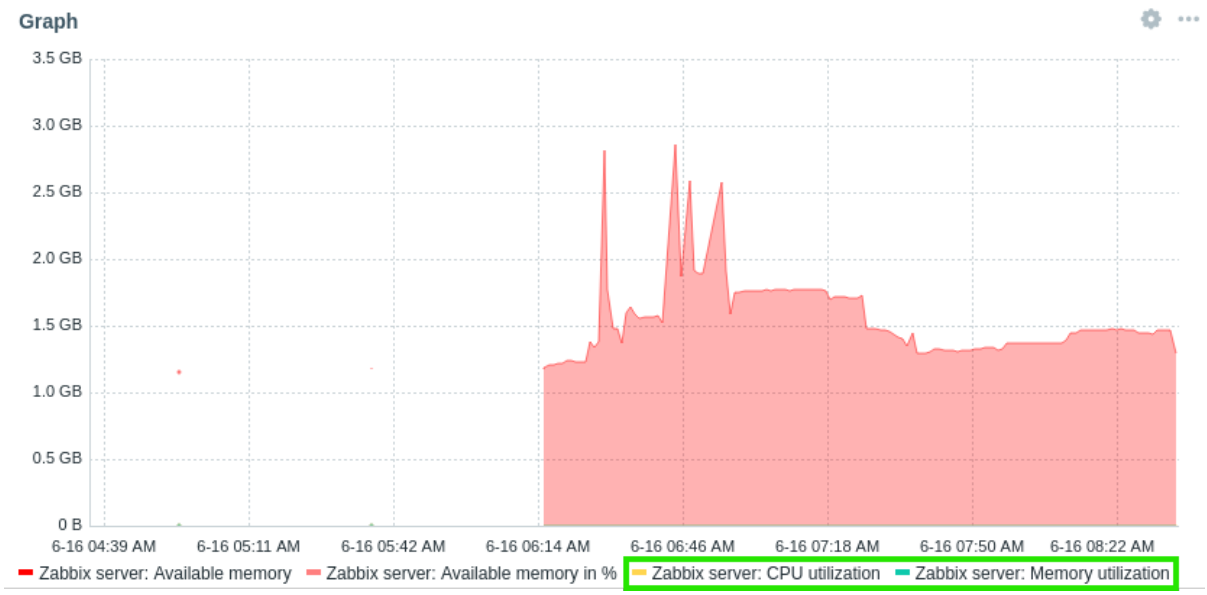
アイテムリストデータセットには、グラフに表示するアイテムを追加できるアイテムの追加ボタンがあります。また、ウィジェットの追加ボタンをクリックすることで、アイテムのデータソースとして互換性のあるウィジェットを追加することもできます。



例えば、アイテムの追加ボタンをクリックすると、ホストパラメータを含むポップアップウィンドウが開きます。ホストを選択すると、そのホストの選択可能なすべてのアイテムがリストに表示されます。



1 つまたは複数のアイテムを選択すると、それらがデータセットのアイテムリストとグラフに表示されます。

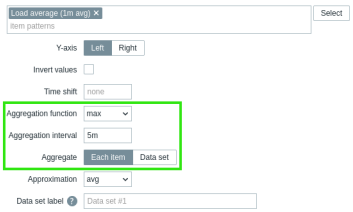


グラフでの集計

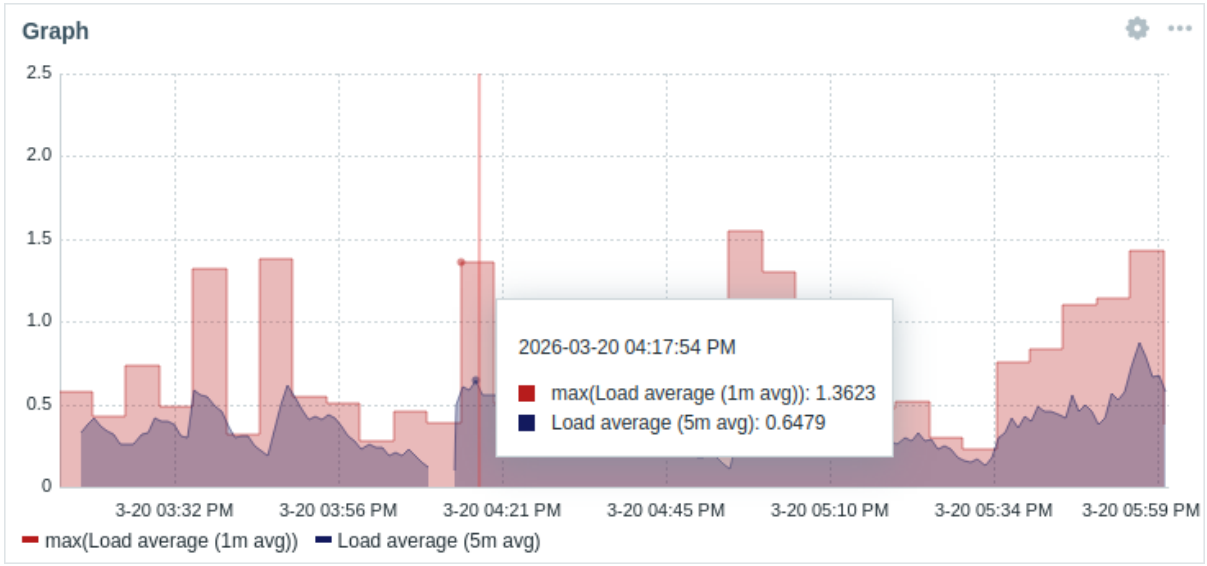
集計を使用すると、すべての個別の値ではなく、指定した間隔（5 分、1 時間、1 日など）に対する集計値（最小値、最大値、平均値など）を表示できます。データセット内では、各アイテムに個別に集計を適用することも、データセット全体に適用することもできます。

集計機能の使用例は次のとおりです。- 1 日あたりの NGINX サーバーへの平均リクエスト数。- 1 週間あたりのクラスタ間で最も少ないディスク容量。

次の例では、5 分間の集計間隔で、アイテムに「max」集計機能を適用しています。



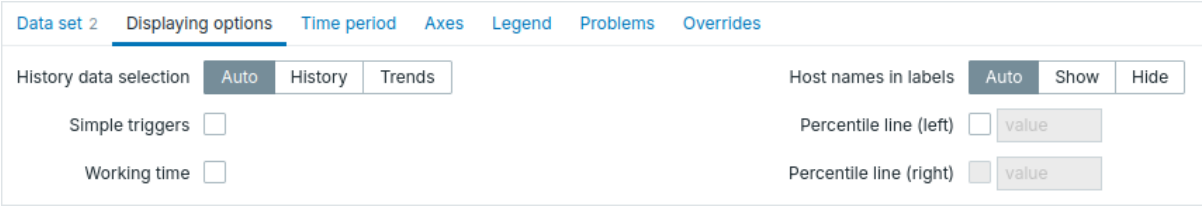
この例では、5 分間の最大 CPU 負荷（1 分間の平均値から集計）を、5 分間の平均 CPU 負荷と比較しています。



凡例で集計機能を表示を選択すると、アイテムは括弧内に表示され、凡例とツールチップでは集計機能が先頭に付加されます。グラフにデータセットラベルを指定すると、ラベルは括弧内に表示され、集計機能が先頭に付加されます。

表示オプション

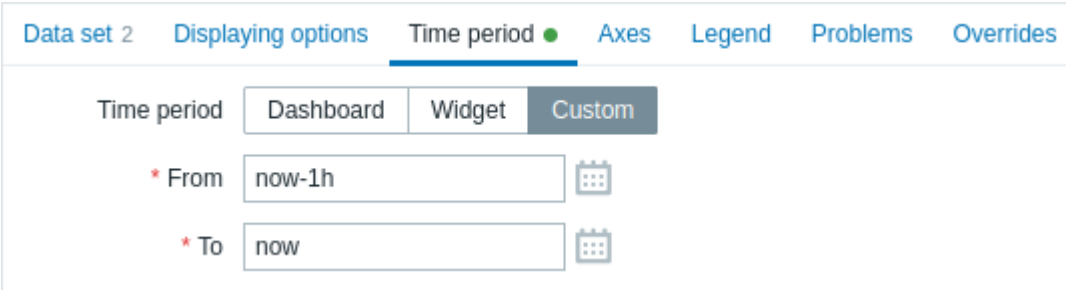
表示オプションタブでは、履歴データの選択を定義できます。



履歴データの選択	グラフデータのソースを設定します。 自動 - クラシックグラフのアルゴリズムに従ってデータを取得します（デフォルト）。 履歴 - 履歴からデータを取得します。 トレンド - トレンドからデータを取得します。
シンプルトリガー	シンプルトリガーのトリガーしきい値を表示するには、チェックボックスをオンにします。しきい値は、トリガーの深刻度の色を使用した破線で描画されます。 シンプルトリガーとは、式内の 1 つのアイテムに対して 1 つの関数（`last`、`max`、`min`、`avg` のみ）を使用するトリガーです。 描画できるトリガーは最大 3 つです。トリガーが描画範囲内にない場合、表示されないことに注意してください。
稼働時間	グラフに稼働時間を表示するには、チェックボックスをオンにします。稼働時間（稼働日）はグラフ上で白い背景として表示され、非稼働時間はグレーで表示されます（*Original blue* のデフォルトの Web インターフェーステーマを使用した場合）。
ラベル内のホスト名	グラフの凡例およびグラフデータ上にマウスを合わせたときに表示されるツールチップに、ホスト名を表示するかどうかを選択します。 自動 - データセット全体に複数のホストが存在する場合にのみ、ホスト名を表示します（デフォルト）。 表示 - ホスト名を表示します。 非表示 - ホスト名を非表示にします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
パーセンタイルライン（左）	チェックボックスをオンにしてパーセンタイル値を入力すると、指定したパーセンタイルをグラフの左 Y 軸上のラインとして表示できます。 たとえば、95% のパーセンタイルを設定すると、値の 95 パーセントがその値を下回るレベルにパーセンタイルラインが表示されます。
パーセンタイルライン（右）	チェックボックスをオンにしてパーセンタイル値を入力すると、指定したパーセンタイルをグラフの右 Y 軸上のラインとして表示できます。 たとえば、95% のパーセンタイルを設定すると、値の 95 パーセントがその値を下回るレベルにパーセンタイルラインが表示されます。

時間範囲

時間範囲タブでは、グラフにデータを表示する時間範囲を設定できます。

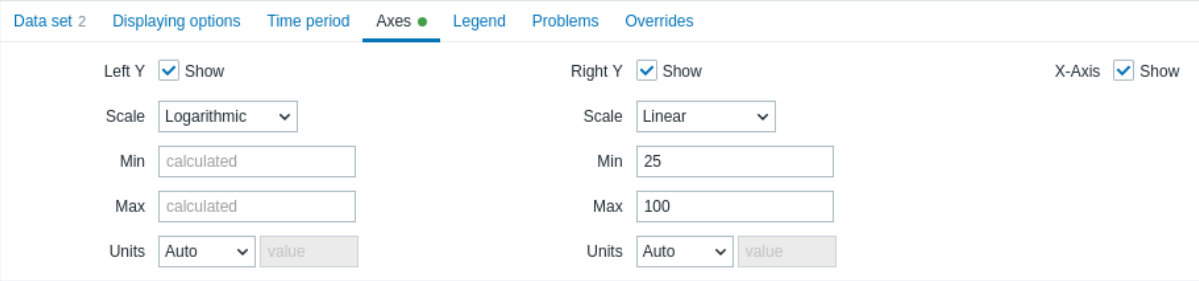


時間範囲	時間範囲のdata sourceを選択します。 ダッシュボード - ダッシュボードの時間範囲セレクターを使用します。 ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します。 カスタム - 開始および終了パラメータで設定したカスタムの時間範囲を使用します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。 ウィジェットの時間範囲設定に関係なく、互換性のあるウィジェットでは、時間範囲のデータソースとして引き続き使用できることに注意してください。
ウィジェット	時間範囲のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 時間範囲が「ウィジェット」に設定されている場合に使用できます。

開始	時間範囲の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 時間範囲が「カスタム」に設定されている場合に使用できます。
終了	時間範囲の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 時間範囲が「カスタム」に設定されている場合に使用できます。

軸

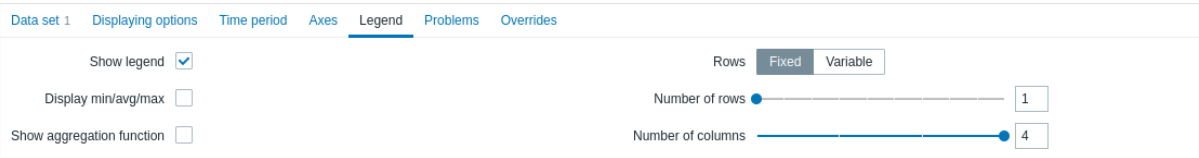
軸タブでは、軸の表示方法をカスタマイズできます。



左 Y	このチェックボックスを選択すると、左 Y 軸が表示されます。 データセットタブまたはオーバーライドタブで選択されていない場合、このチェックボックスは無効になることがあります。
右 Y	このチェックボックスを選択すると、右 Y 軸が表示されます。 データセットタブまたはオーバーライドタブで選択されていない場合、このチェックボックスは無効になることがあります。
X 軸	このチェックボックスの選択を解除すると、X 軸が非表示になります（デフォルトでは選択されています）。
スケール	ドロップダウンからグラフ軸の値のスケールを選択します。 線形 - 軸の値が一定の量（例: 10、20、30）ずつ増加します。値が安定して変化するデータや、小規模から中規模の範囲を対象とするデータに適しています。 対数 - 軸の値が指数関数的に増加します（例: 10、100、1000）。急速に変化するデータや、広い範囲を対象とするデータに適しています。
最小	対応する軸の最小値を設定します。 Y 軸の表示範囲の最小値を指定します。
最大	対応する軸の最大値を設定します。 Y 軸の表示範囲の最大値を指定します。
単位	ドロップダウンからグラフ軸の値の単位を選択します。 自動 - データセット内の最初のアイテムの単位を使用して軸の値を表示します。 静的 - 値入力フィールドで指定した単位を使用して軸の値を表示します。フィールドを空白のままにすると、数値のみが表示されます。

凡例

凡例タブでは、グラフの凡例をカスタマイズできます。

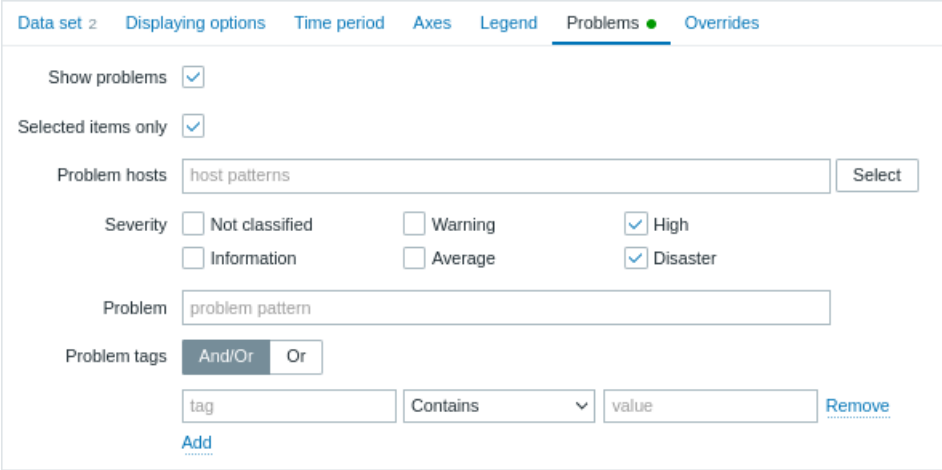


凡例を表示	このチェックボックスの選択を解除すると、グラフ上の凡例が非表示になります（デフォルトでは選択されています）。
最小/平均/最大を表示	このチェックボックスを選択すると、アイテムの最小値、平均値、最大値が凡例に表示されます。
集計関数を表示	このチェックボックスを選択すると、凡例に集計関数が表示されます。
行	凡例の行の表示モードを選択します。 固定 - 表示される行数は、行数パラメータの値によって決まります。 可変 - 表示される行数は、設定されているアイテム数によって決まります。ただし、最大行数パラメータの値を超えることはありません。

行数/ 最大行数	行を「固定」に設定した場合、表示する凡例の行数（1～10）を設定します。 行を「可変」に設定した場合、表示する凡例の最大行数（1～10）を設定します。
列数	表示する凡例の列数（1～4）を設定します。 このパラメータは、最小/平均/最大を表示が選択されていない場合に使用できます。

障害

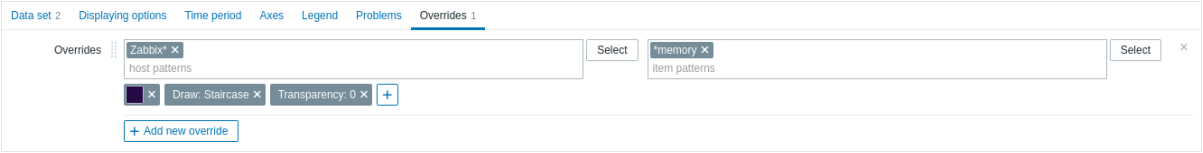
障害タブでは、障害の表示をカスタマイズできます。



障害を表示	このチェックボックスをオンにすると、グラフ上で障害が表示されます（デフォルトではオフ、つまり無効です）。
選択したアイテムのみ	このチェックボックスをオンにすると、選択したアイテムの障害のみがグラフに表示されます。
障害のホスト	グラフに表示する障害のホストを選択します。 ワイルドカードパターンを使用できます（たとえば、* は 0 文字以上の文字に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。 入力中は、一致するすべてのホストがドロップダウンに表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
深刻度	グラフに表示する障害をフィルターするため、障害の深刻度を選択します。 深刻度を 1 つも選択しない場合、すべての障害が表示されます。
障害	グラフに表示する障害の名前を指定します。
障害タグ	ウィジェットに表示する障害の数を制限するため、障害タグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる、指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる、指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - 1 つの条件を満たせば十分です。

オーバーライド

オーバーライドタブでは、データセットにカスタムオーバーライドを追加できます。



オーバーライドは、* ワイルドカードを使用してデータセットに複数のアイテムを選択し、アイテムのデフォルトの表示方法（デフォルトの基本色やその他のプロパティなど）を変更したい場合に便利です。

既存のオーバーライドがある場合は、一覧に表示されます。新しいオーバーライドを追加するには、次の手順を実行します。

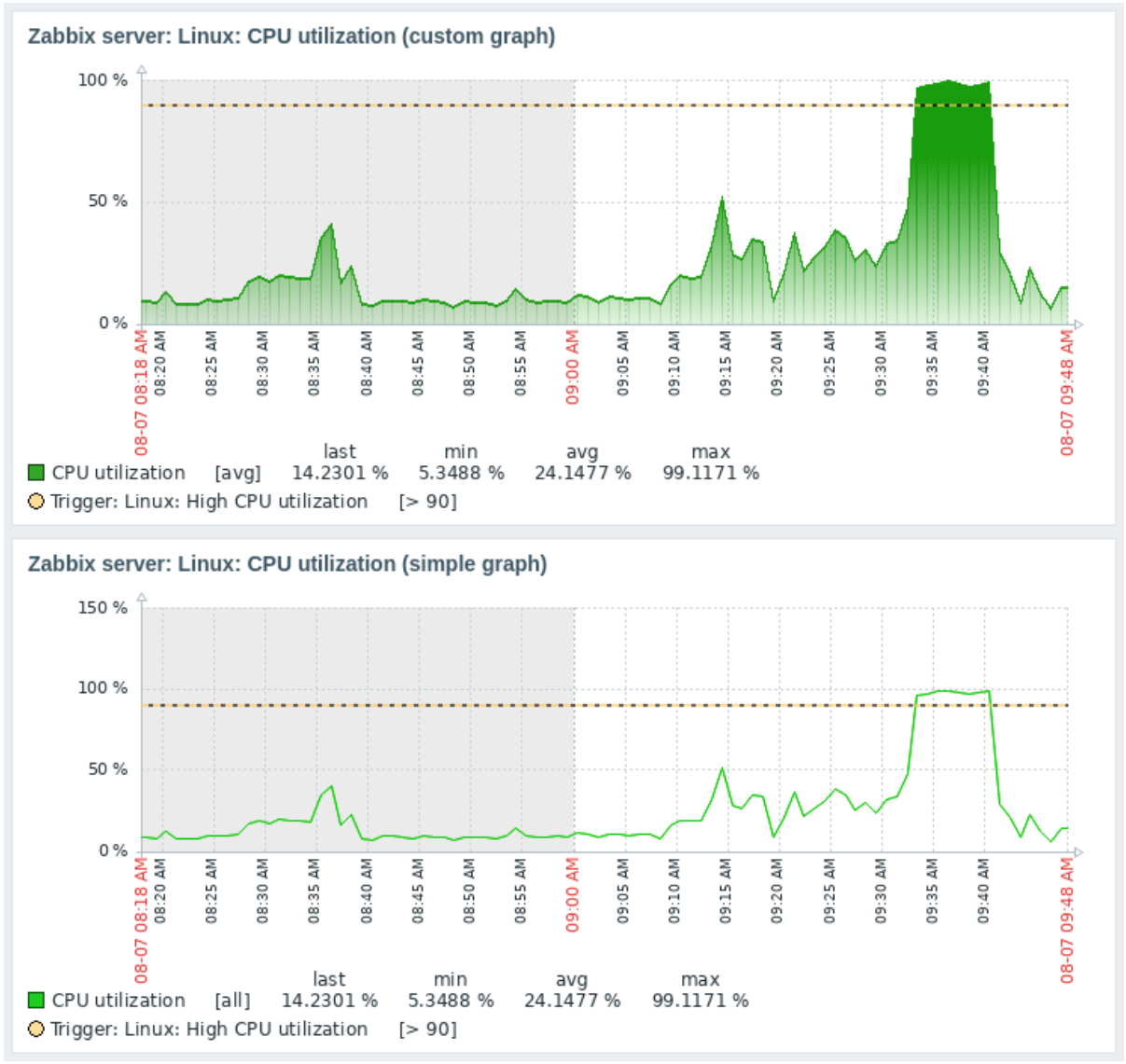
- 

ボタンをクリックします。
- オーバーライドの対象となるホストとアイテムを選択します。または、ホストとアイテムのパターンを入力することもできます。ワイルドカードパターンを使用できます（たとえば、* は 0 文字以上の文字列に一致する結果を返します）。ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* キーを押します。入力中は、一致するすべてのホストがドロップダウンに表示されることに注意してください。ワイルドカード記号は常に解釈されるため、たとえば、他にも一致するアイテム（item2、item3 など）がある場合に、”item*” という名前のアイテムだけを個別に追加することはできません。ホストパターンとアイテムパターンのパラメータは必須です。テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは使用できません。テンプレートダッシュボードでウィジェットを設定する場合、アイテムリストを指定するパラメータでは、テンプレートに設定されたアイテムのみ選択できます。
- 
をクリックして、オーバーライドパラメータを選択します。少なくとも 1 つのオーバーライドパラメータを選択する必要があります。パラメータの説明については、上記のデータセットタブを参照してください。

グラフ（クラシック）

概要

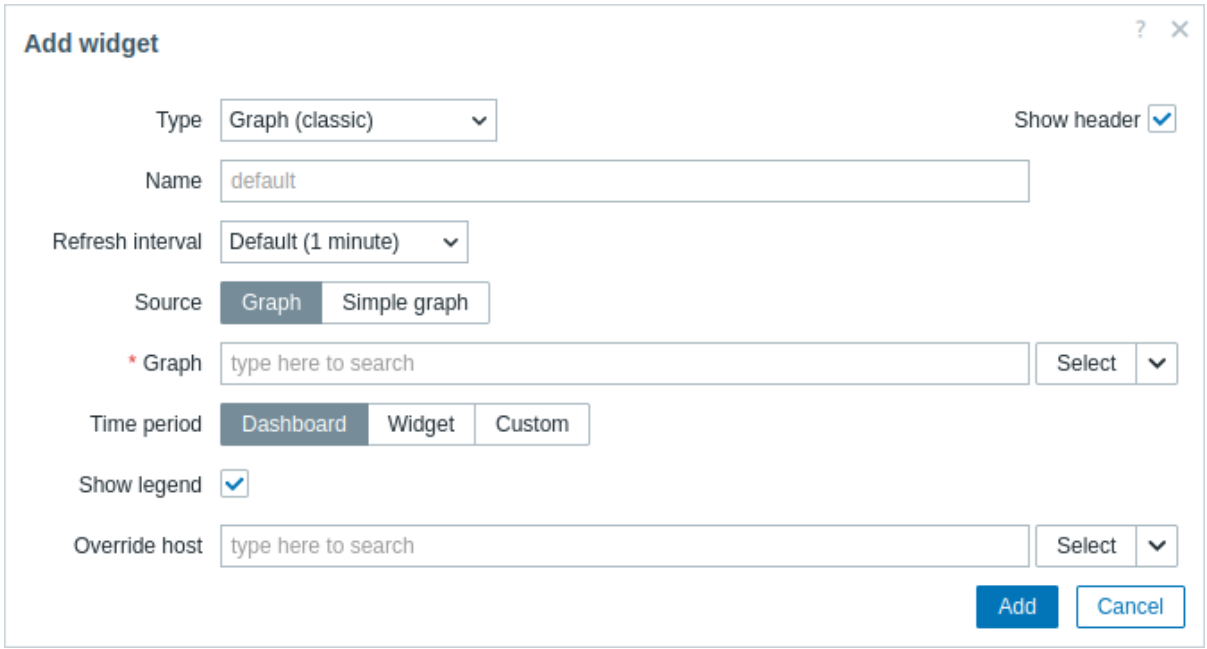
グラフ（クラシック）ウィジェットは、数値アイテムのデータを画像ベースのカスタムグラフまたはシンプルグラフとして表示します。期間の選択、トリガーライン、凡例の表示（オプション）に対応しています。



グラフ（クラシック）ウィジェットに表示される情報は、ウィジェット上にカーソルを合わせると表示されるウィジェットのアクションメニュー ... から画像をダウンロードを選択することで、PNG 画像としてダウンロードできます。ブラウザーの設定に応じて、画像は自動的に `Downloads` フォルダーに保存されるか、保存場所を選択できる保存ダイアログが開きます。

設定

設定するには、タイプとしてグラフ（クラシック）を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ソース	グラフタイプを選択します。 グラフ - カスタムグラフ; シンプルグラフ - シンプルグラフ。
グラフ	表示するカスタムグラフを選択します。 または、グラフのデータソースとして互換性のあるウィジェットを選択します。 このパラメータはソースが「グラフ」に設定されている場合に利用できます。
アイテム	シンプルグラフで表示するアイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 このパラメータはソースが「シンプルグラフ」に設定されている場合に利用できます。
期間	グラフにデータを表示する期間を設定します。期間のデータソースを選択します。 ダッシュボード - ダッシュボードの期間セレクターを使用します。 ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します。 カスタム - *From* および *To* パラメータで設定したカスタム期間を使用します。設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された期間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
凡例を表示	このチェックボックスのチェックを外すと、グラフの凡例が非表示になります（デフォルトでチェックされています）。
ホストを上書き	互換性のあるウィジェットまたはダッシュボードのホストセレクターを、ホストのデータソースとして選択します。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。

グラフの凡例は 3 つのセクションで構成されています。

- アイテムとその集計値
- パーセンタイル（設定されている場合）
- トリガー（表示されているアイテムに関連付けられている場合）

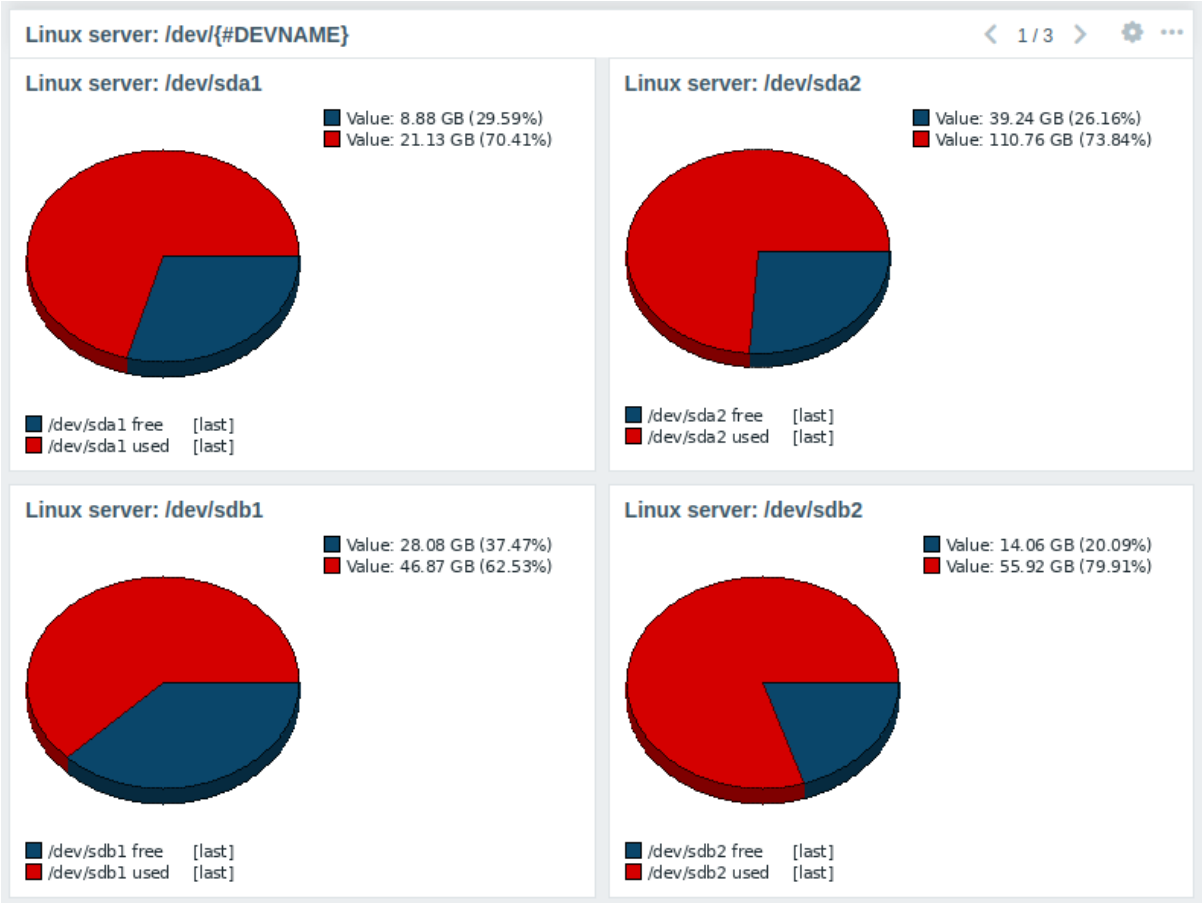
ウィジェット内のグラフの高さが不十分な場合、凡例が表示されないか、部分的にしか表示されない場合があります。最初にトリガーとパーセンタイルが非表示になり、次にアイテムの凡例が非表示になります。完全な凡例を表示するには、ウィジェットの縦サイズを大きくしてください。

Note:
トリガーラインは最大 3 本まで表示できます。トリガーがそれ以上ある場合は、重大度の低いトリガーが優先的に表示されます。

グラフのプロトタイプ

概要

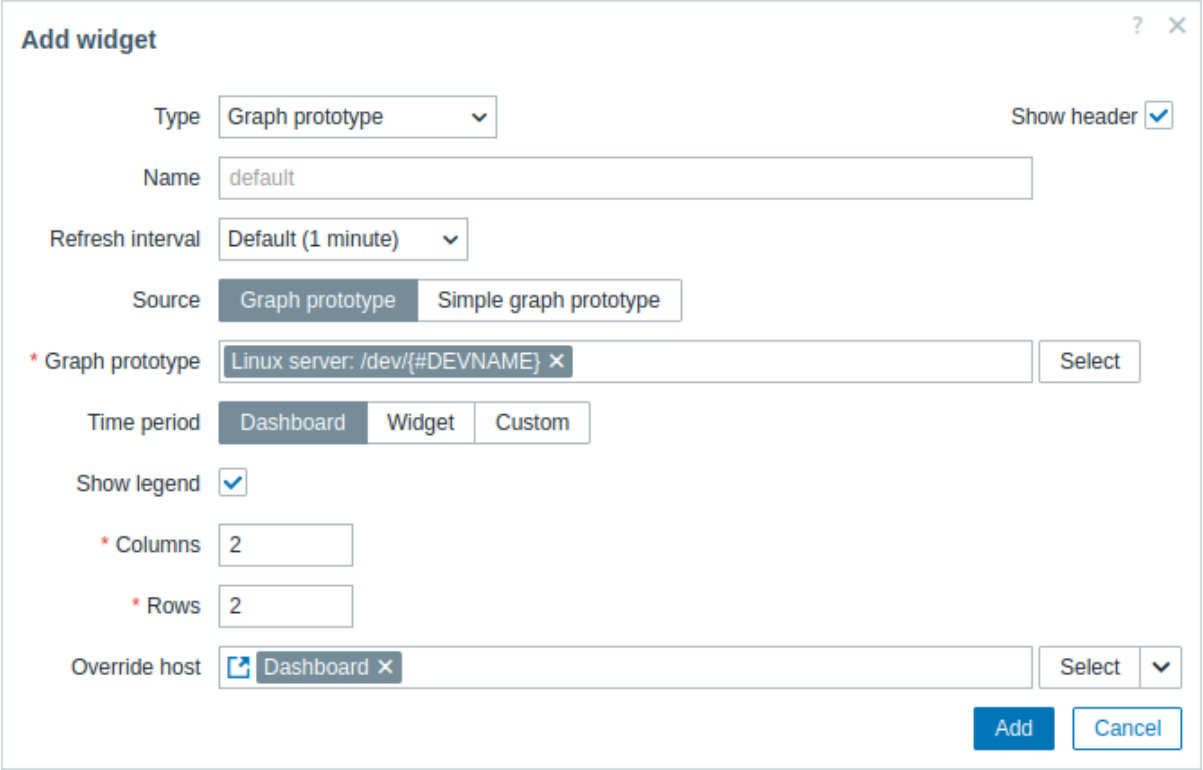
グラフプロトタイプウィジェットは、ローレベルディスカバリールールのグラフプロトタイプまたはアイテムプロトタイプに基づいて自動的に作成されたカスタムグラフのグリッドを表示します。



グラフは設定可能なグリッドレイアウトで配置されます。グラフの数がレイアウトを超える場合、ウィジェットでページングが有効になります。

設定

設定するには、ウィジェットのタイプとしてグラフのプロトタイプを選択します。



すべてのウィジェットに共通するパラメータに加えて、次の固有のオプションを設定できます。

ソース	グラフのソースとして、グラフのプロトタイプまたはシンプルグラフのプロトタイプを選択します。
グラフのプロトタイプ	グラフのプロトタイプによって検出されたグラフを表示するグラフのプロトタイプを選択します。 このパラメータは、ソースに「グラフのプロトタイプ」を設定した場合に使用できます。
アイテムのプロトタイプ	アイテムのプロトタイプによって検出されたアイテムのシンプルグラフを表示するアイテムのプロトタイプを選択します。 このパラメータは、ソースに「シンプルグラフのプロトタイプ」を設定した場合に使用できます。
期間	グラフにデータを表示する期間を設定します。期間のデータソースを選択します。 ダッシュボード - ダッシュボードの期間セレクターを使用します。 ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します。 カスタム - 開始および終了パラメータで設定したカスタム期間を使用します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻を示します。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットでは、期間のデータソースとして引き続き使用できます。
ウィジェット	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 このパラメータは、期間に「ウィジェット」を設定した場合に使用できます。
開始	期間の開始を入力または選択します。 相対時間の構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間に「カスタム」を設定した場合に使用できます。
終了	期間の終了を入力または選択します。 相対時間の構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間に「カスタム」を設定した場合に使用できます。
凡例を表示	グラフの凡例を非表示にするには、このチェックボックスの選択を解除します（デフォルトでは選択されています）。
ホストを上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
列	グラフのプロトタイプウィジェット内に表示するグラフの列数を入力します。
行	グラフのプロトタイプウィジェット内に表示するグラフの行数を入力します。

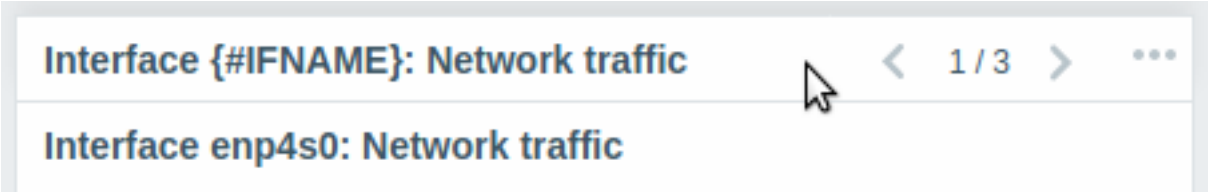
グラフの凡例は、次の 3 つのセクションで構成されます。

- アイテムとその集計値
- パーセンタイル（設定されている場合）
- トリガー（表示されているアイテムに関連付けられている場合）

ウィジェット内のグラフの高さが不十分な場合、凡例が表示されないか、一部のみ表示されることがあります。
最初にトリガーとパーセンタイルが非表示になり、次にアイテムの凡例が非表示になります。
凡例全体を表示するには、ウィジェットの縦方向のサイズを大きくします。

Note:
表示できるトリガーの行は 3 行までです。
トリガーがそれより多い場合は、重要度の低いトリガーが優先して表示されます。

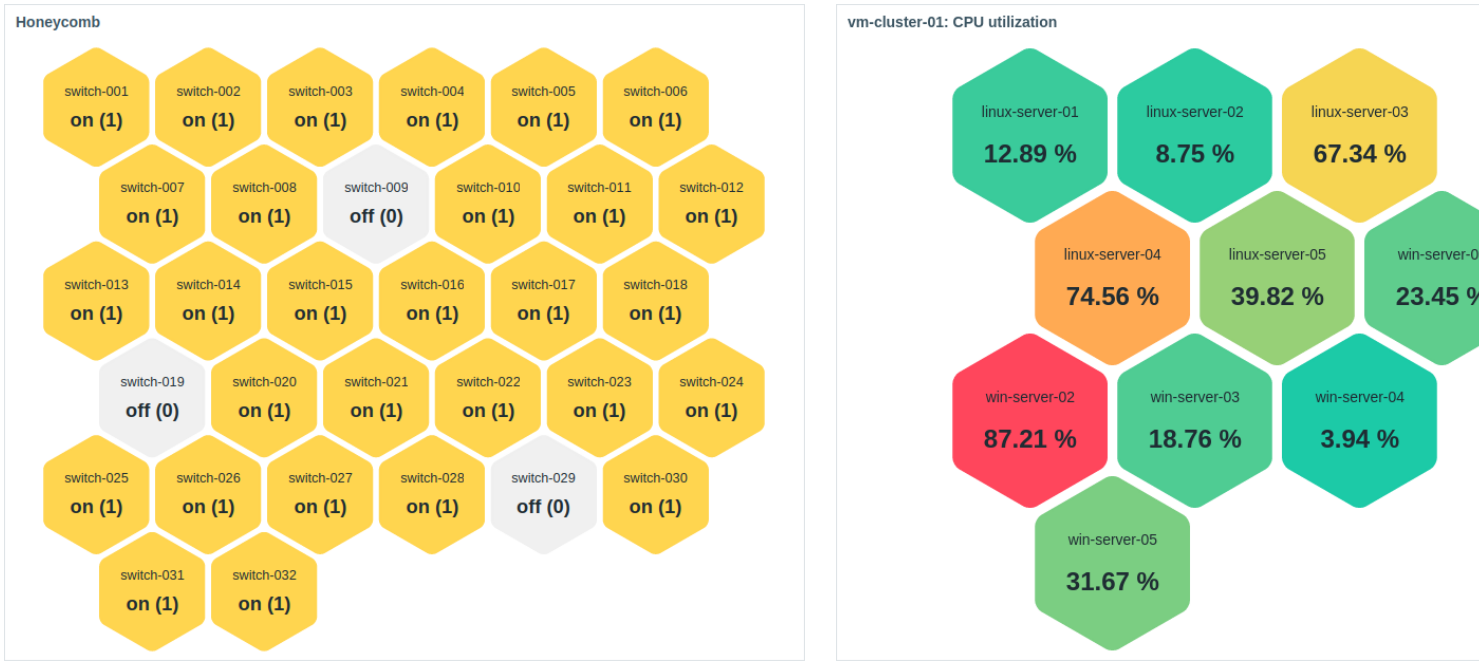
列および行パラメータを使用すると、ウィジェット内に複数のグラフを配置できますが、ウィジェットの列数または行数を超える数の検出されたグラフが存在する場合があります。
この場合、ウィジェットでページングが使用可能になり、スライドアップヘッダーに表示される左右の矢印を使用してページを切り替えられます。



Honeycomb

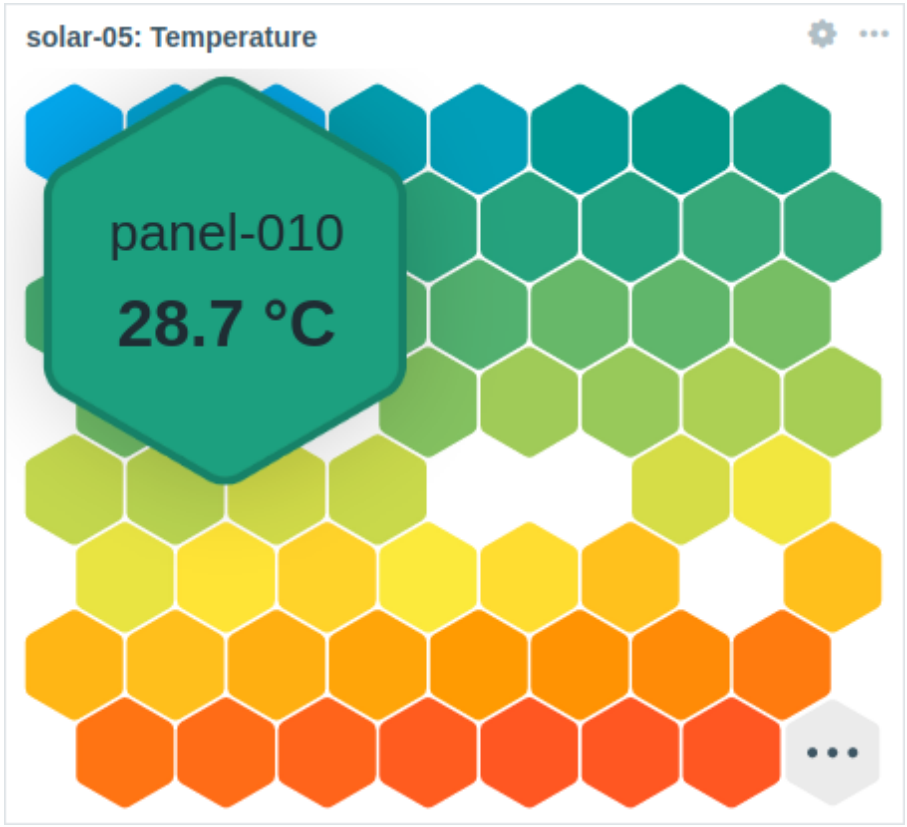
概要

Honeycomb ウィジェットは、ホストまたはアイテムをグリッド内の六角形セルとして表示します。このレイアウトにより、障害が発生しているホストやアイテムの検出、問題の集中箇所の把握、グループ間の比較などを簡単に行えます。



表示されるセルの数は、ウィジェットのサイズとセルの最小サイズ（32px）によって制限されます。すべてのセルを収められない場合、最後のセルに省略記号が表示されます。ウィジェットのサイズを変更すると、セルの数、サイズ、配置が動的に調整されます。

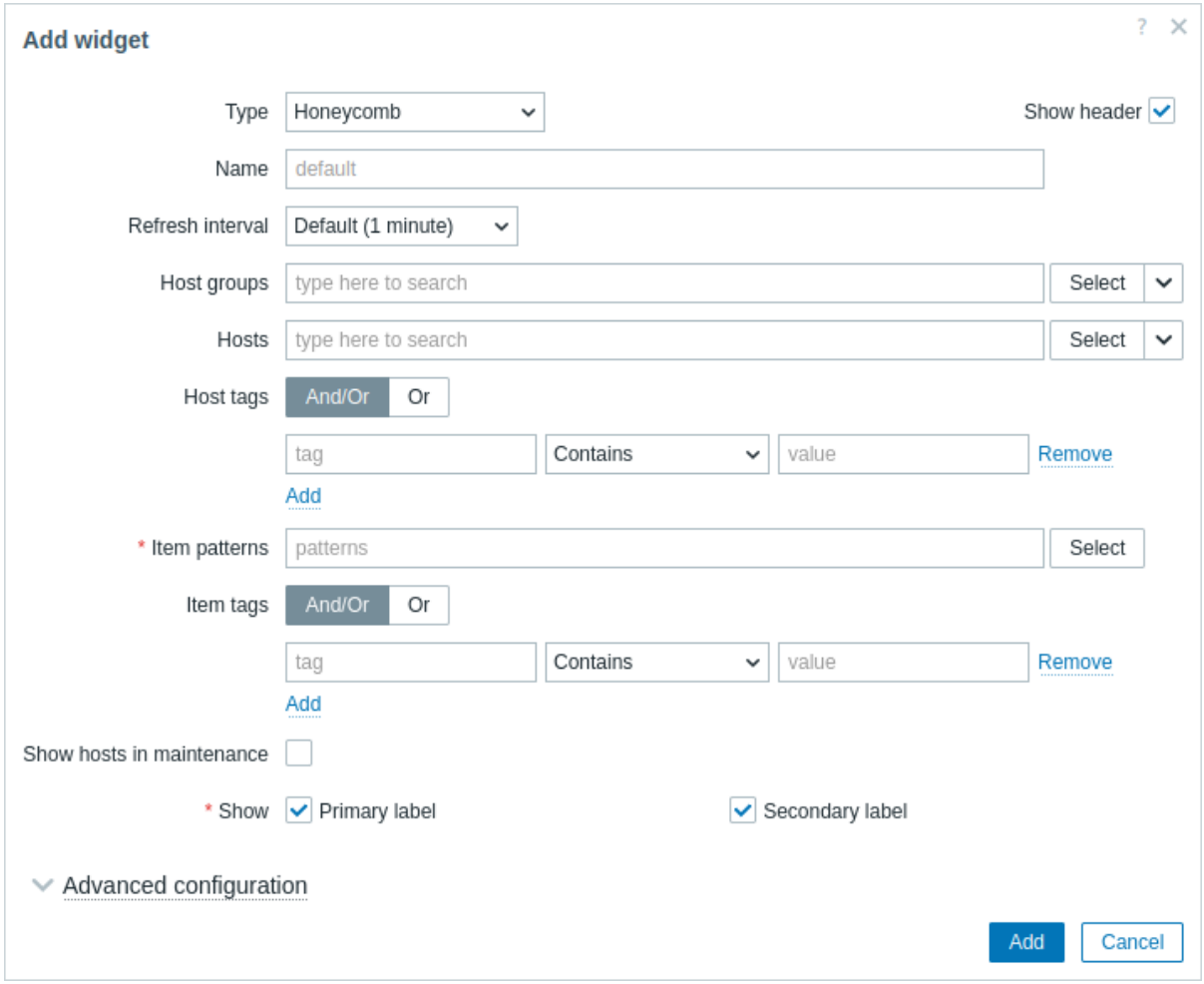
マウスオーバーすると、フォーカスされたセルが拡大され、見やすくなります。セルをクリックすると、その境界線が強調表示され、別のセルを選択するまで維持されます。



Honeycomb ウィジェットに表示される情報は、ウィジェット上にマウスを移動すると表示されるウィジェットのアクションメニュー ... から画像をダウンロードを選択することで、PNG 画像としてダウンロードできます。ブラウザーの設定に応じて、画像は自動的に `Downloads` フォルダーに保存されるか、保存場所を選択できる保存ダイアログが開きます。

設定

設定するには、タイプとして *Honeycomb* を選択します。



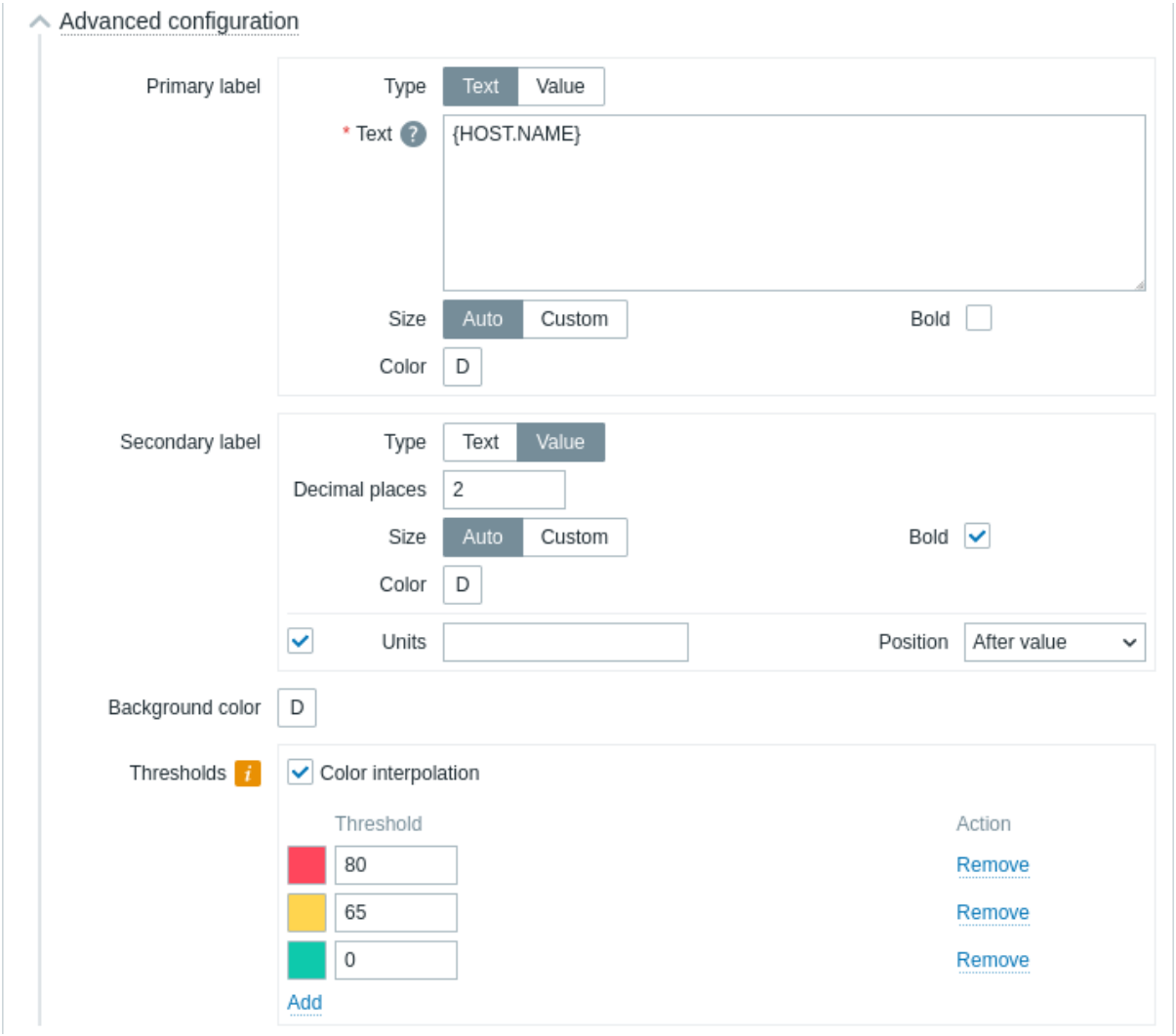
すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、ネストされたすべてのホストグループも暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットには、選択した アイテムパターンに一致するアイテムを持つホストを含むすべてのホストグループが表示されます（以下を参照）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットには、選択した *Item pattern* に一致するアイテムを持つすべてのホストが表示されます（以下を参照）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

ホストタグ	ウィジェットに表示するホスト数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。 タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
アイテムパターン	アイテムパターンを入力するか、既存のアイテムをアイテムパターンとして選択します。入力または選択したパターンに一致するアイテムのデータが honeycomb に表示されます。アイテムパターンパラメータは必須です。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致するアイテムを返し、Zabbix* は”Zabbix” で始まるアイテムを返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力を開始すると、選択した ホスト、または選択した ホストグループ内のホストに属するアイテムに限定された一致アイテムのドロップダウンリストが表示されます。ワイルドカード記号は常に解釈されるため、たとえば *item** という名前のアイテムを、他に一致するアイテム（例: item2、item3）が存在する場合に個別に追加することはできません。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータではテンプレート上で設定されたアイテムのみ選択できます。
アイテムタグ	ウィジェットに表示するアイテム数を制限するためのタグを指定します。詳細については、上記のホストタグを参照してください。
メンテナンス中のホストを表示	このチェックボックスをオンにすると、メンテナンス中のホストを表示します（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータのラベルは メンテナンス中のデータを表示になります。
表示	このチェックボックスをオンにすると、対応する honeycomb セル要素（主ラベル、副ラベル）を表示します。 少なくとも 1 つの要素を選択する必要があります。
高度な設定	高度な設定ヘッダーをクリックすると、高度な設定オプションが表示されます。

高度な設定

高度な設定オプションは、折りたたみ可能な 高度な設定セクションで利用できます。また、表示フィールドで選択された要素（上記参照）およびハニカムセルの背景色またはしきい値に対してのみ使用できます。



主/副ラベル

タイプ
ラベルのタイプを選択します:
テキスト - ラベルには テキストパラメータで指定した文字列が表示されます。
値 - ラベルには 小数点以下の桁数パラメータで指定した小数点以下の桁数でアイテムの値が表示されます。

テキスト
ラベルのテキストを入力します。このテキストはデフォルトのアイテム名を上書きする場合があります。
複数行のテキストをサポートしています。テキストとサポートされているマクロの組み合わせも可能です。
{HOST.*}、{ITEM.*}、{INVENTORY.*}、およびユーザーマクロをサポートしています。
ハニカムセルはホスト名のアルファベット順に並べられ、各ホスト内ではアイテム名順に並べられます。
このパラメータは、タイプが テキストに設定されている場合に使用できます。

小数点以下の桁数
値とともに表示する小数点以下の桁数を入力します。
このパラメータは、タイプが 値に設定されている場合に使用でき、numeric (float) データを返すアイテムにのみ影響します。

サイズ
ラベルサイズを選択します:
自動 - 自動調整されたラベルサイズを使用します。
カスタム - カスタムのラベルサイズを入力します（ハニカムセルサイズに対する割合）。
ハニカムセルサイズに収まらないラベルは切り詰められることに注意してください。

太字
ラベルを太字で表示するには、チェックボックスをオンにします。

色
カラーピッカーからラベルの色を選択します。
「D」はデフォルト色を表し、これは Web インターフェースのテーマによって異なります。デフォルト色に戻すには、カラーピッカーの デフォルトを使用ボタンをクリックします。

単位

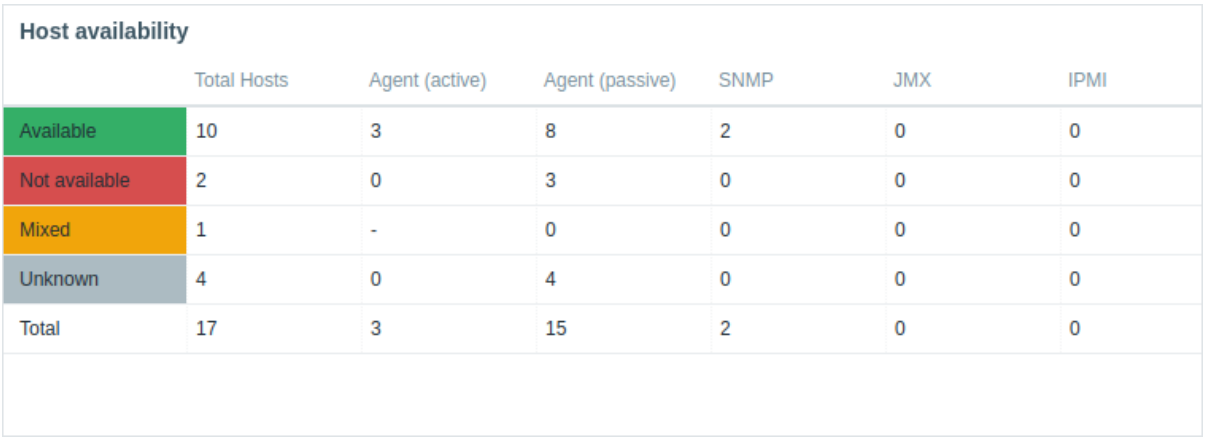
単位
アイテムの値とともに単位を表示するには、チェックボックスをオンにします。
単位名を入力すると、アイテム設定 で設定された単位を上書きします。
このパラメータは、タイプが 値に設定されている場合に使用できます。

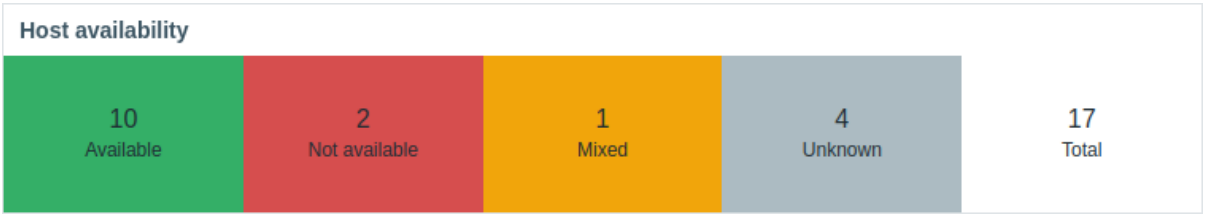
位置	アイテムの単位の位置（アイテムの値の前または後）を選択します。 このパラメータは、次の時間関連の単位では無視されます: unixtime、uptime、s。 このパラメータは、タイプが 値に設定され、かつ 単位オプションが選択されている場合に使用できます。
背景色	
背景色	カラーピッカーからハニカムセルの背景色を選択します。 「D」はデフォルト色を表し、これは Web インターフェースのテーマによって異なります。デフォルト色に戻すには、カラーピッカーの デフォルトを使用ボタンをクリックします。
しきい値	
色補間	ハニカムセルのしきい値色の間で滑らかに遷移するようにするには、チェックボックスをオンにします。 このパラメータは、2 つ以上のしきい値が設定されている場合に使用できます。
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択して、数値を指定します。 しきい値の一覧は、保存時に昇順で並べ替えられます。 しきい値として設定された色は、数値アイテムに対してのみ正しく表示されることに注意してください。 サフィックス（たとえば”1d”、”2w”、”4K”、”8G”）をサポートしています。値のマッピング をサポートしています。

ホストの可用性

概要

ホスト可用性ウィジェットは、選択したホストグループ全体の可用性を表示し、稼働中または停止中のホスト数を監視できます。



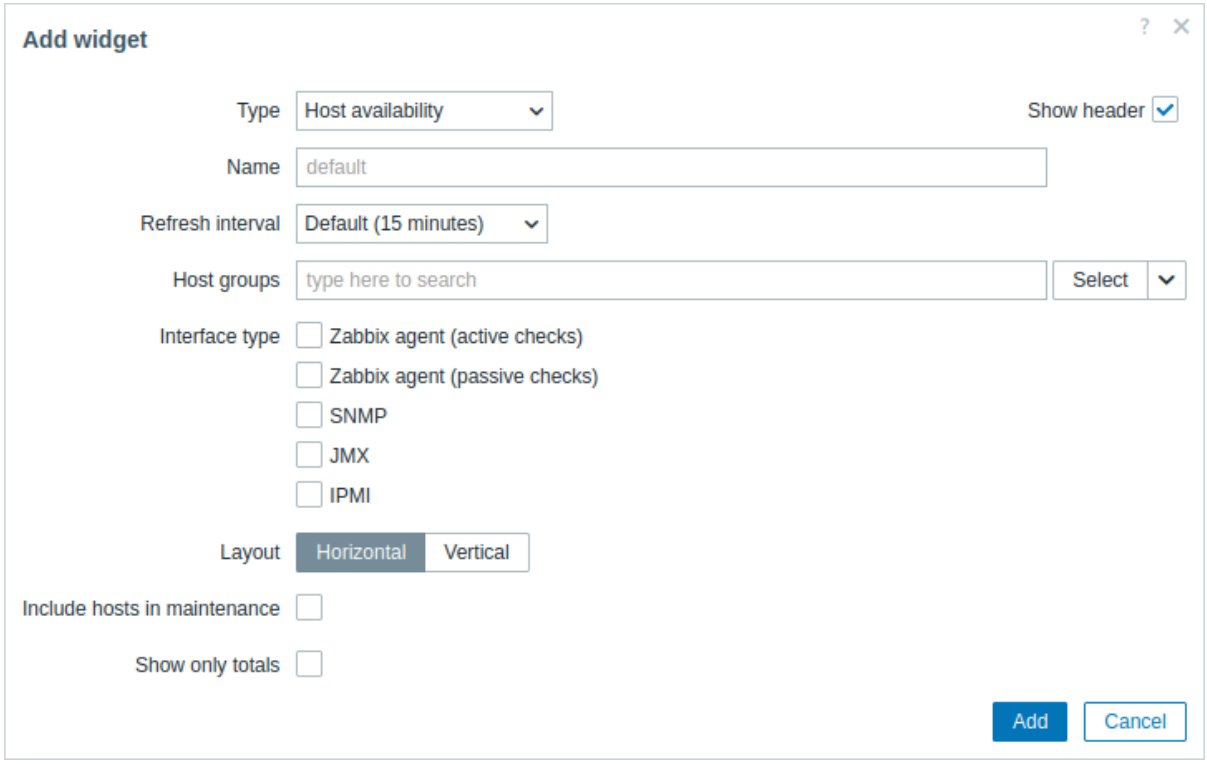


ホストの可用性は次のようにカウントされます:

- 利用可能 - すべてのインターフェースが利用可能なホスト
- 利用不可 - すべてのインターフェースが利用不可なホスト
- 混在 - 少なくとも 1 つのインターフェースが利用不可で、少なくとも 1 つが利用可能または不明なホスト
- 不明 - 少なくとも 1 つのインターフェースが不明で、利用不可なものがないホスト
- 合計 - すべてのホストの合計

設定

設定するには、タイプとして ホストの可用性を選択します。



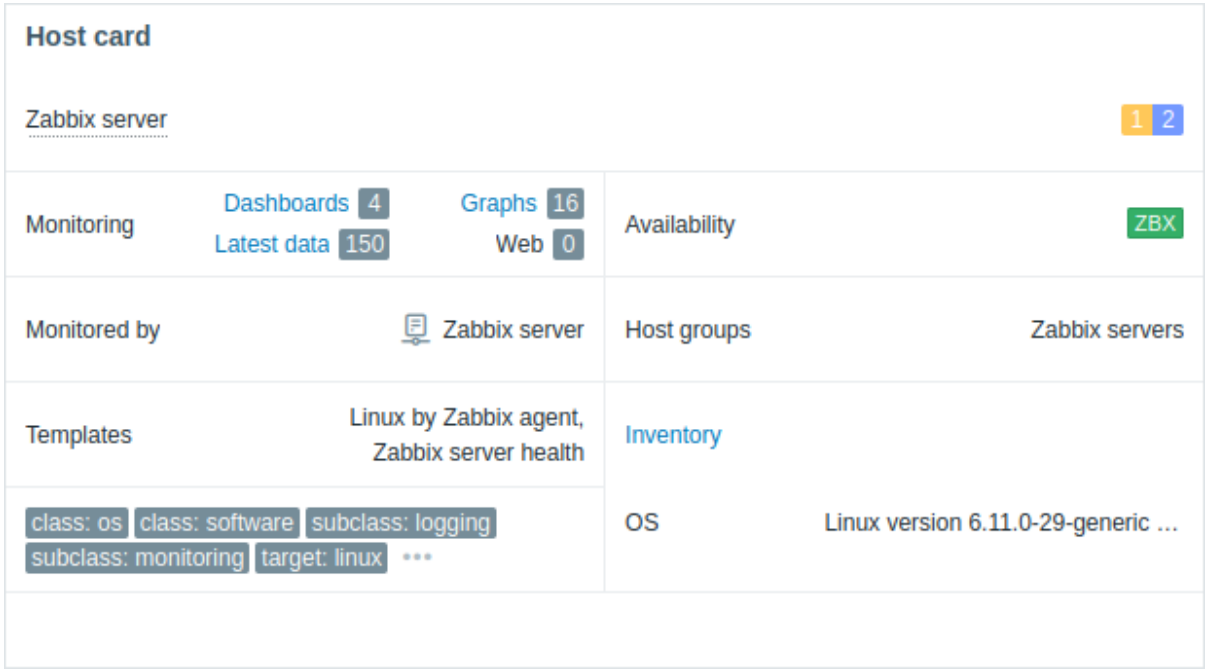
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートなので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
インターフェースタイプ	可用性データを表示するホストインターフェースを選択します—*Zabbix* エージェント（アクティブチェック）、*Zabbix* エージェント（パッシブチェック）、*SNMP*、*JMX*、*IPMI*。 *Zabbix* エージェント（アクティブチェック）の場合、このタイプのアイテムは複数のインターフェースを持つことができないため、混在セルは常に空になります。 何も選択しない場合は、すべてのインターフェースの可用性がデフォルトで表示されます。
レイアウト	横表示（列）または縦表示（行）を選択します。
メンテナンス中のホストを含める	統計にメンテナンス中のホストを含めます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はメンテナンス中のデータを表示と表示されます。
合計のみ表示	チェックを入れると、インターフェースごとの内訳なしでホストの合計が表示されます。このオプションは、1 つのインターフェースのみが選択されている場合は無効になります。

ホストカード

概要

ホストカードウィジェットは、単一のホストの詳細を表示し、ホストのステータスや設定 (可用性、問題、インベントリなど) をすばやく評価できます。

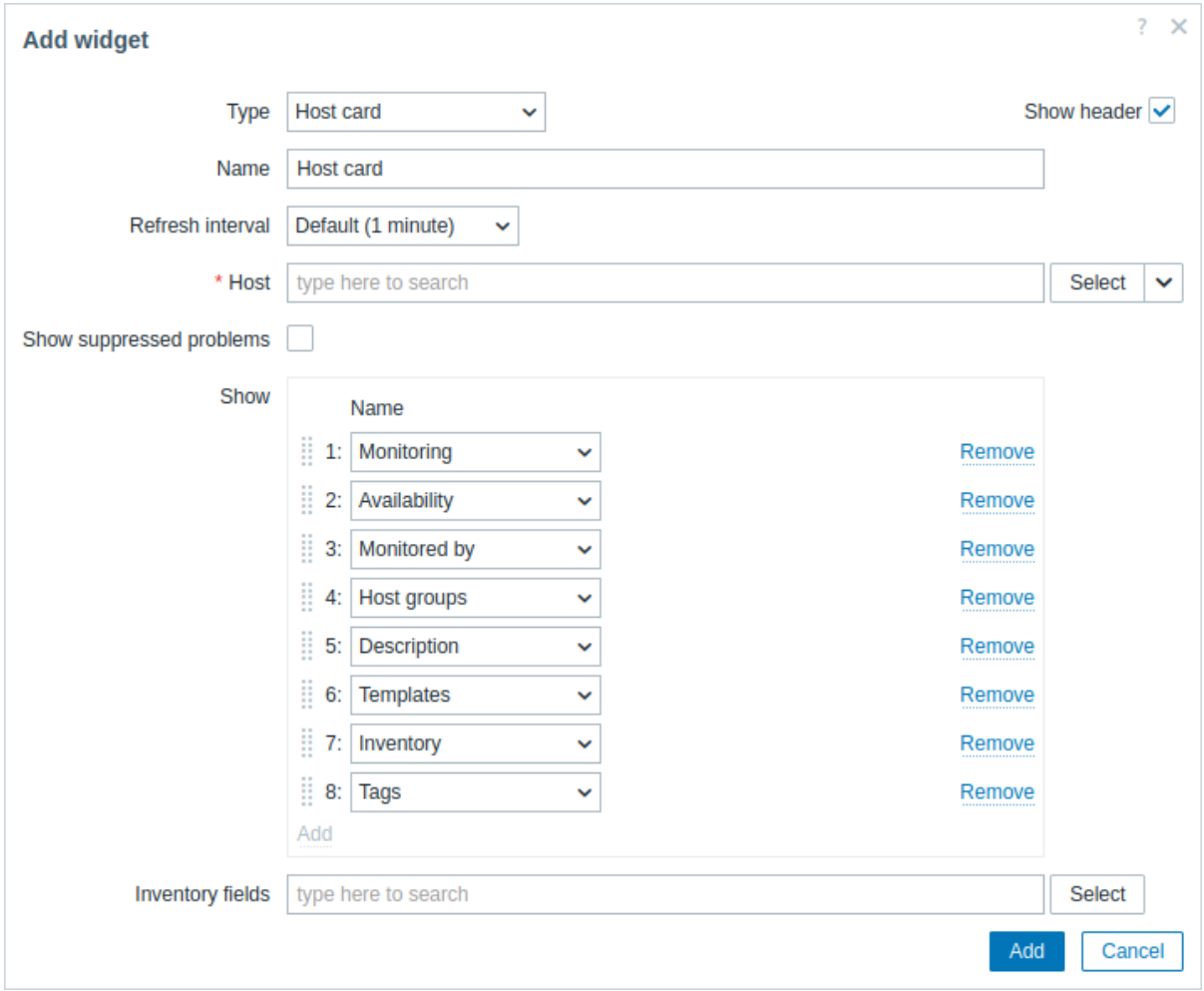


ホストはウィジェットの設定で直接指定するか、互換性のあるウィジェットまたはダッシュボードのホストセレクターから選択できます。

ウィジェットは複数のセクションを表示でき、それぞれが異なる情報を示します。セクションのレイアウトはウィジェットの幅に応じて自動的に調整されます。横に広げると、セクションは自動的に複数の列に再配置されます。

設定

設定するには、タイプとして ホストカードを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有パラメータを設定できます。

ホスト		ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 入力中に一致する候補が表示されます。 ウィジェットの上部セクションには常にホスト名が表示されます（クリックするとホストメニューが開きます）: - ホストが無効の場合、ホスト名の横に赤い無効ラベルが表示されます。 - ホストがメンテナンス中の場合、ホスト名の横にメンテナンスアイコンが表示されます。 ウィジェットの上部セクションには、ホストの障害数も重大度ごとにグループ化して表示されます（クリックするとホストの障害が開きます）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
抑制された障害を表示		このチェックボックスをオンにすると、抑制された障害も障害計算に含めます。
インベントリフィールド		ウィジェットに表示するセクションを追加します。セクションはドラッグ＆ドロップで並べ替えることができます。
	ホストグループ	ホストが所属するホストグループを表示します。
	説明	ホストの説明を表示します。
	監視	ホストのダッシュボード、最新データ、グラフ、Web シナリオへのクイックリンクを表示します。各リンクの横の数字は、それぞれのエンティティの数を示します。
	可用性	インターフェースごとのホストの可用性を表示します。
	監視元	ホストが Zabbix サーバー、特定のプロキシ、またはプロキシグループによって監視されているかどうかを表示します。
	テンプレート	ホストにリンクされているテンプレートを表示します。
	インベントリ	ホストのインベントリフィールドを表示します。
	タグ	ホストのタグを表示します。
		表示するインベントリフィールドを選択します。 入力中に一致する候補が表示されます。 インベントリフィールドが指定されていない場合、すべての入力済みインベントリフィールドが表示されます。 このパラメータは、表示にインベントリが含まれている場合に利用できます。

ホストナビゲーター

概要

ホストナビゲーターウィジェットは、さまざまなフィルタリングおよびグループ化オプションに基づいてホストのリストを表示します。

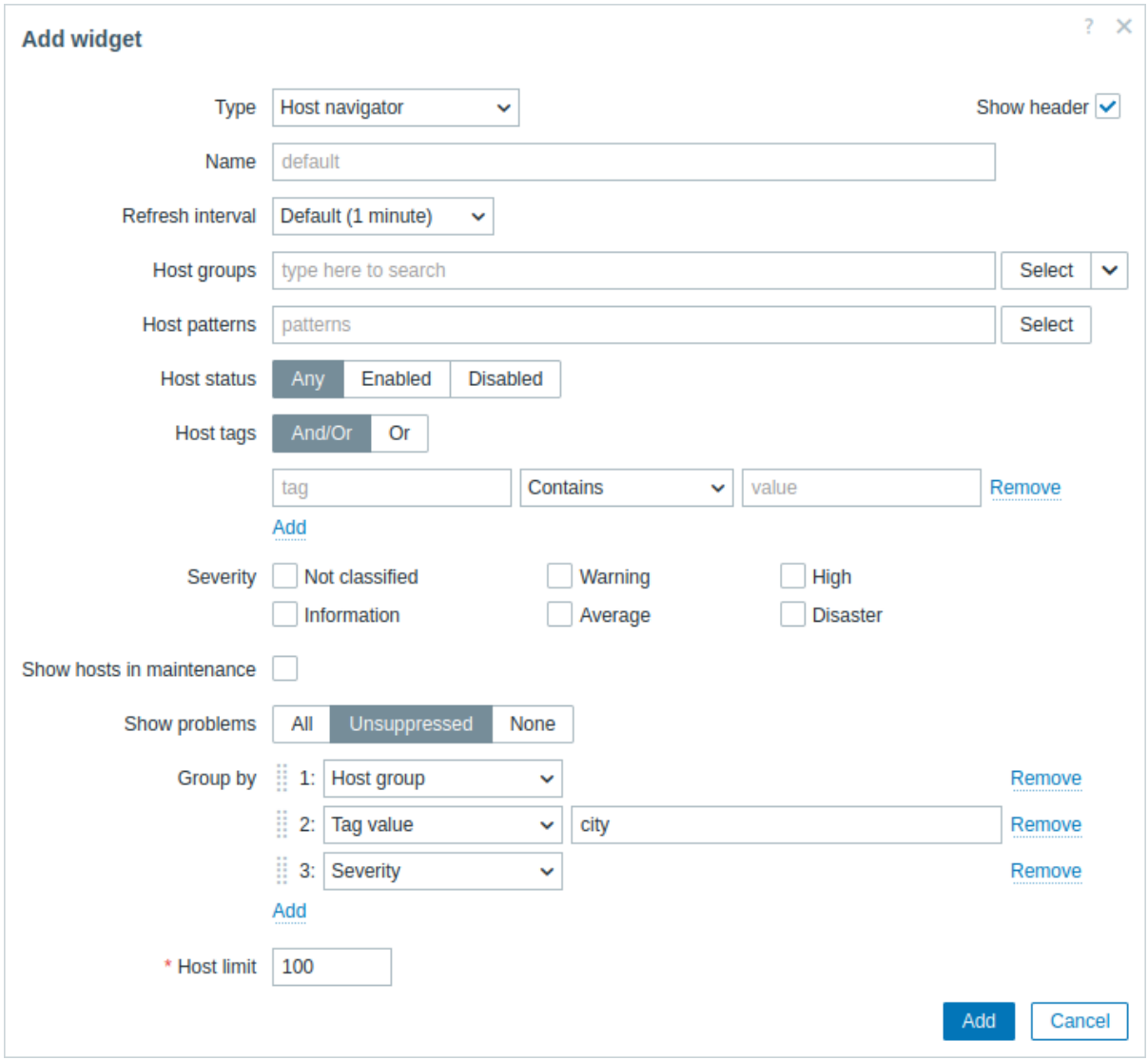
ホストが整理されているグループは、展開または折りたたむことができます。

グループ、障害、メンテナンス中のホストについては、マウスオーバーのヒントで追加の詳細情報にアクセスできます。

このウィジェットは、選択したホストに基づいて他のウィジェットが表示する内容を制御するのに最も便利です。

設定

設定するには、タイプとして *Host navigator* を選択します。



すべてのウィジェットに共通するパラメータに加えて、次の固有のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、互換性のあるウィジェットをホストグループのデータソースとして選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、ネストされたすべてのホストグループが暗黙的に選択されます。 ホストグループを選択しない場合、ウィジェットにはすべてのホストグループのすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

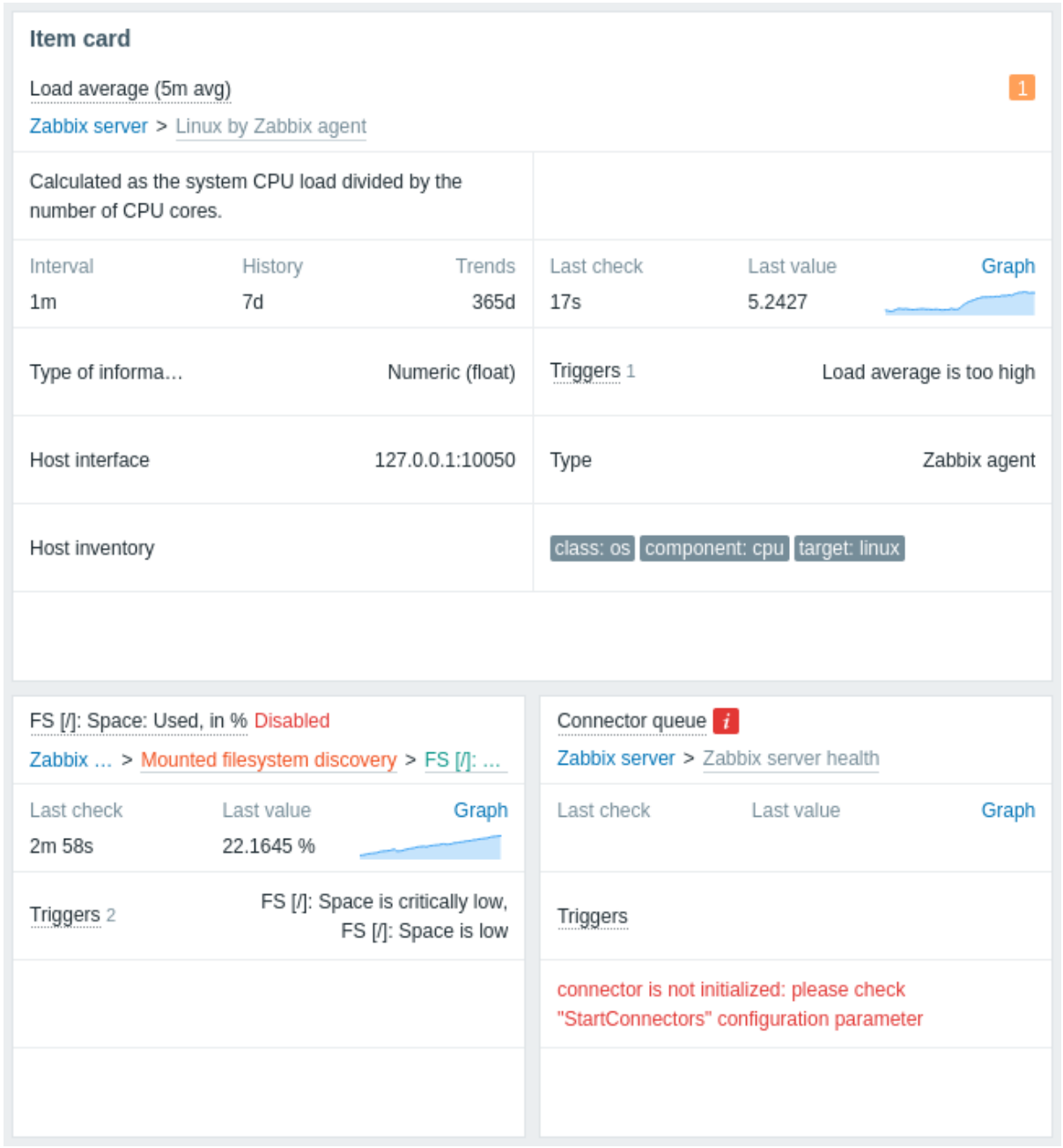
ホストパターン	ホストパターンを入力するか、既存のホストをホストパターンとして選択します。指定したパターンに一致するホストがホストナビゲーターに表示されます。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストを選択しない場合、ウィジェットにはすべてのホストが表示されます。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上の文字に一致するホストを返し、Zabbix* は「Zabbix」で始まるホストを返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力を開始すると、選択したホストグループに属するホストに限定された、一致するホストのドロップダウンリストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホストのステータス	ステータス（すべて、有効、無効）に基づいて表示するホストをフィルターします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホストタグ	ウィジェットに表示するホストをフィルターするタグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
深刻度	障害のあるホストをウィジェットに表示するため、障害の深刻度を選択します。 深刻度を選択しない場合、すべての障害があるすべてのホストが表示されます。
メンテナンス中のホストを表示	メンテナンス中のホストを表示するには、このチェックボックスを選択します（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータのラベルはメンテナンス中のデータを表示になります。
障害を表示	ステータス（すべて、抑制されていないもの、なし）に基づいて表示する障害をフィルターします。 障害は、ホスト行の末尾に色付きのブロックで示されます。色は障害の深刻度に基づいており、管理 > 一般設定 > トリガー表示オプションでカスタマイズできます。障害の更新時に障害の深刻度を変更できます。

グループ化	選択したホストをグループ化する属性を追加します。 ホストグループ - ホストグループごとにホストをグループ化します。 タグ値 - タグ名を入力し、そのタグの値ごとにホストをグループ化します（たとえば、「city」と入力すると、「Riga」、「Tokyo」などの値ごとにホストをグループ化します）。 深刻度 - 障害の深刻度ごとにホストをグループ化します。 障害を表示で障害を表示するように設定した場合、障害は次のように表示されます。 - 各深刻度グループには、対応する障害数のみが表示されます。 - 各ホストには、そのホストのすべての障害数が表示されます。 深刻度パラメータで選択した深刻度のみでホストがグループ化されることに注意してください。深刻度を選択しない場合、すべてのホストがすべての深刻度でグループ化されます。 グループ化属性は、グループ名の前にあるハンドルを上下にドラッグして並べ替えることができます。グループ化属性の順序によって、グループのネスト順が決まります。たとえば、複数のタグ名（1: color、2: city）を指定すると、ホストは color（red、blue など）、次に city（Riga、Tokyo など）の順にグループ化されます。 設定したグループ化属性によっては、1 つのホストが複数のグループに表示される場合があります（たとえば、ホストグループでグループ化し、そのホストが複数のホストグループに属している場合）。このようなホストをクリックすると、すべてのグループで選択され、強調表示されます。 設定したグループ化属性に一致しないホストは、未分類グループに表示されます。 グループ化属性は最大 10 個まで指定でき、すべて一意である必要があります。 グループ化属性を指定しない場合、ホストはグループ化されません。
ホスト数の上限	表示するホストの最大数を入力します。指定できる値は 1～9999 です。 表示可能なホスト数が設定した上限を超える場合、表示されたホストの下に対応するメッセージが表示されます（たとえば、「100 / 100+ ホストを表示しています」）。 設定したホスト数の上限は、設定したグループの表示にも影響することに注意してください。たとえば、ホスト数の上限を 100 に設定し、タグ値（200 個超）でホストをグループ化した場合、対応するホストを持つ最初の 100 個のタグ値のみがウィジェットに表示されます。 このパラメータは、管理 > 一般設定 > *GUI*の検索およびフィルター結果の上限パラメータの影響を受けません。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

アイテムカード

概要

アイテムカードウィジェットは、単一のアイテムの詳細を表示し、アイテムの状態や設定（最新データ、エラー、説明など）をすばやく評価できるようにします。

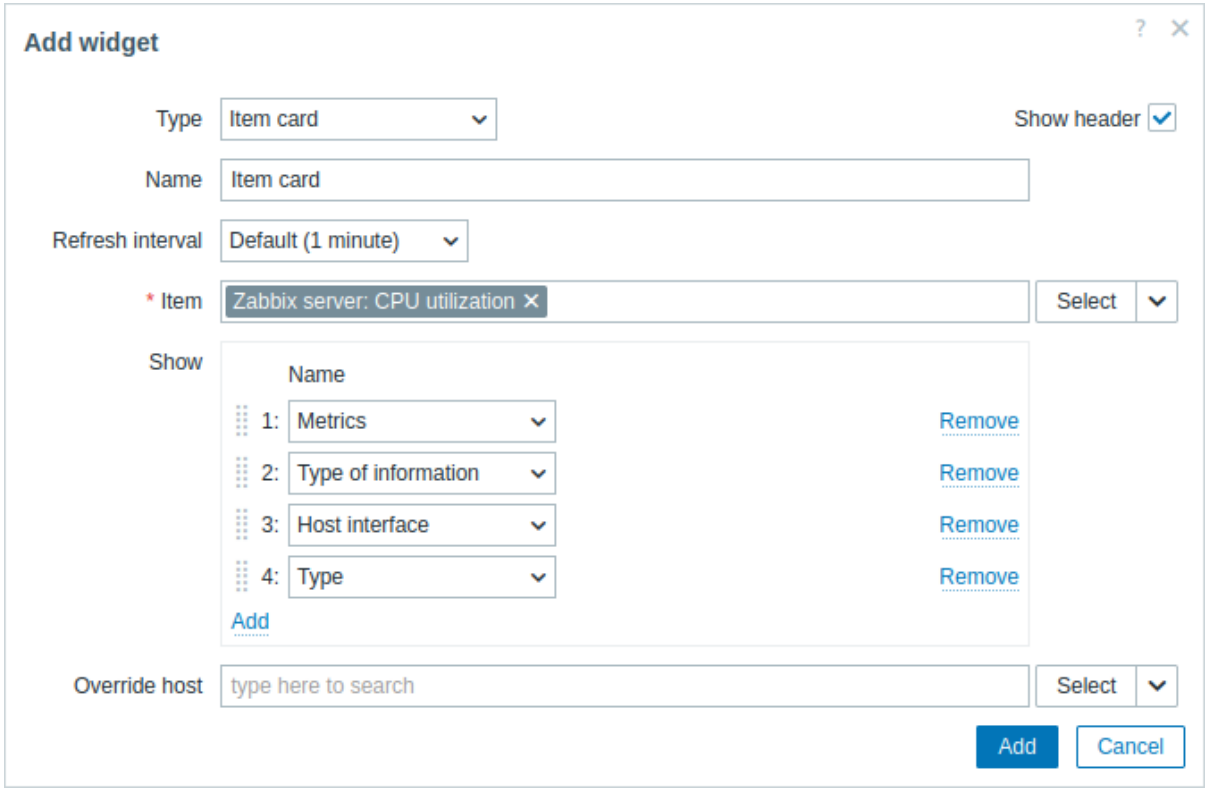


アイテムはウィジェットの設定で直接指定するか、ダッシュボード上の互換性のあるウィジェットから選択できます。

ウィジェットは複数のセクションを表示でき、それぞれが異なる情報を示します。セクションのレイアウトはウィジェットの幅に応じて自動的に調整され、水平方向に拡張すると、セクションは自動的に複数の列に再配置されます。

設定

設定するには、タイプとしてアイテムカードを選択します:



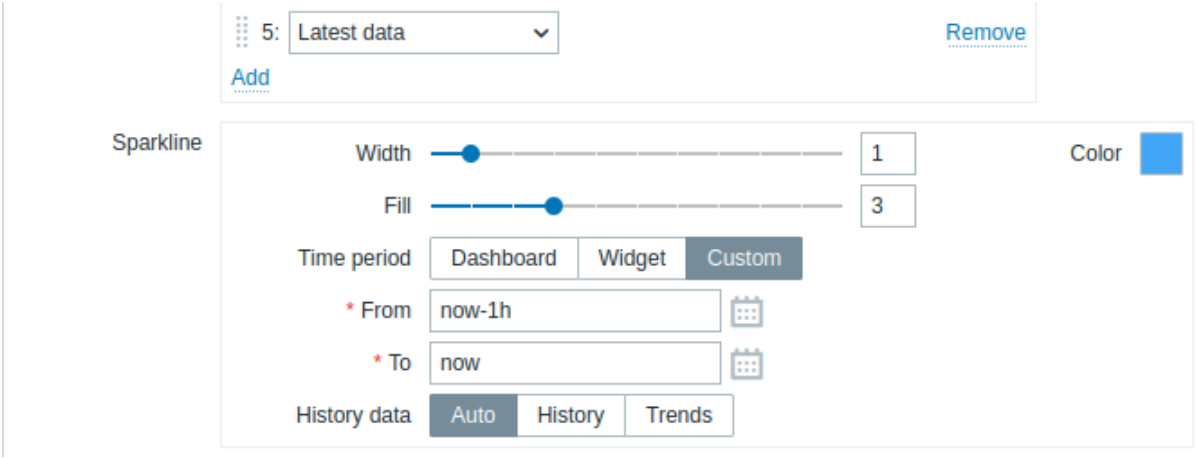
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

アイテム		アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 入力中に一致する候補が表示されます。 ウィジェットの上部セクションには常にアイテム名が表示されます（クリックするとアイテムメニューが開きます）: - アイテムが無効の場合、アイテム名の横に赤い無効ラベルが表示されます。 - エラーがある場合は、四角い「i」アイコンが表示されます。問題がある場合は、問題の深刻度ごとに問題数が表示された四角いアイコンが表示されます。アイコンにカーソルを合わせると詳細のポップアップが表示されます。 - テンプレートダッシュボードでは、アイテムの表示名の代わりに技術名が表示されます。 ウィジェットの上部セクションにはアイテムパス（ホスト名、および該当する場合はテンプレート、低レベルディスカバリ（LLD）ルール、またはマスターアイテム）が表示されます（エンティティをクリックするとその設定が開きます）。 アイテムパスは、該当するエンティティへの権限を持つユーザーにのみ表示されます。
表示		ウィジェットに表示するセクションを追加します。セクションはドラッグ＆ドロップで並べ替えできます。
	説明	アイテムの説明を表示します。マクロが展開されます。
	エラーテキスト	アイテムのエラー（存在する場合）を表示します。
	メトリクス	以下の情報を表示します： - アイテムの更新間隔（トラッパーアイテムを除く）; - アイテムの履歴設定（存在する場合）; - アイテムのトレンド設定（存在する場合）。 マクロが展開されます。 アイテムがカスタム間隔を使用している場合、アイテムの更新間隔の横にヘルプアイコンが表示されます。アイコンにカーソルを合わせるとカスタム間隔の詳細のポップアップが表示されます。

	最新データ	以下の情報を表示します： - 最後のアイテムチェックからの経過時間； - アイテムの最新値； - アイテム値のシンプルグラフ/履歴へのリンク（履歴またはトレンドが保存されている場合）； - 設定可能なスパークラインチャート（アイテムが数値型の場合）。 バイナリ型アイテムの場合、最新値の代わりにサムネイルまたは表示オプションが表示されます。サムネイルにカーソルを合わせると画像のポップアップが表示されます。表示にカーソルを合わせるとアイテム値（Base64 文字列）のポップアップが表示されます。 テンプレートダッシュボードでは、アイテムの最新値の代わりにデータなしが表示されます。
	情報の型	アイテムの情報の型を表示します。
	トリガー	以下の情報を表示します： - トリガー数付きのトリガーラベル（トリガーにカーソルを合わせるとトリガーの詳細のポップアップが表示されます）； - トリガー名のリスト（マクロが展開されます）。
	ホストインターフェース	アイテムで使用されているホストインターフェース、またはホストインターフェースを持たないアイテム（依存アイテム、シンプルチェックなど）の場合はデータなしを表示します。 テンプレートダッシュボードではデータなしが表示されます。
	タイプ	アイテムのタイプを表示します。
	ホストインベントリ	アイテム値が入力されるホストインベントリフィールドを表示します。
	タグ	アイテムのタグを表示します。
ホストの上書き		ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

スパークライン

スパークラインチャートの設定オプションは、最新データセクションがウィジェットに追加されている場合のみ利用できます。



幅	スライダーを使用するか、0 から 10 の範囲で値を手動入力してグラフ線の太さを設定します。
色	線と塗りつぶしの色を選択します。
塗りつぶし	スライダーを使用するか、0 から 10 の範囲で値を手動入力して塗りつぶし色の透明度を設定します。
期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェットを使用（ウィジェットパラメータで設定）； カスタム - *From* および *To* パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。

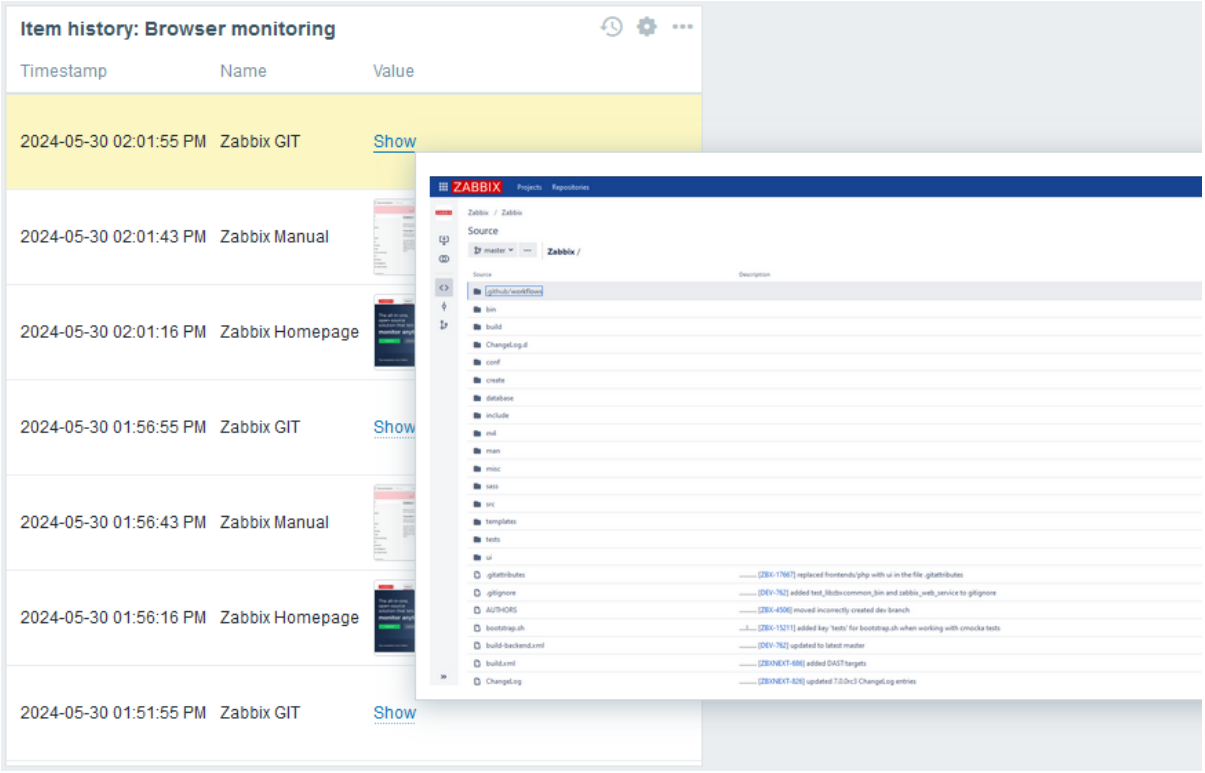
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。 このパラメータは数値データにのみ適用されます。非数値データは常に履歴から取得されます。

アイテム履歴

概要

アイテム履歴ウィジェットは、すべてのアイテムタイプ（数値、テキストなど）の最新データを表形式で表示します。進捗バーや画像を表示でき（browser アイテムで便利です）、値を強調表示することもできます（ログファイル監視で便利です）。

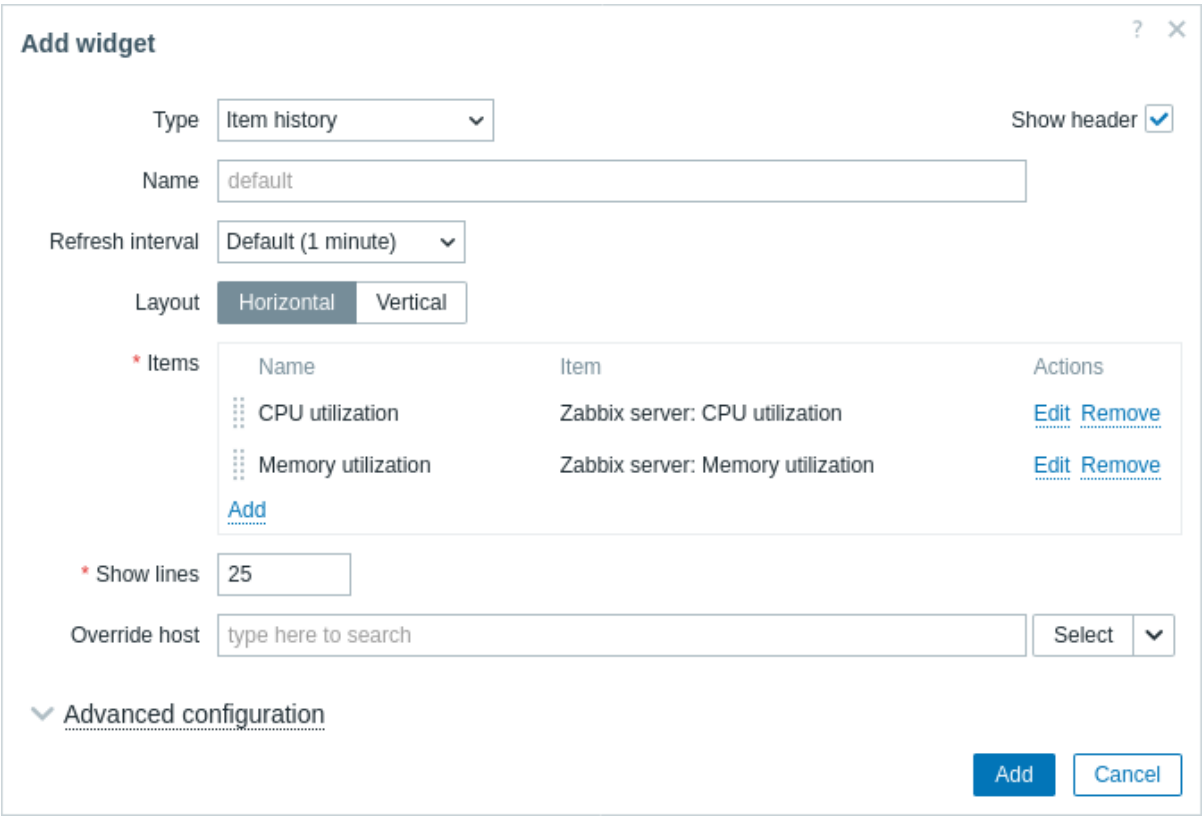




最大 1000 件のレコードを表示できます。

設定

設定するには、タイプとして アイテム履歴を選択します:



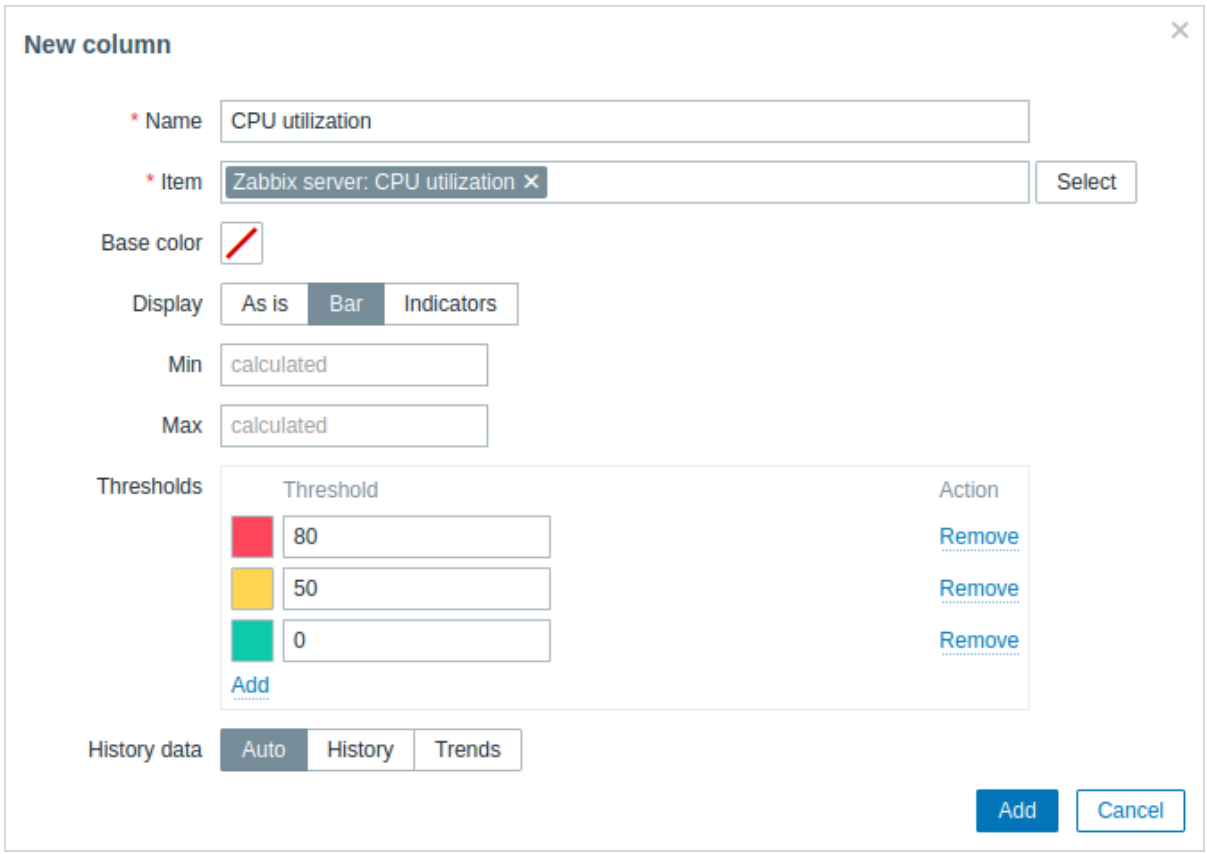
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

レイアウト	アイテム列のレイアウトオプションを選択します: 水平 - アイテムが水平に、値が垂直に表示されます; 垂直 - アイテムが垂直に、値が水平に表示されます。
アイテム	表示する列アイテムを追加します。 アイテムの順序が表示順序を決定します。 アイテム名の前のハンドルで上下にドラッグして並べ替えることができます。

行数の表示	表示するアイテム値の行数を指定します。
ホストの上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードのホストセレクタを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。

カラムの設定

アイテムカラムを設定するには、アイテムパラメータで追加をクリックします。



一般的なカラムパラメータ:

名前	カラムの名前を入力します。 空欄の場合、アイテムパラメータのアイテム名が使用されます。
アイテム	アイテムを選択します。 選択したアイテムの情報タイプによってカラムの設定パラメータが異なることに注意してください。詳細は下記の個別パラメータを参照してください。 テンプレートのダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみ選択できます。
ベースカラー	カラムの背景色、または表示が「バー」や「インジケーター」の場合は塗りつぶし色を選択します。 ベースカラーはしきい値やハイライトカラーで上書きされる場合があることに注意してください。

数値タイプのアイテムに固有のカラムパラメータ:

表示	アイテム値の表示方法を選択します: そのまま - 通常のテキストとして; バー - ソリッドな色付きバーとして; インジケータ - セグメント化された色付きバーとして。
最小	バー/インジケータの最小値を入力します。 空欄の場合、ウィジェットはアイテムの最小値を使用します。 このパラメータは表示が「バー」または「インジケータ」に設定されている場合のみ利用可能です。

最大	バー/インジケータの最大値を入力します。 空欄の場合、ウィジェットはアイテムの最大値を使用します。 このパラメータは表示が「バー」または「インジケータ」に設定されている場合のみ利用可能です。
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 しきい値リストは保存時に昇順でソートされます。 サフィックス（例: "1d", "2w", "4K", "8G"）がサポートされています。値のマッピングもサポートされています。
履歴データ	履歴またはトレンドからデータを取得するか選択します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。

文字、テキスト、ログ、および JSON タイプのアイテムに固有の列パラメータ:

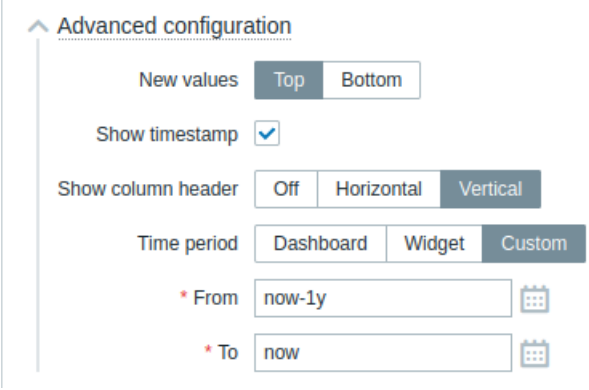
Highlights	*Add* をクリックしてハイライトを追加し、カラーピッカーからハイライト色を選択して、正規表現を指定します。 選択した色は、指定した正規表現がテキストに一致するアイテム値の背景色として使用されます。
Display	アイテム値の表示方法を選択します: **As is** - 受信したとおりにそのまま表示します。値に改行が含まれている場合は、改行が保持されます。*Layout* が"Vertical" に設定され、かつ複数の列が構成されている場合は、ワードラップが適用されます; **HTML** - HTML 形式のテキストとして表示します; **Single line** - 1 行として表示し、指定した長さ (1-500 文字) に切り詰めます。 切り詰められた値にカーソルを合わせるかクリックすると、完全な値を含むポップアップが開きます。
Use monospace font	このチェックボックスをオンにすると、アイテム値を等幅フォントで表示します (既定ではオフ)。
Display local time	このチェックボックスをオンにすると、タイムスタンプ列にタイムスタンプの代わりにローカル時刻を表示します。 advanced configuration の *Show timestamp* チェックボックスもオンにする必要があります。 このパラメータはログタイプのアイテムでのみ使用できます。

バイナリタイプのアイテムに固有のカラムパラメータ:

サムネイルを表示	画像バイナリの場合はサムネイルを、非画像バイナリの場合は「表示」オプションを表示するには、このチェックボックスをオンにします。 このチェックボックスをオフにすると、すべてのバイナリアイテム値に「表示」オプションが表示されます。 「表示」オプションにマウスを合わせるかクリックすると、アイテム値（画像または Base64 文字列）がポップアップウィンドウで開きます。 アイテム値が空文字列の場合、「表示」オプションが表示され、それにマウスを合わせるかクリックすると「空文字列」と表示されたポップアップが開きます。

詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用できます。



新しい値	新しいアイテム値をどこに追加するかを選択します: 上 - 列の一番上に追加; 下 - 列の一番下に追加。
タイムスタンプを表示	このチェックボックスをオンにすると、タイムスタンプ列が表示されます（デフォルトではオフ）。
列ヘッダーを表示	列ヘッダーの表示方法を選択します: オフ - ヘッダーを非表示; 水平 - ヘッダーを水平に表示; 垂直 - ヘッダーを垂直に表示。
期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメーターで設定）を使用; カスタム - 開始および終了パラメーターで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェット（グラフ、グラフ（クラシック）、グラフプロトタイプ）を入力または選択します。 このパラメーターは、期間が「ウィジェット」に設定されている場合に利用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメーターは、期間が「カスタム」に設定されている場合に利用できます。

アイテムナビゲーター

概要

アイテムナビゲーターウィジェットは、さまざまなフィルタリングおよびグループ化オプションに基づいてアイテムのリストを表示します。

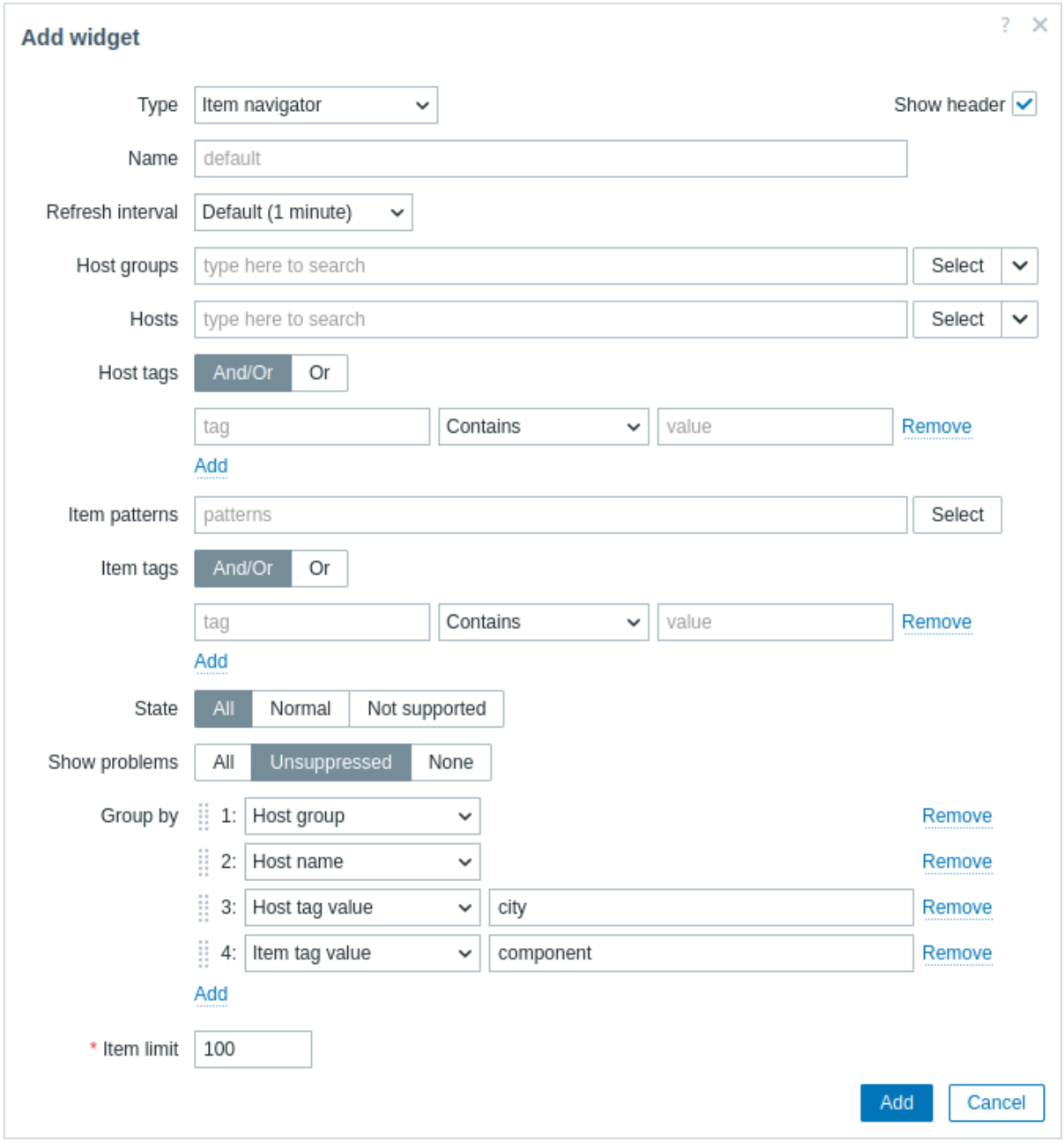
アイテムが整理されているグループは、展開または折りたたむことができます。

グループや問題については、マウスオーバーのヒントで追加の詳細情報にアクセスできます。

このウィジェットは、選択したアイテムに基づいて他のウィジェットが表示する内容を制御するのに最も便利です。

設定

設定するには、タイプとして アイテムナビゲーターを選択します。



すべてのウィジェットに共通するパラメータに加えて、次の固有のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、すべてのネストされたホストグループが暗黙的に選択されます。 ホストグループを選択しない場合、すべてのホストグループに属するすべてのホストのアイテムがウィジェットに表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストを選択しない場合、すべてのホストに属するアイテムがウィジェットに表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

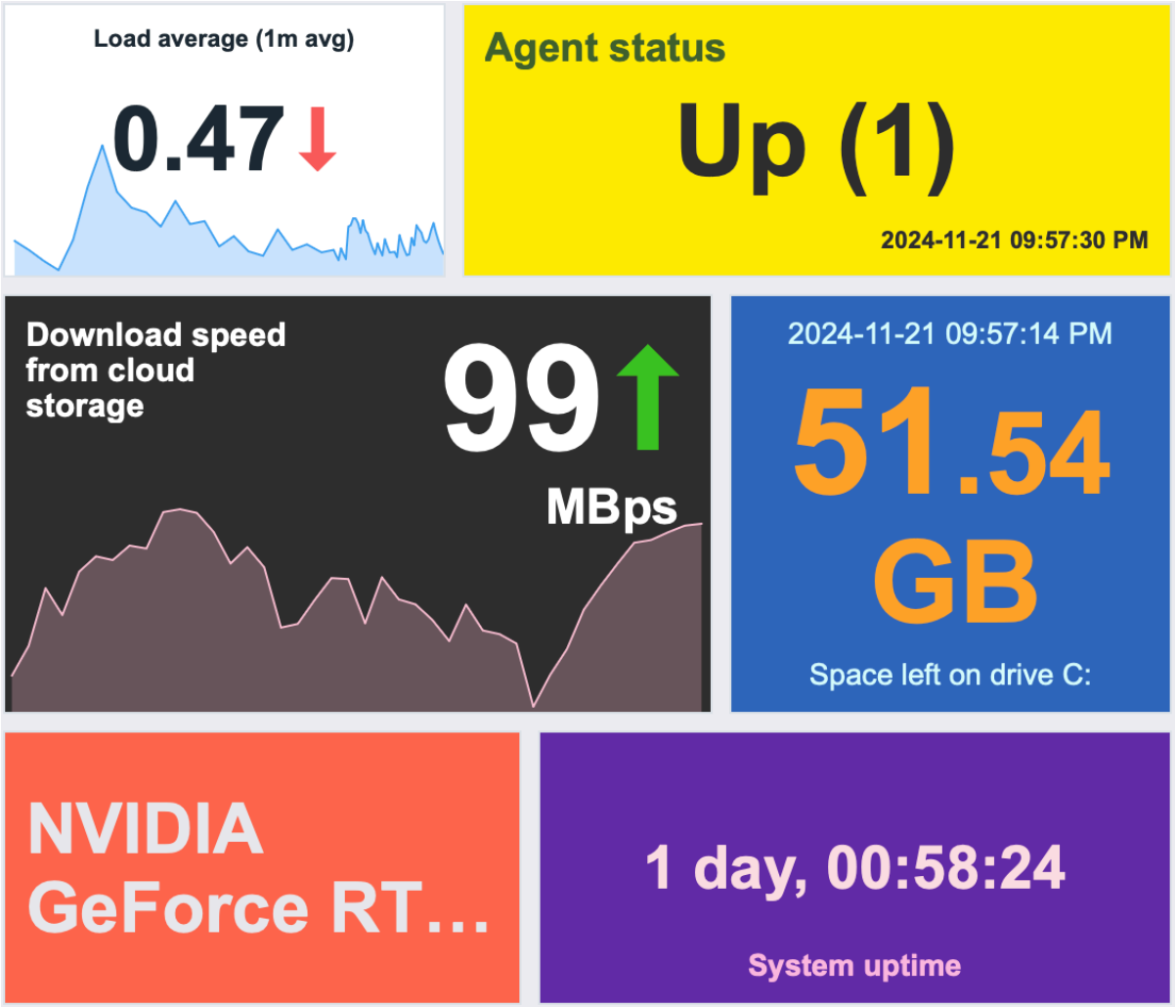
ホストタグ	ウィジェットに表示するアイテムをフィルターするためのホストタグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
アイテムタグ	ウィジェットに表示するアイテムをフィルターするためのアイテムタグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
状態	状態（すべて、正常、未サポート）に基づいて表示するアイテムをフィルターします。
障害を表示	ステータス（すべて、抑制されていない障害、なし）に基づいて表示する障害をフィルターします。 障害は、アイテム行の末尾に色付きのブロックで示されます。色は障害の深刻度に基づいており、管理 > 一般設定 > トリガー表示オプションでカスタマイズできます。障害の更新時に障害の深刻度を変更できます。

グループ化の基準	アイテムをグループ化するためのグループ化属性を追加します。 ホストグループ - ホストのホストグループごとにアイテムをグループ化します。 ホスト名 - ホストごとにアイテムをグループ化します。 ホストタグ値 - このホストタグの値ごとにアイテムをグループ化するタグ名を入力します（たとえば、「city」と入力すると、「Riga」、「Tokyo」などの値ごとにアイテムがグループ化されます）。 アイテムタグ値 - このアイテムタグの値ごとにアイテムをグループ化するタグ名を入力します（たとえば、「component」と入力すると、「cpu」、「memory」などの値ごとにアイテムがグループ化されます）。 グループ化属性は、グループ名の前にあるハンドルを使用して上下にドラッグすることで並べ替えられます。グループ化属性の順序によって、グループのネスト順序が決まることに注意してください。たとえば、複数のホストタグ名（1: color、2: city）を指定すると、アイテムは color（red、blue など）ごとにグループ化され、その後 city（Riga、Tokyo など）ごとにグループ化されます。 設定したグループ化属性によっては、1 つのアイテムが複数のグループに表示される場合があります（たとえば、ホストグループごとにグループ化していて、アイテムのホストが複数のホストグループに属している場合）。このようなアイテムをクリックすると、すべてのグループで選択され、強調表示されます。 設定したグループ化属性に一致しないアイテムは、未分類グループに表示されます。 障害を表示で障害を表示するように設定した場合、障害は次のように表示されます。 - 各グループには、配下にあるすべてのアイテムの障害数が表示されます。 - 各アイテムには、そのアイテムの障害数のみが表示されます。 最大 10 個のグループ化属性を指定できます。すべて一意である必要があります。 グループ化属性を指定しない場合、アイテムはグループ化されません。
アイテム数の上限	表示するアイテムの最大数を入力します。指定できる値は 1～9999 です。 表示可能なアイテム数が設定した上限を超える場合、表示されたアイテムの下に対応するメッセージが表示されます（たとえば、「100 of 100+ items are shown」）。 設定したアイテム数の上限は、設定したグループの表示にも影響することに注意してください。たとえば、アイテム数の上限を 100 に設定し、アイテムをホストごとにグループ化した場合（各ホストに 200 個のアイテムが含まれている場合）、最初のホストとその 100 個のアイテムのみがウィジェットに表示されます。 このパラメータは、管理 > 一般設定 > *GUI*の 検索およびフィルター結果の上限パラメータの影響を受けません。

アイテムの値

概要

アイテム値ウィジェットは、単一の数値アイテムまたは文字列アイテムの値を表示します。重要なメトリックを監視したり、しきい値を可視化したり、データの急激な変化を検出したりするのに役立ちます。



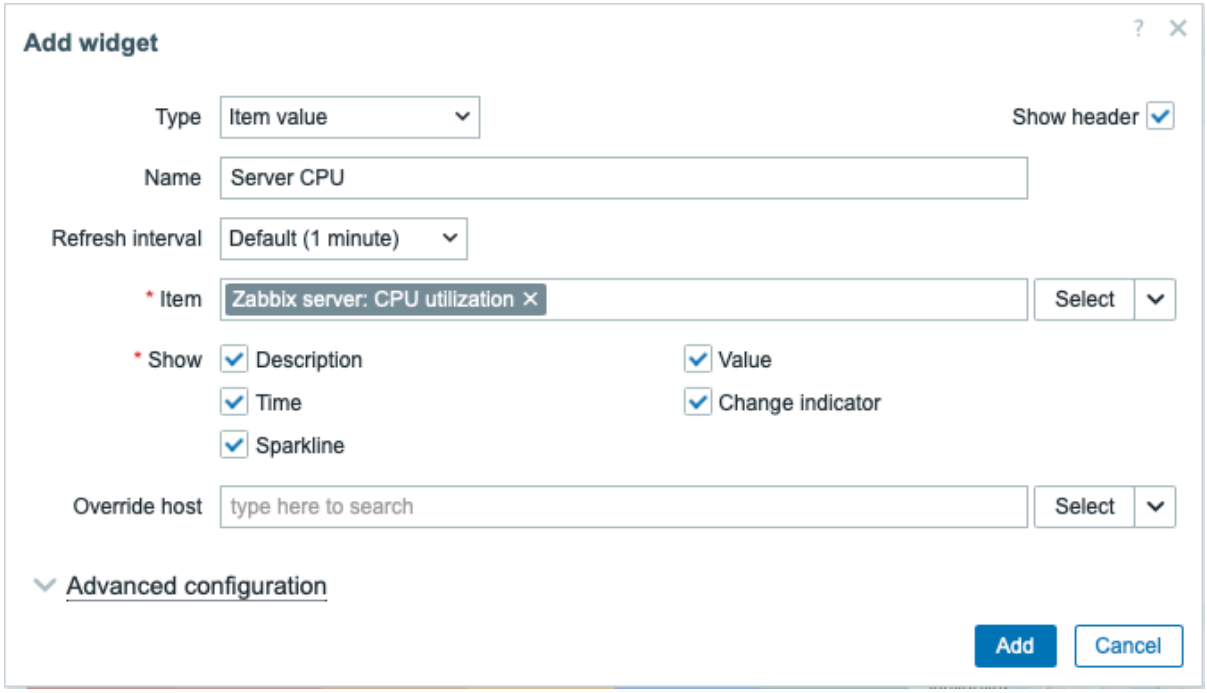
ウィジェットでは、次の内容を表示するように設定できます。

- メトリックのタイムスタンプ (2024-11-21 09:57:30)
- アイテムの説明 (Load average (1m), エージェント status)
- アイテムの値、変化インジケーター、および単位 (0.47 ↓, 99 ↑ Mbps)
- 背景色
- 指定した期間の値を示すスパークラインチャート

ウィジェットをクリックすると、数値アイテムの場合はアドホックグラフ が、文字列アイテムの場合は最新データが開きます。

設定

設定するには、ウィジェットタイプとして アイテムの値を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

アイテム	アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 バイナリデータを返すアイテムはサポートされていません。
表示	チェックボックスをオンにして、対応する要素を表示します: 説明、値、時刻、変化インジケーター、スパークライン。非表示にするにはオフにします。少なくとも 1 つの要素を選択する必要があります。
ホストの上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
高度な設定	高度な設定オプションを表示するには、高度な設定の見出しをクリックします。

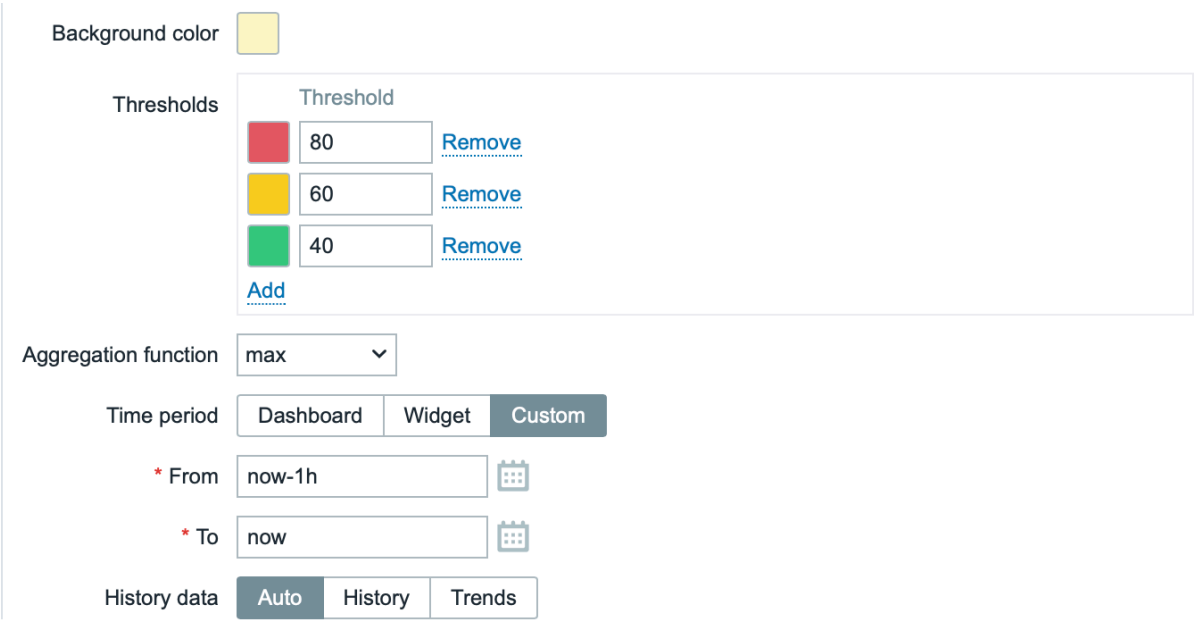
詳細設定

詳細設定オプションは、折りたたみ可能な詳細設定セクションで利用でき、表示フィールドで選択された要素にのみ適用されます (上記参照)。

複数の要素が同じスペースを占有することはできません。同じスペースに配置した場合はエラーメッセージが表示されますのでご注意ください。

汎用ウィジェットパラメータ

これらのパラメータでは、ウィジェット全体の背景色（静的または動的）と、値の表示に使用する集計関数を設定します。

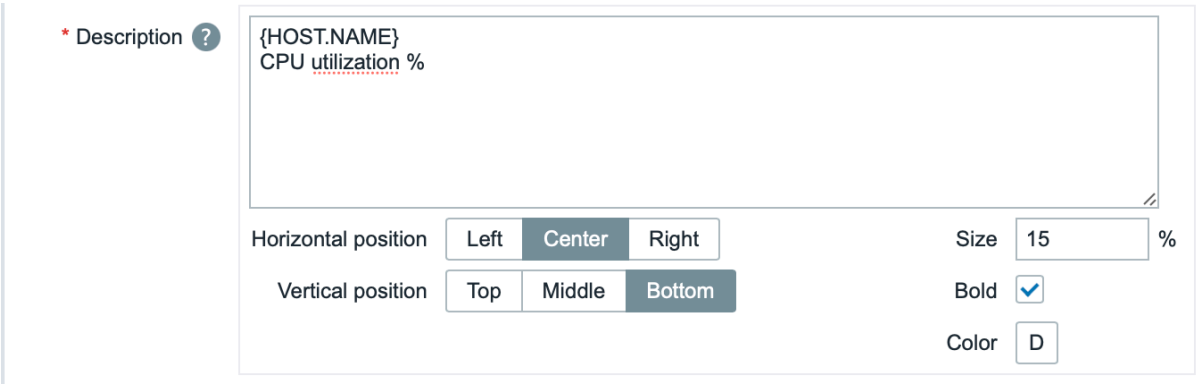


背景色	カラーピッカーから、ウィジェット全体の背景色を選択します。 D はデフォルトの色（Web インターフェースのテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーの デフォルトを使用ボタンをクリックします。
しきい値	ウィジェット全体の動的な背景色を設定します。追加をクリックしてしきい値を追加し、カラーピッカーから背景色を選択して、数値を指定します。アイテムの値がしきい値以上になると、背景色が変わります。 保存時に、リストは昇順に並べ替えられます。 動的な背景色が正しく表示されるのは、数値アイテムの場合のみです。
集計関数	選択した 期間内で使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **count** - 値の件数を表示します。 **sum** - 値の合計を表示します。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します。 使用しないを選択すると、最新の値が表示されます（集計なし）。 集計を使用すると、最新の値ではなく、指定した期間（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* で表示できるのは数値データのみです。*count* の場合、数値以外のデータは数値に変換されます。
期間	値の集計に使用する期間です。期間のデータソース を選択します。 ダッシュボード - ダッシュボードの期間セレクター を使用します。 ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します。 カスタム - 開始および 終了パラメータで設定したカスタム期間を使用します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻を示します。 ウィジェットの 期間設定に関係なく、互換性のあるウィジェットを期間のデータソースとして使用できます。 集計関数が「使用しない」に設定されている場合、このパラメータは非表示になります。
ウィジェット	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 期間が「ウィジェット」に設定されている場合に使用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 期間が「カスタム」に設定されている場合に使用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 期間が「カスタム」に設定されている場合に使用できます。

履歴データ	履歴またはトレンドからデータを取得します。 自動 - 自動的に選択します。 履歴 - 履歴データを取得します。 トレンド - トレンドデータを取得します。 この設定は数値データにのみ適用されます。数値以外のデータは常に履歴から取得されます。

説明

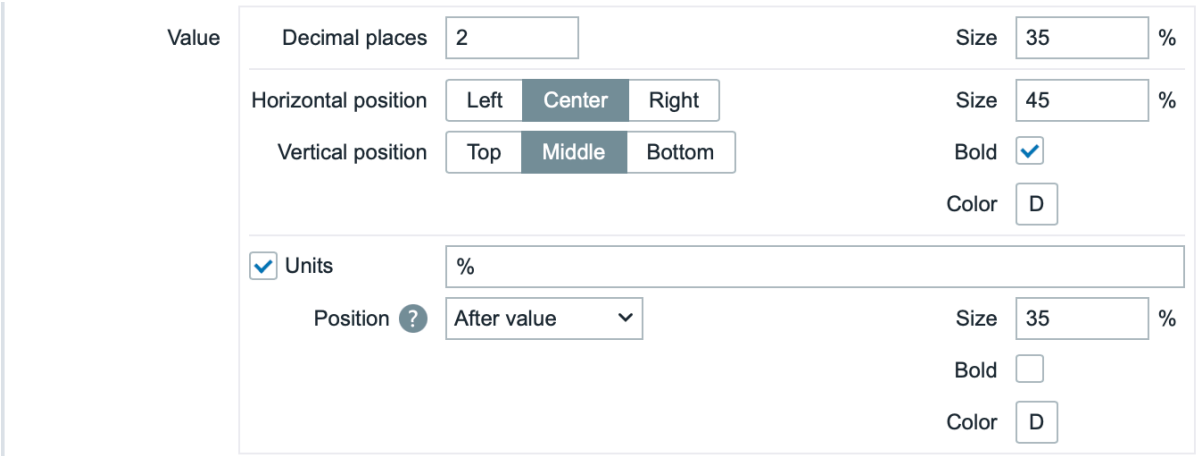
これらのパラメータは、アイテムの説明をどのように表示するかを決定します。



説明	アイテムの説明を入力します。この説明はデフォルトのアイテム名を上書きする場合があります。 複数行の説明がサポートされています。テキストとサポートされているマクロの組み合わせが可能です。 {HOST.*}、{ITEM.*}、{INVENTORY.*} およびユーザーマクロがサポートされています。
水平位置	アイテム説明の水平位置を選択します（左、右、中央）。
垂直位置	アイテム説明の垂直位置を選択します（上、下、中央）。
サイズ	アイテム説明のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。
太字	チェックボックスをオンにすると、アイテム説明が太字で表示されます。
色	カラーピッカーからアイテム説明の色を選択します。 D はデフォルト色（フロントエンドテーマによって異なります）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

値

これらのパラメータは、アイテム値の表示方法を決定します。

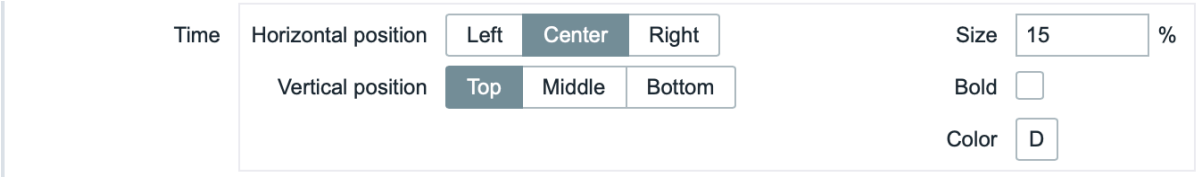


小数点以下の桁数	値に表示する小数点以下の桁数を選択します。この値は float 型のアイテムにのみ影響します。単位が「s」に設定されているアイテムの場合、この値が 0 より大きい場合（デフォルトは 2）、ウィジェットは最も重要な時間単位を四捨五入し、値を完全な時間文字列に変換する代わりに数値（例：「10.43m」）を表示します。0 に設定すると、秒が人間が読みやすい時間文字列（例：「4h 56m 30s」）に変換されます。

サイズ	小数点以下の桁数のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。
水平位置	アイテム値の水平位置を選択します（左、右、中央）。
垂直位置	アイテム値の垂直位置を選択します（上、下、中央）。
サイズ	アイテム値のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。 アイテム値のサイズが優先されることに注意してください。他の要素は値のためにスペースを譲る必要があります。ただし、変化インジケーターがある場合、値が大きすぎると変化インジケーターを表示するために切り捨てられます。
太字	チェックボックスをオンにすると、アイテム値が太字で表示されます。
色	カラーピッカーからアイテム値の色を選択します。 D はデフォルト色（フロントエンドテーマによる）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
単位	
単位	チェックボックスをオンにすると、アイテム値に単位が表示されます。単位名を入力すると、アイテム設定の単位を上書きします。
位置	アイテム単位の位置を選択します（値の上、下、前、後）。
サイズ	アイテム単位のフォントサイズの高さを入力します（ウィジェット全体の高さに対するパーセント）。
太字	チェックボックスをオンにすると、アイテム単位が太字で表示されます。
色	カラーピッカーからアイテム単位の色を選択します。 D はデフォルト色（フロントエンドテーマによる）を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

時刻

これらのパラメータは、時刻 (アイテム履歴のクロック値) をどのように表示するかを決定します。



水平位置	時刻の水平位置を選択します - 左、右、中央。
垂直位置	時刻の垂直位置を選択します - 上、下、中央。
サイズ	時刻のフォントサイズの高さを入力します (ウィジェット全体の高さに対するパーセント)。
太字	チェックボックスをオンにすると、時刻が太字で表示されます。
色	カラーピッカーから時刻の色を選択します。 D はデフォルト色 (フロントエンドのテーマによって異なります) を表します。デフォルト値に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

変化インジケーター

このセクションでは、カラーピッカーから変化インジケーターの色を選択できます。



変化インジケーターは次のとおりです。

- ↑ - アイテムの値が増加（数値アイテムの場合）
- ↓ - アイテムの値が減少（数値アイテムの場合）
- ↕ - アイテムの値が変化（文字列アイテムおよび値のマッピングが設定されたアイテムの場合）

注:

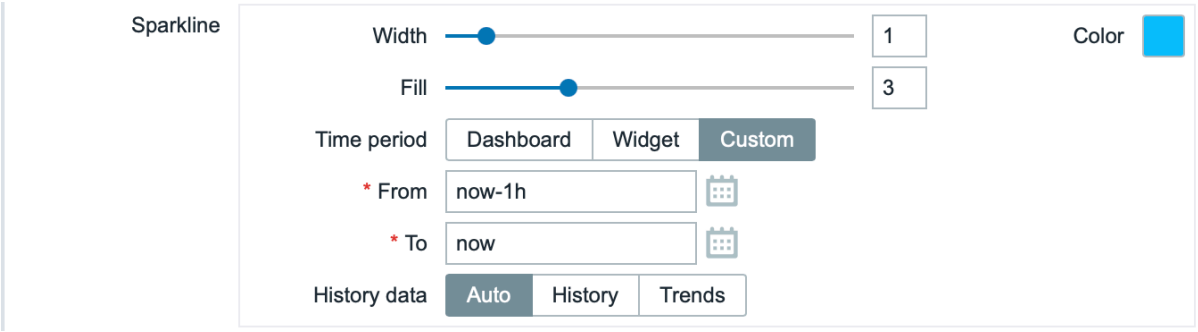
- 値が 1 つしかない場合、増加（↑）および減少（↓）のインジケーターは表示されません。
- 値は常に同じスコープ内で比較されます。例:
  - 最新の値は、前の値と比較されます。
  - 月次の値は、前月と比較されます。
  - 集計の場合、前の期間は選択した期間と同じ長さで、選択した期間の開始直前に終了します。

D はデフォルトの色を表します（Web インターフェースのテーマによって異なります）。デフォルト値に戻すには、カラーピッカーの デフォルトを使用ボタンをクリックします。

変化インジケーターの垂直方向のサイズは、値のサイズ（数値アイテムの場合は値の整数部分）と同じです。

スパークライン

これらのパラメータは、スパークラインチャートの表示方法を決定します。

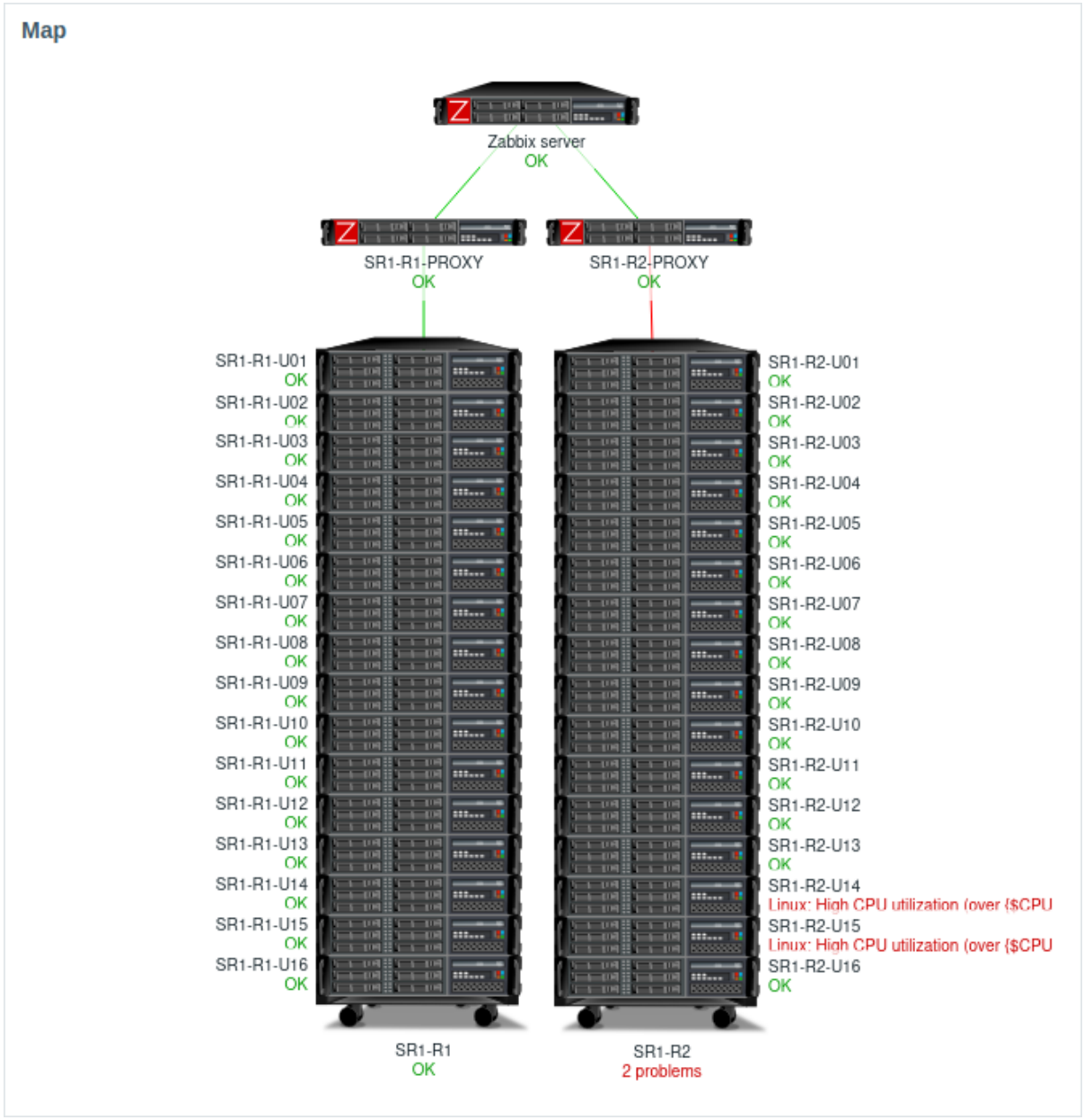


幅	グラフの線の太さをスライダーで設定するか、0 から 10 の範囲で値を手動で入力します。
色	線と塗りつぶしの色を選択します。
塗りつぶし	スライダーで塗りつぶしの透明度を設定するか、0 から 10 の範囲で値を手動で入力します。
期間	スパークラインチャートに含める値の期間を指定します。期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクターを使用; ウィジェット - 互換性のあるウィジェットを使用（ウィジェットパラメータで設定）; カスタム - 開始および終了パラメータで設定したカスタム期間を使用; 設定されている場合、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定された時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。 このパラメータは、集約関数が「使用しない」に設定されている場合は利用できません。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に利用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、期間が「カスタム」に設定されている場合に利用できます。
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得;

マップ

概要

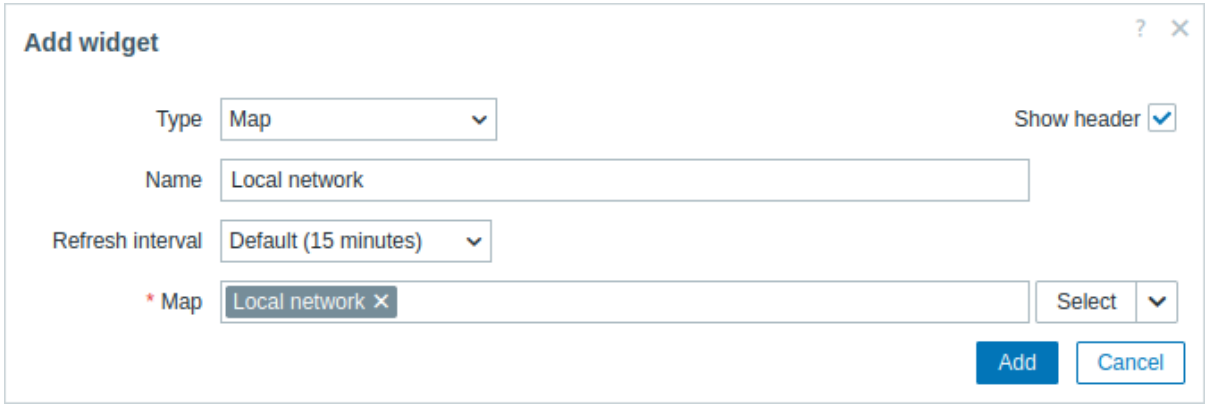
マップウィジェットは、ネットワークマップや類似のビジュアライゼーションを表示し、ネットワークの動的な概要を提供します。



マップナビゲーションツリーウィジェットと組み合わせて、ツリーで選択したマップを動的に表示することもできます。

設定

設定するには、タイプとしてマップを選択します:



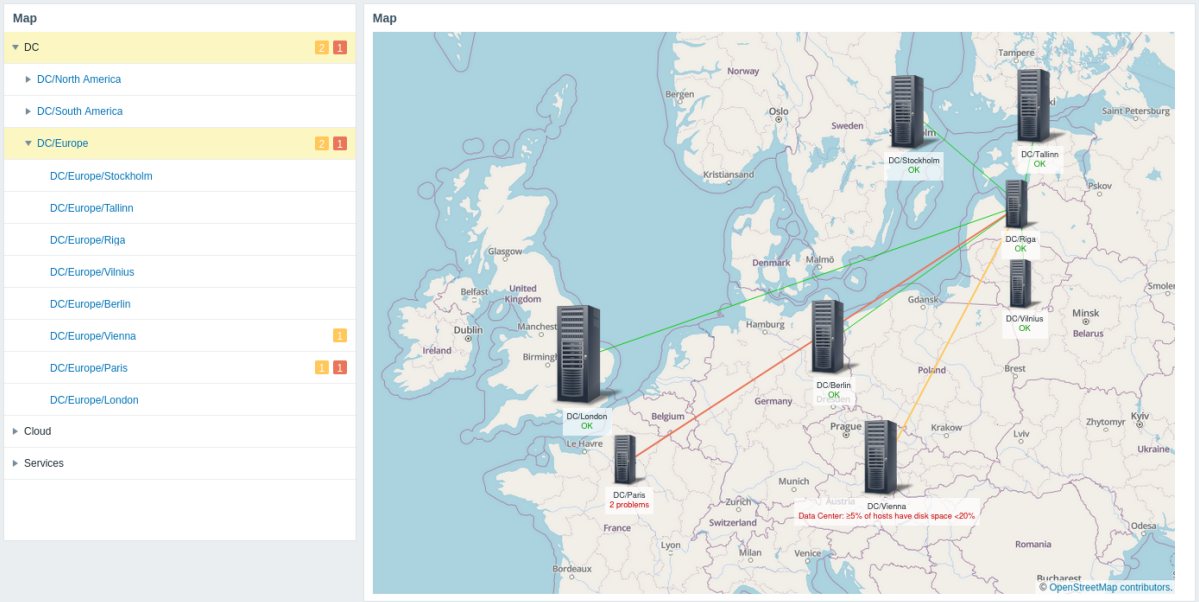
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

マップ	表示するマップを設定します。 または、マップを表示するためのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、マップやウィジェットの名前を入力し始めると、一致するマップやウィジェットのドロップダウンが表示されます。

マップナビゲーションツリー

概要

マップナビゲーションツリーウィジェットは、既存のマップの階層と、各マップおよびマップグループの問題数を表示します。

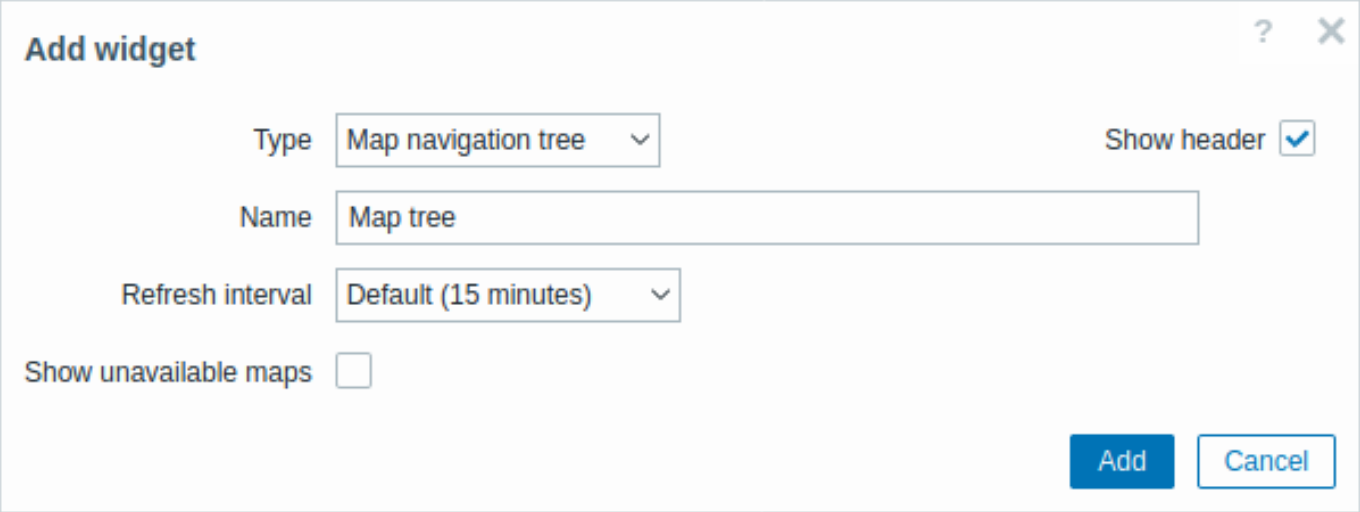


マップウィジェットと組み合わせて、ツリーで選択したマップを動的に表示することができます。

階層内の最上位マップの問題数には、すべてのサブマップの問題と自身の問題が含まれます。

設定

設定するには、タイプとしてマップナビゲーションツリーを選択します:

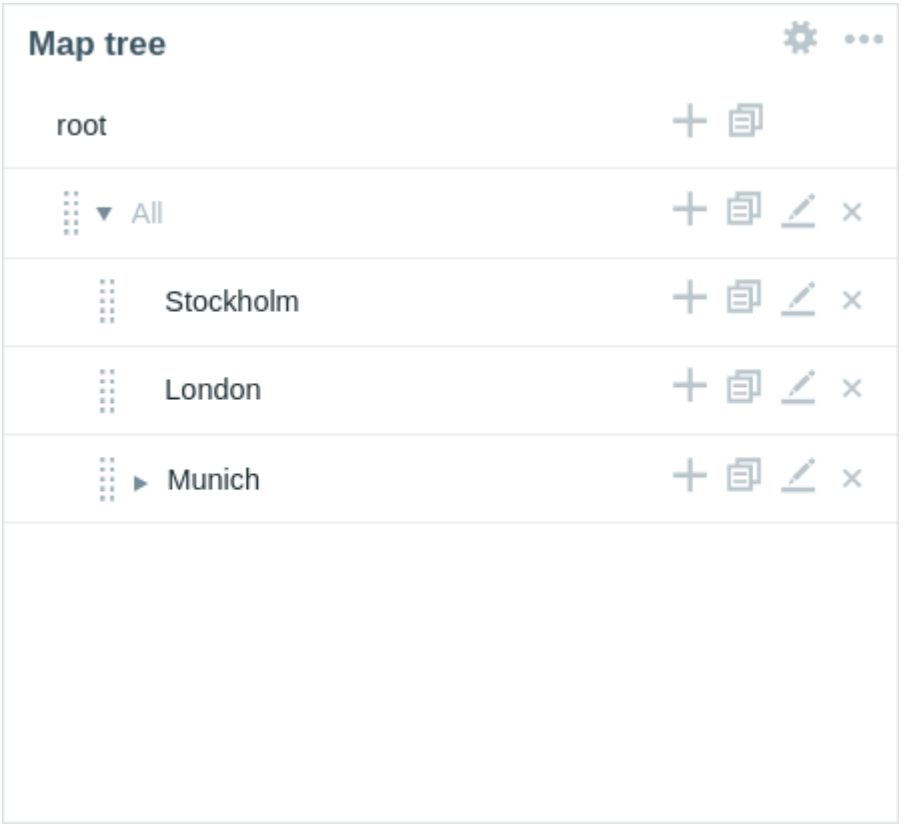


すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

利用できないマップを表示	このチェックボックスをオンにすると、ユーザーが読み取り権限を持たないマップも表示されます。 ナビゲーションツリー内の利用できないマップは、グレーアウトしたアイコンで表示されます。 このチェックボックスをオンにした場合、親レベルのマップが利用できなくても、利用可能なサブマップは表示されます。オフにした場合、利用できない親マップの利用可能なサブマップはまったく表示されません。 問題数は、利用可能なマップおよび利用可能なマップ要素に基づいて計算されます。

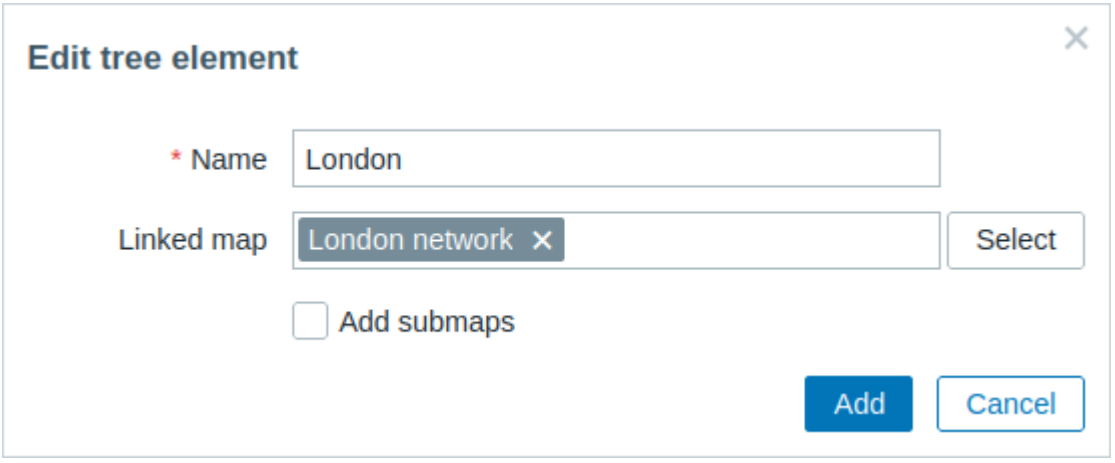
ナビゲーションツリーの要素はリストで表示されます。以下の操作が可能です。

- 要素（子要素を含む）をリスト内の新しい場所にドラッグする
- 要素を展開または折りたたんで、子要素を表示または非表示にする
- 要素に子要素（リンクされたマップの有無にかかわらず）を追加する
- 要素に複数の子要素（リンクされたマップ付き）を追加する
- 要素を編集する
- 要素（子要素を含む）を削除する



要素の設定

ナビゲーションツリー要素を設定するには、新しい要素を追加するか、既存の要素を編集します。



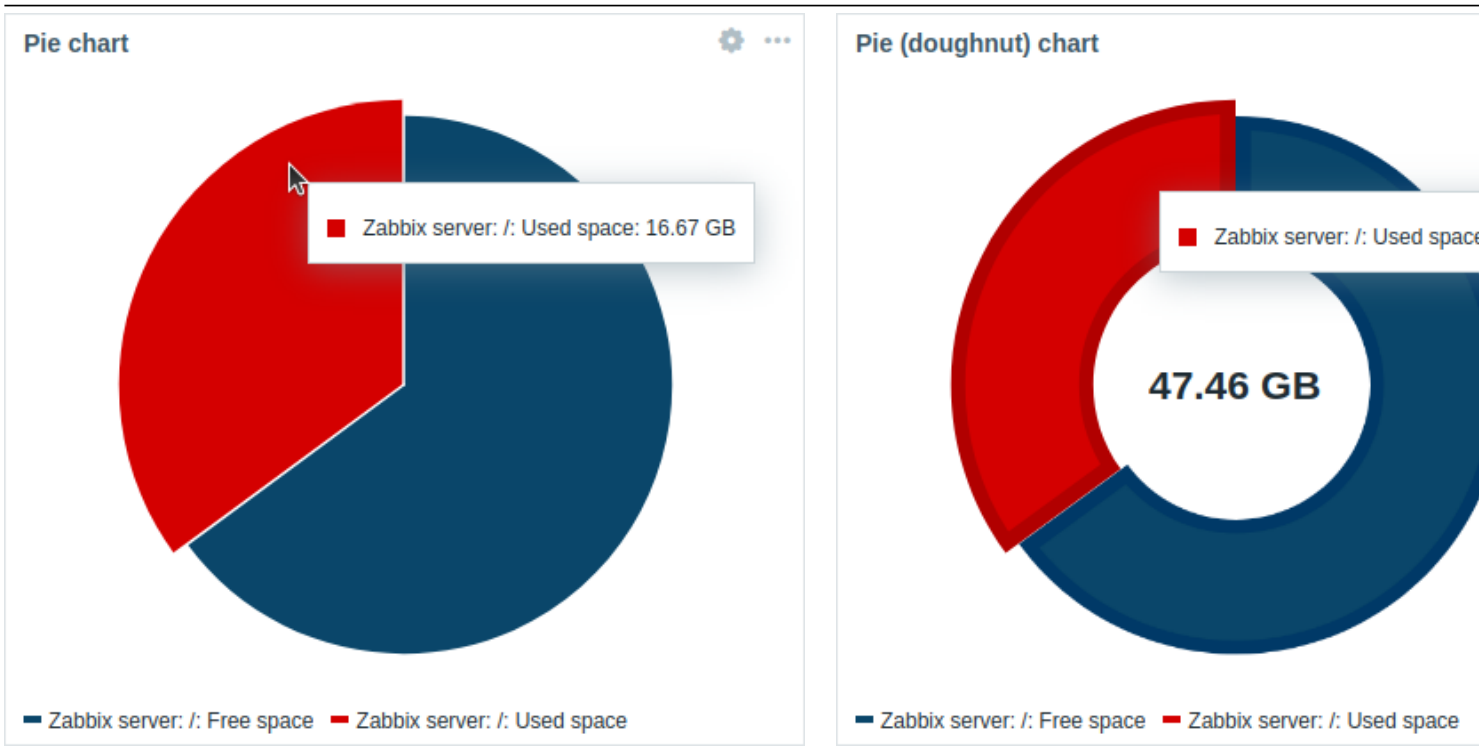
ナビゲーションツリー要素の設定パラメータは以下の通りです。

名前	ナビゲーションツリー要素の名前を入力します。
リンクするマップ	ナビゲーションツリー要素にリンクするマップを選択します。 このフィールドはオートコンプリート対応なので、マップ名の入力を始めると一致するマップのドロップダウンが表示されます。
サブマップを追加	このチェックボックスをオンにすると、リンクされたマップのサブマップをナビゲーションツリー要素の子要素として追加します。

円グラフ

概要

円グラフウィジェットは、数値アイテムデータをベクトルベースの円グラフまたはドーナツグラフとして表示します。このウィジェットを使用すると、アイテムまたはホストがデータセット全体にどの程度寄与しているかを視覚化できます。

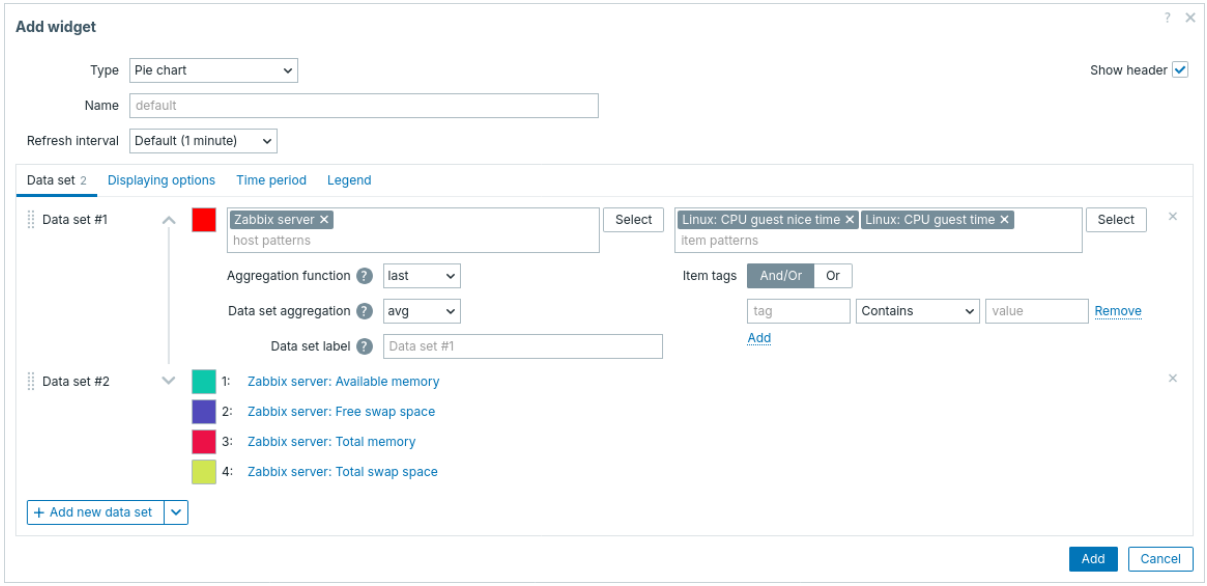


マウスオーバーすると、カーソルを合わせたセクターが拡大し、その凡例が表示されます。セクターをクリックすると、閉じるまで拡大された状態と凡例の表示が維持されます。

円グラフウィジェットに表示される情報は、ウィジェット上にマウスを合わせると表示されるウィジェットのアクションメニュー ... から画像をダウンロードを選択することで、PNG 画像としてダウンロードできます。ブラウザーの設定に応じて、画像は自動的に Downloads フォルダーに保存されるか、保存先を選択できる保存ダイアログが開きます。

設定

設定するには、タイプとして 円グラフを選択します:



データセット

データセットタブでは、データセットを追加して円グラフに表示するデータを選択できます。追加できるデータセットには、次の 2 種類があります。

- アイテムパターン - 一致するアイテムのデータを表示します。単一のベースカラーを選択するか、パレットの行を選択して、一致した各アイテムに異なる色を割り当てることができます。
- アイテムリスト - 選択したアイテムのデータを表示します。カラーピッカーで各アイテムの色を個別に選択できます。

デフォルトでは、アイテムパターンデータセットが追加されます。

データセット	アイテムパターンデータセットの場合: ホストおよびアイテムのパターンを選択または入力します。入力したパターンに一致するアイテムのデータが円グラフに表示されます。表示できるアイテムは最大 50 個です。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上の文字に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* キーを押します。 ワイルドカード記号は常に解釈されるため、他に一致するアイテム（たとえば、*item2*、*item3*）がある場合、*item** という名前のアイテムだけを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストおよびアイテムのパターンを指定する必要があります。 詳細は、データセット設定の詳細を参照してください。 アイテムリストデータセットの場合: アイテムを追加ボタンをクリックして、円グラフに表示するアイテムを選択します。 アイテム名の後にある タイプドロップダウンでは、各アイテムの表示タイプを選択できます。 通常 - アイテムの値を円グラフ上で比例表示します（デフォルト）。 合計 - アイテムの値が円グラフ全体を占めます。1 つの円グラフに「合計」アイテムは 1 つだけ設定でき、円グラフの凡例の先頭に配置されます。アイテムを「合計」に設定すると、データセットの集計パラメータ（下記参照）は無効になり、「使用しない」に設定されます。 ウィジェットを追加ボタンをクリックして、アイテムのデータソースとして互換性のあるウィジェットを選択することもできます。ウィジェットにも個々のアイテムと同じオプションが適用されます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットを指定する必要があります。 詳細は、データセット設定の詳細を参照してください。 数値アイテムタイプのみ使用できます。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは使用できません。また、アイテムリストを指定するパラメータでは、テンプレートで設定されたアイテムのみ選択できます。

集計関数	選択した期間内のデータセットの各アイテムに使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の件数を表示します。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します（デフォルト）。 集計を使用すると、指定した間隔（5 分、1 時間、1 日）に対する集計値を表示できます。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では、数値以外のデータが数値に変換されます。
データセットの集計	データセット全体に使用する集計関数を指定します。 使用しない - 集計せず、アイテムを個別に表示します（デフォルト）。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の件数を表示します。 集計を使用すると、期間タブで選択した、またはダッシュボード全体で使用される指定間隔（5 分、1 時間、1 日）に対する集計値を表示できます。
データセットラベル	データセットのカスタムラベルを指定します。 ラベルは、データセット設定および円グラフの凡例（集計されたデータセットの場合）に表示されます。 データセットラベルを指定したデータセットを含め、すべてのデータセットに番号が付けられます。ラベルを指定しない場合、データセットには番号に基づいて自動的にラベルが付けられます（例: 「データセット #2」、「データセット #3」など）。データセットの番号は、データセットの並べ替えやドラッグ後に再計算されます。 長すぎるデータセットラベルは、表示領域に収まるように短縮されます（例: 「Number of proc...」）。
アイテムタグ	ウィジェットに表示するアイテムをフィルタリングするタグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の照合では、常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる、指定したタグ名を含めます（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる、指定したタグ名を除外します（部分文字列の照合、大文字と小文字を区別しない）。 条件には次の 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

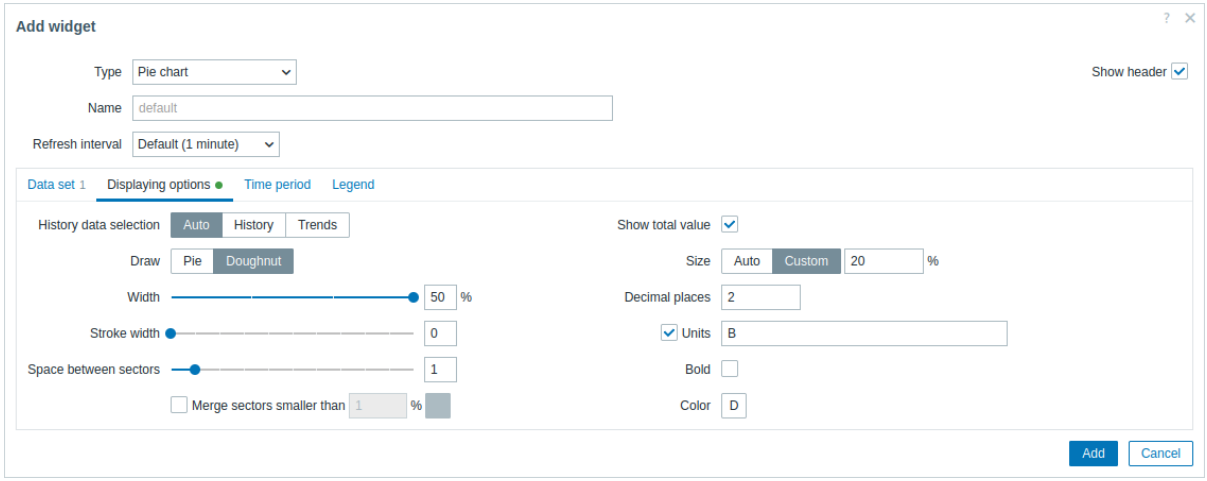
データセットの設定詳細

既存のデータセットはリストで表示されます。これらのデータセットは、並べ替え、展開/折りたたみ、色の変更、クローン作成が可能です。

詳細については、グラフウィジェットのデータセット設定の詳細を参照してください。これらの詳細は円グラフウィジェットにも適用されます。

表示オプション

表示オプションタブでは、円グラフの履歴データの選択および可視化オプションを定義できます。

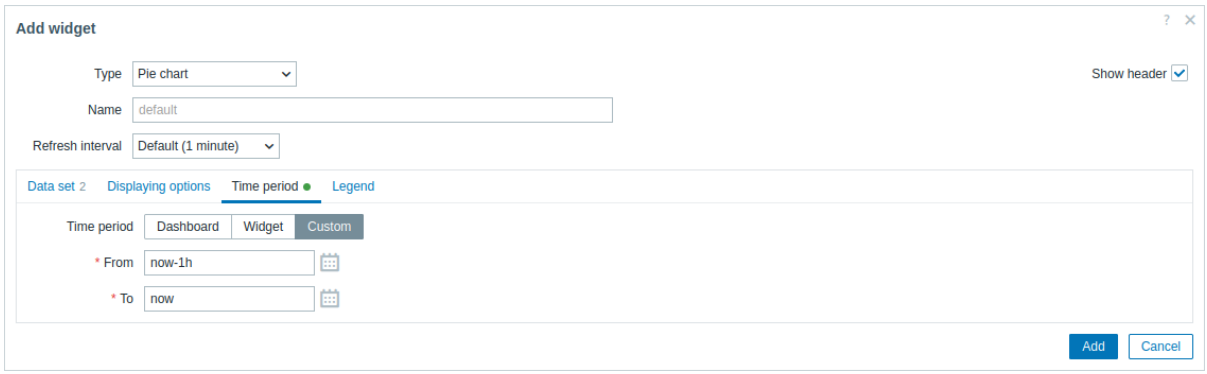


履歴データの選択	データソースを選択します。 自動 - クラシックなアルゴリズムに従ってデータを取得します（デフォルト）。 履歴 - 履歴からデータを取得します。 トレンド - トレンドからデータを取得します。
描画	円グラフの表示スタイルを選択します。 円グラフ - 完全な円グラフ（各セクターが半径の 100% を占めます）。 ドーナツ - 中央が空洞の円グラフ（各セクターが半径の最大 50% を使用します）。
セクター間の間隔	セクター間の間隔のサイズ（0～10 の単位）を選択します（デフォルトは「1」）。
N% 未満のセクターを統合	チェックボックスを選択すると、N% 未満のセクターを統合します。 有効にした場合、統合されたセクターの色と、小さなセクターを統合する割合のしきい値（N）を選択します。
描画スタイル: ドーナツ	
幅	ドーナツの幅を選択します。半径の 20、30、40、または 50%（デフォルト）。
線の幅	ドーナツのセクター境界線の幅（0～10）を選択します。

合計値を表示		チェックボックスを選択すると、ドーナツグラフの中央に合計値を表示します。
	サイズ	合計値のサイズオプションを選択します。 自動 - ドーナツの中央に読みやすく収まるように、テキストサイズを自動的に調整します。 カスタム - ウィジェット全体の高さに対する割合として、テキストサイズを指定します。
	小数点以下の桁数	合計値の小数点以下の桁数（0～6）を指定します。
	単位	合計値の単位を指定します。
	太字	チェックボックスを選択すると、合計値を太字で表示します。
	色	合計値の色を選択します。

時間範囲

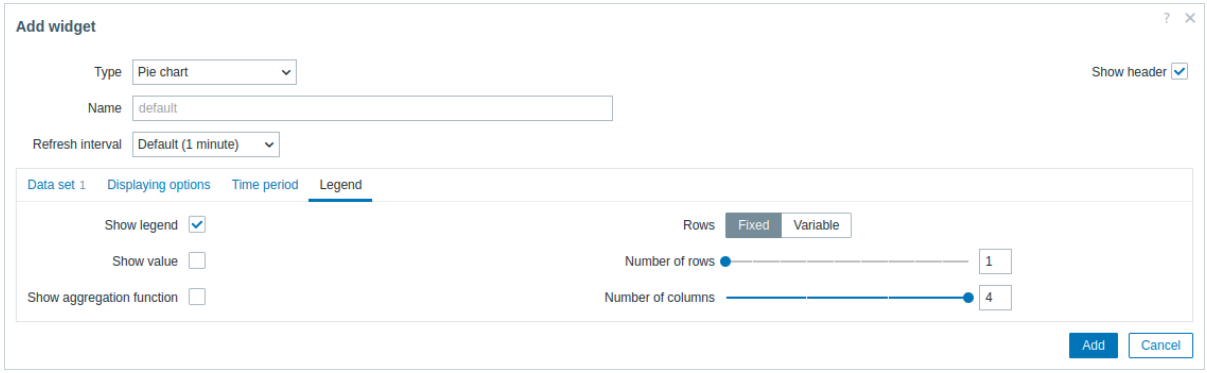
時間範囲タブでは、円グラフの集計設定に使用するカスタム時間範囲を設定できます。



時間範囲	時間範囲のデータソースを選択します。 ダッシュボード - ダッシュボードの時間範囲セレクターを使用します。 ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します。 カスタム - 開始および終了パラメータで指定したカスタム時間範囲を使用します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が表示されます。 ウィジェットの時間範囲設定に関係なく、互換性のあるウィジェットでは、時間範囲のデータソースとして引き続き使用できることに注意してください。
ウィジェット	時間範囲のデータソースとして使用する互換性のあるウィジェット（グラフ、グラフ（クラシック）、グラフのプロトタイプ）を入力または選択します。 このパラメータは、時間範囲が「ウィジェット」に設定されている場合に使用できます。
開始	時間範囲の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、時間範囲が「カスタム」に設定されている場合に使用できます。
終了	時間範囲の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 このパラメータは、時間範囲が「カスタム」に設定されている場合に使用できます。

凡例

凡例タブでは、円グラフの凡例をカスタマイズできます。

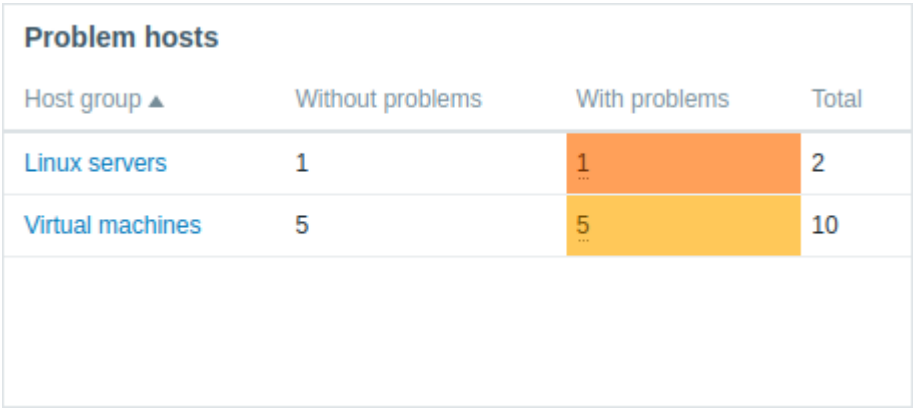


凡例を表示	このチェックボックスの選択を解除すると、円グラフの凡例が非表示になります（デフォルトでは選択されています）。
値を表示	このチェックボックスを選択すると、凡例にアイテムの値が表示されます。
集計関数を表示	このチェックボックスを選択すると、凡例に集計関数が表示されます。
行	凡例の行の表示モードを選択します。 固定 - 表示される行数は、行数パラメータの値によって決まります。 可変 - 表示される行数は、設定されたアイテム数によって決まります。ただし、最大行数パラメータの値を超えることはありません。
行数/ 最大行数	行を「固定」に設定した場合、表示する凡例の行数（1～10）を設定します。 行を「可変」に設定した場合、表示する凡例の最大行数（1～10）を設定します。
列数	表示する凡例の列数（1～4）を設定します。 このパラメータは、値を表示の選択が解除されている場合に使用できます。

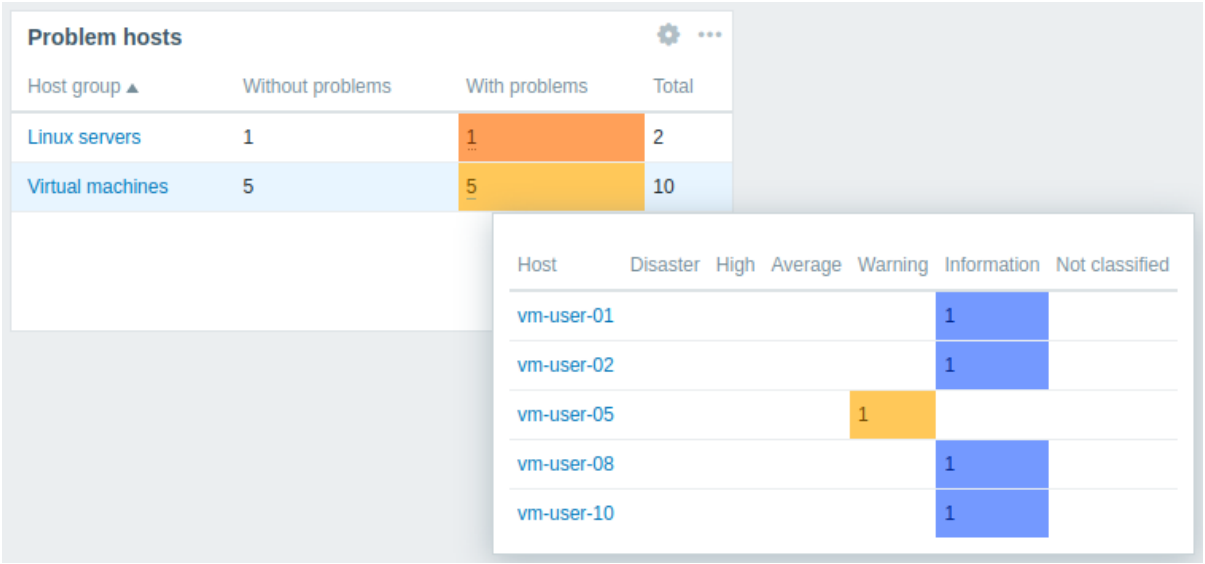
障害ホスト

概要

問題ホストウィジェットは、ホストグループごとの原因となる障害の数を表示し、各グループで最も高い障害の深刻度を示します。さまざまなパラメータ（ホストグループ、ホスト、障害名など）で障害をフィルタリングできます。

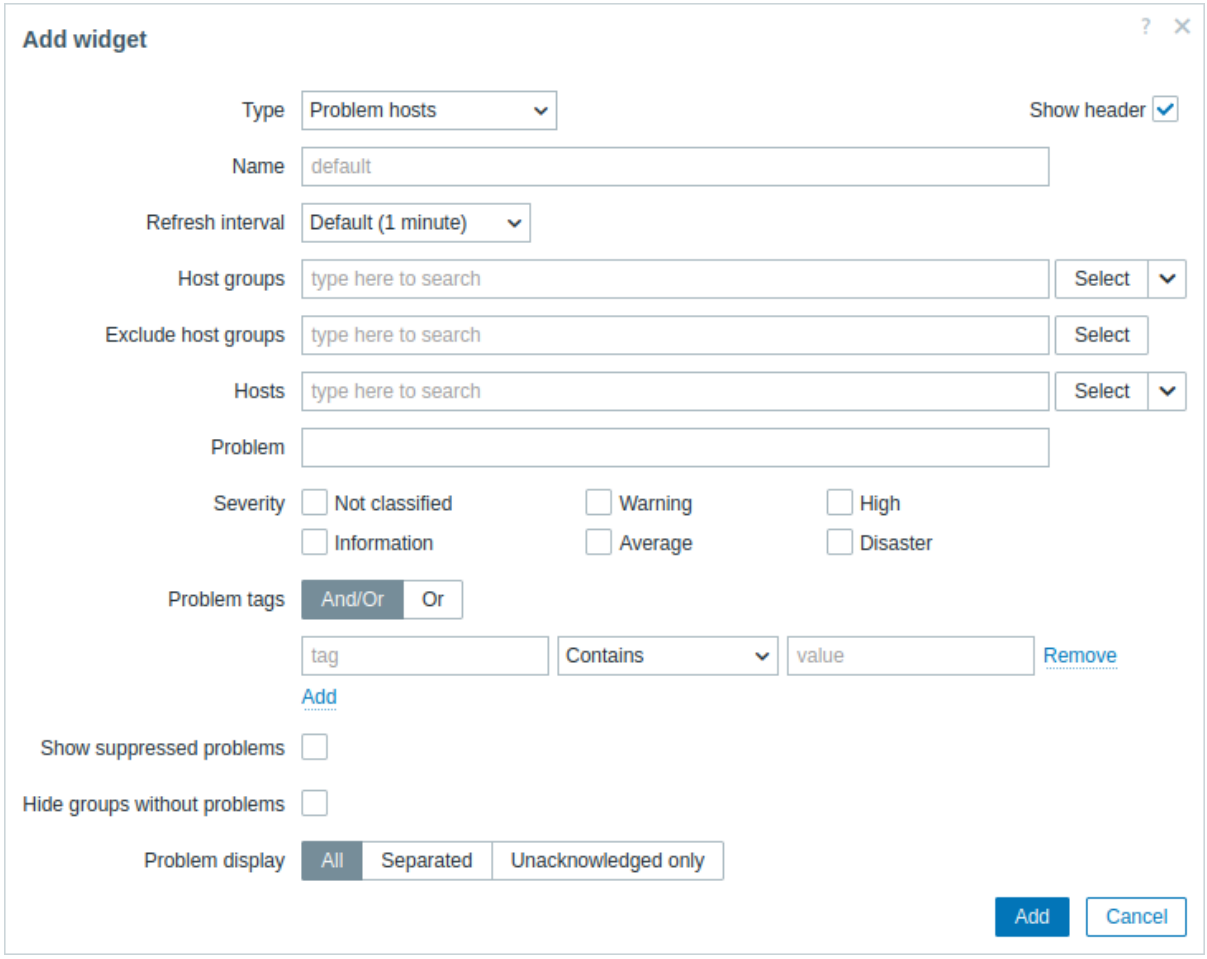


障害数にカーソルを合わせると、障害が発生しているホストのリストが表示されます:



設定

設定するには、タイプとして 問題のあるホストを選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

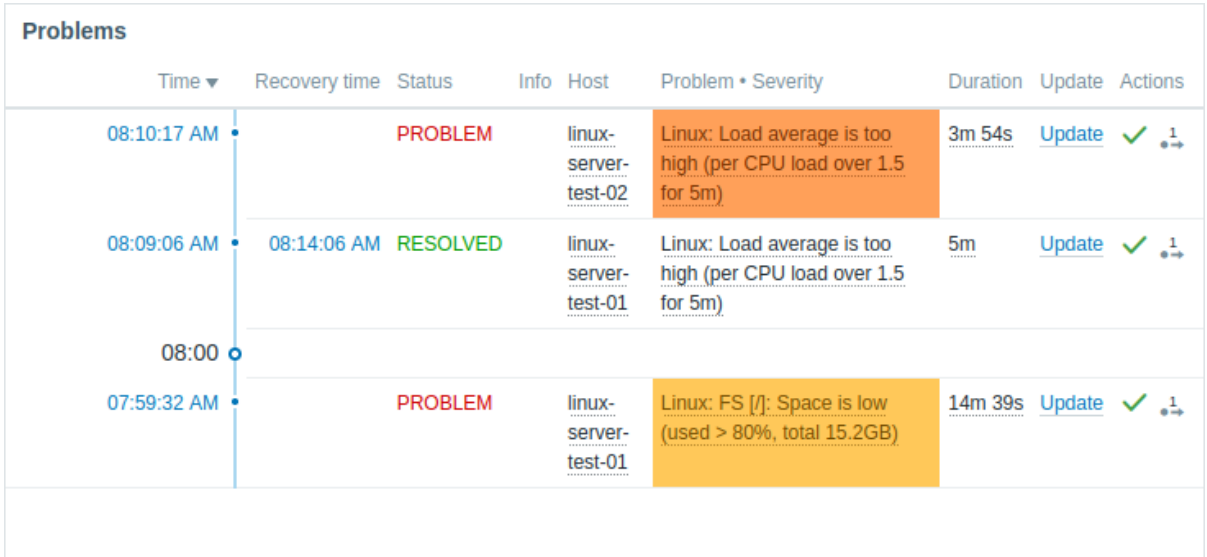
ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストグループの除外	ウィジェットから非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータはウィジェットに表示されません。例えば、ホスト 001、002、003 がグループ A に、ホスト 002、003 がグループ B にも所属している場合、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

問題	問題名で表示する問題ホストの数を制限できます。 ここに文字列を入力すると、名前に入力した文字列が含まれる問題を持つホストのみが表示されます。 マクロは展開されません。
深刻度	ウィジェットに表示する問題をフィルタリングするために、問題の深刻度を選択します。 深刻度を選択しない場合は、すべての問題が表示されます。
問題タグ	ウィジェットに表示する問題の数を制限するために、問題タグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
抑制された問題を表示	ホストのメンテナンスのために抑制（非表示）される問題も表示するには、チェックボックスをオンにします。
問題のないグループを非表示	問題のないグループを非表示オプションを選択すると、ウィジェットで問題のないホストグループのデータが非表示になります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
問題の表示	問題数の表示方法: すべて - 問題の総数を表示; 分離 - 未確認の問題数を、問題の総数の数値として分離して表示; 未確認のみ - 未確認の問題数のみを表示;

障害

概要

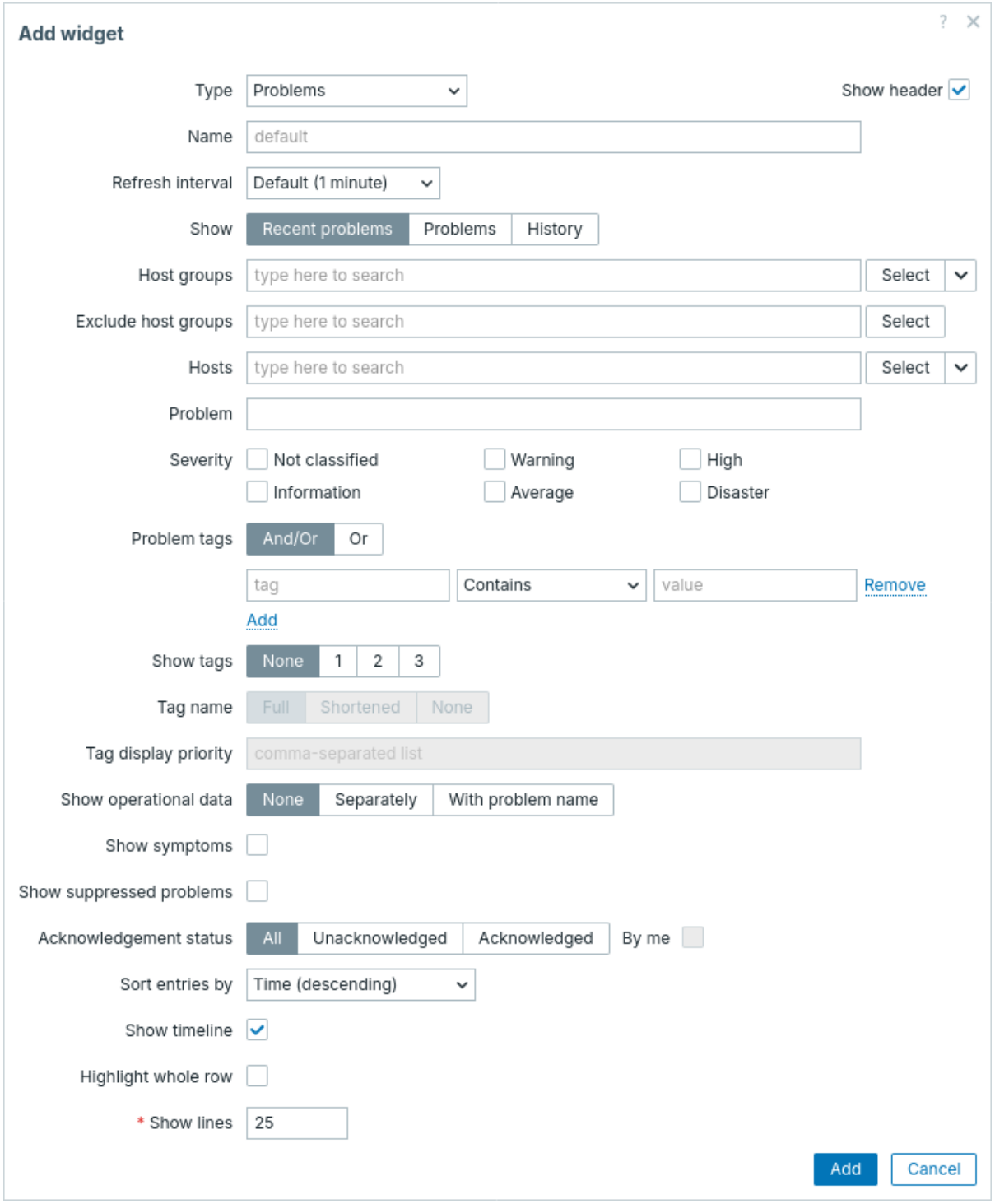
問題ウィジェットは、さまざまなパラメータ（ホストグループ、ホスト、問題名など）でフィルタリングされた現在の問題を表示し、注意が必要な内容を明確に把握できます。



監視 > 問題と同じデータを表示し、最大 1000 件まで表示できます。

設定

設定するには、タイプとして 障害を選択します:



ウィジェットに表示する障害の数は、障害ステータス、障害名、深刻度、ホストグループ、ホスト、イベントタグ、確認ステータスなど、さまざまな方法で制限できます。

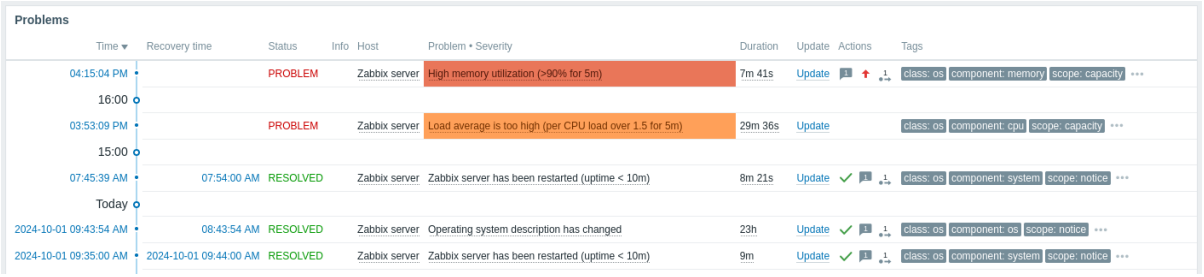
すべてのウィジェットに共通のパラメータに加えて、次の固有オプションを設定できます。

Show	障害ステータスでフィルターします: **Recent problems** - 未解決および最近解決された障害が表示されます（デフォルト）; **Problems** - 未解決の障害が表示されます; **History** - すべてのイベントの履歴が表示されます。

Host groups	ウィジェットに障害を表示するホストグループを選択します。 または、ホストグループのデータソースとして、互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を開始すると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、その配下のすべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループの障害がウィジェットに表示されます。ホストグループを指定しない場合は、すべてのホストグループの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Exclude host groups	ウィジェットから障害を非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を開始すると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、その配下のすべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループの障害は表示されません。たとえば、ホスト 001、002、003 が Group A にあり、ホスト 002、003 が Group B にもあるとします。このとき、Group A を *show* し、同時に Group B を *exclude* すると、ホスト 001 の障害のみがウィジェットに表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Hosts	ウィジェットに障害を表示するホストを選択します。 または、ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応のため、ホスト名の入力を開始すると一致するホストのドロップダウンが表示されます。 ホストを指定しない場合は、すべてのホストの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Problem	名前によって表示する障害数を制限できます。 ここに文字列を入力すると、名前に入力した文字列を含む障害のみが表示されます。 マクロは展開されません。
Severity	ウィジェットに表示する障害をフィルターする深刻度を選択します。 何も選択しない場合は、すべての障害が表示されます。
Problem tags	ウィジェットに表示する障害数を制限するために、障害タグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです: **Exists** - 指定したタグ名を含める; **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別）; **Contains** - タグ値に入力した文字列を含む指定タグ名を含める（部分一致、大文字と小文字を区別しない）; **Does not exist** - 指定したタグ名を除外する; **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別）; **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外する（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。 フィルター適用時、ここで指定したタグは、*Tag display priority*（下記）で上書きされない限り、障害とともに最初に表示されます。
Show tags	表示するタグ数を選択します: **None** - *Tags* 列を表示しません; **1** - *Tags* 列に 1 つのタグを表示します; **2** - *Tags* 列に 2 つのタグを表示します; **3** - *Tags* 列に 3 つのタグを表示します。 障害のすべてのタグを表示するには、3 つのドットのアイコンにマウスを重ねます。

Tag name	タグ名の表示モードを選択します: **Full** - タグ名と値を完全に表示します; **Shortened** - タグ名は 3 文字に短縮されますが、タグ値は完全に表示されます; **None** - タグ値のみを表示し、名前は表示しません。
Tag display priority	障害に対するタグの表示優先順位を、カンマ区切りのタグ一覧として入力します。 値ではなく、タグ名のみを使用してください。 例: `Services,Applications,Application` この一覧のタグは、アルファベット順の自然な並び順を上書きして、常に最初に表示されます。
Show operational data	運用データの表示モードを選択します: **None** - 運用データを表示しません; **Separately** - 運用データを別の列に表示します; **With problem name** - 運用データを括弧付きで障害名に追加します。 詳細は、トリガー設定の *Operational data* フィールドの説明を参照してください。
Show symptoms	チェックボックスをオンにすると、症状として分類された障害を独自の行に表示します。
Show suppressed problems	チェックボックスをオンにすると、ホストメンテナンスまたは単一の障害抑制により本来は抑制される（表示されない）障害を表示します。
Acknowledgement status	すべての障害、未確認の障害のみ、または確認済みの障害のみを表示するようにフィルターします。 追加のチェックボックスをオンにすると、自分が過去に確認した障害を除外できます。
Sort entries by	次の基準でエントリを並べ替えます: **Time**（降順または昇順）; **Severity**（降順または昇順）; **Problem name**（降順または昇順）; **Host**（降順または昇順）。 テンプレートダッシュボードでウィジェットを設定する場合、**Host**（降順または昇順）による並べ替えは使用できません。
Show timeline	チェックボックスをオンにすると、視覚的なタイムラインを表示します。
Highlight whole row	チェックボックスをオンにすると、未解決の障害の行全体を強調表示します。強調表示には障害の深刻度の色が使用されます。 *Highlight whole row* は高コントラストテーマでは使用できません。
Show lines	表示する障害行数を指定します。

ウィジェットの使用

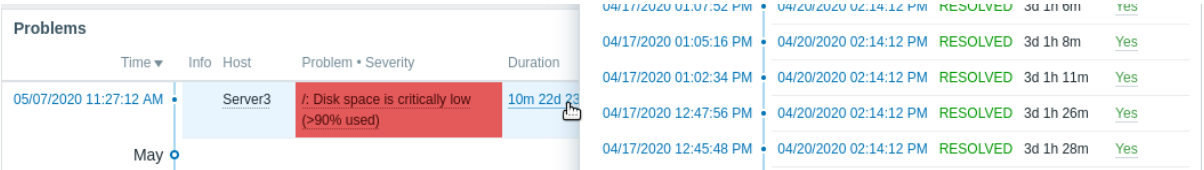


障害ウィジェットでは、追加情報にすばやくアクセスできます。

- 問題の日付（Time 列）または復旧日（Recovery Time 列）をクリックすると、イベント詳細を表示できます。
- Info フィールドが空でない場合は、表示されているアイコンにマウスオーバーすると追加詳細を表示できます。
- ホスト名をクリックすると、ホストメニューを開けます。
- 障害名をクリックすると、イベントメニューを開けます。
- 障害の継続時間にマウスオーバーするかクリックすると、障害イベントポップアップを表示できます。
- *Update* をクリックすると、障害の更新ウィンドウが開きます。
- Actions 列の灰色の矢印アイコンにマウスオーバーするかクリックすると、実行されたアクションの一覧を表示できます。

障害イベントポップアップ

障害イベントポップアップには、このトリガーに対する障害イベントの一覧と、定義されている場合はトリガーの説明およびクリック可能な URL が含まれます。



障害イベントポップアップを表示するには、次のいずれかを行います。

- *Problems* ウィジェットの *Duration* 列にある障害の継続時間にマウスを重ねます。マウスを継続時間から外すと、ポップアップは消えます。
- *Problems* ウィジェットの *Duration* 列にある継続時間をクリックします。ポップアップは、もう一度継続時間をクリックした場合にのみ消えます。

深刻度別の障害

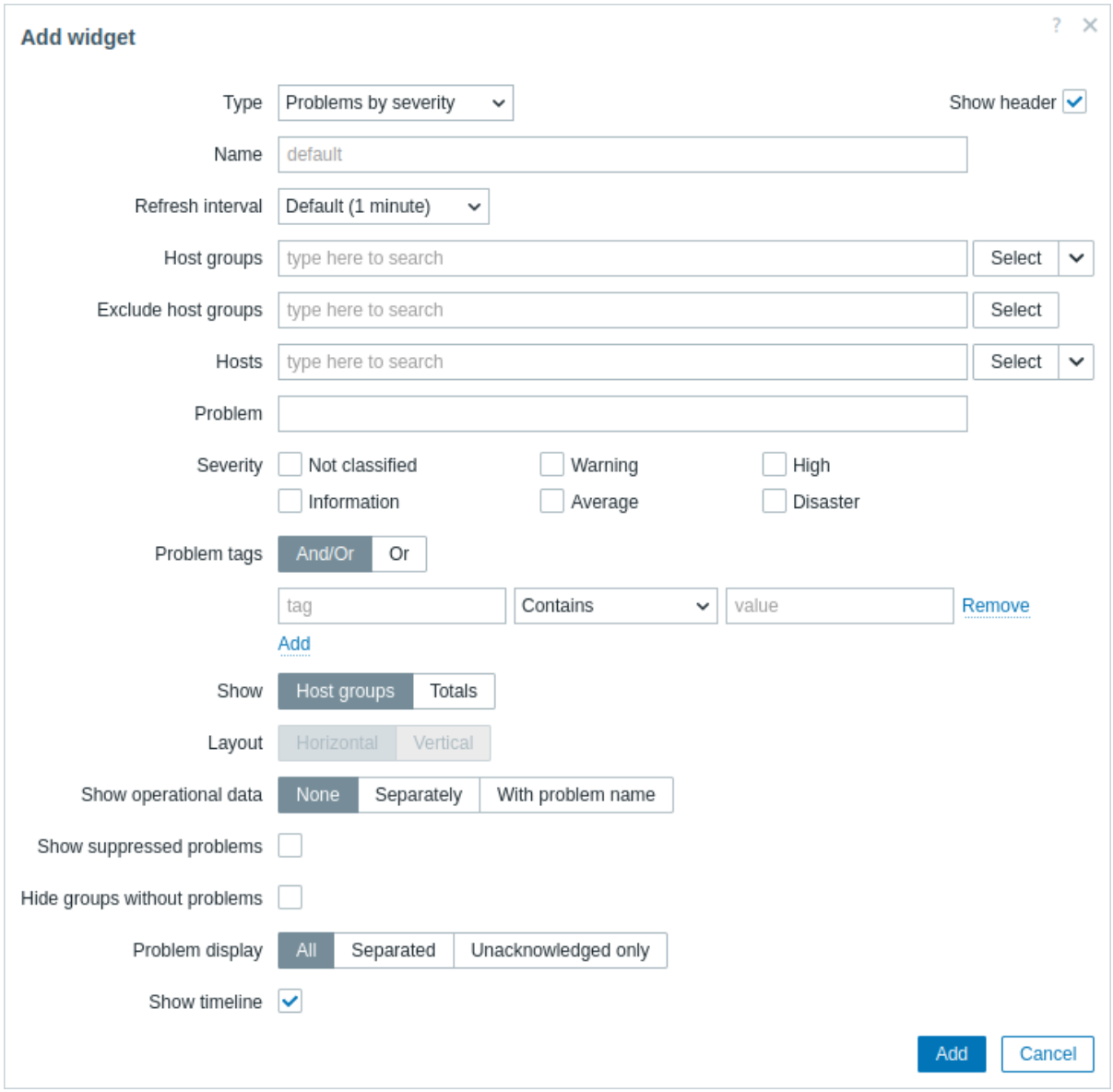
概要

重大度別の問題ウィジェットは、重大度ごとにグループ化された原因となる問題の数を表示します。さまざまなパラメータ（ホストグループ、ホスト、問題名など）で問題をフィルタリングできます。

問題数にカーソルを合わせると、関連する問題のリストが表示されます:

設定

設定するには、タイプとして 深刻度別の障害を選択します:



すべてのウィジェットに共通の共通パラメータに加えて、次の固有オプションを設定できます:

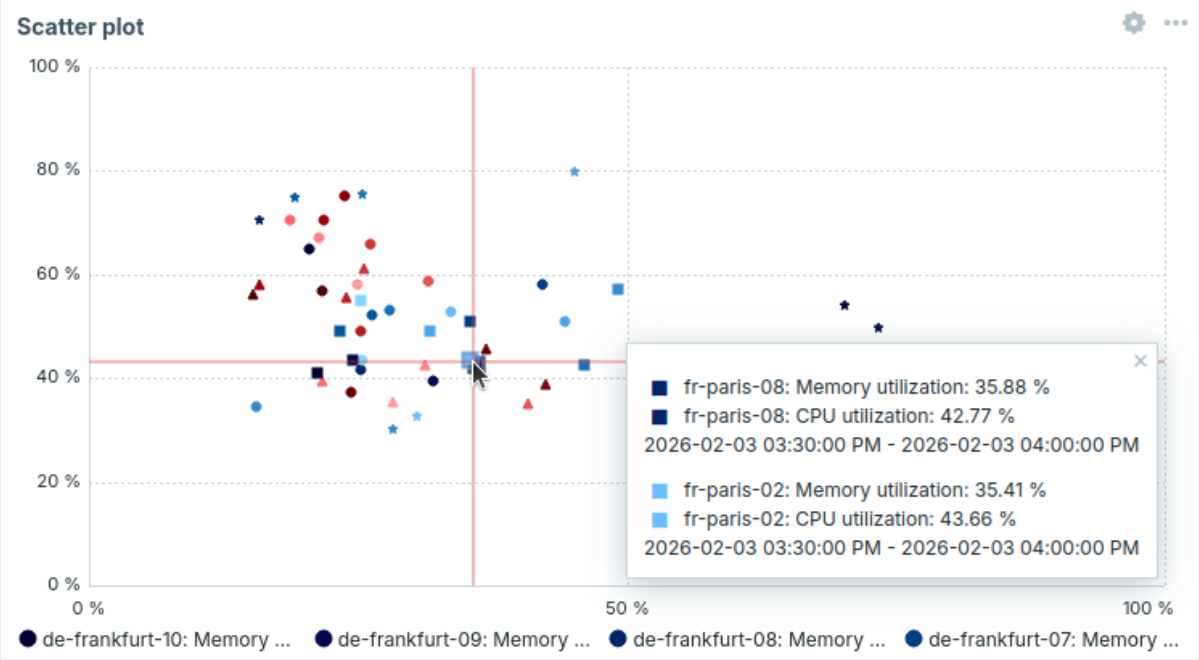
ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、配下のすべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
除外するホストグループ	ウィジェットから非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、配下のすべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータは表示されません。たとえば、ホスト 001、002、003 が Group A にあり、ホスト 002、003 が Group B にもあるとします。このとき、Group A を表示し、同時に Group B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応のため、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害	障害ホストの表示数を障害名で制限できます。 ここに文字列を入力すると、名前に入力した文字列を含む障害を持つホストのみが表示されます。 マクロは展開されません。
深刻度	ウィジェットに表示する障害をフィルタリングするために、障害の深刻度を選択します。 深刻度が選択されていない場合は、すべての障害が表示されます。
障害タグ	ウィジェットに表示する障害数を制限するために障害タグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです: **Exists** - 指定したタグ名を含める; **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別）; **Contains** - タグ値に入力した文字列を含むタグ名を含める（部分一致、大文字と小文字を区別しない）; **Does not exist** - 指定したタグ名を除外する; **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別）; **Does not contain** - タグ値に入力した文字列を含むタグ名を除外する（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
表示	表示オプションを選択します: ホストグループ - ホストグループごとに障害を表示します; 合計 - 選択したすべてのホストグループの障害合計を、障害の深刻度に対応する色付きブロックで表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できず、障害合計のみが表示されます。
レイアウト	レイアウトオプションを選択します: **Horizontal** - 合計の色付きブロックを横方向に表示します; **Vertical** - 合計の色付きブロックを縦方向に表示します。 このパラメータは、表示が"Totals" に設定されている場合に使用できます。

運用データを表示	運用データの表示方法を選択します: **None** - 運用データを表示しません **Separately** - 運用データを別の行として表示します **With problem name** - 運用データを障害名と一緒に表示します。 詳細は、トリガー設定の 運用データフィールドの説明を参照してください。
抑制された障害を表示	ホストメンテナンスのために抑制される（表示されない）障害を表示するには、このチェックボックスをオンにします。
障害のないグループを隠す	障害のないグループを隠すオプションをオンにすると、ウィジェット内で障害のないホストグループのデータを非表示にできます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害表示	障害数の表示方法を選択します: **All** - 障害の総数を表示します; **Separated** - 未確認の障害数を、障害総数とは別に数値として表示します; **Unacknowledged only** - 未確認の障害数のみを表示します。
タイムラインを表示	視覚的なタイムラインを表示するには、このチェックボックスをオンにします。

散布図

概要

散布図ウィジェットは、X 軸と Y 軸に個々のデータポイントをプロットして、2 つのメトリクス間の関係を表示します。これにより、データセット内のパターン、クラスター、相関関係、外れ値を明らかにできます。



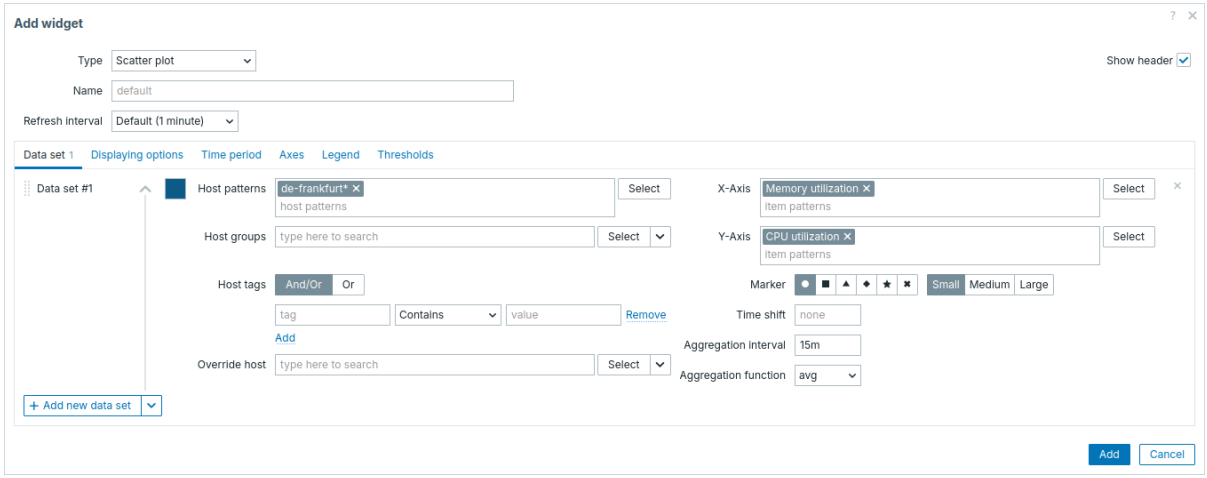
データポイントにカーソルを合わせると、そのデータポイントおよび半径 6px 以内にある他のデータポイントのアイテム一覧を含むツールチップが表示されます。一覧は、まず散布図の X 軸の値の昇順、次に Y 軸の値の降順で並べ替えられます。

ツールチップ内のアイテムを選択すると、そのデータが他のウィジェットにブロードキャストされます。

散布図ウィジェットに表示される情報は、ウィジェット上にカーソルを合わせると表示されるウィジェットのアクションメニュー ... から画像をダウンロードを選択して、PNG 画像としてダウンロードできます。ブラウザーの設定に応じて、画像は自動的に `Downloads` フォルダーに保存されるか、保存場所を選択できる保存ダイアログが開きます。

設定

設定するには、タイプとして散布図を選択します:



データセット

データセットタブでは、データセットを追加して散布図に表示するデータを選択できます。次の 2 種類のデータセットを追加できます。

- アイテムパターン - 一致するアイテムを持つホストがデータポイントとして表示されます。単一のベースカラーを選択するか、パレットの行を選択して、一致するアイテムを持つ各ホストに異なる色を割り当てることができます。
- アイテムリスト - 選択したアイテムがデータポイントとして表示されます。各ポイントはアイテムのペア（X と Y）を表します。各データセットの色は、カラーピッカーから個別に選択できます。

データポイントは、両方の軸に少なくとも 1 つの値があるホストとアイテムのペアに対してのみ作成されます。

デフォルトでは、アイテムパターンデータセットが追加されます。

データセット	アイテムパターンデータセットの場合: ホストとアイテムのパターンを選択または入力します。これらのパターンに一致するアイテムのデータが散布図に表示されます。データポイントの数が 8,000 を超える場合は、`php.ini` 設定ファイルの`memory_limit`を増やす必要があります。データポイントが 20,000 の場合、推奨されるメモリ制限は 512 MB です。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。 ワイルドカード記号は常に解釈されるため、たとえば、他に一致するアイテム（*item2*、*item3*）がある場合に、*item** という名前のアイテムだけを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストとアイテムのパターンを指定する必要があります。 詳細については、データセット設定の詳細を参照してください。 アイテムリストデータセットの場合: アイテムを追加ボタンをクリックして、散布図に表示するアイテムを選択します。 ウィジェットを追加ボタンをクリックして、アイテムのデータソースとして互換性のあるウィジェットを選択することもできます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットを指定する必要があります。 詳細については、データセット設定の詳細を参照してください。 数値アイテムタイプのみ使用できます。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは使用できません。また、アイテムリストを指定するパラメータでは、テンプレートに設定されたアイテムのみ選択できます。
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータはアイテムパターンデータセットでのみ使用でき、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

ホストタグ	ウィジェットに表示するホストをフィルターするタグを指定します。 特定のタグおよびタグ値を含めたり除外したりできます。複数の条件を設定できます。 タグ名の照合では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - 指定したタグ名を含め、タグ値に入力した文字列が含まれるものを対象にします（部分文字列の照合、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - 指定したタグ名を除外し、タグ値に入力した文字列が含まれるものを対象外にします(部分文字列の照合、大文字と小文字を区別しない)。 条件の計算タイプには次の 2 種類があります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - 1 つの条件を満たせば十分です。 このパラメータはアイテムパターンデータセットでのみ使用でき、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホストの上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードのホストセレクターを選択します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータは使用できません。
マーカー	データセット内のホストデータポイントに使用するマーカーのタイプとサイズを選択します。
時間シフト	必要に応じて時間シフトを指定します。 このフィールドでは時間のサフィックスを使用できます。負の値も使用できます。
集計関数	選択した集計間隔内で、データセット内の各アイテムに使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の数を表示します（両方の軸に少なくとも 1 つの値がある場合）。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します。 集計を使用すると、すべての値ではなく、指定した間隔（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では、数値以外のデータが数値に変換されます。
集計間隔	値を集計する間隔を指定します。 このフィールドでは時間のサフィックスを使用できます。サフィックスのない数値は秒として扱われます。 ウィジェットがトレンドに基づいて履歴データを表示するように設定されている場合（履歴データの選択がトレンドまたは自動に設定されている場合）、1 時間の倍数となる集計間隔（3600、60m、1h、3h など）を使用することを推奨します。トレンドには 1 時間ごとの集計値が保存されるため、1 時間の倍数ではない集計間隔（100s、7min、15min、90min など）を使用すると、解釈が難しい結果になる場合があります。 集計間隔の指定は必須です。 集計間隔によって、データポイントの時間間隔と座標が決まります。たとえば、X 軸のアイテムに 4 つの値があり、Y 軸のアイテムに 1 つの値がある場合、座標は指定した間隔内で X 軸の 4 つの値と Y 軸の 1 つの値を集計した結果を反映します。

データセットの設定の詳細

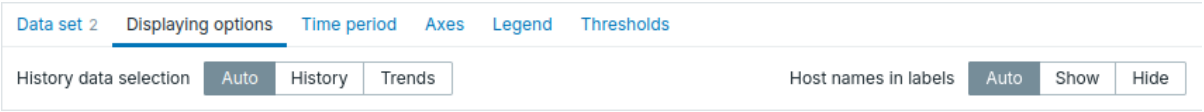
既存のデータセットはリストに表示されます。これらのデータセットは、並べ替え、展開/折りたたみ、色の変更、クローン作成が可能です。

詳細については、グラフウィジェットのデータセット設定の詳細を参照してください。これらの詳細は散布図ウィジェットにも適用

されます。

表示オプション

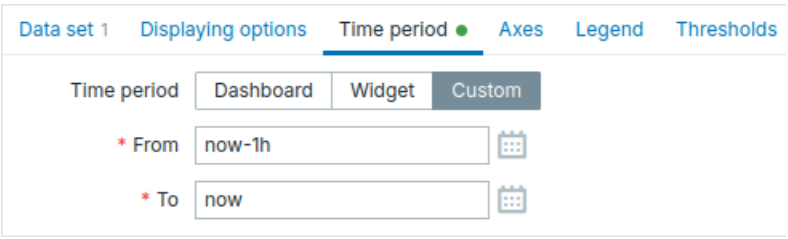
表示オプションタブでは、履歴データの選択を定義できます。



履歴データの選択	散布図データのソースを設定します。 自動 - クラシックグラフのアルゴリズムに従ってデータを取得します（デフォルト）。 履歴 - 履歴からデータを取得します。 トレンド - トレンドからデータを取得します。
ラベル内のホスト名	グラフの凡例およびグラフデータにカーソルを合わせたときに表示されるツールチップに、ホスト名を表示するかどうかを選択します。 自動 - データセット全体に複数のホストが存在する場合にのみホスト名を表示します（デフォルト）。 表示 - ホスト名を表示します。 非表示 - ホスト名を非表示にします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

期間

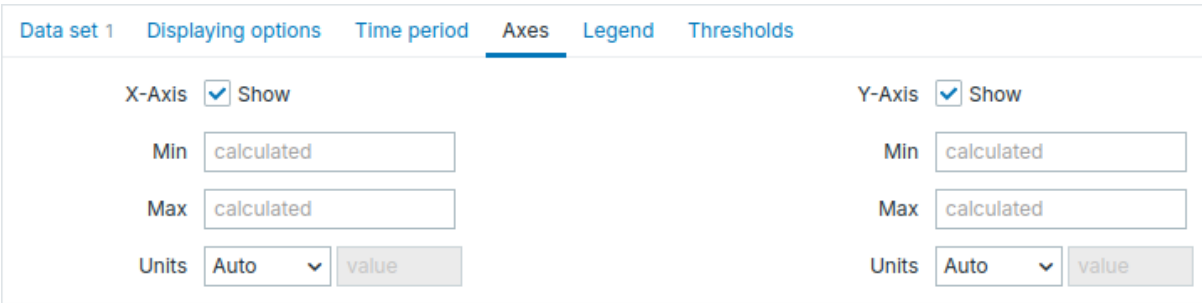
期間タブでは、散布図にデータを表示する期間を設定できます。



期間	期間のdata sourceを選択します。 ダッシュボード - ダッシュボードの期間セレクターを使用します。 ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用します。 カスタム - 開始および終了パラメータで設定したカスタム期間を使用します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻を示します。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットでは、期間のデータソースとして引き続き使用できることに注意してください。
ウィジェット	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 期間が「ウィジェット」に設定されている場合に使用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 期間が「カスタム」に設定されている場合に使用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）がサポートされています。 期間が「カスタム」に設定されている場合に使用できます。

軸

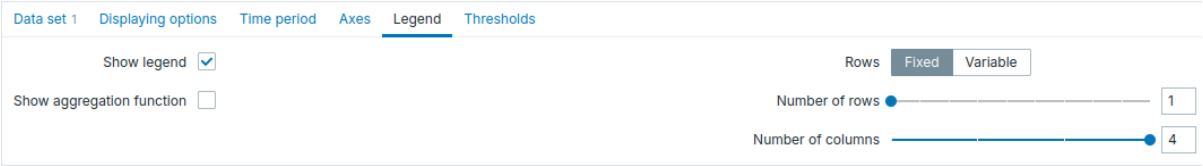
軸タブでは、軸の表示方法をカスタマイズできます。



X 軸/*Y* 軸	このチェックボックスの選択を解除すると、X 軸/Y 軸が非表示になります（デフォルトでは選択されています）。
最小	対応する軸の最小値を設定します。
最大	対応する軸の最大値を設定します。
単位	ドロップダウンから散布図の軸の値に使用する単位を選択します。 自動 - データセット内の最初のアイテムの単位を使用して軸の値を表示します。 静的 - *value* 入力フィールドで指定した単位を使用して軸の値を表示します。フィールドを空白のままにすると、数値のみが表示されます。

凡例

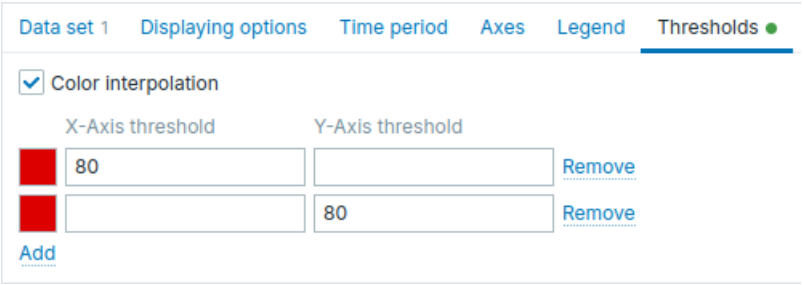
凡例タブでは、散布図の凡例をカスタマイズできます。



凡例を表示	このチェックボックスの選択を解除すると、散布図の凡例が非表示になります（デフォルトでは選択されています）。
集計関数を表示	このチェックボックスを選択すると、凡例に集計関数が表示されます。
行	凡例の行の表示モードを選択します。 固定 - 表示される行数は、行数パラメータの値によって決まります。 可変 - 表示される行数は、設定されたアイテムの数によって決まります。ただし、最大行数パラメータの値を超えることはありません。
行数/ 最大行数	行を「固定」に設定した場合、表示する凡例の行数を設定します（1～10）。 行を「可変」に設定した場合、表示する凡例の最大行数を設定します（1～10）。
列数	表示する凡例の列数を設定します（1～4）。 このパラメータは、最小値/平均値/最大値を表示が選択されていない場合に使用できます。

しきい値

しきい値タブでは、ホストマーカーの色を決定するしきい値を設定できます。

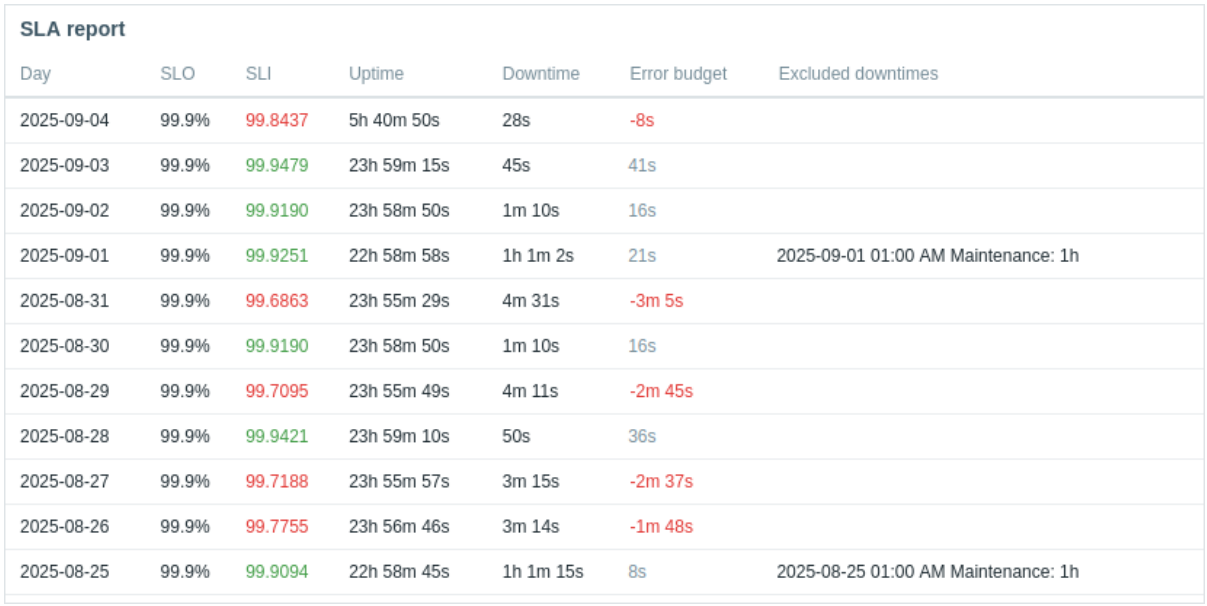


色の補間	チェックボックスを選択すると、ホストマーカーのしきい値間の色が滑らかに変化します。 このパラメータは、X 軸、Y 軸、または両方の軸に 2 つ以上のしきい値が設定されている場合に適用されます。
X 軸/*Y* 軸のしきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択して、数値を指定します。 しきい値は、1 つの軸だけ、X 軸と Y 軸に個別、または両方の軸に同時に適用できます。 たとえば、X=80 および Y=80 の単一のしきい値は、両方の条件を満たした場合にマーカーを着色します。 保存時に、しきい値の一覧は昇順に並べ替えられます。 サフィックス（例: ”1d”、”2w”、”4K”、”8G”）がサポートされています。値のマッピングがサポートされています。

SLA レポート

概要

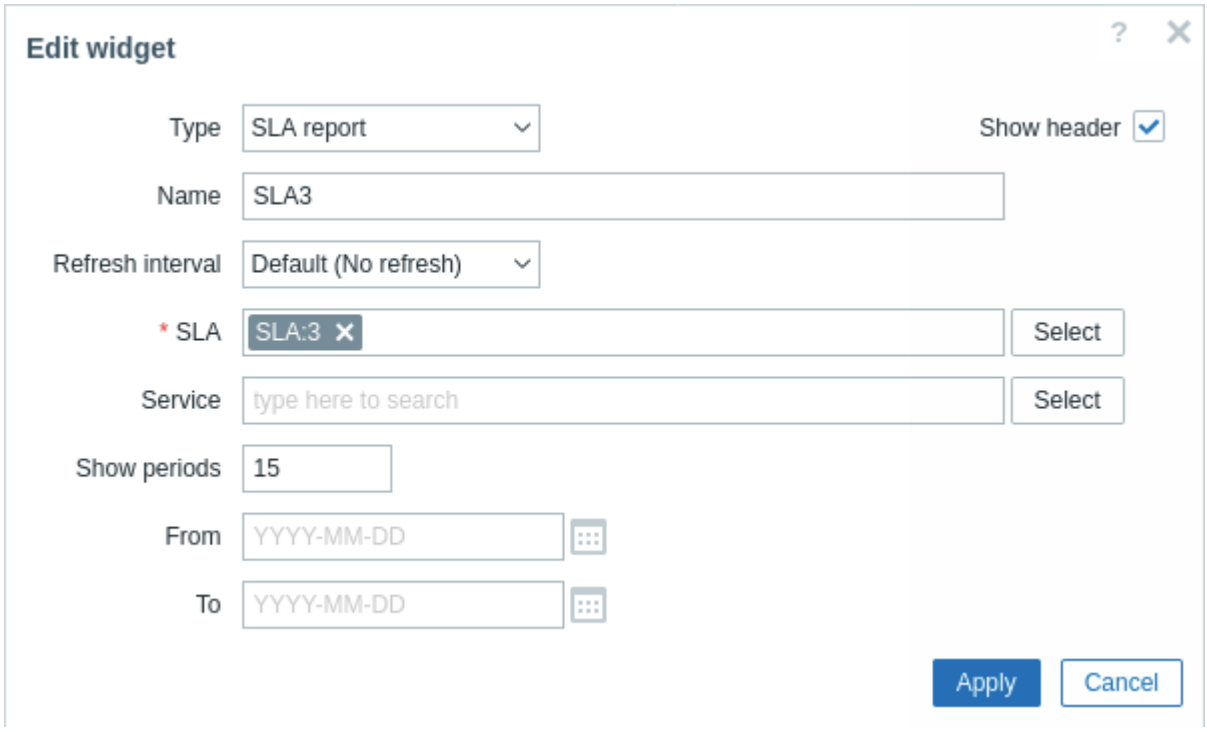
SLA レポートウィジェットは、SLA レポートを表示し、定義された目標に対するサービスのパフォーマンスを監視するのに役立ちます。



サービス > *SLA* レポートと同じデータを表示します。

設定

設定するには、タイプとして *SLA* レポートを選択します:



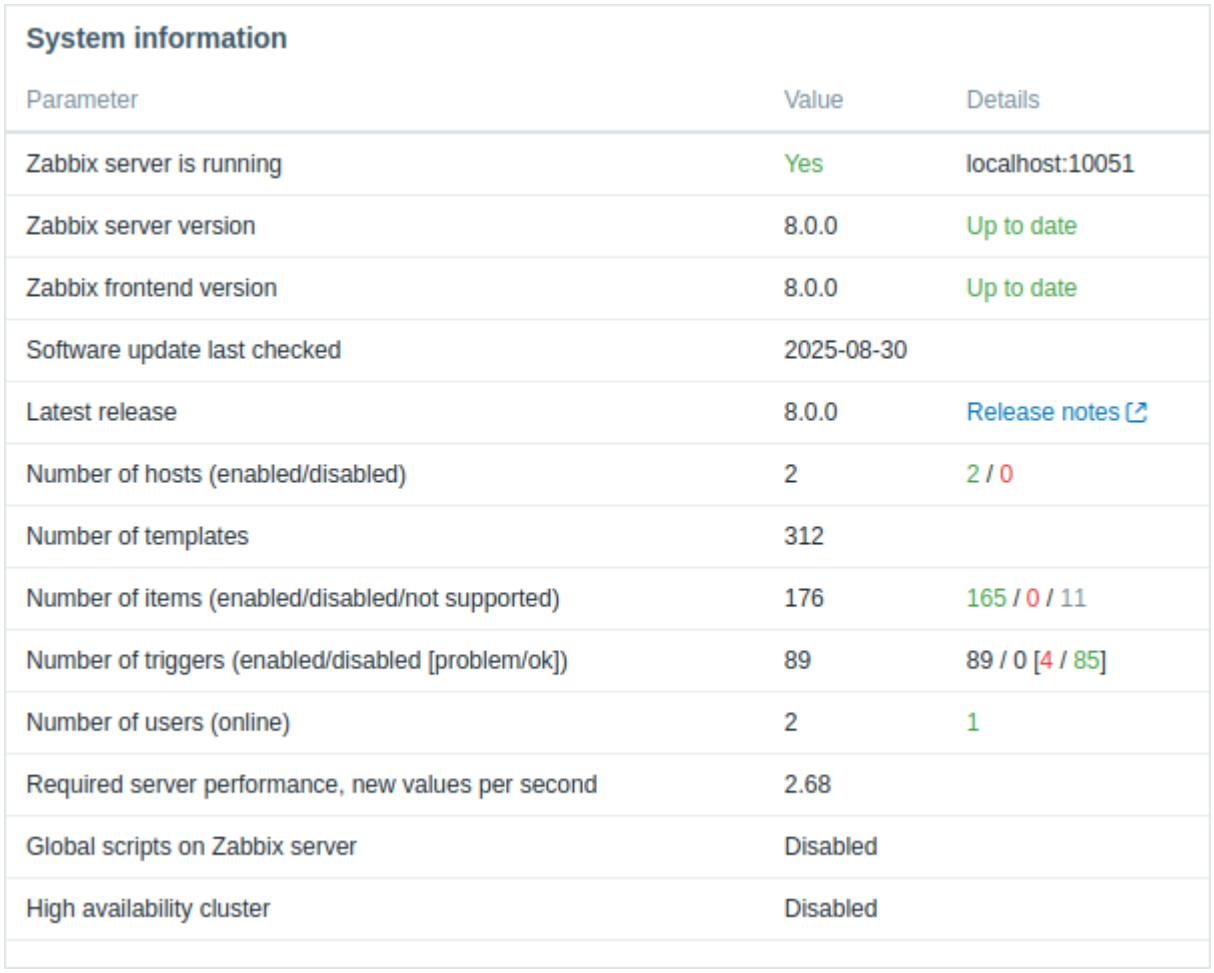
すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

SLA	レポートの SLA を選択します。
サービス	レポートのサービスを選択します。
期間の表示	ウィジェットに表示する期間の数を設定します (デフォルトは 20、最大 100)。
開始	レポートの開始日を選択します。 相対日付がサポートされています: `now`, `now/d`, `now/w-1w` など; サポートされている日付修飾子: `d`, `w`, `M`, `y`。
終了	レポートの終了日を選択します。 相対日付がサポートされています: `now`, `now/d`, `now/w-1w` など; サポートされている日付修飾子: `d`, `w`, `M`, `y`。

システム情報

概要

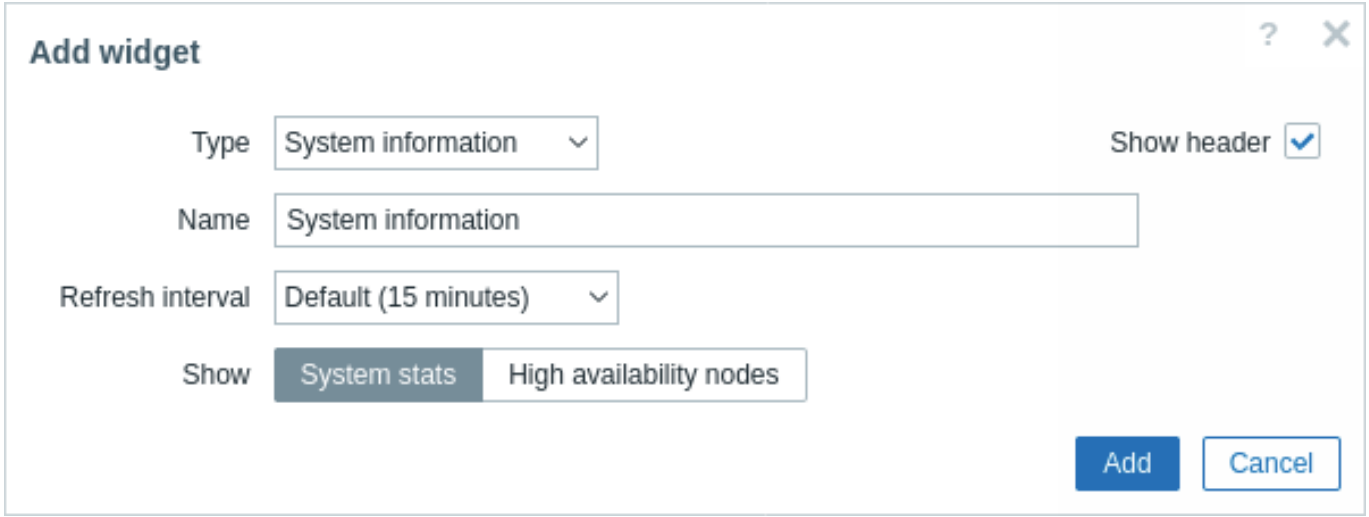
System information ウィジェットは、主要な Zabbix サーバーおよびシステムデータの概要を表示し、高可用性が有効な場合は高可用性ノードの詳細も表示します。



Reports > *System information* と同じデータを表示しますが、*Zabbix server ID* は除きます。

設定

設定するには、タイプとして システム情報を選択します。



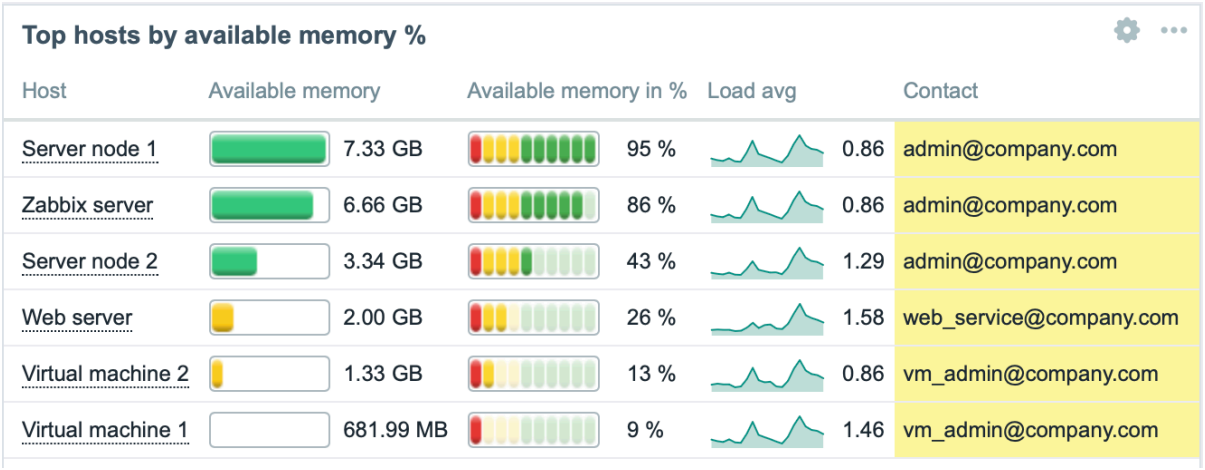
すべてのウィジェットに共通するパラメータに加えて、次の固有のオプションを設定できます。

表示	表示する内容を選択します。 システム統計 - Zabbix サーバーおよびシステムの主要なデータの概要を表示します。 高可用性ノード - 高可用性ノードのステータスを表示します（高可用性クラスターが有効な場合）。
ソフトウェア更新チェックの詳細を表示	Zabbix ソフトウェア更新チェックの詳細を表示するには、チェックボックスをオンにします。 このオプションは、Zabbix のサーバー設定でソフトウェア更新チェックが有効になっており、表示フィールドで「システム統計」が選択されている場合にのみ使用できます。

上位ホスト

概要

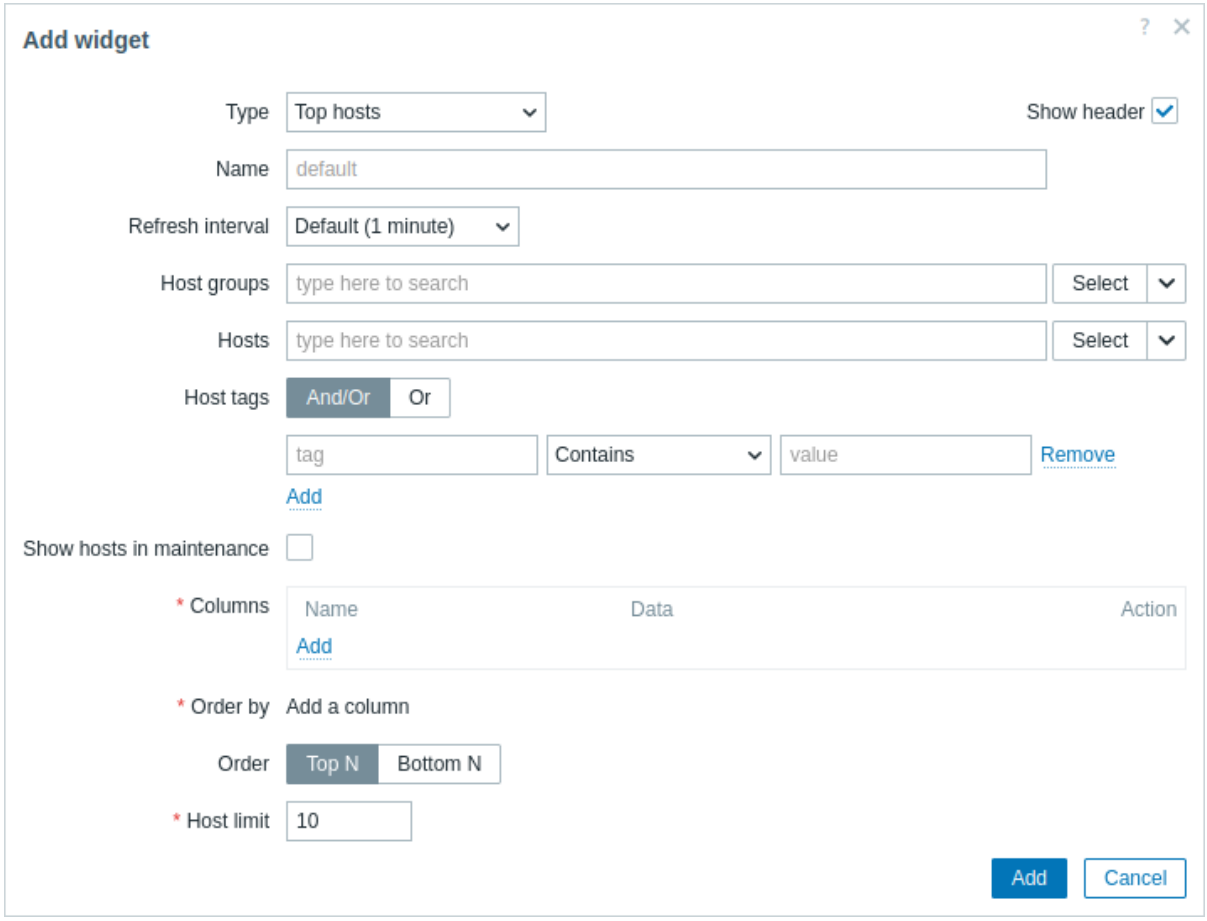
Top hosts ウィジェットは、複数のホストから選択したアイテムのトップ/ボトムの値を表示します。



このウィジェットは、最大 1000 台のホストを同時に表示することができます。

設定

設定するにはタイプとして上位ホストを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホストタグ	ウィジェットに表示するホスト数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字・小文字を区別します。 各条件にはいくつかの演算子が利用できます: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含む（部分一致、大文字・小文字を区別しない）タグ名を含める; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含む（部分一致、大文字・小文字を区別しない）タグ名を除外する。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
メンテナンス中のホストを表示	このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。 メンテナンス中のホストも表示する場合はこのチェックボックスをオンにします（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。デフォルトではオフです。

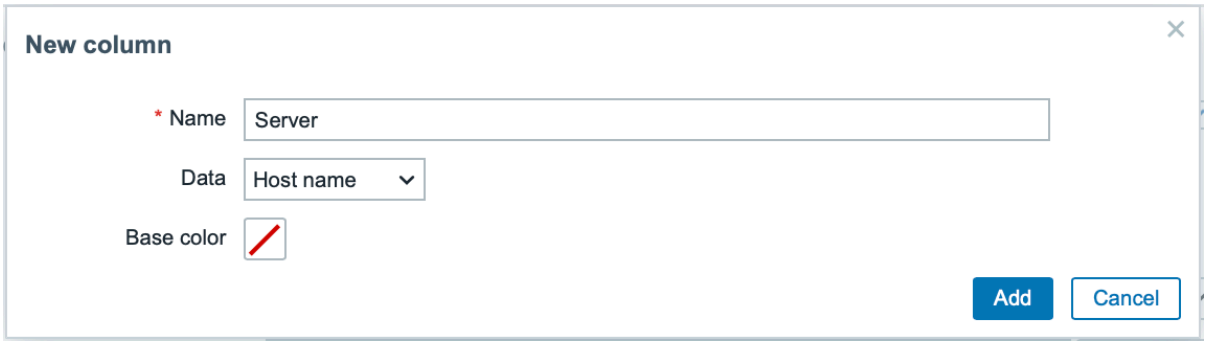
カラム	表示するデータのカラムを追加します。 カラムの順序は、左から右への表示順を決定します。 カラムはカラム名の前のハンドルをドラッグして上下に並べ替えることができます。
並び替え	*Top N* または *Bottom N* の並び替えに使用するカラムリストからカラムを指定します。
順序	行の並び順を指定します: **Top N** - 並び替えで集計した値の降順; **Bottom N** - 並び替えで集計した値の昇順。
ホストの上限	表示するホスト行数（1～1000）。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。

カラムの設定

3 つのデータ型のカラムを追加できます: ホスト名、アイテム値、またはテキストです。利用可能なカラムパラメータのリストは、カラムのデータ型、およびアイテム値タイプの場合は値のフォーマットによって異なります。

ホスト名カラム

ホスト名カラムは、ホストの名前を表示するために使用されます。



サポートされているパラメータ:

名前	カラムの名前。
データ	カラムに表示するデータタイプ。ホスト名を選択します。
基本色	カラムの背景色。

テキストカラム

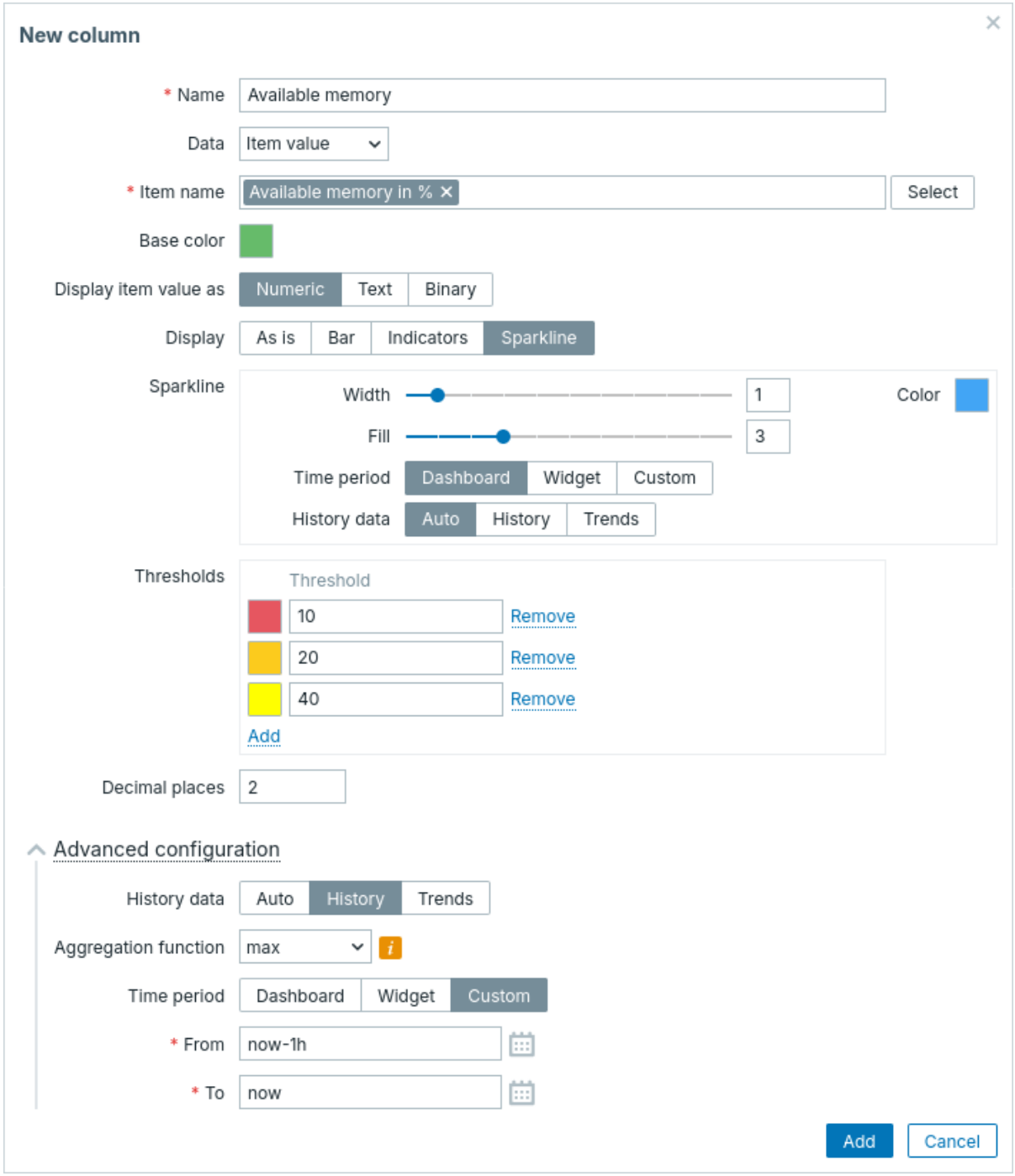
テキストカラムは、指定した任意のテキスト文字列を表示するために使用します。

サポートされているパラメータ:

名前	カラムの名前。
データ	カラムに表示するデータタイプ。テキストを選択します。
ベースカラー	カラムの背景色。
テキスト	表示する文字列を入力します。 ホストやインベントリのマクロを含めることができます。

アイテムの値列

アイテムの値列は、指定したアイテムの値を表示するために使用します。複数の値表示形式とオプションに対応しています。



サポートされているパラメータ:

Name	列の名前。
Data	列に表示するデータの種類。**Item value** を選択します。
Item name	アイテムを選択します。その名前を使用して、選択したすべてのホストで同じ名前のアイテムを照合し、表示します。ホストに該当するアイテムが複数ある場合、ウィジェットにはキーがアルファベット順で最初のアイテムが表示されます（例: `system.cpu.util` ではなく、キーが `proc.cpu.util` の *CPU utilization*）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみ選択できます。
Base color	列の背景色。*Item value* データをバーまたはインジケーターとして表示する場合は、塗りつぶしの色になります。アイテムの値が指定したしきい値のいずれかを超える場合、デフォルトの色をカスタム色で上書きできます。
Display item value as	アイテムの値の表示形式: **Numeric**、**Text**、または **Binary**。 選択したオプションに応じて、使用できる追加パラメータが決まります。各形式のパラメーター覧については、以下を参照してください。
Advanced configuration	

History data	ヒストリまたはトレンドからデータを取得します: **Auto** - 自動選択; **History** - ヒストリデータを取得; **Trends** - トレンドデータを取得。 このパラメータは、数値アイテムの値でのみ使用できます。テキスト値およびバイナリ値の場合、データは常にヒストリから取得されます。
Aggregation function	選択した *Time period* 内で使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の件数を表示; **sum** - 値の合計を表示; **first** - 最初の値を表示; **last** - 最後の値を表示。 **not used** を選択すると、最新の値が表示されます（集計なし）。 集計を使用すると、最新の値の代わりに、指定した期間（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* で表示できるのは数値データのみです。*count* では、数値以外のデータは数値に変換されます。
Time period	値の集計に使用する期間を指定します。期間のデータソースを選択します: **Dashboard** - ダッシュボードの期間セレクターを使用; **Widget** - 互換性のあるウィジェット（*Widget* パラメータで設定）を使用; **Custom** - *From* および *To* パラメータで設定したカスタム期間を使用。設定すると、ウィジェットの右上に時計アイコンが表示され、マウスオーバー時に設定された時刻を示します。 ウィジェットの *Time period* 設定に関係なく、互換性のあるウィジェットを期間のデータソースとして使用できます。 *Aggregation function* が「not used」に設定されている場合、このパラメータは使用できません。
Widget	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 *Time period* が「Widget」に設定されている場合に使用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 *Time period* が「Custom」に設定されている場合に使用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 *Time period* が「Custom」に設定されている場合に使用できます。

数値アイテムの値のパラメータ:

Display	値の表示方法を定義します: **As is** - 通常のテキストとして表示; **Bar** - 単色で塗りつぶされたバーとして表示; **Indicators** - 区切られた色付きのバーとして表示; **Sparkline** - ミニ折れ線グラフとして表示。
Thresholds	背景色または塗りつぶしの色を変更するしきい値を指定します。 保存時に、一覧は昇順に並べ替えられます。 スパークラインでは、しきい値はアイテムの最後の値にのみ適用されます。
Decimal places	値とともに表示する小数点以下の桁数を指定します。
Bar/Indicators	
Min	最小値。
Max	最大値。
Sparkline	
Width	スライダーを使用するか、0 から 10 の範囲の値を手動で入力して、グラフの線の太さを設定します。
Color	線と塗りつぶしの色を選択します。
Fill	スライダーを使用するか、0 から 10 の範囲の値を手動で入力して、塗りつぶしの透明度を設定します。

Time period	スパークラインチャートに含める値の期間を指定します。期間のデータソースを選択します: **Dashboard** - ダッシュボードの期間セレクターを使用; **Widget** - 互換性のあるウィジェット（*Widget* パラメータで設定）を使用; **Custom** - *From* および *To* パラメータで設定したカスタム期間を使用。設定すると、ウィジェットの右上に時計アイコンが表示され、マウスオーバー時に設定された時刻を示します。 ウィジェットの *Time period* 設定に関係なく、互換性のあるウィジェットを期間のデータソースとして使用できます。
Widget	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 *Time period* が「Widget」に設定されている場合に使用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 *Time period* が「Custom」に設定されている場合に使用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 *Time period* が「Custom」に設定されている場合に使用できます。
History data	ヒストリまたはトレンドからデータを取得します: **Auto** - 自動選択; **History** - ヒストリデータを取得; **Trends** - トレンドデータを取得。

テキストアイテムの値のパラメータ:

Highlights	一致した場合に背景色または塗りつぶしの色を変更する正規表現を指定します。

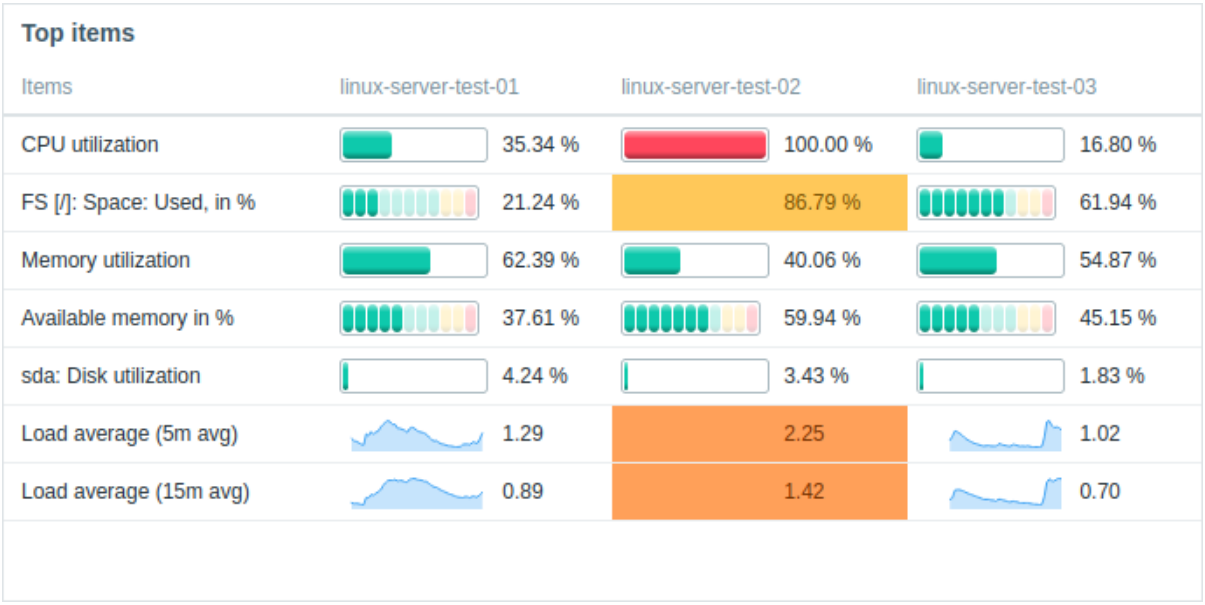
バイナリアイテムの値のパラメータ:

Show thumbnail	バイナリデータを含む画像のサムネイルを作成して表示するか、値列にフルサイズ画像へのハイパーリンク *Show* を表示するかを指定します。

上位アイテム

概要

Top items ウィジェットは、選択したアイテムのトップ/ボトム値を表示し、そのパフォーマンスの概要を素早く把握できます。

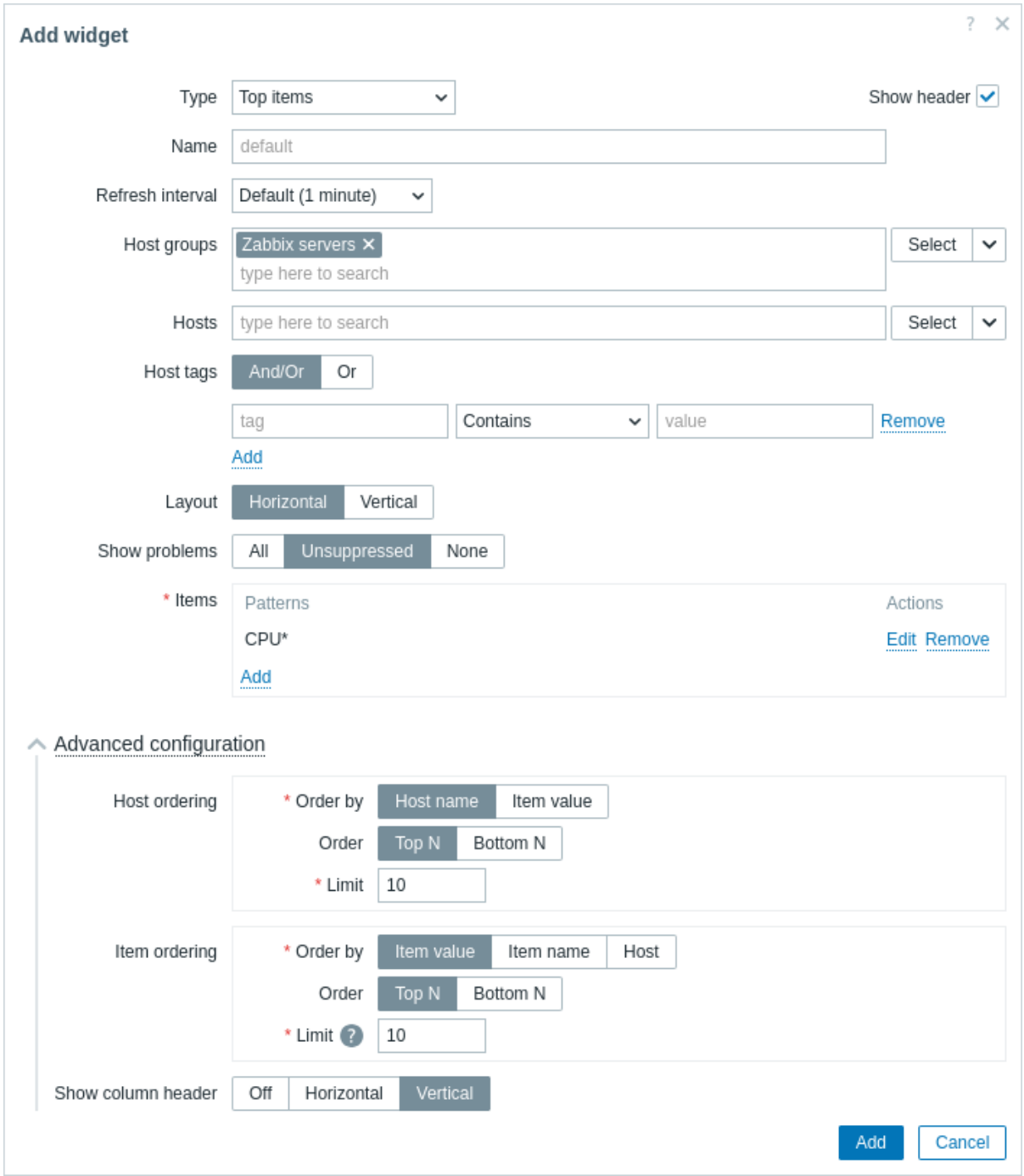


デフォルトでは、ウィジェットは直近 24 時間の値を表示します。この期間は、管理 > 一般 > *GUI*（最大履歴表示期間）で変更できます。

アイテム値をクリックすると、アイテムメニューが開きます。

設定

設定するには、タイプとして *Top items* を選択します:



すべてのウィジェットに共通のパラメータに加えて、次の固有オプションを設定できます:

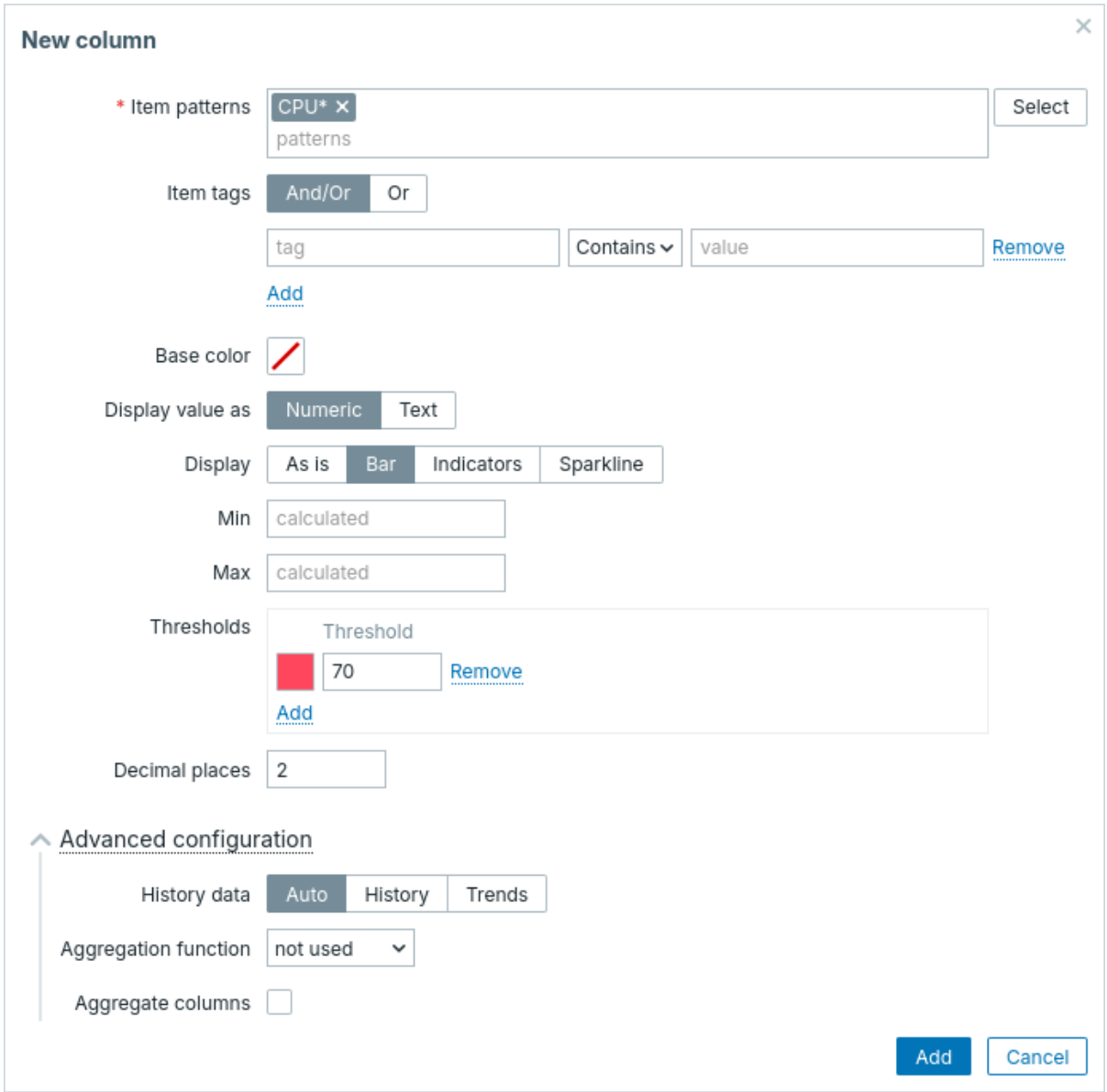
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして、互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を開始すると一致するグループのドロップダウン候補が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応のため、ホスト名の入力を開始すると一致するホストのドロップダウン候補が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

ホストタグ		ウィジェットに表示するホストデータの数を制限するタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件で使用できる演算子は次のとおりです: **Exists** - 指定したタグ名を含める; **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別）; **Contains** - タグ値に入力した文字列を含む指定タグ名を含める（部分一致、大文字と小文字を区別しない）; **Does not exist** - 指定したタグ名を除外する; **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別）; **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外する（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
レイアウト		レイアウトオプションを選択します: **Horizontal** - ホスト名を左側に表示します; **Vertical** - ホスト名を上部に表示します。
障害を表示		ステータスに基づいて表示する障害をフィルタリングします（all、unsuppressed、none）。 障害は色付きのアイテム値で示され、アイテム列の表示設定より優先されます。色は障害の深刻度に基づきます。深刻度は *Administration* > *General* > *Trigger displaying options* でカスタマイズできます。障害の深刻度は、障害を更新すると変更できます。
アイテム詳細設定		表示するアイテムパターンまたは特定のアイテムを追加します（列の設定を参照）。
ホストの並び順		ホスト列/行の並び順オプションを選択します。
	並び順の基準	ホストの並び順を次の基準で指定します: **Host name** - ホスト名で並べ替えます; **Item value** - 選択したアイテムの値で並べ替えます。
	順序	表示する値を高い順または低い順のどちらにするかを選択します: **TopN** - 上位 N 件の値; **BottomN** - 下位 N 件の値。 N の値は *Limit* フィールドで指定します。
	Limit	表示可能なホスト数の上限を入力します（範囲 1-1000、既定値 10）。 この値が 順序フィールドの N の値になります。
アイテムの並び順		アイテム列/行の並び順オプションを選択します。

	並び順の基準	アイテムの並び順を次の基準で指定します: **Item value** - アイテム値で並べ替えます; **Item name** - アイテム名で並べ替えます; **Host** - 選択したホストパターンで並べ替えます。
	順序	表示する値を高い順または低い順のどちらにするかを選択します: **TopN** - 上位 N 件の値; **BottomN** - 下位 N 件の値。 N の値は *Limit* フィールドで指定します。
	Limit	表示可能なアイテム数の上限を入力します（範囲 1-1000、既定値 10）。 この値が 順序フィールドの N の値になります。
列ヘッダーを表示		列ヘッダーの表示オプションを選択します: **Off** - 列ヘッダーを表示しません; **Horizontal** - ヘッダー内のテキストを横向きに表示します; **Vertical** - ヘッダー内のテキストを縦向きに表示します。

カラムの設定

アイテムカラムを設定するには、アイテムパラメータで追加をクリックします。



共通のカラムパラメータ:

アイテムパターン	ワイルドカード文字を使用して、1 つまたは複数のアイテムパターンを指定します。 または、アイテムを選択します。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみ選択できます。
アイテムタグ	ウィジェットに表示されるアイテムデータの数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める (大文字・小文字を区別); 含む - タグ値が入力した文字列を含むタグ名を含める (部分一致、大文字・小文字を区別しない); 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する (大文字・小文字を区別); 含まない - タグ値が入力した文字列を含むタグ名を除外する (部分一致、大文字・小文字を区別しない)。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
ベースカラー	表示が「バー」または「インジケーター」に設定されている場合、カラムの背景色または塗りつぶし色を選択します。 ベースカラーはしきい値やハイライトカラーで上書きされる場合があります。
値の表示形式	アイテム値の表示形式: 数値またはテキスト。 選択したオプションによって、利用可能な追加パラメータが決まります。各形式のパラメータリストを参照してください。

数値用の列パラメータ:

表示	値の表示方法を定義します: そのまま - 通常のテキストとして表示; バー - 単色で塗りつぶされたバーとして表示; インジケーター - 複数の区切りに分かれた、色付きのバーとして表示; スパークライン - 小さな折れ線グラフとして表示。
最小	バー/インジケーター表示の最小値。
最大	バー/インジケーター表示の最大値。
幅	スライダーを使用するか、0～10 の範囲で値を手動入力して、グラフの線の太さを設定します。 このパラメータはスパークライン表示でのみ使用できます。
塗りつぶし	スライダーを使用するか、0～10 の範囲で値を手動入力して、塗りつぶし色の透明度を設定します。 このパラメータはスパークライン表示でのみ使用できます。
色	線と塗りつぶしの色を選択します。 このパラメータはスパークライン表示でのみ使用できます。
時間期間および履歴データの選択に関連するフィールドについては、詳細設定を参照してください。	
しきい値	背景/塗りつぶしの色を変更するしきい値を指定します。 保存時に、リストは昇順に並べ替えられます。 スパークラインでは、しきい値はアイテムの最新の値にのみ適用されます。
小数点以下の桁数	値とともに表示する小数点以下の桁数を指定します。
詳細設定	
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動的に選択; 履歴 - 履歴データを取得; トレンド - トレンドデータを取得。 このパラメータは数値アイテムの値でのみ使用できます。テキスト値の場合、データは常に履歴から取得されます。

集計関数	選択した時間期間内で使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の件数を表示; **sum** - 値の合計を表示; **first** - 最初の値を表示; **last** - 最後の値を表示。 使用しないを選択した場合は、最新の値が表示されます（集計なし）。 集計を使用すると、最新の値の代わりに、指定した期間（5 分、1 時間、1 日など）の集計値を表示できます。 *min*、*max*、*avg*、*sum* で表示できるのは数値データのみです。*count* では、数値以外のデータは数値に変換されます。
時間期間	値の集計に使用する時間期間を指定します。時間期間のデータソースを選択します: ダッシュボード - ダッシュボードの時間期間セレクターを使用; ウィジェット - 互換性のあるウィジェット（ウィジェットパラメータで設定）を使用; カスタム - 開始および終了パラメータで設定したカスタム時間期間を使用; 設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が表示されます。 ウィジェットの時間期間設定に関係なく、互換性のあるウィジェットを時間期間のデータソースとして使用できることに注意してください。 集計関数が「使用しない」に設定されている場合、このパラメータは使用できません。
ウィジェット	時間期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 このパラメータは、時間期間が「ウィジェット」に設定されている場合に使用できます。
開始	時間期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、時間期間が「カスタム」に設定されている場合に使用できます。
終了	時間期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）がサポートされています。 このパラメータは、時間期間が「カスタム」に設定されている場合に使用できます。
列を集計	有効にすると、列のアイテムパターンに一致するすべてのアイテムを、1 つの結合された列/行にグループ化します。一致するアイテムの値は、選択した列の集計関数を使用して集計され、結合された列名の下に表示されます。デフォルト: 無効（未選択）。このパラメータは、値の表示方法が数値の場合にのみ表示されます。
列の集計関数	グループ化されたアイテムパターンの値を結合する方法を選択します。使用可能なオプション: **min**、**max**、**avg**、**count**、**sum**（デフォルト）。このパラメータは、列を集計が有効な場合にのみ表示されます。
結合された列名	集計された列の表示名を入力します。列を集計が有効な場合、このフィールドは必須となり、結合された値の列ヘッダーになります。

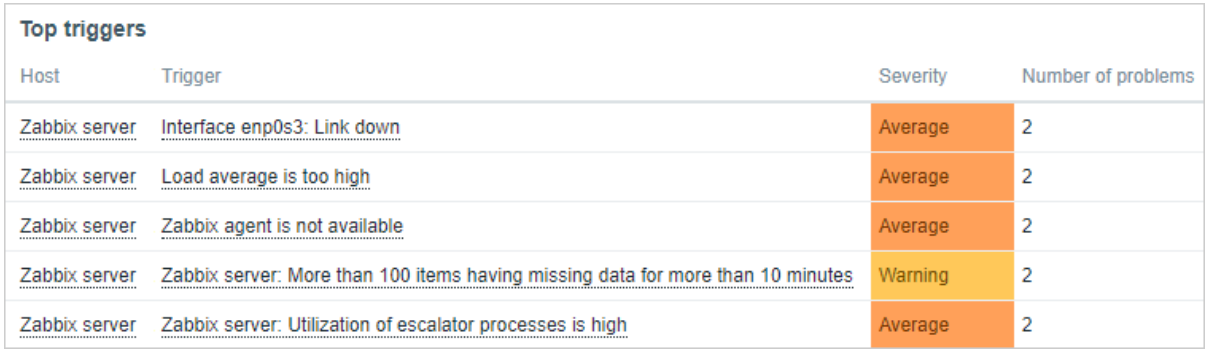
テキスト値用の列パラメータ:

ハイライト	一致したときに背景/塗りつぶしの色を変更する正規表現を指定します。

上位トリガー

概要

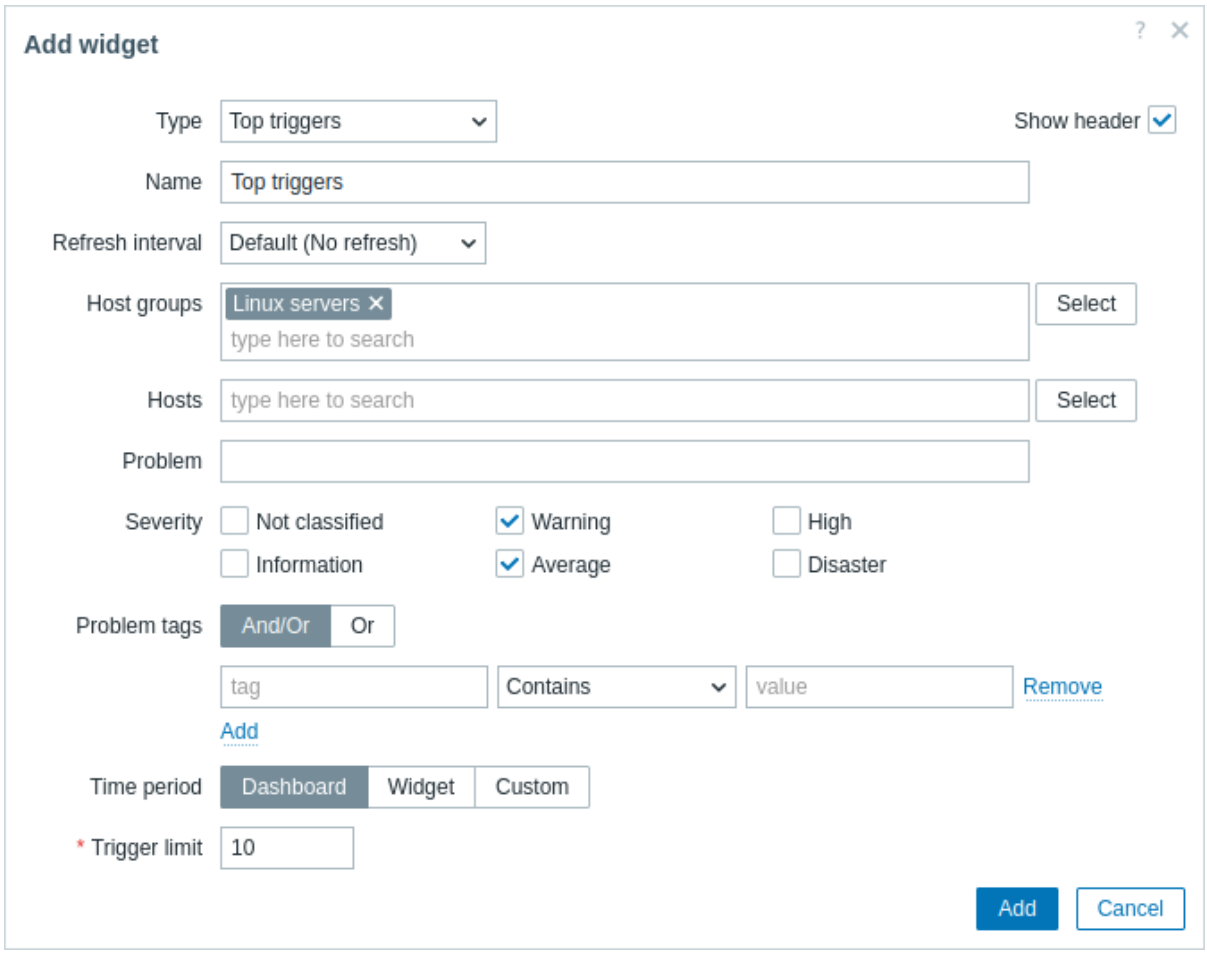
Top triggers ウィジェットは、最も多くの問題が発生しているトリガーを表示します。



* レポート > Top 100 triggers*と同じデータを表示し、最大 1000 件まで表示できます。

設定

設定するには、ウィジェットタイプとして トップトリガーを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ウィジェットに表示する問題のホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループの問題がウィジェットに表示されます。ホストグループを入力しない場合、すべてのホストグループの問題が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示する問題のホストを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合、すべてのホストの問題が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

問題	特定の問題のみのトリガーを表示できます。そのためには、問題名に一致する文字列を入力します。 マクロは展開されません。
深刻度	ウィジェットに表示するトリガーをフィルタリングするために、トリガーの深刻度をマークします。 深刻度をマークしない場合、すべてのトリガーが表示されます。
問題タグ	ウィジェットに表示する問題のタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字・小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字・小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字・小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字・小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字・小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
期間	期間のデータソースを選択します: ダッシュボード - ダッシュボードの期間セレクタを使用; ウィジェット - 互換性のあるウィジェットを使用（ウィジェットパラメータで設定）; カスタム - *From* および *To* パラメータで設定したカスタム期間を使用。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバーで設定した時間が表示されます。 ウィジェットの期間設定に関係なく、互換性のあるウィジェットは引き続き期間のデータソースとして使用できます。
ウィジェット	期間のデータソースとして互換性のあるウィジェットを入力または選択します。 このパラメータは期間が「ウィジェット」に設定されている場合に利用できます。
From	期間の開始を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
To	期間の終了を入力または選択します。 相対時間構文（`now`、`now/d`、`now/w-1w` など）に対応しています。 このパラメータは期間が「カスタム」に設定されている場合に利用できます。
トリガー数の上限	表示するトリガー数を設定します。設定可能な値の範囲: 1-1000。

トリガー概要

概要

Trigger overview ウィジェットは、ホストのトリガーの現在の状態を色付きブロックの表として表示します。これにより、さまざまなホスト上のトリガーの健全性とアクティビティをすばやく視覚的に把握できます。



このウィジェットでは、トリガーの詳細を示すために複数の視覚要素を使用します。

- PROBLEM のトリガーのブロック色は、障害の深刻度に対応します。
- 点滅するブロックは、最近のトリガー状態の変化（直近 2 分以内）を示します。
- 灰色の矢印は、依存関係のあるトリガーを示します（詳細を表示するには、マウスオーバーしてください）。
- チェックボックスのアイコンは、確認済みの障害を示します。このアイコンが表示されるには、そのトリガーに関するすべてのアクティブまたは解決済みの障害が確認済みである必要があります。

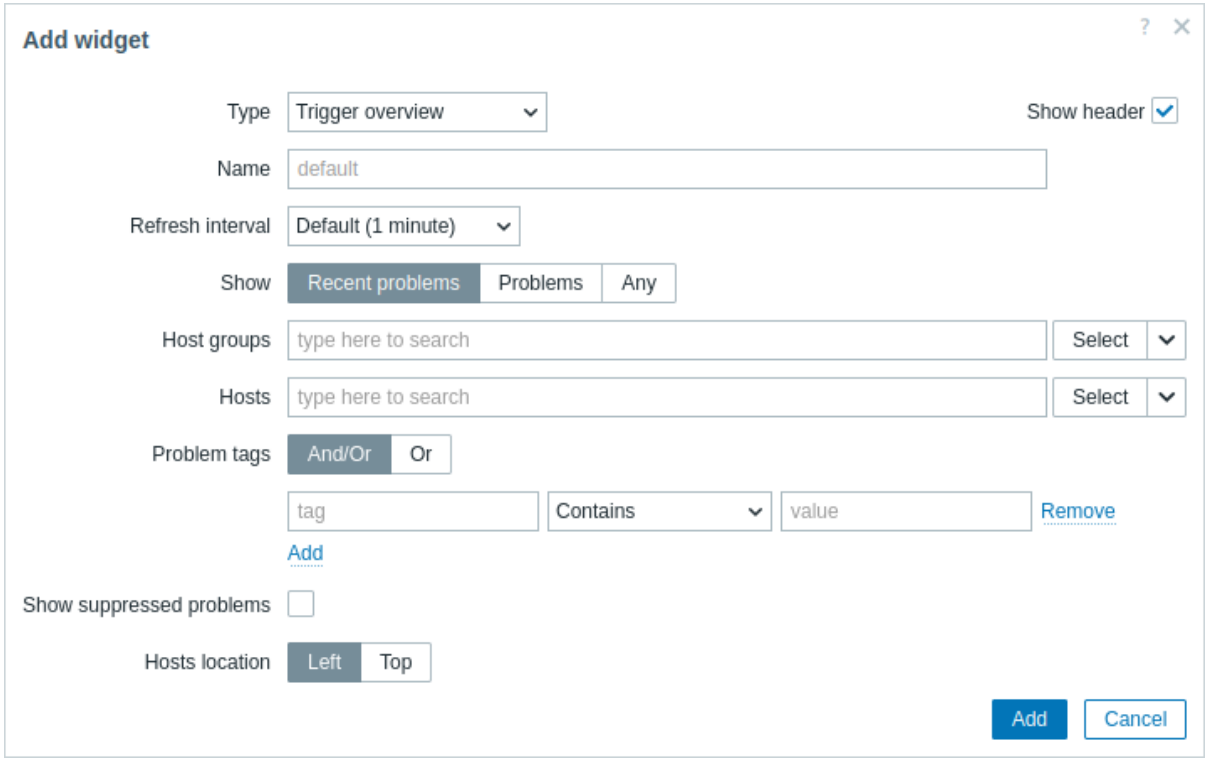
Note:
トリガーの深刻度の色は、*Administration* > *General* で調整できます。トリガーの深刻度は、障害の更新 時に変更できます。

ブロックをクリックすると、トリガー関連の情報（障害、トリガー設定など）へのリンクが表示されます。

デフォルトでは、最大 50 件のレコードが表示されます（*Administration* > *General* > *GUI* の *Max number of columns and rows in overview tables* で設定可能です）。それ以上のトリガーが存在する場合は、フィルタ条件を絞り込むようメッセージが表示されます。この制限は、追加のフィルタリング（たとえばタグによる絞り込み）を行う前に適用されます。

設定

設定するには、タイプとして トリガーの概要を選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます:

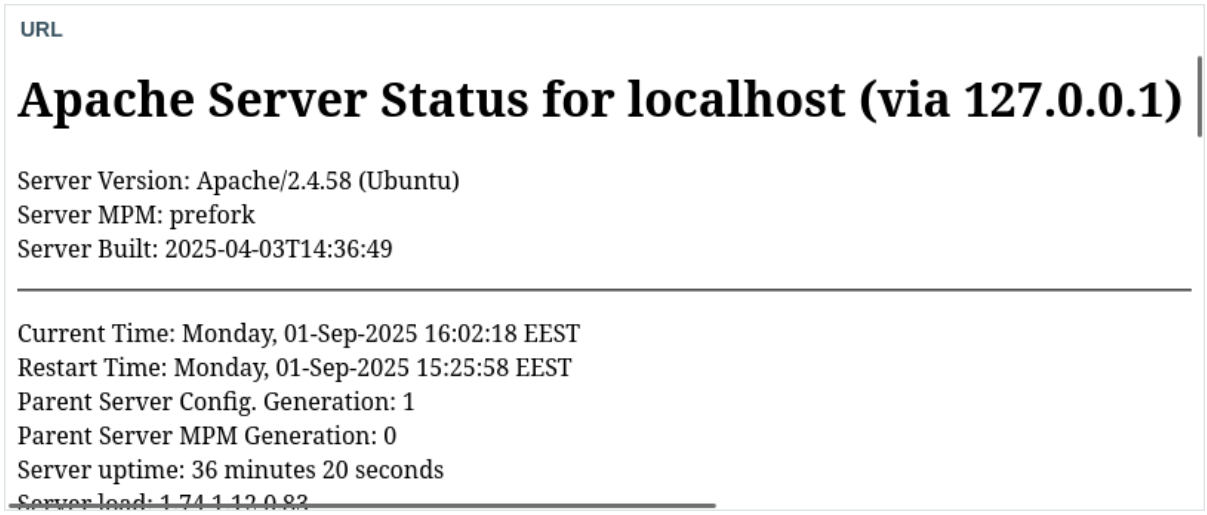
表示	トリガーの状態でフィルタリング: 最近の障害 - *(デフォルト)* 最近障害状態になった、または現在も障害状態のトリガー (解決済みと未解決) を表示; 障害 - 障害状態 (未解決) のトリガーを表示; すべて - すべてのトリガーを表示。
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応なので、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応なので、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合は利用できません。
障害タグ	ウィジェットに表示するトリガーをフィルタリングするタグを指定します。 特定のタグやタグ値を含めたり除外したりできます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 注意**:** パラメータ 表示が「すべて」に設定されている場合、タグを指定してもすべてのトリガーが表示されます。ただし、最近のトリガー状態の変更 (点滅ブロックとして表示) はすべてのトリガーで更新されますが、トリガー状態の詳細 (障害の深刻度の色や障害が認識済みかどうか) は、指定したタグに一致するトリガーのみ更新されます。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める (大文字小文字を区別); 含む - タグ値が入力した文字列を含むタグ名を含める (部分一致、大文字小文字を区別しない); 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する (大文字小文字を区別); 含まない - タグ値が入力した文字列を含むタグ名を除外する (部分一致、大文字小文字を区別しない)。 条件の計算タイプは 2 つあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

抑制された障害を表示	ホストのメンテナンスによって抑制 (非表示) される障害も表示するにはチェックボックスをオンにします。
レイアウト	レイアウトオプションを選択: 横 - ホスト名が左側に表示されます; 縦 - ホスト名が上部に表示されます。

URL

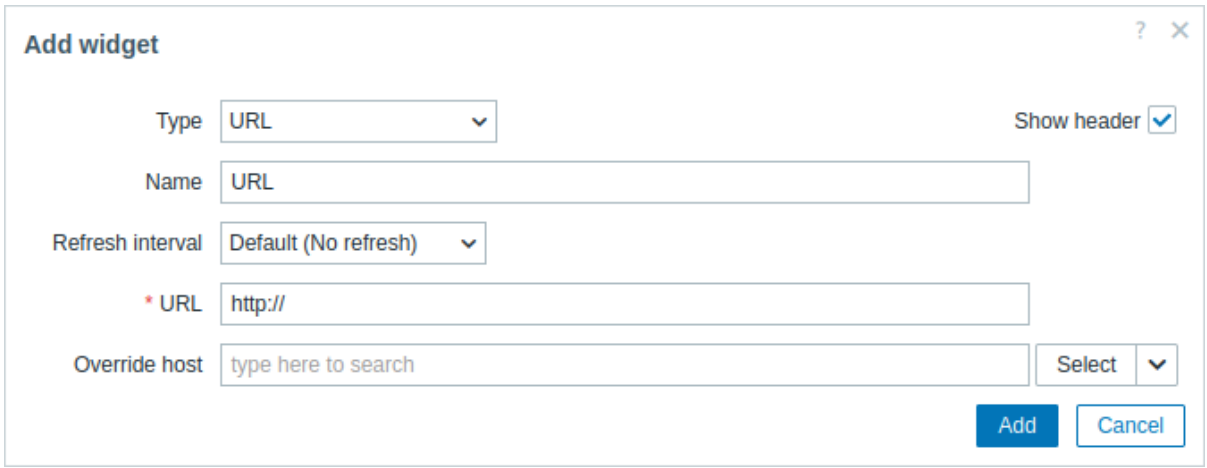
概要

URL ウィジェットは、URL から取得したコンテンツを表示します。



設定

設定するには、タイプとして *URL* を選択します:



すべてのウィジェットに共通のパラメータに加えて、次の固有オプションを設定できます:

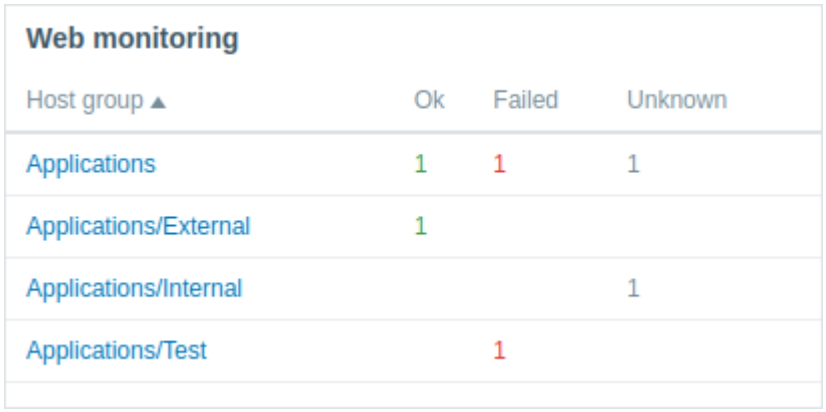
URL	表示する URL を入力します（最大 2048 文字）。 外部 URL は `http://` または `https://` で始まる必要があります。 内部 URL は相対パスに対応しています（例: `zabbix.php?action=report.status`）。 `{HOST.*}` マクロがサポートされています。
Override host	ホストのdata sourceとして、互換性のあるウィジェットまたはダッシュボードのhost selectorを選択します。 このパラメータは、template dashboardでウィジェットを設定する場合は使用できません。

Attention:
Zabbix Web インターフェースに HTTPS 経由でアクセスしている場合、ブラウザーがウィジェットで設定された HTTP ページを読み込まないことがあります。また、Zabbix Web インターフェース外のコンテンツを表示するには、X-Frame-Options HTTP response header を変更する必要がある場合があります。

Web 監視

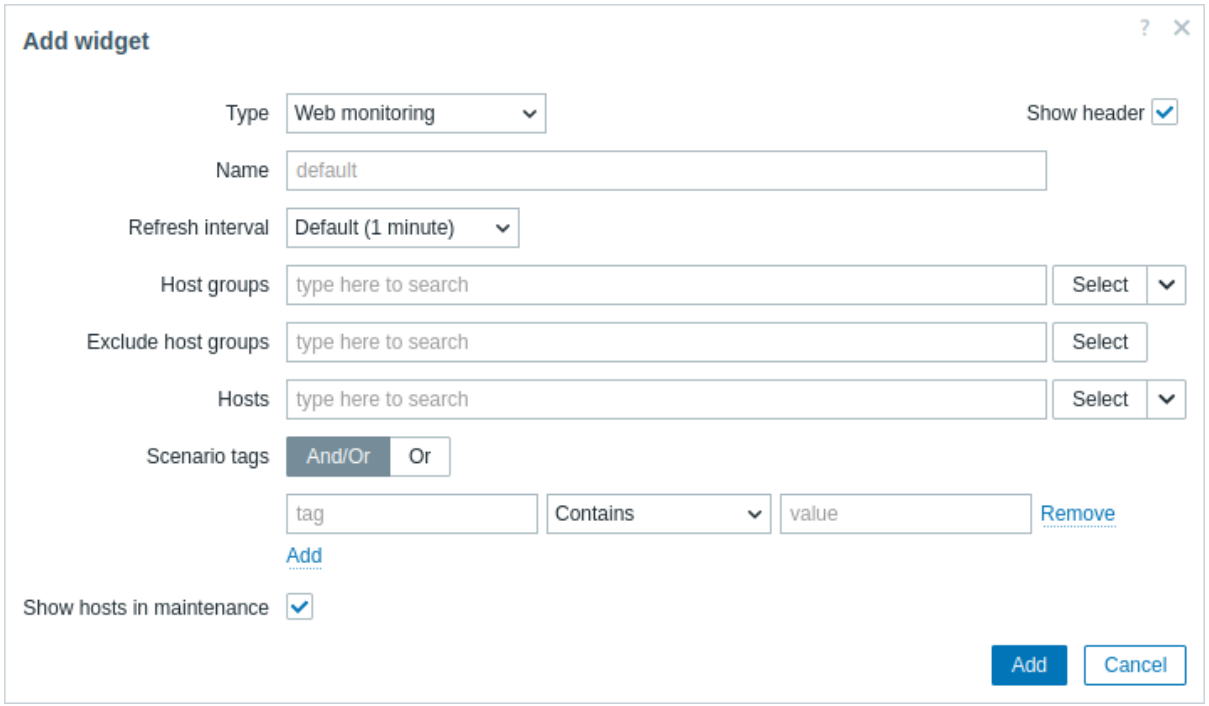
概要

Web 監視ウィジェットは、アクティブな Web シナリオのステータスを表示します。



ホストグループ名をクリックすると、監視データ > ホスト > *Web* シナリオが開き、シナリオの詳細を確認できます。

設定



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます。

ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。

ホストグループの除外	ウィジェットから非表示にするホストグループを選択します。 このフィールドはオートコンプリート対応で、グループ名の入力を始めると一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。これらのホストグループのホストデータはウィジェットに表示されません。たとえば、ホスト 001、002、003 がグループ A に、ホスト 002、003 がグループ B にも所属している場合、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリート対応で、ホスト名の入力を始めると一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は利用できません。
シナリオタグ	ウィジェットに表示する Web シナリオの数を制限するタグを指定します。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件にはいくつかの演算子があります: 存在する - 指定したタグ名を含める; 等しい - 指定したタグ名と値を含める（大文字小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含める（部分一致、大文字小文字を区別しない）; 存在しない - 指定したタグ名を除外する; 等しくない - 指定したタグ名と値を除外する（大文字小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外する（部分一致、大文字小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は *Or* 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。
メンテナンス中のホストを表示	メンテナンス中のホストを統計に含めます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はメンテナンス中のデータを表示と表示されます。

監視

概要

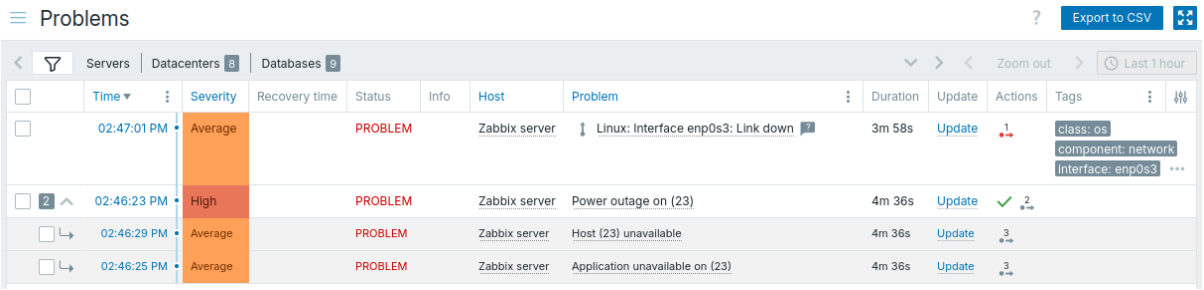
[監視] メニューでは、データの表示を行います。Zabbix で収集、可視化、処理するように設定された情報は、[監視] メニューの各セクションに表示されます。

障害

概要

Monitoring > *Problems* では、現在発生している障害を確認できます。障害とは、”Problem” 状態になっているトリガーのことです。

デフォルトでは、新しい障害はすべて原因障害として分類されます。特定の障害を、原因障害の症状障害として手動で再分類することもできます。詳細については、原因イベントと症状イベント を参照してください。



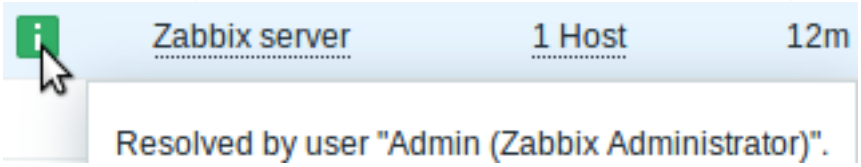
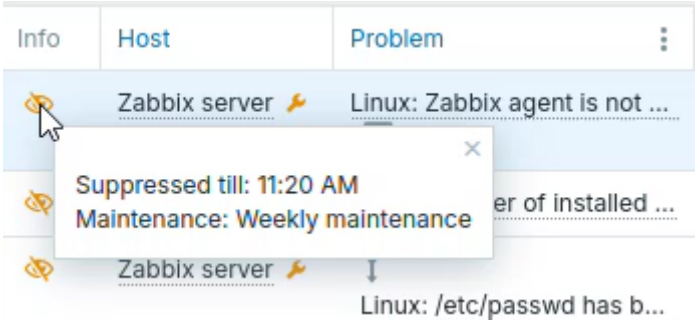
障害の詳細は、カスタマイズ可能なテーブルに表示されます。テーブルヘッダーの *Customize table* ボタンを使用すると、列の表示/非表示や並べ替え、レイアウトのデフォルトへのリセットができます。次のオプションも利用できます。

- *Compact view* - 行の高さを小さくして、より多くの障害を画面に表示します。
- *Highlight full row* - 障害の深刻度の色を使って、未解決の障害を強調表示します（高コントラストテーマでは使用できません）。

各列は右端をドラッグして幅を変更できます。境界線をダブルクリックすると、幅をリセットできます。すべてのカスタマイズは即時に適用され、アクティブなフィルター タブ（デフォルトまたは保存済み）内で、現在のユーザーに対して常に記憶されます。

利用可能な列は次のとおりです。

Column	Description
Checkbox	障害選択用のチェックボックスが表示されます。 チェックボックスの横にあるアイコンの意味は次のとおりです。 2 - 原因障害に対する症状イベントの数; - 展開して症状イベントを表示; - 折りたたんで症状イベントを非表示; - これは症状イベントです。
Time	障害の開始時刻が表示されます。 列ヘッダーの *Options* ボタンをクリックすると、*Show timeline* オプションを使用できます。これにより、視覚的なタイムラインと障害のグループ化を切り替えられます（*Compact view* が有効な場合は使用できません）。
Severity	障害の深刻度が表示されます。 障害の深刻度は元々、基になる障害トリガーの深刻度に基づいていますが、イベント発生後に *Update problem* 画面 を使用して更新できます。障害発生中は、障害の深刻度の色がセルの背景色として使用されます。
Recovery time	障害の復旧時刻が表示されます。 この列は、フィルター の *Show* が *Problems* に設定されている場合は使用できません。
Status	障害の状態が表示されます。 **Problem** - 未解決の障害 **Resolved** - 最近解決された障害。フィルターを使用して最近解決された障害を非表示にできます。 新しい障害と最近解決された障害は 2 分間点滅します。解決済みの障害は合計 5 分間表示されます。 これら 2 つの値は、*Administration* > *General* > *Trigger displaying options* で設定できます。 この列は、フィルター の *Show* が *Problems* に設定されている場合は使用できません。

Column	Description
Info	障害がグローバル相関または障害更新時の手動操作によってクローズされた場合、緑色の情報アイコンが表示されます。アイコンにカーソルを合わせると、詳細が表示されます。  抑制された障害が表示される場合は、次のアイコンが表示されます: 。イベントがホストメンテナンスによって抑制されている場合、アイコンはオレンジ色で表示されます: 。メンテナンスと手動抑制の両方が適用される場合は、アイコンの色では手動抑制が優先されます。アイコンにカーソルを合わせると、詳細が表示されます。 
Host	障害のホストが表示されます。 ホスト名をクリックすると、ホストメニュー が表示されます。
Problem	障害名が表示されます。 障害名は、基になる障害トリガーの名前に基づいています。 トリガー名内のマクロは障害発生時に展開され、展開後の値はその後更新されません。 障害名をクリックすると、イベントメニュー が表示されます。 障害名の後にある アイコンにカーソルを合わせると、トリガーの説明が表示されます（説明が設定されている障害の場合）。 列ヘッダーの *Options* ボタンをクリックすると、次の 2 つのオプションを使用できます。 - *Show operational data* は、障害名の後ろに括弧付きで運用データを表示します（トリガー設定 の *Operational data* フィールドが空でない場合のみ）。 - *Show trigger expression* は、障害名の下にその障害のトリガー式を表示します（*Compact view* が有効な場合は使用できません）。
Operational data	障害の動的なリアルタイムデータが表示されます。内容はトリガー設定 の *Operational data* フィールドに依存します。マクロを含む文字列が定義されている場合はそれが表示され、フィールドが空の場合は、代わりにトリガー式内のすべてのアイテムの最新値が表示されます。
Duration	障害の継続時間が表示されます。 こちらも参照: 負の障害継続時間
Update	*Update* リンクをクリックすると、障害の更新 画面に移動し、コメントの追加や障害の承認など、障害に対してさまざまな操作を行えます。

Column	Description
Actions	障害に関するアクティビティ履歴が、記号アイコンで表示されます。  - 障害は承認されています。このアイコンは常に最初に表示されます。  - コメントが追加されています。コメント数も表示されます。  - 障害の深刻度が上がっています（例: Information > Warning）。  - 障害の深刻度が下がっています（例: Warning > Information）。  - 障害の深刻度が変更されましたが、元のレベルに戻りました（例: Warning > Information > Warning）。  - 障害は抑制されています。  - 障害の抑制が解除されています。  - ホストメンテナンスにより障害が抑制されています。  - ホストメンテナンス後に障害の抑制が解除されています。  - アクションが実行されています。アクション数も表示されます。  - アクションが実行されており、少なくとも 1 つは進行中です。アクション数も表示されます。  - アクションが実行されており、少なくとも 1 つは失敗しています。アクション数も表示されます。 アイコンにマウスを重ねると、アクティビティの詳細を示すポップアップが表示されます。実行されたアクションに対してポップアップ内で使用されるアイコンの詳細については、詳細の表示 を参照してください。
Tags	障害のタグ が表示されます。 さらに、外部チケットシステムのタグも表示される場合があります（webhook を設定する際の *Process tags* オプションを参照）。 表示されたタグの後ろにある ••• 3 点アイコンにカーソルを合わせるかクリックすると、すべてのタグを含むポップアップが開きます。 列ヘッダーの ⋮ *Options* ボタンをクリックすると、表示するタグ数（1〜3）、タグ名の表示方法（完全表示、短縮表示、または非表示）、タグ表示の優先順位を設定したり、列を複製したりできます（複製した列は名前変更または削除できます）。
Tag value	障害のタグ の値が表示されます。 列ヘッダーの ⋮ *Options* ボタンをクリックすると、列名の変更、表示するタグ値の選択、または列の複製ができます（複製した列は名前変更または削除できます）。
Custom text	カスタムテキスト、マクロ値、またはその両方の組み合わせが表示されます。 列ヘッダーの ⋮ *Options* ボタンをクリックすると、列名の変更、カスタムテキストまたはマクロの入力、または列の複製ができます（複製した列は名前変更または削除できます）。 サポートされるマクロ: `{HOST.*}`（`{HOST.METADATA}` を除く）、`{INVENTORY.*}`、`{ITEM.*}`（`{ITEM.STATE.ERROR}` を除く）、`{TRIGGER.ID}`、`{EVENT.ID}`、およびユーザーマクロ。

負の障害継続時間

実際には、障害の解決時刻が障害の発生時刻よりも早い場合、つまり負の障害継続時間になることが、いくつかの一般的な状況で発生する可能性があります。例えば:

- あるホストがプロキシによって監視されていて、ネットワークエラーが発生し、しばらくの間プロキシからデータが受信されなくなった場合、nodata(/host/key) トリガーがサーバーによって発火されます。接続が回復すると、サーバーは過去の時刻を持つアイテムデータをプロキシから受信します。その後、nodata(/host/key) の障害が解決され、負の障害継続時間となります。
- 障害イベントを解決するアイテムデータが Zabbix sender によって送信され、そのタイムスタンプが障害発生時刻よりも早い場合も、負の障害継続時間が表示されます。

Note:
負の障害継続時間は、特定のトリガーのSLA 計算や可用性レポートには一切影響しません。障害時間を短縮も延長もしません。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 一括更新 - 障害の更新画面に移動して、選択した障害を更新します

このオプションを使用するには、対象の障害の前にあるチェックボックスをオンにしてから、一括更新ボタンをクリックします。

ボタン

右側のボタンでは、以下のオプションを利用できます。

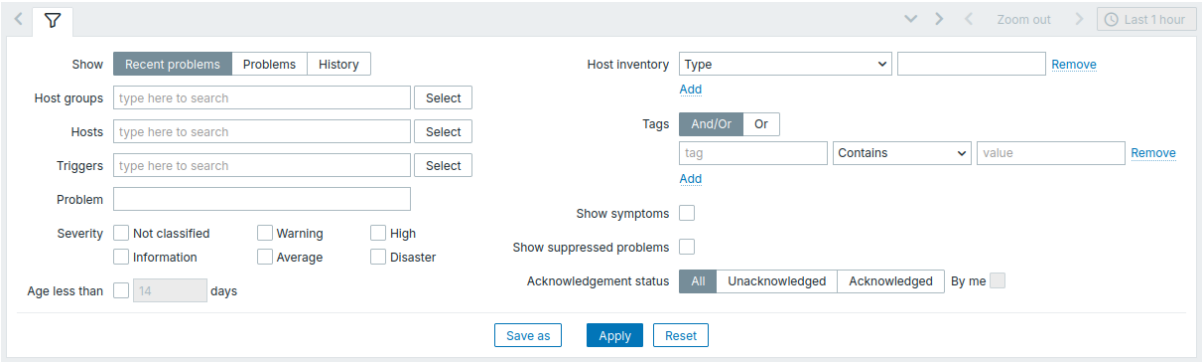
Export to CSV	すべてのページの内容を CSV ファイルにエクスポートします。

表示モードのボタンはすべてのセクションで共通のため、監視ページで説明しています。

フィルターの使用

フィルターを使用して、関心のある障害のみを表示できます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

フィルターは表の上にあります。お気に入りのフィルター設定はタブとして保存でき、フィルター上部のタブ をクリックしてすばやくアクセスできます。



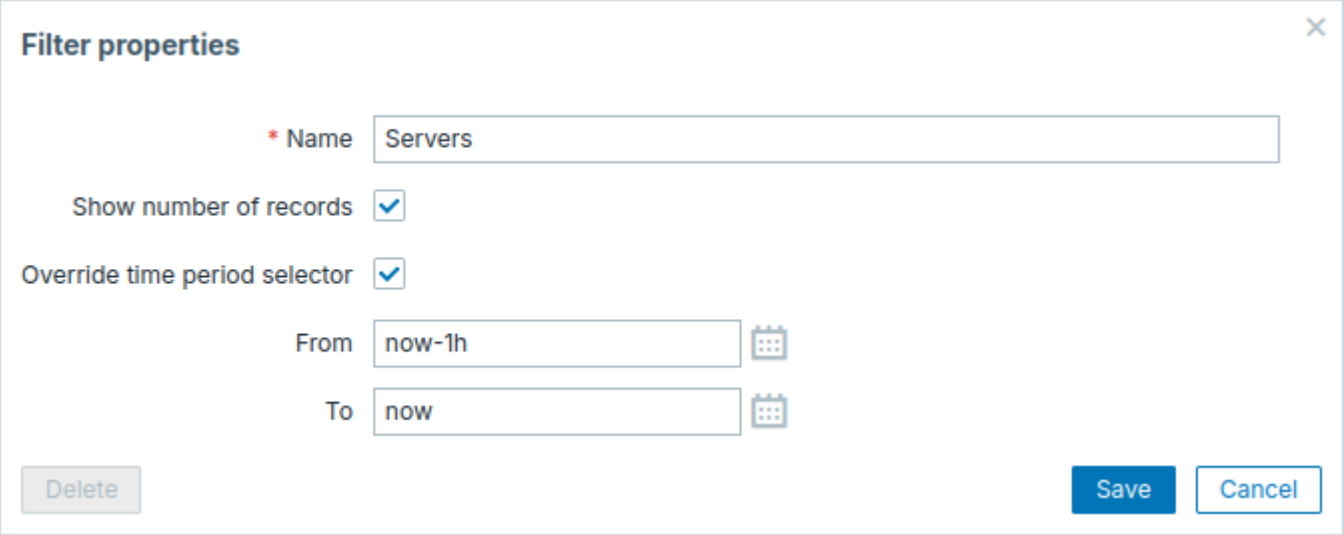
Parameter	Description
Show	障害ステータスでフィルターします: **Recent problems** - 未解決および最近解決された障害を表示します（デフォルト） **Problems** - 未解決の障害を表示します **History** - すべてのイベントの履歴を表示します
Host groups	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
Hosts	1 つ以上のホストでフィルターします。
Triggers	1 つ以上のトリガーでフィルターします。
Problem	障害名でフィルターします。
Severity	トリガー（障害）の深刻度でフィルターします。
Age less than	障害の経過時間でフィルターします。
Host inventory	インベントリの種類と値でフィルターします。
Tags	イベントタグ の名前と値でフィルターします。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件には、次の演算子を使用できます: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別） **Contains** - 入力した文字列をタグ値に含む指定したタグ名を含める（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別） **Does not contain** - 入力した文字列をタグ値に含む指定したタグ名を除外する（部分一致、大文字と小文字を区別しない） 条件の計算方法には 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です フィルター適用時、ここで指定したタグは、*Tag display priority*（下記）リストで上書きされない限り、障害とともに最初に表示されます。
Show symptoms	チェックボックスをオンにすると、症状として分類された障害を別行で表示します。

Parameter	Description
Show suppressed problems	チェックボックスをオンにすると、ホストメンテナンスのため、またはユーザーによって手動で抑制 されたために、通常は抑制されて非表示になる障害を表示します。
Acknowledgement status	すべての障害、未承認の障害のみ、または承認済みの障害のみを表示するようにフィルターします。 追加のチェックボックスをオンにすると、これまでに自分が承認した障害を除外できます。

お気に入りフィルターのタブ

よく使用するフィルターパラメーターのセットは、タブとして保存できます。

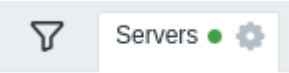
新しいフィルターパラメーターのセットを保存するには、メインタブを開いてフィルター設定を構成し、*Save as* ボタンをクリックします。新しいポップアップウィンドウで、*Filter properties* を定義します。



Parameter	Description
Name	タブ一覧に表示するフィルターの名前。
Show number of records	タブ名の横に障害の件数を表示したい場合にチェックします。
Override time period selector	このフィルターセットに特定のデフォルト期間を設定する場合にチェックします。設定すると、このタブの期間はフィルター設定を更新することでのみ変更できます。カスタム期間がないタブでは、右上の時間セレクターボタンをクリックして期間を変更できます（ボタン名は選択した時間間隔によって異なります: This week、Last 30 minutes、Yesterday など）。 このオプションは *Monitoring* > *Problems* のフィルターでのみ使用できます。
From/To	時間期間 の開始と終了を、絶対時刻 (`Y-m-d H:i:s`) または相対時刻の構文 (`now-1d`) で指定します。*Set custom time period* がチェックされている場合に使用できます。

保存すると、フィルターは名前付きフィルタータブとして作成され、すぐに有効になります。

既存のフィルターのフィルター属性を編集するには、アクティブなタブ名の横にある歯車アイコンをクリックします。



注意:

- フィルター領域を非表示にするには、現在のタブ名をクリックします。アクティブなタブ名をもう一度クリックすると、フィルター領域が再び開きます。
- キーボード操作に対応しています。矢印キーでタブを切り替え、*Enter* を押して開きます。
- フィルターの上にある左/右ボタンで、保存済みフィルターを切り替えられます。また、下向きボタンを押すと、保存済みフィルターの一覧を含むドロップダウンメニューが開き、必要なものをクリックできます。
- 保存済みフィルターの設定が変更されている場合（ただし保存はされていない場合）、フィルター名の後ろに緑の点が表示されます。新しい設定に従ってフィルターを更新するには、*Save as* ボタンの代わりに表示される *Update* ボタンをクリックします。
- テーブルの列レイアウトはフィルタータブごとに保存されます。保存済みフィルターのレイアウトをカスタマイズするには、まずそのフィルタータブを開いてから列をカスタマイズします。
- フィルタータブ、フィルター設定、および各タブのテーブル列レイアウトはユーザープロファイルに記憶され、ユーザーがページを再度開いたときに復元されます。

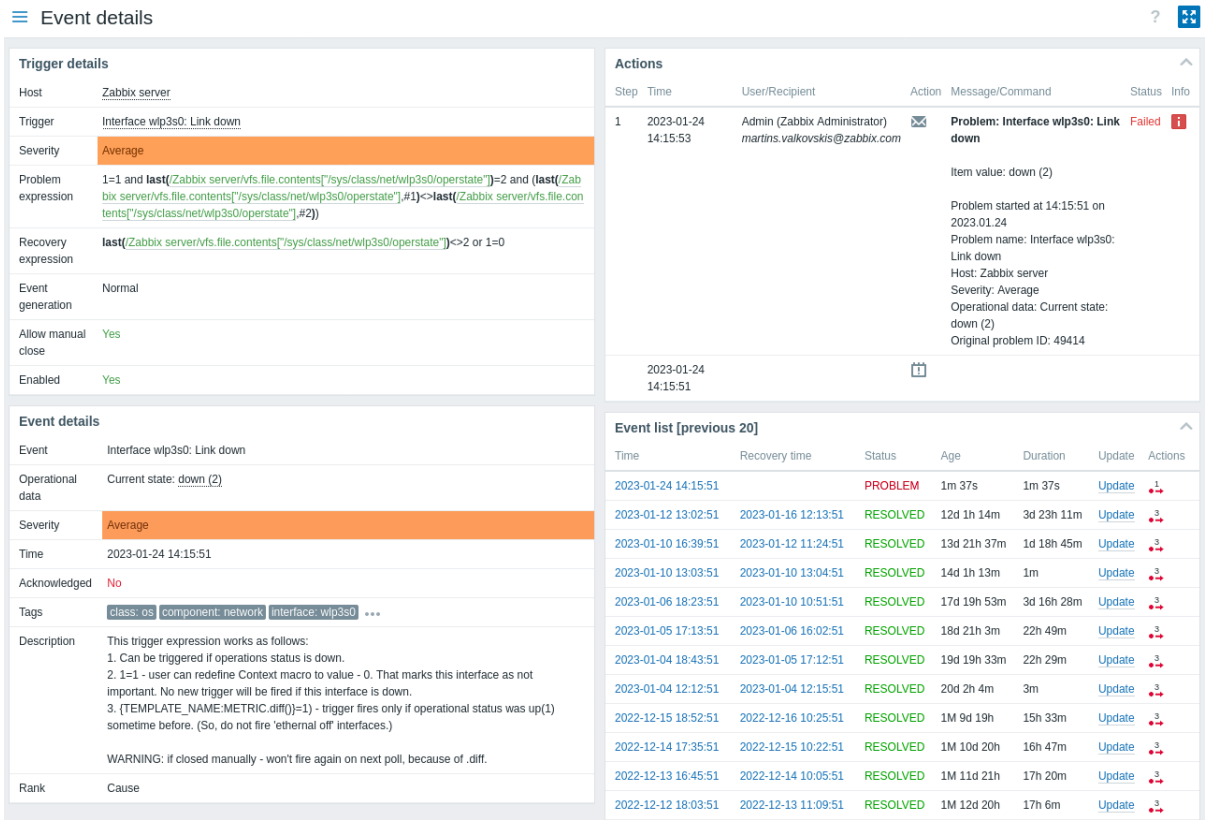
Note:
フィルターを共有するには、アクティブなフィルターの URL をコピーして他のユーザーに送信します。この URL を開くと、他のユーザーはこのパラメーターセットを Zabbix アカウントの永続的なフィルターとして保存できます。
 関連項目: Page parameters.

フィルターのボタン

Apply	指定されたフィルタリングを（保存せずに）適用します
Reset	現在のフィルターをリセットし、現在のタブに保存されたパラメーターに戻ります。メインタブの場合はフィルターがクリアされます。
Save as	現在のフィルタリングパラメーターを新しいタブに保存します。メインタブでのみ使用できます。
Update	タブに保存されているパラメータを現在指定されているパラメータに置き換えます。メインタブでは使用できません。

詳細の表示

Monitoring > *Problems* における障害発生時刻と復旧時刻はリンクです。
それらをクリックすると、イベントの詳細が表示されます。



イベントの詳細では、メンテナンスのためにイベントが抑制されたのか、ユーザーによって手動で抑制されたのかを確認できます。

障害の深刻度は、トリガーと障害イベントで異なる場合があることに注意してください。これは、*Update problem* 画面 を使用して障害イベントの深刻度が更新されている場合です。

アクション一覧では、アクティビティの種類を示すために次のアイコンが使用されます。

- - 障害イベントが生成された。
- - メッセージが送信された。

-  - 障害イベントが確認された。
-  - 障害イベントが未確認である。
-  - コメントが追加された。
-  - 障害の深刻度が上がった（例: Information > Warning）。
-  - 障害の深刻度が下がった（例: Warning > Information）。
-  - 障害の深刻度が変更されたが、元のレベルに戻った（例: Warning > Information > Warning）。
-  - リモートコマンドが実行された。
-  - 障害イベントが復旧した。
-  - 障害が手動でクローズされた。
-  - 障害が抑制された。
-  - 障害の抑制が解除された。
-  - ホストのメンテナンスにより障害が抑制された。
-  - ホストのメンテナンス後に障害の抑制が解除された。
-  - 障害が症状障害に変換された。
-  - 障害が原因障害に変換された。

原因と症状の障害

概要

デフォルトでは、すべての新しい問題は原因問題として分類されます。特定の問題を手動で原因問題の症状問題として再分類することが可能です。

たとえば、停電が実際の根本原因であり、あるホストに到達できない、またはサービスがダウンしている理由である場合があります。この場合、「ホストに到達できない」や「サービスがダウンしている」という問題は、「停電」という原因問題の症状問題として分類する必要があります。

原因-症状の階層は 2 レベルのみをサポートします。すでに症状となっている問題には「従属」する症状問題を割り当てることはできません。症状問題に割り当てられた問題は、すべて同じ原因問題の症状となります。

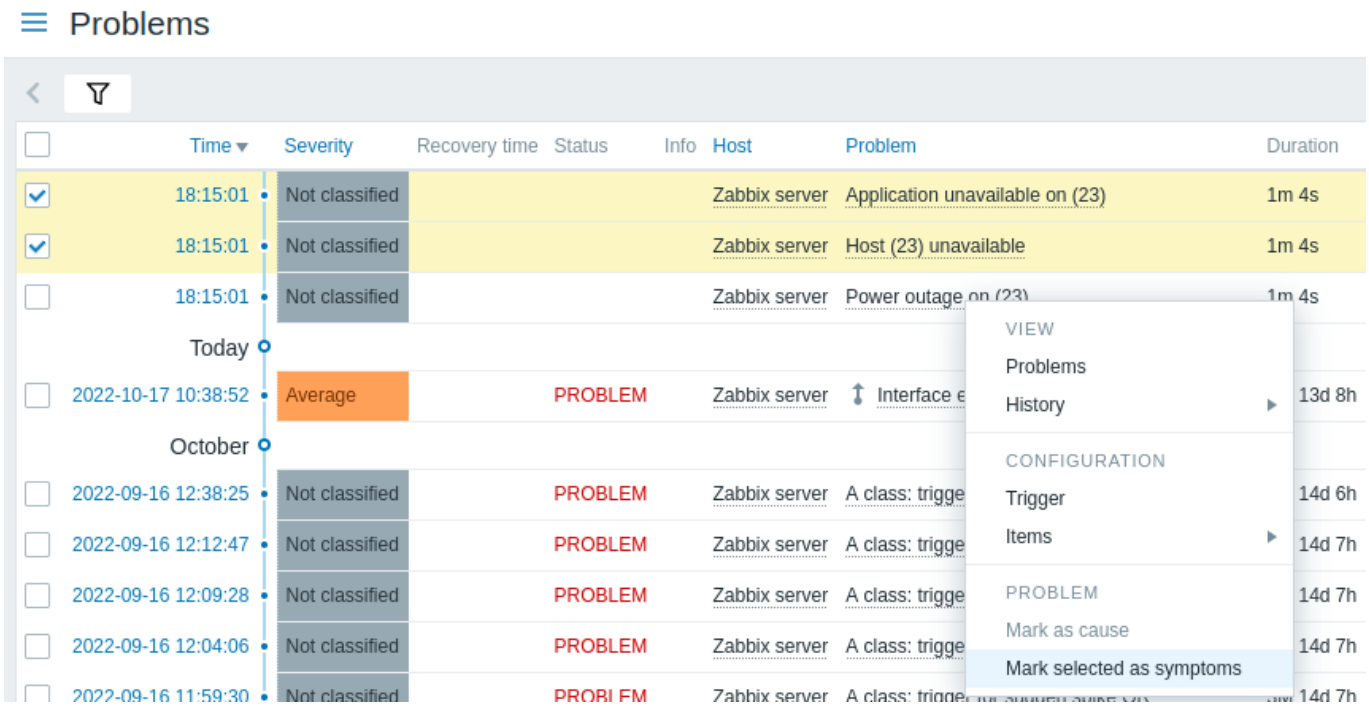
原因問題のみが、マップや深刻度別の問題や問題のあるホストなどのダッシュボードウィジェットの問題合計にカウントされます。ただし、問題のランク付けはサービスには影響しません。

症状問題は 1 つの原因問題のみにリンクできます。原因問題が解決またはクローズされても、症状問題は自動的には解決されません。

設定

問題を症状問題として再分類するには、まず問題の一覧でその問題を選択します。1 つまたは複数の問題を選択できます。

次に原因問題へ移動し、そのコンテキストメニューで 選択したものを症状としてマークオプションをクリックします。



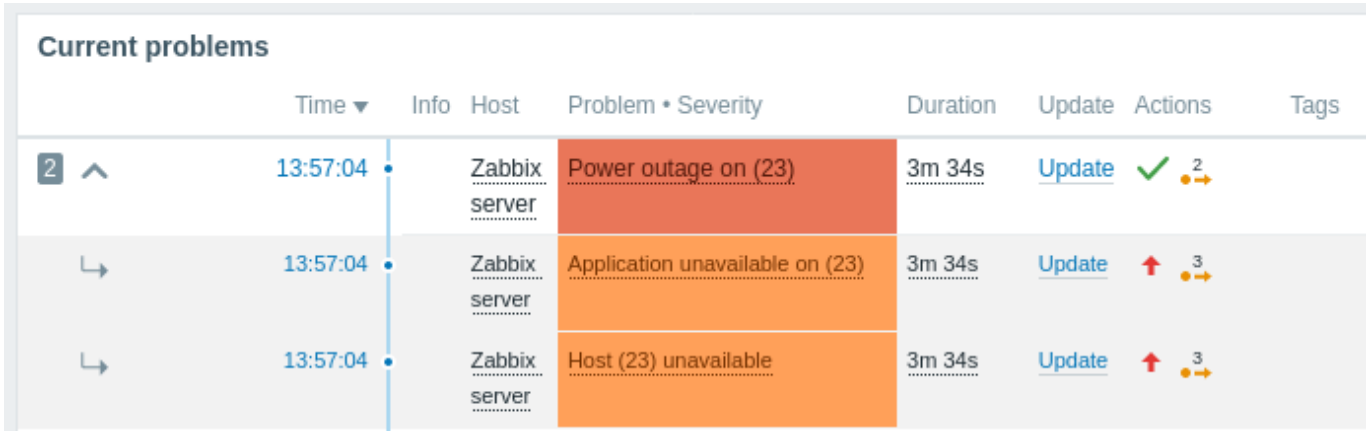
その後、選択した問題はサーバーによって原因問題の症状問題として更新されます。

問題のステータスが更新されている間は、次のいずれかの方法で表示されます。

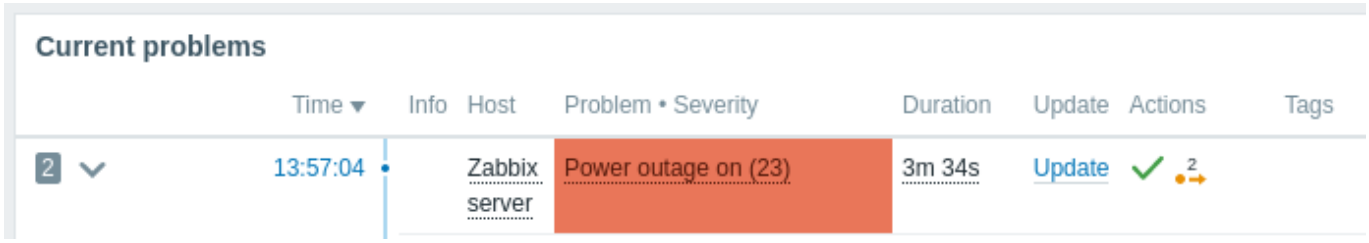
- Status 列に点滅する”UPDATING” ステータスが表示される。
- Info 列に点滅する または アイコンが表示される（これはフィルターで *Problems* のみが選択されており、そのため Status 列が表示されない場合に有効です）。

表示

症状の障害は、原因の障害の下に表示され、*Monitoring* > *Problems*（および *Problems* ダッシュボードウィジェット）で、アイコン、小さいフォント、異なる背景によって区別されます。



折りたたみ表示では、原因の障害のみが表示されます。症状の障害が存在する場合は、行の先頭にある数値と、表示を展開するためのアイコンで示されます。



また、症状の障害を通常のフォントで個別の行に追加表示することもできます。その場合は、フィルター設定またはウィジェット設定で *Show symptoms* を選択します。

障害の原因への戻し

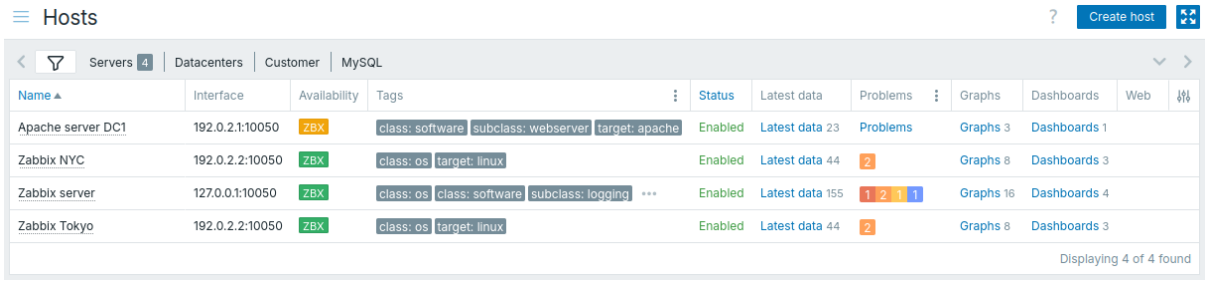
症状障害は原因障害に戻すことができます。これを行うには、次のいずれかを実行します。

- 症状障害のコンテキストメニューで 原因としてマークオプションをクリックします。
- 障害の更新 画面で 原因に変換オプションを選択し、更新をクリックします（このオプションは、複数の障害が選択されている場合にも機能します）。

ホスト

概要

Monitoring > *Hosts* セクションには、監視対象ホストの一覧が詳細情報とともに表示されます。表示される情報には、ホストインターフェース、可用性、タグ、現在の障害、ステータス（有効/無効）、およびホストの最新データ、障害履歴、グラフ、ダッシュボード、Web シナリオへ簡単に移動するためのリンクが含まれます。



ホストの詳細は、カスタマイズ可能なテーブルで表示されます。テーブルヘッダーの *Customize table* ボタンを使用すると、列の表示/非表示や並べ替え、レイアウトの初期化ができます。各列の右端をドラッグして幅を変更することもでき、境界線をダブルクリックすると幅をリセットできます。すべてのカスタマイズは即時に適用され、アクティブなフィルタータブ（デフォルトまたは保存済み）内で、現在のユーザーに対して常に記憶されます。

利用可能な列は次のとおりです。

Column	Description
Name	表示されるホスト名です。名前をクリックするとホストメニューが開きます。 名前の後ろにオレンジ色のレンチアイコン が表示されている場合、そのホストはメンテナンス中です。 列ヘッダーをクリックすると、ホストを名前の昇順または降順で並べ替えられます。
Interface	ホストのメインインターフェースが表示されます。

Column	Description
Availability	設定済みインターフェースごとのホストの可用性が表示されます。 可用性アイコンは、Zabbix サーバー上のホストインターフェースの現在の状態を表します。そのため、Web インターフェースでホストを無効にした場合、Zabbix サーバーが設定変更を同期した後に可用性が更新されます。同様に、ホストを有効にした場合も、Zabbix サーバーが設定変更を同期し、ホストをポーリングした後に可用性が更新されます。 可用性アイコンは、設定されているインターフェース種別（Agent、SNMP、IPMI、JMX）のみを表します。 アイコンにマウスオーバーすると、同じ種別のすべてのインターフェースの一覧がポップアップで表示され、詳細、状態、エラーを確認できます。Agent インターフェースの場合、ポップアップにはインターフェース（パッシブ）とアクティブチェックが表示されます。ホストにアクティブチェックのみがある場合、Agent インターフェースが設定されていなくても Agent インターフェースアイコンが表示されます。 インターフェースがないホストでは、この列は空になります。 単一のホストインターフェースの状態は、そのインターフェースを使用する有効なアイテムとホスト間の接続によって決まります。状態は次のいずれかです。 **Available** - ホストへの接続に成功しました。 **Not available** - ホストへの接続に失敗しました（タイムアウト、ファイアウォールの問題など）。 **Unknown** - ホストへの接続が試行されていないか、結果が不明です。 Zabbix サーバーがインターフェース状態をどのように判定するかの詳細は、Unknown interface status およびUnreachable/unavailable host interface settings を参照してください。 単一の種別（Agent、SNMP、IPMI、JMX）のすべてのホストインターフェースの状態は、少なくとも 1 つの有効なアイテムで使用されているインターフェースによって決まります。状態はアイコンの色で示されます。 **Green** - すべてのインターフェースが利用可能です。 **Yellow** - 少なくとも 1 つのインターフェースが利用不可で、少なくとも 1 つが利用可能または不明です。 **Red** - すべてのインターフェースが利用不可です。 **Gray** - 少なくとも 1 つのインターフェースが不明ですが、利用不可のものはありません。 **Active check availability.** ホストで少なくとも 1 つのアクティブチェックが有効な場合、アクティブチェックも上記のとおり全体の Agent インターフェース可用性に影響します。アクティブチェックの可用性を判定するために、ハートビートメッセージがエージェントのアクティブチェック用スレッドで送信されます。ハートビートメッセージの頻度は、Zabbix agent またはagent 2 の設定にある `HeartbeatFrequency` パラメータで制御されます（デフォルト 60 秒、範囲 0-3600）。アクティブチェックは、アクティブチェックのハートビートが 2 x `HeartbeatFrequency` 秒より古い場合に利用不可と見なされます。 **Note:** バージョン 6.2.x より古い Zabbix エージェントはアクティブチェックのハートビートを送信しないため、それらのホストの可用性は不明のままです。
Tags	ホストのタグが表示されます。 表示されたタグの後ろにある 3 点アイコンにマウスオーバーまたはクリックすると、すべてのタグを含むポップアップが開きます。 列ヘッダーの *Options* ボタンをクリックすると、表示するタグ数（1〜3）、タグ名の表示方法（完全表示、短縮表示、非表示）、タグ表示の優先順位を設定したり、列を複製したりできます（複製した列は名前変更または削除できます）。
Tag value	ホストのタグ値が表示されます。 列ヘッダーの *Options* ボタンをクリックすると、列名の変更、表示するタグ値の選択、列の複製ができます（複製した列は名前変更または削除できます）。
Status	ホストのステータス - *Enabled* または *Disabled*。 列ヘッダーをクリックすると、ホストをステータスの昇順または降順で並べ替えられます。
Latest data	リンクをクリックすると、*Monitoring* > *Latest data* セクションが開き、ホストから収集された最新データがすべて表示されます。 最新データを持つアイテム数が灰色で表示されます。

Column	Description
Problems	ホストの未解決の障害数が、深刻度順に表示されます。 各障害アイコンの色は深刻度を示し、数字はその深刻度の障害数を示します。アイコンをクリックすると、そのホストの *Monitoring* > *Problems* が開きます。ホストに障害がない場合は、代わりに *Problems* リンクが表示されます。 列ヘッダーの *Options* ボタンをクリックすると、*Show suppressed problems* オプションを利用できます。これにより、ホストのメンテナンスや単一の障害抑制のために通常は抑制（非表示）される障害も含めて表示できます。
Graphs	リンクをクリックすると、そのホストに設定されているグラフが表示されます。グラフ数は灰色で表示されます。 ホストにグラフがない場合、リンクは表示されません。
Dashboards	リンクをクリックすると、そのホストに設定されているダッシュボードが表示されます。ダッシュボード数は灰色で表示されます。 ホストにダッシュボードがない場合、リンクは表示されません。
Web	リンクをクリックすると、そのホストに設定されている Web シナリオが表示されます。Web シナリオ数は灰色で表示されます。 ホストに Web シナリオがない場合、リンクは表示されません。
Custom text	カスタムテキスト、マクロ値、またはその両方の組み合わせが表示されます。 列ヘッダーの *Options* ボタンをクリックすると、列名の変更、カスタムテキストまたはマクロの入力、列の複製ができます（複製した列は名前変更または削除できます）。 サポートされるマクロ: `{HOST.*}`（`{HOST.METADATA}` を除く）、`{INVENTORY.*}`、およびユーザーマクロ。

ボタン

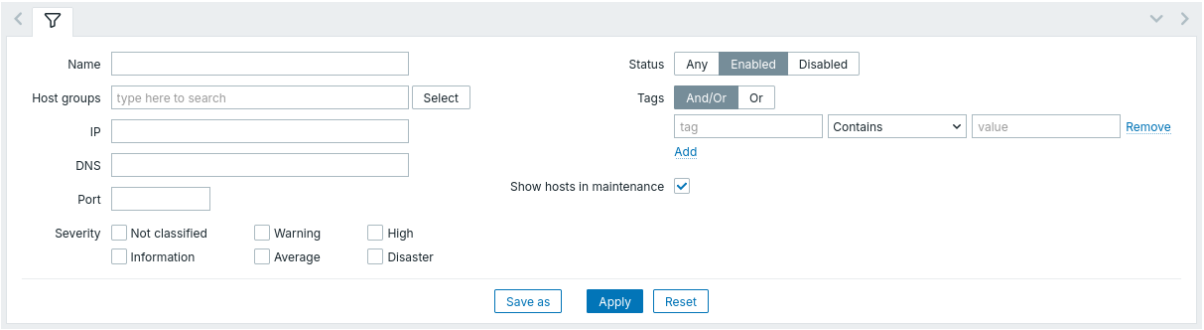
ホストの作成をクリックすると、新しいホストを作成できます。このボタンは、管理者および特権管理者ユーザーのみ使用できます。

すべてのセクションに共通する表示モードボタンについては、監視ページで説明しています。

フィルターの使用

フィルターを使用して、関心のあるホストだけを表示できます。
検索性能を向上させるため、データはマクロが未展開の状態で検索されます。

フィルターは表の上にあります。ホストは次の条件で絞り込めます。名前、ホストグループ、IP または DNS、インターフェースポート、タグ、障害の深刻度、ステータス（有効/無効/任意）。また、現在メンテナンス中のホストを表示するかどうかも選択できます。



Parameter	Description
Name	表示されるホスト名でフィルターします。
Host groups	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。
IP	IP アドレスでフィルターします。
DNS	DNS 名でフィルターします。
Port	ポート番号でフィルターします。
Severity	障害の深刻度でフィルターします。既定では、すべての深刻度の障害が表示されます。障害は抑制されていない場合に表示されます。
Status	ホストのステータスでフィルターします。

Parameter	Description
Tags	ホストタグの名前と値でフィルターします。ホストは、ホストレベルのタグに加えて、ネストされたテンプレートを含む、リンクされたすべてのテンプレートのタグでもフィルターできます。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです。 **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別） **Contains** - 入力した文字列をタグ値に含む指定タグ名を含める（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別） **Does not contain** - 入力した文字列をタグ値に含む指定タグ名を除外する（部分一致、大文字と小文字を区別しない） 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です
Show hosts in maintenance	チェックボックスをオンにすると、メンテナンス中のホストを表示します（既定で表示されます）。

フィルターの保存

お気に入りのフィルター設定はタブとして保存でき、フィルターの上にある該当タブをクリックしてすばやくアクセスできます。

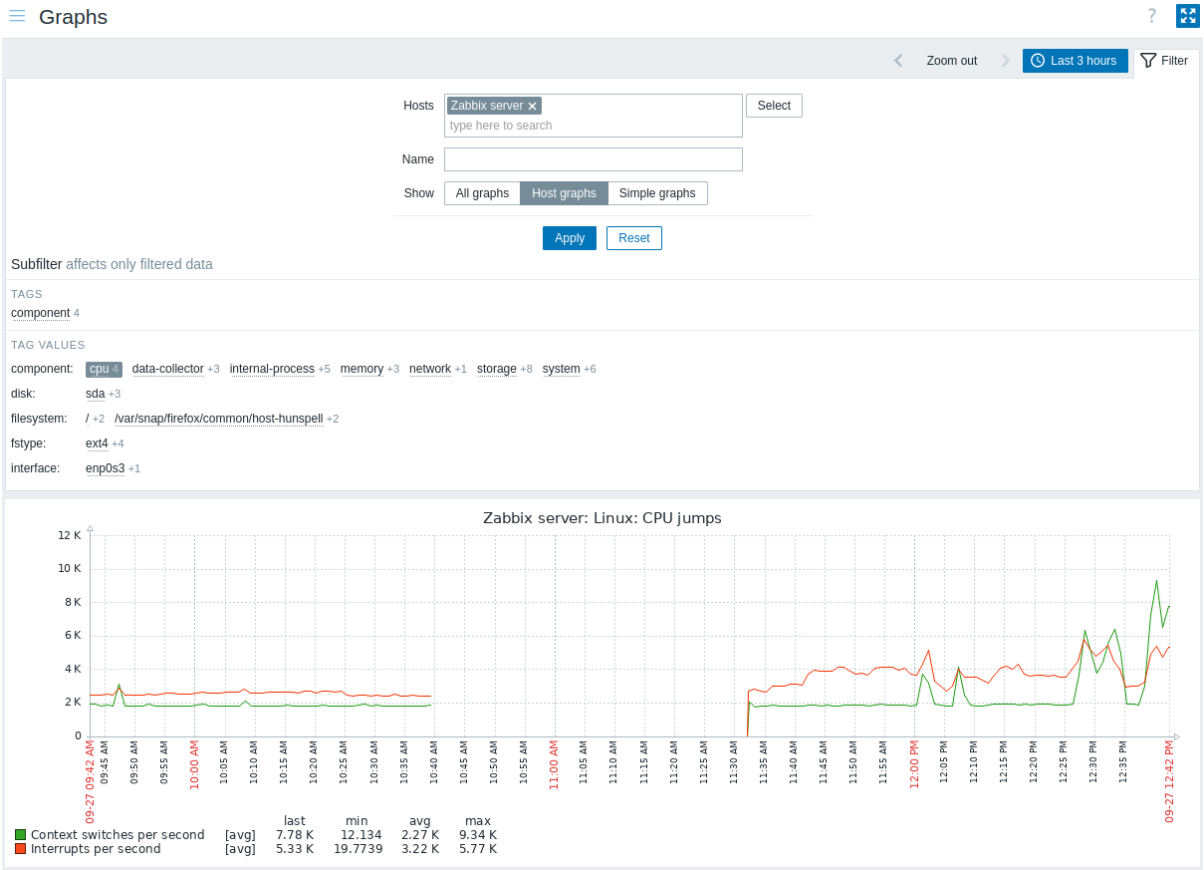
詳細はフィルターの保存を参照してください。

グラフ

概要

ホストのグラフは、*Monitoring* > *Hosts* から、該当するホストのグラフをクリックしてアクセスできます。

ホストに設定されている任意のカスタムグラフ や、任意のシンプルグラフ を表示できます。



グラフは次の順で並べ替えられます。

- グラフ名（カスタムグラフ）
- アイテム名（シンプルグラフ）

無効化されたホストのグラフにもアクセスできます。

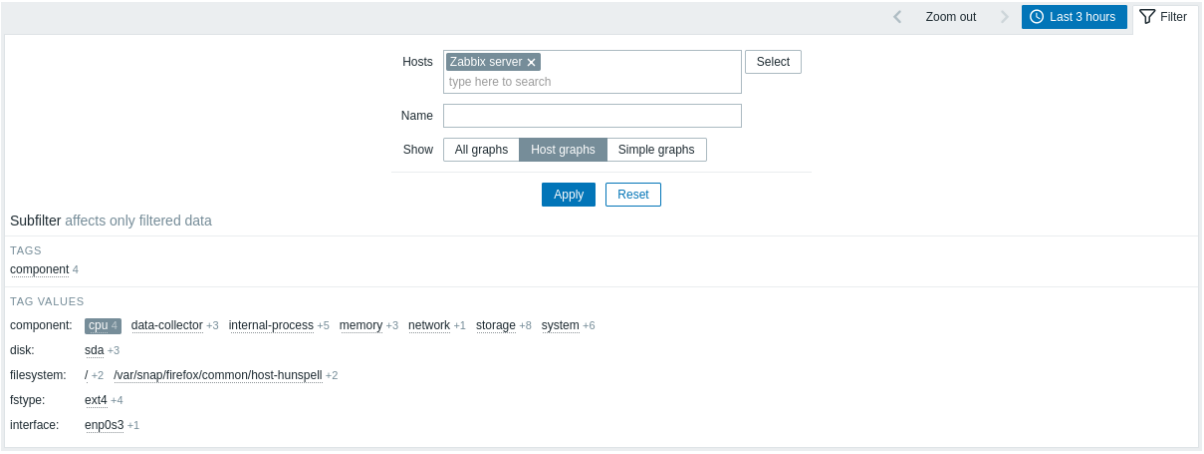
期間セレクター

グラフの上にある期間セレクターに注目してください。よく使う期間をワンクリックで選択できます。

参考: 期間セレクター

フィルターの使用

特定のグラフを表示するには、フィルターで選択します。フィルターでは、ホスト、グラフ名、および表示オプション（すべて/ホストグラフ/シンプルグラフ）を指定できます。



フィルターでホストを選択していない場合、グラフは表示されません。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターで適用したフィルタリングをさらに変更できます。

サブフィルターには、関連するグラフにすばやくアクセスするためのクリック可能なリンクが含まれています。グラフは、共通のエンティティ（タグ名またはタグ値）によって関連付けられています。タグ名またはタグ値をクリックすると、灰色の背景で強調表示され、グラフがすぐにフィルタリングされます（メインフィルターで 適用をクリックする必要はありません）。別のタグ名またはタグ値をクリックすると、フィルタリングされた結果に追加されます。タグ名またはタグ値をもう一度クリックすると、そのフィルタリングが解除されます。

サブフィルターは、フィルタリングされたデータに基づいて生成されます。このデータは最大 1000 レコードに制限されています。サブフィルターにさらに多くのレコードを表示するには、*Administration* > *General* > *GUI* にある *Limit for search and filter results* パラメータの値を増やす必要があります。

メインフィルターとは異なり、サブフィルターは各テーブル更新リクエストとともに更新されます。これにより、利用可能なフィルタリングオプションとその件数を常に最新の状態に保つことができます。

表示されるエンティティ数は、水平方向では 100 個に制限されています。それを超える場合は、末尾に三点アイコンが表示されます。このアイコンはクリックできません。垂直方向のリスト（値を含むタグなど）は 20 項目に制限されています。それを超える場合は、三点アイコンが表示されます。このアイコンはクリックできません。

各クリック可能なエンティティの横に表示される数字は、メインフィルターの結果に含まれる、そのエンティティに関連するグラフの数を示します。

1 つのエンティティを選択すると、その他の利用可能なエンティティの件数の前にプラス記号が表示され、現在の選択に追加できるグラフの数が示されます。

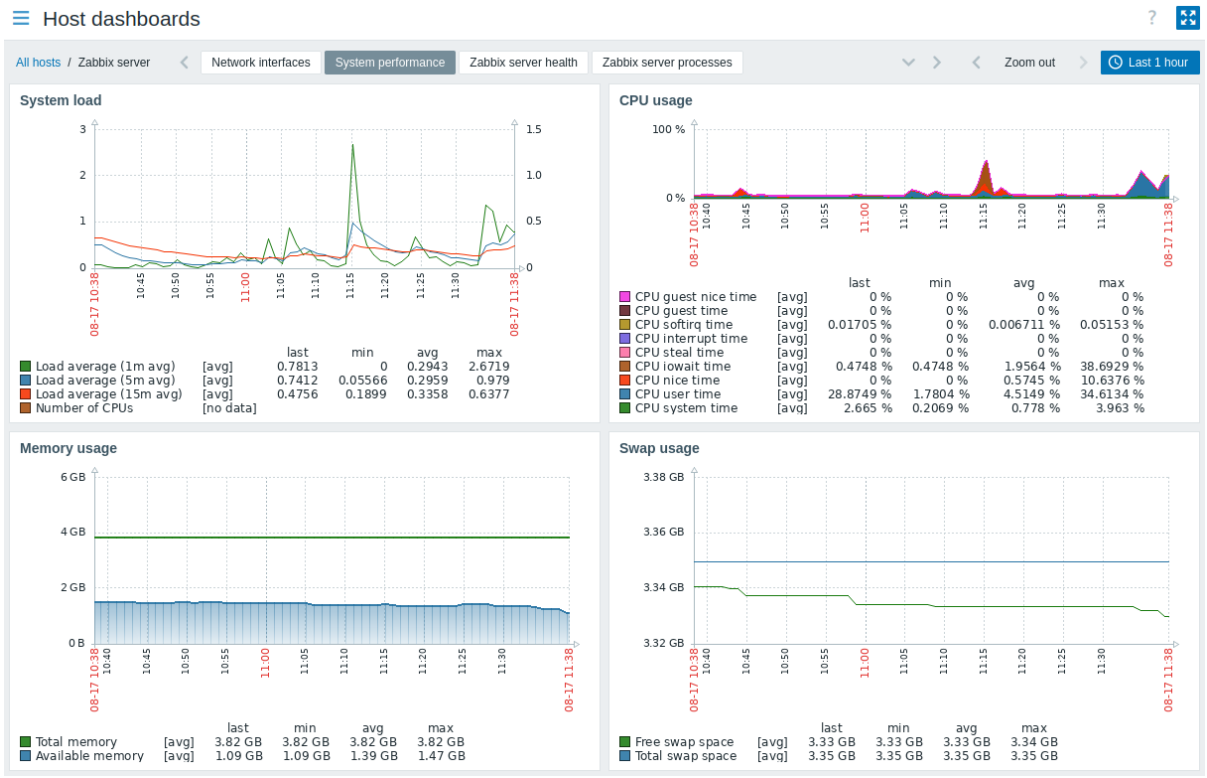
ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoring ページで説明しています。

ホストダッシュボード

概要

ホストダッシュボードはグローバルダッシュボードと似ていますが、ホストダッシュボードには所有者がなく、選択したホストのデータのみが表示されます。



ホストダッシュボードを表示しているときは、次のいずれかをクリックして、設定済みのダッシュボードを切り替えることができます。

- ダッシュボードのタブ
- ヘッダーの下にある  矢印ボタン
- ヘッダーの下にある  矢印ボタン。このボタンをクリックすると、利用可能なすべてのホストダッシュボードの一覧が表示されます。

Monitoring > *Hosts* セクションに切り替えるには、左上のヘッダーの下にある *All hosts* ナビゲーションリンクをクリックします。

設定

ホストダッシュボードはテンプレートレベルで設定します。テンプレートがホストにリンクされると、そのホスト用のホストダッシュボードが生成されます。ホストダッシュボードはダッシュボードセクションで設定できないことに注意してください。このセクションはグローバルダッシュボード専用です。

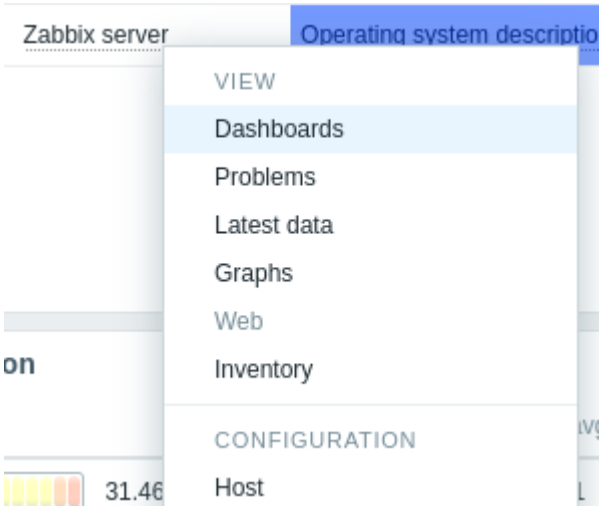
ホストダッシュボードのウィジェットも、テンプレートレベルでのみ設定できます（更新間隔の変更を除く）。さらに、ホストダッシュボードのウィジェットは、同じテンプレート内の他のホストダッシュボードにのみコピーできます。グローバルダッシュボードのウィジェットはホストダッシュボードにコピーできないことに注意してください。

Note:
Zabbix 5.2 以前では、ホストダッシュボードはホストスクリーンでした。スクリーンを含む古いテンプレートをインポートする場合、スクリーンのインポートは無視されます。

アクセス

ホストダッシュボードには、次の方法でアクセスできます。

- グローバル検索でホスト名を検索した後、検索結果に表示される *Dashboards* リンクをクリックする。
- *Inventory* > ホスト でホスト名をクリックした後、ホスト概要に表示される *Dashboards* リンクをクリックする。
- ホストメニュー から *Dashboards* をクリックする。

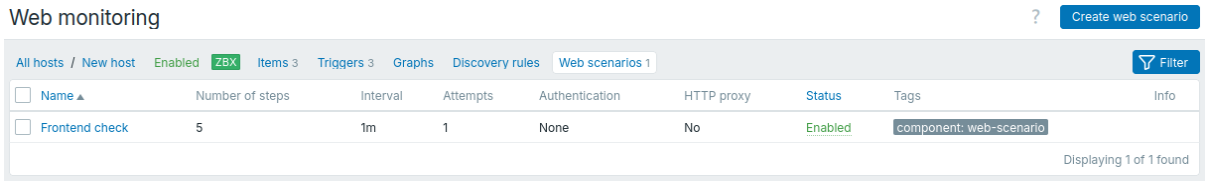


ホストダッシュボードは、グローバルダッシュボード専用の *Dashboards* セクションからは直接アクセスできないことに注意してください。

Web シナリオ

概要

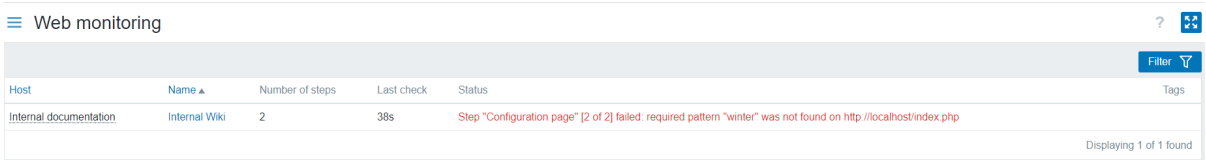
ホストのweb scenario 情報は、対象のホストの *Monitoring* > *Hosts* で Web をクリックすると表示できます。



ホスト名をクリックすると、ホストメニュー が表示されます。無効化されたホストのデータも参照できます。無効化されたホスト名は赤色で表示されます。

web scenario のステータスは次のいずれかです。

- OK (緑) - すべてのステップが成功し、Zabbix がその scenario のデータを収集しています。
- Error (赤) - 1 つ以上のステップが失敗しています（どのパラメーターを修正すべきかの手がかりはエラーメッセージを参照してください）。
- Unknown (灰色) - まだステップのステータスが利用できないか、データが収集されていません。



1 ページあたりに表示される scenario の最大数は、ユーザープロファイルの *Rows per page* 設定 に依存します。

デフォルトでは、直近 24 時間以内の値のみが表示されます。この制限は、最新データの件数が多いページの初回読み込み時間を改善する目的で導入されています。*Administration* > *General* > *GUI* メニューセクションの *Max history display period* パラメーターの値を変更することで、この期間を延長できます。

scenario 名は、より詳細な統計情報へのリンクになっています。



フィルターの使用

このページには、選択したホストのすべての Web シナリオの一覧が表示されます。*Monitoring* > *Hosts* ページに戻らずに別のホストまたはホストグループの Web シナリオを表示するには、フィルターでそのホストまたはグループを選択します。タグに基づいてシナリオを絞り込むこともできます。

ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoring ページで説明しています。

最新データ

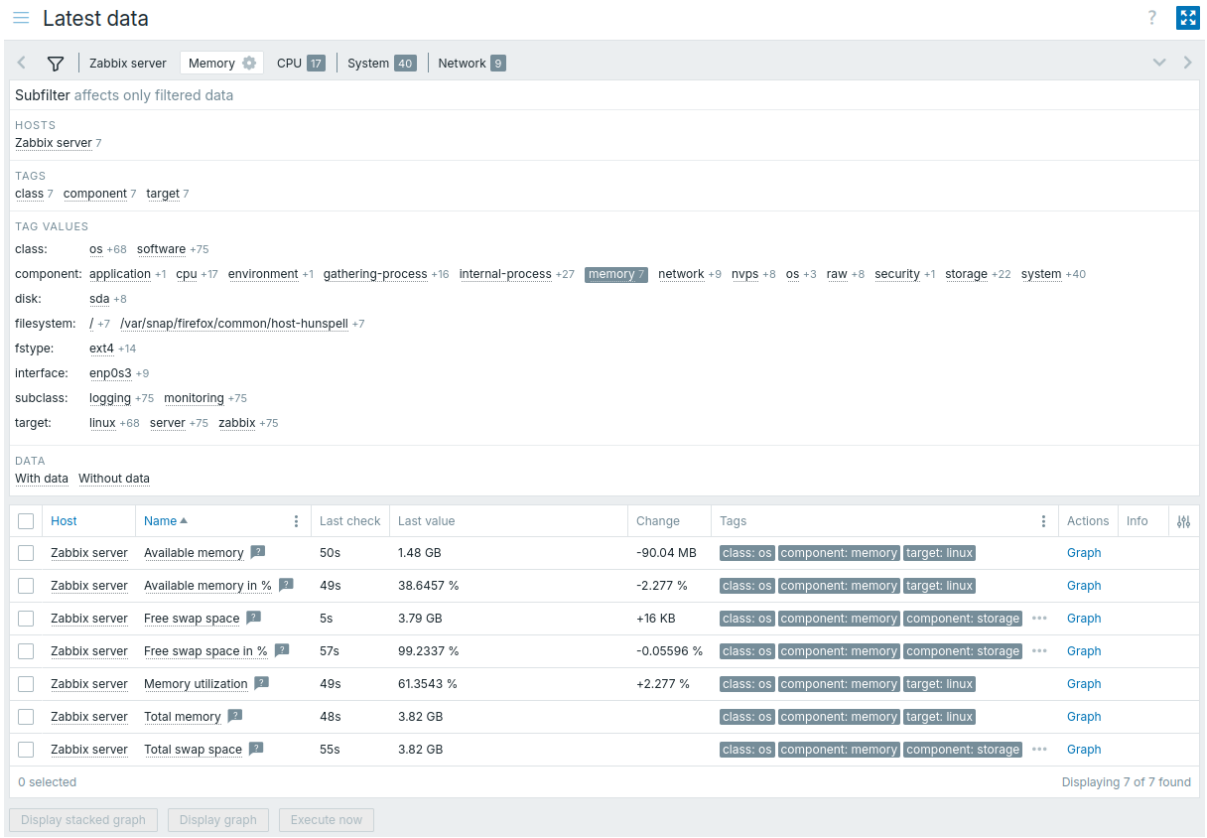
概要

Monitoring > *Latest data* セクションには、アイテムによって収集された最新の値が表示されます。

このセクションには、次の要素が含まれます。

- フィルター
- サブフィルター
- アイテム一覧

Note:
サブフィルターとアイテム一覧は、フィルターが設定され、表示できる結果がある場合にのみ表示されます。



最新値の詳細は、カスタマイズ可能な表に表示されます。表ヘッダーの テーブルをカスタマイズボタンを使用すると、列の表示/非表示や並べ替え、レイアウトの初期状態へのリセットができます。また、各列の右境界線をドラッグして幅を変更できます。境界線をダブルクリックすると、幅がリセットされます。すべてのカスタマイズは即時に適用され、アクティブなフィルター タブ（デフォルトまたは保存済み）内で、現在のユーザーに対して常に記憶されます。

利用可能な列は次のとおりです。

Column	Description
ホスト	アイテムが属するホストの名前です。 名前をクリックすると、ホストのコンテキストメニューが表示されます。 ホストがメンテナンス中の場合は、ホスト名の後ろにオレンジ色のレンチ アイコンが表示されます。 ホストが無効化されている場合は、ホスト名が赤色で表示されます。無効化されたホストのデータ(グラフやアイテム値一覧を含む）は、*Latest data* セクションで参照できることに注意してください。
名前	アイテムの名前です。 名前をクリックすると、アイテムメニューが表示されます。 説明が設定されているすべてのアイテムについて、アイテム名の横に疑問符 アイコンが表示されます。アイコンにマウスオーバーすると、アイテムの説明を含むツールチップが表示されます。 列ヘッダーの オプションボタンをクリックすると、アイテムキーを表示オプションにアクセスでき、各アイテム名の下にアイテムキーを表示できます。
間隔	アイテムの更新間隔です。
履歴	アイテムの履歴保存期間です。
トレンド	アイテムのトレンド保存期間です。
タイプ	アイテムのタイプです。
最終チェック	最後のアイテムチェックからの経過時間です。
最終値	アイテムの最新値です。 値は単位変換と値マッピングを適用した状態で表示されます。値にマウスオーバーすると、生データ（8192 文字で切り詰められます）が表示されます。 バイナリ値は、実際の値の代わりにプレースホルダー（*binary value*）として表示されます。 デフォルトでは、直近 24 時間以内に受信した値のみが表示されます。この制限により、最新データの大きなページでも初期読み込み時間が改善されます。これを延長するには、*Administration* > *General* > *GUI* の *Max history display period* パラメータの値を更新してください。

Column	Description
変化	前回の値と最新の値の差です。 更新頻度が 1 日以上のアイテムでは、(デフォルト設定では) 変化量は表示されません。この場合、最終値が 24 時間以上前に受信されたものであれば、最終値自体も表示されません。
タグ	アイテムに関連付けられたタグです。 タグをクリックすると、サブフィルター でそのタグが有効になり、そのタグ (およびサブフィルターで選択された他のタグ) でアイテム一覧が絞り込まれます。一度クリックしたタグはクリックできなくなるため、タグによる絞り込みを調整するにはサブフィルターを使用する必要があります。 列ヘッダーの ⋮ オプションボタンをクリックすると、表示するタグ数 (1〜3)、タグ名の表示方法 (完全表示、短縮表示、または非表示)、タグ表示の優先順位を設定したり、列を複製したりできます (複製した列は名前変更または削除できます)。
タグ値	アイテムのタグ 値が表示されます。 列ヘッダーの ⋮ オプションボタンをクリックすると、列名の変更、表示するタグ値の選択、または列の複製ができます (複製した列は名前変更または削除できます)。
アクション	アイテム値のシンプルグラフ/履歴 へのリンクです。
情報	アイテムに関する追加情報です。 アイテムにエラーがある場合 (たとえば、サポート対象外になった場合) は、情報 アイコンが表示されます。詳細はアイコンにマウスオーバーしてください。
カスタムテキスト	カスタムテキスト、マクロ値、またはその両方の組み合わせが表示されます。 列ヘッダーの ⋮ オプションボタンをクリックすると、列名の変更、カスタムテキストまたはマクロの入力、または列の複製ができます (複製した列は名前変更または削除できます)。 サポートされるマクロ: {HOST.*} ({HOST.METADATA} を除く)、{INVENTORY.*}、{ITEM.*} ({ITEM.STATE.ERROR} を除く)、およびユーザーマクロ。

ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoringページで説明されています。

一括操作

一覧の下にあるボタンでは、1 つまたは複数の選択したアイテムに対して一括操作を実行できます。

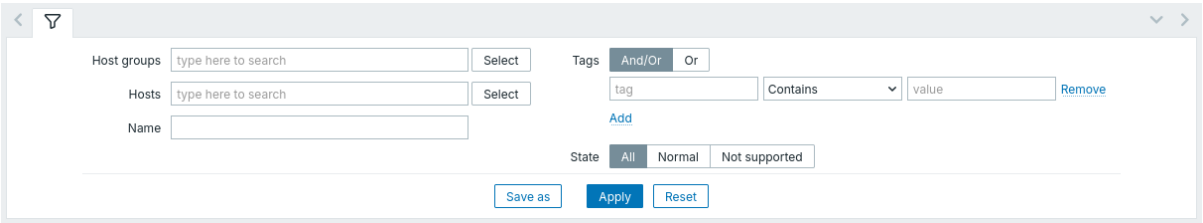
- 積み重ねグラフを表示 - 積み重ねアドホックグラフ を表示します。
- グラフを表示 - シンプルなアドホックグラフ を表示します。
- 今すぐ実行 - 新しいアイテム値のチェックを直ちに実行します。パッシブチェックでのみサポートされます (詳細はこちら を参照してください)。このオプションは、読み取り/書き込み権限を持つホストでのみ利用できます。読み取り専用権限のホストでこのオプションにアクセスできるかどうかは、読み取り専用ホストで「今すぐ実行」を呼び出すというユーザー権限 オプションに依存します。

これらのオプションを使用するには、該当するアイテムの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるアイテムのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決の状態で検索されます。

フィルター アイコンは、アイテム一覧とサブフィルターの上にあります。クリックするとフィルターが展開されます。



フィルターを使用すると、ホストグループ、ホスト、アイテム名、タグ、状態でアイテム一覧を絞り込むことができます。フィルターで親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。タグによるフィルタリングの詳細については、監視 > 障害 を参照してください。

詳細を表示フィルターオプションを使用すると、アイテムに表示される情報を拡張できます。これを選択すると、アイテムの更新間隔、ヒストリとトレンドの設定、アイテムのタイプ、アイテムのエラー (正常/未サポート) などの詳細が表示されます。

フィルターの保存

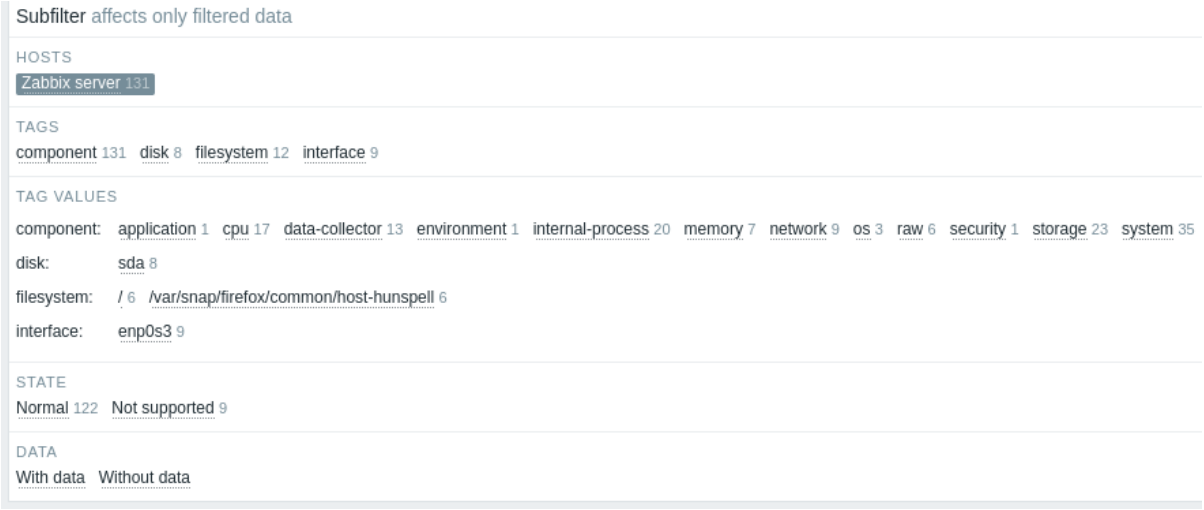
お気に入りのフィルター設定はタブとして保存でき、フィルターの上にある該当のタブをクリックすることで、すばやくアクセスできます。

詳細は、フィルターの保存を参照してください。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターのフィルタリングをさらに変更できます。

サブフィルターには、関連するアイテムにすばやくアクセスするためのクリック可能なリンクが含まれています。アイテムは、共通のエンティティ（ホスト、タグ名または値、アイテムの状態、データのステータス）によって関連付けられます。エンティティをクリックすると、そのエンティティが灰色の背景で強調表示され、アイテムがすぐにフィルタリングされます（メインフィルターで 適用をクリックする必要はありません）。別のエンティティをクリックすると、フィルタリングされた結果に追加されます。エンティティをもう一度クリックすると、そのフィルタリングが解除されます。



サブフィルターは、フィルタリングされたデータに基づいて生成されます。対象となるデータは最大 1000 レコードに制限されています。各 100 アイテムを持つホストが 20 台ある場合（合計 2000 レコード）、サブフィルターにはホストの半分しか表示されません。サブフィルターにさらに多くのレコードを表示するには、検索およびフィルター結果の上限パラメータの値を増やす必要があります（管理 > 一般 > *GUI*）。

メインフィルターとは異なり、サブフィルターはテーブルの更新リクエストごとに更新されます。これにより、利用可能なフィルタリングオプションとその件数が常に最新の情報になります。

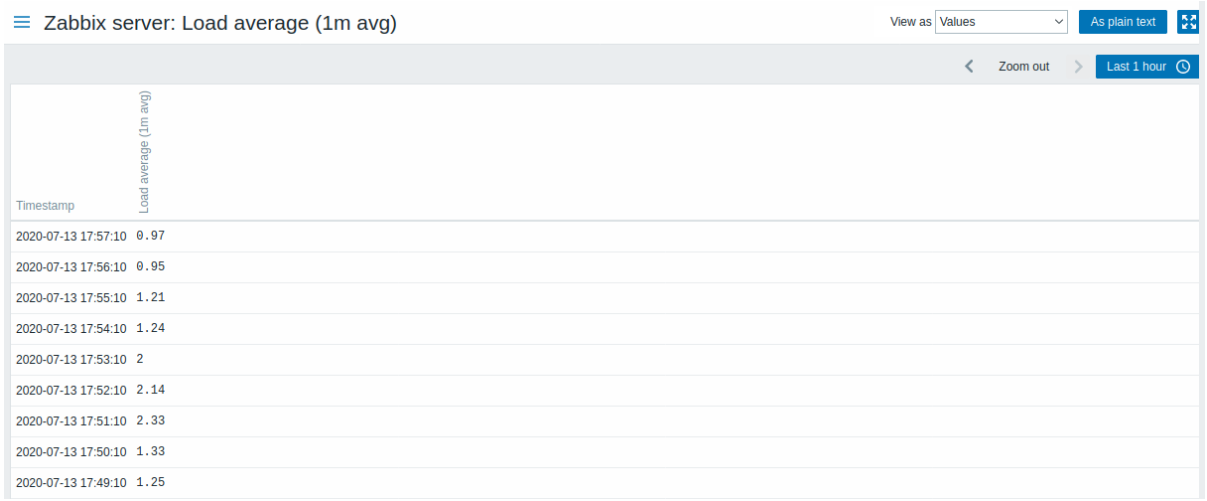
各エンティティグループ（ホスト、タグ、タグの値など）では、最大 10 行のエンティティが表示されます。エンティティがさらに存在する場合は、リストの末尾にある 3 点 *** アイコンをクリックすると、最大 1000 件（Web インターフェースの定義における SUBFILTER_VALUES_PER_GROUP の値）まで表示するようにリストを展開できます。タグの値では、対応する値とともに最大 200 個のタグ名を表示するようにリストを展開できます。一度完全に展開すると、リストを折りたたむことはできません。

クリック可能な各エンティティの横に表示される数字は、そのエンティティにグループ化されたアイテムの数を示します（メインフィルターの結果に基づきます）。エンティティをクリックすると、その他の利用可能なエンティティの横にプラス記号付きの数字が表示されます。この数字は、現在の選択に追加できるアイテムの数を示します。アイテムのないエンティティは、サブフィルターで以前に選択されていない限り表示されません。

グラフと履歴

アイテム一覧の *Graph/History* 列には、次のリンクがあります。

- **History** - すべてのテキストアイテム向けで、過去のアイテム値の履歴を表示する一覧（*Values/500 latest values*）に移動します。
- **Graph** - すべての数値アイテム向けで、シンプルグラフ に移動します。グラフが表示されると、右上のドロップダウンから *Values/500 latest values* に切り替えることもできます。



この一覧に表示される値は生の値であり、後処理は適用されません。

Note:
表示される値の総数は、*Administration* > *General* > *GUI* で設定される *Limit for search and filter results* パラメーターの値によって決まります。

マップ

概要

Monitoring > *Maps* セクションでは、ネットワークマップを設定、管理、表示できます。

このセクションを開くと、最後にアクセスしたマップ、またはアクセス権のあるすべてのマップの一覧のいずれかが表示されます。

すべてのマップは、公開または非公開にできます。公開マップはすべてのユーザーが利用でき、非公開マップは所有者と、そのマップを共有されたユーザーのみがアクセスできます。

マップ一覧



表示されるデータ:

Column	Description
Name	マップの名前です。名前をクリックすると、マップを表示できます。
Width	マップの幅が表示されます。
Height	マップの高さが表示されます。
Actions	2 つの操作を利用できます: **Properties** - 一般的なマップのプロパティを設定します **Edit** - マップ要素を追加するためのグリッドにアクセスします

新しいマップを設定するには、右上隅の *Create map* ボタンをクリックします。
YAML、XML、または JSON ファイルからマップをインポートするには、右上隅の *Import* ボタンをクリックします。マップをインポートしたユーザーが、その所有者として設定されます。

一覧の下にある 2 つのボタンでは、一括編集のオプションを利用できます:

- *Export* - マップを YAML、XML、または JSON ファイルにエクスポートします
- *Delete* - マップを削除します

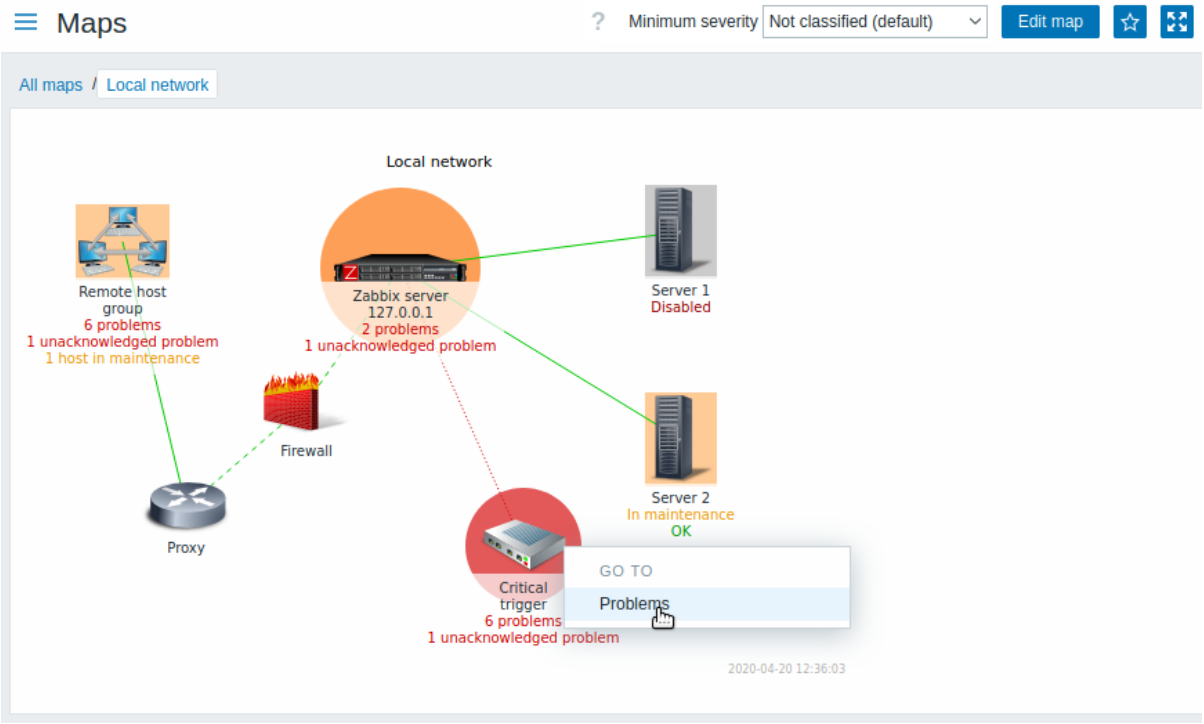
これらのオプションを使用するには、各マップの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のマップのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

マップの表示

マップを表示するには、すべてのマップの一覧でその名前をクリックします。



マップのタイトルバーにあるドロップダウンを使用して、表示する問題トリガーの最も低い深刻度レベルを選択できます。*default* と表示されている深刻度は、マップの設定で指定されたレベルです。マップにサブマップが含まれている場合、サブマップへ移動すると、上位マップの深刻度が保持されます（ただし、*Not classified* の場合は除きます。この場合、サブマップには引き継がれません）。

アイコンの強調表示

マップ要素が障害状態の場合、円形の円で強調表示されます。円の塗りつぶし色は、障害の深刻度の色に対応します。選択した深刻度以上の障害のみ、要素とともに表示されます。すべての障害が認識済みの場合は、円の周囲に太い緑色の枠が表示されます。

さらに、次のように表示されます。

- メンテナンス中のホストは、オレンジ色の塗りつぶされた四角形で強調表示されます。マップ要素がホストの場合、メンテナンスの強調表示が障害の深刻度による強調表示よりも優先されることに注意してください。
- 無効（監視対象外）のホストは、灰色の塗りつぶされた四角形で強調表示されます。

マップ設定でアイコンの強調表示チェックボックスが選択されている場合に、強調表示が表示されます。

最近の変更マーカー

要素の周囲にある内向きの赤い三角形は、トリガーの状態が最近変更されたことを示します。これは、過去 30 分以内に発生した変更です。
これらの三角形は、マップの設定でトリガーの状態変更時に要素をマークチェックボックスが選択されている場合に表示されます。

リンク

マップ要素をクリックすると、利用可能なリンクを含むメニューが開きます。ホスト名をクリックすると、ホストメニューが表示されます。

ボタン

右側のボタンでは、以下のオプションを利用できます。

Edit map	マップ内容の編集に移動します。
☆	マップをダッシュボードの「お気に入り」ウィジェットに追加します。
★	このマップはダッシュボードの「お気に入り」ウィジェットに追加されています。クリックすると、「お気に入り」ウィ

すべてのセクションに共通する表示モードのボタンについては、監視 ページで説明しています。

マップの読み上げ可能な概要

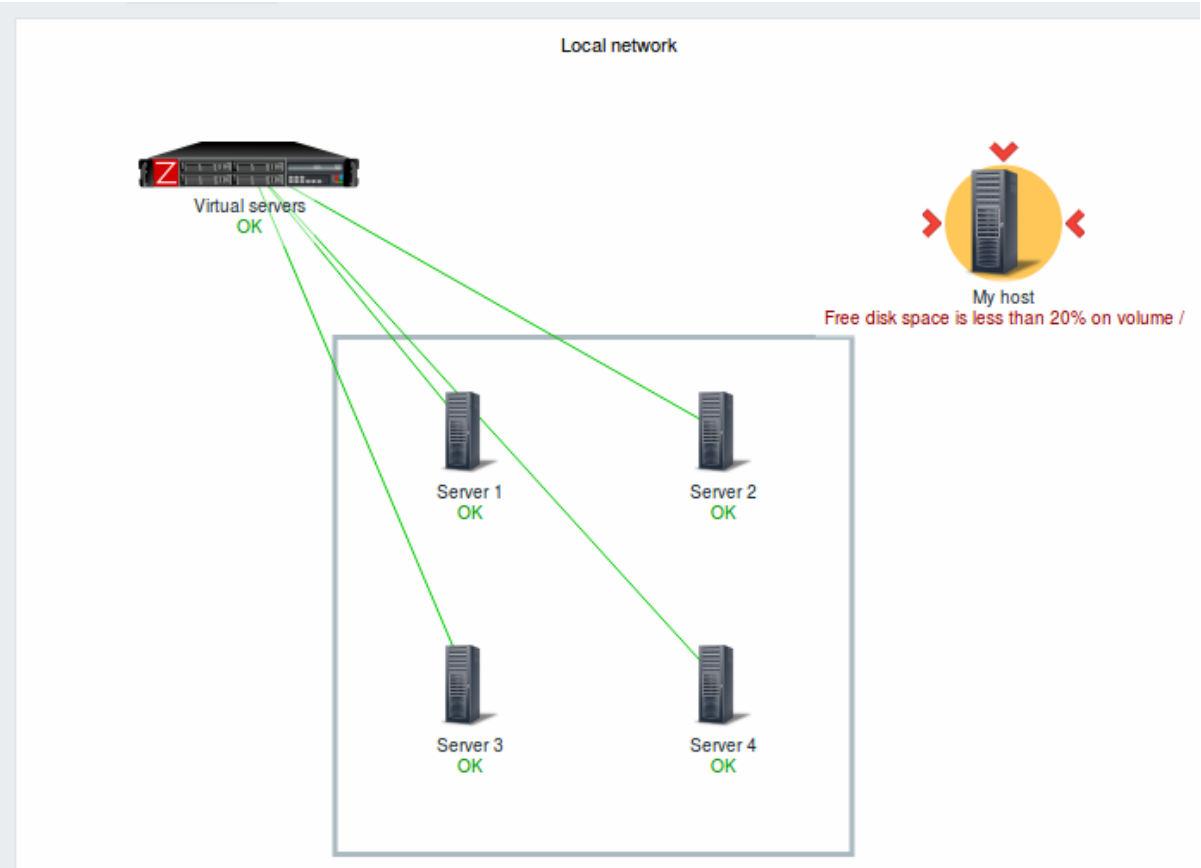
非表示の「aria-label」プロパティを使用すると、スクリーンリーダーでマップ情報を読み上げることができます。マップ全体の説明と個々の要素の説明は、次の形式で利用できます。

- マップの説明: < マップ名 >, < 障害状態にあるアイテム数 / アイテム総数 >, < 障害の総数 >.
- 1 つの障害がある要素の説明: < 要素タイプ >, ステータス < 要素のステータス >, < 要素名 >, < 障害の説明 >.
- 複数の障害がある要素の説明: < 要素タイプ >, ステータス < 要素のステータス >, < 要素名 >, < 障害数 >.
- 障害がない要素の説明: < 要素タイプ >, ステータス < 要素のステータス >, < 要素名 >.

例えば、次のマップでは、以下の説明を利用できます。

'ローカルネットワーク、6個の要素のうち1個が障害状態、障害の総数は1。ホスト、ステータス 障害、My host、ボ'

次のマップに対して:



ネットワークマップの参照

ネットワークマップは sysmapid と mapname の両方の GET パラメータで参照できます。例えば、

http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network

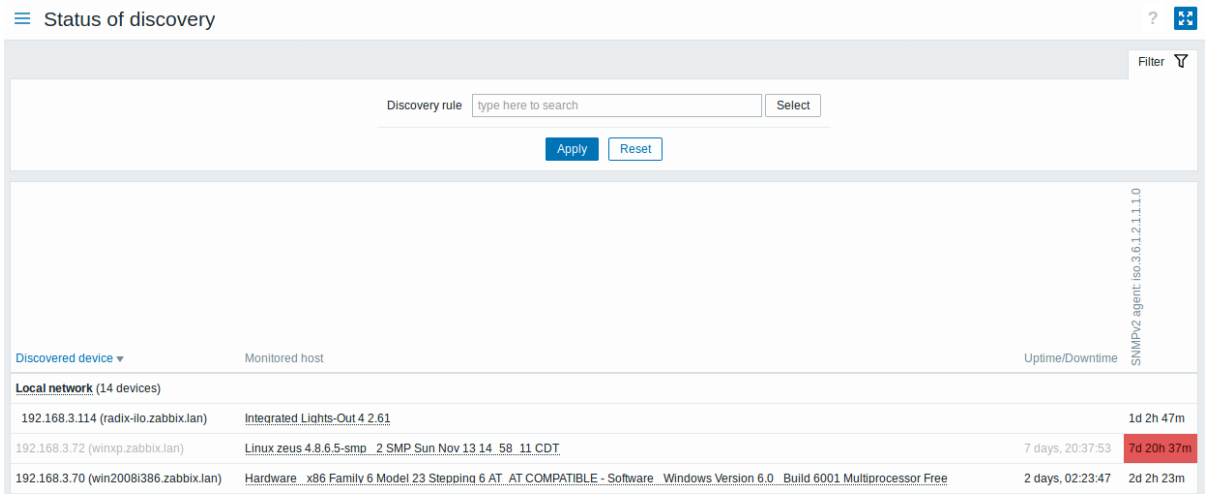
指定された名前のマップを開きます。(Local network)

sysmapid（マップ ID）と mapname（マップ名）の両方が指定されている場合、mapname の優先度が高くなります。

ディスカバリ

概要

監視 > ディスカバリセクションには、ネットワークディスカバリの結果が表示されます。検出されたデバイスは、ディスカバリルールごとに並べ替えられます。



表示されるデータ:

列	説明
検出されたデバイス	検出されたデバイスは、ディスカバリルールごとにグループ化されて一覧表示されます。ディスカバリルールをクリックすると、ディスカバリルール設定フォームへのリンクを含むルールメニューが表示されます。 デバイスに複数の IP アドレスがある場合（ディスカバリルールのデバイスの一意性の基準によって決定）、それらはまとめて一覧表示されます。最初にデバイスのプライマリ IP アドレスが表示され、その後に他の IP アドレスがインデントされて表示されます。プライマリ IP アドレスは、監視対象ホスト列にホストが表示されている IP アドレスのうち最も小さいものです。そのようなアドレスがない場合は、デバイスの IP アドレスのうち最も小さいものになります。 デバイスは、プライマリ IP アドレスの昇順または降順で並べ替えられます。デバイスの他の IP アドレスは、常にプライマリ IP アドレスの下に昇順で表示されます。
監視対象ホスト	デバイスがすでに監視されている場合、この列にホスト名が表示されます。 ホスト名をクリックすると、ホストメニューが表示されます。
稼働時間/停止時間	前回の検出後にデバイスが検出されている、または失われている期間がこの列に表示されます。 複数の IP アドレスを持つデバイスの場合、期間はプライマリ IP アドレスに対してのみ表示されます。
ディスカバリチェック	検出された各デバイスについて、個々のサービス（ディスカバリチェック）の状態が表示されます。赤いセルは、サービスが停止していることを示します。サービスの稼働時間または停止時間もセル内に表示されます。 この列は、少なくとも 1 台の検出されたデバイスでサービスが検出された場合にのみ表示されます。

Note:
このセクションに表示されるディスカバリルールの数と、検出されたデバイスの合計数は、管理 > 一般設定 > *GUI*の検索およびフィルター結果の上限パラメータによって制限されます。上限を超えたルールまたはデバイスは表示されず、そのことを示す表示もありません。

ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoring ページを参照してください。

フィルターの使用

フィルターを使用して、関心のあるディスカバリルールのみを表示できます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

フィルターで何も選択されていない場合は、有効なすべてのディスカバリルールが表示されます。特定のディスカバリルールを表示するには、フィルターにその名前を入力し始めます。一致する有効なディスカバリルールがすべて選択候補として一覧表示されます。複数のディスカバリルールを選択できます。

APM

概要

APM メニューでは、Zabbix のアプリケーションパフォーマンス監視（APM）機能を使用できます。このメニューには、テレメトリデータを収集するためのオープン標準である OpenTelemetry を使用して、アプリケーションやサービスから送信されたテレメトリデータが表示されます。このデータにより、アプリケーションやサービスを通過するリクエストの流れを確認し、パフォーマンスを測定して、処理が遅い、または失敗した原因を特定できます。

このメニューには、OpenTelemetry の シグナルタイプ に対応する次のセクションがあります。

- トレース
- メトリクス
- ログ

Zabbix のアイテムによって収集された値を表示する 監視 > 最新データ とは異なり、*APM* セクションにはデータソースから直接テレメトリデータが表示されるため、アイテムを作成する必要はありません。*APM* セクションにデータを表示するには、次の手順を実行します。

1. 管理 > データソース でテレメトリデータソースを指定します。
2. APM を有効にした Zabbix プロキシ、または OpenTelemetry Collector によって、テレメトリデータがデータソースに書き込まれていることを確認します。Zabbix サーバーはテレメトリデータを受信できず、Zabbix プロキシのみが受信できることに注意してください。

トリガーまたはグラフでテレメトリデータを使用するには、テレメトリクエリ アイテムを使用します。

トレース

概要

APM > トレースセクションには、アプリケーションおよびサービスから収集されたトレースが表示されます。

トレースは、アプリケーションおよびサービスを通過するリクエストの経路を表します。たとえば、オンラインショップでユーザーが 注文するボタンをクリックすると、リクエストは Web インターフェースサービスからチェックアウトサービス、決済サービスを経由して、データベースに到達します。

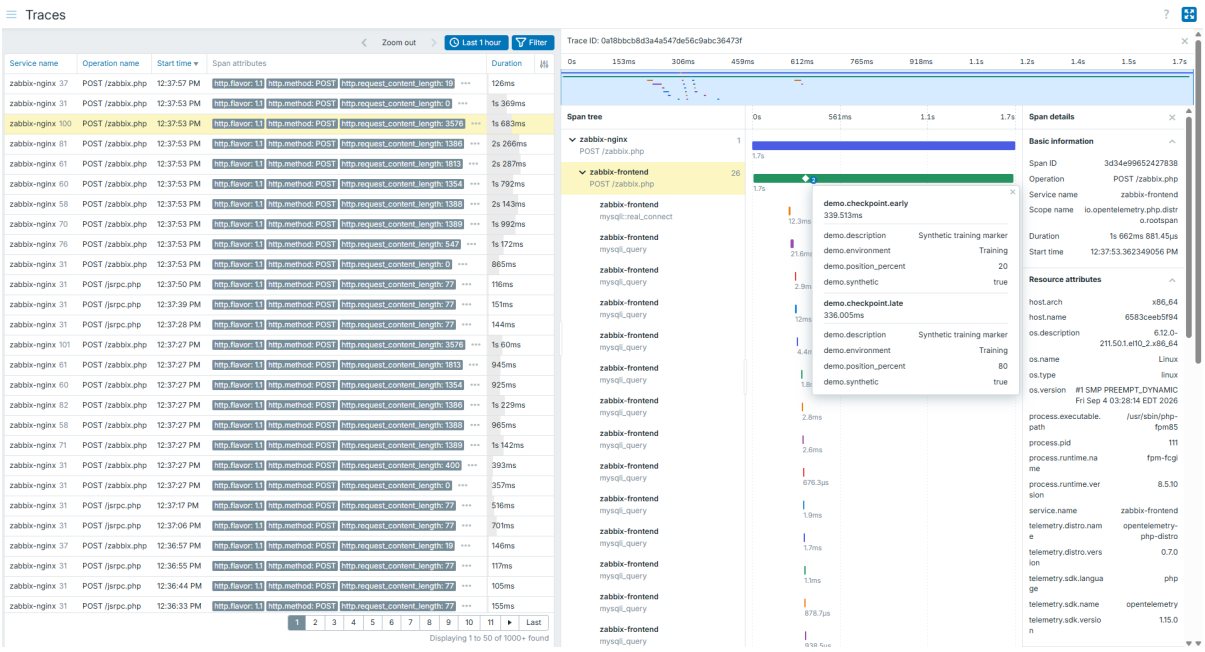
トレースはスパンで構成されます。各スパンは、Web インターフェースサービスからチェックアウトサービスに送信される HTTP リクエスト、注文を保存するデータベースクエリ、配送料を計算する関数呼び出しなど、単一の操作を表します。

スパンは階層を形成します。ルートスパンは、リクエストの最初の操作を表します。その他のすべてのスパンは、その操作が開始されたスパンの子スパンです。たとえば、注文するリクエストを受信する Web インターフェースサービスがルートスパンであり、注文を処理するチェックアウトサービスがその子スパンです。

各スパンは、サービス（例: `zabbix-frontend`; OpenTelemetry の `service.name` リソース属性によって名前が付けられます）と、スパンを作成したコードの部分であるインストルメンテーションスコープ（例: HTTP リクエストをトレースするライブラリ）によって生成されます。

各操作に関するコンテキストを提供するため、スパンにはキーと値の属性もあります。スパン属性は操作自体（例: `http.method: POST`）を説明し、リソース属性はサービスとその実行環境（例: `host.name: eabf8c2a5ddc`）を説明します。

トレースの詳細については、OpenTelemetry のドキュメントを参照してください。

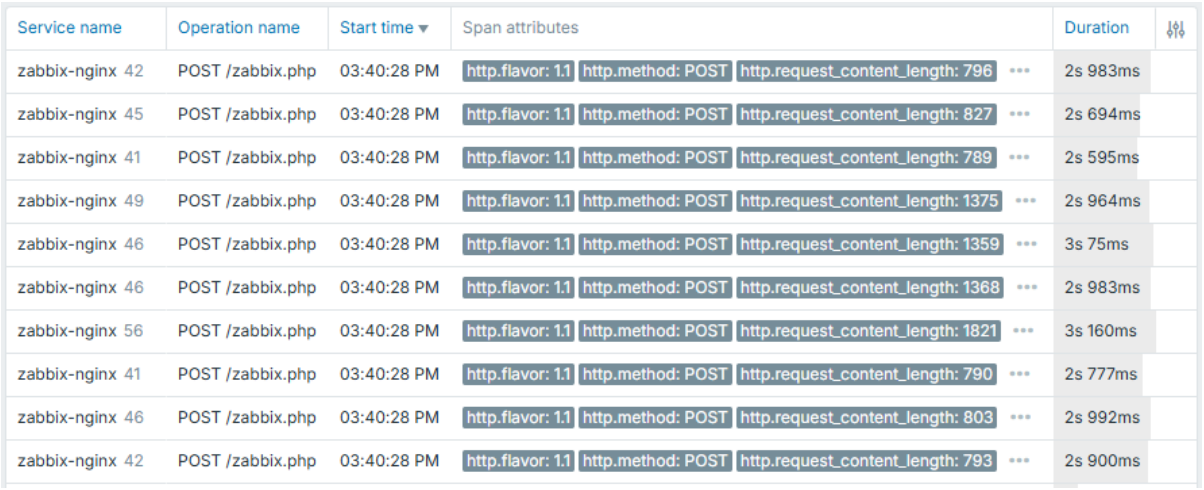


APM > トレースセクションには、次の要素が含まれます。

- キオスクモードボタン - メニューとヘッダーを表示せず、ページのコンテンツのみを表示します。
- 期間セレクター - トレースを表示する期間を選択します。
- フィルター - 関心のあるトレースのみを表示します。
- トレース一覧 - ルートスパンによって説明されたトレースを表示します。
- トレースパネル - 選択したトレースのすべてのスパンをタイムライン上に表示します。

トレース一覧

トレース一覧の各行は 1 つのトレースを表し、各列にはそのルートスパンのデータが表示されます。



Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796 •••	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827 •••	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789 •••	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375 •••	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359 •••	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368 •••	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821 •••	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790 •••	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803 •••	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793 •••	2s 900ms

トレースはカスタマイズ可能なテーブルに表示されます。テーブルヘッダーの テーブルをカスタマイズボタンを使用すると、列の表示、非表示、並べ替え、およびレイアウトのデフォルトへのリセットができます。各列の右端をドラッグして列の幅を変更することもできます。右端をダブルクリックすると、幅がリセットされます。すべてのカスタマイズは直ちに適用され、現在のユーザーに対して保存されます。

トレース一覧には、次の列を表示できます。

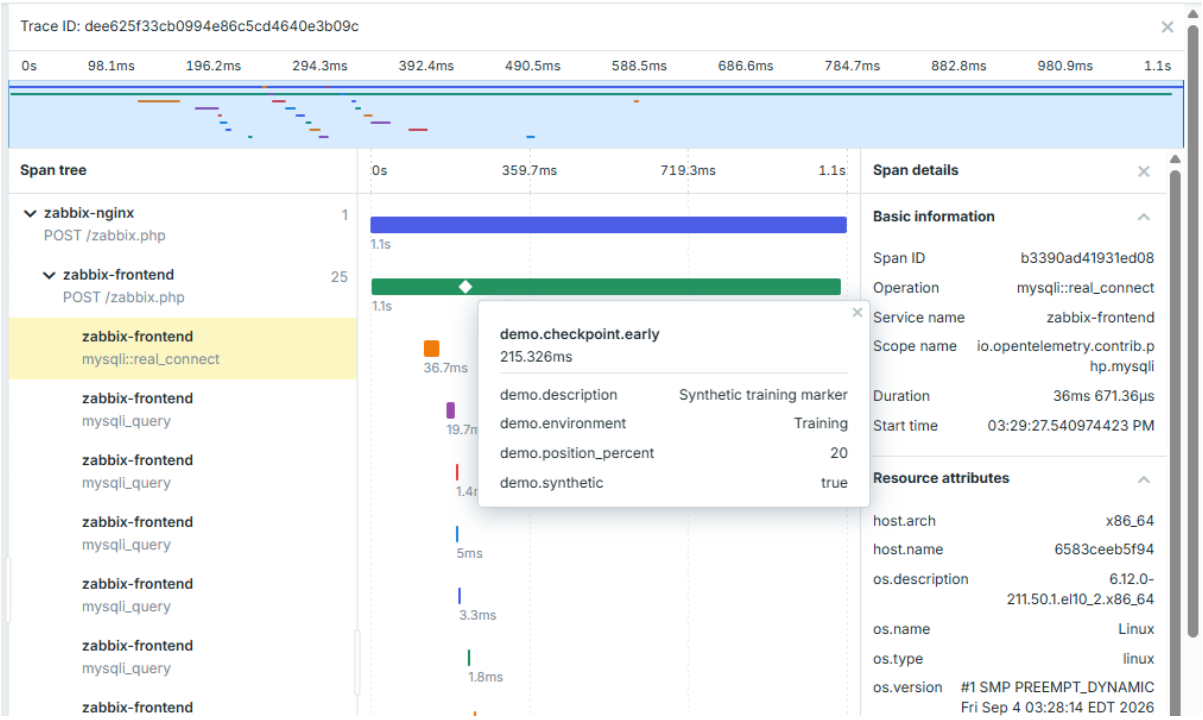
Note:
各列は、テレメトリデータソースの `otel_traces` テーブルの列に対応しています。列の一覧については、サポートされているテレメトリデータ列を参照してください。ほとんどの列はデータソースと同じ名前です（例: トレース *ID* は `TraceId` 列に対応します）。名前が異なる場合は、下の表にデータソースの列名を記載しています。

列	説明
トレース *ID*	トレースに属するすべてのスパンで共有される、トレースの ID。
スパン *ID*	ルートスパンの ID。
トレース状態	トレースツールがトレースに付加する追加データ（存在する場合）（W3C Trace Contextを参照）。
サービス名	ルートスパンを生成したサービスと、トレース内のスパン数。 *Error* ステータスコード（下記参照）のスパンがある場合は、赤いアイコンにその数が表示されます。
操作名	ルートスパンが表す操作（例: `POST /zabbix.php`）。 データソースの列: `SpanName`。
スパン種別	ルートスパンの役割: **Unspecified** - 指定なし。 **Internal** - アプリケーション内で実行された操作。 **Server** - 受信リクエストの処理。 **Client** - 送信リクエスト。 **Producer** - 非同期操作の作成（例: メッセージの送信）。 **Consumer** - 非同期操作の処理。
スコープ名	ルートスパンを生成したインストルメンテーションスコープ。
スコープバージョン	インストルメンテーションスコープのバージョン。
ステータスコード	ルートスパンが表す操作のステータス: **Unset** - ルートスパンに明示的なステータスが割り当てられていません。 **Ok** - 操作が正常に完了しました。 **Error** - 操作がエラーで完了しました。
ステータスメッセージ	ステータスに関する追加情報（存在する場合）。
開始時刻	ルートスパンが開始した日時。 データソースの列: `Timestamp`。

列	説明
スパン属性	ルートスパンのスパン属性。 *** 3 点ボタンをクリックすると、すべての属性が表示されます。
リソース属性	ルートスパンのリソース属性。 *** 3 点ボタンをクリックすると、すべての属性が表示されます。
期間	トレースの期間。灰色のバーは、他のトレースと比較した期間を示します。

トレースパネル

トレースリストでトレースをクリックすると、トレースパネルが開き、トレースのすべてのスパンがタイムラインに表示されます。実行された操作、その順序、および各操作にかかった時間を確認できます。

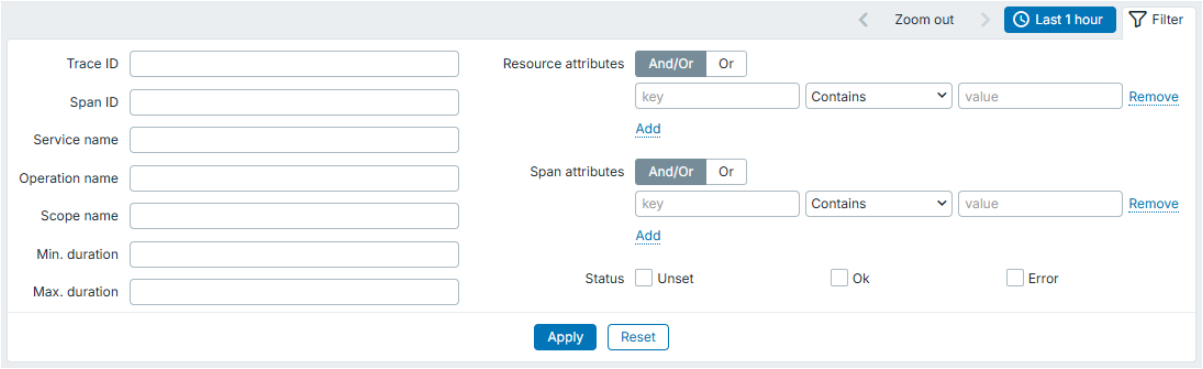


トレースパネルには、次の要素が表示されます。

要素	説明
トレース *ID*	選択したトレースの ID。
トレースタイムラインの概要	最大 8 行に表示されたスパンを含む、トレース全体（複数のスパンが 1 行を共有する場合があります）。 選択範囲の境界または範囲自体をドラッグすると、トレースのその部分にスパンタイムラインをズームできます。
スパンツリー	階層構造で表示されたスパン。各スパンには、サービス（太字）、操作名、および直接の子スパン数が表示されます。
スパンタイムライン	バーとして表示されたスパン。位置と長さによって、各スパンの開始時刻と所要時間が示されます。 マーカーはイベント（例: 例外）を示します。詳細を確認するには、マーカーにカーソルを合わせるか、クリックします。
スパンの詳細	スパンツリーでスパンをクリックすると、その詳細と属性が表示されます。

フィルターの使用

フィルターを使用すると、関心のあるトレースのみを表示できます。



属性でフィルターするには、属性キーを入力し、演算子を選択して、属性値を入力します。

次の演算子がサポートされています。

- **Contains** - *(デフォルト)* 属性値に入力した文字列が含まれます。
- **Equals** - 属性値が入力した文字列と一致します。
- **Does not contain** - 属性値に入力した文字列が含まれません。
- **Does not equal** - 属性値が入力した文字列と一致しません。
- **Exists** - 入力したキーを持つ属性が存在します。
- **Does not exist** - 入力したキーを持つ属性が存在しません。

複数の条件を追加する場合は、条件の評価方法を選択します。

- **And/Or** - *(デフォルト)* すべての条件を満たす必要があります。同じ属性キーを持つ条件は *Or* で結合されます。
- **Or** - 少なくとも 1 つの条件を満たす必要があります。

メトリクス

概要

APM > メトリクスセクションには、アプリケーションおよびサービスから収集されたメトリクスのデータポイントが表示されます。

メトリクスとは、測定値（リクエスト時間、処理された注文数、メモリ使用量など）です。アプリケーションは、特定の時点で測定された値（または集計値）を含む データポイントとして値を送信します。メトリクスは Zabbix のアイテムに、データポイントはそのアイテムの値に似ています。

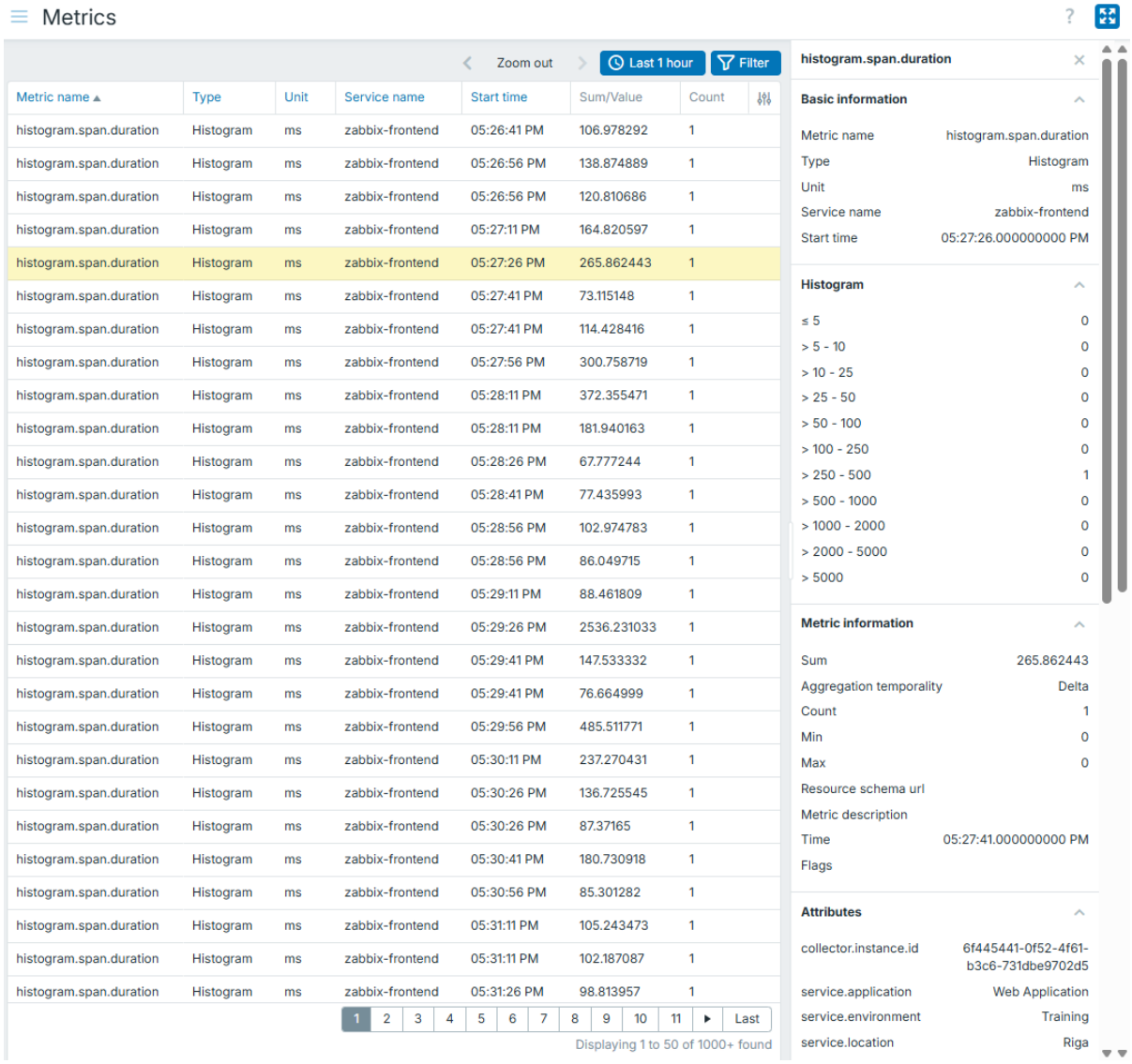
データポイントには、次のいずれかのタイプがあります。

- **Gauge** - 特定の時点で測定された値（現在のメモリ使用量など）。
- **Sum** - 一定期間にわたる測定値の合計（リクエストの総数など）。
- **Histogram** - バケットと呼ばれる複数の固定された値の範囲ごとに、一定期間内にその範囲に該当した測定値の数（0～100 ミリ秒、100～250 ミリ秒、250～500 ミリ秒かかったリクエストの数など）。
- **Exponential histogram** - ヒストグラムと同様ですが、各バケットの幅が同じ倍率で前のバケットより広くなります（1～2 ミリ秒、2～4 ミリ秒、4～8 ミリ秒など）。

各データポイントは、サービス（`zabbix-frontend` など。OpenTelemetry の `service.name` リソース属性によって名前が付けられます）と、データポイントを作成したコードの一部である インストルメンテーションスコープ（HTTP リクエスト時間を測定するライブラリなど）によって生成されます。

各測定値のコンテキストを提供するため、データポイントにはキーと値の 属性も含まれます。データポイント属性 は測定対象を示し、同じメトリクスのデータポイントを区別します（`http.request.method: POST` など）。リソース属性 は、サービスとその実行環境を示します（`host.name: eabf8c2a5ddc` など）。

メトリクスの詳細については、OpenTelemetry のドキュメントを参照してください。



APM > メトリクスセクションには、次の要素が含まれます。

- キオスクモード ボタン - メニューとヘッダーを表示せず、ページのコンテンツのみを表示します。
- 期間セレクター - データポイントを表示する期間を選択します。
- フィルター - 関心のあるデータポイントのみを表示します。
- メトリクス一覧 - データポイントを表示します。
- メトリクスの詳細 - 選択したデータポイントの詳細を表示します。

メトリック一覧

メトリック一覧の各行は、1 つのデータポイントを表します。

Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

データポイントは、カスタマイズ可能なテーブルに表示されます。テーブルヘッダーの テーブルをカスタマイズボタンを使用すると、列の表示、非表示、並べ替え、およびレイアウトのデフォルト設定へのリセットを行えます。各列の右端をドラッグして列の幅を変更することもできます。右端をダブルクリックすると、幅がリセットされます。すべてのカスタマイズは直ちに適用され、現在のユーザーに対して保存されます。

メトリック一覧には、次の列を表示できます。

Note:
タイプを除く各列は、テレメトリデータソースのメトリックテーブル（otel_metrics_gauge、otel_metrics_sum、otel_metrics_histogram、otel_metrics_exponential_histogram）の列に対応しています。列の一覧については、サポートされているテレメトリデータ列を参照してください。ほとんどの列は、データソース内と同じ名前です（例: メトリック名は MetricName 列に対応します）。名前が異なる場合は、下の表にデータソースの列名を記載しています。

列	説明
メトリック名	データポイントが属するメトリック（例: http.server.duration）。
タイプ	データポイントのタイプ: ゲージ、合計、ヒストグラム、または 指数ヒストグラム。 データポイントを保持するメトリックテーブルによって決まります。
単位	メトリック値の単位（存在する場合）（例: s または By。UCUMを参照）。 データソースの列: MetricUnit。
サービス名	データポイントを生成したサービス。
スコープ名	データポイントを生成したインストルメンテーションスコープ。
開始時刻	データポイントの測定期間が開始した日時。 データソースの列: StartTimeUnix。
合計/値	すべての測定値の合計（ヒストグラム、指数ヒストグラム）、またはデータポイントの値（ゲージ、合計）。
カウント	ヒストグラムまたは 指数ヒストグラムデータポイントの測定値の数。
時刻	データポイントの日時。合計、ヒストグラム、および 指数ヒストグラムデータポイントの場合は、測定期間の終了時刻。 データソースの列: TimeUnix。
フラグ	データポイントのフラグ: **0** - フラグは設定されていません。 **1** - データポイントの値は記録されていません。
集計の一時性	合計、ヒストグラム、または 指数ヒストグラムデータポイントの値を時間の経過に伴って集計する方法: 未指定 - 指定されていません。 デルタ - 前のデータポイント以降の期間で集計されます。 累積 - 開始時刻以降の期間で集計されます。
メトリックの説明	メトリックの説明（存在する場合）。
属性	データポイントの属性。 3 点ボタンをクリックすると、すべての属性が表示されます。
リソース属性	データポイントのリソース属性。 3 点ボタンをクリックすると、すべての属性が表示されます。

列	説明
スコープ属性	インストルメンテーションスコープの属性。 *** 3 点ボタンをクリックすると、すべての属性が表示されます。

メトリックの詳細

メトリック一覧でデータポイントをクリックすると、データポイントに関するすべての情報を表示するメトリックの詳細パネルが開きます。

exponential.span.calls

Basic information

Metric name	exponential.span.calls
Type	Summary
Unit	
Service name	zabbix-frontend
Start time	02:22:15.000000000 PM

Metric information

Value	1
Aggregation temporality	Delta
Is monotonic	true
Resource schema url	
Metric description	
Time	02:22:30.000000000 PM
Flags	

Attributes

Resource attributes

Scope attributes

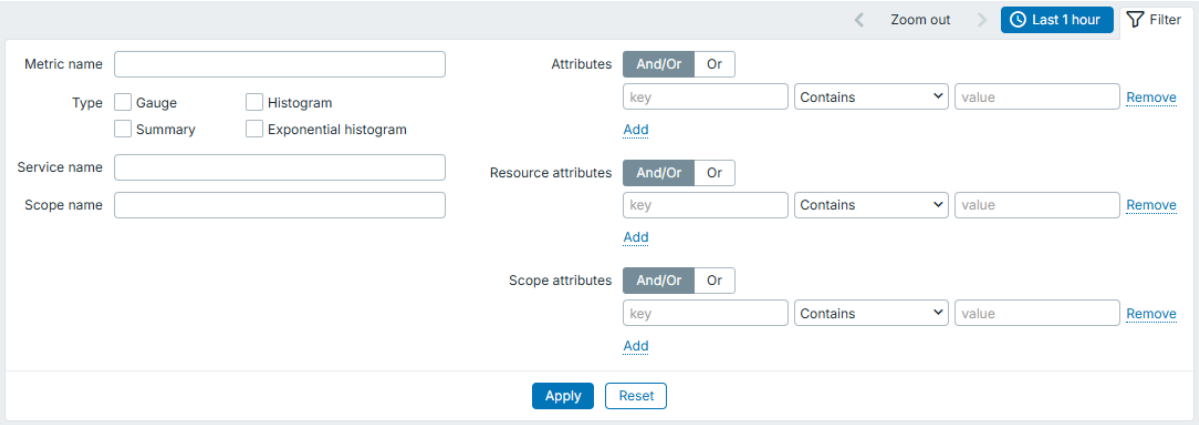
Scope

メトリックの詳細パネルには、次のセクションが表示されます。

セクション	説明
基本情報	メトリック名、タイプ、単位、サービス名、開始時刻。
ヒストグラム	バケット（例: > 5 – 10）と、それぞれに含まれる測定値の数。 ヒストグラムデータポイントの場合のみ表示されます。
指数ヒストグラム	バケット（例: >= 249.7885 – 249.7887）と、それぞれに含まれる測定値の数。 指数ヒストグラムデータポイントの場合のみ表示されます。
メトリック情報	データポイントのタイプに応じた値、およびメトリックの説明、時刻、フラグ、リソーススキーマ URL。
属性	データポイントの属性。
リソース属性	リソース属性。
スコープ属性	インストルメンテーションスコープの属性（存在する場合）。
スコープ	インストルメンテーションスコープの名前、バージョン、スキーマ URL。

フィルターの使用

フィルターを使用すると、関心のあるデータポイントのみを表示できます。



属性でフィルタリングするには、属性キーを入力し、演算子を選択して、属性値を入力します。

次の演算子がサポートされています。

- **Contains** - *(デフォルト)* 属性値に入力した文字列が含まれます。
- **Equals** - 属性値が入力した文字列と一致します。
- **Does not contain** - 属性値に入力した文字列が含まれません。
- **Does not equal** - 属性値が入力した文字列と一致しません。
- **Exists** - 入力したキーを持つ属性が存在します。
- **Does not exist** - 入力したキーを持つ属性が存在しません。

複数の条件を追加する場合は、条件の評価方法を選択します。

- **And/Or** - *(デフォルト)* すべての条件を満たす必要があります。同じ属性キーを持つ条件は *Or* で結合されます。
- **Or** - 少なくとも 1 つの条件を満たす必要があります。

ログ

概要

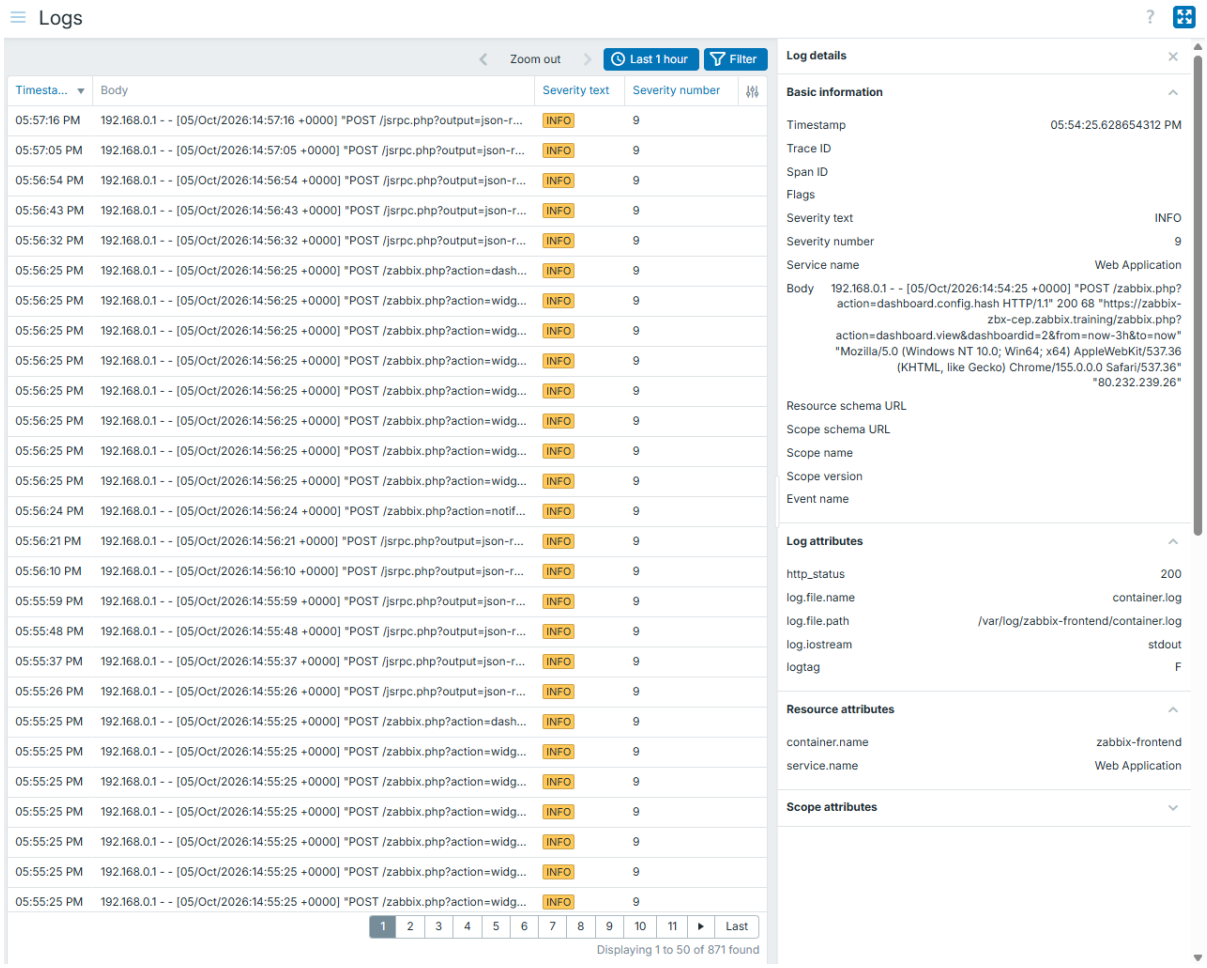
APM > ログセクションには、アプリケーションおよびサービスから収集されたログレコードが表示されます。

ログレコードは、イベント（支払いの失敗やデータベースのタイムアウトなど）を記録したテキストです。各ログレコードには、重大度とメッセージ（本文）が含まれます。ログレコードがトレースの一部である操作中に生成された場合、そのトレースおよびスパンの ID も含まれます。

各ログレコードは、サービス（`zabbix-frontend` など。OpenTelemetry の `service.name` リソース属性によって名前が付けられます）と、ログレコードを作成したコードの部分であるインストルメンテーションスコープ（ロギングライブラリなど）によって生成されます。

各イベントのコンテキストを提供するため、ログレコードにはキーと値の属性も含まれます。ログ属性はイベント自体（`log.file.path: /var/log/zabbix/zabbix_frontend.log` など）を説明し、リソース属性はサービスと、そのサービスが実行される環境（`host.name: eabf8c2a5ddc` など）を説明します。

ログの詳細については、OpenTelemetry のドキュメントを参照してください。



APM > ログセクションには、次の要素が含まれます。

- キオスクモードボタン - メニューとヘッダーを表示せず、ページのコンテンツのみを表示します。
- 期間セレクター - ログレコードを表示する期間を選択します。
- フィルター - 関心のあるログレコードのみを表示します。
- ログ一覧 - ログレコードを表示します。
- ログの詳細 - 選択したログレコードの詳細を表示します。

ログリスト

ログリストの各行は、1 つのログレコードを表します。

Timestamp	Body	Severity text	Severity number
06:04:03 PM	192.168.0.1 - - [05/Oct/2026:15:04:03 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:52 PM	192.168.0.1 - - [05/Oct/2026:15:03:52 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:41 PM	192.168.0.1 - - [05/Oct/2026:15:03:41 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:30 PM	192.168.0.1 - - [05/Oct/2026:15:03:30 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=dashboard.config.hash HTTP...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.problems.view HTTP/...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=notifications.get&output=ajax...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:19 PM	192.168.0.1 - - [05/Oct/2026:15:03:19 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9

ログレコードは、カスタマイズ可能なテーブルに表示されます。テーブルヘッダーの テーブルをカスタマイズボタンを使用する

と、列の表示、非表示、並べ替え、およびレイアウトのデフォルトへのリセットを行えます。各列の右端をドラッグして列の幅を変更することもできます。境界線をダブルクリックすると、幅がリセットされます。すべてのカスタマイズはすぐに適用され、現在のユーザー用に記憶されます。

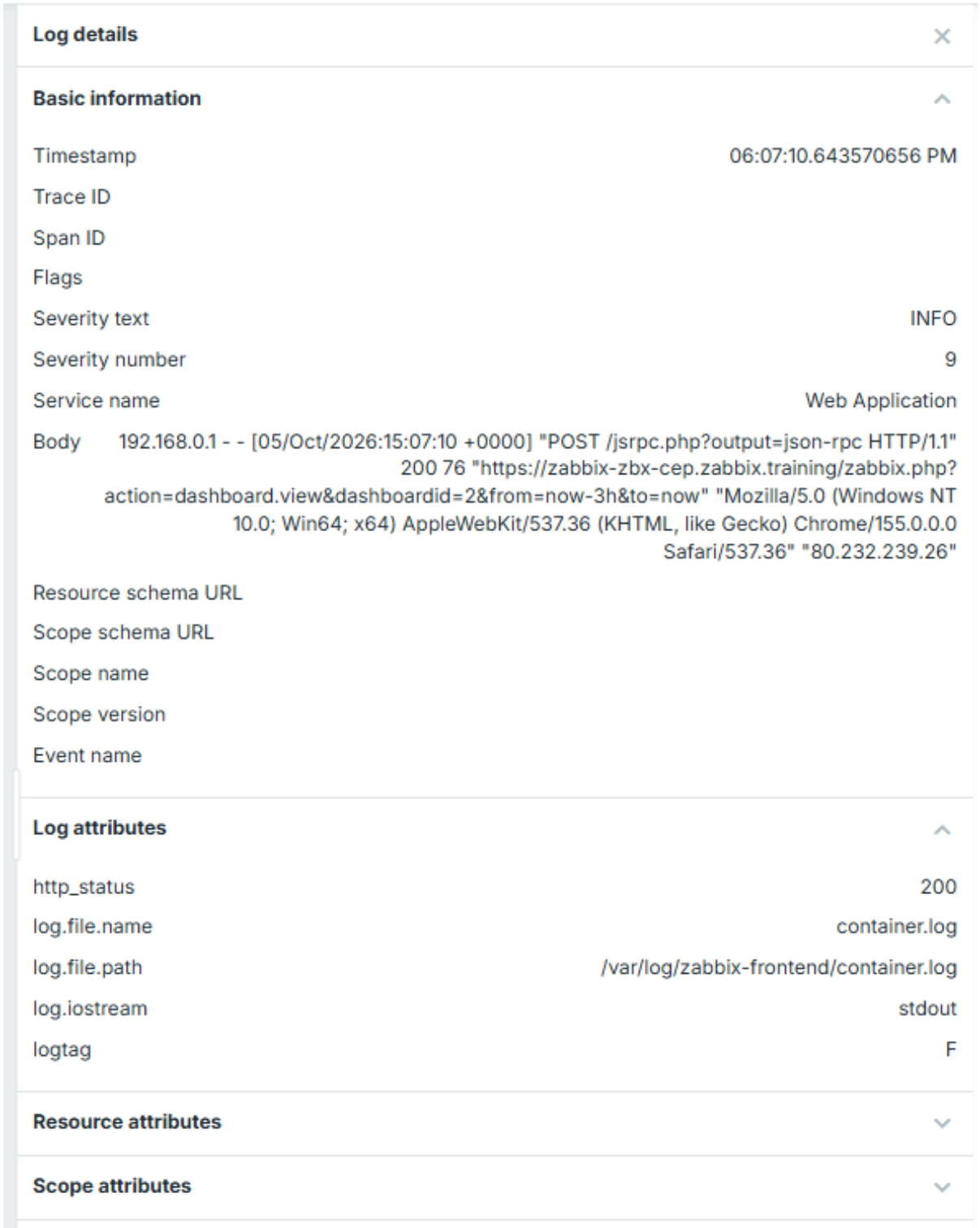
ログリストには、次の列を表示できます。

> **Note:**
> 各列は、テレメトリデータソースの `otel_logs` テーブル内の列に対応します。列の一覧については、サポートされているテレメトリデータ列を参照してください。ほとんどの列は、データソース内と同じ名前です（例: *Severity text* は `SeverityText` 列に対応します）。名前が異なる場合は、以下の表にデータソースの列名を記載しています。

列	説明
Timestamp	イベントが発生した日時。
Body	ログレコードのメッセージ。
Severity text	アプリケーションによって提供された重大度レベル（例: `INFO`、`WARN`、`ERROR`）。 ラベルの色は重大度によって異なります。
Severity number	重大度レベルを数値で表したもの: **0** - UNSPECIFIED。 **1-4** - TRACE。 **5-8** - DEBUG。 **9-12** - INFO。 **13-16** - WARN。 **17-20** - ERROR。 **21-24** - FATAL。
Trace ID	ログレコードが生成されたトレースの ID（存在する場合）。
Span ID	ログレコードが生成されたスパンの ID（存在する場合）。
Flags	ログレコードのトレースフラグ: **0** - トレースはサンプリングされていません。 **1** - トレースはサンプリングされています（記録済み）。 データソースの列: `TraceFlags`。
Service name	ログレコードを生成したサービス。
Scope name	ログレコードを生成したインストルメンテーションスコープ。
Scope version	インストルメンテーションスコープのバージョン。
Event name	ログレコードが表すイベントタイプの名前（存在する場合）。
Log attributes	ログレコードのログ属性。 ⋯ 3 点ボタンをクリックすると、すべての属性が表示されます。
Resource attributes	ログレコードのリソース属性。 ⋯ 3 点ボタンをクリックすると、すべての属性が表示されます。
Scope attributes	インストルメンテーションスコープの属性。 ⋯ 3 点ボタンをクリックすると、すべての属性が表示されます。

ログの詳細

ログ一覧でログレコードをクリックすると、ログの詳細パネルが開き、ログレコードに関するすべての情報が表示されます。

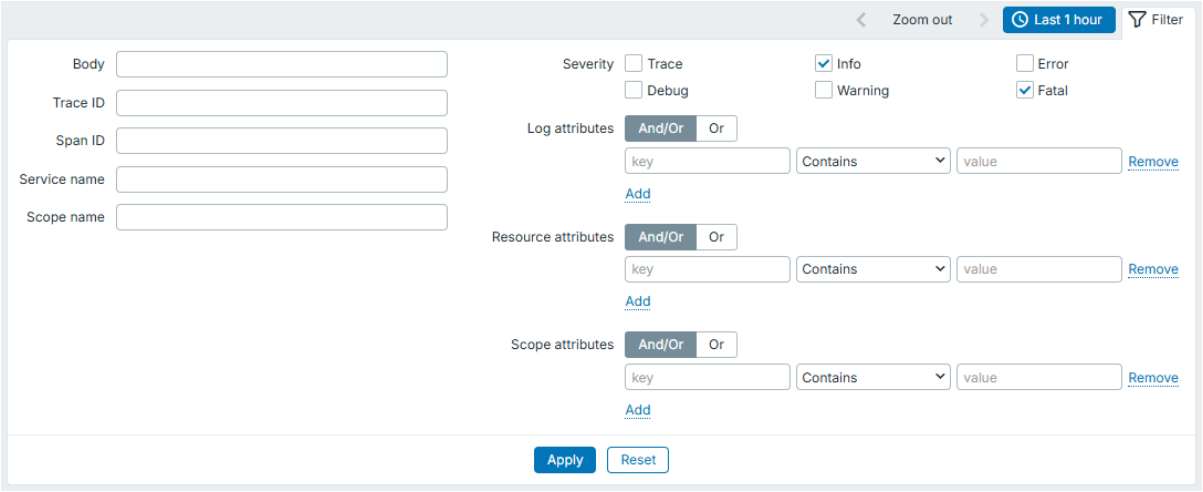


ログの詳細パネルには、次のセクションがあります。

セクション	説明
基本情報	ログレコードのトレース ID、スパン ID、トレースフラグ。
ログ属性	ログレコードのログ属性。
リソース属性	スキーマ URL とリソース属性。
スコープ属性	インストルメンテーションスコープのスキーマ URL、名前、バージョン、属性。

フィルターの使用

フィルターを使用すると、関心のあるログレコードのみを表示できます。



属性でフィルターするには、属性キーを入力し、演算子を選択して、属性値を入力します。

次の演算子がサポートされています。

- 含む - *(デフォルト)* 属性値に入力した文字列が含まれます。
- 等しい - 属性値が入力した文字列と等しくなります。
- 含まない - 属性値に入力した文字列が含まれません。
- 等しくない - 属性値が入力した文字列と等しくなりません。
- 存在する - 入力したキーを持つ属性が存在します。
- 存在しない - 入力したキーを持つ属性が存在しません。

複数の条件を追加する場合は、条件の評価方法を選択します。

- **And/Or** - *(デフォルト)* すべての条件を満たす必要があります。同じ属性キーを持つ条件は *Or* で結合されます。
- **Or** - 少なくとも 1 つの条件を満たす必要があります。

サービス

概要

サービスメニューは、Zabbix のサービスモニタリング機能です。

サービス

概要

このセクションでは、インフラストラクチャに基づいて Zabbix で設定されたサービス全体の概要ステータスを確認できます。

サービスは、「子」サービスと呼ばれる複数の他のサービスから構成される、複数レベルの階層になる場合があります。子サービスは、サービス全体のステータスに影響します（サービス監視機能の概要も参照してください）。

サービスの状態の主なカテゴリは正常または障害です。障害状態は、対応する障害の深刻度名と色で表されます。

表示モードでは、サービスのステータスやその他の詳細を確認できます。また、このセクションで編集モードに切り替えることにより、サービス階層（サービスや子サービスの追加・編集）を設定することもできます。

表示モードと編集モードを切り替えるには、右上隅にある該当するボタンをクリックします。

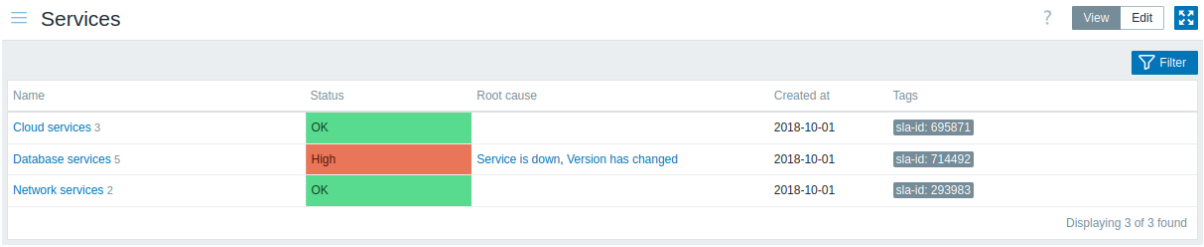
- 

- サービスを表示
- 

- サービスおよび子サービスを追加・編集

編集へのアクセスは、ユーザーの役割の設定によって異なることに注意してください。

サービスの表示



既存のサービスの一覧が表示されます。

表示されるデータ:

Parameter	Description
Name	サービス名。 サービス名はサービス詳細へのリンクです。 名前の後の数値は、そのサービスにいくつの子サービスがあるかを示します。
Status	サービスの状態: **OK** - 問題なし **<problem color and severity>** - 障害とその深刻度を示します。複数の障害がある場合は、最も重大な障害の色と深刻度が表示されます。
Root cause	サービスの状態に直接または間接的に影響する根本原因の障害が一覧表示されます。 同じ障害は {SERVICE.ROOTCAUSE} macro が返す内容として一覧表示されます。 障害名をクリックすると、*Monitoring* > *Problems* でその詳細を確認できます。 サービスの状態に影響しない障害は一覧に含まれません。
Created at	サービスが作成された時刻が表示されます。
Tags	サービスのタグが表示されます。タグは、サービスのアクションおよびSLAsでサービスを識別するために使用されます。

ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoring ページで説明しています。

フィルターの使用

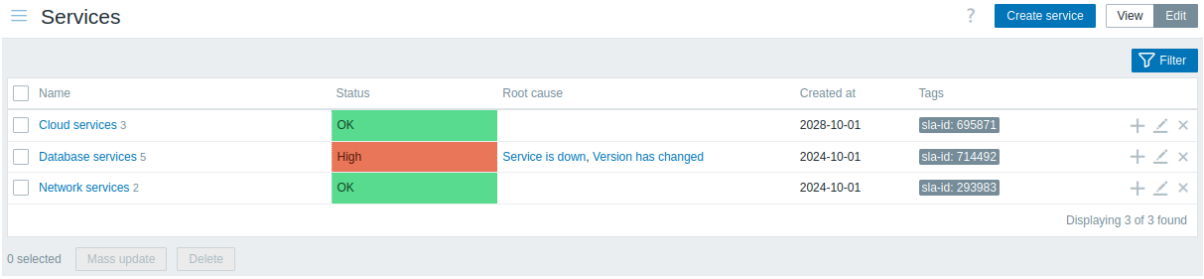
フィルターを使用して、関心のあるサービスのみを表示できます。

パラメータ	説明
名前	サービス名でフィルターします。
状態	サービスの状態でフィルターします。
子サービスのないサービスのみ	チェックボックスをオンにすると、子サービスのないサービスのみが表示されます。このパラメータはサービスの編集モードでのみ利用可能です。
問題タグのないサービスのみ	チェックボックスをオンにすると、問題タグのないサービスのみが表示されます。このパラメータはサービスの編集モードでのみ利用可能です。
タグ	サービスのタグ名と値、またはサービスの問題タグ名と値でフィルターします（サービスの編集モードの場合）。 特定のタグやタグ値を含めたり除外したりすることができます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件にはいくつかの演算子があります： 存在する - 指定したタグ名を含めます; 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）; 含む - タグ値が入力した文字列を含むタグ名を含めます（部分一致、大文字と小文字を区別しない）; 存在しない - 指定したタグ名を除外します; 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）; 含まない - タグ値が入力した文字列を含むタグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります： **And/Or** - すべての条件を満たす必要があり、同じタグ名の条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

サービスの編集

編集ボタンをクリックすると、編集モードに入ります。編集モードでは、一覧の各項目の前にチェックボックスが追加され、さらに次のオプションが表示されます。

-  - このサービスに子サービスを追加する
-  - このサービスを編集する
-  - このサービスを削除する

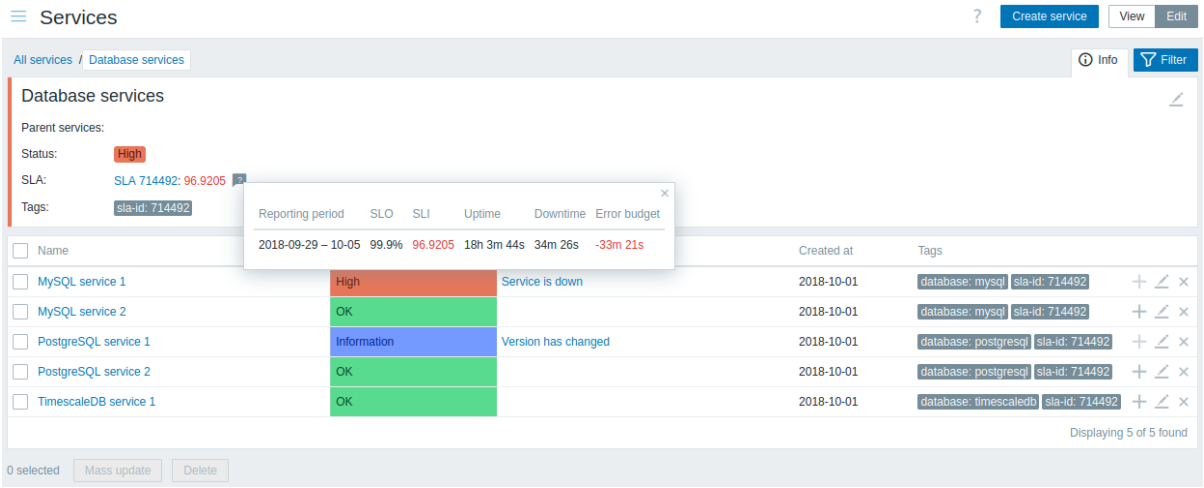


新しいサービスを設定するには、右上隅の サービスの作成ボタンをクリックします。

サービス詳細

サービス詳細にアクセスするには、サービス名をクリックします。すべてのサービスの一覧に戻るには、*All services* をクリックします。

サービス詳細には、情報ボックスと子サービスの一覧が含まれます。



情報ボックスにアクセスするには、*Info* タブをクリックします。情報ボックスには、次の項目が含まれます。

- 親サービスの名前（存在する場合）
- このサービスの現在のステータス
- このサービスの現在の SLA。形式は `SLA name:service level indicator` です。'SLA name' は、このサービスの SLA レポートへのリンクでもあります。サービスレベルインジケーター（SLI）の横にある情報ボックス上にマウスを置くと、SLI の詳細を含むポップアップの情報一覧が表示されます。サービスレベルインジケーターは、現在のサービスレベルをパーセンテージで表示します。
- サービスタグ

情報ボックスには、サービス設定 へのリンクも含まれます。

子サービスのフィルターを使用するには、*Filter* タブをクリックします。

編集モードでは、子サービスの一覧に追加の編集オプションが表示されます。

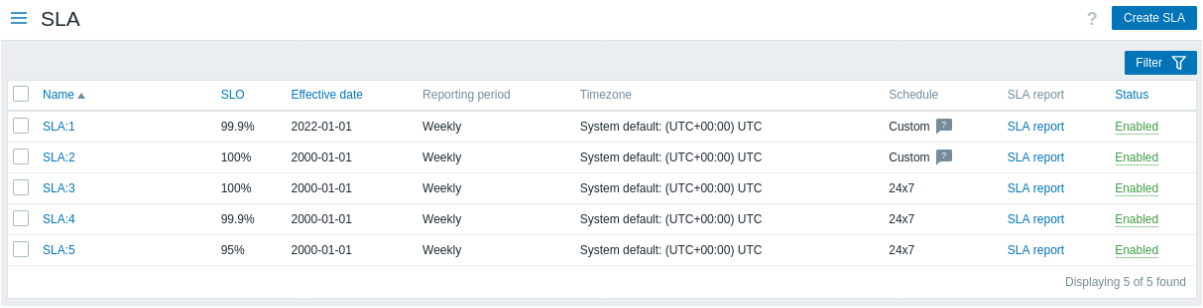
-  - このサービスに子サービスを追加する
-  - このサービスを編集する
-  - このサービスを削除する

SLA

概要

このセクションでは、SLA の表示と設定を行えます。

SLA



設定済みの SLA の一覧が表示されます。注ユーザーがアクセスできるサービスに関連する SLA のみが表示されます（ユーザーのロールで *Manage SLA* が有効になっていない限り、読み取り専用です）。

表示されるデータ:

Parameter	Description
Name	SLA 名が表示されます。 名前はSLA configuration へのリンクです。
SLO	サービスレベル目標（SLO）が表示されます。
Effective date	SLA 計算の開始日が表示されます。
Reporting period	SLA レポートで使用される期間が表示されます - *daily*、*weekly*、*monthly*、*quarterly*、または *annually*。
Time zone	SLA のタイムゾーンが表示されます。
Schedule	SLA のスケジュールが表示されます - 24x7 またはカスタム。
SLA report	リンクをクリックすると、この SLA の SLA レポートを表示できます。
Status	SLA のステータスが表示されます - 有効または無効。

SLA レポート

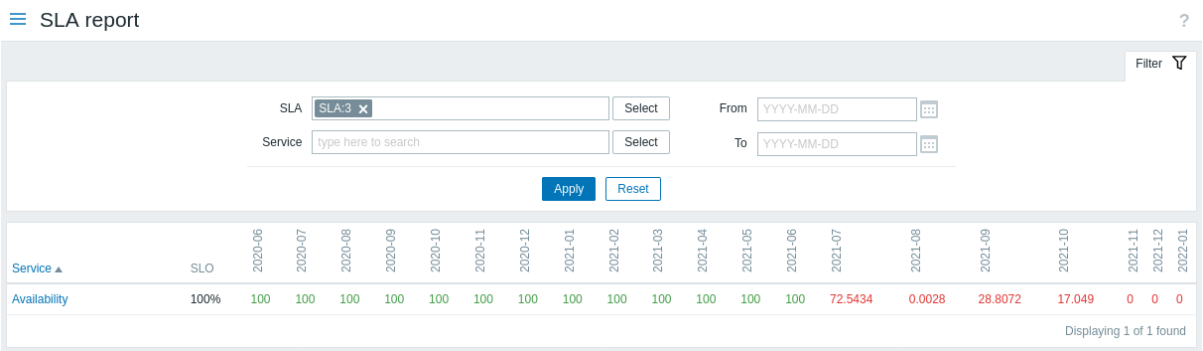
概要

このセクションでは、フィルターで選択した条件に基づいてSLA レポートを表示できます。

SLA レポートはダッシュボードウィジェットとして表示することもできます。

レポート

フィルターを使用すると、SLA 名およびサービス名に基づいてレポートを選択できます。また、表示する期間を制限することもできます。

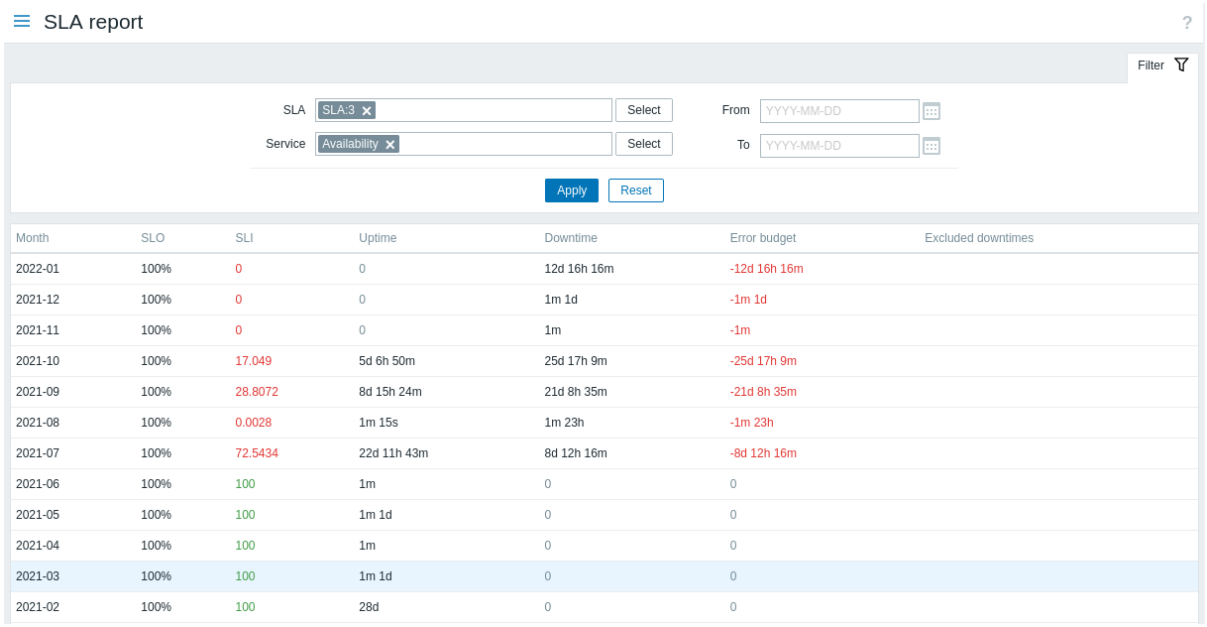


各列（期間）には、その期間の SLI が表示されます。設定された SLO に違反している SLI は赤色で強調表示されます。

レポートには 20 期間が表示されます。開始日と終了日の両方を指定した場合は、最大 100 期間を表示できます。

レポートの詳細

レポート内のサービス名をクリックすると、より詳細な表示を行う別のレポートにアクセスできます。



負の問題継続時間 は、SLA の計算やレポート作成には影響しないことに注意してください。

インベントリ

概要

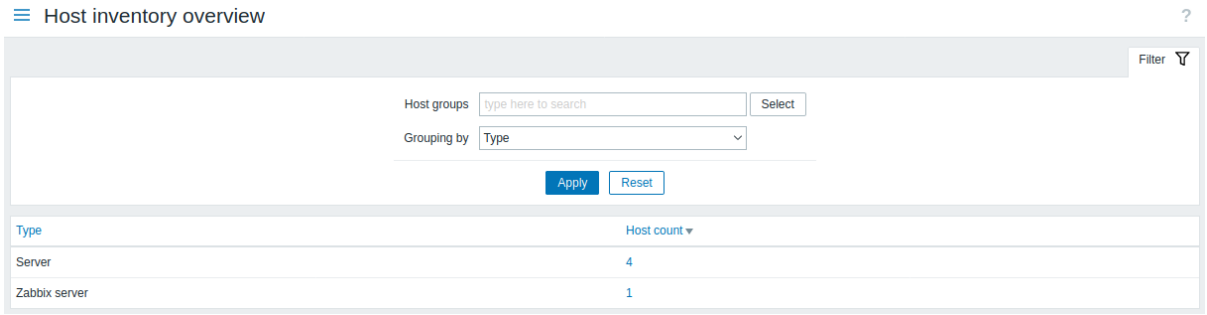
インベントリメニューには、選択したパラメータごとのホストインベントリデータの概要と、ホストインベントリの詳細を表示するセクションがあります。

概要

概要

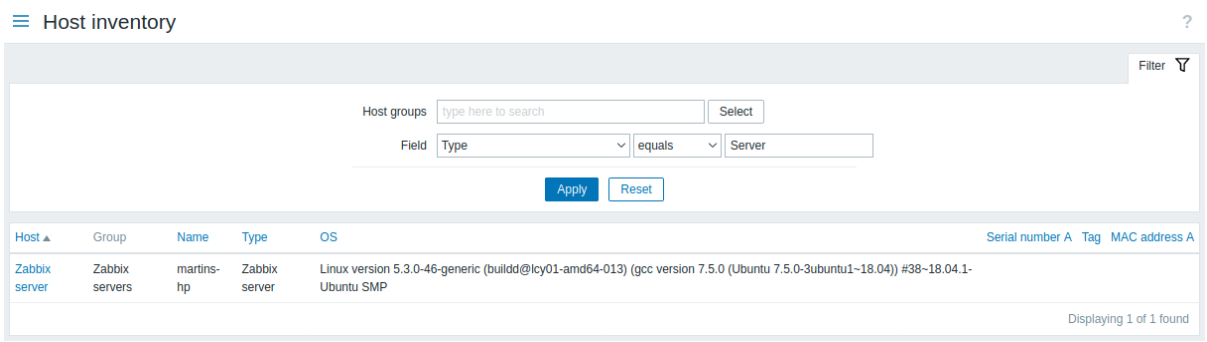
インベントリ > 概要セクションでは、ホストインベントリ データの概要を表示する方法を提供します。

概要を表示するには、ホストグループ（または指定なし）と、データを表示するためのインベントリフィールドを選択します。選択したフィールドの各項目に対応するホスト数が表示されます。



概要の完全性は、ホストにどれだけのインベントリ情報が保持されているかによって決まります。

ホスト数列の数値はリンクになっており、クリックすると ホストインベントリテーブルで該当ホストが絞り込まれます。

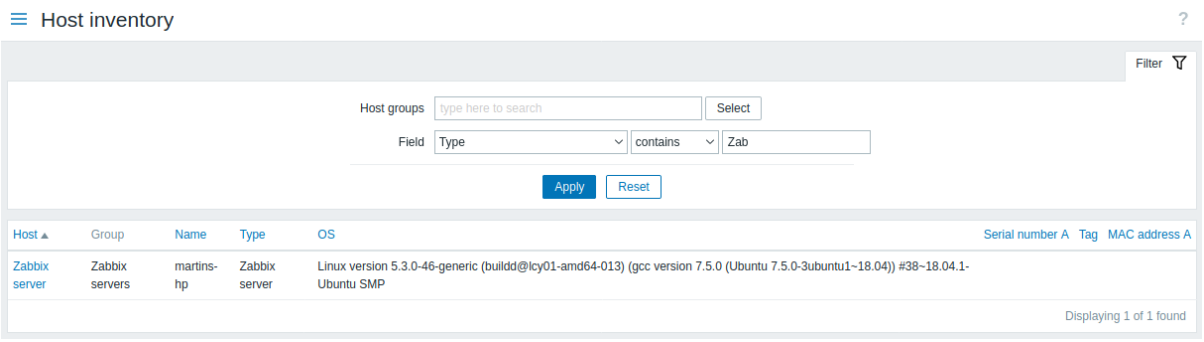


ホスト

概要

Inventory > *Hosts* セクションでは、ホストのinventory data が表示されます。

ホストグループや任意の inventory フィールドでホストを絞り込み、関心のあるホストだけを表示できます。

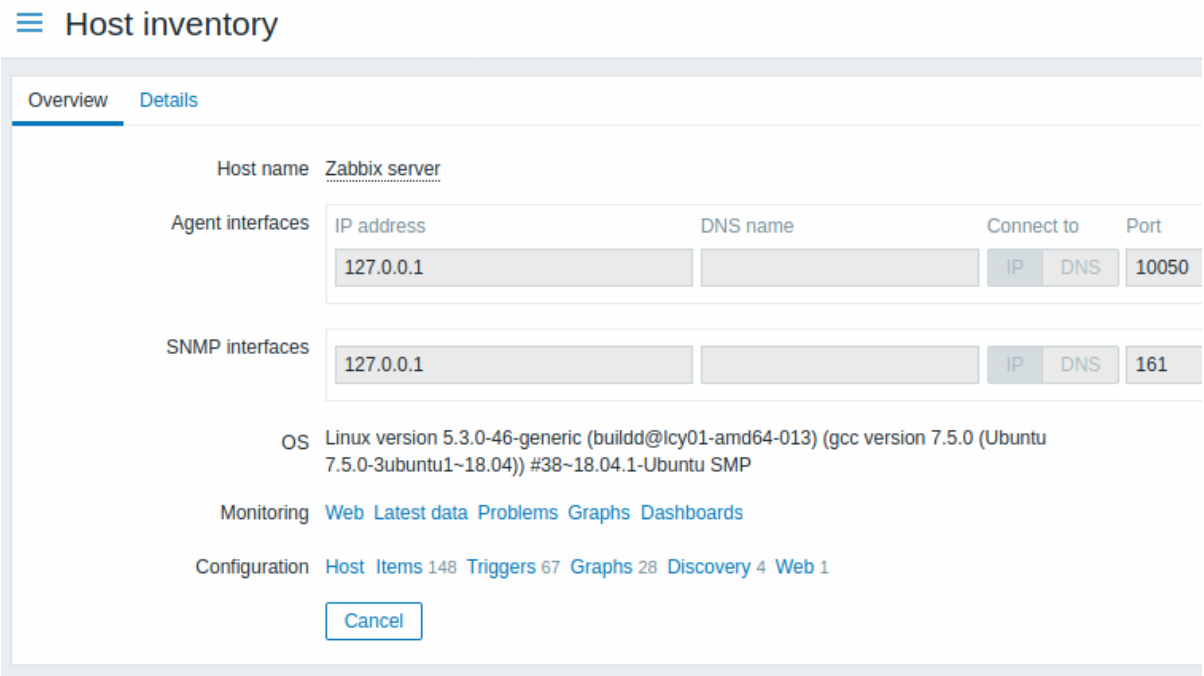


すべてのホストの inventory を表示するには、*Reset* ボタンをクリックします。

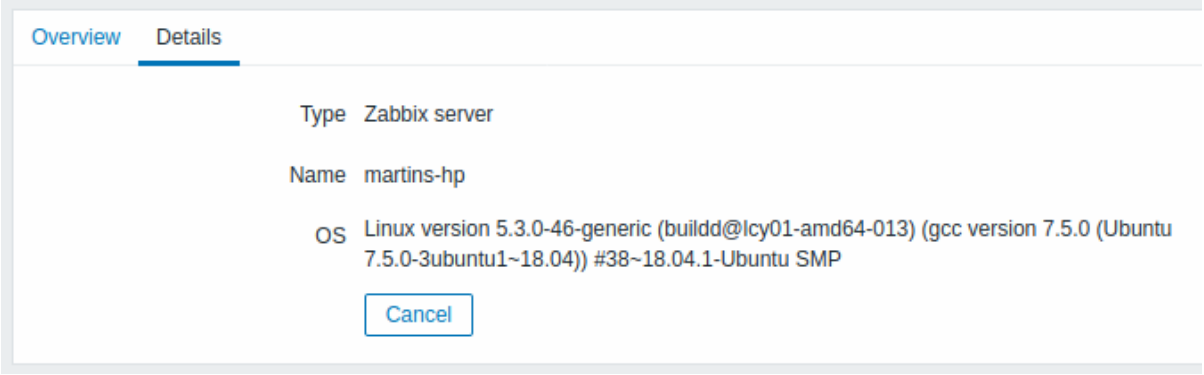
表には一部の主要な inventory フィールドのみが表示されますが、そのホストの利用可能な inventory 情報をすべて確認することもできます。その場合は、最初の列にあるホスト名をクリックします。

インベントリの詳細

概要タブには、ホストに関する一般的な情報、最新の監視データ、およびホストの設定オプションが含まれています。



詳細タブには、ホストのすべての利用可能なインベントリの詳細が含まれています。



インベントリデータの完全性は、ホストで管理されているインベントリ情報の量によって異なります。情報が管理されていない場合、詳細タブは無効になります。

レポート

概要

レポートメニューには、システム情報、トリガー、収集されたデータなどのパラメーターの表示に焦点を当てた、事前定義されたユーザーがカスタマイズ可能なさまざまなレポートを含むいくつかのセクションがあります。

システム情報

概要

レポート > システム情報では、Zabbix サーバーおよびシステムの主要なデータの概要が表示されます。システムデータは、内部アイテムを使用して収集されます。

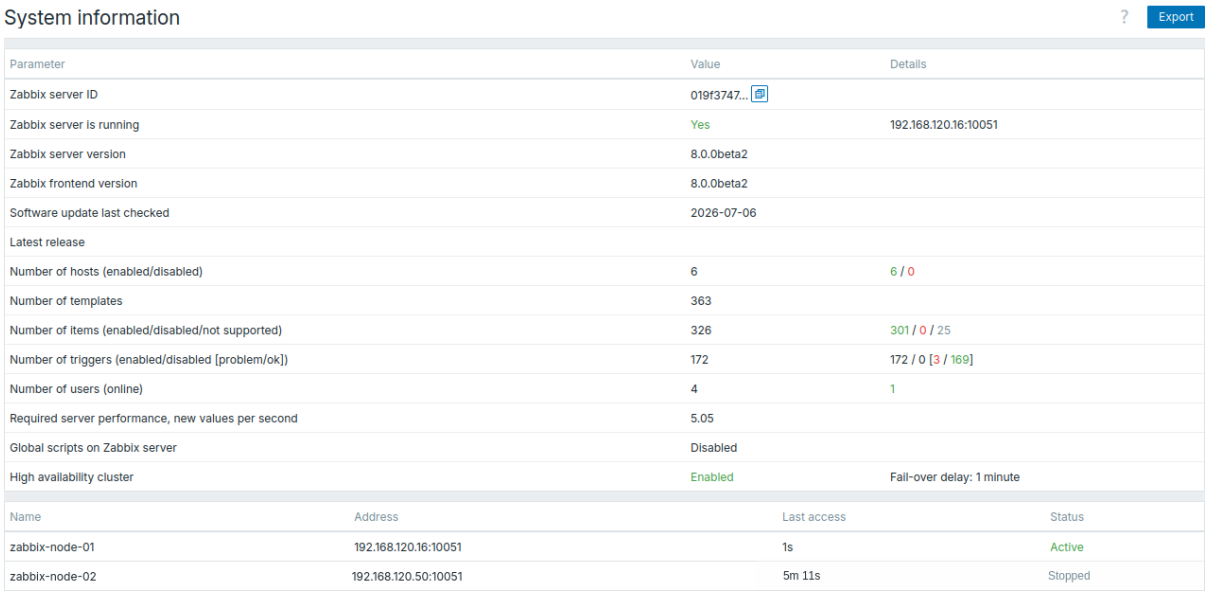
システム情報レポートは、Zabbix の特権管理者および Zabbix の管理者ユーザーのみが表示できます。

高可用性構成では、システム情報のソース（サーバーインスタンス）をリダイレクトできます。これを行うには、Web インターフェースの設定ファイルを編集し、$ZBX_SERVER、または $ZBX_SERVER と $ZBX_SERVER_PORT の両方のコメントを解除して、現在アクティブとして表示されているサーバー以外のサーバーを設定します。$ZBX_SERVER のみを設定した場合、$ZBX_SERVER_PORT にはデフォルト値（10051）が使用されます。

高可用性構成を有効にすると、システム統計の下に、高可用性ノードの詳細を示す個別のブロックが表示されます。

システム情報は、ダッシュボードのウィジェットとしても利用できます。

システム情報



表示されるデータ:

パラメータ	値	詳細
Zabbix サーバー *ID*	Zabbix サーバーの一意の識別子（UUIDv7）。	サーバー ID は、サーバーの初回起動時に生成されます。最初の 8 文字の後に省略記号が表示されます。サーバー ID の値にカーソルを合わせるかクリックすると、完全な値を示すツールチップが表示されます。 値の横にあるアイコン をクリックすると、ID がクリップボードにコピーされます。 高可用性が有効な場合、各ノードにはコマンドラインアドレス指定用の独自のノード ID があります。ただし、すべてのノードは、現在アクティブなノードに関係なく、最初に起動されたノード（または以前にスタンドアロンサーバーだったノード）によって生成された単一のサーバー ID の下にグループ化されます。
Zabbix サーバーは稼働中	Zabbix サーバーのステータス: はい - サーバーは稼働中です いいえ - サーバーは稼働していません 注: 残りの情報を表示するには、Web インターフェースでサーバーが稼働している必要があり、サーバー上で少なくとも 1 つの trapper プロセスが起動している必要があります（`zabbix_server.conf`ファイルの StartTrappers パラメータが 0 より大きい値）。	Zabbix サーバーの場所とポート。
Zabbix サーバーのバージョン	現在のサーバーバージョン番号が表示されます。 注: Zabbix サーバーが稼働している場合にのみ表示されます。	サーバーバージョンのステータスが表示されます: 最新 - 最新バージョンを使用中; 新しいアップデートが利用可能 - より新しいバージョンが利用可能; 古い - このバージョンの完全サポート期間が終了。 この情報は、Zabbix サーバー設定でソフトウェアアップデートの確認が有効になっている場合にのみ利用できます。最後のソフトウェアアップデート確認が 1 週間以上前に実行された場合、または現在のバージョンに関するデータが存在しない場合は、何も表示されません。
Zabbix Web インターフェースのバージョン	Zabbix Web インターフェースのバージョン番号が表示されます。	Zabbix Web インターフェースのバージョンステータスが表示されます: 最新 - 最新バージョンを使用中; 新しいアップデートが利用可能 - より新しいバージョンが利用可能; 古い - このバージョンの完全サポート期間が終了。 この情報は、Zabbix サーバー設定でソフトウェアアップデートの確認が有効になっている場合にのみ利用できます。最後のソフトウェアアップデート確認が 1 週間以上前に実行された場合、または現在のバージョンに関するデータが存在しない場合は、何も表示されません。
ソフトウェアアップデートの最終確認	Zabbix ソフトウェアアップデートを最後に確認した日付が表示されます。 この情報は、Zabbix サーバー設定でソフトウェアアップデートの確認が有効になっている場合にのみ利用できます。	
最新リリース	現在の Zabbix バージョンで利用可能な、より新しいリリースの番号（存在する場合）が表示されます。 この情報は、Zabbix サーバー設定でソフトウェアアップデートの確認が有効になっている場合にのみ利用できます。最後のソフトウェアアップデート確認が 1 週間以上前に実行された場合、または現在のバージョンに関するデータが存在しない場合は、何も表示されません。	利用可能な最新の Zabbix リリースのリリースノートへのリンクが表示されます。
ホスト数	設定されているホストの合計数が表示されます。	監視中のホスト数/監視されていないホスト数。
テンプレート数	テンプレートの合計数が表示されます。	

パラメータ	値	詳細
アイテム数	アイテムの合計数が表示されます。	監視中/無効/サポートされていないホストレベルのアイテム数。 無効なホスト上のアイテムは、無効としてカウントされます。
トリガー数	トリガーの合計数が表示されます。	有効/無効なホストレベルのトリガー数。「障害」/「正常」状態に応じた有効なトリガーの内訳。 「正常」状態に一覧表示されるトリガーには、ステータスが「不明」のトリガーも含まれます。 無効なアイテムに依存するトリガー、または無効なホストに割り当てられたトリガーは、無効としてカウントされます。
ユーザー数	設定されているユーザーの合計数が表示されます。	オンラインユーザー数。
必要なサーバーパフォーマンス、*1* 秒あたりの新しい値	Zabbix サーバーが 1 秒あたりに処理すると予想される新しい値の数が表示されます。	必要なサーバーパフォーマンスは推定値であり、目安として役立ちます。処理された値の正確な数を確認するには、`zabbix[wcache,values,all]` 内部アイテムを使用してください。 監視中のホストの有効なアイテムが計算に含まれます。ログアイテムは、アイテムの更新間隔ごとに 1 つの値としてカウントされます。通常の間隔の値はカウントされますが、柔軟な間隔およびスケジュール間隔の値はカウントされません。「データなし」メンテナンス期間中に計算は調整されません。 trapper アイテムはカウントされません。
Zabbix サーバー上のグローバルスクリプト	サーバー設定で EnableGlobalScripts=0 を設定して Zabbix サーバー上のグローバルスクリプトを無効にすると、このフィールドに無効と表示されます。	
高可用性クラスター	Zabbix サーバーの高可用性クラスターのステータス: 無効 - スタンドアロンサーバー 有効 - 少なくとも 1 つの高可用性ノードが存在	有効な場合、フェイルオーバー遅延が表示されます。

システム情報には、次の条件でエラーメッセージも表示されます:

- 使用されているデータベースに必要な文字セットまたは照合順序（UTF-8）がありません。
- データベースのバージョンがサポート範囲を下回っている、または上回っている（スーパー管理者ロールタイプのユーザーのみ利用可能）。
- TimescaleDBのハウスキーピングが正しく設定されていません（履歴またはトレンドテーブルに圧縮チャンクが含まれているにもかかわらず、アイテムの履歴期間を上書きまたはアイテムのトレンド期間を上書きオプションが無効になっている）。

ボタン

ページ右上のボタンには、次のオプションがあります。

Export	システム情報を JSON ファイルにエクスポートします。 ファイルは `Zabbix-system-information-{time}.json` として保存されます。ここで `{time}` は、サーバー上でデータが準備された時刻を示す Unix タイムスタンプです。 高可用性が有効な場合は、高可用性ノードの詳細もエクスポートされます。

高可用性ノード

高可用性クラスターが有効な場合、各高可用性ノードの状態を示す追加のデータブロックが表示されます。



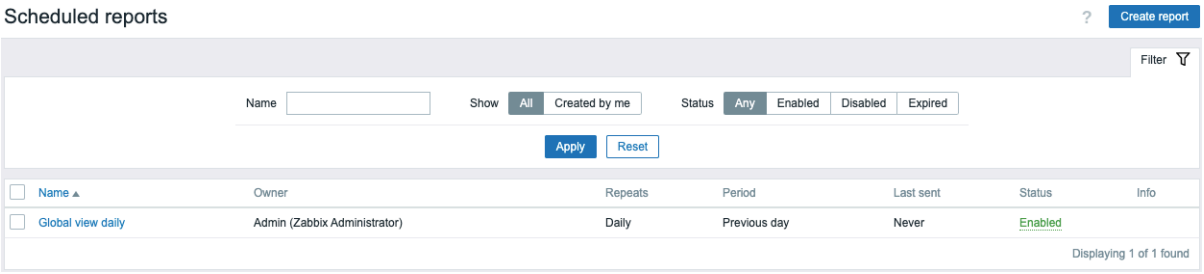
表示されるデータ:

Column	Description
Name	ノード名。サーバー設定で定義されます。
Address	ノードの IP アドレスとポート。
Last access	ノードの最終アクセス時刻。 セルにカーソルを合わせると、最終アクセス時刻が長い形式で表示されます。
Status	ノードの状態。テーブル行は、次の優先順位でこれらの状態に従って並べ替えられます: **Active** - ノードは稼働しており、動作しています **Unavailable** - ノードが failover delay を超えて確認されていません（原因を確認することをお勧めします） **Stopped** - ノードが停止しているか、起動できませんでした（起動するか削除することをお勧めします） **Standby** - ノードは稼働しており、待機しています

スケジュールされたレポート

概要

レポート > スケジュールレポートでは、十分な権限を持つユーザーが、ダッシュボードの PDF 版を定期的に生成し、指定した受信者にメールで送信するよう設定できます。



開始画面には、スケジュールレポートに関する情報が表示されます。情報はナビゲーションしやすいようにフィルターできます。詳細については、以下のフィルターの使用セクションを参照してください。

表示されるデータ:

列	説明
名前	レポートの名前。クリックするとレポート設定フォームが開きます。
所有者	レポートを作成したユーザー。
繰り返し	レポート生成の頻度（日次/週次/月次/年次）。
期間	レポートを作成する対象期間（前日、前週、前月、または前年）。
最終送信日時	最新のレポートが送信された日時。
ステータス	レポートの現在のステータス（有効/無効/期限切れ）。十分な権限を持つユーザーは、ステータスをクリックして「有効」から「無効」（またはその逆）に変更できます。「期限切れ」から「無効」（またはその逆）に変更することもできます。権限が不十分なユーザーの場合、ステータスはクリックできません。

列	説明
情報	情報アイコンが表示されます: 赤いアイコンは、レポートの生成に失敗したことを示します。アイコンにカーソルを合わせると、エラー情報を含むツールチップが表示されます。 黄色いアイコンは、レポートは生成されたものの、一部またはすべての受信者への送信に失敗したこと、またはレポートの有効期限が切れていることを示します。アイコンにカーソルを合わせると、追加情報を含むツールチップが表示されます。

フィルターの使用

フィルターを使用して、レポートのリストを絞り込むことができます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

利用可能なフィルターオプションは次のとおりです。

- 名前 - 部分一致が可能
- 表示 - 現在のユーザーが作成したレポートまたはすべてのレポート
- ステータス - 「すべて」(すべてのレポートを表示)、「有効」、「無効」、「期限切れ」から選択

フィルターは定期レポートセクション名の下にあります。右上のフィルタータブをクリックすることで、フィルターを開いたり折りたたんだりできます。

一括更新

複数のレポートのステータスを一度に削除または変更したい場合があります。その場合、各レポートを個別に編集する代わりに、一括更新機能を使用できます。

複数のレポートを一括更新するには、以下の手順を実行します。

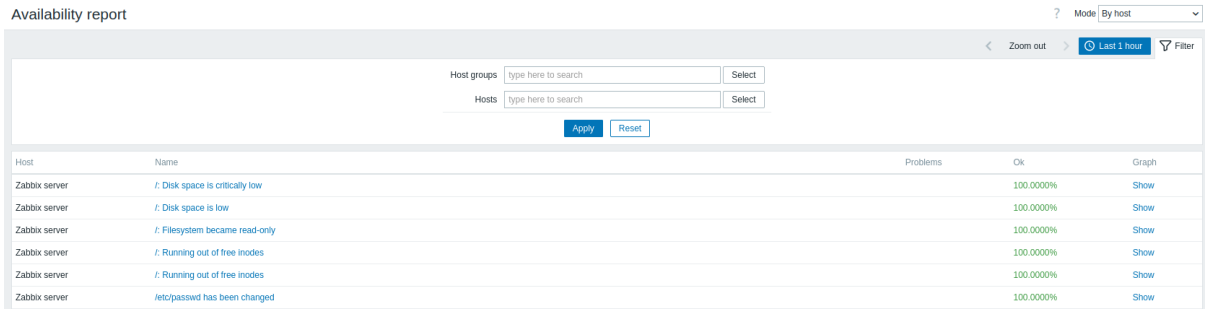
- 一覧で更新したいレポートのチェックボックスをオンにします
- 一覧の下にある必要なボタン（有効、無効、削除）をクリックして変更を行います

可用性レポート

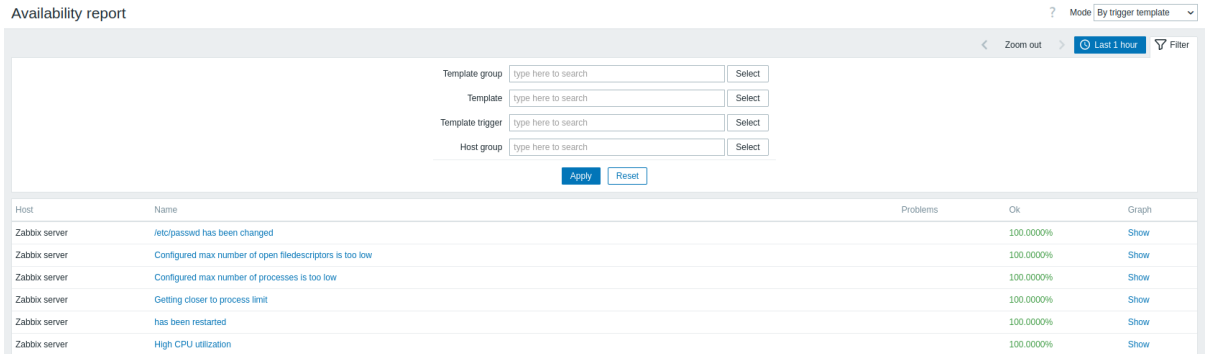
概要

レポート > 可用性レポートでは、各トリガーが問題/正常状態にあった時間の割合を確認できます。

各状態ごとに時間の割合が表示されるため、システム上のさまざまな要素の可用性を簡単に判断できます。



右上のドロップダウンで選択モードを選択できます。ホストごとにトリガーを表示するか、テンプレートに属するトリガーごとに表示するかを選択できます。



トリガー名は、そのトリガーの最新イベントへのリンクです。

フィルターの使用

フィルターを使用すると、表示されるホストやトリガーの数を絞り込むことができます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

フィルターは、*Availability report* セクション名の下にあります。右側の *Filter* タブをクリックすると、展開および折りたたみができます。

トリガーテンプレートによるフィルタリング

トリガーテンプレートによるモードでは、以下のいずれかまたは複数のパラメータで結果をフィルタリングできます。

パラメータ	説明
テンプレートグループ	選択したテンプレートグループに属するテンプレートから継承されたトリガーによってホストをフィルタリングします。 親テンプレートグループを指定すると、すべてのネストされたテンプレートグループが暗黙的に選択されます。
テンプレート	選択したテンプレート（ネストされたテンプレートを含む）から継承されたトリガーによってホストをフィルタリングします。 ネストされたテンプレートに独自のトリガーがある場合、それらのトリガーは表示されません。
テンプレートトリガー	選択したトリガーによってホストをフィルタリングします。フィルタリングされたホストの他のトリガーは表示されません。
ホストグループ	選択したホストグループに属するホストをフィルタリングします。

ホストによるフィルタリング

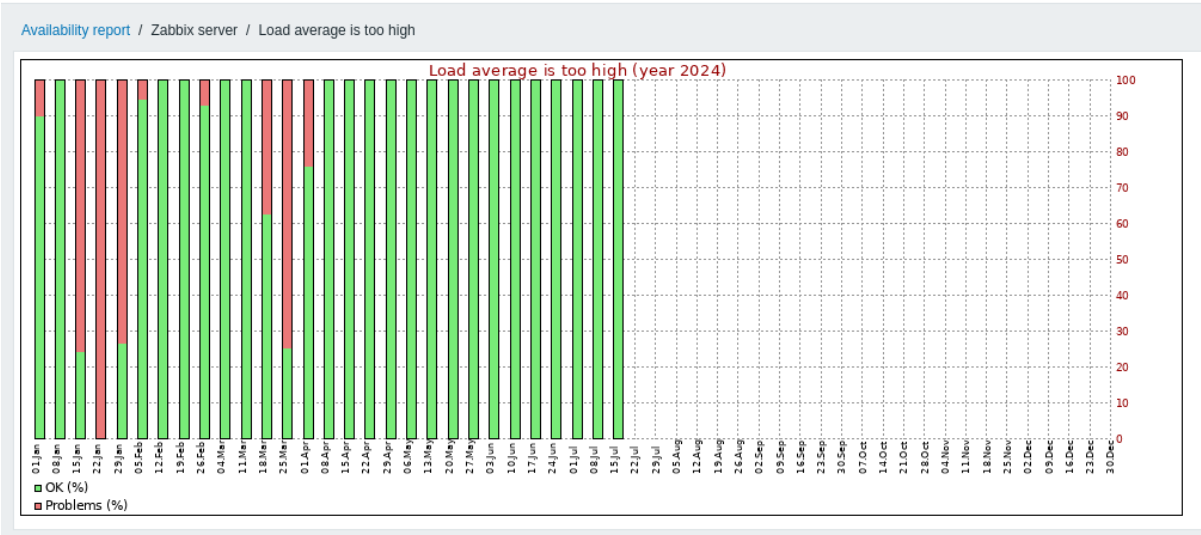
ホストによるモードでは、ホストまたはホストグループで結果をフィルタリングできます。親ホストグループを指定すると、すべてのネストされたホストグループが暗黙的に選択されます。

期間セレクター

期間セレクターを使用すると、よく使用する期間をワンクリックで選択できます。セレクターは、フィルターの横にある期間タブをクリックして展開および折りたたむことができます。

グラフ列の表示をクリックすると、現在の年の経過した週をそれぞれ 1 本のバーで表した棒グラフに、可用性情報が表示されます。

Availability report graph



バーの緑色は正常な時間を、赤色は障害の時間を表します。

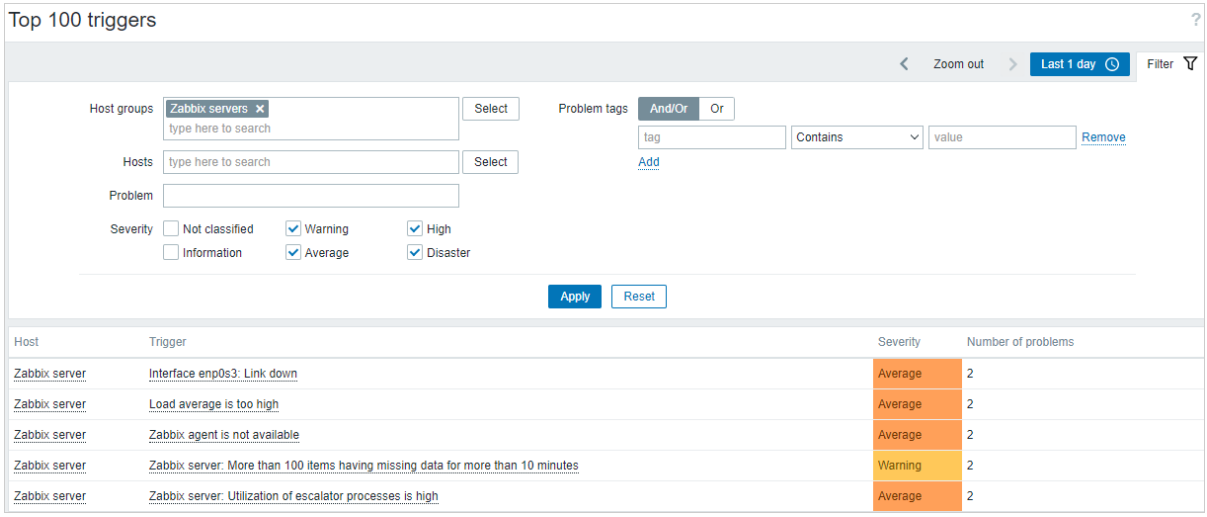
メンテナンス期間の影響

メンテナンスは、可用性レポートから時間を自動的に除外しません。データ収集なしとして設定されたメンテナンスのみがポーリングを停止し（そのため障害は生成されません）、結果として影響を受けるトリガーの可用性率は変わりません。

トップ **100** トリガー

概要

レポート > 上位 *100* トリガーでは、選択した期間中に検出された障害の数が最も多いトリガーを確認できます。



ホスト列とトリガー列の項目は、便利なオプションを提供するリンクになっています。

- ホストの場合 - ホスト名をクリックすると、ホストメニューが表示されます
- トリガーの場合 - トリガー名をクリックすると、最新のイベント、各トリガーアイテムのシンプルグラフ、トリガー自体および各トリガーアイテムの設定フォームへのリンクが表示されます

フィルターの使用

ホストグループ、ホスト、障害名、タグ、またはトリガーの深刻度でトリガーを表示するには、フィルターを使用します。親ホストグループを指定すると、ネストされたすべてのホストグループが暗黙的に選択されます。検索パフォーマンスを向上させるため、データはマクロを未解決の状態で検索されます。

フィルターは、上位 *100* トリガーセクション名の下にあります。右側のフィルタータブをクリックすると、開いたり折りたたんだりできます。

期間セレクター

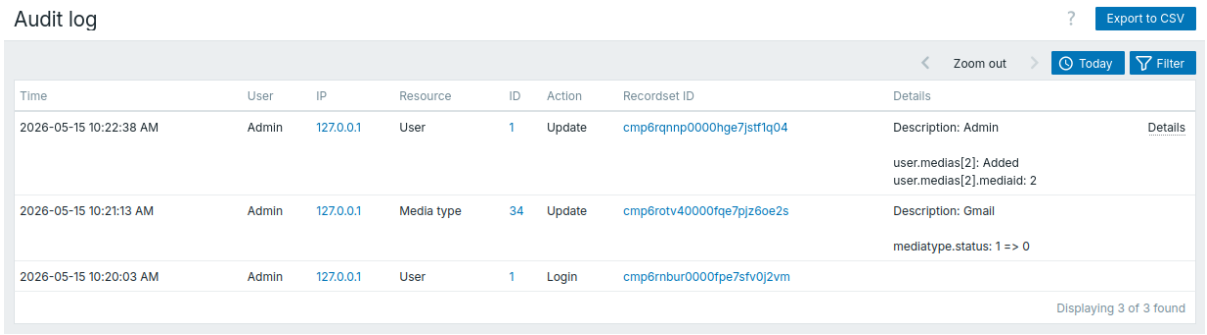
フィルターの横にある期間セレクターを使用すると、よく使用する期間をマウスのワンクリックで選択できます。詳細については、期間とホストのセレクターを参照してください。

監査ログ

概要

Reports > *Audit log* セクションでは、ユーザーおよびシステムのアクティビティの記録を表示できます。

Note:
監査記録を収集して表示するには、*Administration* > *Audit log* セクションで *Enable audit logging* チェックボックスをオンにする必要があります。この設定を有効にしない場合、アクティビティの履歴はデータベースに記録されず、監査ログにも表示されません。



監査ログには、次のデータが表示されます。

Column	Description
Time	監査記録のタイムスタンプ。
User	アクティビティを実行したユーザー。

Column	Description
IP	アクティビティが開始された IP。 ハイパーリンクをクリックすると、この IP で監査ログ記録がフィルタリングされます。
Resource	影響を受けたリソースの種類（*API token*、*Action*、*Authentication*、*Autoregistration* など）。
ID	影響を受けたリソースの ID。 ハイパーリンクをクリックすると、このリソース ID で監査ログ記録がフィルタリングされます。
Action	アクティビティの種類（*Add*、*Configuration refresh*、*Delete*、*Execute*、*Failed login*、*History clear*、*Login*、*Logout*、*Push*、*Update*）。
Recordset ID	同じ操作の結果として作成されたすべての監査ログ記録で共有される ID。 たとえば、テンプレートをホストにリンクすると、継承された各テンプレートエンティティ（アイテム、トリガーなど）ごとに個別の監査ログ記録が作成されます。これらの記録はすべて同じ *Recordset ID* を持ちます。 ハイパーリンクをクリックすると、この *Recordset ID* で監査ログ記録がフィルタリングされます。
Details	リソースの説明と、実行されたアクティビティの詳細情報。 記録に 2 行を超える内容が含まれる場合は、追加の *Details* リンクが表示されます。このリンクをクリックすると、変更内容の全一覧を表示できます。

Note:
trapper アイテム またはHTTP agent アイテム（トラッピング有効）がデータを受信した場合、監査ログにエントリが追加されるのは、そのデータが`history.push` API メソッドを使用して送信された場合のみであり、Zabbix sender ユーティリティを使用した場合は追加されません。

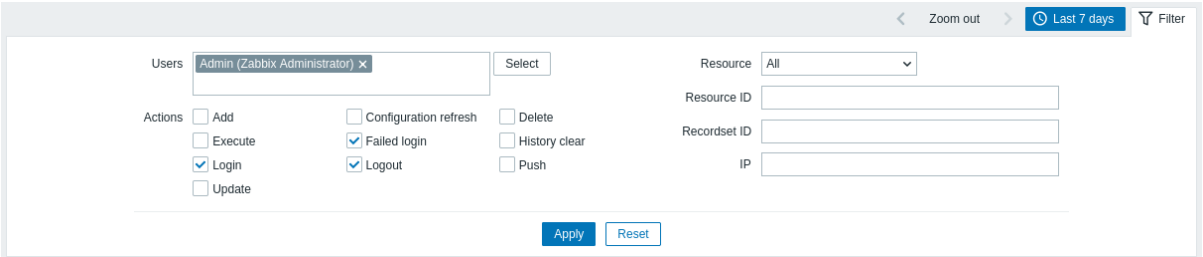
ボタン

ページの右上隅にあるボタンには、次のオプションがあります。

Export to CSV	すべてのページの監査ログレコードを CSV ファイルにエクスポートします。フィルターが適用されている場合は、フィルターされたレコードのみがエクスポートされます。

フィルターの使用

フィルターは監査ログバーの下にあります。右上隅のフィルタータブをクリックすると、フィルターを開いたり折りたたんだりできます。



フィルターを使用して、ユーザー、影響を受けたリソース、リソース ID、実行された操作（レコードセット *ID*）、IP でレコードを絞り込むことができます。リソースによっては、フィルターで 1 つまたは複数の特定のアクションを選択できます。

検索パフォーマンスを向上させるために、すべてのデータはマクロを解決せずに検索されます。

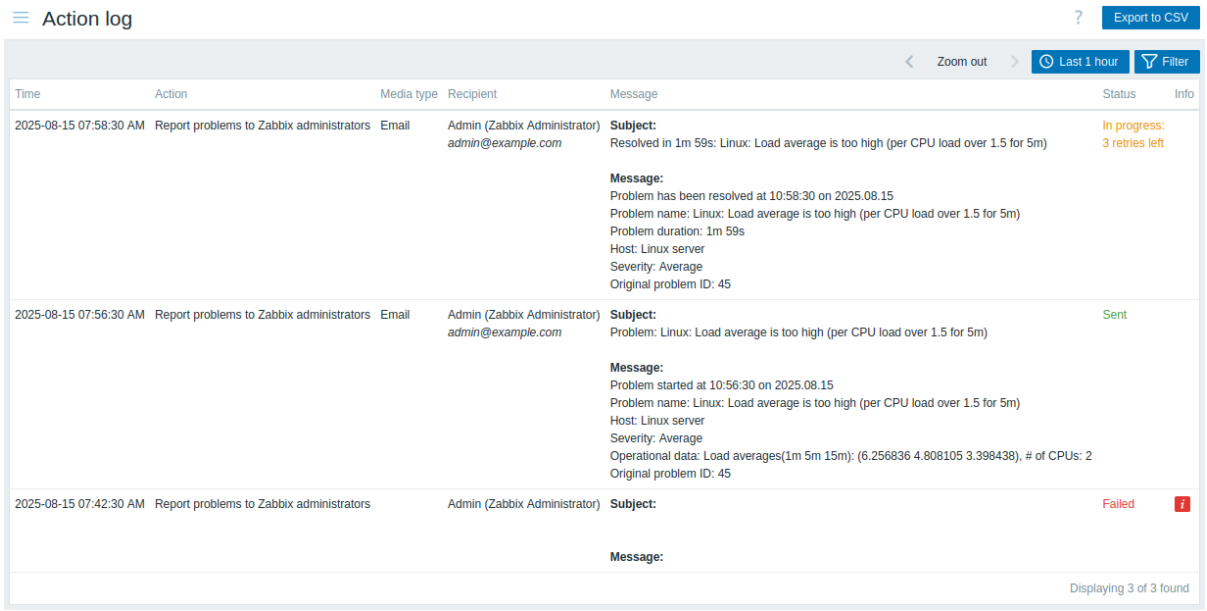
期間セレクター

フィルターの横にある期間セレクターを使用すると、頻繁に使用する期間をマウスクリック 1 回で選択できます。詳細については、期間およびホストセレクターを参照してください。

アクションログ

概要

レポート > アクションログでは、アクション内で実行された操作（通知、リモートコマンド）の詳細を確認できます。



表示されるデータ:

Column	Description
Time	操作のタイムスタンプ。
Action	操作の原因となったアクション名。
Media type	通知の送信に使用されたメディアタイプ（例: Email、Jira など）。 リモートコマンドを実行した操作では、この列は空になります。
Recipient	通知の受信者に関する情報 - ユーザー名、名前と姓（括弧内）、およびメディアタイプに応じた追加情報（email、ユーザー名など）。 リモートコマンドを実行した操作では、この列は空になります。
Message	メッセージの内容、リモートコマンド、またはグローバルスクリプト名。 リモートコマンドは対象ホストとコロンで区切られます: `<host>:<command>`。たとえば、リモートコマンドが Zabbix server で実行された場合、情報は次の形式になります: `Zabbix server:<command>`。
Status	操作のステータス: *In progress* - 通知送信の操作が進行中です（サーバーが通知の送信を試行する残り回数も表示されます） *Sent* - 通知が送信されました *Executed* - リモートコマンドが実行されました *Failed* - 操作が完了しませんでした
Info	操作の実行に関するエラー情報（ある場合）。

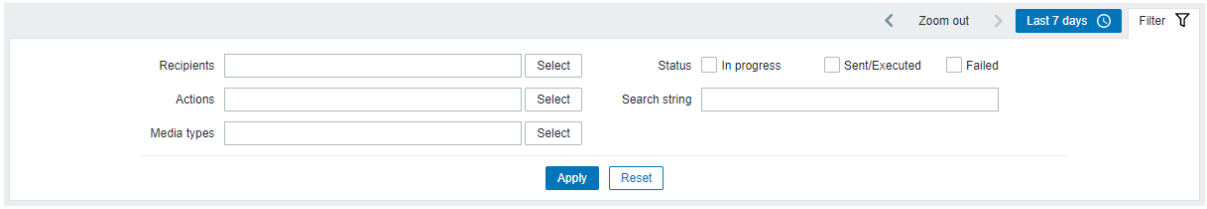
ボタン

ページの右上隅にあるボタンには、次のオプションがあります。

Export to CSV	現在のページのアクションログレコードを CSV ファイルにエクスポートします。フィルターが適用されている場合は、フィルターされたレコードのみがエクスポートされます。 エクスポートされた CSV ファイルでは、「Recipient」と「Message」列が複数の列に分割されます - 「Recipient's Zabbix username」、「Recipient's name」、「Recipient's surname」、「Recipient」、および「Subject」、「Message」、「Command」。

フィルターの使用

フィルターは *Action log* バーの下にあります。ページ右上隅の *Filter* タブをクリックすると、展開および折りたたみができます。



フィルターを使用すると、通知の受信者、アクション、メディアタイプ、ステータス、またはメッセージ/リモートコマンドの内容（*Search string*）でレコードを絞り込むことができます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

期間セレクター

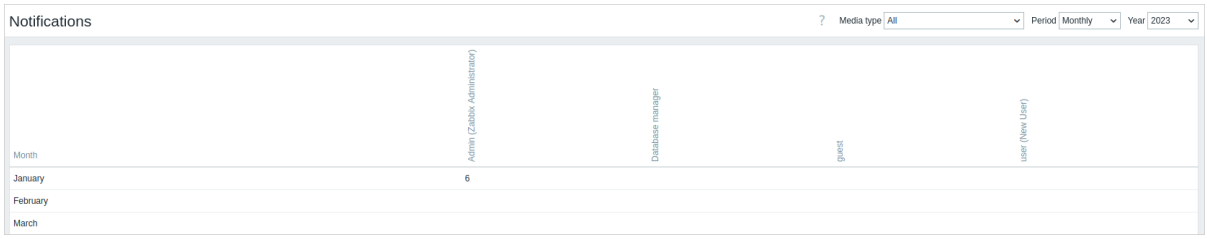
フィルターの横にある期間セレクターを使用すると、頻繁に必要となる期間をマウスのワンクリックで選択できます。詳細については、期間およびホストセレクターを参照してください。

通知

概要

Reports > *Notifications* セクションでは、各ユーザーに送信された通知数のレポートが表示されます。

右上のドロップダウンから、送信された通知のメディアタイプ（またはすべて）、期間（各日/週/月/年のデータ）、および年を選択できます。



各列には、1 人のシステムユーザーごとの合計が表示されます。

データ収集

概要

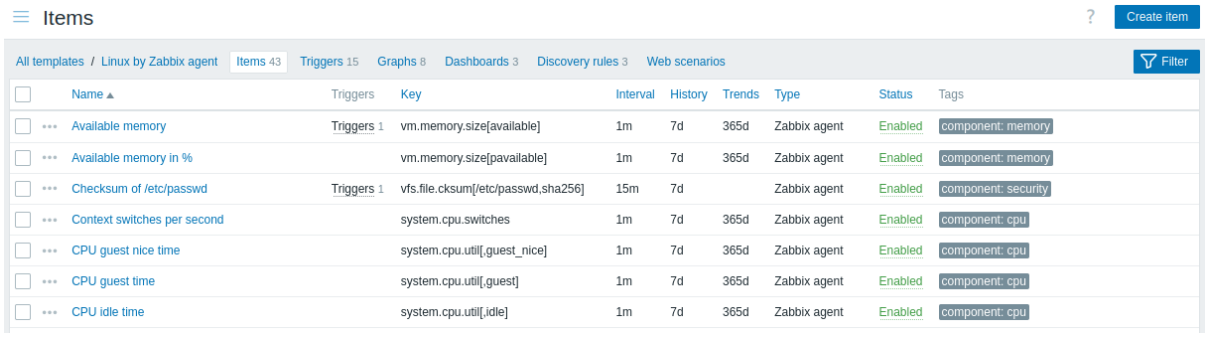
このメニューには、データ収集の設定に関連するセクションがあります。

アイテム

概要

テンプレートのアイテム一覧には、データ収集 > テンプレートから、対象のテンプレートの アイテムをクリックしてアクセスできます。

既存のアイテム一覧が表示されます。



表示されるデータ:

列	説明
アイテムメニュー	3 点アイコンをクリックすると、アイテムメニューが開きます。
テンプレート	アイテムが所属するテンプレート。 テンプレート名をクリックすると、テンプレート設定フォームが開きます。 フィルターで複数のテンプレートまたはテンプレートが選択されていない場合にのみ、この列が表示されます。

列	説明
名前	アイテムの名前。アイテムの詳細への青色のリンクとして表示されます。 アイテム名のリンクをクリックすると、アイテム設定フォームが開きます。 アイテムが別のテンプレートから継承されている場合、アイテム名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのアイテム一覧が開きます。
トリガー	「トリガー」にマウスを合わせると、アイテムに関連付けられたトリガーを表示するインフォボックスが表示されます。 トリガーの数は灰色で表示されます。
キー	アイテムキーが表示されます。
監視間隔	チェックの実行頻度が表示されます。
ヒストリ	アイテムデータのヒストリを保持する日数が表示されます。
トレンド	アイテムのトレンド履歴を保持する日数が表示されます。
タイプ	アイテムのタイプ（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）が表示されます。
ステータス	アイテムのステータス（有効または 無効）が表示されます。ステータスをクリックすると、有効から無効（またはその逆）に変更できます。
タグ	アイテムのタグが表示されます。 最大 3 つのタグ（名前: 値のペア）を表示できます。タグが 4 つ以上ある場合は、マウスを合わせるとすべてのタグを表示できる「...」リンクが表示されます。

新しいアイテムを設定するには、右上隅にある アイテムの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

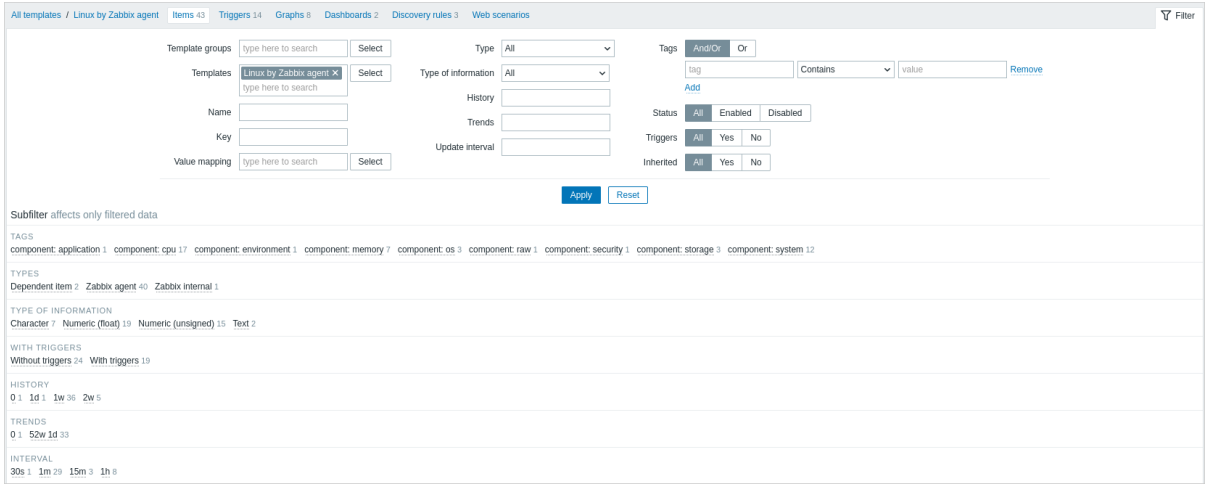
- *Enable* - アイテムのステータスを *Enabled* に変更します。
- *Disable* - アイテムのステータスを *Disabled* に変更します。
- *Copy* - アイテムを他のホストまたはテンプレートにコピーします。
- *Mass update* - 複数のアイテムの複数のプロパティを更新します。
- *Delete* - アイテムを削除します。

これらのオプションを使用するには、該当するアイテムの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

アイテム一覧には多数のアイテムが含まれる場合があります。フィルターを使用すると、その一部を絞り込み、探しているアイテムをすばやく見つけることができます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

Filter アイコンは右上隅にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



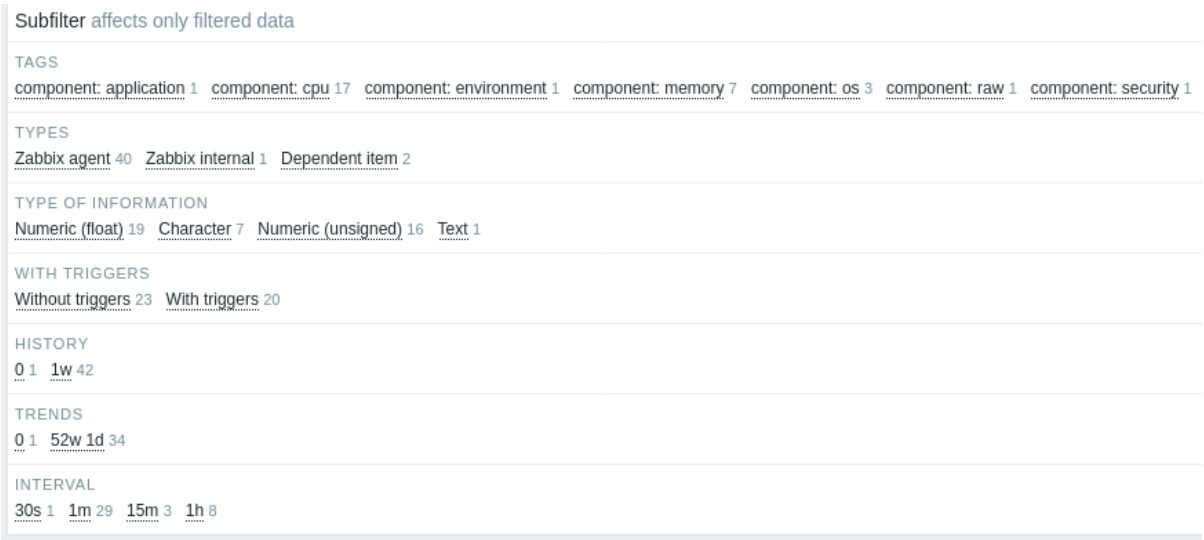
Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、すべてのネストされたグループが暗黙的に選択されます。
Templates	1 つ以上のテンプレートで絞り込みます。
Name	アイテム名で絞り込みます。
Key	アイテムキーで絞り込みます。

Parameter	Description
Value mapping	使用されている値マップで絞り込みます。 *Templates* オプションが空の場合、このパラメーターは表示されません。
Type	アイテムタイプ（Zabbix agent、SNMP agent など）で絞り込みます。
Type of information	情報の種類（Numeric unsigned、float など）で絞り込みます。
History	アイテムの履歴データの保持期間で絞り込みます。
Trends	アイテムのトレンドデータの保持期間で絞り込みます。
Update interval	アイテムの更新間隔で絞り込みます。
Tags	表示するアイテム数を制限するためにタグを指定します。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件には次の演算子を使用できます: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別） **Contains** - タグ値に入力した文字列を含む指定タグ名を含める（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外する（部分一致、大文字と小文字を区別しない） 条件の計算方法には 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
Status	アイテムのステータスで絞り込みます - *Enabled* または *Disabled*。
Triggers	トリガーがあるアイテム、またはないアイテムで絞り込みます。
Inherited	リンクされたテンプレートから継承されたアイテム、または継承されていないアイテムで絞り込みます。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターで適用したフィルタリングをさらに変更できます。

サブフィルターには、関連するアイテムにすばやくアクセスするためのクリック可能なリンクが含まれています。アイテムは、共通のエンティティ（タグ、アイテムのタイプ、アイテムの更新間隔など）によって関連付けられています。エンティティをクリックすると、そのエンティティが灰色の背景で強調表示され、アイテムがすぐにフィルタリングされます（メインフィルターで 適用をクリックする必要はありません）。別のエンティティをクリックすると、そのエンティティがフィルタリング結果に追加されます。エンティティをもう一度クリックすると、そのフィルタリングが解除されます。



サブフィルターは、フィルタリングされたデータに基づいて生成されます。対象となるデータは最大 1000 レコードに制限されています。サブフィルターにさらに多くのレコードを表示するには、検索およびフィルター結果の上限パラメータの値を増やす必要があります（管理 > 一般 > *GUI*）。

メインフィルターとは異なり、サブフィルターはテーブルの更新リクエストごとに更新されます。これにより、利用可能なフィルタリングオプションとその件数が常に最新の状態になります。

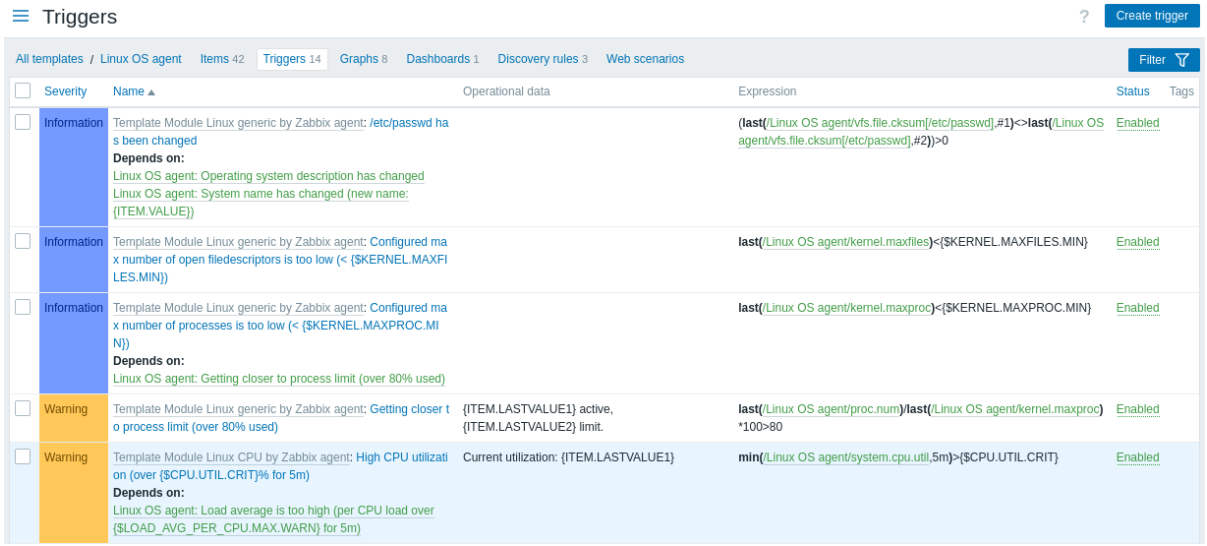
表示されるエンティティの数は、水平方向に最大 100 個に制限されています。100 を超える場合は、末尾に 3 点アイコンが表示されます。このアイコンはクリックできません。

各クリック可能なエンティティの横に表示される数字は、そこにグループ化されたアイテムの数を示します（メインフィルターの結果に基づきます）。エンティティをクリックすると、その他の利用可能なエンティティの横にプラス記号付きの数字が表示されます。この数字は、現在の選択に追加できるアイテムの数を示します。アイテムのないエンティティは、サブフィルターで以前に選択されていない限り表示されません。

トリガー

概要

テンプレートのトリガー一覧には、データ収集 > テンプレートから、対象のテンプレートの トリガーをクリックしてアクセスできます。



表示されるデータ:

列	説明
深刻度	トリガーの深刻度が、名前とセルの背景色の両方で表示されます。
テンプレート	トリガーが属するテンプレート。 テンプレート名をクリックすると、テンプレート設定フォームが開きます。 この列は、フィルターで複数のテンプレートまたはテンプレートが選択されていない場合にのみ表示されます。
名前	トリガーの名前が、トリガーの詳細への青色のリンクとして表示されます。 トリガー名のリンクをクリックすると、トリガー設定フォームが開きます。 トリガーが別のテンプレートから継承されている場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのトリガー一覧が開きます。
運用データ	トリガーの運用データ定義。任意の文字列と、監視 > 障害で動的に解決されるマクロが含まれます。
条件式	トリガーの条件式が表示されます。条件式のテンプレート-アイテム部分はリンクとして表示され、アイテム設定フォームに移動できます。
ステータス	トリガーのステータス（有効または 無効）が表示されます。ステータスをクリックすると、有効から無効（またはその逆）に変更できます。
タグ	トリガーにタグが含まれている場合、この列にタグ名と値が表示されます。

新しいトリガーを設定するには、右上隅にある トリガーの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- *Enable* - トリガーのステータスを *Enabled* に変更します
- *Disable* - トリガーのステータスを *Disabled* に変更します
- *Copy* - トリガーを他のホストまたはテンプレートにコピーします
- *Mass update* - 複数のトリガーに対して、いくつかのプロパティを一度に更新します

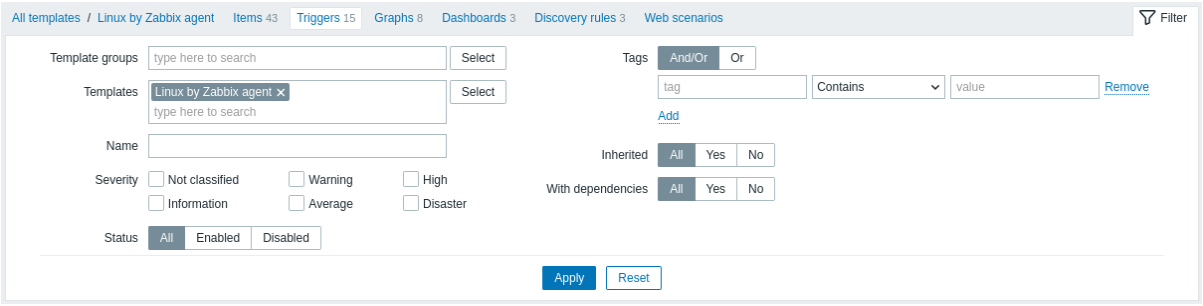
- *Delete* - トリガーを削除します

これらのオプションを使用するには、対象のトリガーの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるトリガーのみを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

フィルターアイコンは右上隅にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



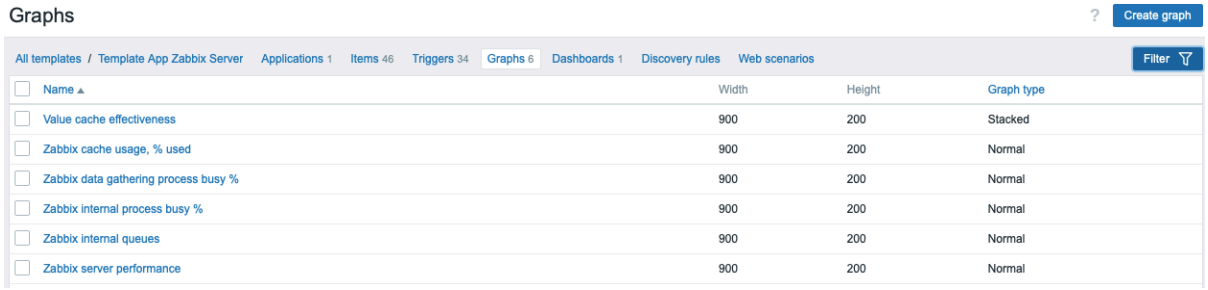
Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループが暗黙的に選択されます。
Templates	1 つ以上のテンプレートで絞り込みます。 上でテンプレートグループがすでに選択されている場合、テンプレートの選択はそれらのグループに限定されます。
Name	トリガー名で絞り込みます。
Severity	1 つ以上のトリガー深刻度で絞り込む場合に選択します。
Status	トリガーの有効化状態（Enabled/Disabled）で絞り込みます。
Tags	トリガーのタグ名と値で絞り込みます。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件には次の演算子を使用できます: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字小文字を区別） **Contains** - タグ値に入力した文字列を含む指定タグ名を含める（部分一致、大文字小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字小文字を区別） **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外する（部分一致、大文字小文字を区別しない） 条件の計算タイプは 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です タグ名およびタグ値のフィールドではマクロとmacro functionsがサポートされています。
Inherited	リンクされたテンプレートから継承されたトリガー（または継承されていないトリガー）で絞り込みます。
With dependencies	依存関係があるトリガー（またはないトリガー）で絞り込みます。

グラフ

概要

テンプレートのカスタムグラフ一覧には、データ収集 > テンプレートから、対象のテンプレートの グラフをクリックしてアクセスできます。

既存のグラフ一覧が表示されます。



表示されるデータ:

列	説明
テンプレート	グラフが属するテンプレート。 テンプレート名をクリックすると、テンプレート設定フォームが開きます。 この列は、フィルターで複数のテンプレートまたはテンプレートが選択されていない場合にのみ表示されます。
名前	カスタムグラフの名前。グラフの詳細への青色のリンクとして表示されます。 グラフ名のリンクをクリックすると、グラフ設定フォームが開きます。 グラフが別のテンプレートから継承されている場合、グラフ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのグラフ一覧が開きます。
幅	グラフの幅が表示されます。
高さ	グラフの高さが表示されます。
グラフの種類	グラフの種類（ノーマル、積み上げ、円、分解）が表示されます。

新しいグラフを設定するには、右上隅にある グラフの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- *Copy* - グラフを他のホストまたはテンプレートにコピーします
- *Delete* - グラフを削除します

これらのオプションを使用するには、対象のグラフの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

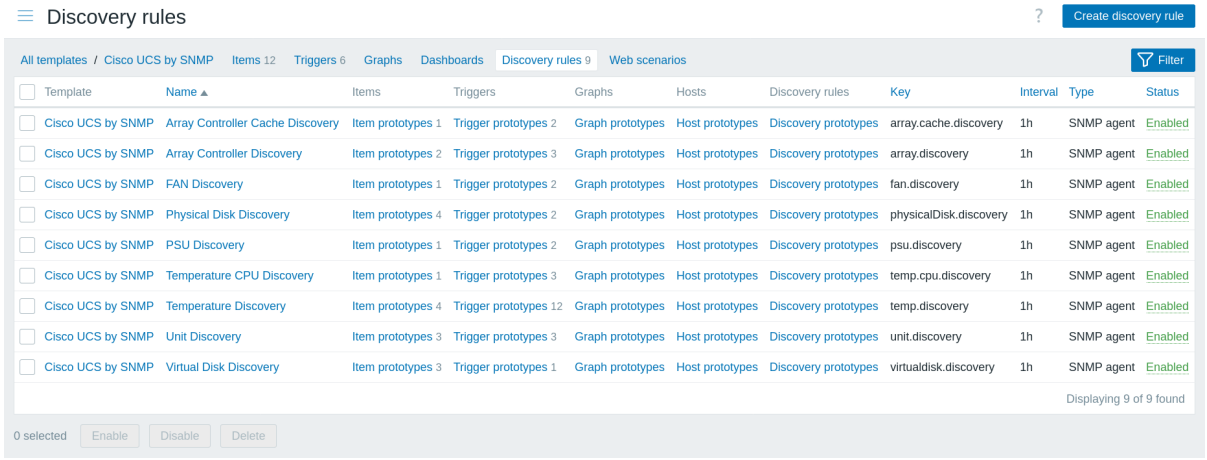
テンプレートグループおよびテンプレートでグラフをフィルタリングできます。検索パフォーマンスを向上させるために、データはマクロが解決されていない状態で検索されます。

ディスカバリルール

概要

テンプレートのローレベルディスカバリルールの一覧には、データ収集 > テンプレートから、対象のテンプレートの ディスカバリをクリックしてアクセスできます。

既存のローレベルディスカバリルールの一覧が表示されます。フィルター設定を変更すると、テンプレートに依存しないすべてのディスカバリルール、または特定のテンプレートグループに属するすべてのディスカバリルールを表示することもできます。



表示されるデータ:

列	説明
テンプレート	テンプレートのディスカバリルールが属するテンプレート。 テンプレート名をクリックすると、テンプレート設定フォームが開きます。
名前	青色のリンクとして表示されるルールの名前。 ルール名をクリックすると、ローレベルディスカバリルール設定フォームが開きます。 ディスカバリルールが別のテンプレートから継承されている場合、ルール名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのディスカバリルール一覧が開きます。
アイテム	アイテムのプロトタイプ一覧へのリンクが表示されます。 既存のアイテムのプロトタイプ数が灰色で表示されます。
トリガー	トリガーのプロトタイプ一覧へのリンクが表示されます。 既存のトリガーのプロトタイプ数が灰色で表示されます。
グラフ	グラフのプロトタイプ一覧へのリンクが表示されます。 既存のグラフのプロトタイプ数が灰色で表示されます。
ホスト	ホストのプロトタイプ一覧へのリンクが表示されます。 既存のホストのプロトタイプ数が灰色で表示されます。
キー	ディスカバリに使用されるアイテムキーが表示されます。
間隔	ディスカバリを実行する頻度が表示されます。
タイプ	ディスカバリに使用されるアイテムのタイプ（Zabbix エージェント、SNMP エージェントなど）が表示されます。
ステータス	ディスカバリルールのステータス（有効または 無効）が表示されます。ステータスをクリックすると、有効から無効へ（またはその逆に）変更できます。

新しいローレベルディスカバリルールを設定するには、右上にある ディスカバリルールの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

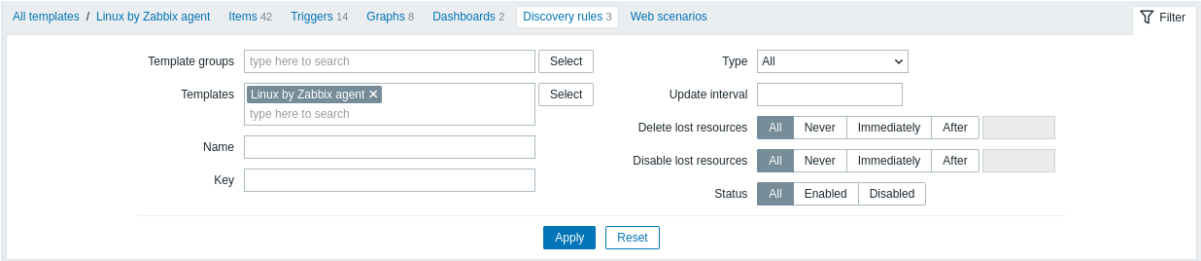
- *Enable* - ローレベルディスカバリルールのステータスを *Enabled* に変更します
- *Disable* - ローレベルディスカバリルールのステータスを *Disabled* に変更します
- *Delete* - ローレベルディスカバリルールを削除します

これらのオプションを使用するには、対象のディスカバリルールの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるディスカバリルールのみを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

右上隅に *Filter* アイコンがあります。これをクリックするとフィルターが開き、テンプレート、ディスカバリルール名、アイテムキー、アイテムタイプなど、必要な絞り込み条件を指定できます。



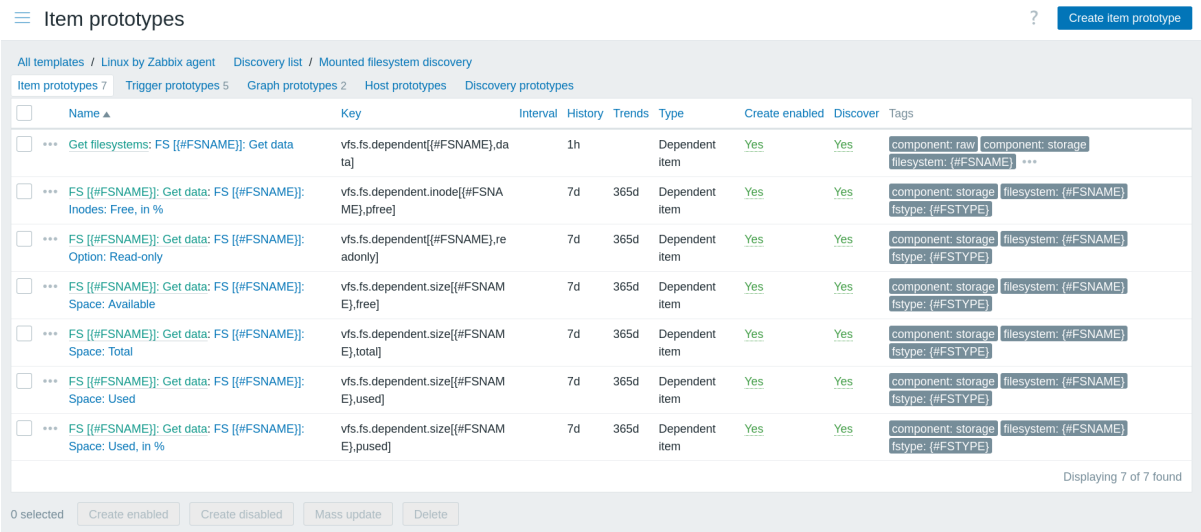
Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループが暗黙的に選択されます。
Templates	1 つ以上のテンプレートで絞り込みます。
Name	ディスカバリルール名で絞り込みます。
Key	ディスカバリアイテムキーで絞り込みます。
Type	ディスカバリアイテムタイプで絞り込みます。
Update interval	更新間隔で絞り込みます。 Zabbix trapper および依存アイテムでは使用できません。
Delete lost resources	*Delete lost resources* の期間で絞り込みます。
Disable lost resources	*Disable lost resources* の期間で絞り込みます。
Status	ディスカバリルールのステータスで絞り込みます (All/Enabled/Disabled)。

アイテムプロトタイプ

概要

このセクションには、テンプレート上のローレベルディスカバリルールに設定されたアイテムのプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ローレベルディスカバリ中に実際のホストのアイテムを作成するための基礎として、アイテムのプロトタイプが使用されます。



表示されるデータ:

列	説明
名前	アイテムのプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、アイテムのプロトタイプの設定フォームが開きます。 アイテムのプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がアイテムのプロトタイプ名の前に灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのアイテムのプロトタイプ一覧が開きます。 アイテムのプロトタイプが「従属アイテム」タイプの場合、マスターアイテムの名前がアイテムのプロトタイプ名の前に緑色で表示されます（ホストレベルのアイテムのプロトタイプの場合と同様です）。
キー	アイテムのプロトタイプのキーが表示されます。

列	説明
間隔	チェックの実行間隔が表示されます。
ヒストリ	アイテムデータのヒストリを保持する日数が表示されます。
トレンド	アイテムのトレンドヒストリを保持する日数が表示されます。
タイプ	アイテムのプロトタイプのタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
作成を有効化	このプロトタイプに基づくアイテムの作成状態が次のように表示されます。 はい - 有効 いいえ - 無効。表示されている「はい」または「いいえ」をクリックして切り替えられます。
ディスカバリ	このプロトタイプに基づくアイテムのディスカバリ状態が次のように表示されます。 はい - ディスカバリする いいえ - ディスカバリしない。表示されている「はい」または「いいえ」をクリックして切り替えられます。
タグ	アイテムのプロトタイプのタグが表示されます。

新しいアイテムのプロトタイプを設定するには、右上隅にあるアイテムのプロトタイプを作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 有効で作成 - これらのアイテムを 有効として作成します
- 無効で作成 - これらのアイテムを 無効として作成します
- 一括更新 - これらのアイテムプロトタイプを一括更新します
- 削除 - これらのアイテムプロトタイプを削除します

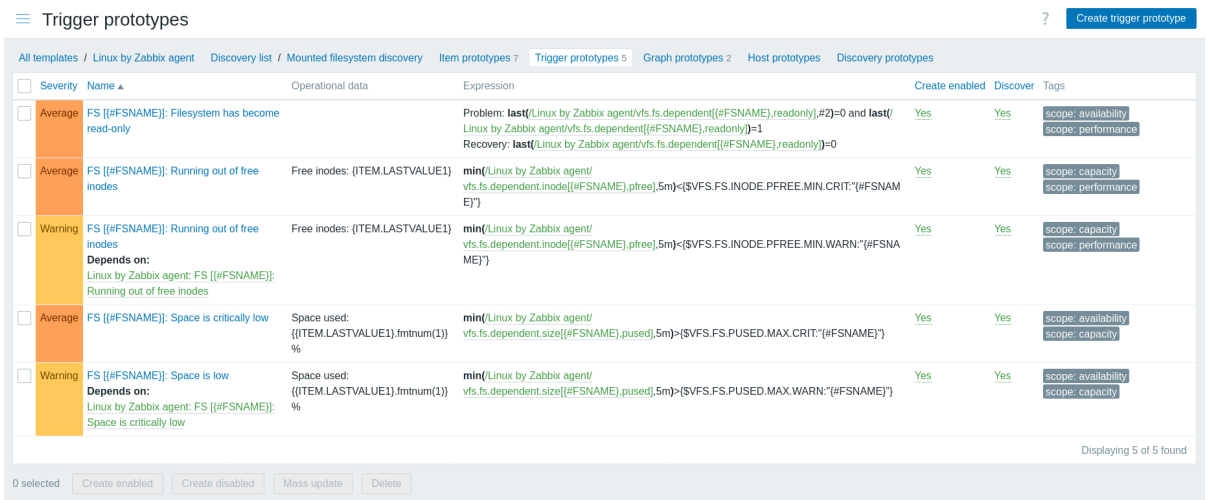
これらのオプションを使用するには、該当するアイテムプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

トリガーのプロトタイプ

概要

このセクションには、テンプレート上のローレベルディスカバリールールに設定されたトリガープロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ローレベルディスカバリー中に実際のホストトリガーを作成するための基礎として、トリガープロトタイプが使用されます。



表示されるデータ:

列	説明
深刻度	トリガーの深刻度が、名前とセルの背景色で表示されます。
名前	トリガープロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、トリガープロトタイプ設定フォームが開きます。 トリガープロトタイプがリンクされたテンプレートに属している場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのトリガープロトタイプ一覧が開きます。

列	説明
条件式	トリガーの条件式が表示されます。条件式のテンプレート-アイテム部分はリンクとして表示され、アイテム設定フォームに移動できます。
運用データ	トリガーの運用データの形式が表示されます。これには、監視 > 障害で動的に解決される任意の文字列とマクロが含まれます。
作成を有効化	このプロトタイプに基づくトリガーの作成状態: はい - 有効 いいえ - 無効。クリックすると「はい」と「いいえ」を切り替えられます。
ディスカバリー	このプロトタイプに基づくトリガーをディスカバリーするかどうか: はい - ディスカバリーする いいえ - ディスカバリーしない。クリックすると「はい」と「いいえ」を切り替えられます。
タグ	トリガープロトタイプのタグが表示されます。

新しいトリガープロトタイプを設定するには、右上隅にあるトリガープロトタイプを作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 有効で作成 - これらのトリガーを 有効として作成します
- 無効で作成 - これらのトリガーを 無効として作成します
- 一括更新 - これらのトリガーのプロトタイプを一括更新します
- 削除 - これらのトリガーのプロトタイプを削除します

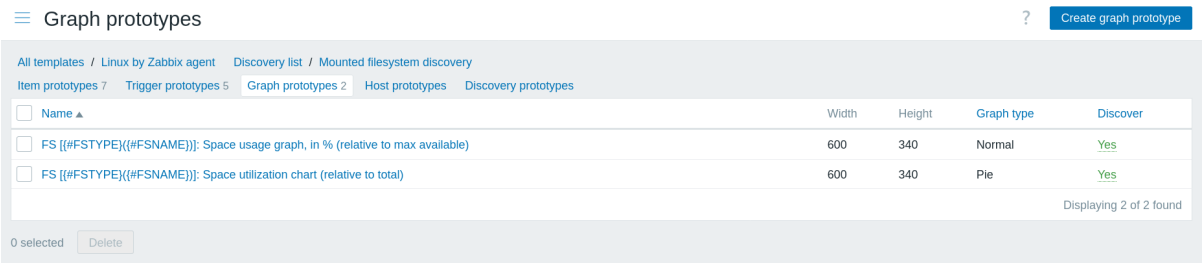
これらのオプションを使用するには、該当するトリガーのプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

グラフのプロトタイプ

概要

このセクションには、テンプレート上のローレベルディスカバリールールに設定されたグラフのプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ローレベルディスカバリー中に実際のホストグラフを作成するための基礎として、グラフのプロトタイプが使用されます。



表示されるデータ:

Column	Description
Name	グラフのプロトタイプ名が青色のリンクとして表示されます。 名前をクリックすると、グラフのプロトタイプ設定フォームが開きます。 グラフのプロトタイプがリンクされたテンプレートに属している場合、グラフ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのグラフのプロトタイプ一覧が開きます。
Width	グラフのプロトタイプの幅が表示されます。
Height	グラフのプロトタイプの高さが表示されます。
Type	グラフのプロトタイプのタイプ（*Normal*、*Stacked*、*Pie*、または *Exploded*）が表示されます。
Discover	このプロトタイプに基づいてグラフをディスカバリーするかどうか: **Yes** - ディスカバリーする **No** - ディスカバリーしない。クリックすると、'Yes' と'No' を切り替えられます。

新しいグラフのプロトタイプを設定するには、右上隅にある グラフのプロトタイプを作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 削除 - これらのグラフプロトタイプを削除します

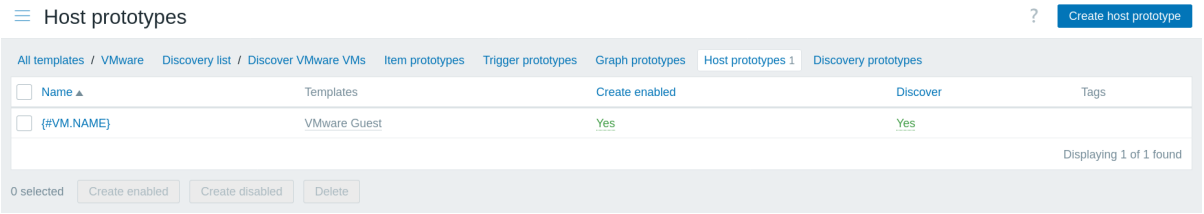
これらのオプションを使用するには、対象のグラフプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

ホストプロトタイプ

概要

このセクションには、テンプレート上のローレベルディスカバリルールに設定されたホストプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ローレベルディスカバリ中に実際のホストを作成するための基礎として、ホストプロトタイプが使用されます。



表示されるデータ:

Column	Description
Name	ホストプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、ホストプロトタイプ設定フォームが開きます。 ホストプロトタイプがリンクされたテンプレートに属している場合、ホスト名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのホストプロトタイプ一覧が開きます。
Templates	ホストプロトタイプのテンプレートが表示されます。
Create enabled	このプロトタイプに基づくホストの作成を次のように設定します: **Yes** - 有効 **No** - 無効。クリックして「Yes」と「No」を切り替えられます。
Discover	このプロトタイプに基づくホストを次のように検出します: **Yes** - 検出する **No** - 検出しない。クリックして「Yes」と「No」を切り替えられます。
Tags	ホストプロトタイプのタグが表示されます。

新しいホストプロトタイプを設定するには、右上隅にあるホストプロトタイプの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 有効で作成 - これらのホストを 有効として作成します
- 無効で作成 - これらのホストを 無効として作成します
- 削除 - これらのホストプロトタイプを削除します

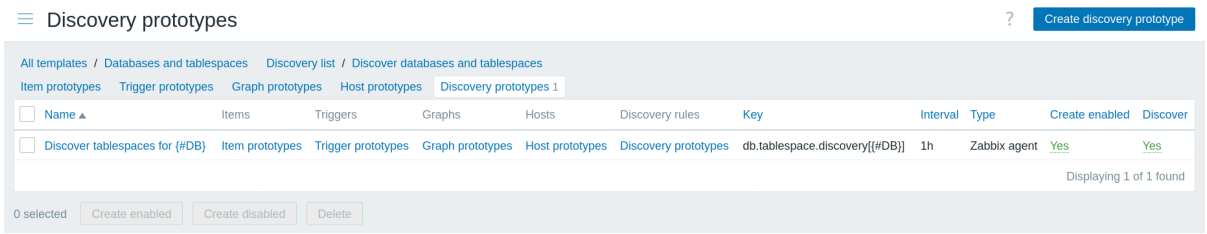
これらのオプションを使用するには、該当するホストプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

ディスカバリープロトタイプ

概要

このセクションには、テンプレート上のローレベルディスカバリルールに設定されているディスカバリプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ディスカバリプロトタイプは、親ディスカバリルールの実際のディスカバリプロトタイプを作成するための基礎になります。



表示されるデータ:

Column	Description
Name	ディスカバリプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、ディスカバリプロトタイプ設定フォームが開きます。 ディスカバリプロトタイプがリンクされたテンプレートに属している場合、ディスカバリプロトタイプ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのディスカバリプロトタイプ一覧が開きます。
Items	アイテムプロトタイプ一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数が灰色で表示されます。
Triggers	トリガープロトタイプ一覧へのリンクが表示されます。 既存のトリガープロトタイプ数が灰色で表示されます。
Graphs	グラフプロトタイプ一覧へのリンクが表示されます。 既存のグラフプロトタイプ数が灰色で表示されます。
Hosts	ホストプロトタイプ一覧へのリンクが表示されます。 既存のホストプロトタイプ数が灰色で表示されます。
Discovery rules	ディスカバリプロトタイプ一覧へのリンクが表示されます。 既存のディスカバリプロトタイプ数が灰色で表示されます。
Key	ディスカバリに使用されるアイテムキーが表示されます。
Interval	ディスカバリの実行頻度が表示されます。
Type	ディスカバリに使用されるアイテムのタイプ（Zabbix エージェント、SNMP エージェントなど）が表示されます。
Create enabled	ディスカバリプロトタイプの作成ステータス（作成有効（*Yes*）または作成無効（*No*））が表示されます。ステータスをクリックすると、Yes から No（またはその逆）に変更できます。
Discover	ディスカバリプロトタイプのディスカバリステータス（ディスカバリ実行（*Yes*）またはディスカバリを実行しない（*No*））が表示されます。ステータスをクリックすると、Yes から No（またはその逆）に変更できます。

新しいディスカバリプロトタイプを設定するには、右上隅にある *Create discovery prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 有効で作成 - これらのディスカバリプロトタイプを 有効として作成します
- 無効で作成 - これらのディスカバリプロトタイプを 無効として作成します
- 削除 - これらのディスカバリプロトタイプを削除します

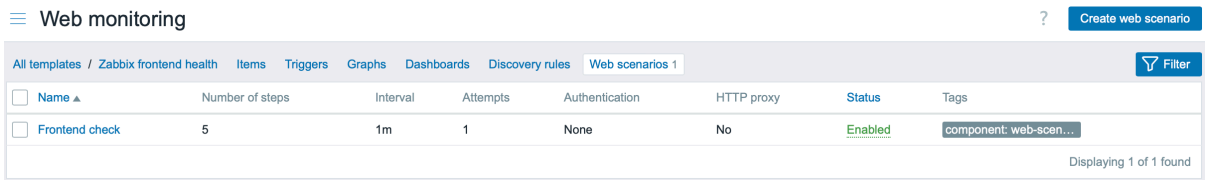
これらのオプションを使用するには、該当するディスカバリプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

Web シナリオ

概要

テンプレートのWeb シナリオ リストには、データ収集 > テンプレートから対象のテンプレートの *Web* をクリックしてアクセスできます。

既存の Web シナリオのリストが表示されます。



表示されるデータ:

列	説明
名前	Web シナリオの名前。Web シナリオ名をクリックすると、Web シナリオ設定フォームが開きます。 Web シナリオが別のテンプレートから継承されている場合、Web シナリオ名の前にテンプレート名がグレーのリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルの Web シナリオリストが開きます。
ステップ数	シナリオに含まれるステップの数。
更新間隔	シナリオを実行する頻度。
試行回数	Web シナリオのステップを実行する試行回数。
認証	認証方式（Basic、NTLM、Kerberos、Digest、またはなし）が表示されます。
HTTP プロキシ	HTTP プロキシを使用している場合はそのプロキシが表示され、使用していない場合は「いいえ」と表示されます。
ステータス	Web シナリオのステータス（有効または無効）が表示されます。 ステータスをクリックすると変更できます。
タグ	Web シナリオのタグが表示されます。 最大 3 つのタグ（名前: 値のペア）を表示できます。タグが 4 つ以上ある場合は「...」リンクが表示され、マウスオーバーで全タグを確認できます。

新しい Web シナリオを設定するには、右上隅にある *Web* シナリオの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

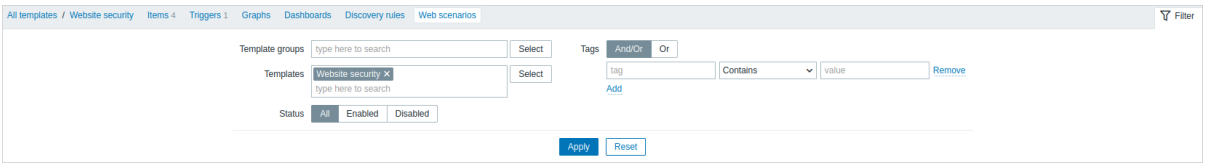
- 有効化 - シナリオのステータスを 有効に変更します
- 無効化 - シナリオのステータスを 無効に変更します
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、対象の Web シナリオの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるシナリオだけを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

Filter リンクは、Web シナリオの一覧の上にあります。これをクリックするとフィルターが表示され、テンプレートグループ、テンプレート、ステータス、タグでシナリオを絞り込めます。

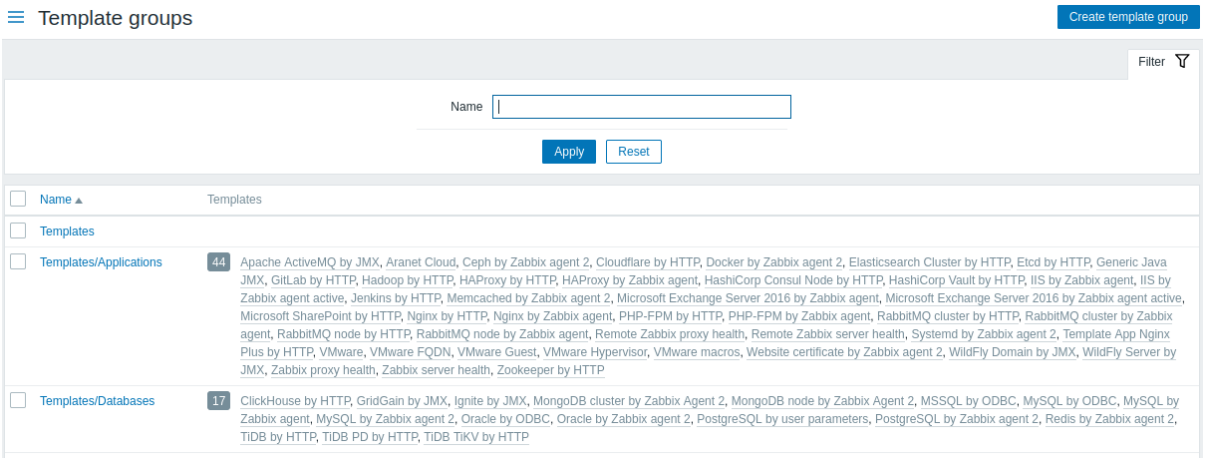


テンプレートグループ

概要

データ収集 > テンプレートグループセクションでは、テンプレートグループを設定および管理できます。

既存のテンプレートグループとその詳細が一覧表示されます。名前でテンプレートグループを検索およびフィルターできます。



表示されるデータ:

列	説明
名前	テンプレートグループの名前。グループ名をクリックすると、テンプレートグループ設定フォームが開きます。
テンプレート	グループ内のテンプレート数（灰色で表示）と、それに続くグループメンバーの一覧。 テンプレート名をクリックすると、テンプレート設定フォームが開きます。 数値をクリックすると、このグループ内のテンプレート一覧が開きます。

一括編集オプション

複数のテンプレートグループを一度に削除するには、各グループの前にあるチェックボックスをオンにしてから、一覧の下にある「削除」ボタンをクリックします。

フィルターの使用

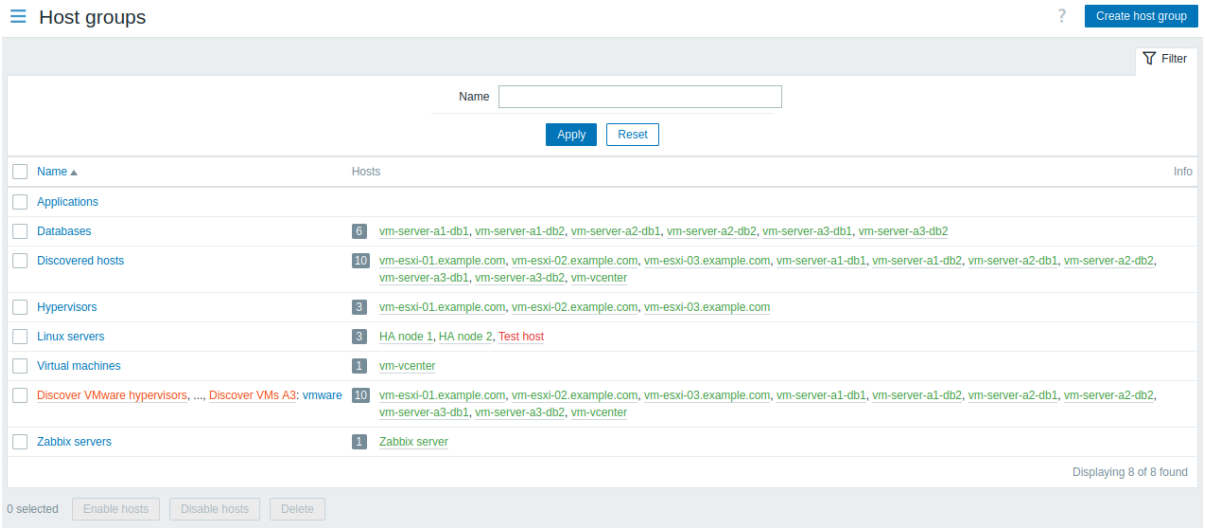
フィルターを使用して、関心のあるテンプレートグループのみを表示できます。検索パフォーマンスを向上させるために、データはマクロを展開せずに検索されます。

ホストグループ

概要

データ収集 > ホストグループセクションでは、ホストグループを設定および管理できます。

既存のホストグループとその詳細が一覧表示されます。名前でホストグループを検索およびフィルターできます。



表示されるデータ:

列	説明
名前	ホストグループの名前。グループ名をクリックすると、ホストグループ設定フォームが開きます。 ディスカバリされたホストグループには、プレフィックスとしてローレベルディスカバリルール名が表示されます。LLD ルール名をクリックすると、ホストプロトタイプ設定フォームが開きます。 ディスカバリを実行したすべての LLD ルールが削除されると、ディスカバリされたホストグループも削除されることに注意してください。
ホスト	グループ内のホスト数（灰色で表示）と、それに続くグループメンバーの一覧。 ホスト名をクリックすると、ホスト設定フォームが開きます。 ホスト数をクリックすると、ホストの一覧全体から、そのグループに属するホストだけがフィルターされます。
情報	ホストグループに関するエラー情報（存在する場合）が表示されます。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- ホストを有効化 - グループ内のすべてのホストのステータスを” 監視中” に変更します
- ホストを無効化 - グループ内のすべてのホストのステータスを” 監視対象外” に変更します
- 削除 - ホストグループを削除します

これらのオプションを使用するには、対象のホストグループの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルタの使用

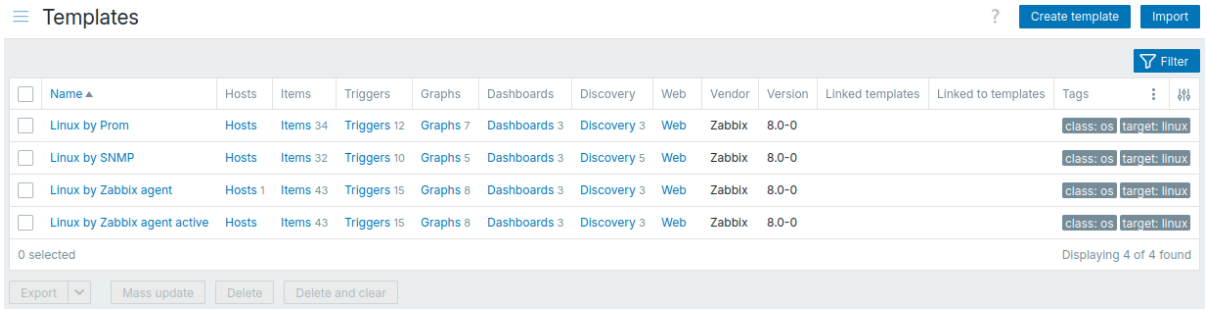
フィルタを使用して、任意のホストグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

テンプレート

概要

データ収集 > テンプレートセクションでは、テンプレートを設定および管理できます。

既存のテンプレートとその詳細が一覧表示されます。



テンプレートの詳細は、カスタマイズ可能なテーブルに表示されます。テーブルヘッダーの テーブルをカスタマイズボタンを使用すると、列の表示、非表示、並べ替え、およびレイアウトのデフォルトへのリセットができます。各列の右端をドラッグして列の幅を変更することもできます。境界線をダブルクリックすると、幅がリセットされます。すべてのカスタマイズは直ちに適用され、現在のユーザーに対して常に記憶されます。

使用可能な列は次のとおりです。

列	説明
名前	テンプレートの名前。 テンプレート名をクリックすると、テンプレート設定フォームが開きます。
ホスト	テンプレートにリンクされている編集可能なホストの数。読み取り専用のホストは含まれません。 ホストをクリックすると、テンプレートにリンクされているホストだけがフィルターされたホスト一覧が開きます。
エンティティ（アイテム、トリガー、グラフ、ダッシュボード、ディスカバリ、*Web*）	テンプレート内の該当するエンティティの数（グレーで表示）。 エンティティ名をクリックすると、そのエンティティの一覧全体で、テンプレートに属するものだけがフィルターされます。
リンクされたテンプレート	テンプレートにリンクされているテンプレート。

列	説明
リンク先のテンプレート	このテンプレートがリンクされているテンプレート。
ベンダー、バージョン	テンプレートのベンダーとバージョン。テンプレート設定にこの情報が含まれている場合に表示され、標準搭載テンプレート、インポートされたテンプレート、またはTemplate APIを使用して変更されたテンプレートに対してのみ表示されます。 標準搭載テンプレートの場合、バージョンは次の形式で表示されます。Zabbix のメジャーバージョン、区切り文字（”-”）、リビジョン番号（テンプレートの新しいバージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされます）。例：7.0-0、7.0-3、8.0-0、8.0-3。
タグ	テンプレートのタグが、マクロを解決せずに表示されます。 表示されたタグの後にある 3 点アイコンにカーソルを合わせるかクリックすると、すべてのタグを含むポップアップが開きます。 列ヘッダーの オプションボタンをクリックすると、表示するタグ数（1～3）の設定、タグ名の表示方法（完全、短縮、または非表示）の選択、タグ表示の優先度の設定、または列の複製ができます（複製した列は名前の変更または削除が可能です）。
タグの値	テンプレートのタグの値が、マクロを解決せずに表示されます。 列ヘッダーの オプションボタンをクリックすると、列名の変更、表示するタグの値の選択、または列の複製ができます（複製した列は名前の変更または削除が可能です）。
カスタムテキスト	カスタムテキスト、マクロの値、またはその両方の組み合わせが表示されます。 列ヘッダーの オプションボタンをクリックすると、列名の変更、カスタムテキストまたはマクロの入力、または列の複製ができます（複製した列は名前の変更または削除が可能です）。 サポートされているマクロ：{HOST.ID}、{HOST.HOST}、{HOST.NAME}、{HOST.DESCRIPTION}、およびユーザーマクロ。

新しいテンプレートを設定するには、右上隅にある テンプレートを作成ボタンをクリックします。

YAML、XML、または JSON ファイルからテンプレートをインポートするには、右上隅にある インポートボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

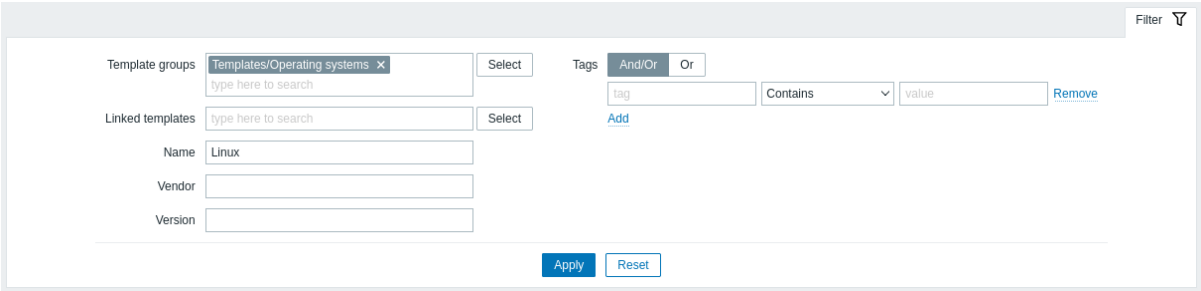
- エクスポート - テンプレートを YAML、XML、または JSON ファイルにエクスポートします。
- 一括更新 - 複数のテンプレートに対して、複数のプロパティを更新します。
- 削除 - テンプレートを削除しますが、リンクされたエンティティ（アイテム、トリガーなど）はホストに残します。
- 削除してクリア - テンプレートと、そのリンクされたエンティティをホストから削除します。

これらのオプションを使用するには、対象のテンプレートの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるテンプレートのみを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

フィルターリンクは、テンプレートの作成ボタンと インポートボタンの下にあります。これをクリックするとフィルターが利用可能になり、テンプレートグループ、リンクされたテンプレート、名前、およびタグでテンプレートを絞り込めます。



Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループが暗黙的に選択されます。
Linked templates	直接リンクされたテンプレートで絞り込みます。
Name	テンプレート名で絞り込みます。
Vendor	テンプレートのベンダーで絞り込みます。
Version	テンプレートのバージョンで絞り込みます。

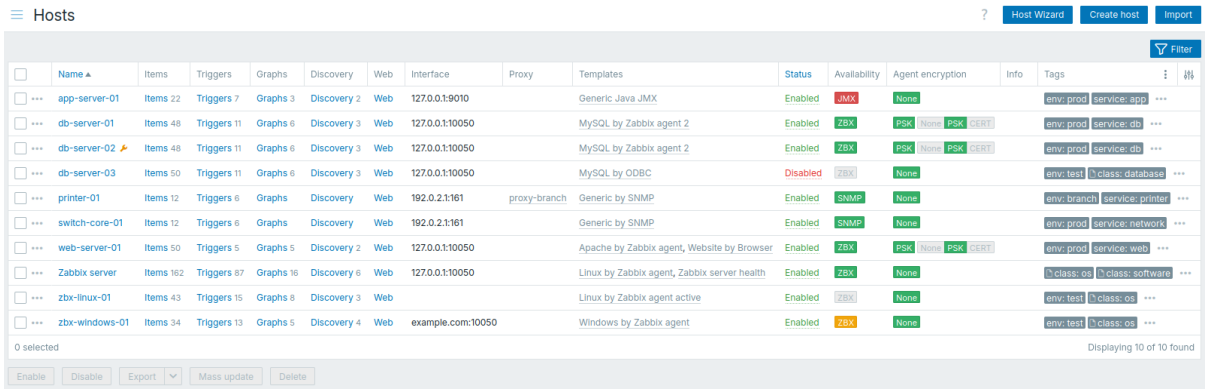
Parameter	Description
Tags	テンプレートのタグ名と値で絞り込みます。 絞り込みはテンプレートレベルのタグのみ可能です（継承されたタグは対象外）。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件には、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - 入力した文字列を含むタグ値を持つ指定したタグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - 入力した文字列を含むタグ値を持つ指定したタグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

ホスト

概要

データ収集 > ホストセクションでは、ホストを設定および管理できます。

既存のホストとその詳細が一覧表示されます。



ホストの詳細は、カスタマイズ可能なテーブルに表示されます。テーブルヘッダーの テーブルをカスタマイズボタンを使用すると、列の表示、非表示、並べ替え、およびレイアウトのデフォルトへのリセットができます。各列の右端をドラッグして列幅を変更することもできます。境界線をダブルクリックすると、幅がリセットされます。すべてのカスタマイズはすぐに適用され、現在のユーザーに対して常に記憶されます。

使用可能な列は次のとおりです。

列	説明
名前	ホストの名前です。 ホスト名をクリックすると、ホスト設定フォームが開きます。 ホスト名の後にオレンジ色のレンチアイコン が表示されている場合、そのホストはメンテナンス中です。アイコン上にマウスポインターを置くと、メンテナンスの詳細が表示されます。
エンティティ（アイテム、トリガー、グラフ、ディスカバリ、*Web*）	エンティティ名をクリックすると、そのホストのアイテム、トリガーなどが表示されます。 各エンティティの数は灰色で表示されます。
インターフェース	ホストのメインインターフェースが表示されます。

列	説明
プロキシ	割り当てられたプロキシがこの列に表示されます。 < プロキシ名 > - ホストがスタンドアロンプロキシによって監視されています（プロキシがプロキシグループの一部である場合も同様）。 < プロキシグループ名: プロキシ名 > - ホストがプロキシグループによって監視され、Zabbix サーバーがホストを監視するプロキシを割り当てています。 < プロキシグループ名 > - ホストがプロキシのないプロキシグループによって監視されているか、Zabbix サーバーがホストを監視するプロキシを割り当てていません。 何も表示されない場合 - ホストはプロキシまたはプロキシグループのいずれによっても監視されていません。 監視元フィルターオプションが「任意」、「プロキシ」、または「プロキシグループ」に設定されている場合にのみ、この列が表示されます。
テンプレート	ホストにリンクされているテンプレートが表示されます。 リンクされたテンプレートに他のテンプレートが含まれている場合、それらは括弧内にカンマ区切りで表示されます。 テンプレート名をクリックすると、その設定フォームが開きます。
ステータス	ホストのステータス（有効または 無効）が表示されます。 ステータスをクリックすると、手動で変更できます。 失われたディスカバリ済みホストには情報アイコンが表示されます。ツールチップにステータスの詳細が表示されます。

列	説明
可用性	設定されたインターフェースごとのホストの可用性が表示されます。 可用性アイコンは、Zabbix サーバー上のホストインターフェースの現在のステータスを表します。そのため、Web インターフェースでホストを無効にした場合、Zabbix サーバーが設定変更を同期した後に可用性が更新されます。同様に、ホストを有効にした場合、Zabbix サーバーが設定変更を同期し、ホストをポーリングした後に可用性が更新されます。 可用性アイコンは、設定されているインターフェースタイプ（Agent、SNMP、IPMI、JMX）のみを表します。 アイコン上にマウスポインターを置くと、同じタイプのすべてのインターフェースについて、詳細、ステータス、エラーを含む一覧がポップアップ表示されます。Agent インターフェースの場合、ポップアップにはインターフェース（パッシブ）とアクティブチェックが表示されます。アクティブチェックのみを使用するホストでは、Agent インターフェースが設定されていなくても Agent インターフェースアイコンが表示されます。 インターフェースがないホストでは、この列は空になります。 単一のホストインターフェースのステータスは、そのインターフェースを使用する有効なアイテムとホスト間の接続によって決まります。ステータスは次のいずれかです。 使用可能 - ホストへの接続に成功しました。 使用不可 - ホストへの接続に失敗しました（タイムアウト、ファイアウォールの問題など）。 不明 - ホストへの接続が試行されていないか、結果が不明です。 Zabbix サーバーがインターフェースのステータスを判定する方法の詳細については、不明なインターフェースステータスおよび到達不能/使用不可なホストインターフェースの設定を参照してください。 単一タイプ（Agent、SNMP、IPMI、JMX）のすべてのホストインターフェースのステータスは、少なくとも 1 つの有効なアイテムで使用されているインターフェースによって決まります。ステータスはアイコンの色で示されます。 緑 - すべてのインターフェースが使用可能です。 黄色 - 少なくとも 1 つのインターフェースが使用不可で、少なくとも 1 つが使用可能または不明です。 赤 - すべてのインターフェースが使用不可です。 灰色 - 少なくとも 1 つのインターフェースが不明ですが、使用不可のインターフェースはありません。 アクティブチェックの可用性。ホストで少なくとも 1 つのアクティブチェックが有効になっている場合、上記の説明に従い、アクティブチェックも Agent インターフェース全体の可用性に影響します。アクティブチェックの可用性を判定するため、エージェントのアクティブチェック処理でハートビートメッセージが送信されます。ハートビートメッセージの頻度は、Zabbix エージェントまたはエージェント 2の設定にある HeartbeatFrequency パラメータで制御されます（デフォルトは 60 秒、範囲は 0～3600）。アクティブチェックのハートビートが 2 x HeartbeatFrequency 秒より古い場合、アクティブチェックは使用不可とみなされます。 注**:** バージョン 6.2.x より前の Zabbix エージェントはアクティブチェックのハートビートを送信しないため、それらのホストの可用性は不明のままです。
エージェントの暗号化	ホストへの接続およびホストからの接続に対する暗号化ステータスが表示されます。 なし - 暗号化なし。 **PSK** - 事前共有鍵を使用。 証明書 - 証明書を使用。
情報	ホストに関するエラー情報（存在する場合）が表示されます。
タグ	ホストのタグが、マクロを解決せずに表示されます。 表示されたタグの後にある 3 点アイコンにマウスポインターを置くかクリックすると、すべてのタグを含むポップアップが開きます。 列ヘッダーの オプションボタンをクリックすると、表示するタグ数（1～3）の設定、タグ名の表示方法（完全、短縮、または非表示）の選択、タグ表示の優先度の設定、または列の複製ができます(複製した列は名前の変更または削除が可能です)。
タグの値	ホストのタグの値が、マクロを解決せずに表示されます。 列ヘッダーの オプションボタンをクリックすると、列名の変更、表示するタグの値の選択、または列の複製ができます（複製した列は名前の変更または削除が可能です）。

列	説明
カスタムテキスト	カスタムテキスト、マクロの値、またはその両方の組み合わせが表示されます。 列ヘッダーの ⋮ オプションボタンをクリックすると、列名の変更、カスタムテキストまたはマクロの入力、または列の複製ができます（複製した列は名前の変更または削除が可能です）。 サポートされているマクロ: {HOST.*}（{HOST.METADATA} を除く）、{INVENTORY.*}、およびユーザーマクロ。

ホストメニューを開くには、ホスト名の前にある ⋮ オプションボタンをクリックします。

新しいホストを設定するには、右上隅にある ホストウィザードまたは ホストの作成ボタンをクリックします。

YAML、XML、または JSON ファイルからホストをインポートするには、右上隅にある インポートボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

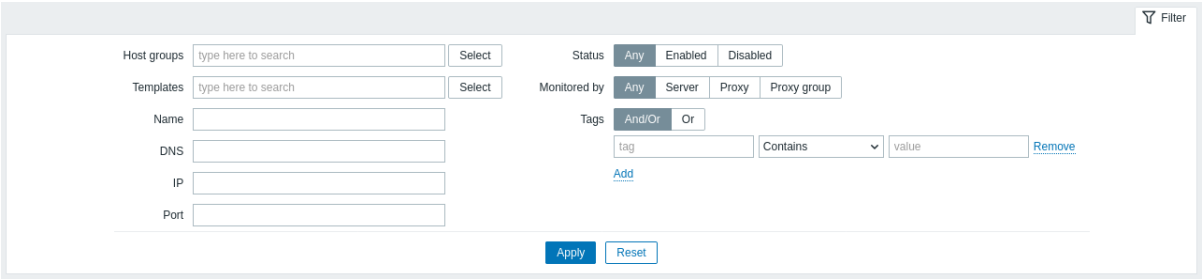
- *Enable* - ホストのステータスを *Monitored* に変更します。
- *Disable* - ホストのステータスを *Not monitored* に変更します。
- *Export* - ホストを YAML、XML、または JSON ファイルにエクスポートします。
- *Mass update* - 複数のホストに対して、複数のプロパティを更新します。
- *Delete* - ホストを削除します。

これらのオプションを使用するには、対象のホストの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるホストのみを表示できます。検索性能を向上させるため、データはマクロが未展開の状態で検索されます。

Filter アイコンは右上隅にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



Parameter	Description
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、ネストされたすべての子ホストグループが暗黙的に選択されます。
Templates	リンクされたテンプレートで絞り込みます。
Name	表示名で絞り込みます。
DNS	DNS 名で絞り込みます。
IP	IP アドレスで絞り込みます。
Port	ポート番号で絞り込みます。
Status	ホストのステータスで絞り込みます。
Monitored by	Zabbix サーバー、プロキシ、またはプロキシグループによって監視されているホストを絞り込みます。
Proxies	ここで指定したプロキシによって監視されているホストを絞り込みます。このフィールドは、*Monitored by* フィールドで”Proxy” が選択されている場合にのみ使用できます。
Proxy groups	ここで指定したプロキシグループによって監視されているホストを絞り込みます。このフィールドは、*Monitored by* フィールドで”Proxy group” が選択されている場合にのみ使用できます。

Parameter	Description
Tags	ホストタグの名前と値で絞り込みます。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです。 **Exists** - 指定したタグ名を含める; **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別）; **Contains** - 入力した文字列をタグ値に含む指定タグ名を含める（部分一致、大文字と小文字を区別しない）; **Does not exist** - 指定したタグ名を除外する; **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別）; **Does not contain** - 入力した文字列をタグ値に含む指定タグ名を除外する（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

不明なインターフェース状態

Zabbix サーバーは、次のケースでホストインターフェース（Agent、SNMP、IPMI、JMX）に対して「Unknown」状態を表示します。

- ホストが無効になっている。
- ホストがプロキシ、別のプロキシ、または以前プロキシで監視されていた場合はサーバーによる監視対象に設定されている。
- ホストが、オフラインのように見えるプロキシによって監視されている（プロキシからの更新が最大ハートビート間隔 - 1 時間の間受信されていない）。
- そのインターフェース種別のホストアイテムがすべて無効になっている。
- そのインターフェース種別のポーラーが設定されていない（たとえば、StartAgentPollers またはStartSNMPPollers サーバー設定パラメーターが 0 に設定されている）。

インターフェースの可用性は、Zabbix サーバーの設定キャッシュの同期後に「Unknown」に設定されます。

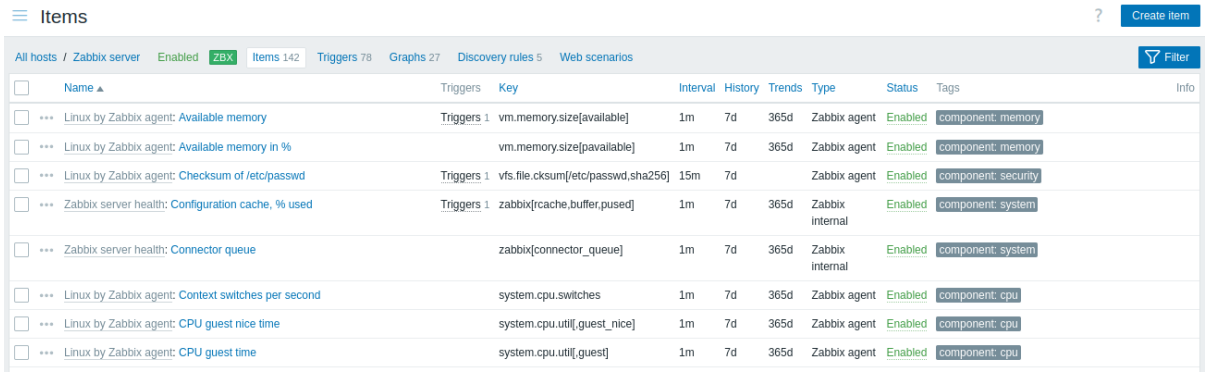
プロキシで監視されているホストのインターフェースの可用性（Available/Not available）は、プロキシの設定キャッシュの同期後に復元されます。

アイテム

概要

ホストのアイテム一覧には、データ収集 > ホストから対象のホストの アイテムをクリックしてアクセスできます。

既存のアイテム一覧が表示されます。



表示されるデータ:

列	説明
アイテムメニュー	3 点アイコンをクリックすると、アイテムメニューが開きます。

列	説明
ホスト	アイテムのホスト。 ホスト名をクリックすると、ホスト設定フォームが開きます。 この列は、フィルターで複数のホストまたはホストが選択されていない場合にのみ表示されます。
名前	アイテムの名前が青色のリンクとして表示されます。 アイテム名のリンクをクリックすると、アイテム設定フォームが開きます。 ホストのアイテムがテンプレートに属している場合、テンプレート名がアイテム名の前に灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのアイテム一覧が開きます。 アイテムがアイテムプロトタイプから作成された場合、その名前の前に低レベルディスカバリルール名がオレンジ色で表示されます。ディスカバリルール名をクリックすると、アイテムプロトタイプ一覧が開きます。
トリガー	トリガーにマウスを合わせると、アイテムに関連付けられたトリガーを表示するインフォボックスが表示されます。 トリガーの数は灰色で表示されます。
キー	アイテムキーが表示されます。
間隔	チェックの頻度が表示されます。 注パッシブアイテムは、今すぐ実行ボタンを押して即時にチェックすることもできます。
ヒストリ	アイテムデータのヒストリを保持する日数が表示されます。
トレンド	アイテムのトレンド履歴を保持する日数が表示されます。
タイプ	アイテムのタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
ステータス	アイテムのステータス（有効、無効、サポートされていない）が表示されます。ステータスをクリックすると、手動で変更できます。有効から 無効（およびその逆）、サポートされていないから 無効（およびその逆）に変更できます。 失われたディスカバリ済みアイテムには情報アイコンが表示されます。ツールチップにステータスの詳細が表示されます。
タグ	アイテムのタグが表示されます。 最大 3 つのタグ（名前: 値のペア）を表示できます。タグがそれ以上ある場合は、マウスを合わせるとすべてのタグを表示できる「...」リンクが表示されます。
情報	アイテムが正常に動作している場合、この列にはアイコンは表示されません。エラーが発生した場合は、文字「i」が表示された四角形のアイコンが表示されます。アイコンにマウスを合わせると、エラーの説明を含むツールチップが表示されます。

新しいアイテムを設定するには、右上隅にある アイテムの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

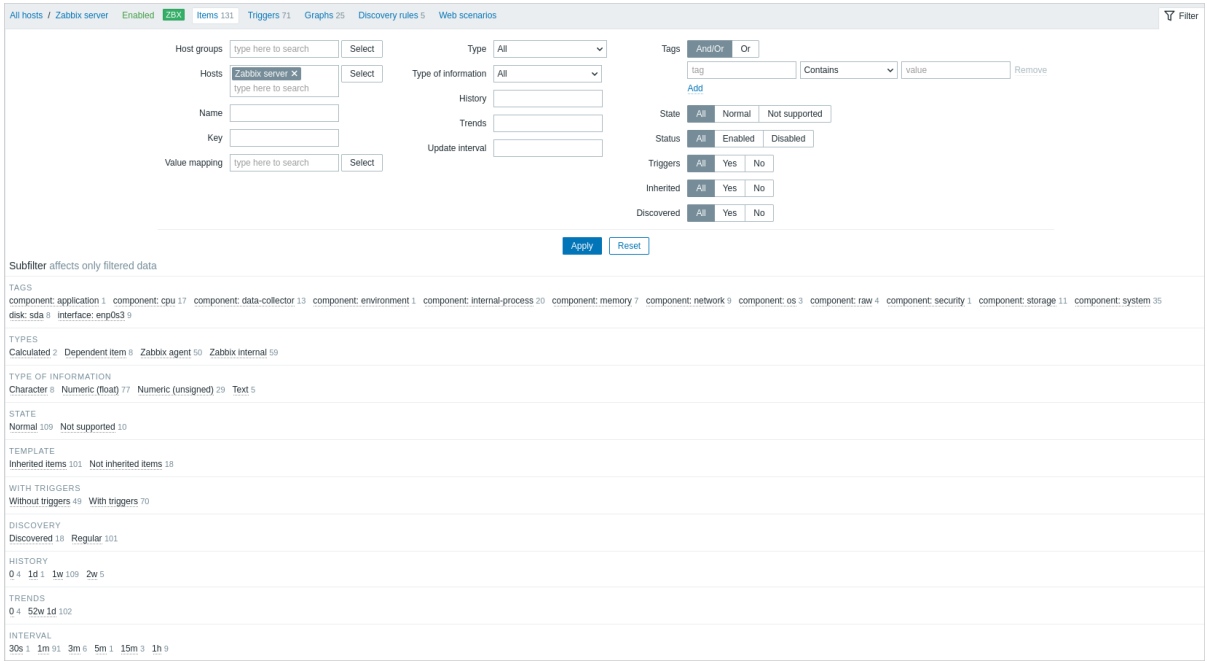
- 有効化 - アイテムのステータスを 有効に変更します
- 無効化 - アイテムのステータスを 無効に変更します
- 今すぐ実行 - 新しいアイテム値のチェックを直ちに実行します。受動的チェックでのみサポートされます（詳細 を参照）。値を直ちにチェックしても設定キャッシュは更新されないため、値にはアイテム設定のごく最近の変更が反映されません。
- 履歴とトレンドを削除 - アイテムの履歴データとトレンドデータを削除します。
- コピー - アイテムを他のホストまたはテンプレートにコピーします。
- 一括更新 - 複数のアイテムの複数のプロパティを更新します。
- 削除 - アイテムを削除します。

これらのオプションを使用するには、対象のアイテムの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるアイテムだけを表示できます。検索性能を向上させるため、データはマクロが未展開の状態で検索されます。

右上に *Filter* アイコンがあります。これをクリックすると、必要なフィルタリング条件を指定できるフィルターが開きます。

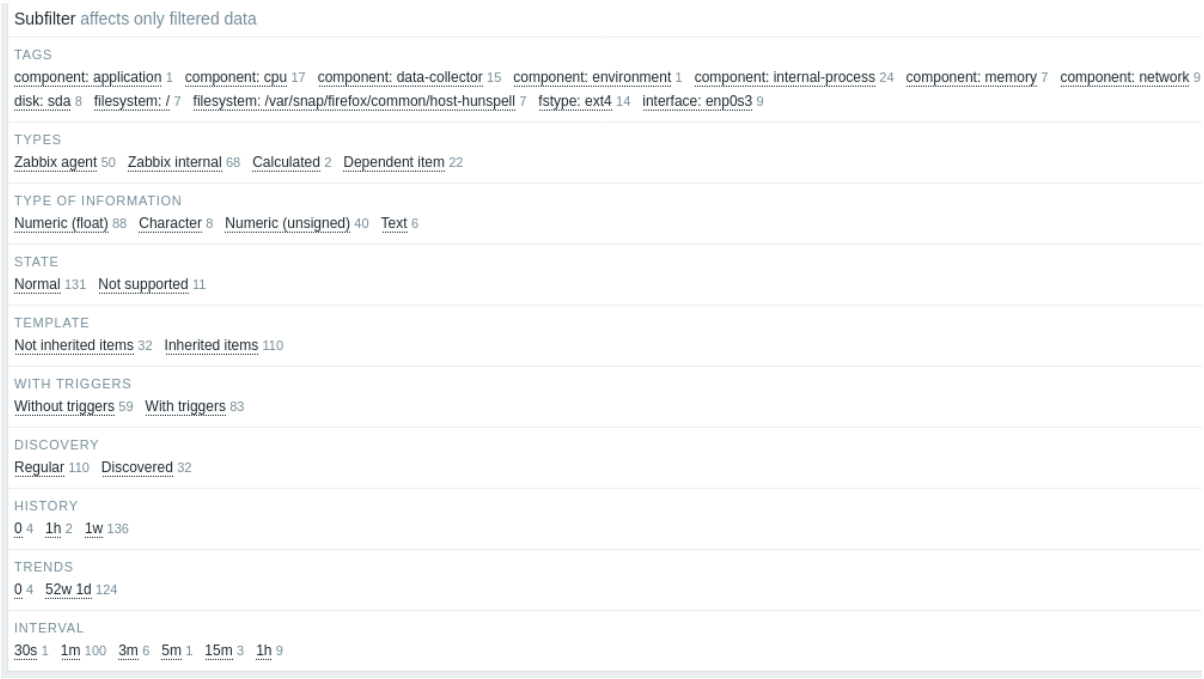


Parameter	Description
Host groups	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、配下のすべてのネストされたホストグループが暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
Hosts	1 つ以上のホストでフィルターします。
Name	アイテム名でフィルターします。
Key	アイテムキーでフィルターします。
Value mapping	使用する値マップでフィルターします。 このパラメータは、*Hosts* オプションが空の場合は表示されません。
Type	アイテムタイプ（Zabbix エージェント、SNMP エージェントなど）でフィルターします。
Type of information	情報の種類（符号なし数値、浮動小数点数など）でフィルターします。
History	アイテム履歴の保存期間でフィルターします。
Trends	アイテムトレンドの保存期間でフィルターします。
Update interval	アイテムの更新間隔でフィルターします。
Tags	表示するアイテム数を制限するためにタグを指定します。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件には次の演算子を使用できます: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別） **Contains** - タグ値に入力した文字列を含む指定タグ名を含める（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外する（部分一致、大文字と小文字を区別しない） 条件の計算方法には 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
State	アイテムの状態でフィルターします - *Normal* または *Not supported*。
Status	アイテムのステータスでフィルターします - *Enabled* または *Disabled*。
Triggers	トリガーがある（またはない）アイテムでフィルターします。
Inherited	テンプレートから継承された（または継承されていない）アイテムでフィルターします。
Discovery	低レベルディスカバリによって発見された（または発見されていない）アイテムでフィルターします。

サブフィルターの使用

サブフィルターは、メインフィルターで適用されたフィルタリングをさらに変更します。

関連するアイテムにすばやくアクセスするための、クリック可能なリンクが含まれています。アイテムは、共通のエンティティ（タグ、アイテムのタイプ、アイテムの状態、アイテムの更新間隔など）によって関連付けられています。エンティティをクリックすると、そのエンティティが灰色の背景で強調表示され、アイテムがすぐにフィルタリングされます（メインフィルターで 適用をクリックする必要はありません）。別のエンティティをクリックすると、フィルタリングされた結果に追加されます。エンティティをもう一度クリックすると、そのフィルタリングが解除されます。



サブフィルターは、フィルタリングされたデータに基づいて生成されます。フィルタリングされたデータは、最大 1000 レコードに制限されます。サブフィルターにさらに多くのレコードを表示するには、検索およびフィルター結果の上限パラメータの値を増やす必要があります（管理 > 一般 > *GUI*）。

メインフィルターとは異なり、サブフィルターはテーブルを更新するたびに更新されます。これにより、利用可能なフィルタリングオプションとその件数が常に最新の状態になります。

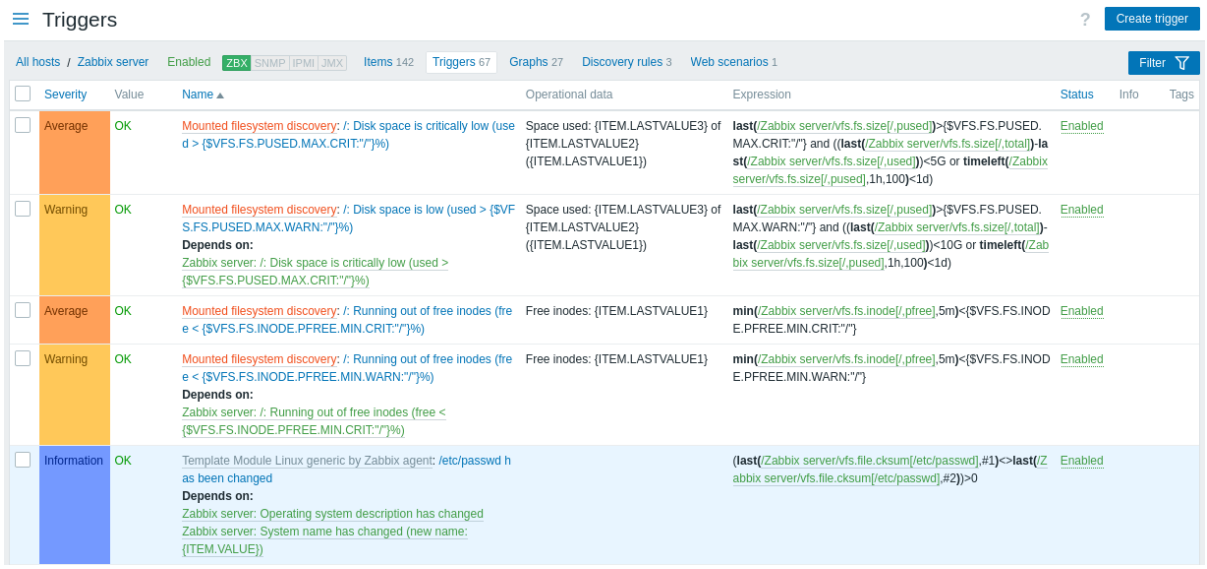
表示されるエンティティ数は、水平方向に最大 100 個に制限されています。それを超える場合は、末尾に 3 点アイコンが表示されます。このアイコンはクリックできません。

クリック可能な各エンティティの横に表示される数字は、そのエンティティにグループ化されているアイテムの数を示します（メインフィルターの結果に基づきます）。エンティティをクリックすると、その他の利用可能なエンティティの横に、プラス記号付きの数字が表示されます。この数字は、現在の選択に追加できるアイテムの数を示します。アイテムがないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

トリガー

概要

ホストのトリガー一覧には、データ収集 > ホストから、対象のホストの トリガーをクリックしてアクセスできます。



表示されるデータ:

列	説明
深刻度	トリガーの深刻度が、名前とセルの背景色で表示されます。
値	トリガーの状態が表示されます: **OK** - OK 状態 **PROBLEM** - 障害状態
ホスト	トリガーのホスト。 ホスト名をクリックすると、ホスト設定フォームが開きます。 フィルターで複数のホストまたはホストが選択されていない場合にのみ、この列が表示されます。
名前	トリガーの名前。トリガーの詳細への青色のリンクとして表示されます。 トリガー名のリンクをクリックすると、トリガー設定フォームが開きます。 ホストのトリガーがテンプレートに属している場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのトリガー一覧が開きます。 トリガーがトリガープロトタイプから作成されている場合、その名前の前にローレベルディスカバリールール名がオレンジ色で表示されます。ディスカバリールール名をクリックすると、トリガープロトタイプの一覧が開きます。
運用データ	トリガーの運用データ定義。任意の文字列と、監視 > 障害で動的に解決されるマクロが含まれます。
条件式	トリガーの条件式が表示されます。条件式のホスト-アイテム部分は、アイテム設定フォームへのリンクとして表示されます。
ステータス	トリガーのステータス（有効、無効、または不明）が表示されます。ステータスをクリックすると、手動で変更できます。有効から無効（またはその逆）、不明から無効（またはその逆）に変更できます。 無効なトリガーの障害は Web インターフェースに表示されなくなりますが、削除はされません。 失われたディスカバリートリガーには情報アイコンが表示されます。ツールチップにステータスの詳細が表示されます。
情報	すべて正常に動作している場合、この列にはアイコンは表示されません。エラーが発生した場合は、文字「i」が表示された四角形のアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明がツールチップに表示されます。
タグ	トリガーにタグが含まれている場合、この列にタグ名と値が表示されます。

新しいトリガーを設定するには、右上隅にある トリガーの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

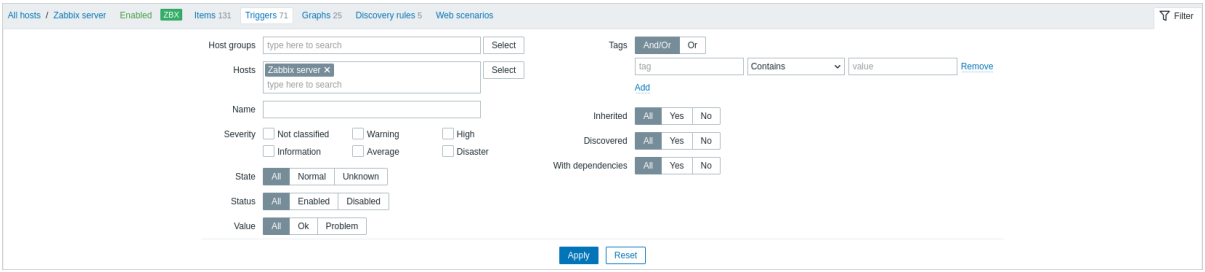
- *Enable* - トリガーの状態を *Enabled* に変更します。
- *Disable* - トリガーの状態を *Disabled* に変更します。
- *Copy* - トリガーを他のホストまたはテンプレートにコピーします。
- *Mass update* - 複数のトリガーに対して、いくつかのプロパティを一度に更新します。
- *Delete* - トリガーを削除します。

これらのオプションを使用するには、対象のトリガーの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるトリガーだけを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

右上にある *Filter* アイコンを使用できます。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



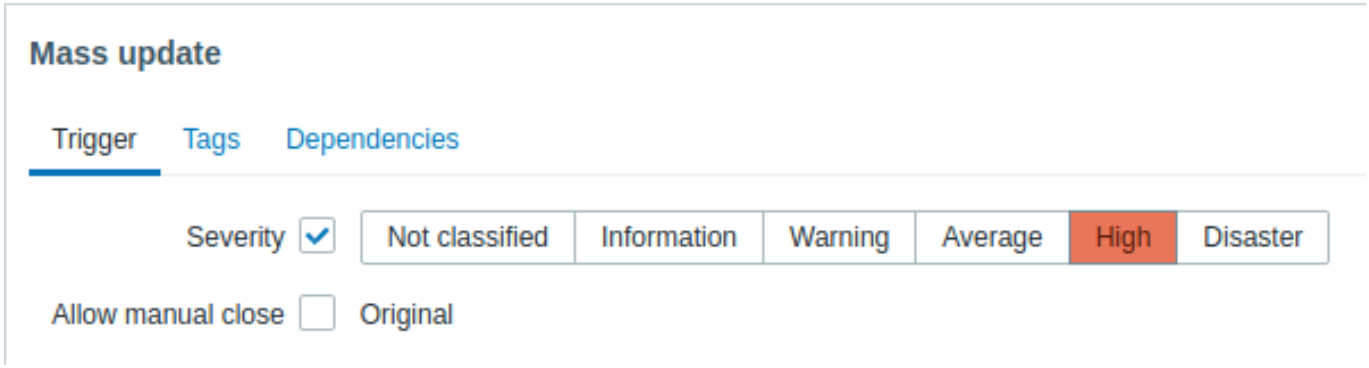
Parameter	Description
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、配下のすべてのネストされたホストグループも暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
Hosts	1 つ以上のホストで絞り込みます。 上でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに制限されます。
Name	トリガー名で絞り込みます。
Severity	1 つまたは複数のトリガーの深刻度で絞り込みます。
State	トリガーのexpression state（Normal/Unknown）で絞り込みます。
Status	トリガーの有効化状態（Enabled/Disabled）で絞り込みます。
Value	トリガーの値で絞り込みます。
Tags	トリガーのタグ名と値で絞り込みます。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字小文字を区別します。 各条件には次の演算子を使用できます: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字小文字を区別） **Contains** - タグ値に入力した文字列を含む指定タグ名を含める（部分一致、大文字小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字小文字を区別） **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外する（部分一致、大文字小文字を区別しない） 条件の計算方法は 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です タグ名とタグ値の両方のフィールドでマクロとmacro functionsがサポートされています。
Inherited	テンプレートから継承されたトリガー（または継承されていないトリガー）で絞り込みます。
Discovered	low-level discovery によって検出されたトリガー（または検出されていないトリガー）で絞り込みます。
With dependencies	依存関係を持つトリガー（または持たないトリガー）で絞り込みます。

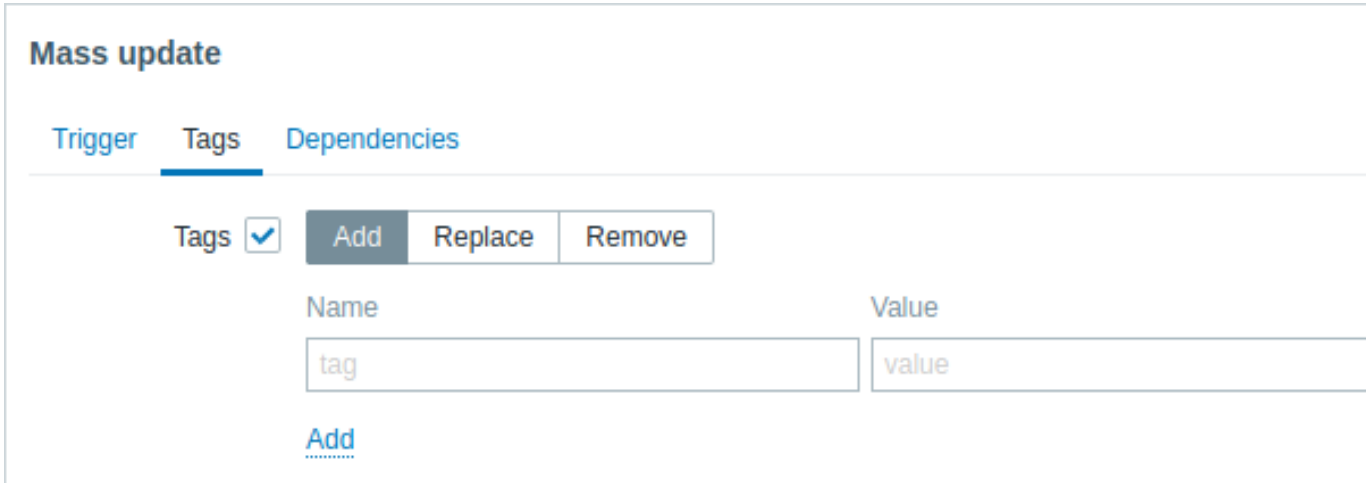
一括更新の使用

一括更新を使用すると、複数のトリガーの属性を一度に変更できるため、各トリガーを個別に開いて編集する必要がなくなります。

トリガーを一括更新するには、次の手順を実行します。

- リストで、更新するトリガーのチェックボックスを選択します
- リストの下にある 一括更新をクリックします
- 必要な属性（トリガー、タグ、または 依存関係）のタブに移動します
- 更新する属性のチェックボックスを選択します

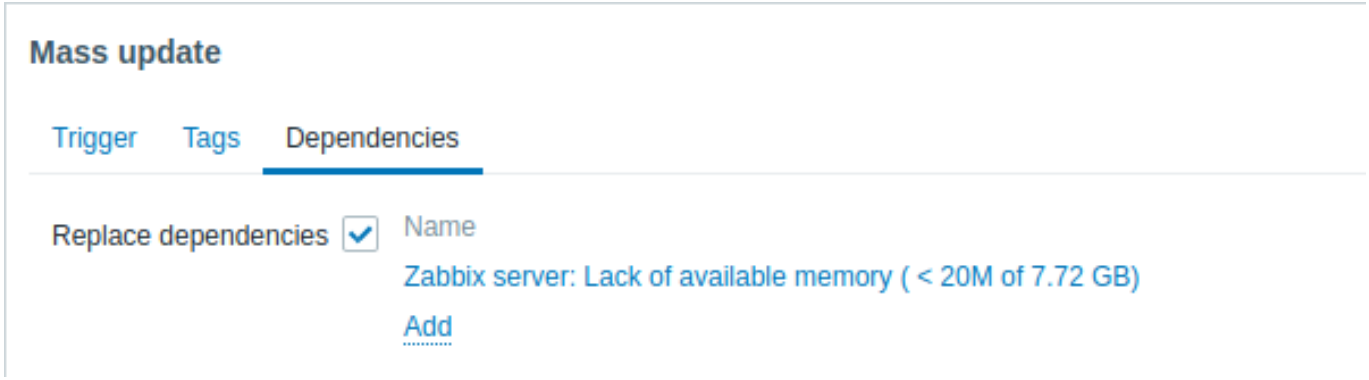




タグ更新で該当するボタンを選択すると、次のオプションを使用できます。

- 追加 - トリガーに新しいタグを追加します。
- 置換 - トリガーに設定されている既存のタグを削除し、下記で指定したタグに置き換えます。
- 削除 - 指定したタグをトリガーから削除します。

同じ名前で値が異なるタグは「重複」とみなされないため、同じトリガーに追加できることに注意してください。



依存関係を置換 - トリガーに設定されている既存の依存関係を削除し、指定した依存関係に置き換えます。

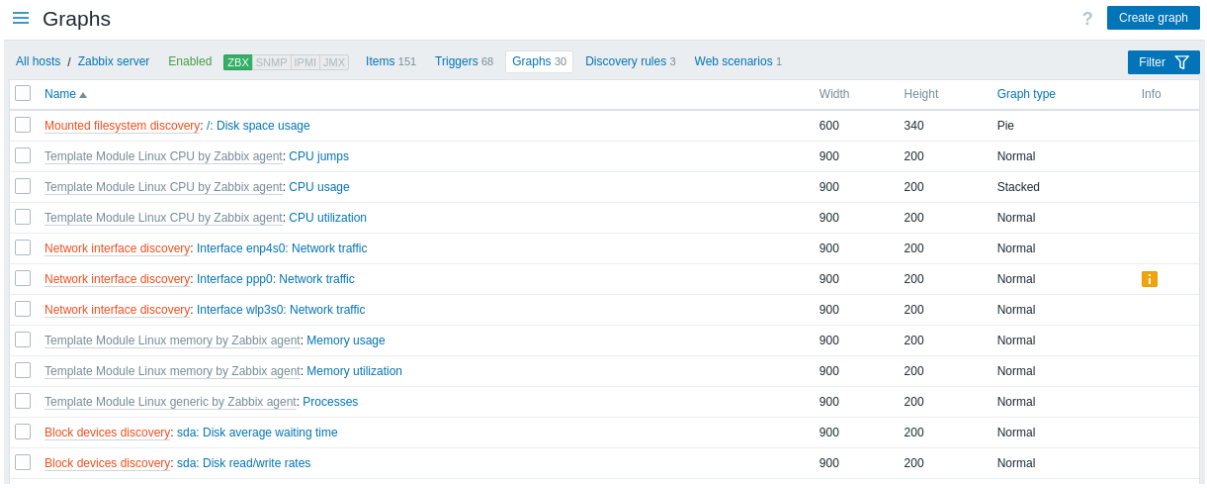
更新をクリックして変更を適用します。

グラフ

概要

ホストのカスタムグラフ一覧には、データ収集 > ホストから対象のホストの グラフをクリックしてアクセスできます。

既存のグラフ一覧が表示されます。



表示されるデータ:

列	説明
ホスト	グラフのホスト。 ホスト名をクリックすると、ホスト設定フォームが開きます。 この列は、フィルターで複数のホストまたはホストが選択されていない場合にのみ表示されます。
名前	カスタムグラフの名前。グラフの詳細への青色のリンクとして表示されます。 グラフ名のリンクをクリックすると、グラフ設定フォームが開きます。 ホストのグラフがテンプレートに属している場合、テンプレート名がグラフ名の前に灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのグラフ一覧が開きます。 グラフがグラフのプロトタイプから作成されている場合、その名前の前にローレベルディスカバリルール名がオレンジ色で表示されます。ディスカバリルール名をクリックすると、グラフのプロトタイプ一覧が開きます。
幅	グラフの幅が表示されます。
高さ	グラフの高さが表示されます。
グラフのタイプ	グラフのタイプ（ノーマル、積み上げ、円、または分解）が表示されます。
情報	グラフが正常に動作している場合、この列にはアイコンは表示されません。エラーが発生した場合は、文字「i」が表示された四角形のアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいグラフを設定するには、右上隅にある グラフの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- *Copy* - グラフを他のホストまたはテンプレートにコピーします
- *Delete* - グラフを削除します

これらのオプションを使用するには、対象のグラフの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

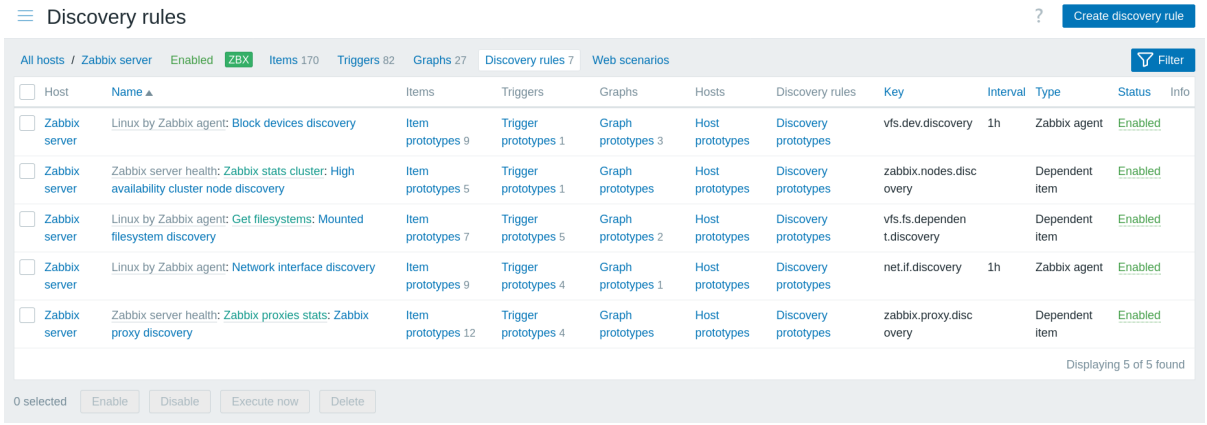
グラフはホストグループとホストでフィルタリングできます。検索パフォーマンスを向上させるために、データはマクロを解決せずに検索されます。

ディスカバリルール

概要

ホストのローレベルディスカバリルールの一覧には、データ収集 > ホストから対象のホストの ディスカバリをクリックしてアクセスできます。

既存のローレベルディスカバリルールの一覧が表示されます。フィルター設定を変更すると、ホストに依存しないすべてのディスカバリルール、または特定のホストグループに属するすべてのディスカバリルールを表示することもできます。



表示されるデータ:

列	説明
ホスト	表示名が表示されます。 ホスト名をクリックすると、ホスト設定フォームが開きます。 表示名がない場合は、技術的なホスト名が表示されます。
名前	ルール名が青色のリンクとして表示されます。 ルール名をクリックすると、ローレベルディスカバリルール設定フォームが開きます。 ディスカバリルールがテンプレートに属している場合、ルール名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのルール一覧が開きます。
アイテム	アイテムのプロトタイプ一覧へのリンクが表示されます。 既存のアイテムのプロトタイプ数が灰色で表示されます。
トリガー	トリガーのプロトタイプ一覧へのリンクが表示されます。 既存のトリガーのプロトタイプ数が灰色で表示されます。
グラフ	グラフのプロトタイプ一覧へのリンクが表示されます。 既存のグラフのプロトタイプ数が灰色で表示されます。
ホスト	ホストのプロトタイプ一覧へのリンクが表示されます。 既存のホストのプロトタイプ数が灰色で表示されます。
キー	ディスカバリに使用されるアイテムキーが表示されます。
間隔	ディスカバリを実行する頻度が表示されます。 注ディスカバリは、一覧の下にある 今すぐ実行ボタンを押して直ちに実行することもできます。
タイプ	ディスカバリに使用されるアイテムのタイプ（Zabbix エージェント、SNMP エージェントなど）が表示されます。
ステータス	ディスカバリルールのステータス（有効、無効、または 未サポート）が表示されます。ステータスをクリックすると変更できます。有効から 無効（およびその逆）、または 未サポートから 無効（およびその逆）に変更できます。
情報	問題がない場合、この列にはアイコンは表示されません。エラーがある場合は、文字「i」が表示された四角形のアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいローレベルディスカバリルールを設定するには、右上隅にある ディスカバリルールの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

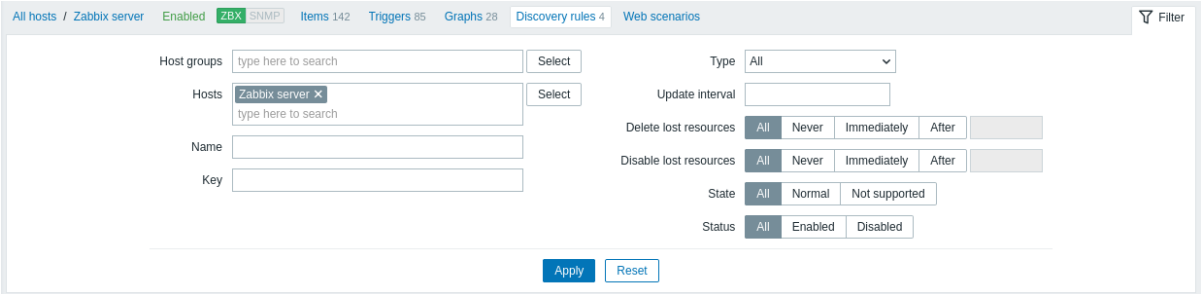
- *Enable* - ローレベルディスカバリルールのステータスを *Enabled* に変更します。
- *Disable* - ローレベルディスカバリルールのステータスを *Disabled* に変更します。
- *Execute now* - ディスカバリルールに基づくディスカバリを直ちに実行します。詳細 を参照してください。なお、ディスカバリを即時実行した場合、設定キャッシュは更新されないため、結果にはディスカバリルール設定へのごく最近の変更が反映されません。
- *Delete* - ローレベルディスカバリルールを削除します。

これらのオプションを使用するには、各ディスカバリルールの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のある発見ルールのみを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

Filter リンクは発見ルール一覧の上部にあります。これをクリックするとフィルターが表示され、ホストグループ、ホスト、名前、アイテムキー、アイテムタイプ、その他のパラメーターで発見ルールを絞り込めます。

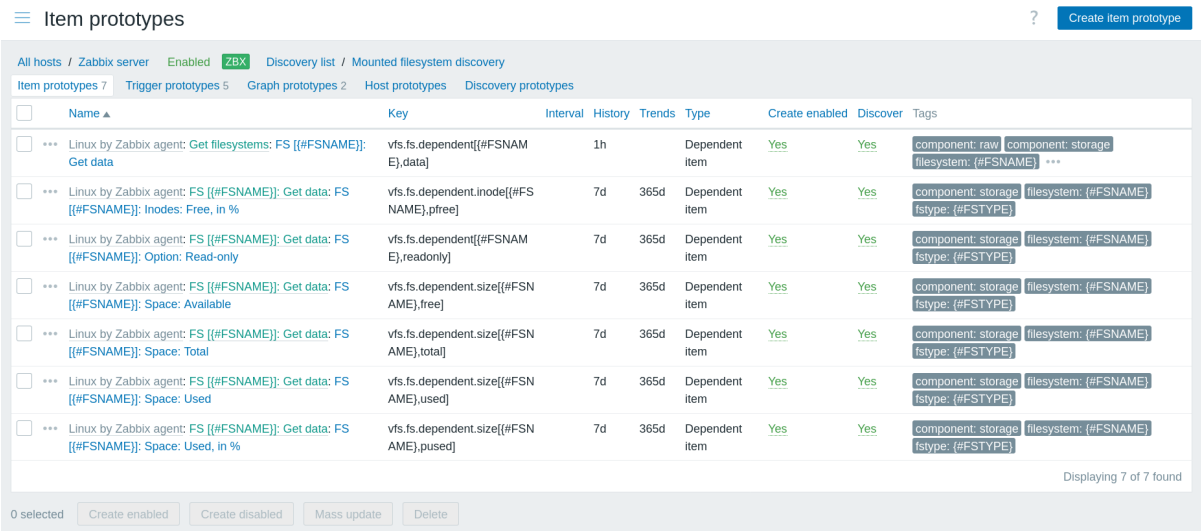


Parameter	Description
Host groups	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、配下のすべてのネストされたホストグループが暗黙的に選択されます。
Hosts	1 つ以上のホストでフィルターします。
Name	発見ルール名でフィルターします。
Key	発見アイテムのキーでフィルターします。
Type	発見アイテムのタイプでフィルターします。
Update interval	更新間隔でフィルターします。 Zabbix trapper と依存アイテムでは使用できません。
Delete lost resources	*Delete lost resources* の期間でフィルターします。
Disable lost resources	*Disable lost resources* の期間でフィルターします。
SNMP OID	SNMP OID でフィルターします。 *SNMP agent* がタイプとして選択されている場合のみ使用できます。
State	発見ルールの状態でフィルターします (All/Normal/Not supported)。
Status	発見ルールのステータスでフィルターします (All/Enabled/Disabled)。

アイテムプロトタイプ

概要

このセクションには、ホスト上のローレベルディスカバリルールのアイテムプロトタイプが表示されます。アイテムプロトタイプは、ローレベルディスカバリ中に作成される実際のホストアイテムの基礎となります。



表示されるデータ:

Column	Description
Name	アイテムプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、アイテムプロトタイプ設定フォームが開きます。 アイテムプロトタイプがテンプレートに属している場合、ルール名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのアイテムプロトタイプ一覧が開きます。 アイテムプロトタイプのタイプが「従属アイテム」の場合、マスターアイテム名がアイテムプロトタイプ名の前に緑色で表示されます。
Key	アイテムプロトタイプのキーが表示されます。
Interval	チェックの実行間隔が表示されます。
History	アイテムデータの履歴を保持する日数が表示されます。
Trends	アイテムのトレンド履歴を保持する日数が表示されます。
Type	アイテムプロトタイプのタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
Create enabled	このプロトタイプに基づくアイテムの作成状態: **Yes** - 有効 **No** - 無効。Yes と No はクリックして切り替えられます。
Discover	このプロトタイプに基づくアイテムのディスカバリ状態: **Yes** - ディスカバリする **No** - ディスカバリしない。Yes と No はクリックして切り替えられます。
Tags	アイテムプロトタイプのタグが表示されます。

新しいアイテムプロトタイプを設定するには、右上隅にあるアイテムプロトタイプの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

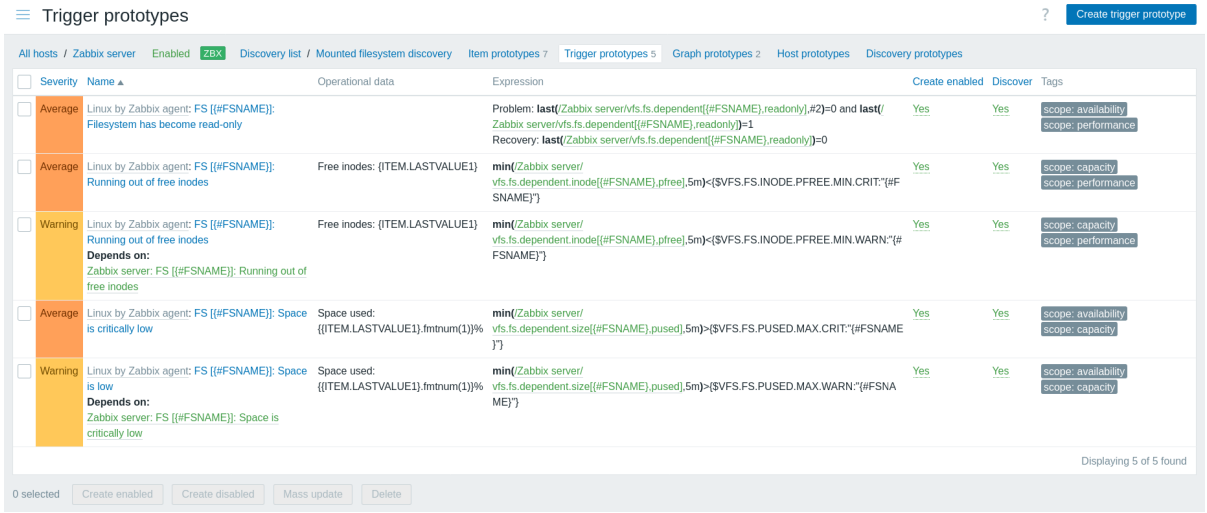
- 有効で作成 - これらのアイテムを 有効として作成します
- 無効で作成 - これらのアイテムを 無効として作成します
- 一括更新 - これらのアイテムプロトタイプを一括更新します
- 削除 - これらのアイテムプロトタイプを削除します

これらのオプションを使用するには、該当するアイテムプロトタイプの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

トリガーのプロトタイプ

概要

このセクションには、ホスト上のローレベルディスカバリルールのトリガーのプロトタイプが表示されます。トリガーのプロトタイプは、ローレベルディスカバリ中に作成される実際のホストのトリガーの基礎となります。



表示されるデータ:

列	説明
名前	トリガーのプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、トリガーのプロトタイプ設定フォームが開きます。 トリガーのプロトタイプがリンクされたテンプレートに属している場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのトリガーのプロトタイプ一覧が開きます。
運用データ	トリガーの運用データの形式が表示されます。これには、監視 > 障害で動的に解決される任意の文字列とマクロが含まれます。
作成を有効化	このプロトタイプに基づくトリガーの作成状態: はい - 有効 いいえ - 無効。表示されている「はい」または「いいえ」をクリックして切り替えられます。
ディスカバリ	このプロトタイプに基づくトリガーをディスカバリするかどうか: はい - ディスカバリする いいえ - ディスカバリしない。表示されている「はい」または「いいえ」をクリックして切り替えられます。
タグ	トリガーのプロトタイプのタグが表示されます。

新しいトリガーのプロトタイプを設定するには、右上隅にあるトリガーのプロトタイプを作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

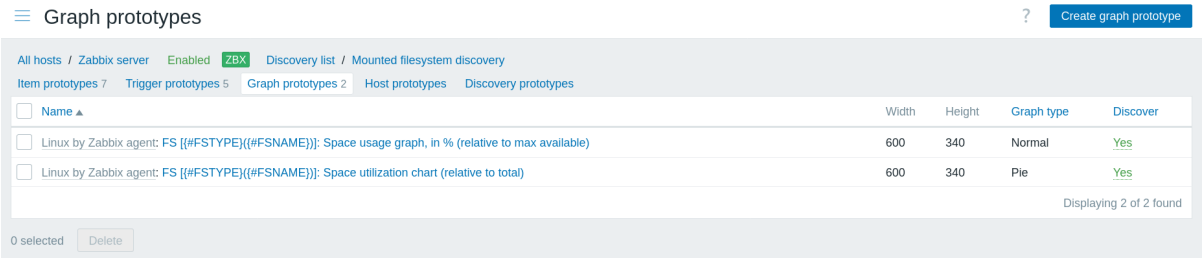
- 有効で作成 - これらのトリガーを 有効として作成します
- 無効で作成 - これらのトリガーを 無効として作成します
- 一括更新 - これらのトリガーのプロトタイプを一括更新します
- 削除 - これらのトリガーのプロトタイプを削除します

これらのオプションを使用するには、該当するトリガーのプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

グラフプロトタイプ

概要

このセクションには、ホスト上のローレベルディスカバリールールのグラフプロトタイプが表示されます。グラフプロトタイプは、ローレベルディスカバリー中に作成される実際のホストのグラフの基礎となります。



表示されるデータ:

列	説明
名前	グラフプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、グラフプロトタイプ設定フォームが開きます。 グラフプロトタイプがリンクされたテンプレートに属している場合、グラフ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのグラフプロトタイプ一覧が開きます。
幅	グラフプロトタイプの幅が表示されます。
高さ	グラフプロトタイプの高さが表示されます。
タイプ	グラフプロトタイプのタイプ（*Normal*、*Stacked*、*Pie*、または *Exploded*）が表示されます。
ディスカバー	このプロトタイプに基づいてグラフをディスカバリーするかどうか: **Yes** - ディスカバリーする **No** - ディスカバリーしない。クリックすると「Yes」と「No」を切り替えられます。

新しいグラフプロトタイプを設定するには、右上隅にあるグラフプロトタイプの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

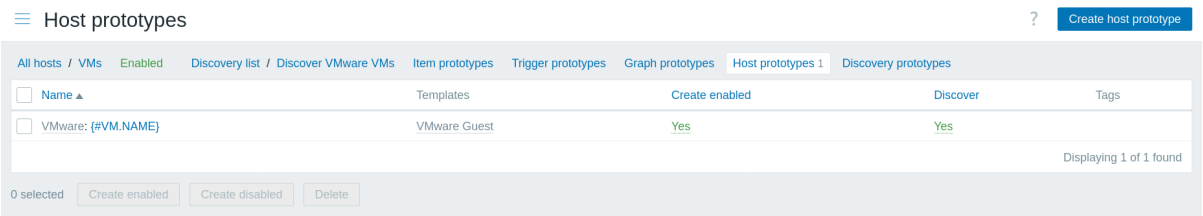
- 削除 - これらのグラフプロトタイプを削除します

これらのオプションを使用するには、対象のグラフプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

ホストプロトタイプ

概要

このセクションには、ホスト上のローレベルディスカバリルールのホストプロトタイプが表示されます。ホストプロトタイプは、ローレベルディスカバリ中に作成される実際のホストの基礎となります。



表示されるデータ:

列	説明
名前	ホストプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、ホストプロトタイプ設定フォームが開きます。 ホストプロトタイプがリンクされたテンプレートに属している場合、ホスト名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのホストプロトタイプ一覧が開きます。
テンプレート	ホストプロトタイプのテンプレートが表示されます。
作成を有効化	このプロトタイプに基づくホストの作成: はい - 有効 いいえ - 無効。クリックして「はい」と「いいえ」を切り替えられます。
ディスカバリ	このプロトタイプに基づくホストのディスカバリ: はい - ディスカバリする いいえ - ディスカバリしない。クリックして「はい」と「いいえ」を切り替えられます。
タグ	ホストプロトタイプのタグが表示されます。

新しいホストプロトタイプを設定するには、右上隅にあるホストプロトタイプの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンで一括編集オプションを利用できます:

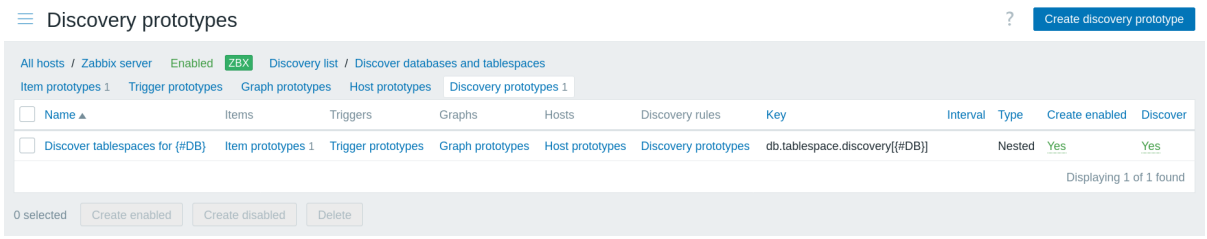
- 有効で作成 - これらのホストを有効として作成します
- 無効で作成 - これらのホストを無効として作成します
- 削除 - これらのホストプロトタイプを削除します

これらのオプションを使用するには、該当するホストプロトタイプの前のチェックボックスをオンにし、必要なボタンをクリックします。

ディスカバリープロトタイプ

概要

このセクションには、ホスト上のローレベルディスカバリルールのディスカバリプロトタイプが表示されます。ディスカバリプロトタイプは、親ディスカバリルールにネストされたディスカバリルールです。



表示されるデータ:

Column	Description
Name	ディスカバリプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、ディスカバリプロトタイプ設定フォームが開きます。 ディスカバリプロトタイプがリンクされたテンプレートに属している場合、テンプレート名がホスト名の前に灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、リンクされたテンプレートレベルのディスカバリプロトタイプ一覧が開きます。
Items	アイテムプロトタイプの一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数が灰色で表示されます。
Triggers	トリガープロトタイプの一覧へのリンクが表示されます。 既存のトリガープロトタイプ数が灰色で表示されます。
Graphs	グラフプロトタイプの一覧へのリンクが表示されます。 既存のグラフプロトタイプ数が灰色で表示されます。
Hosts	ホストプロトタイプの一覧へのリンクが表示されます。 既存のホストプロトタイプ数が灰色で表示されます。
Discovery rules	ディスカバリプロトタイプの一覧へのリンクが表示されます。 既存のディスカバリプロトタイプ数が灰色で表示されます。
Key	ディスカバリに使用されるアイテムキーが表示されます。
Interval	ディスカバリの実行頻度が表示されます。 注ディスカバリは、一覧の下にある今すぐ実行ボタンを押して直ちに実行することもできます。
Type	ディスカバリに使用されるアイテムのタイプ（Zabbix エージェント、SNMP エージェントなど）が表示されます。
Create enabled	ディスカバリプロトタイプの作成ステータス（作成有効（*Yes*）または作成無効（*No*））が表示されます。ステータスをクリックすると、Yes から No（またはその逆）に変更できます。
Discover	ディスカバリプロトタイプのディスカバリステータス（ディスカバリ実行（*Yes*）またはディスカバリを実行しない（*No*））が表示されます。ステータスをクリックすると、Yes から No（またはその逆）に変更できます。

新しいディスカバリプロトタイプを設定するには、右上隅にあるディスカバリプロトタイプの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 有効で作成 - これらのディスカバリプロトタイプを 有効として作成します
- 無効で作成 - これらのディスカバリプロトタイプを 無効として作成します
- 削除 - これらのディスカバリプロトタイプを削除します

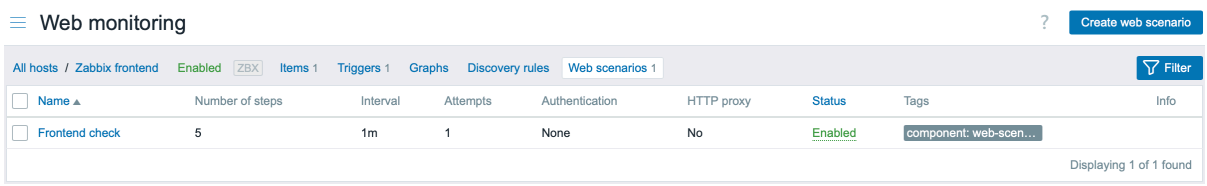
これらのオプションを使用するには、該当するディスカバリプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

Web シナリオ

概要

ホストのWeb シナリオ リストには、データ収集 > ホストから、対象のホストの *Web* をクリックしてアクセスできます。

既存の Web シナリオのリストが表示されます。



表示されるデータ:

列	説明
名前	Web シナリオの名前。Web シナリオ名をクリックすると、Web シナリオ設定フォームが開きます。 ホストの Web シナリオがテンプレートに属している場合、Web シナリオ名の前にテンプレート名がグレーのリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルの Web シナリオリストが開きます。
ステップ数	シナリオに含まれるステップ数。
更新間隔	シナリオを実行する頻度。
試行回数	Web シナリオのステップを実行する試行回数。
認証	認証方式（Basic、NTLM、Kerberos、Digest、None）が表示されます。
HTTP プロキシ	HTTP プロキシが表示されます。使用していない場合は「No」と表示されます。
ステータス	Web シナリオのステータス（有効または無効）が表示されます。 ステータスをクリックすると変更できます。
タグ	Web シナリオのタグが表示されます。 最大 3 つのタグ（名前: 値のペア）を表示できます。タグが 4 つ以上ある場合は「...」リンクが表示され、マウスオーバーするとすべてのタグを確認できます。
情報	すべて正常に動作している場合、この列にはアイコンは表示されません。エラーが発生した場合は、「i」の文字が入った四角形のアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明がツールチップに表示されます。

新しい Web シナリオを設定するには、右上隅にある *Web* シナリオの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

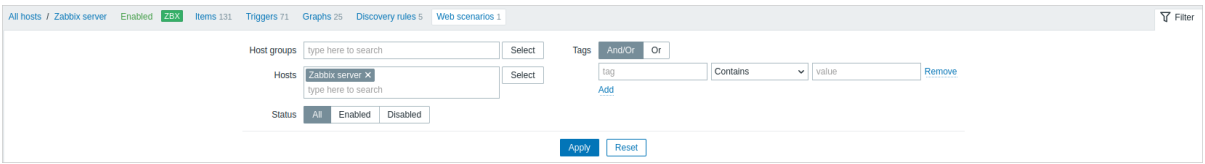
- *Enable* - シナリオのステータスを *Enabled* に変更します
- *Disable* - シナリオのステータスを *Disabled* に変更します
- *Clear history and trends* - シナリオの履歴データとトレンドデータを削除します
- *Delete* - Web シナリオを削除します

これらのオプションを使用するには、対象の Web シナリオの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるシナリオのみを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

Filter リンクは、Web シナリオ一覧の上にあります。これをクリックすると、必要なフィルタリング条件を指定できるフィルターが開きます。



Parameter	Description
Host groups	1 つ以上のホストグループでフィルターします。 親ホストグループを指定すると、配下のすべてのネストされたホストグループが暗黙的に選択されます。
Hosts	1 つ以上のホストでフィルターします。 上でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに限定されます。
Status	Web シナリオのステータスでフィルターします。

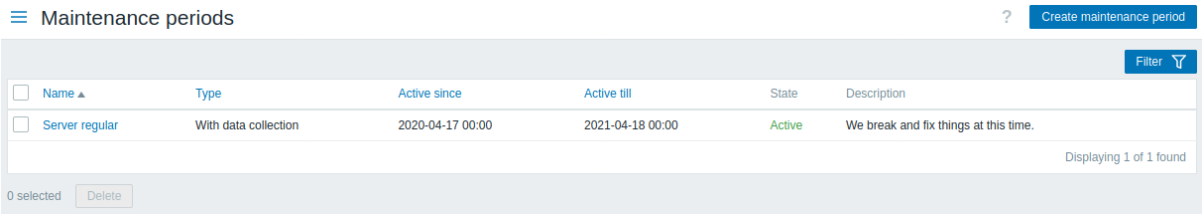
Parameter	Description
Tags	Web シナリオのタグ名と値でフィルターします。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件には、次の演算子を使用できます: **Exists** - 指定したタグ名を含める; **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別）; **Contains** - タグ値に入力した文字列を含む指定したタグ名を含める（部分一致、大文字と小文字を区別しない）; **Does not exist** - 指定したタグ名を除外する; **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別）; **Does not contain** - タグ値に入力した文字列を含む指定したタグ名を除外する（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます; **Or** - いずれか 1 つの条件を満たせば十分です。

メンテナンス

概要

データ収集 > メンテナンスセクションでは、ユーザーはホストのメンテナンス期間を設定および管理できます。

既存のメンテナンス期間とその詳細が一覧表示されます。



表示されるデータ:

Column	Description
Name	メンテナンス期間の名前。メンテナンス期間の名前をクリックすると、メンテナンス期間の設定フォームが開きます。
Type	メンテナンスのタイプが表示されます: データ収集ありまたは データ収集なし
Active since	メンテナンス期間の実行がアクティブになる日時。 注: この時刻によってメンテナンス期間が有効になるわけではありません。メンテナンス期間は別途設定する必要があります。
Active till	メンテナンス期間の実行がアクティブでなくなる日時。
State	メンテナンス期間の状態: **Upcoming** - まもなくアクティブになる **Active** - アクティブ **Expired** - すでにアクティブではない
Description	メンテナンス期間の説明が表示されます。

新しいメンテナンス期間を設定するには、右上にある メンテナンス期間の作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、1 つの一括編集オプションを利用できます。

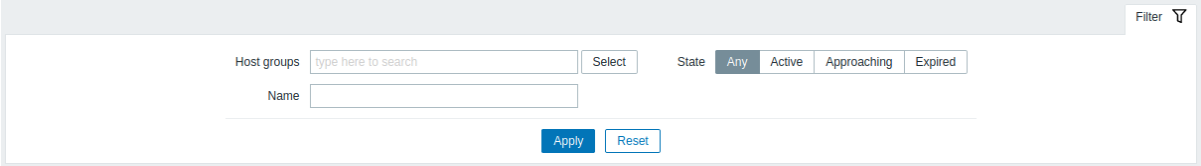
- 削除 - メンテナンス期間を削除します

このオプションを使用するには、対象のメンテナンス期間の前にあるチェックボックスを選択し、削除をクリックします。

フィルターの使用

フィルターを使用すると、関心のあるメンテナンス期間のみを表示できます。検索性能を向上させるため、データはマクロを展開せずに検索されます。

Filter リンクは、メンテナンス期間の一覧の上にあります。これをクリックするとフィルターが表示され、ホストグループ、名前、状態でメンテナンス期間を絞り込めます。



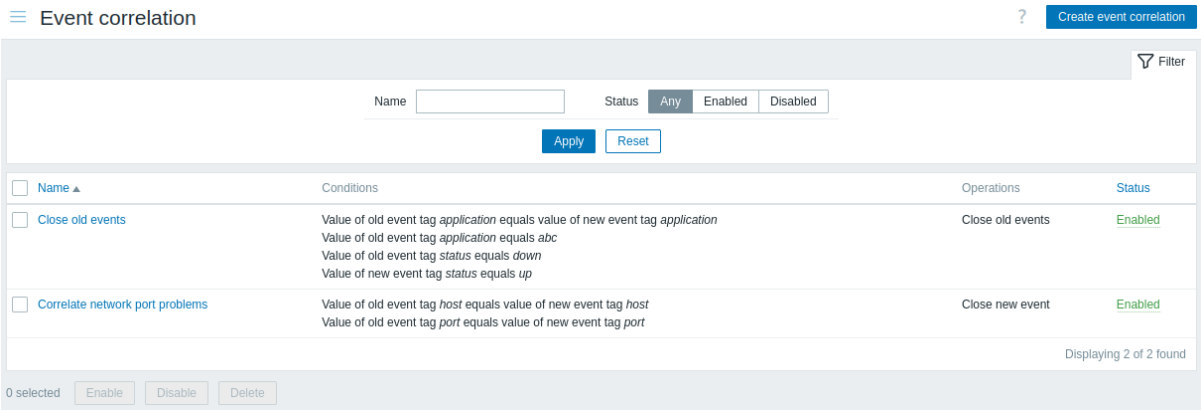
メンテナンス中のキューの計算

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

イベント相関

概要

データ収集 > イベント相関セクションでは、Zabbix イベントのグローバル相関ルールを設定および管理できます。



表示されるデータ:

列	説明
名前	相関ルールの名前。相関ルール名をクリックすると、イベント相関設定フォームが開きます。
条件	相関ルールの条件が表示されます。
操作	相関ルールの操作が表示されます。
ステータス	相関ルールのステータス（有効または 無効）が表示されます。 ステータスをクリックすると変更できます。

新しい相関ルールを設定するには、右上隅にあるイベント相関の作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

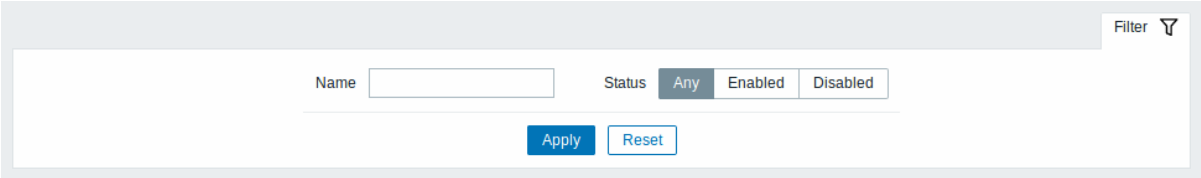
- *Enable* - 相関ルールのステータスを *Enabled* に変更します
- *Disable* - 相関ルールのステータスを *Disabled* に変更します
- *Delete* - 相関ルールを削除します

これらのオプションを使用するには、対象の相関ルールの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のある相関ルールのみを表示できます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

Filter リンクは、相関ルールの一覧の上にあります。これをクリックするとフィルターが表示され、名前とステータスで相関ルールを絞り込めます。

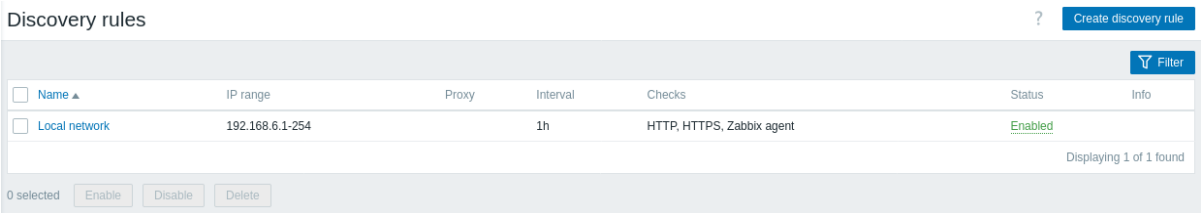


ディスカバリ

概要

データ収集 > ディスカバリセクションでは、ディスカバリルールを設定および管理できます。

既存のディスカバリルールとその詳細が一覧表示されます。



表示されるデータ:

列	説明
名前	ディスカバリルールの名前。ディスカバリルール名をクリックすると、ディスカバリルール設定フォームが開きます。
IP 範囲	ネットワークスキャンに使用する IP アドレスの範囲が表示されます。
プロキシ	プロキシによってディスカバリが実行される場合、プロキシ名が表示されます。
間隔	ディスカバリを実行する頻度が表示されます。
チェック	ディスカバリに使用されるチェックの種類が表示されます。
ステータス	ディスカバリルールのステータス（有効または 無効）が表示されます。 ステータスをクリックすると変更できます。
情報	すべて正常に動作している場合、この列には何も表示されません。エラーが発生した場合は、文字「i」が表示された赤い情報アイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいディスカバリルールを設定するには、右上隅にあるディスカバリルールの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

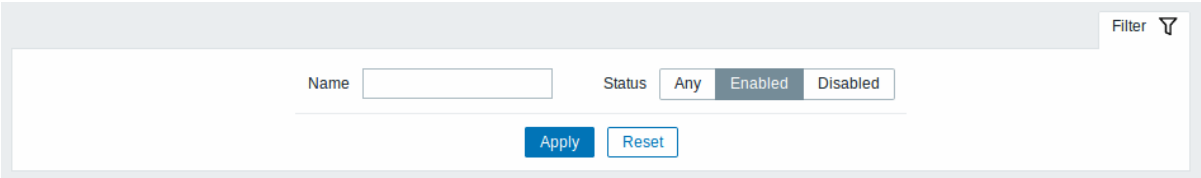
- 有効 - ディスカバリルールのステータスを 有効に変更します
- 無効 - ディスカバリルールのステータスを 無効に変更します
- 削除 - ディスカバリルールを削除します

これらのオプションを使用するには、該当するディスカバリルールの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるディスカバリルールのみを表示できます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

Filter リンクは、ディスカバリルールの一覧の上にあります。これをクリックするとフィルターが表示され、名前とステータスでディスカバリルールを絞り込めます。



アラート

概要

このメニューには、Zabbix でのアラートの設定に関連するセクションがあります。

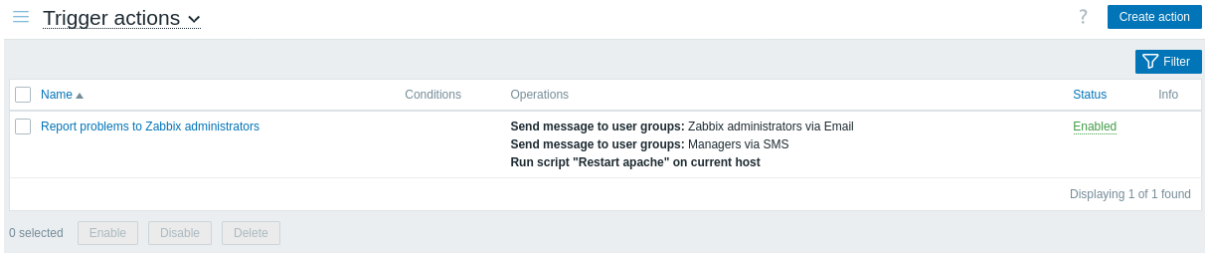
アクション

概要

障害 > アクションセクションでは、アクションを設定および管理できます。

表示されるアクションは、選択したイベントソース（トリガー、サービス、ディスカバリ、自動登録、内部アクション）に割り当てられたアクションです。

別のイベントソースに切り替えるには、障害メニューセクションの アクションをクリックします。左上のタイトルドロップダウンを使用してソースを切り替えることもできます。



表示されるデータ:

列	説明
名前	アクションの名前。アクション名をクリックすると、アクション設定フォームが開きます。
条件	アクションの条件が表示されます。
実行内容	アクションの実行内容が表示されます。 実行内容の一覧には、通知に使用されるメディアタイプ（メール、SMS、スクリプト）と、通知先の名前および姓（ユーザー名の後ろに括弧内で表示）も表示されます。 アクションの実行内容は、選択した実行内容のタイプに応じて、通知またはリモートコマンドになります。
ステータス	アクションのステータス（有効または 無効）が表示されます。 ステータスをクリックすると変更できます。 エスカレーションの進行中にアクションを無効にした場合の動作については、エスカレーションセクションを参照してください。
情報	すべて正常に動作している場合、この列にはアイコンは表示されません。アップグレードまたはローレベルディスカバリの後にアクションの実行内容や条件が不足している場合など、エラーが発生すると警告アイコンが表示されます。アイコンにカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいアクションを設定するには、右上の アクションの作成ボタンをクリックします。

スーパーユーザーの権限を持たないユーザーには、権限設定に従ってアクションが表示されます。つまり、特定の権限制限により、スーパーユーザーの権限を持たないユーザーがアクションの一覧全体を表示できない場合があります。スーパーユーザーの権限を持たないユーザーにアクションが表示されるのは、次の条件をすべて満たす場合です。

- ユーザーが、アクションの条件に含まれるホストグループ、ホスト、テンプレート、およびトリガーに対する読み書きアクセス権を持っている
- ユーザーが、アクションの実行内容、復旧時の実行内容、および更新時の実行内容に含まれるホストグループ、ホスト、およびテンプレートに対する読み書きアクセス権を持っている
- ユーザーが、アクションの実行内容、復旧時の実行内容、および更新時の実行内容に含まれるユーザーグループとユーザーに対する読み取りアクセス権を持っている

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

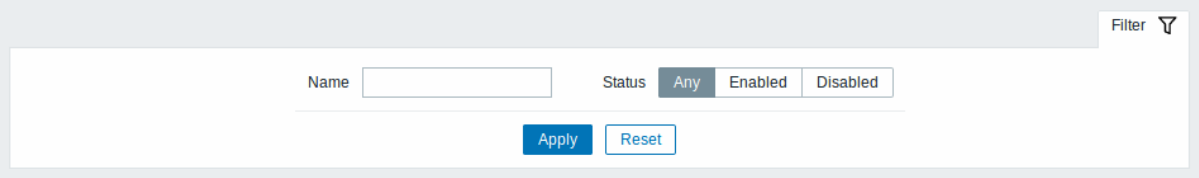
- *Enable* - アクションのステータスを *Enabled* に変更します
- *Disable* - アクションのステータスを *Disabled* に変更します
- *Delete* - アクションを削除します

これらのオプションを使用するには、対象のアクションの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のアクションのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクはアクションリストの上にあります。クリックすると名前とステータスでアクションをフィルタリングできるフィルターが使用可能になります。



メディアタイプ

概要

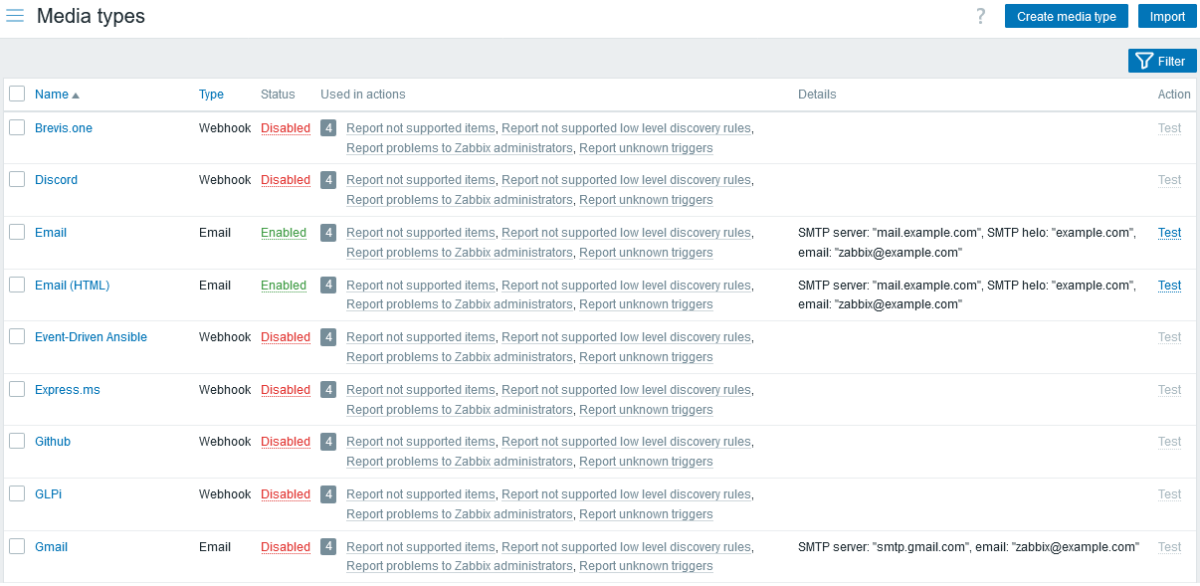
アラート > メディアタイプセクションでは、メディアタイプの情報を設定および管理できます。

このメニューは、SuperAdmin ユーザータイプのユーザーのみ利用できます。

メディアタイプ情報には、通知の配信チャネルとしてメディアを使用するための一般的な手順が含まれます。通知の送信先となる個々のメールアドレスなどの具体的な詳細は、個々のユーザーに設定します。

既存のメディアタイプとその詳細が一覧表示されます。

Note:
Web インターフェース設定ファイルの `$ZBX_FEATURE_FLAGS['media_type_denylist']` 機能フラグで配信方法（*Email*、*SMS*、*Webhook*、*Script*、または *Push*）が指定されている場合、対応する事前設定済みのメディアタイプはこのセクションで無効になります。



表示されるデータ:

Column	Description
Name	メディアタイプの名前。名前をクリックすると、メディアタイプ設定フォームが開きます。
Type	メディアのタイプ（メール、SMS など）が表示されます。
Status	メディアタイプのステータス（*Enabled* または *Disabled*）が表示されます。 ステータスをクリックすると変更できます。
Used in actions	メディアタイプが使用されているアクションが、該当するアクションの合計数とともに表示されます。 アクション名をクリックすると、アクション設定フォームが開きます。ユーザーにアクションへの権限がない場合、名前はクリックできません。
Details	メディアタイプの詳細情報が表示されます。

Column	Description
Actions	次の操作を利用できます: **Test** - クリックするとテストフォームが開きます。このフォームでメディアタイプのパラメータ（テスト用の件名と本文を含む受信者アドレスなど）を入力し、テストメッセージを送信して、設定したメディアタイプが機能することを確認できます。詳細については、メディアタイプのテストも参照してください。

新しいメディアタイプを設定するには、右上隅にある *Create media type* ボタンをクリックします。

メディアタイプをインポートするには、右上隅にある *Import* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

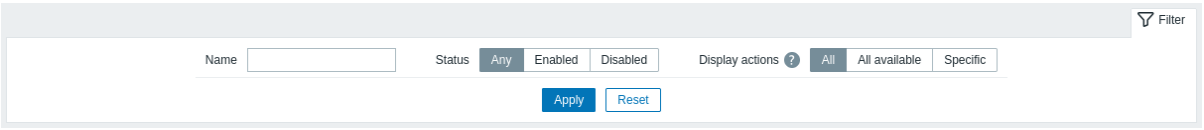
- *Enable* - メディアタイプのステータスを *Enabled* に変更します
- *Disable* - メディアタイプのステータスを *Disabled* に変更します
- *Export* - メディアタイプを YAML、XML、または JSON ファイルにエクスポートします
- *Delete* - メディアタイプを削除します

これらのオプションを使用するには、対象のメディアタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるメディアタイプのみを表示できます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

フィルターリンクは、メディアタイプの一覧の上にあります。これをクリックするとフィルターが表示され、名前とステータスでメディアタイプを絞り込めます。さらに、アクションの操作の詳細 にある *Send to media type* パラメータで定義されたメディアタイプの使用範囲に基づいて、*Used in actions* 列に表示されるアクションをフィルターで表示できます。



スクリプト

概要

アラート > スクリプトセクションでは、ユーザー定義のグローバルスクリプトを設定および管理できます。

このメニューは、SuperAdmin ユーザータイプのユーザーのみ利用できます。

グローバルスクリプトは、設定されたスコープおよびユーザー権限に応じて、次の場所から実行できます。

- さまざまな Web インターフェースの場所（ダッシュボード、障害、最新データ、マップなど）にあるホストメニューから。
- イベントメニューから。
- アクションのオペレーションとして実行。

スクリプトは、Zabbix エージェント、Zabbix サーバー（プロキシ）、または Zabbix サーバーのみで実行されます。コマンドの実行も参照してください。

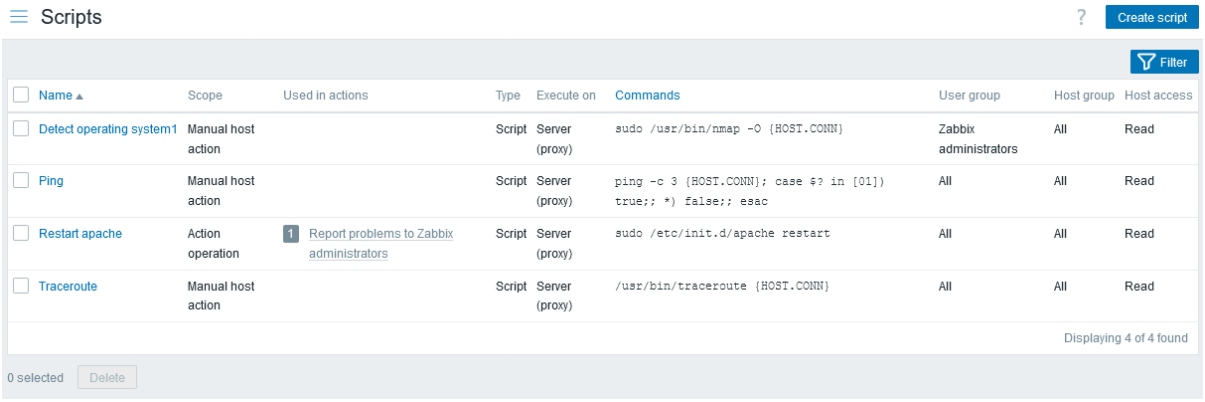
リモートコマンドは、Zabbix エージェントと Zabbix プロキシの両方でデフォルトでは無効になっています。次のように有効にできます。

- Zabbix エージェントで実行するリモートコマンドの場合、エージェント設定に、許可する各コマンドの `AllowKey=system.run[<comma` または `AllowKeyRegexp=^system\.run\[<command>,.*\]$` パラメータを追加します。ここで、`*` は実行モード（`wait` または `nowait`）のいずれかに一致します。
- Zabbix プロキシで実行するリモートコマンドの場合、プロキシ設定で `EnableRemoteCommands` パラメータを 1 に設定します。プロキシによって監視されているホスト上の Zabbix エージェントでリモートコマンドを実行する場合、Zabbix プロキシでリモートコマンドを有効にする必要はありません。

Attention:
ホスト上の任意のコマンドの実行を許可するため、必要な場合を除き `AllowKey=system.run[*]` または `AllowKeyRegexp=^system\.run\[.*\]$` は使用しないでください。セキュリティ上の理由から、完全一致するコマンドのみを許可することを推奨します。`system.run[*]` はデフォルトで拒否されます（`DenyKey=system.run[*]`）。

Zabbix サーバーでのグローバルスクリプトの実行は、サーバー設定で EnableGlobalScripts=0 を設定することにより無効にできます。Zabbix 7.0 以降で新規インストールした環境では、Zabbix サーバーでのグローバルスクリプトの実行はデフォルトで無効になっています。

既存のスクリプトとその詳細の一覧が表示されます。



表示されるデータ：

列	説明
名前	スクリプトの名前。スクリプト名をクリックすると、スクリプト設定フォームが開きます。
スコープ	スクリプトのスコープ（アクションのオペレーション、手動ホストアクション、または手動イベントアクション）。この設定により、スクリプトを利用できる場所が決まります。つまり、アクションのオペレーションのリモートコマンド、ホストメニュー、またはイベントメニューのいずれかです。
アクションで使用	スクリプトが使用されているすべてのアクションが、該当するアクションの総数とともに表示されます。 アクション名をクリックすると、アクション設定フォームが開きます。ユーザーにアクションへの権限がない場合、名前はクリックできません。
タイプ	スクリプトのタイプ（*URL*、*Webhook*、スクリプト、*SSH*、*Telnet*、または *IPMI* コマンド）が表示されます。
実行先	スクリプトを Zabbix エージェント、Zabbix プロキシ、またはサーバーで実行するか、Zabbix サーバーのみで実行するかが表示されます。
コマンド	スクリプト内で実行されるすべてのコマンドが表示されます。 webhook の場合、ここには何も表示されません。
ユーザーグループ	スクリプトを利用できるユーザーグループが表示されます（すべてのユーザーグループの場合は すべて）。
ホストグループ	スクリプトを利用できるホストグループが表示されます（すべてのホストグループの場合は すべて）。
ホストへのアクセス	ホストグループに対する権限レベル（読み取りまたは 書き込み）が表示されます。必要な権限レベルを持つユーザーのみ、スクリプトを実行できます。

新しいスクリプトを設定するには、右上隅にある スクリプトの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、1 つの一括編集オプションを利用できます。

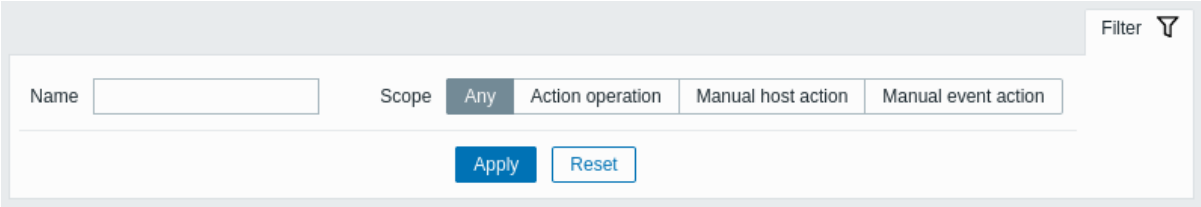
- 削除 - スクリプトを削除します

このオプションを使用するには、対象のスクリプトの前にあるチェックボックスを選択し、削除をクリックします。

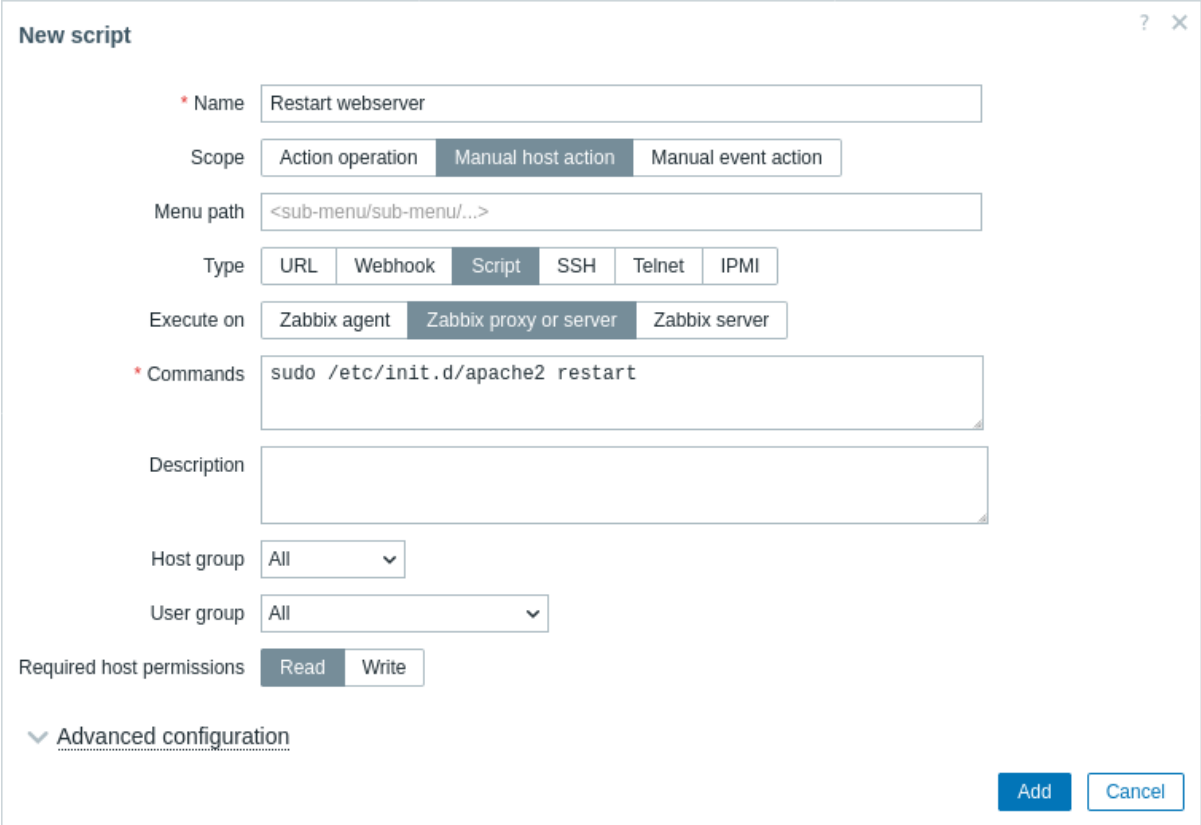
フィルターの使用

フィルターを使用すると、関心のあるスクリプトだけを表示できます。検索性能を向上させるため、データはマクロが展開されていない状態で検索されます。

フィルターリンクは、スクリプト一覧の上にあります。これをクリックするとフィルターが表示され、名前とスコープでスクリプトを絞り込めます。



グローバルスクリプトの設定



Note:
グローバルスクリプトでは、セキュリティを強化するため、通常のマクロではなくマクロ関数を使用することを推奨します。マクロは自動的にエスケープされないためです。

スクリプトの属性:

パラメータ	説明
名前	スクリプトの一意の名前。 例: `Clear /tmp filesystem`

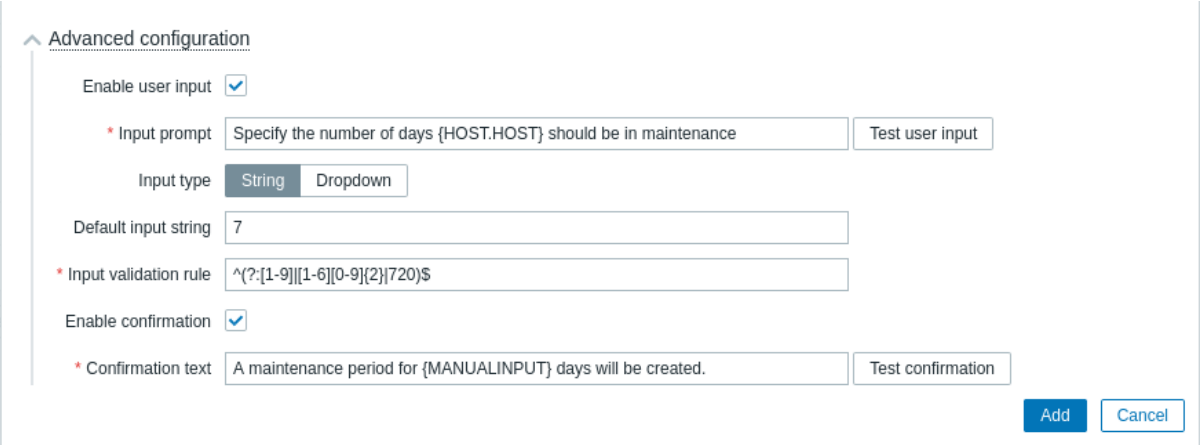
パラメータ		説明
スコープ		スクリプトのスコープ - アクションの実行、手動ホストアクション、または手動イベントアクション。この設定により、スクリプトを使用できる場所が決まります。つまり、アクションの実行のリモートコマンド、ホストメニュー、またはイベントメニューで使用できます。 スコープを「アクションの実行」に設定すると、*Alerts* > *Actions* にアクセスできるすべてのユーザーがスクリプトを使用できます。 スクリプトが実際にアクションで使用されている場合、スコープを「アクションの実行」以外に変更することはできません。 マクロのサポート スコープは、使用できるマクロの範囲に影響します。例えば、スクリプトを起動したユーザーの情報を渡せるように、ユーザー関連のマクロ（{USER.*}）がスクリプトでサポートされています。ただし、アクションの実行は自動的に実行されるため、スクリプトのスコープがアクションの実行の場合はサポートされません。 {MANUALINPUT} マクロを使用すると、スクリプトの実行時に手動入力を指定できます。これは手動ホストアクションおよび手動イベントアクションのスクリプトでサポートされています。 その他にサポートされているマクロについては、サポートされているマクロの表で、「トリガーベースの通知とコマンド/トリガーベースのコマンド」、「手動ホストアクションスクリプト」、「手動イベントアクションスクリプト」を検索してください。マクロがスペースを含む値（ホスト名など）に展開される可能性がある場合は、必要に応じて引用符で囲むことを忘れないでください。
メニューパス		スクリプトの目的のメニューパス。例えば、Default または Default/ と指定すると、スクリプトがそれぞれのディレクトリに表示されます。メニューは Main menu/Sub menu1/Sub menu2 のようにネストできます。監視セクションのホスト/イベントメニューからスクリプトにアクセスすると、指定したディレクトリに従って整理されます。 スコープで「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ、このフィールドが表示されます。
タイプ		該当するボタンをクリックして、スクリプトのタイプを選択します: **URL**、**Webhook**、スクリプト、**SSH**、**Telnet**、または **IPMI** コマンド。 タイプ **URL** は、スコープで「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ使用できます。
	スクリプトタイプ: **URL**	
	URL	ホストメニューまたはイベントメニューからすばやくアクセスするための URL を指定します。 マクロおよびカスタムのユーザーマクロがサポートされています。マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。 スクリプトの実行時に手動入力を指定できるように、このフィールドで {MANUALINPUT} マクロを使用します。例: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` マクロの値は URL エンコードしないでください。
	新しいウィンドウで開く	URL を新しいブラウザータブで開くか、同じブラウザータブで開くかを指定します。
	スクリプトタイプ: **Webhook**	
	パラメータ	Webhook 変数を属性と値のペアとして指定します。 次も参照してください: Webhookメディア設定。 パラメータ値では、マクロおよびカスタムのユーザーマクロがサポートされています。マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。
	スクリプト	パラメータフィールド内またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターに、JavaScript コードを入力します。 マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。 次も参照してください: Webhookメディア設定、追加の Javascript オブジェクト。
	タイムアウト	JavaScript の実行タイムアウト（1～60 秒、デフォルトは 30 秒）。 時間のサフィックスがサポートされています（例: 30s、1m）。
	スクリプトタイプ: スクリプト	

パラメータ		説明
	実行場所	該当するボタンをクリックして、シェルスクリプトを実行する場所を選択します: **Zabbix** エージェント - ホスト上で Zabbix エージェントがスクリプトを実行します (system.run アイテムが許可されている場合)。 **Zabbix** プロキシまたはサーバー - ホストがプロキシとサーバーのどちらで監視されているかに応じて、Zabbix プロキシまたはサーバーがスクリプトを実行します。 EnableRemoteCommandsで有効になっている場合、プロキシ上で実行されます。 サーバーパラメータEnableGlobalScriptsでグローバルスクリプトが有効になっている場合、サーバー上で実行されます。 **Zabbix** サーバー - Zabbix サーバーのみがスクリプトを実行します。 サーバーパラメータEnableGlobalScriptsでグローバルスクリプトが無効になっている場合、このオプションは使用できません。
	コマンド	スクリプト内で実行するコマンドの完全なパスを入力します。 マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。カスタムのユーザーマクロがサポートされています。
	スクリプトタイプ: **SSH**	
	認証方式	認証方式（パスワードまたは公開鍵）を選択します。
	ユーザー名	ユーザー名を入力します。
	パスワード	パスワードを入力します。 認証方式として「パスワード」が選択されている場合に、このフィールドを使用できます。
	公開鍵ファイル	公開鍵ファイルへのパスを入力します。 認証方式として「公開鍵」が選択されている場合に、このフィールドを使用できます。
	秘密鍵ファイル	秘密鍵ファイルへのパスを入力します。 認証方式として「公開鍵」が選択されている場合に、このフィールドを使用できます。
	パスフレーズ	パスフレーズを入力します。 認証方式として「公開鍵」が選択されている場合に、このフィールドを使用できます。
	ポート	Zabbix が接続する対象ホスト上のリモート SSH サービスのポートを入力します。
	コマンド	コマンドを入力します。 マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。カスタムのユーザーマクロがサポートされています。
	スクリプトタイプ: **Telnet**	
	ユーザー名	ユーザー名を入力します。
	パスワード	パスワードを入力します。
	ポート	Zabbix が接続する対象ホスト上のリモート Telnet サービスのポートを入力します。
	コマンド	コマンドを入力します。 マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。カスタムのユーザーマクロがサポートされています。
	スクリプトタイプ: **IPMI**	
	コマンド	IPMI コマンドを入力します。 マクロのサポートはスクリプトのスコープによって異なります（上記のスコープを参照）。カスタムのユーザーマクロがサポートされています。
説明		スクリプトの説明を入力します。
ホストグループ		スクリプトを使用できるホストグループを選択します（すべてのホストグループの場合はすべて）。

パラメータ	説明
ユーザーグループ	スクリプトを使用できるユーザーグループを選択します（すべてのユーザーグループの場合はすべて）。 スコープで「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ、このフィールドが表示されます。
必要なホスト権限	ホストグループの権限レベル（読み取りまたは書き込み）を選択します。必要な権限レベルを持つユーザーのみがスクリプトを実行できます。 スコープで「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ、このフィールドが表示されます。
詳細設定	詳細設定ヘッダーをクリックすると、詳細設定のオプションが表示されます。 スコープで「手動ホストアクション」または「手動イベントアクション」が選択されている場合のみ、このフィールドが表示されます。

高度な設定

高度な設定オプションは、折りたたみ可能な 高度な設定セクションで使用できます。



パラメータ	説明
ユーザー入力を有効にする	スクリプトの実行前にユーザーが手動で入力できるようにするには、チェックボックスを選択します。 手動で入力した値は、スクリプト内の {MANUALINPUT} マクロの値を置き換えます。 関連項目: 手動ユーザー入力。
入力プロンプト	ユーザー入力を促すカスタムテキストを入力します。このテキストは、手動入力ポップアップの入力フィールドの上に表示されます。 手動入力ポップアップのプレビューを表示するには、ユーザー入力をテストをクリックします。プレビューでは、入力文字列が入力検証ルール（以下のパラメータを参照）に準拠しているかどうかもテストできます。 マクロおよびユーザーマクロのサポートは、スクリプトのスコープによって異なります（一般的なスクリプト設定パラメータの スコープを参照）。
入力タイプ	手動入力のタイプを選択します。 文字列 - 1 つの文字列; ドロップダウン - 複数のドロップダウンオプションから値を選択します。

パラメータ	説明
ドロップダウンオプション	ユーザー入力のドロップダウンに表示する重複しない値を、カンマ区切りのリストで入力します。 ドロップダウンに空のオプションを含めるには、リストの先頭、途中、または末尾にカンマを追加します。 このフィールドは、入力タイプで「ドロップダウン」を選択した場合にのみ表示されます。
デフォルトの入力文字列	ユーザー入力のデフォルト文字列（または何も入力しない）を入力します。 このフィールドは、入力検証ルールフィールドに指定された正規表現に対して検証されます。 ここで入力した値は、手動入力ポップアップにデフォルトで表示されます。 このフィールドは、入力タイプで「文字列」を選択した場合にのみ表示されます。
入力検証ルール	ユーザー入力文字列を検証する正規表現を入力します。 グローバル正規表現がサポートされています。 このフィールドは、入力タイプで「文字列」を選択した場合にのみ表示されます。
確認を有効にする	スクリプトの実行前に確認メッセージを表示するには、チェックボックスを選択します。この機能は、危険を伴う可能性のある操作（再起動スクリプトなど）や、完了までに長時間かかる可能性のある操作で特に役立ちます。
確認テキスト	上記のチェックボックスで有効にした確認ポップアップに表示するカスタム確認テキストを入力します（例: リモートシステムを再起動します。よろしいですか?)。テキストの表示を確認するには、フィールドの横にある 確認をテストをクリックします。 マクロおよびカスタムユーザーマクロがサポートされています。 注: 確認メッセージのテスト時には、マクロは展開されません。

手動ユーザー入力と確認メッセージの両方が設定されている場合、それらは連続するポップアップウィンドウに表示されます。

手動ユーザー入力

手動ユーザー入力を使用すると、スクリプトを実行するたびにカスタムパラメータを指定できます。これにより、パラメータが 1 つだけ異なる類似のユーザースクリプトを複数作成する必要がなくなります。

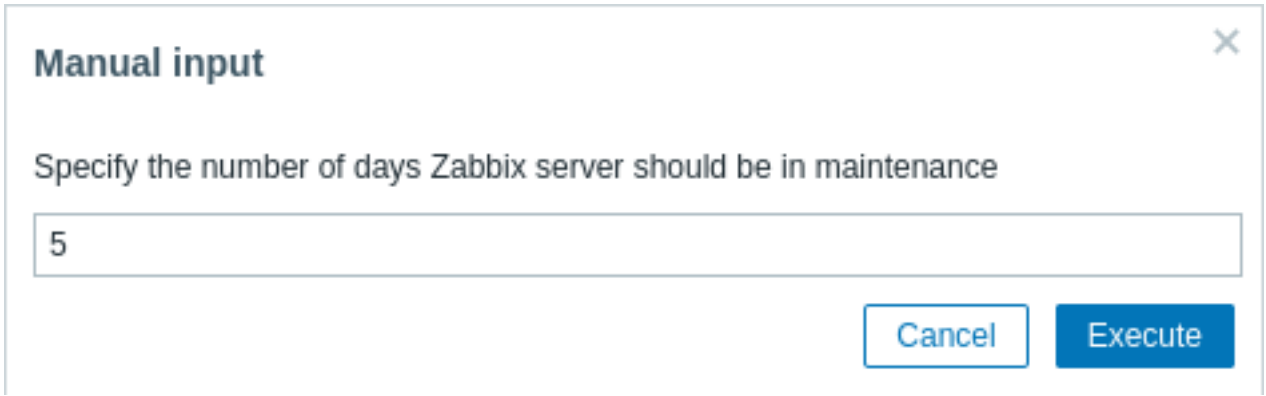
例えば、スクリプトの実行時に異なる整数値や URL アドレスを指定できます。

手動ユーザー入力を有効にするには、次の手順を実行します。

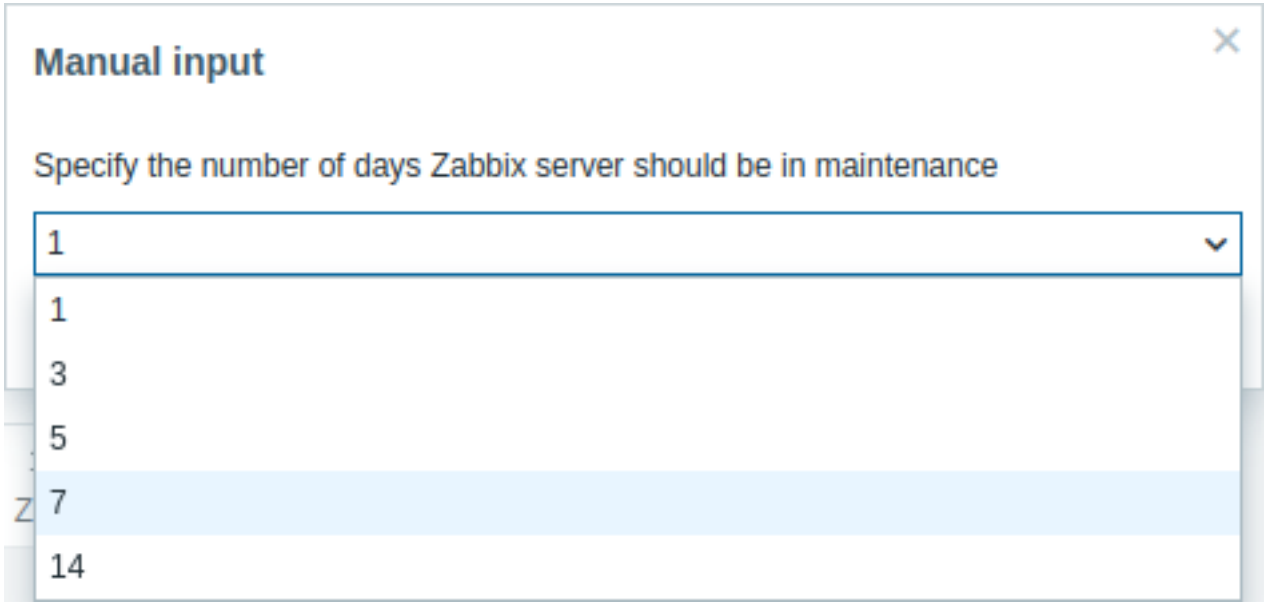
- 必要な箇所（コマンド、スクリプト、スクリプトパラメータ）でスクリプト内の {MANUALINPUT} マクロを使用します。または、URL スクリプトの URL フィールドで使用します。
- スクリプトの詳細設定で、手動ユーザー入力を有効にし、入力オプションを設定します。

ユーザー入力を有効にすると、スクリプトの実行前に、カスタム値の入力を求める手動入力ポップアップが表示されます。入力された値は、スクリプト内の {MANUALINPUT} に置き換えられます。

設定に応じて、ユーザーは文字列値を入力します。



または、あらかじめ定義されたオプションのドロップダウンから値を選択します。



手動ユーザー入力は、スコープが「手動ホストアクション」または「手動イベントアクション」のスクリプトでのみ使用できます。

スクリプトの実行と結果

Zabbix サーバーによって実行されるスクリプトは、コマンド実行ページで説明されている順序で実行されます。

スクリプトの結果は、スクリプトの実行後に表示されるポップアップウィンドウに表示されます。スクリプトの戻り値は標準出力です。

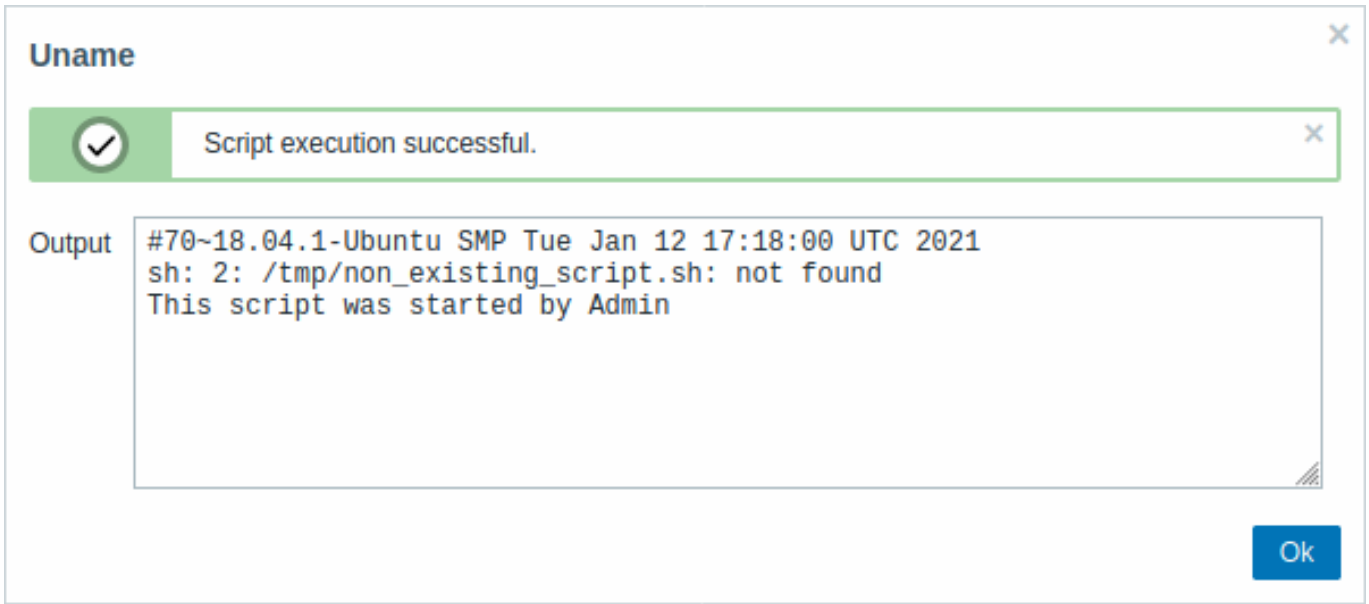
- スクリプトが正常に終了した場合（終了コード 0）、戻り値は 16MB に制限されます（切り捨てられる末尾の空白文字を含む）。
- スクリプトがエラーで終了した場合（終了コードが 0 以外）、戻り値は標準エラーとなり、2KB に制限されます。

Zabbix は、デフォルトでは拡張されたスクリプト出力を保存しません。出力の詳細全体を保持するには、スクリプト自体にロギングを実装できます（例: 出力をローカルのログファイルにリダイレクトする）。

Zabbix サーバーまたは Zabbix プロキシ上で実行されるスクリプトには、データベースの制限も適用されることに注意してください。

以下は、スクリプトと結果ウィンドウの例です。

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



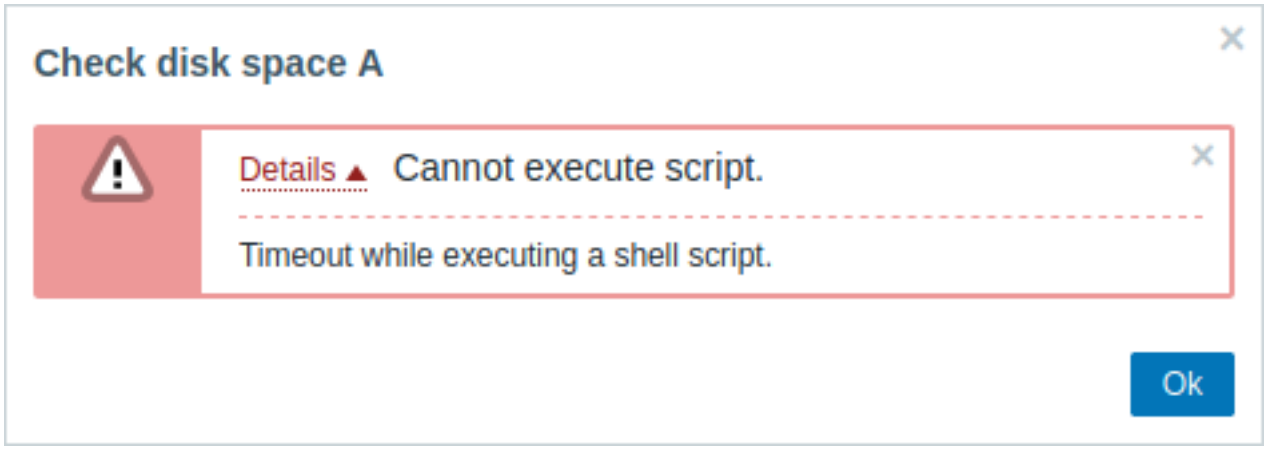
スクリプトの結果には、スクリプト自体は表示されません。

Zabbix エージェントでのスクリプトのタイムアウト

スクリプトの実行中にタイムアウトが発生する場合があります。

以下は、Zabbix エージェントで実行されるスクリプトと、その結果ウィンドウの例です。

```
sleep 5
df -h
```



この場合、次のエラーメッセージが表示されます。

```
Timeout while executing a shell script.
```

このような状況を避けるには、Zabbix エージェントの設定および Zabbix サーバーの設定で Timeout パラメータを調整するのではなく、スクリプト自体（上記の例では「5」）を最適化することを推奨します。

Zabbix エージェントの設定で Timeout パラメータを変更した場合、次のエラーメッセージが表示されます。

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

これは、Zabbix エージェントの設定で変更が行われたものの、Zabbix サーバーの設定でも Timeout パラメータを変更する必要があることを意味します。

アクティブエージェントでのスクリプトタイムアウト

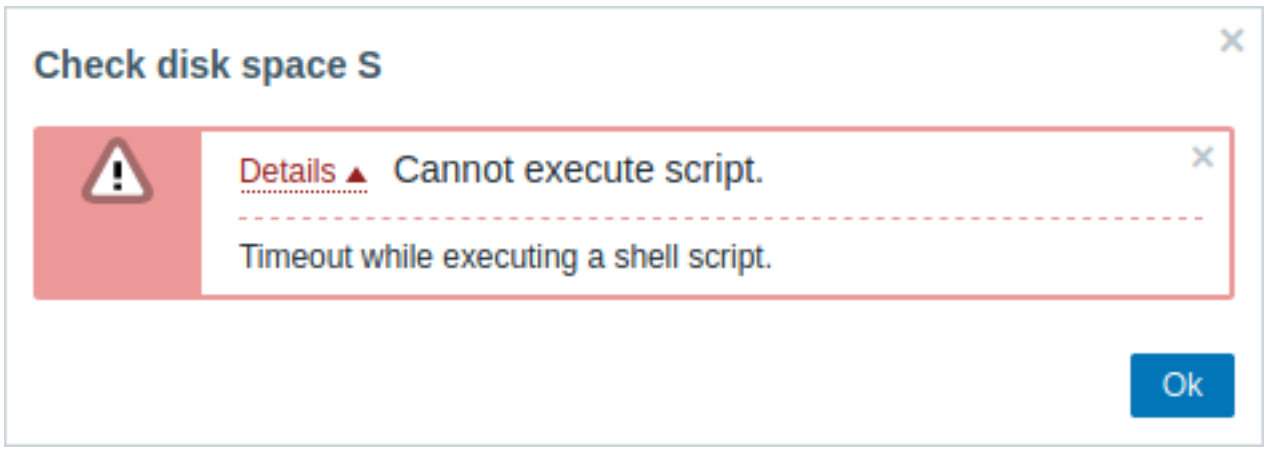
scripts は、アクティブモードで動作しているエージェント上でのみ実行できます。スクリプトの実行は、アクションのoperationまたは手動でのスクリプト実行によってトリガーされると、そのコマンドはアクティブチェックの設定に含まれ、アクティブエージェントが受信した時点で実行されます。

手動スクリプトは、スクリプト実行用のサーバー/プロキシのタイムアウトとともにアクティブエージェントへ送信されます。デフォルトのserver/proxyのスクリプト実行タイムアウトを増やしてください。このタイムアウトは、アクティブチェックの更新頻度よりも大きくする必要があります。そうでない場合、アクティブエージェントがスクリプトを受信して結果を返せるようになる前にタイムアウトしてしまいます。

Zabbix サーバー/プロキシでのスクリプトタイムアウト

Zabbix サーバーで実行されるスクリプトと、その結果ウィンドウの例を以下に示します。

```
sleep 11
df -h
```



TrapperTimeout パラメータを対応する値（この場合は 11 より大きい値）に調整するために Zabbix のサーバー設定を変更するのではなく、スクリプト自体を最適化することも推奨されます。

ユーザー

概要

このメニューには、Zabbix でのユーザーの設定に関連するセクションがあります。このメニューは、SuperAdminタイプのユーザーのみが利用できます。

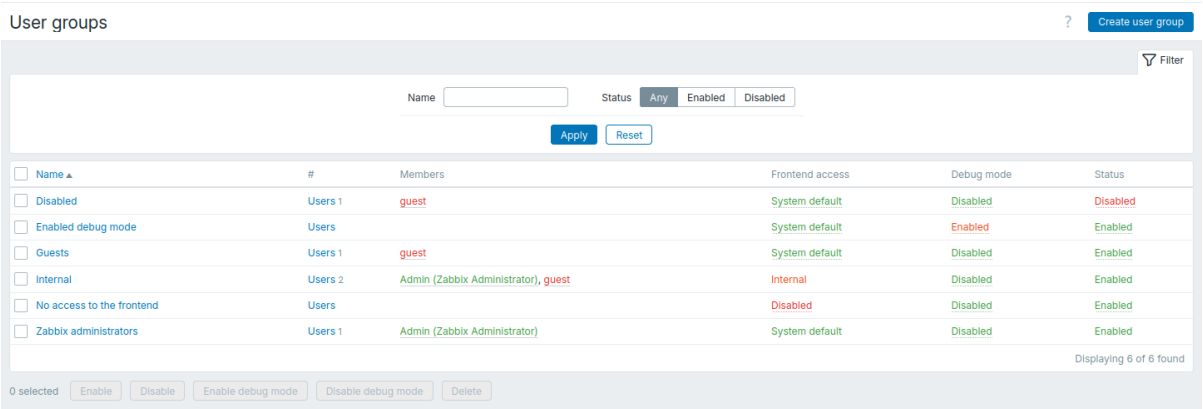
ユーザーグループ

概要

Users > *User groups* セクションでは、システムのユーザーグループを管理します。

ユーザーグループ

既存のユーザーグループとその詳細が一覧表示されます。



表示されるデータ:

列	説明
名前	ユーザーグループの名前。ユーザーグループ名をクリックすると、ユーザーグループ設定フォームが開きます。
#	グループ内のユーザー数。ユーザーをクリックすると、ユーザー一覧に該当するユーザーがフィルタリングされて表示されます。
メンバー	ユーザーグループに所属する個々のユーザーのユーザー名（括弧内に名前と姓）。ユーザー名をクリックすると、ユーザー設定フォームが開きます。無効なグループのユーザーは赤色で表示されます。
Web インターフェースへのアクセス	Web インターフェースへのアクセスレベルが表示されます: システムデフォルト - ユーザーは、グローバルに設定された認証方式に応じて、Zabbix、LDAP、または HTTP によって認証されます。 内部 - ユーザーは Zabbix によって認証されます。HTTP 認証がグローバルなデフォルトの場合は無視されます。 **LDAP** - ユーザーは LDAP によって認証されます。HTTP 認証がグローバルなデフォルトの場合は無視されます。 無効 - このグループの Web インターフェースから Zabbix へのアクセスは禁止されます。 現在のレベルをクリックすると、変更できます。
デバッグモード	デバッグモードの状態（有効または無効）が表示されます。状態をクリックすると、変更できます。
ステータス	ユーザーグループのステータス（有効または無効）が表示されます。ステータスをクリックすると、変更できます。

新しいユーザーグループを設定するには、右上隅にあるユーザーグループの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

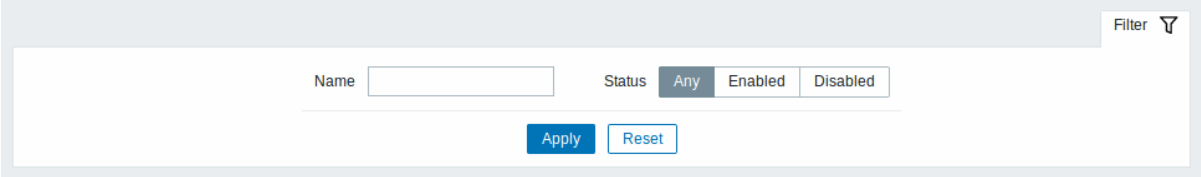
- *Enable* - ユーザーグループのステータスを *Enabled* に変更します
- *Disable* - ユーザーグループのステータスを *Disabled* に変更します
- *Enable debug mode* - ユーザーグループのデバッグモードを有効にします
- *Disable debug mode* - ユーザーグループのデバッグモードを無効にします
- *Delete* - ユーザーグループを削除します

これらのオプションを使用するには、該当するユーザーグループの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるユーザーグループだけを表示できます。検索性能を向上させるため、データはマクロが未解決の状態で検索されます。

Filter リンクは、ユーザーグループ一覧の上にあります。これをクリックするとフィルターが表示され、名前とステータスでユーザーグループを絞り込めます。



ユーザー権限ロール

概要

ユーザー > ユーザーの役割セクションでは、ユーザーの役割を作成できます。

ユーザーの役割を使用すると、最初に選択したユーザータイプ（ユーザー、管理者、特権管理者）に基づいて、きめ細かな権限を作成できます。

ユーザータイプを選択すると、そのユーザータイプで利用可能なすべての権限が付与されます（デフォルトでチェックされています）。

権限は、ユーザータイプで利用可能な権限の範囲内でのみ取り消すことができます。ユーザータイプで利用可能な範囲を超えて権限を拡張することはできません。

利用できない権限のチェックボックスはグレー表示されます。ユーザーは、ブラウザーにその要素への直接 URL を入力した場合でも、その要素にアクセスできません。

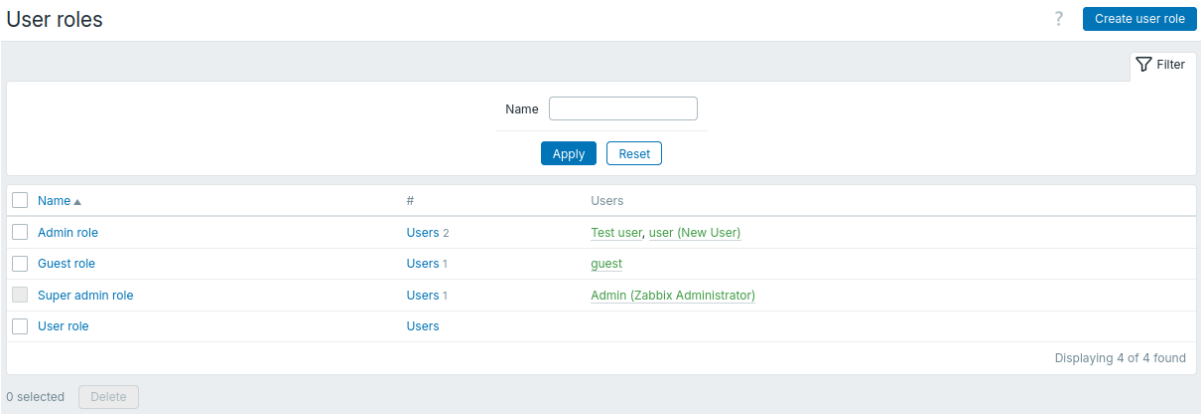
Note:
一部の UI 要素へのアクセスを制限しても、そのページを開けなくなるだけであり、Web インターフェースの他の部分から基盤となるデータにアクセスする機能が削除されるわけではありません。

ユーザーの役割は、システムユーザーに割り当てることができます。各ユーザーに割り当てられる役割は 1 つだけです。

デフォルトのユーザーロール

デフォルトでは、Zabbix には事前定義された権限セットを持つ 4 つのユーザーロールが設定されています。

- Guest ロール
- User ロール
- Admin ロール
- Super admin ロール



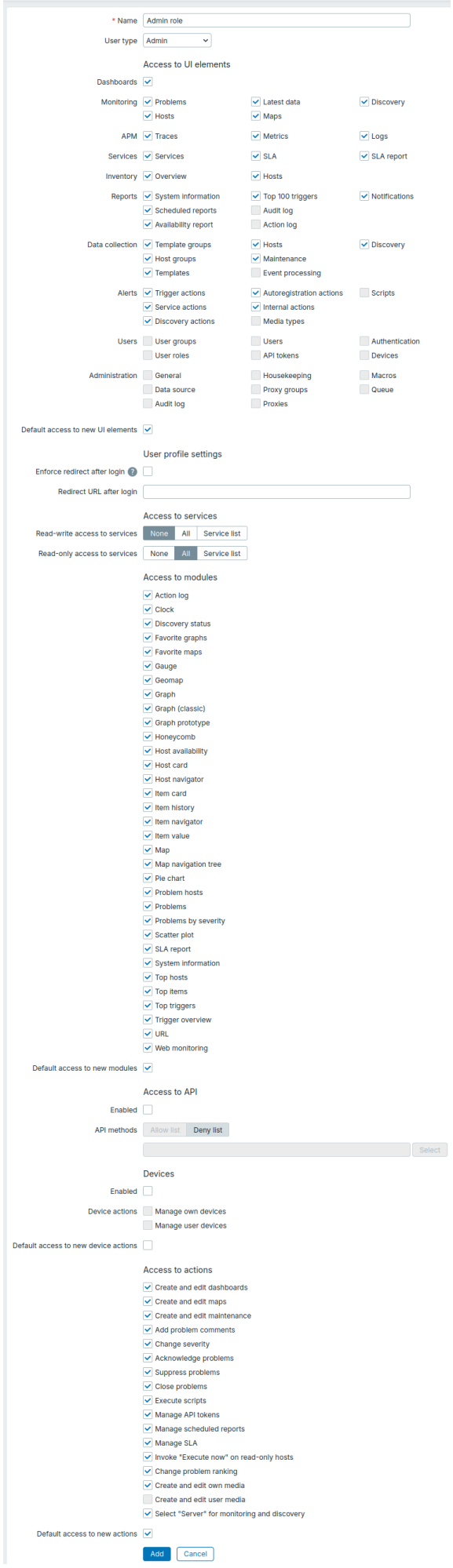
これらは、Zabbix の主なユーザータイプに基づいています。それぞれのロールが割り当てられているすべてのユーザーの一覧が表示されます。無効なグループに含まれているユーザーは赤色で表示されます。*Guest* ロールは、一部の Web インターフェースセクションを表示する権限のみを持つユーザータイプのロールです。

Note:
デフォルトの *Super admin* ロールは変更または削除できません。これは、無制限の権限を持つ Super admin ユーザーが少なくとも 1 人、Zabbix に存在する必要があるためです。*Super admin* タイプのユーザーは、自分のロールの設定を変更できますが、ユーザータイプは変更できません。

設定

新しいロールを作成するには、右上隅にある ユーザーロールを作成ボタンをクリックします。既存のロールを更新するには、ロール名をクリックして設定フォームを開きます。

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

利用可能な権限が表示されます。特定の権限を取り消すには、そのチェックボックスの選択を解除します。

Zabbix で事前設定されている各ユーザーロールのデフォルト設定と利用可能な権限について、以下で説明します。

デフォルトの権限

UI 要素へのアクセス

メニューセクションへのデフォルトのアクセス権は、ユーザータイプによって異なります。詳細については、権限のページを参照してください。

その他のオプションへのアクセス

以下の表に記載されているセクションの一部は非表示になる場合があります。

- Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['modules_config_enabled']` が `false` に設定されている場合、モジュールへのアクセスは非表示になります。
- `EnableMobileDevices` が無効になっている場合、デバイスは非表示になります。

パラメータ		説明	デフォルトのユーザーロール スーパーユーザーロール	管理者ロール	ユーザーロール	ゲストロール
新しい *UI* 要素へのデフォルトアクセス		このオプションでは、Zabbix のアップグレード後に新しいメニューセクションへどのようにアクセスできるかを指定します。既存のモジュールのメニューセクションには影響しません。	はい	はい	はい	はい
サービスへのアクセス	サービスへの読み書きアクセス	サービスへの読み書きアクセスを選択します。 なし - アクセスなし すべて - すべてのサービスへの読み書きアクセス サービスリスト - 読み書きアクセスを許可するサービスを選択 読み書きアクセスが許可されている場合、読み取り専用アクセスの設定より優先され、子サービスに動的に継承されます。	すべて	すべて	なし	なし

	タグによるサービスへの読み書きアクセス	タグ名と、必要に応じて値を指定して、タグに一致するサービスへの読み書きアクセスを追加で許可します。 このオプションは、サービスへの読み書きアクセスパラメータで サービスリストが選択されている場合に使用できます。 読み書きアクセスが許可されている場合、読み取り専用アクセスの設定より優先され、子サービスに動的に継承されます。				
	サービスへの読み取り専用アクセス	サービスへの読み取り専用アクセスを選択します。 なし - アクセスなし すべて - すべてのサービスへの読み取り専用アクセス サービスリスト - 読み取り専用アクセスを許可するサービスを選択 読み取り専用アクセスは読み書きアクセスより優先されず、子サービスに動的に継承されます。			すべて	すべて
	タグによるサービスへの読み取り専用アクセス	タグ名と、必要に応じて値を指定して、タグに一致するサービスへの読み取り専用アクセスを追加で許可します。 このオプションは、サービスへの読み取り専用アクセスパラメータで サービスリストが選択されている場合に使用できます。 読み取り専用アクセスは読み書きアクセスより優先されず、子サービスに動的に継承されます。				
モジュールへのアクセス						
	< モジュール名 >	特定のモジュールへのアクセスを許可または拒否します。有効なモジュールのみがこのセクションに表示されます。現在無効になっているモジュールへのアクセスを許可または制限することはできません。	はい	はい	はい	はい
	新しいモジュールへのデフォルトアクセス	このオプションでは、Zabbix のアップグレード後に新しいモジュールおよびウィジェットへどのようにアクセスできるかを指定します。管理 > 一般 > モジュールセクションで追加されたモジュールおよびウィジェットにも適用されます。				
API へのアクセス						
	有効	API へのアクセスを有効または無効にします。	いいえ	いいえ	いいえ	いいえ

	API メソッド	検索フィールドで指定した API メソッドを許可するには 許可リストを、拒否するには 拒否リストを選択します。一部の API メソッドを許可し、他の API メソッドを拒否することはできません。 検索フィールドにメソッド名の入力を開始し、オートコンプリートリストからメソッドを選択します。 選択ボタンをクリックして、このユーザータイプで使用可能な完全なリストからメソッドを選択することもできます。なお、アクションへのアクセスブロックで特定のアクションのチェックを外すと、そのアクションに関連する API メソッドをユーザーは使用できなくなります。 ワイルドカードがサポートされています。例: `dashboard.*`（'dashboard.' API サービスのすべてのメソッド）、`*`（任意のメソッド）、`*.export`（すべての API サービスにおける、名前に'.export' を含むメソッド）。 メソッドが指定されていない場合、許可/拒否リストのルールは無視されます。				
アクションへのアクセス	ダッシュボードの作成と編集	このチェックボックスをオフにすると、対応する要素に対する `.create`、`.update`、`.delete` API メソッドを使用する権限も取り消されます。	はい	はい	はい	いいえ
	マップの作成と編集					
	メンテナンスの作成と編集				いいえ	
	障害コメントの追加	このチェックボックスをオフにすると、`event.acknowledge` API メソッドによって対応するアクションを実行する権限も取り消されます。			はい	
	深刻度の変更					
	障害の承認					
	障害の抑制					
	障害のクローズ					
	スクリプトの実行	このチェックボックスをオフにすると、`script.execute` API メソッドを使用する権限も取り消されます。				
	API トークンの管理	このチェックボックスをオフにすると、すべての `token.` API メソッドを使用する権限も取り消されます。				
	スケジュールレポートの管理	このチェックボックスをオフにすると、すべての `report.` API メソッドを使用する権限も取り消されます。			いいえ	
	SLA の管理	SLAを管理する権限を有効または無効にします。				
	読み取り専用ホストで「今すぐ実行」を実行	読み取り専用ホスト上のアイテムに対して、ユーザーが最新データの「今すぐ実行」オプションを使用できるようにします。			はい	
	障害のランキングの変更	ユーザーが障害のランキングを原因から症状へ、またはその逆に変更できるようにします。				

	自身のメディアの作成と編集	ユーザーが自身のメディアを作成および編集できるようにします。				
	ユーザーメディアの作成と編集	ユーザーが他のユーザーのメディアを作成および編集できるようにします。このオプションはスーパーユーザーにのみ使用できます。		いいえ	いいえ	
	監視とディスカバリに「サーバー」を選択	このチェックボックスをオフにすると、サーバーによって監視されるホストの作成、サーバーによって実行されるネットワークディスカバリルールの作成、およびサーバーを使用したアイテムのテストが制限されます。監視方法パラメータのホスト設定、ディスカバリ方法パラメータのネットワークディスカバリルール設定、および テスト方法パラメータのアイテムのテストで、サーバーオプションが無効になります。		はい	いいえ	
	新しいアクションへのデフォルトアクセス	このオプションでは、Zabbix のアップグレード後に新しいアクションへどのようにアクセスできるかを指定します。		はい	はい	
デバイス						
	有効	ユーザー設定 > デバイス（すべてのユーザーがアクセス可能）および ユーザー > デバイス（スーパーユーザーのみアクセス可能）へのアクセスを有効または無効にします。	はい	いいえ	いいえ	いいえ
	自身のデバイスの管理	ユーザーが ユーザー設定 > デバイスで自身のデバイスを管理できるようにします。				
	ユーザーデバイスの管理	ユーザーが ユーザー > デバイスで他のユーザーに属するデバイスを管理できるようにします。				
	新しいアクションへのデフォルトアクセス	このオプションでは、Zabbix のアップグレード後に新しいアクションへどのようにアクセスできるかを指定します。				

関連項目:

- ユーザーの設定

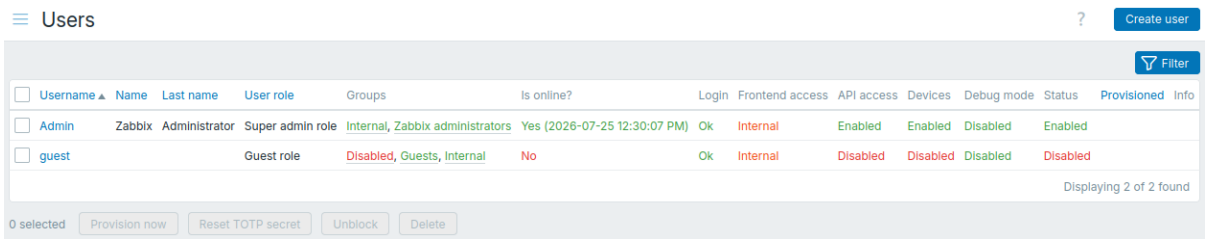
ユーザー

概要

ユーザー > ユーザーセクションでは、システムのユーザーを管理します。

ユーザー

既存のユーザーとその詳細の一覧が表示されます。



表示されるデータ:

列	説明
ユーザー名	Zabbix へのログインに使用するユーザー名。ユーザー名をクリックすると、ユーザー設定フォームが開きます。
名前	ユーザーの名。
姓	ユーザーの姓。
ユーザーの役割	ユーザーの役割が表示されます。

列	説明
グループ	ユーザーが所属するグループが一覧表示されます。ユーザーグループ名をクリックすると、ユーザーグループ設定フォームが開きます。無効なグループは赤色で表示されます。
オンライン?	ユーザーのオンライン状態（はいまたはいいえ）が表示されます。括弧内には、最後にユーザーがアクティビティを行った時刻が表示されます。
ログイン	ユーザーのログイン状態（*OK* またはブロック）が表示されます。管理 > 一般 > その他セクションで設定されたログイン失敗回数（デフォルトは 5 回）を超えると、ユーザーは一時的にブロックされます。ブロックをクリックすると、ユーザーのブロックを解除できます。
Web インターフェースへのアクセス	ユーザーグループ全体に設定された値に応じて、Web インターフェースへのアクセスレベル（システムデフォルト、内部、*LDAP*、または無効）が表示されます。
API アクセス	ユーザーの役割に設定された値に応じて、API アクセスの状態（有効または無効）が表示されます。
デバイス	ユーザーの役割に設定された値に応じて、デバイスアクセスの状態（有効または無効）が表示されます。 この列は、`EnableMobileDevices`が有効な場合に表示されます。
デバッグモード	ユーザーグループ全体に設定された値に応じて、デバッグモードの状態（有効または無効）が表示されます。
ステータス	ユーザーグループ全体に設定された値に応じて、ユーザーのステータス（有効または無効）が表示されます。
プロビジョニング済み	ユーザーが最後にプロビジョニングされた日付が表示されます。 LDAP/SAML からの JIT プロビジョニングによって作成されたユーザーに使用されます。
情報	エラーに関する情報が表示されます。 ユーザーグループに所属していないユーザーには黄色の警告が表示されます。 役割がないユーザー、および役割とユーザーグループの両方がないユーザーには赤色の警告が表示されます。

新しいユーザーを設定するには、右上隅にあるユーザーの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、次の一括編集オプションを使用できます。

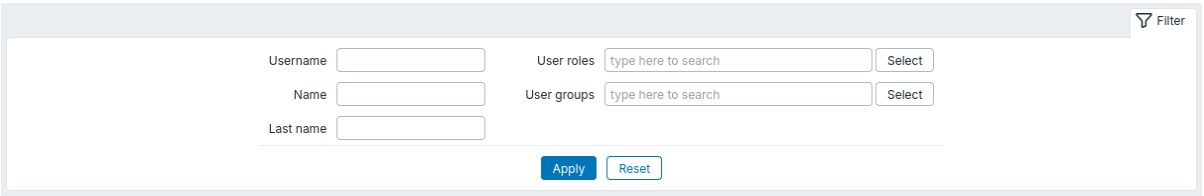
- 今すぐプロビジョニング - LDAP からユーザー情報を更新します（このオプションは、LDAPユーザーが選択されている場合にのみ有効になります）
- *TOTP* シークレットをリセット - すべての TOTP 方式のユーザー TOTP シークレットをリセットし、ユーザーセッションを削除します（MFAが有効になっている場合にのみ、このオプションは有効になります。TOTP シークレットを持たないユーザーの場合、セッションは削除されません）
- ブロック解除 - ブロックされたユーザーのシステムアクセスを再び有効にします
- 削除 - ユーザーを削除します

これらのオプションを使用するには、対象ユーザーの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるユーザーのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決の状態で検索されます。

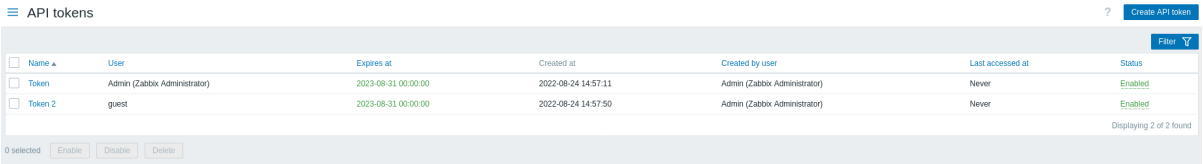
ユーザーの一覧の上にある フィルターリンクを使用できます。クリックするとフィルターが表示され、ユーザー名、名前、姓、ユーザーロール、ユーザーグループでユーザーを絞り込めます。



API トークン

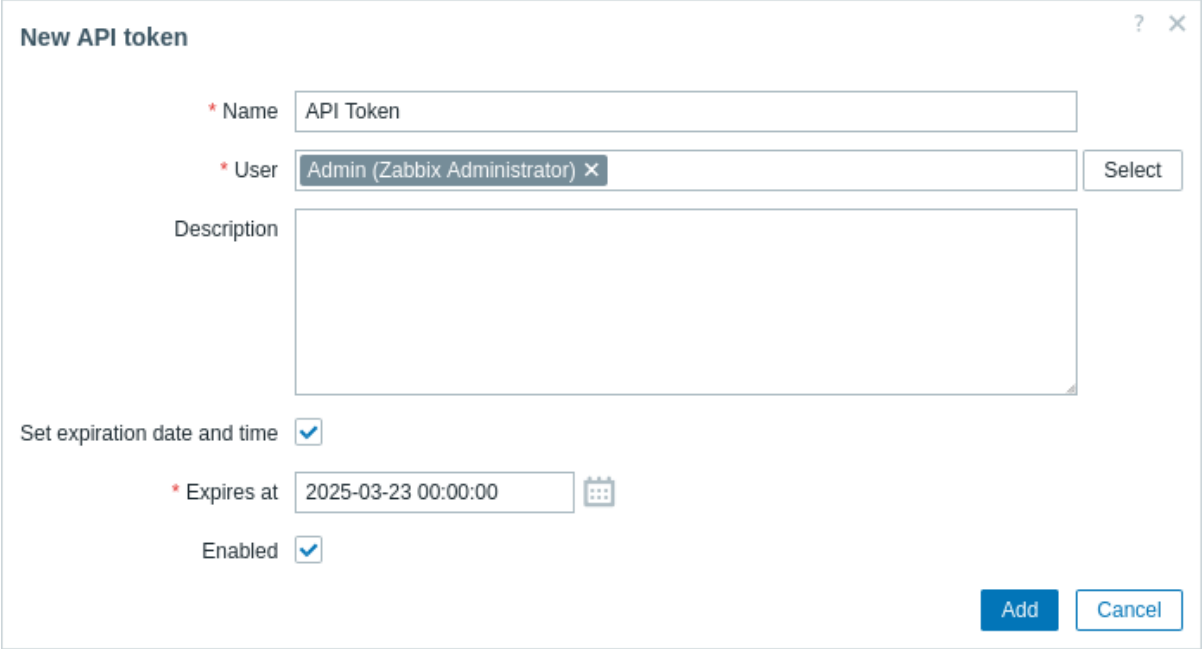
概要

このセクションでは、API トークンを作成および管理できます。
 API トークンは、Zabbix API での認証に使用されます。
 各 API トークンは 1 人のユーザーに割り当てられ、そのユーザーの権限を継承します。API トークンで認証されたリクエストは、関連付けられたユーザーのロールと権限に従って認可されます。



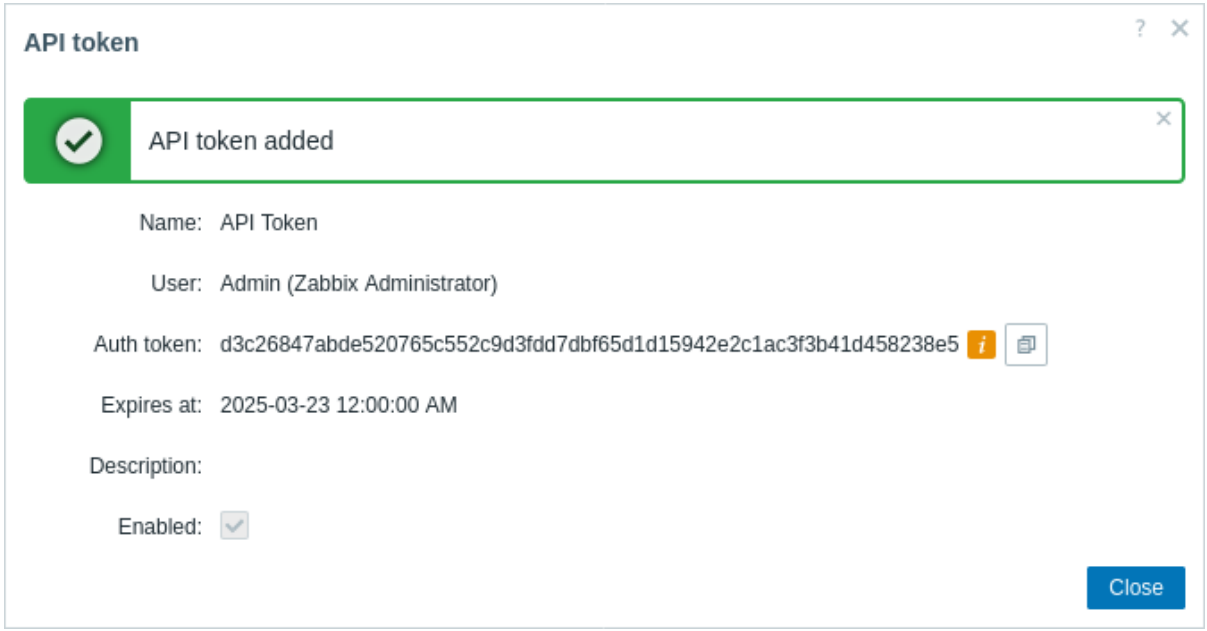
API トークンは、名前、トークンが割り当てられているユーザー、有効期限、トークンを作成したユーザー、またはステータス（有効/無効）でフィルタリングできます。リスト内のトークンのステータスをクリックすると、すぐにトークンを有効/無効にできます。また、リストでトークンを選択してから、リストの下にある有効化/無効化ボタンをクリックすることで、複数のトークンを一括して有効/無効にすることもできます。

新しいトークンを作成するには、右上隅にある *API* トークンの作成ボタンをクリックし、トークン設定画面で必須フィールドに入力します。



パラメーター	説明
名前	トークンの表示名。
ユーザー	トークンを割り当てるユーザー。ユーザーをすばやく選択するには、ユーザー名、名、または姓の入力を開始し、オートコンプリートリストから必要なユーザーを選択します。または、選択ボタンをクリックして、ユーザーの一覧からユーザーを選択できます。トークンを割り当てられるユーザーは 1 人だけです。
説明	任意のトークンの説明。
有効期限の日時を設定	トークンに有効期限を設定しない場合は、このチェックボックスの選択を解除します。
有効期限	カレンダーアイコンをクリックしてトークンの有効期限を選択するか、YYYY-MM-DD hh:mm:ss 形式で日付を手動入力します。
有効	無効な状態でトークンを作成する場合は、このチェックボックスの選択を解除します。

追加をクリックしてトークンを作成します。



認証トークンの値をコピーして安全な場所に保存してから、ページを閉じる前に、閉じるをクリックします。トークンがリストに表示されます。

Warning:
認証トークンの値は、後から再度表示することはできません。トークンの作成直後にのみ使用できます。保存したトークンを紛失した場合は、再生成する必要があります。再生成すると、新しい認証文字列が作成されます。

トークン名をクリックすると、名前、説明、有効期限の設定、またはトークンのステータスを編集できます。トークンが割り当てられているユーザーを変更することはできません。更新ボタンをクリックして変更を保存します。トークンを紛失または漏えいした場合は、再生成ボタンをクリックして新しいトークン値を生成できます。操作の実行後は以前に生成されたトークンが無効になるため、この操作の確認を求めるダイアログボックスが表示されます。

管理メニューセクションへのアクセス権がないユーザーは、ユーザーロールの権限で *API* トークンの管理が許可されている場合に限り、ユーザープロフィール > *API* トークンセクションで、自分に割り当てられたトークンの詳細を表示および変更できます。

認証

概要

ユーザー > 認証セクションでは、Zabbix のユーザー認証方式と内部パスワードの要件を指定できます。

利用可能な認証方式は、内部認証、HTTP 認証、LDAP 認証、SAML 認証、MFA 認証です。

デフォルトの認証

デフォルトでは、Zabbix はすべてのユーザーに対して内部 Zabbix 認証を使用します。

デフォルトの認証方式は、システム全体でLDAPに変更できます。変更するには、*LDAP* タブに移動して LDAP パラメータを設定し、認証タブに戻って、デフォルトの認証セレクターを LDAP に切り替えます。

認証方式は、ユーザーグループレベルで細かく調整できます。LDAP 認証がグローバルに設定されている場合でも、一部のユーザーグループでは Zabbix 認証を使用できます。これらのグループでは、Web インターフェースへのアクセスを「内部」に設定する必要があります。

内部認証がグローバルに使用されている場合でも、特定のユーザーグループに対してのみ LDAP 認証を有効にできます。この場合、LDAP 認証の詳細を指定して、特定のユーザーグループで使用できます。その場合、対象となるユーザーグループのWeb インターフェースへのアクセスを LDAP に設定する必要があります。LDAP 認証を使用するユーザーグループに 1 つでも所属しているユーザーは、内部認証方式を使用できません。

HTTP、SAML 2.0、および MFA 認証方式は、デフォルトの認証方式に加えて使用できます。

Zabbix は、ジャストインタイム（JIT）プロビジョニングをサポートしています。これにより、外部ユーザーが初めて認証したときに Zabbix でユーザーアカウントを作成し、プロビジョニングできます。JIT プロビジョニングは LDAP および SAML でサポートされています。

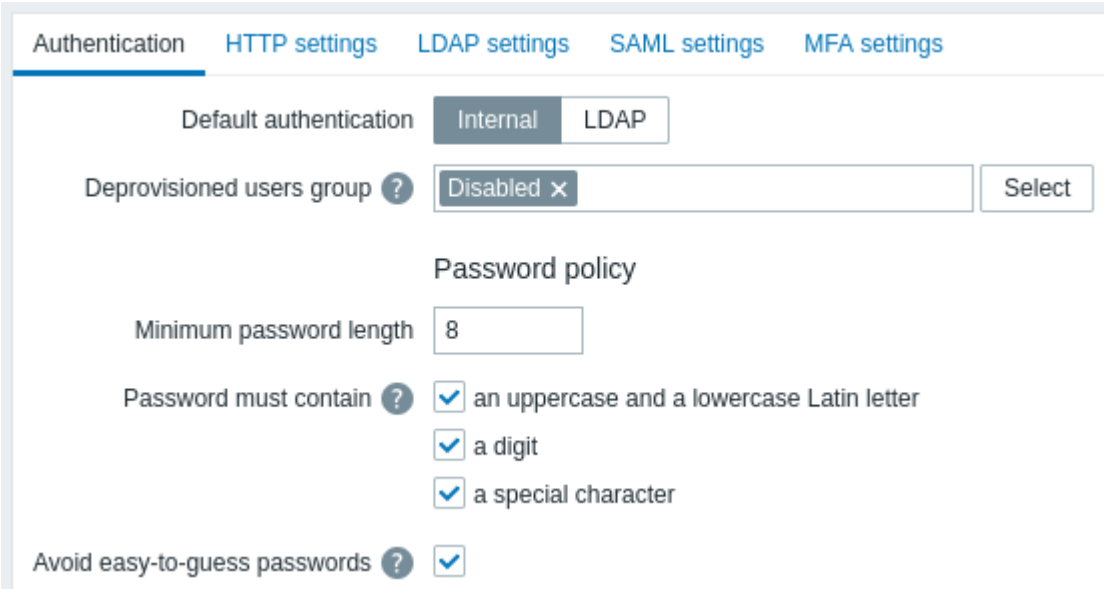
関連項目:

- HTTP 認証

- LDAP 認証
- SAML 認証
- MFA 認証

設定

認証タブでは、デフォルトの認証方式、プロビジョニング解除されたユーザーのグループ、および Zabbix ユーザーのパスワードの複雑性要件を設定できます。



設定パラメータ:

パラメータ	説明
デフォルトの認証	Zabbix のデフォルトの認証方式として、内部または *LDAP* を選択します。
プロビジョニング解除されたユーザーのグループ	プロビジョニング解除されたユーザーのユーザーグループを指定します。この設定は、LDAP または SAML システムから Zabbix に作成されたものの、プロビジョニングする必要がなくなったユーザーに関する JIT プロビジョニングの場合にのみ必要です。 無効なユーザーグループを指定する必要があります。
パスワードの最小長	デフォルトでは、パスワードの最小長は 8 に設定されています。サポートされる範囲: 1～70。72 文字を超えるパスワードは切り詰められることに注意してください。
パスワードに含める必要がある文字	パスワードに指定した文字を使用するようにするには、1 つ以上のチェックボックスを選択します: - 大文字と小文字のラテン文字 - 数字 - 特殊文字 疑問符の上にマウスカーソルを置くと、各オプションで使用できる文字の一覧がヒントとして表示されます。
推測されやすいパスワードを避ける	選択すると、パスワードが次の要件に照らして確認されます: - ユーザーの名前、姓、またはユーザー名を含めてはならない - 一般的なパスワードまたはコンテキスト固有のパスワードであってはならない 一般的なパスワードとコンテキスト固有のパスワードの一覧は、NCSC の「Top 100k passwords」、SecLists の「Top 1M passwords」、および Zabbix のコンテキスト固有のパスワードの一覧から自動的に生成されます。内部ユーザーは、この一覧に含まれるパスワードを設定できません。このようなパスワードは広く使用されているため、脆弱と見なされます。

パスワードの複雑性要件を変更しても、既存のユーザーのパスワードには影響しません。ただし、既存のユーザーがパスワードの変更を選択した場合、新しいパスワードは現在の要件を満たす必要があります。要件の一覧を含むヒントは、ユーザープロファイルのパスワードフィールドの横、およびユーザー > ユーザーメニューからアクセスできるユーザー設定フォームに表示されます。

HTTP

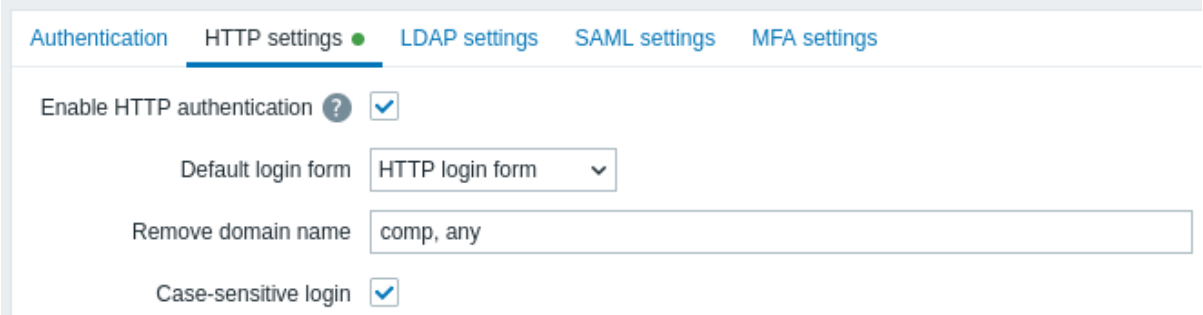
概要

HTTP または Web サーバーベースの認証 （例：BasicAuthentication、NTLM/Kerberos）を使用して、ユーザー名とパスワードを確認できます。ただし、ユーザーは Zabbix にも存在している必要があります。ただし、その Zabbix パスワードは使用されません。

Attention:
注意してください。HTTP サーバー認証を有効にする前に、設定が完了し、正常に動作していることを確認してください。

Note:
HTTP 認証の設定は、Web インターフェース設定ファイル（zabbix.conf.php）で `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` を設定すると無効にできます。この場合、HTTP 認証オプションを含む *HTTP* 設定タブは非表示になります。

設定



設定パラメータ:

パラメータ	説明
HTTP 認証を有効にする	HTTP 認証を有効にするには、チェックボックスを選択します。にマウスを合わせると、Web サーバー認証を使用する場合、Web インターフェースへのアクセスを LDAP/Internal に設定しているユーザーを含むすべてのユーザーが、Zabbix ではなく Web サーバーによって認証されることを警告するヒントボックスが表示されます。
デフォルトのログインフォーム	認証されていないユーザーのリダイレクト先を指定します。 **Zabbix** ログインフォーム - 標準の Zabbix ログインページ。 **HTTP** ログインフォーム - HTTP ログインページ。 Web サーバーを使用した認証は、`index_http.php` ページに対してのみ有効にすることを推奨します。デフォルトのログインフォームを「HTTP ログインページ」に設定した場合、Web サーバーの認証モジュールが `$_SERVER` 変数に有効なユーザーログインを設定すると、ユーザーは自動的にログインします。 サポートされる `$_SERVER` キーは、`PHP_AUTH_USER`、`REMOTE_USER`、`AUTH_USER` です。
ドメイン名を削除	ユーザー名から削除するドメイン名をカンマ区切りで指定します。 例: `comp,any` - ユーザー名が「Admin@any」または「comp\Admin」の場合、ユーザーは「Admin」としてログインします。「notacompany\Admin」の場合、ログインは拒否されます。
大文字と小文字を区別するログイン	ユーザー名の大文字と小文字を区別するログインを無効にするには、チェックボックスの選択を解除します（デフォルトでは有効）。 大文字と小文字を区別するログインを無効にすると、Zabbix ユーザーが「Admin」または「ADMIN」であっても、たとえば「admin」としてログインできます。 大文字と小文字を区別するログインを無効にし、類似したユーザー名（例: Admin と admin）を持つ Zabbix ユーザーが複数存在する場合、次のエラーメッセージが表示され、これらのユーザーのログインは常に拒否されることに注意してください: 「Authentication failed: supplied credentials are not unique.」

Note:
HTTP 認証情報を使用してログインできず（HTTP ログインフォームをデフォルトに設定している場合）、401 エラーが発生する内部ユーザーについては、基本認証ディレクティブに `ErrorDocument 401 /index.php?form=default` の行を追加することを検討してください。これにより、通常の Zabbix ログインフォームにリダイレクトされます。

LDAP

概要

外部 LDAP認証を使用して、ユーザー名とパスワードを確認できます。

Zabbix の LDAP 認証は、少なくとも Microsoft Active Directory および OpenLDAP で動作します。

LDAP サインインのみが設定されている場合、ユーザーは Zabbix にも存在する必要がありますが、その Zabbix パスワードは使用されません。認証が成功すると、Zabbix は LDAP から返されたユーザー名属性とローカルのユーザー名を照合します。

ユーザーのプロビジョニング

LDAP ユーザーに対して、JIT（Just-In-Time）ユーザーのプロビジョニングを設定できます。この場合、ユーザーが Zabbix にすでに存在している必要はありません。ユーザーアカウントは、ユーザーが初めて Zabbix にログインしたときに作成できます。

LDAP ユーザーが LDAP のログイン名とパスワードを入力すると、Zabbix はこのユーザーがデフォルトの LDAP サーバーに存在するかどうかを確認します。ユーザーが存在し、まだ Zabbix にアカウントを持っていない場合、Zabbix に新しいユーザーが作成され、ユーザーはログインできるようになります。

JIT プロビジョニングによって作成されたユーザーは、作成時にデフォルトとして設定されていた LDAP サーバー（ディレクトリ）に関連付けられます。後からデフォルトの LDAP サーバーを変更しても、すでにプロビジョニングされたユーザーにリンクされている LDAP サーバーは変更または更新されません。

> **Attention:**
> JIT プロビジョニングが有効な場合、認証タブでプロビジョニング解除されたユーザー用のユーザーグループを指定する必要があります。

JIT プロビジョニングを使用すると、LDAP の変更に基づいて、プロビジョニングされたユーザーアカウントを更新することもできます。たとえば、ユーザーがある LDAP グループから別の LDAP グループに移動された場合、Zabbix でも別のグループに移動されます。ユーザーが LDAP グループから削除された場合、Zabbix でもそのグループから削除され、他のグループにも所属していなければ、プロビジョニング解除されたユーザー用のユーザーグループに追加されます。プロビジョニングされたユーザーアカウントは、設定されたプロビジョニング期間に基づいて、またはユーザーが Zabbix にログインしたときに更新されます。

バックグラウンドプロビジョニングは、ユーザーが Web インターフェースを操作している場合、または少なくともブラウザーで Web インターフェースのページを開いている場合に、Zabbix の Web インターフェースによって実行されることに注意してください。ユーザーをプロビジョニングする専用のバックグラウンドプロセスはありません。

LDAP では、認証および検索のためにディレクトリにバインドする方法として、次の 3 つがサポートされています。

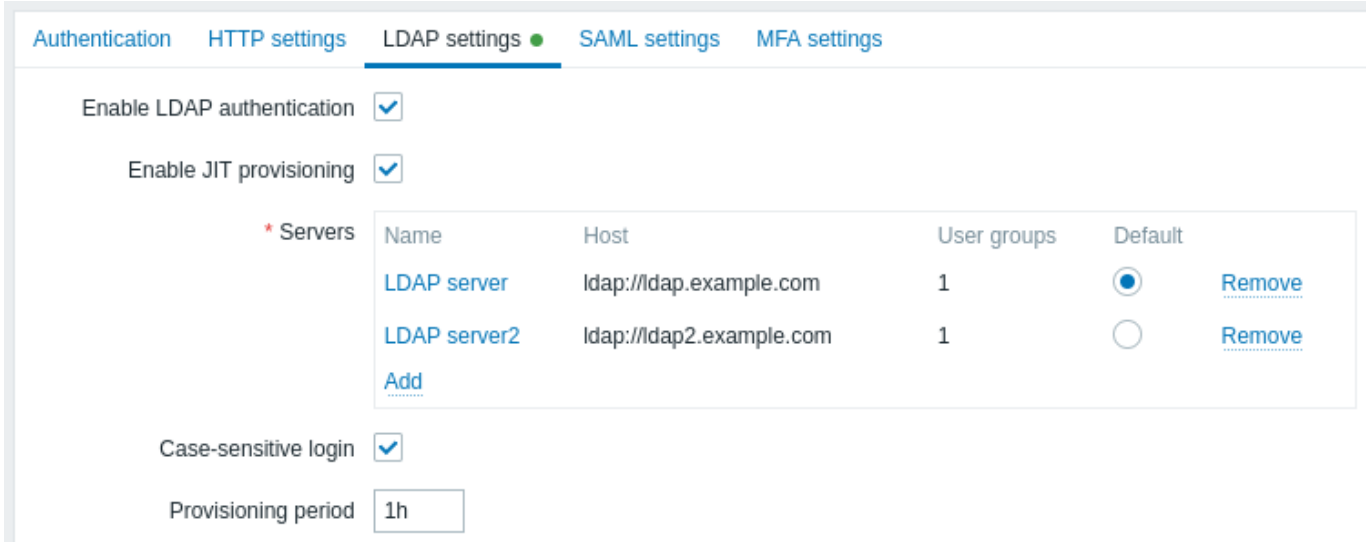
- 匿名バインド - *Bind DN* / *Bind password* を指定せず、LDAP サーバーが匿名クエリを許可する方法です。
- 専用バインドユーザー（サービスアカウント）- 特定の LDAP アカウントを *Bind DN* / *Bind password* に設定し、Zabbix が検索およびプロビジョニングに使用する方法です。これは、Zabbix がエンドユーザーの認証情報なしで検索およびバックグラウンドプロビジョニングを実行できるため、推奨される最も柔軟な方法です。
- 直接ユーザーバインド - ユーザーがログイン時に入力した認証情報を使用して Zabbix がバインドを実行する方法です（*Bind DN* / *Bind password* は設定しません）。これは、Base DN に `uid=%{user}` のようなプレースホルダーを含めることで設定します。このモードでは、Zabbix は対話型ログイン中にのみユーザーのパスワードにアクセスできます。そのため、ユーザーのログインセッション外で LDAP への認証を必要とするプロビジョニング操作（たとえば、今すぐプロビジョニングボタンの使用や、ユーザーがアクティブにログインしていない状態でのバックグラウンドプロビジョニングの実行）は認証できず、機能しません。直接ユーザーバインドでのプロビジョニングおよび更新は、ユーザーがログインした時点でのみ実行されます。

複数のサーバー

必要に応じて、複数の LDAP サーバーを定義できます。たとえば、異なるユーザーグループを認証するために異なるサーバーを使用できます。LDAP サーバーが設定されると、ユーザーグループの設定で、該当するユーザーグループに必要な LDAP サーバーを選択できるようになります。

ユーザーが複数のユーザーグループおよび複数の LDAP サーバーに属している場合、名前で昇順にソートされた LDAP サーバーのリストの最初のサーバーが認証に使用されます。

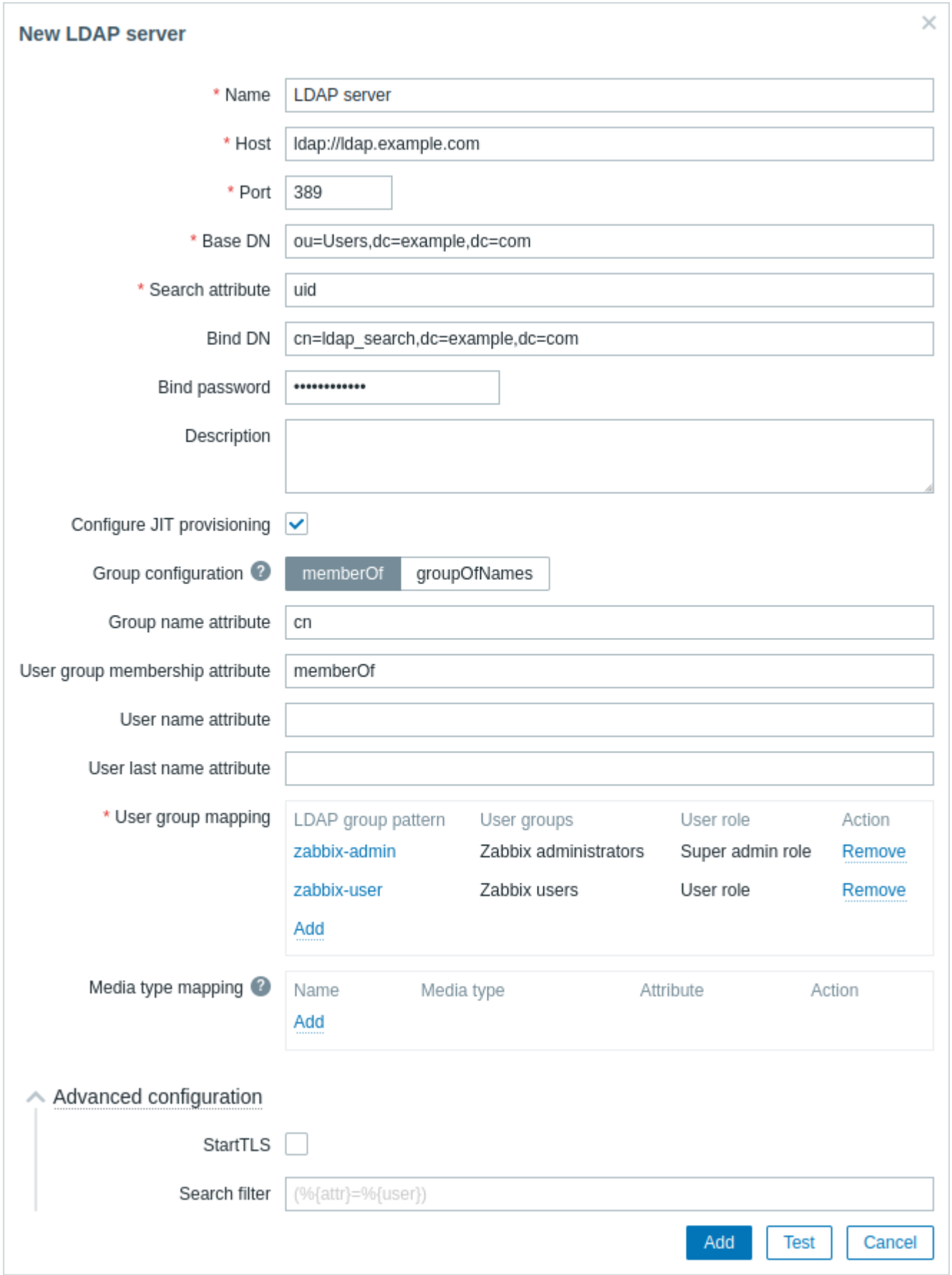
設定



設定パラメータ:

パラメータ	説明
LDAP 認証を有効にする	LDAP 認証を有効にするには、チェックボックスをオンにします。
JIT プロビジョニングを有効にする	JIT プロビジョニングを有効にするには、チェックボックスをオンにします。
サーバー	LDAP サーバーを設定するには、追加をクリックします（下記のLDAP サーバーの設定を参照）。
大文字と小文字を区別するログイン	ユーザー名の大文字と小文字を区別するログインを無効にするには、チェックボックスをオフにします（デフォルトでは有効）。 大文字と小文字を区別するログインを無効にすると、Zabbix ユーザーが「Admin」または「ADMIN」であっても、例えば「admin」としてログインできます。 大文字と小文字を区別するログインが無効で、類似したユーザー名（例: Admin と admin）を持つ Zabbix ユーザーが複数存在する場合、それらのユーザーのログインは常に拒否され、次のエラーメッセージが表示されます: 「Authentication failed: supplied credentials are not unique.」
プロビジョニング期間	プロビジョニング期間、つまり、ログインしたユーザーが Web インターフェースの操作中にプロビジョニングされる頻度を設定します。

LDAP サーバー設定



LDAP サーバーの設定パラメーター:

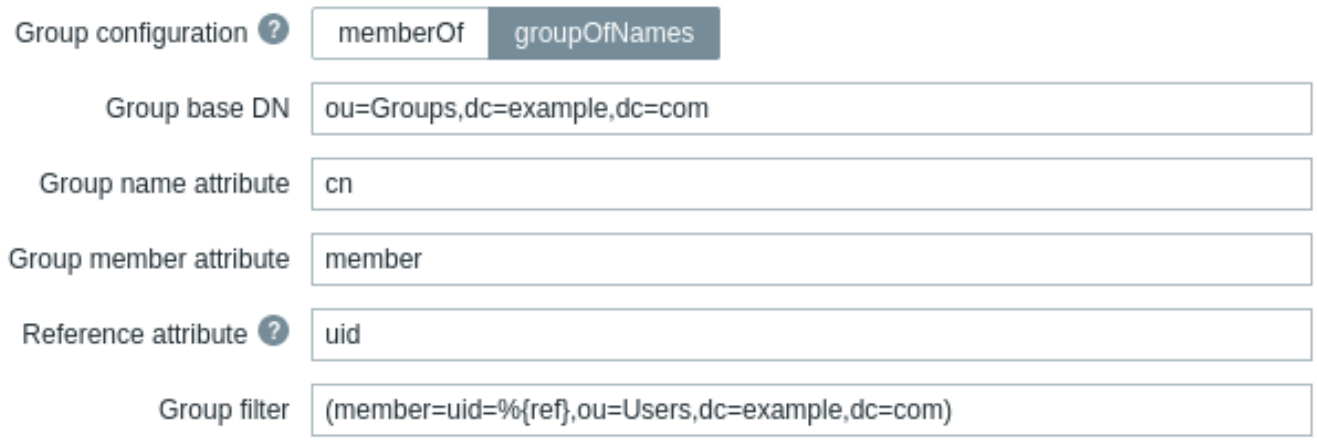
Parameter	Description
Name	Zabbix 設定における LDAP サーバーの名前。

Parameter	Description
Host	LDAP サーバーのホスト名、IP、または URI。例: ldap.example.com, 127.0.0.1, ldap://ldap.example.com セキュアな LDAP サーバーでは、*ldaps* プロトコルとホスト名を使用します。例: ldaps://ldap.example.com OpenLDAP 2.x.x 以降では、ldap://hostname:port または ldaps://hostname:port 形式の完全な LDAP URI を使用できます。
Port	LDAP サーバーのポート。デフォルトは 389 です。 セキュアな LDAP 接続では、通常ポート番号は 636 です。 完全な LDAP URI を使用する場合は使用されません。
Base DN	LDAP サーバー上のユーザーアカウントへのベースパス: ou=Users,ou=system (OpenLDAP の場合), DC=company,DC=com (Microsoft Active Directory の場合) uid=%{user},dc=example,dc=com (直接ユーザー bind の場合。下記の注記を参照)
Search attribute	検索に使用する LDAP アカウント属性: uid (OpenLDAP の場合), sAMAccountName (Microsoft Active Directory の場合)
Bind DN	LDAP サーバーへの bind および検索に使用する LDAP アカウント。例: uid=ldap_search,ou=system (OpenLDAP の場合), CN=ldap_search,OU=user_group,DC=company,DC=com (Microsoft Active Directory の場合) 匿名 bind もサポートされています。なお、匿名 bind ではドメイン設定が未認証ユーザーに公開される可能性があります（ユーザー、コンピューター、サーバー、グループ、サービスなどの情報）。 セキュリティ上の理由から、LDAP ホストでの匿名 bind を無効にし、代わりに認証済みアクセスを使用してください。
Bind password	LDAP サーバーへの bind および検索に使用するアカウントの LDAP パスワード。
Description	LDAP サーバーの説明。
Configure JIT provisioning	このチェックボックスをオンにすると、JIT provisioning に関連するオプションが表示されます。
Group configuration	グループ設定方法を選択します: **memberOf** - ユーザーとそのグループ所属属性を検索する方法 **groupOfNames** - member 属性を通じてグループを検索する方法 memberOf のほうが高速なため推奨されます。LDAP サーバーが `memberOf` をサポートしていない場合、またはグループフィルタリングが必要な場合は groupOfNames を使用してください。
Group name attribute	`memberOf` 属性内のすべてのオブジェクトからグループ名を取得する属性を指定します（*User group membership attribute* フィールドを参照）。 グループ名はユーザーグループのマッピングに必要です。
User group membership attribute	ユーザーが所属するグループに関する情報を含む属性を指定します（例: `memberOf`）。 たとえば、memberOf 属性には次のような情報が含まれる場合があります: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` このフィールドは memberOf 方式でのみ使用できます。
User name attribute	ユーザーの名を含む属性を指定します。
User last name attribute	ユーザーの姓を含む属性を指定します。
User group mapping	LDAP ユーザーグループのパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングします。 これは、プロビジョニングされたユーザーに Zabbix でどのユーザーグループ/ロールが付与されるかを決定するために必要です。 マッピングを追加するには *Add* をクリックします。 *LDAP group pattern* フィールドはワイルドカードをサポートします。グループ名は既存のグループと一致する必要があります。 LDAP ユーザーが複数の Zabbix ユーザーグループに一致する場合、そのユーザーはそれらすべてのメンバーになります。 LDAP ユーザーが複数の Zabbix ユーザーロールに一致する場合、その中で最も高い権限レベルのロールが付与されます。
Media type mapping	通知送信用に、ユーザーの LDAP media 属性（例: email）を Zabbix ユーザーメディアにマッピングします（属性値はメディアの*Send to* フィールドとして使用されます）。
Advanced configuration	詳細設定オプションを表示するには、*Advanced configuration* 見出しをクリックします（下記参照）。
StartTLS	LDAP サーバーへの接続時に StartTLS 操作を使用する場合は、このチェックボックスをオンにします。サーバーが StartTLS をサポートしていない場合、接続は失敗します。 StartTLS は *ldaps* プロトコルを使用するサーバーでは使用できません。

Parameter	Description
Search filter	LDAP でユーザーを認証する際のカスタム文字列を定義します。次のプレースホルダーがサポートされています: %{attr} - 検索属性名 (uid, sAMAccountName) %{user} - 認証するユーザー名の値 たとえば、大文字小文字を区別しない LDAP または Microsoft Active Directory 環境で大文字小文字を区別する検索を行うには、次のように文字列を定義します: (%{attr}:caseExactMatch:=%{user}). フィルターをカスタマイズしない場合、LDAP は既定値 (%{attr}=%{user}) を使用します。

Note:
直接ユーザー **bind** 用に LDAP サーバーを設定するには、*Base DN* パラメーターに属性 uid=%{user} を追加し（例: *uid=%{user},dc=example,dc=com*）、*BindDN* と *Bind password* パラメーターは空のままにします。認証時、プレースホルダー%{user} はログイン時に入力されたユーザー名に置き換えられます。
 直接ユーザー bind では、Zabbix がユーザーの資格情報にアクセスできるのは対話型ログイン中のみです。そのため、対話型ログインなしで動作するプロビジョニングタスク（たとえば *Provision now* ボタンや`user.provision` API メソッド）は、直接ユーザー bind で認証したユーザーを無視します。これは、Zabbix がそのユーザーの代わりに bind できないためです。Web インターフェースでエンドユーザーのログインを必須にせずにプロビジョニングや LDAP 検索を動作させる必要がある場合は、匿名 bind または専用の bind ユーザー（サービスアカウント）を使用してください。

以下のフィールドは、*Group configuration* 方法として”groupOfNames” を使用する場合に固有です:



Parameter	Description
Group base DN	LDAP サーバー上のグループへのベースパス。
Group name attribute	指定したグループのベースパス内でグループ名を取得する属性を指定します。 グループ名はユーザーグループのマッピングに必要です。
Group member attribute	LDAP 内のグループメンバーに関する情報を含む属性を指定します（例: member）。
Reference attribute	グループフィルター用の参照属性を指定します（*Group filter* フィールドを参照）。 その後、ここで指定した属性の値を取得するために、グループフィルター内で %{ref} を使用します。
Group filter	ユーザーがメンバーであるグループを取得するためのフィルターを指定します。 たとえば、(member=uid=%{ref},ou=Users,dc=example,dc=com) は、グループの member 属性が uid=User1,ou=Users,dc=example,dc=com の場合に”User1” に一致し、”User1” がメンバーであるグループを返します。

Warning:
証明書に問題がある場合、セキュアな LDAP 接続（ldaps）を動作させるには、/etc/openldap/ldap.conf 設定ファイルに TLS_REQCERT allow 行を追加する必要がある場合があります。これにより、LDAP カタログへの接続のセキュリティが低下する可能性があります。

Note:
実際のユーザーアカウント（Zabbix Web インターフェースへのログインに使用するアカウント）ではなく、LDAP サーバーへの bind および検索を最小権限で実行するための別個の LDAP アカウント（*Bind DN*）を作成することを推奨します。
この方法はより安全であり、ユーザーが LDAP サーバー上で自分のパスワードを変更しても *Bind password* を変更する必要がありません。
 上の表では、これが *ldap_search* アカウント名です。

アクセスのテスト

テストボタンを使用すると、ユーザーアクセスをテストできます。

パラメーター	説明
ログイン	テストする LDAP ユーザー名（Zabbix の Web インターフェースに現在ログインしているユーザー名があらかじめ入力されています）。このユーザー名は LDAP サーバーに存在している必要があります。 テストユーザーを認証できない場合、Zabbix は LDAP 認証を有効にしません。
ユーザーパスワード	テストする LDAP ユーザーのパスワード。

SAML

概要

SAML 2.0 認証を使用して Zabbix にサインインできます。

SAML サインインが JIT ユーザープロビジョニングなしで設定されている場合、ユーザーは事前に Zabbix に存在している必要があります。ユーザーの Zabbix パスワードは認証には使用されません。SAML 認証が成功すると、Zabbix は SAML から受信した username 属性を既存のローカル Zabbix ユーザー名と照合します。

ユーザーのプロビジョニング

SAML ユーザーに対して JIT（ジャストインタイム）ユーザープロビジョニングを設定することができます。この場合、Zabbix にユーザーがすでに存在している必要はありません。ユーザーアカウントは、ユーザーが初めて Zabbix にログインしたときに作成できます。

Attention:
JIT プロビジョニングが有効な場合、認証タブでプロビジョニング解除されたユーザー用のユーザーグループを指定する必要があります。

JIT プロビジョニングに加えて、SCIM（System for Cross-domain Identity Management）プロビジョニング、つまりユーザープロビジョニングによって作成されたユーザーの継続的なユーザーアカウント管理を有効化および設定することも可能です。SCIM プロビジョニングには、Zabbix への認証のために Zabbix のAPI トークン（スーパ管理者権限付き）が必要です。

たとえば、ユーザーがある SAML グループから別のグループに移動された場合、Zabbix でもユーザーは一方のグループからもう一方のグループに移動されます。ユーザーが SAML グループから削除された場合、Zabbix でもそのグループから削除され、他のグループに属していない場合はプロビジョニング解除されたユーザー用のユーザーグループに追加されます。

SCIM が有効かつ設定されている場合、SAML ユーザーは Zabbix にログインした時点でプロビジョニングされ、その後 SAML の変更に基づいて継続的に更新されます。すでに存在する SAML ユーザーはプロビジョニングされず、プロビジョニングされたユーザーのみが更新されます。ユーザーがプロビジョニングまたは更新される際、有効なメディアのみがユーザーに追加されることに注意してください。

SCIM が有効でない場合、SAML ユーザーは Zabbix にログインした時点でプロビジョニング（および後で更新）されます。

Note:
SAML 認証が有効な場合、ユーザーはローカルログインまたは SAML シングルサインオンのいずれかを選択できます。JIT プロビジョニングを使用する場合、シングルサインオンのみが可能です。

アイデンティティプロバイダーの設定

Zabbix で動作させるためには、SAML アイデンティティプロバイダー (onelogin.com、auth0.com、okta.comなど) を以下のように設定する必要があります。

- *Assertion Consumer URL* には `<path_to_zabbix_ui>/index_sso.php?acs` を設定します
- *Single Logout URL* には `<path_to_zabbix_ui>/index_sso.php?sls` を設定します

<path_to_zabbix_ui> の例: https://example.com/zabbix/ui, http://another.example.com/zabbix, http://<any_public_

Zabbix の設定

Attention:
Web インターフェースで SAML 認証を使用する場合は、php-openssl をインストールする必要があります。

SAML 認証を使用するには、Zabbix を次のように設定する必要があります。

1. SP の秘密鍵と証明書を指定します。Web インターフェースで SAML を有効にするには、IdP 証明書を指定する必要があります（テキストを貼り付けるか、ファイルを選択します）。*Sign* または *Encrypt* オプションを選択する場合は、SP 証明書と SP 秘密鍵を指定する必要があります。

Web インターフェース設定ファイル（zabbix.conf.php）で $SSO['CERT_STORAGE'] = 'database'が設定されている場合、SAML の設定中に Web インターフェースで証明書のテキストを貼り付けるか、証明書ファイルをアップロードできます。ファイルシステム上にファイルを用意する必要はありません。

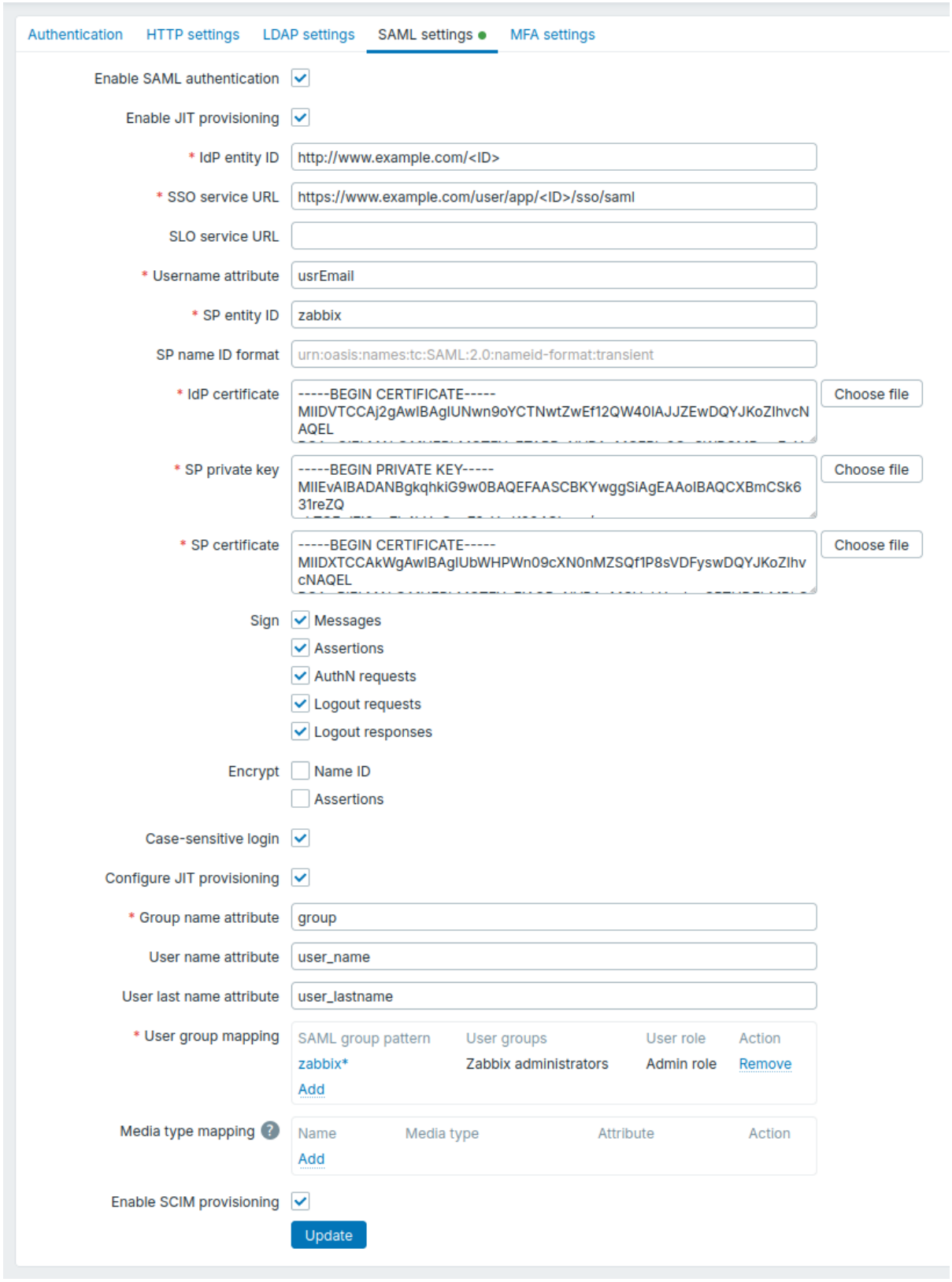
設定ファイルで $SSO['CERT_STORAGE'] = 'file'が設定されている場合、証明書をファイルシステム上（デフォルトでは ui/conf/certs、または設定ファイルで指定したパス）で使用できるようにする必要があります。この場合、Web インターフェースは証明書をデータベースに保存しません。$SSO['CERT_STORAGE'] が設定されていないか、コメントアウトされている場合は、ファイルストレージが使用されることに注意してください。

デフォルトでは、Zabbix は次の場所を検索します。

- ui/conf/certs/sp.key - SP 秘密鍵ファイル
- ui/conf/certs/sp.crt - SP 証明書ファイル
- ui/conf/certs/idp.crt - IDP 証明書ファイル

Attention:
証明書の内容は、テキストを貼り付けるかファイルをアップロードすることで、Web インターフェースからインポートできます。一部の環境では利便性のため暗号化されていないインポートも許可されていますが、暗号化された通信を使用することを強く推奨します（たとえば、Zabbix Web インターフェースで HTTPS/TLS を有効にします）。SAML 証明書/鍵をデータベースに保存する場合は、アプリケーションとデータベース間の通信を保護するため、データベース接続でTLS/SSL を有効にしてください。
 SAML 証明書または秘密鍵を含む可能性のあるデータベースバックアップを暗号化し、バックアップファイルおよび SAML の内容を保存するデータベーステーブルへのアクセスを制限してください。データベースユーザーの権限は、必要最小限に制限してください。

2. 最も重要な設定はすべて Zabbix Web インターフェースで設定できます。ただし、設定ファイルで追加設定を指定することもできます。



Zabbix Web インターフェースで使用できる設定パラメータ:

パラメータ	説明
SAML 認証を有効にする	チェックボックスを選択して、SAML 認証を有効にします。
JIT プロビジョニングを有効にする	チェックボックスを選択して、JIT ユーザープロビジョニングを有効にします。
IDP エンティティ *ID*	SAML アイデンティティプロバイダー内の一意のエンティティ識別子。
SSO サービス *URL*	ログイン時にユーザーがリダイレクトされる URL。
SLO サービス *URL*	ログアウト時にユーザーがリダイレクトされる URL。空の場合、SLO サービスは使用されません。

パラメータ	説明
ユーザー名属性	Zabbix へのログイン時にユーザー名として使用する SAML 属性。 サポートされる値の一覧は、アイデンティティプロバイダーによって決まります。 例: uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
SP エンティティ *ID*	一意のサービスプロバイダー識別子（一致しない場合、操作は拒否されます）。 URL または任意の文字列データを指定できます。
SP name ID 形式	応答で特定の name ID 形式を要求します。 例: urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified urn:oasis:names:tc:SAML:2.0:nameid-format:transient
IdP 証明書	SAML シングルサインオン（SSO）サービスの設定に使用するサービスプロバイダー（SP）証明書の内容。
SP 秘密鍵	SAML シングルサインオン（SSO）サービスの設定に使用するサービスプロバイダー（SP）秘密鍵の内容。アイデンティティプロバイダー（IdP）との安全な認証およびデータ交換を提供します。
SP 証明書	SAML シングルサインオン（SSO）サービスの設定に使用するサービスプロバイダー（SP）証明書の内容。
署名	対応するエンティティの SAML 署名を無効にするには、チェックボックスの選択を解除します: メッセージ アサーション *AuthN* リクエスト ログアウトリクエスト ログアウトレスポンス
暗号化	SAML 暗号化を有効にするエンティティを選択するには、チェックボックスを選択します: *Name ID* アサーション
大文字と小文字を区別するログイン	ユーザー名の大文字と小文字を区別するログインを無効にするには、チェックボックスの選択を解除します（デフォルトでは有効）。 大文字と小文字を区別するログインを無効にすると、たとえば Zabbix ユーザーが「Admin」または「ADMIN」であっても、「admin」としてログインできます。 大文字と小文字を区別するログインが無効で、類似したユーザー名（例: Admin と admin）を持つ Zabbix ユーザーが複数存在する場合、次のエラーメッセージが表示され、これらのユーザーのログインは常に拒否されることに注意してください: 「Authentication failed: supplied credentials are not unique.」
JIT プロビジョニングを設定する	このチェックボックスを選択して、JIT ユーザープロビジョニングに関連するオプションを表示します。
グループ名属性	JIT ユーザープロビジョニングのグループ名属性を指定します。
ユーザー名属性	JIT ユーザープロビジョニングのユーザー名属性を指定します。
ユーザー姓属性	JIT ユーザープロビジョニングのユーザー姓属性を指定します。
ユーザーグループマッピング	SAML ユーザーグループのパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングします。 プロビジョニングされたユーザーに Zabbix で付与するユーザーグループ/ロールを決定するために必要です。 追加をクリックしてマッピングを追加します。 *SAML* グループパターンフィールドではワイルドカードを使用できます。グループ名は既存のグループと一致する必要があります。 SAML ユーザーが複数の Zabbix ユーザーグループに一致する場合、そのユーザーはすべてのグループのメンバーになります。 ユーザーが複数の Zabbix ユーザーロールに一致する場合、それらの中で最も高い権限レベルが付与されます。
メディアタイプマッピング	通知の送信に使用するユーザーの SAML メディア属性（例: email）を Zabbix ユーザーメディアにマッピングします。

パラメータ	説明
SCIM プロビジョニングを有効にする	このチェックボックスを選択して、SCIM 2.0 プロビジョニングを有効にします。

次のアイデンティティプロバイダーを使用して、Zabbix へのサインインおよびユーザープロビジョニング用に SAML を設定する例を参照してください。

- Microsoft Azure AD
- Okta
- Onelogin

SCIM プロビジョニングに関する注意事項

SCIM プロビジョニングでは、ID プロバイダー側で Zabbix の Web インターフェースへのパスを指定し、`api_scim.php` を追加します。例:

`https://<path-to-zabbix-ui>/api_scim.php`

Zabbix で使用するユーザー属性(ユーザー名、ユーザーの名、ユーザーの姓、メディア属性)はカスタム属性として追加する必要があります。また、必要に応じて、外部名前空間はユーザースキーマと同じにします: `urn:ietf:params:scim:schemas:core:2.0:User`。

高度な設定

追加の SAML パラメータは、Zabbix のWeb インターフェース設定ファイル（`zabbix.conf.php`）で設定できます。

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

設定例

例 1: TLS/SSL を使用するロードバランサー経由で Zabbix インスタンスに接続する

ロードバランサーを使用して Zabbix インスタンスに接続し、ロードバランサーでは TLS/SSL を使用する一方で Zabbix では使用しない場合は、`baseurl`、`strict`、`use_proxy_headers` パラメータを次のように指定します。

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

例 2: カスタム SAML 署名およびダイジェストアルゴリズムを設定する

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

例 3: ADFS を使用して Kerberos 認証による SSO を設定する

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

この場合、SAML 設定の *SP name ID* フィールドに次の値を設定します。

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

MFA

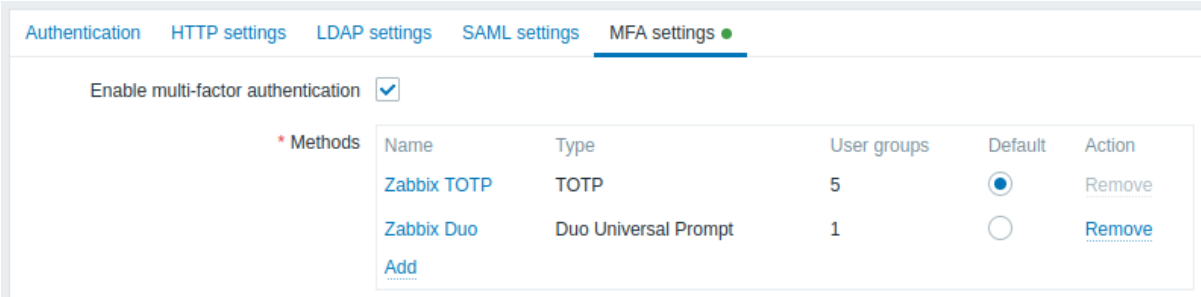
概要

Zabbix へのサインインには、多要素認証（MFA）を使用でき、ユーザー名とパスワードだけでなく、追加のセキュリティ層を提供します。

MFA を使用する場合、ユーザーは Zabbix に存在し、ログイン時に Zabbix の認証情報を提供し、さらに通常はユーザーの携帯電話の認証アプリによって生成されたコードなど、他の手段で本人確認を行う必要があります。

複数の MFA 方式が利用可能であり、ユーザーは自分のセキュリティ要件や好みに最適なオプションを選択できます。これらの方式には、時刻ベースのワンタイムパスワード（TOTP）と Duo Universal Prompt があります。

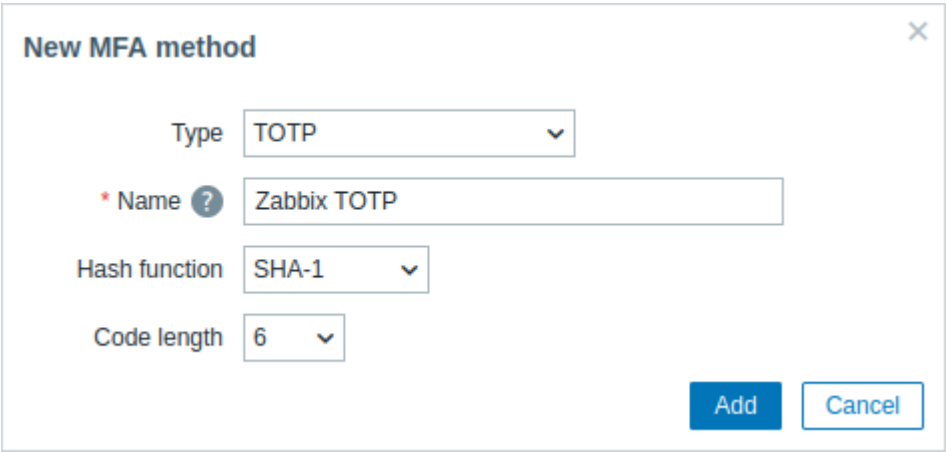
設定



設定パラメータ:

パラメータ	説明
多要素認証を有効にする	チェックボックスをオンにすると多要素認証が有効になります。
メソッド	追加をクリックして MFA メソッドを設定します（下記のメソッド設定を参照）。

メソッドの設定



メソッド設定パラメータ:

パラメータ	説明
タイプ	MFA メソッドのタイプを選択: **TOTP** - 認証アプリを使用して時限ワンタイムパスワードを生成します。 **Duo Universal Prompt** - Duo認証サービスを使用して多要素認証を提供します。
名前	認証アプリで全ての MFA ユーザーにアカウント名として表示される名前を入力します（例：”Zabbix”）。
ハッシュ関数	TOTP コードを生成するためのハッシュ関数（SHA-1、SHA-256、または SHA-512）を選択します。 このパラメータは、MFA メソッドタイプが「TOTP」に設定されている場合に利用できます。 SHA-256 または SHA-512 を選択すると、多くのアプリケーションがこれらの関数をサポートしていないため、互換性が大幅に制限される可能性があることに注意してください。
コード長	認証コードの長さ（6 または 8）を選択します。 このパラメータは、MFA メソッドタイプが「TOTP」に設定されている場合に利用できます。

パラメータ	説明
API ホスト名	Duo 認証サービスから提供された API ホスト名を入力します。 このパラメータは、MFA メソッドタイプが「Duo Universal Prompt」に設定されている場合に利用できます。
クライアント *ID*	Duo 認証サービスから提供されたクライアント ID を入力します。 このパラメータは、MFA メソッドタイプが「Duo Universal Prompt」に設定されている場合に利用できます。
クライアントシークレット	Duo 認証サービスから提供されたクライアントシークレットを入力します。 このパラメータは、MFA メソッドタイプが「Duo Universal Prompt」に設定されている場合に利用できます。

設定例

このセクションでは、Time-Based One-Time Password (TOTP) およびDuo Universal Prompt を使用した MFA の設定例を示します。

TOTP

TOTP では、ユーザーは認証アプリ（たとえば、Google Authenticator アプリ）を使用して本人確認を行う必要があります。

1. Zabbix の *Users* > *Authentication* にある MFA 設定に移動し、多要素認証を有効にします。

2. 次の設定で新しい MFA method を追加します。

- Type: TOTP
- Name: Zabbix TOTP
- Hash function: SHA-1
- Code length: 6

3. *Add* ボタンをクリックし、次に *Update* ボタンをクリックします。

4. *Users* > *User groups* に移動し、次のconfiguration で新しいユーザーグループを作成します。

- Group name: TOTP group
- Users: Admin
- Multi-factor authentication: Default（または、デフォルトに設定されていない場合は”Zabbix TOTP”）

5. Zabbix からログアウトし、資格情報を使用して再度ログインします。
ログインに成功すると、MFA の登録を求められ、QR コードとシークレットキーが表示されます。



6. QR コードをスキャンするか、シークレットキーを Google Authenticator アプリに入力します。
アプリが検証コードを生成するので、それを入力してログイン処理を完了します。

7. 以降のログインでは、Google Authenticator アプリから検証コードを取得し、ログイン時に入力します。

Duo Universal Prompt

Duo Universal Prompt では、ユーザーは Duo Mobile 認証アプリを使用して本人確認を行う必要があります。

Attention:
Duo Universal Prompt の MFA メソッドを使用するには、php-curl 拡張モジュールのインストール、HTTPS 経由での Zabbix へのアクセス、および Duo サーバーへのアウトバウンド接続の許可が必要です。さらに、Web サーバーで Content Security Policy (CSP) を有効にしている場合 は、仮想ホストの設定ファイル内の CSP ディレクティブに”duo.com” を追加してください。

1. Duo Signup で無料の Duo 管理者アカウントに登録します。

2. Duo Admin Panel を開き、*Applications* > *Protect an Application* に移動し、*Web SDK* アプリケーションを検索して、*Protect* をクリックします。

3. Zabbix で MFA メソッドを設定するために必要な認証情報（Client ID、Client secret、API hostname）を控えます。

4. Zabbix の *Users* > *Authentication* で MFA 設定に移動し、多要素認証を有効にします。

5. 次の設定で新しい MFA method を追加します。

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: (Duo の API hostname を使用)
- Client ID: (Duo の Client ID を使用)
- Client secret: (Duo の Client secret を使用)

6. *Add* ボタンをクリックし、次に *Update* ボタンをクリックします。

7. *Users* > *User groups* に移動し、次のconfiguration で新しいユーザーグループを作成します。

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default（または、デフォルトに設定されていない場合は”Zabbix Duo”）

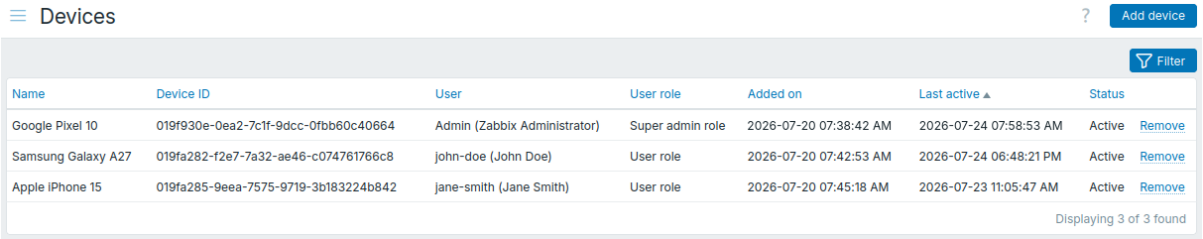
8. Zabbix からログアウトし、資格情報を使用して再度ログインします。ログインに成功すると、MFA の登録を求められ、Duo にリダイレクトされます。Duo のセットアップを完了し、スマートフォンの Duo アプリでユーザー確認を行ってログインします。

9. 以降のログインでは、Duo アプリで提供される適切な MFA メソッド（確認コードの取得、プッシュ通知への応答、ハードキーの使用など）を使用し、ログイン時に必要な情報を入力します。

デバイス

このセクションでは、Zabbix モバイルアプリで使用するモバイルデバイスをスーパーユーザーが管理できます。

このセクションは、`EnableMobileDevices`が有効な場合に表示されます。



デバイスを追加するには、次の手順に従います。

1. デバイスを追加をクリックします。
2. ユーザーを選択し、追加をクリックします。ユーザーにはデバイスへのアクセスが有効になっている必要があります。
3. QR コードが表示されます。ユーザーは Zabbix モバイルアプリでこのコードをスキャンする必要があります。

Note:
ユーザーのデバイスを追加すると、Zabbix はそのユーザーのユーザーメディアにプッシュ通知メディアタイプも追加します。デバイスを削除すると、そのユーザーメディアも削除されます。複数のプッシュ通知メディアタイプが設定されている場合、Zabbix は有効になっているもののうち、アルファベット順で最初のメディアタイプを追加します。

追加されたデバイスは一覧に表示され、フィルターを使用できます。

デバイスを削除することもできます。デバイスを削除すると、そのアクセス認証情報は無効化され、Zabbix から削除されます。その後、そのデバイスは Zabbix にアクセスできなくなります。

一覧には次の詳細が表示されます。

Column	Description
名前	デバイス名。
デバイス *ID*	デバイス ID。
ユーザー	デバイスが追加されたユーザー。このユーザーが削除されている場合は、アクセスできないユーザーと表示されます。
ユーザーの役割	ユーザーの役割。
追加日時	デバイスが追加された日時。
最終アクティブ日時	デバイスが最後にアクティブだった日時。
ステータス	デバイスのステータス: 新規 - QR コードをスキャンせずにデバイスが追加されたため、再度追加する必要があります。 新規（トークンの有効期限切れ） - QR コードをスキャンせずにデバイスが追加され、QR コード内のトークンの有効期限が切れています。デバイスを再度追加する必要があります。 アクティブ - デバイスが追加され、QR コードがスキャンされており、Zabbix モバイルアプリを使用できます。 未割り当て - デバイスが追加されましたが、そのユーザーが削除されているため、Zabbix モバイルアプリを使用できません。

十分な権限を持つユーザーは、ユーザー設定 > デバイスで自分のデバイスを管理することもできます。

管理

概要

管理メニューは、Zabbix の管理機能のためのものです。このメニューは、SuperAdminユーザータイプのユーザーのみが利用できます。

データソース

概要

管理 > データソースセクションでは、テレメトリデータソース、つまりテレメトリデータ（トレース、メトリクス、ログ）を保持するデータベースを指定できます。テレメトリデータは、アプリケーションパフォーマンスモニタリング（APM）に使用されます。APM を使用すると、アプリケーションやサービスを通過するリクエストの流れを確認し、パフォーマンスを測定して、処理の遅延や失敗の原因を特定できます。

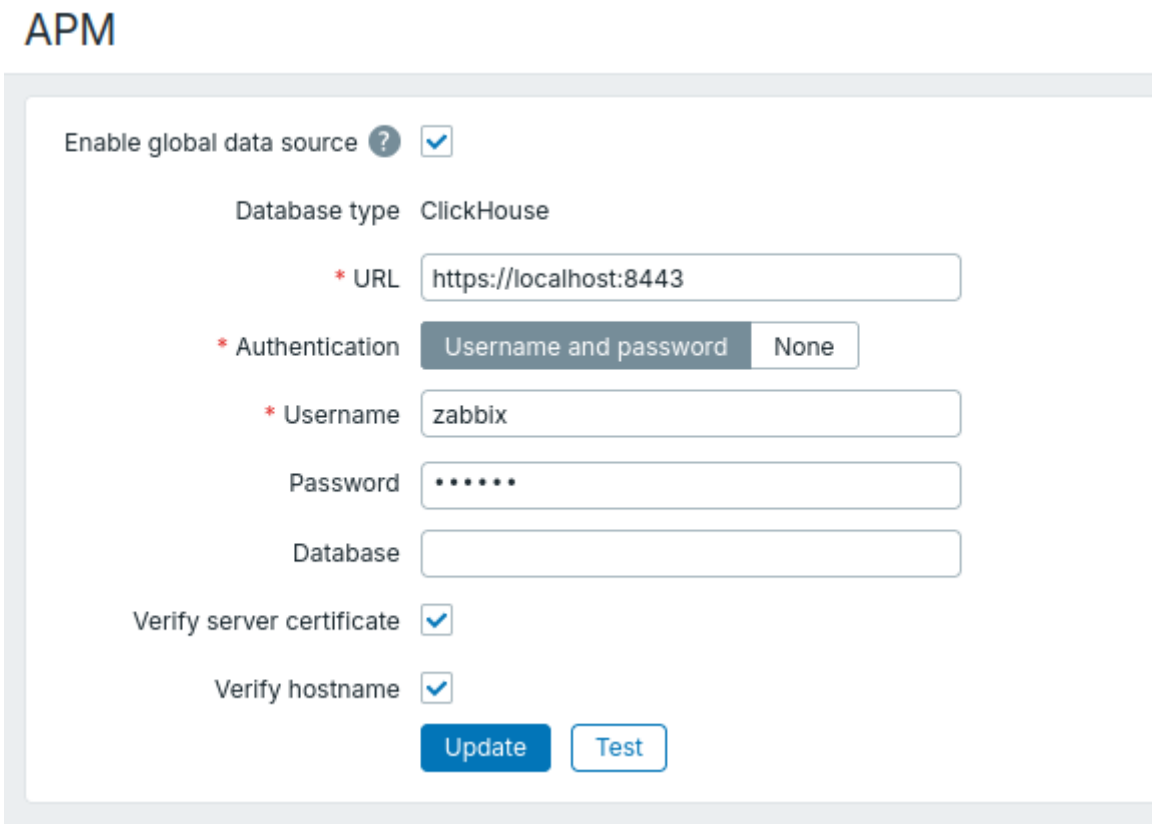
テレメトリデータソースは、OpenTelemetryをサポートしている必要があります。

このセクションで指定するテレメトリデータソースは、グローバルデータソースと呼ばれます。これは、Zabbix サーバー、プロキシ、Web インターフェースで共有されるためです。ただし、サーバー、プロキシ、またはWeb インターフェースの設定で別のテレメトリデータソースを指定することで、これらのコンポーネントごとに上書きできます。

テレメトリクエリアイテムは、コンポーネントの設定で別のデータソースが指定されていない限り、このデータソースに対してクエリを実行します。

APM

グローバルデータソースを指定するには、グローバルデータソースを有効にするチェックボックスを選択し、以下に説明するパラメータを入力します。保存する前に、テストをクリックして、このデータソースへの接続を確認できます（php-curl拡張機能が必要です）。



パラメータ	説明
データベースタイプ	データベースタイプは *ClickHouse* に固定されており、変更できません。 関連項目：要件およびClickHouse のセットアップ。
URL	データベースへの接続に使用する HTTP または HTTPS URL を入力します。
認証	データベースへのログインに使用する認証方式として、ユーザー名とパスワードまたはなしを選択します。
ユーザー名	データベースへのログインに使用するユーザー名を指定します。 認証でユーザー名とパスワードを設定した場合に、このパラメータを使用できます。
パスワード	データベースへのログインに使用するパスワードを指定します。データベースユーザーにパスワードがない場合は空のままにします。 URL を変更すると、パスワードはリセットされます。 認証でユーザー名とパスワードを設定した場合に、このパラメータを使用できます。
データベース	テレメトリデータを格納するデータベースの名前を入力します。 空のままにした場合、Zabbix はユーザーに対して ClickHouse で設定されているデフォルトデータベースを使用します。
サーバー証明書を検証	サーバー証明書を検証するには、チェックボックスを選択します。 選択した場合、関連する TLS オプションをサーバー、プロキシ、またはWeb インターフェースの設定で指定する必要があります。 *URL* に HTTPS URL が含まれている場合に、このパラメータを使用できます。
ホスト名を検証	*URL* 内のホスト名がサーバー証明書のホスト名と一致することを検証するには、チェックボックスを選択します。 選択した場合、関連する TLS オプションをサーバー、プロキシ、またはWeb インターフェースの設定で指定する必要があります。 *URL* に HTTPS URL が含まれている場合に、このパラメータを使用できます。

全般

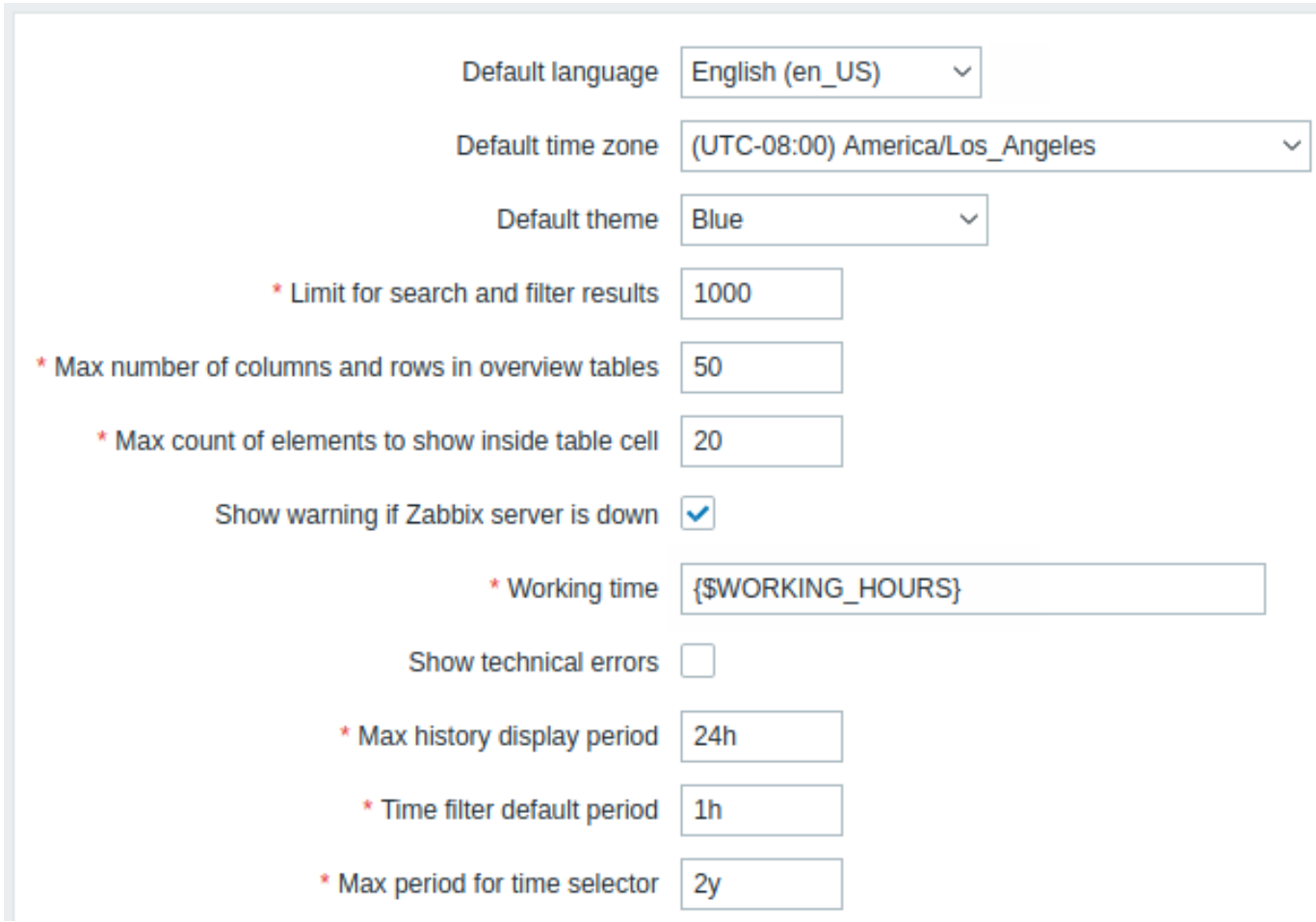
概要

Administration > *General* セクションには、Web インターフェース関連のデフォルト設定や Zabbix のカスタマイズを行うための複数のサブセクションがあります。

利用可能なサブセクションの一覧は、*Administration* メニューセクションで *General* をクリックすると表示されます。また、左上隅のタイトルのドロップダウンを使用して、サブセクション間を切り替えることもできます。

GUI

このセクションでは、いくつかの Web インターフェース関連の既定値をカスタマイズできます。



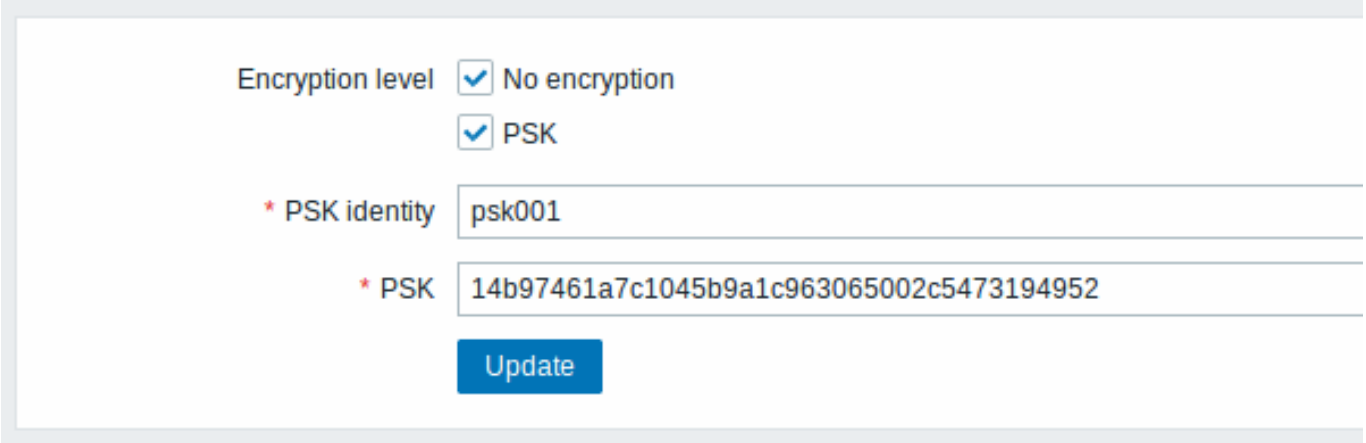
設定パラメーター:

Parameter	Description
Default language	プロファイルで言語を指定していないユーザーおよびゲストユーザーに対する既定の言語。 詳細は追加の Web インターフェース言語のインストール を参照してください。
Default time zone	プロファイルでタイムゾーンを指定していないユーザーおよびゲストユーザーに対する既定のタイムゾーン。
Default theme	プロファイルでテーマを指定していないユーザーおよびゲストユーザーに対する既定のテーマ。
Limit for search and filter results	Web インターフェースの一覧に表示される要素数（行数）の上限です。たとえば、*Data collection* > *Hosts* などで使用されます。 *Note*: たとえば'50' に設定すると、影響を受けるすべての frontend 一覧では最初の 50 要素のみが表示されます。ある一覧に 50 を超える要素が含まれる場合、そのことは *"Displaying 1 to 50 of **50+** found"* の「+」記号で示されます。また、フィルタリングを使用しても 50 件を超える一致がある場合は、最初の 50 件のみが表示されます。 このパラメーターの値を大きくすると、Web サーバー側でパフォーマンスの低下やメモリ消費の増加につながる可能性があることに注意してください。
Max number of columns and rows in overview tables	*Trigger overview* ダッシュボードウィジェットに表示する列数と行数の最大値です。列と行の両方に同じ上限が適用されます。表示数を超える行や列が存在する場合、システムは表の下部に次の警告を表示します: "Not all results are displayed. Please provide more specific search criteria."
Max count of elements to show inside table cell	1 つの表セル内に表示される項目について、ここで設定した数を超えて表示しません。
Show warning if Zabbix server is down	このパラメーターを有効にすると、Zabbix サーバーに接続できない場合（停止している可能性があります）にブラウザーウィンドウへ警告メッセージを表示します。メッセージは、ユーザーがページを下へスクロールしても表示されたままです。マウスオーバーすると一時的に非表示になり、その下の内容を確認できます。
Working time	このシステム全体のパラメーターは勤務時間を定義します。グラフでは、勤務時間は白い背景で、非勤務時間は灰色で表示されます。 時間形式の説明はTime period specification ページを参照してください。 ユーザーマクロ をサポートしています。

Parameter	Description
Show technical errors	チェックすると、登録済みのすべてのユーザーが技術的エラー（PHP/SQL）を表示できるようになります。チェックしない場合、この情報はZabbix Super Admins と、debug mode が有効なユーザーグループに属するユーザーのみが利用できます。
Max history display period	*Monitoring* > *Latest data*、*Monitoring* > *Hosts* のホストweb scenario 情報、および *Top items* ダッシュボードウィジェットで履歴データを表示する最大期間です。 許可範囲: 24 時間（既定）- 1 週間。Time suffixes である 1w（1 週間）、36h（36 時間）などをサポートしています。
Time filter default period	グラフおよびダッシュボードで既定として使用する期間です。許可範囲: 1 分 - 10 年 (既定: 1 時間)。 Time suffixes である 10m（10 分）、5w（5 週間）などをサポートしています。 Note: ユーザーがグラフ表示中に期間を変更すると、その期間はユーザー設定として保存され、グローバル既定値または以前のユーザー選択が置き換えられます。
Max period for time selector	グラフおよびダッシュボードで選択可能な最大期間です。ユーザーはそれより古いデータを表示できません。許可範囲: 1 年 - 10 年（既定: 2 年）。 Time suffixes である 1y（1 年）、365w（365 週間）などをサポートしています。

自動登録

このセクションでは、アクティブエージェントの自動登録に対する暗号化レベルを設定できます。



アスタリスクが付いたパラメータは必須です。

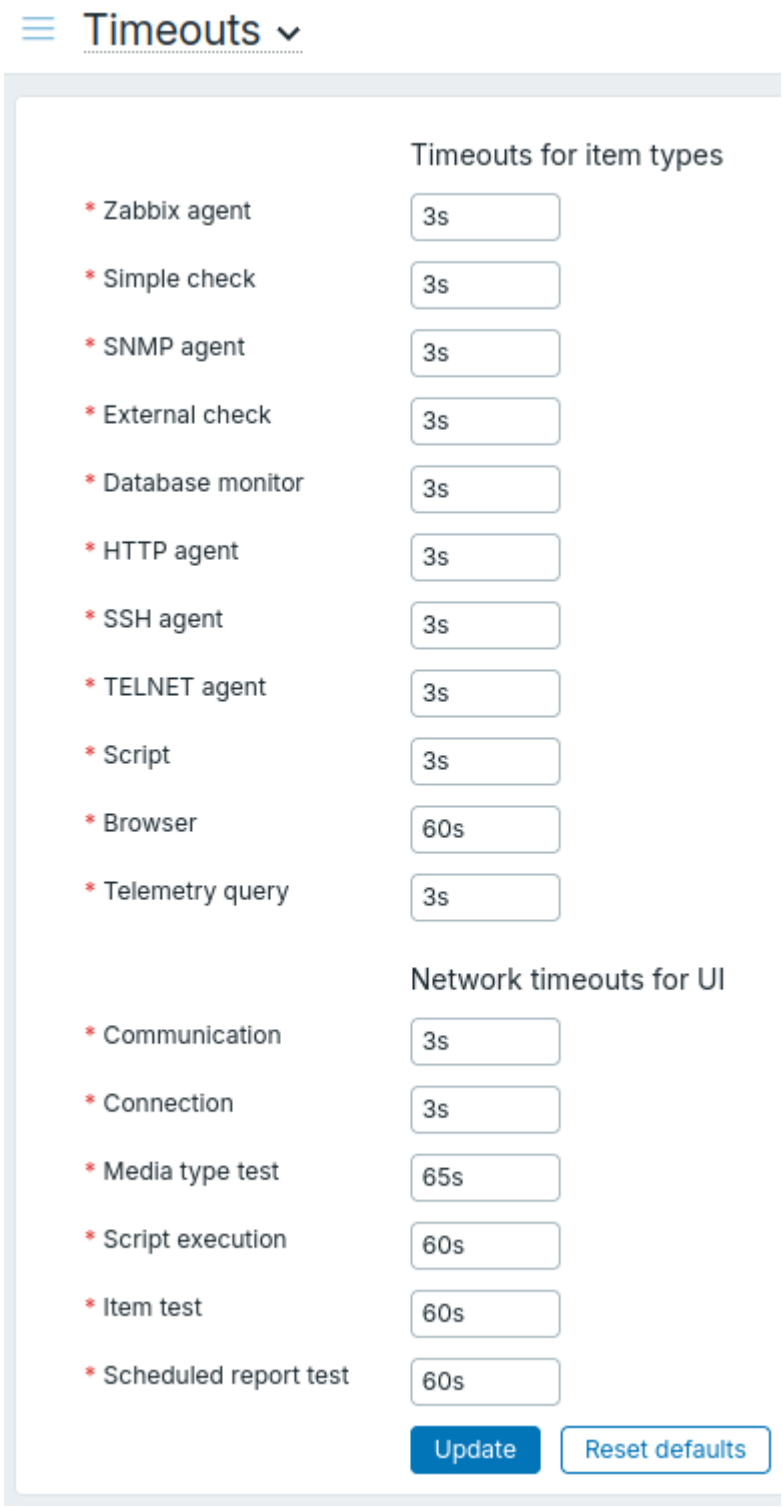
設定パラメータ:

Parameter	Description
暗号化レベル	暗号化レベルとして 1 つまたは両方のオプションを選択します: 暗号化なし - 暗号化されていない接続を許可します **PSK** - 事前共有鍵を使用した TLS 暗号化接続を許可します
PSK identity	事前共有鍵の識別文字列を入力します。 このフィールドは、暗号化レベルで'PSK' が選択されている場合にのみ使用できます。 PSK identity に機密情報を含めないでください。受信側にどの PSK を使用するかを通知するために、ネットワーク上へ暗号化されずに送信されます。
PSK	事前共有鍵を入力します（16 進文字の偶数桁）。 最大長: Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 16 進桁（256 バイトの PSK）、Zabbix が mbed TLS（PolarSSL）ライブラリを使用する場合は 64 16 進桁（32 バイトの PSK）。 例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 このフィールドは、暗号化レベルで'PSK' が選択されている場合にのみ使用できます。

参照: 安全な自動登録

タイムアウト

このセクションでは、アイテムタイプのグローバルタイムアウトとネットワークタイムアウトを設定できます。
このフォームのすべてのフィールドは必須です。

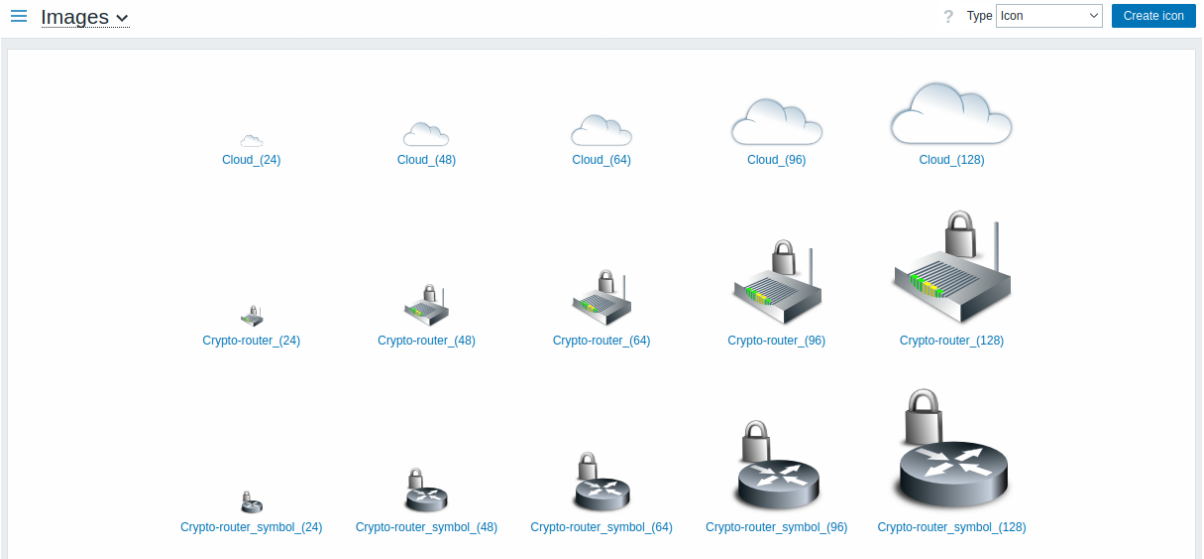


Parameter	Description
アイテムタイプのタイムアウト	Zabbix がチェックを中止するまでに待機する秒数です（タイプに基づく）。 詳細と制限については、item timeout を参照してください。 許容範囲: 1 - 600s（デフォルト: 3s; browser アイテムのデフォルト: 60s）。 Time suffixes（例: 30s、1m）およびuser macros がサポートされています。 プロキシレベルのtimeouts が設定されている場合は、これらのグローバル設定よりも優先されることに注意してください。さらに、特定のitems のタイムアウトが設定されている場合は、グローバル設定とプロキシ設定の両方よりも優先されます。
UI のネットワークタイムアウト	

Parameter		Description
	通信	アイドル状態のソケットを閉じるまでに待機する秒数です（Zabbix サーバーへの接続が以前に確立されているが、Web インターフェースがこの時間内にデータの読み取り/送信操作を完了できない場合、接続は切断されます）。許容範囲: 1 - 300s（デフォルト: 3s）。
	接続	Zabbix サーバーへの接続試行を停止するまでに待機する秒数です。許容範囲: 1 - 30s（デフォルト: 3s）。
	メディアタイプのテスト	メディアタイプのテスト時に応答を待機する秒数です。許容範囲: 1 - 300s（デフォルト: 65s）。
	スクリプト実行	スクリプト実行時に応答を待機する秒数です。許容範囲: 1 - 300s（デフォルト: 60s）。 このタイムアウトはスクリプト実行チェーン全体に適用され、長さはさまざまです。たとえば、スクリプトがエージェント上で実行される場合、サーバー（場合によってはプロキシも）からエージェントへの往復処理が発生します。 エージェント上で実行されるスクリプトには、server またはproxy のタイムアウトが適用されます。 アクティブエージェントでのみ実行されるスクリプトの場合、既定のサーバー/プロキシのタイムアウトを引き上げる必要がある可能性が高いです。サーバー/プロキシのタイムアウトは、アクティブチェックの更新頻度よりも大きくなければなりません。そうでない場合、アクティブエージェントがスクリプトを受信する前にサーバー/プロキシのタイムアウトに達する可能性があります。 関連項目: Script timeout。
	アイテムテスト	アイテムのテスト時に返されるデータを待機する秒数です。許容範囲: 1 - 600s（デフォルト: 60s）。
	スケジュールレポートのテスト	スケジュールレポートのテスト時に返されるデータを待機する秒数です。許容範囲: 1 - 300s（デフォルト: 60s）。

画像

「画像」セクションには、Zabbix で使用可能なすべての画像が表示されます。画像はデータベースに保存されます。

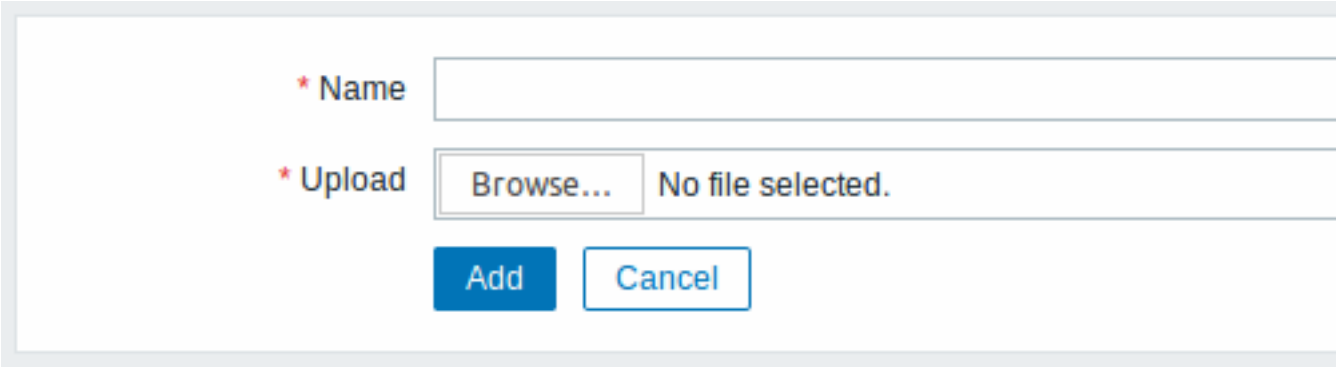


タイプドロップダウンでは、アイコン画像と背景画像を切り替えることができます。

- アイコンは、ネットワークマップの要素の表示に使用されます。
- 背景は、ネットワークマップの背景画像として使用されます。

画像の追加

右上にあるアイコンの作成または背景の作成ボタンをクリックすると、独自の画像を追加できます。



画像の属性:

パラメーター	説明
名前	画像の一意の名前。
アップロード	ローカルシステムから Zabbix にアップロードするファイル（PNG、JPEG、GIF、または WebP）を選択します。 注：アップロード時に PNG に変換される他の形式もアップロードできる場合があります。画像の処理には GD ライブラリが使用されるため、サポートされる形式は使用するライブラリのバージョンによって異なります（Zabbix では 2.0.28 以降が必要です）。

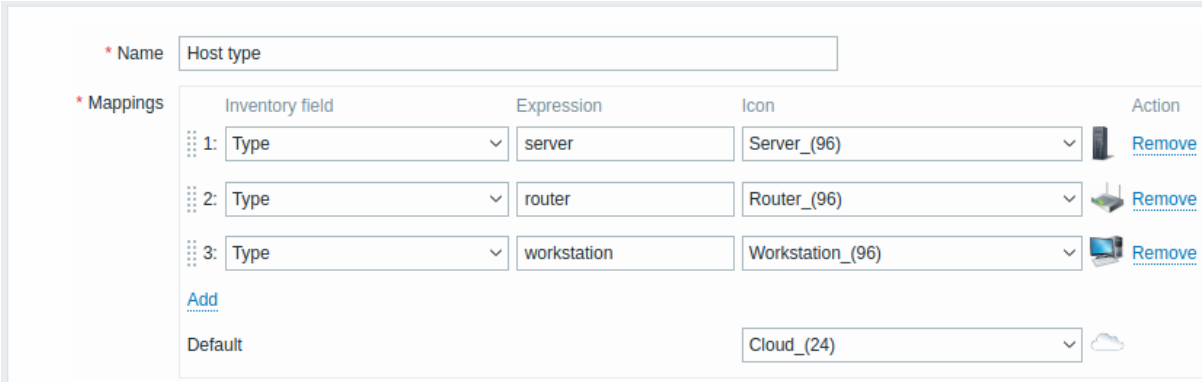
Note:
アップロードファイルの最大サイズは、1024x1024 バイト（1 MB）である ZBX_MAX_IMAGE_SIZE の値によって制限されます。
画像サイズが 1 MB に近く、MySQL の設定パラメーター max_allowed_packet がデフォルトの 1 MB に設定されている場合、画像のアップロードに失敗することがあります。この場合は、max_allowed_packetパラメーターの値を増やしてください。

アイコンマッピング

このセクションでは、特定のホストを特定のアイコンにマッピングできます。ホストのインベントリフィールド情報を使用してマッピングを作成します。

作成したマッピングは、ネットワークマップの設定で使用でき、条件に一致するホストに適切なアイコンを自動的に割り当てられます。

新しいアイコンマップを作成するには、右上隅の アイコンマップの作成をクリックします。



設定パラメータ:

Parameter	Description
Name	アイコンマップの一意の名前。
Mappings	マッピングの一覧。マッピングの順序によって、どれが優先されるかが決まります。ドラッグアンドドロップで、マッピングを一覧内で上下に移動できます。
Inventory field	一致を検索するために参照するホストのインベントリフィールド。
Expression	一致条件を表す正規表現。
Icon	式に一致する場合に使用するアイコン。
Default	使用するデフォルトのアイコン。

正規表現

このセクションでは、Web インターフェースの複数の場所で使用できるカスタム正規表現を作成できます。詳細は正規表現セクションを参照してください。

トリガー表示オプション

このセクションでは、Web インターフェースでのトリガー状態の表示方法と、トリガーの深刻度の名前および色をカスタマイズできます。

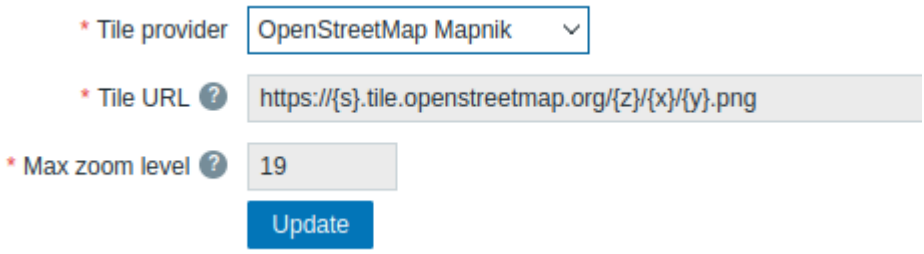


Parameter	Description
Use custom event status colors	このパラメータを有効にすると、確認済み/未確認の問題に対する色のカスタマイズが有効になります。
Unacknowledged PROBLEM events, *Acknowledged PROBLEM events*, *Unacknowledged RESOLVED events*, *Acknowledged RESOLVED events*	新しいカラーコードを入力するか、色をクリックして、表示されたパレットから新しい色を選択します。 *blinking* チェックボックスをオンにすると、状態変化後にトリガーがしばらく点滅し、より目立つようになります。
Display OK triggers for	OK トリガーを表示する期間です。許容範囲は 0 - 24 時間です。時間サフィックス（例: 5m、2h、1d）を使用できます。
On status change triggers blink for	トリガーの点滅時間です。許容範囲は 0 - 24 時間です。時間サフィックス（例: 5m、2h、1d）を使用できます。
Not classified, *Information*, *Warning*, *Average*, *High*, *Disaster*	システムの既定値の代わりに表示する、カスタムの深刻度名および/または色です。 新しいカラーコードを入力するか、色をクリックして表示されたパレットから新しい色を選択します。 ここで入力したカスタムの深刻度名は、すべてのロケールで使用される点に注意してください。特定のユーザー向けに他の言語へ翻訳する必要がある場合は、トリガーの深刻度のカスタマイズページを参照してください。

地理マップ

このセクションでは、地理マップのタイルサービスプロバイダーを選択し、Geomap ダッシュボードウィジェット のサービスプロバイダー設定を構成できます。
地理マップを使用した可視化を提供するために、Zabbix はオープンソースの JavaScript インタラクティブマップライブラリ Leaflet を使用します。

Zabbix は、あらかじめ定義されたタイルプロバイダーを含む、サードパーティのタイルプロバイダーが提供する画像の品質を管理できないことにご注意ください。



Parameter	Description
Tile provider	利用可能なタイルサービスプロバイダーのいずれかを選択するか、*Other* を選択して別のタイルプロバイダーまたはセルフホストのタイルを追加します（カスタムタイルサービスプロバイダーの使用 を参照）。
Tile URL	地理マップ上でタイルレイヤーを読み込み、表示するための URL テンプレート（最大 2048 文字）です。 このフィールドは、*Tile provider* が *Other* に設定されている場合にのみ編集できます。 次のプレースホルダーがサポートされています: *{s}* は利用可能なサブドメインのいずれかを表します; *{z}* は URL 内のズームレベルパラメーターを表します; *{x}* と *{y}* はタイル座標を表します; *{r}* は URL に"@2x" を追加して retina タイルを読み込むために使用できます。 例: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attribution text	地図上の小さなテキストボックスに表示される、タイルプロバイダーの帰属表示テキストです。このフィールドは、*Tile provider* が *Other* に設定されている場合にのみ表示されます。
Max zoom level	地図の最大ズームレベルです。このフィールドは、*Tile provider* が *Other* に設定されている場合にのみ編集できます。

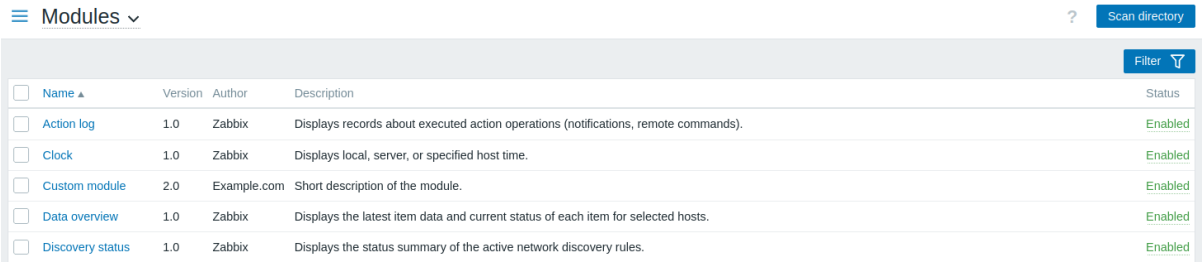
カスタムタイルサービスプロバイダーの使用

Geomap ウィジェットは、カスタムのセルフホスト型またはサードパーティのタイルプロバイダーサービスからラスタータイル画像を読み込むことができます。カスタムのサードパーティタイルプロバイダーサービス、またはセルフホスト型のタイルフォルダやサーバーを使用するには、*Tile provider* フィールドで *Other* を選択し、適切なプレースホルダーを使用して *Tile URL* フィールドにカスタム URL を指定します。

モジュール

このセクションでは、カスタムモジュールおよび組み込みのWeb インターフェースモジュールを管理できます。

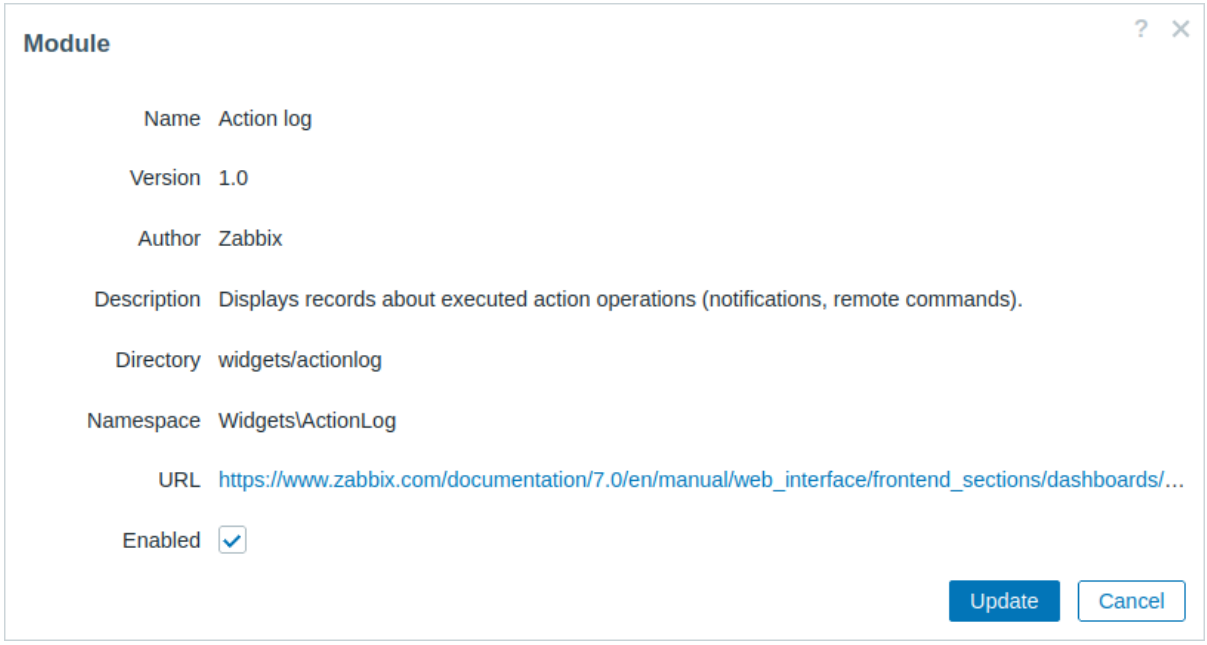
Note:
Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['modules_config_enabled']` 機能フラグが `false` に設定されている場合、このセクションへのアクセスは無効になります。



ディレクトリをスキャンをクリックすると、カスタムモジュールを登録または登録解除できます。登録されたモジュールは一覧に表示され、登録解除されたモジュールは一覧から削除されます。

一覧でモジュールのステータスをクリックすると、モジュールを有効または無効にできます。一覧でモジュールを選択してから、一覧の下にある有効/無効ボタンをクリックすることで、複数のモジュールを一括して有効または無効にすることもできます。

一覧でモジュール名をクリックすると、ポップアップウィンドウでモジュールの詳細を表示できます。

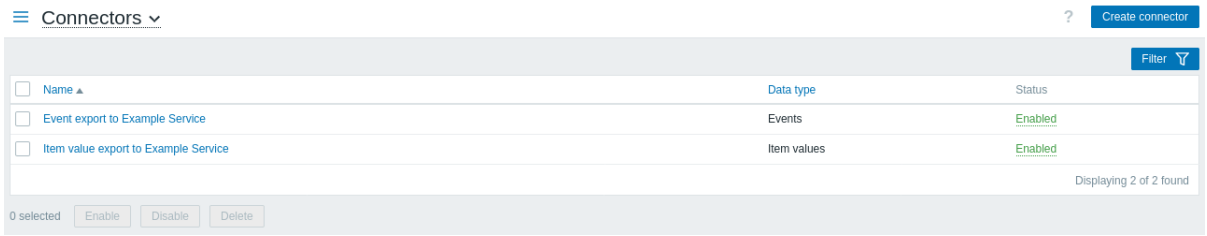


モジュールの詳細ポップアップウィンドウでも、モジュールのステータスを更新できます。これを行うには、有効チェックボックスを選択または選択解除してから、更新をクリックします。

モジュールは、名前またはステータス（有効/無効）でフィルターできます。

コネクター

このセクションでは、HTTP 経由で Zabbix データを外部システムにストリーミングするためのコネクターを設定できます。



コネクターの作成をクリックして、新しいコネクターを設定します。

コネクターは、名前またはステータス（有効/無効）でフィルタリングできます。一覧でコネクターのステータスをクリックすると、コネクターを有効または無効にできます。また、一覧でコネクターを選択してから一覧の下にある有効化/無効化ボタンをクリックすると、複数のコネクターを一括で有効または無効にできます。

その他

このセクションでは、その他の各種 Web インターフェースパラメータを設定できます。



パラメータ	説明
Web インターフェース *URL*	Zabbix Web インターフェースの URL（最大 2048 文字）。 このパラメータは、Zabbix web service が Web インターフェースと通信するために使用され、定期レポートを有効にするには指定が必要です。
ディスカバリされたホスト用のグループ	ネットワークディスカバリ およびエージェント自動登録 によってディスカバリされたホストは、ここで選択したホストグループに自動的に配置されます。
デフォルトのホストインベントリモード	ホストインベントリのデフォルトのモードです。新しいホストまたはホストプロトタイプがサーバーまたは Web インターフェースによって作成される際、ホストディスカバリ/自動登録中に ホストインベントリモードの設定操作で上書きされない限り、この設定に従います。

パラメータ	説明
データベース停止メッセージ用のユーザーグループ	データベース停止アラームメッセージの送信先となるユーザーグループです。 Zabbix サーバーはバックエンドデータベースの可用性に依存しています。データベースなしでは動作できません。データベースが停止した場合、選択したユーザーに対して Zabbix から通知できます。通知は、ここで設定したユーザーグループに対して、有効なユーザーメディアエントリを使用して送信されます。通知は次のメディアタイプを使用して送信されます（有効な場合）: メール、SMS、カスタムアラートスクリプト。webhook メディアエントリが設定され有効になっていても、通知はトリガーされません。 Zabbix サーバーは停止せず、データベースが復旧するまで待機して処理を継続します。 通知の内容は次のとおりです。 `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> は空の値として定義されている場合はメッセージに追加されず、<DB Port> はデフォルト値（"0"）の場合は追加されません。アラートマネージャー（Zabbix サーバーの特別なプロセス）は、10 秒ごとにデータベースへの新しい接続の確立を試みます。データベースが引き続き停止している場合、アラートマネージャーはアラート送信を繰り返しますが、その頻度は 15 分に 1 回を超えません。
一致しない *SNMP* トラップをログに記録	対応する SNMP インターフェースが見つからない場合に、SNMP トラップをログに記録します。

認証

パラメータ	説明
ログイン試行回数	ログイン可能性がブロックされるまでの、失敗したログイン試行回数。
ログインブロック間隔	ログイン試行回数の上限を超えた場合に、ログインが禁止される期間。許容範囲: 30～3600 秒。時間のサフィックス（例: 90s、5m、1h）をサポートしています。

シークレットの保存

Parameter	Description
Vault provider	ユーザーマクロ の値を保存するためのシークレット管理ソフトウェアを選択します - *HashiCorp Vault*（デフォルト）または *CyberArk Vault*。
Resolve secret vault macros by	シークレット vault マクロの解決方法: **Zabbix server** - シークレットは Zabbix server によって取得され、必要に応じてプロキシに転送されます（デフォルト）。 **Zabbix server and proxy** - シークレットは Zabbix server とプロキシの両方によって取得され、それぞれが独立してマクロを解決できるようになります。

関連項目: シークレットの保存。

セキュリティ

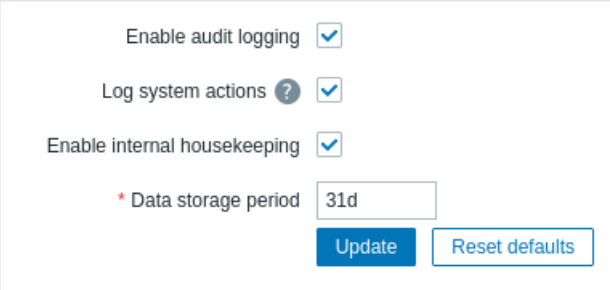
パラメータ	説明
URI スキームの検証	このチェックボックスを外すと、URI スキームの検証が無効になります (デフォルトは有効)。 チェックを入れると、許可する URI スキームのカンマ区切りリストを指定できます (デフォルト: http,https,ftp,file,mailto,tel,ssh)。URI が使用されるフロントエンドのすべてのフィールド (例: マップ要素の URL) に適用されます。
X-Frame-Options HTTP ヘッダーの使用	このチェックボックスを外すと、HTTP X-Frame-options ヘッダーが無効になります (非推奨)。 チェックを入れると、HTTP X-Frame-options ヘッダーの値を指定できます。サポートされる値: **SAMEORIGIN**(デフォルト) または**'self'**(シングルクォート必須) - ページ自身と同じオリジンのフレームでのみページを表示できます; **DENY** または**'none'**(シングルクォート必須) - ページをフレームで表示できません。どのサイトが表示しようとしても無効です; スペース区切りのホスト名の文字列; リストに**'self'**(シングルクォート必須) を追加すると、ページ自身と同じオリジンのフレームでページを表示できます。 **'self'** や**'none'** をシングルクォートなしで使用すると、ホスト名として扱われることに注意してください。

パラメータ	説明
iframe サンドボックスの使用	このチェックボックスを外すと、取得した URL コンテンツをサンドボックスに入れることが無効になります (非推奨)。 チェックを入れると、iframe サンドボックスの例外を指定できます。指定されていない制限は引き続き適用されます。このフィールドが空の場合、すべてのサンドボックス属性の制限が適用されます。 詳細は、sandbox属性の説明を参照してください。

監査ログ

概要

このセクションでは、監査ログの設定を行うことができます。



以下のパラメータが利用可能です。

パラメータ	説明
監査ログを有効にする	監査ログを有効（デフォルト）または無効にします。
システムアクションのログ	サーバー（システムユーザー）によって実行される低レベルディスカバリ、ネットワークディスカバリ、自動登録アクティビティの監査ログを有効（デフォルト）または無効にします。
内部ハウスキーピングを有効にする	監査ログレコードの内部ハウスキーピングを有効（デフォルト）または無効にします。
データ保存期間	ハウスキーパーによって削除されるまで監査ログレコードを保持する日数。 ハウスキーピングが有効な場合は必須です。 デフォルト：31 日。

監査ログ圧縮のパラメータは、TimescaleDBを使用している場合に表示されるヒストリ、トレンドおよび監査ログ圧縮ブロック内の管理 > ハウスキーピングセクションで利用できます。

ハウスキーピング

概要

Administration > *Housekeeping* セクションでは、Zabbix の内部 housekeeping を設定できます。これは、古くなったデータやユーザーによって削除されたデータをデータベースから自動的に削除します。これにより、データベースが際限なく増大するのを防ぎ、不要なディスク領域の消費やデータベース性能の低下を回避できます。

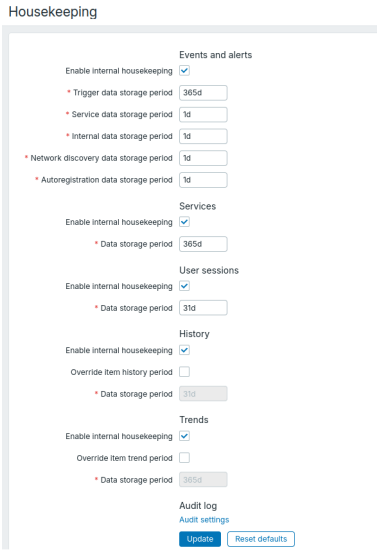
Housekeeping は、次の種類のデータに対して有効化および設定できます。

- トリガー、サービス、内部データ、ネットワークディスカバリ、および自動登録のイベントとアラート
- サービス
- ユーザーセッション
- アイテムの履歴とトレンド

Note:
監査ログの housekeeping は別途 設定します。

設定

次のパラメータを使用できます。



Parameter	Description
内部ハウスキーピングを有効にする	内部ハウスキーピングを有効または無効にします（デフォルトでは有効）。 有効にすると、ハウスキーピング手順 により、データ保存期間を超えたデータがデータベースから自動的に削除されます。
データ保存期間	ハウスキーパーによって削除されるまで、データをどのくらい保持するかを指定します。 内部ハウスキーピングが有効な場合は必須です。 範囲: 1 日（履歴は 1 時間）- 25 年、または”0”。時間サフィックス（例: 1d、1w）を使用できます。 障害とアラートでは、データ保存期間はトリガー、サービス、内部データ、ネットワークディスカバリ、オートレジストレーションごとに個別に設定されます。 ハウスキーパーは、障害に関連付けられていないイベントのみを削除します。たとえば、データ保存期間を超えた障害/復旧イベントでも、未解決の障害に関連付けられている場合は削除されません。 ハウスキーパーが期限切れのエンティティを削除する際は、まず障害を削除し、その後にイベントを削除します。 履歴と トレンドでは、内部ハウスキーピングが無効でも、データ保存期間によって 監視 > 最新データにデータが表示される期間も決まります。 アイテムまたはトリガーが削除されると、それに関連する障害とイベントも削除されることに注意してください。
アイテム履歴期間を上書きする	有効にすると、アイテム設定 で指定された履歴/トレンドの保存期間が データ保存期間の設定で上書きされます（保存しないオプションが有効なアイテムを除く）。 このオプションは、内部ハウスキーピングが無効で外部ハウスキーパーを使用している場合でも使用できます。
アイテムトレンド期間を上書きする	

ClickHouse を使用している場合、*ClickHouse* データ保存期間セクションが利用可能になります。ここには、各アイテム値タイプごとのデータ保存期間が表示されます。期間を変更するには、ClickHouse データ保存期間の設定 を参照してください。ハウスキーパーは ClickHouse からデータを削除しないことに注意してください。

Elasticsearch を使用している場合、*Elasticsearch* データ保存期間セクションが利用可能になります。ここには、Elasticsearch に保存されるアイテム値タイプが表示され、データ保存期間は *Unknown* と表示されます。データの保持期間を制御するには、Index Lifecycle Management を参照してください。ハウスキーパーは Elasticsearch からデータを削除しないことに注意してください。

TimescaleDB を使用している場合、履歴、トレンド、および監査ログの圧縮セクションが利用可能になります。

Note:
TimescaleDB では、自動パーティショニングの利点を最大限に活用するために、履歴とトレンドについて アイテム履歴期間を上書きする、アイテムトレンド期間を上書きする、および 内部ハウスキーピングを有効にするを有効にしてください。これらのオプションが無効な場合でも、履歴およびトレンドのテーブルに保持されるデータは引き続きパーティショニングされますが、ハウスキーパーは期限切れのパーティションを削除せず、設定警告が表示されます。期限切れのパーティションの削除が有効な場合、Zabbix サーバーと Web インターフェースは削除されたアイテムを追跡しなくなり、それらのアイテムの履歴は期限切れのパーティションが削除されると消去されます。

ハウスキーパーは、デフォルトでは 1 時間ごとに自動実行されます。実行頻度は、`HousekeepingFrequency` Zabbix サーバー設定パラメータを指定して変更できます。

デフォルトに戻すボタンを使用すると、加えた変更を元に戻せます。

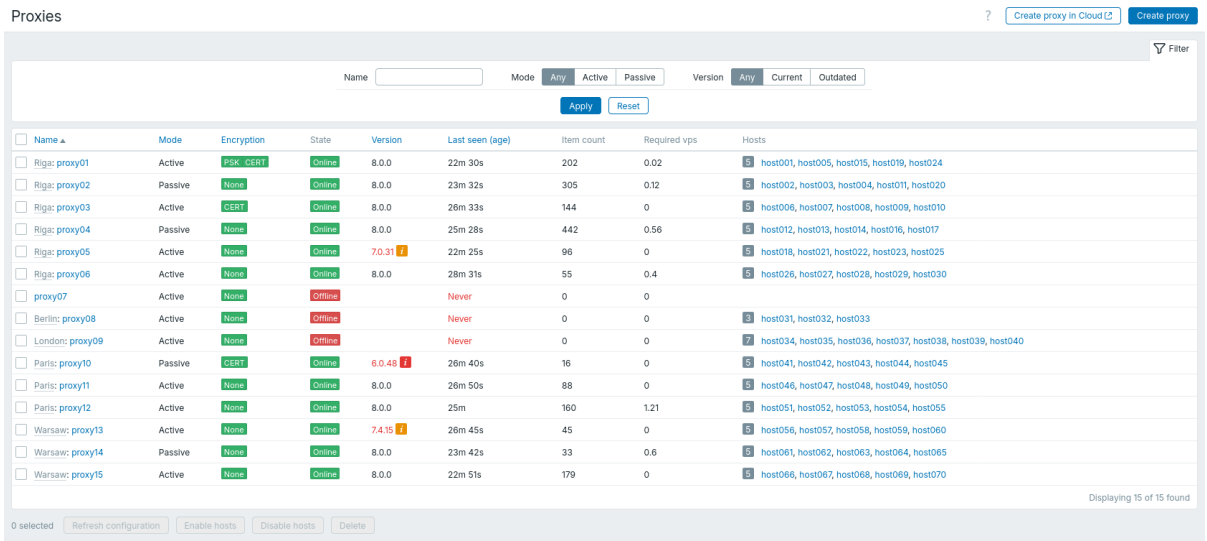
プロキシ

概要

Zabbix の Web インターフェースでは、管理 > プロキシセクションで分散監視用のプロキシを設定できます。

プロキシ

既存のプロキシとその詳細が一覧表示されます。



表示されるデータ:

列	説明
名前	プロキシの名前。 プロキシ名をクリックすると、プロキシ設定フォームが開きます。 プロキシがプロキシグループに属している場合、プロキシ名の前にグループ名がグレーのリンクとして表示されます。グループ名をクリックすると、プロキシグループ設定フォームが開きます。
モード	プロキシのモード - アクティブまたはパッシブ。
暗号化	プロキシからの接続における暗号化の状態: なし - 暗号化なし; **PSK** - 事前共有鍵を使用; 証明書 - 証明書を使用。
状態	プロキシの状態: 不明 - Zabbix サーバーが停止している間にプロキシが作成された、またはサーバーがまだ状態を更新していない; オンライン - フェイルオーバー期間内にプロキシが Zabbix サーバーと通信した（パッシブプロキシがサーバーのリクエストに応答した、またはアクティブプロキシがリクエストを送信した）; オフライン - フェイルオーバー期間内にプロキシが Zabbix サーバーと通信していない。
バージョン	プロキシのバージョン（3 桁のバージョン番号）。プロキシが古い、またはサポート対象外の場合、バージョン番号が強調表示（赤色）され、情報ステータスアイコン（黄色または赤色）が表示されます。詳細を確認するには、アイコンにマウスカーソルを合わせます。
最終確認時刻（経過時間）	サーバーが最後にプロキシを確認した時刻。

列	説明
アイテム数	プロキシに割り当てられた有効なホスト上の、有効なアイテムの数。
必要な *vps*	必要なプロキシのパフォーマンス（1 秒あたりに収集する必要がある値の数）。
ホスト	プロキシに割り当てられた有効なホストの数、およびプロキシによって監視されているホストの一覧。 ホスト名をクリックすると、ホスト設定フォームが開きます。

新しいプロキシを設定するには、右上隅にあるプロキシの作成ボタンをクリックします。Zabbix Cloud に新しいプロキシをデプロイするには、その隣にある *Cloud* でプロキシを作成ボタンをクリックします。これにより、新しいブラウザータブで Zabbix Cloud が開き、プロキシノードを作成できます。プロキシノードを作成したら、プロキシの作成ボタンを使用してプロキシを設定します。詳細については、Zabbix Cloud プロキシのセットアップを参照してください。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

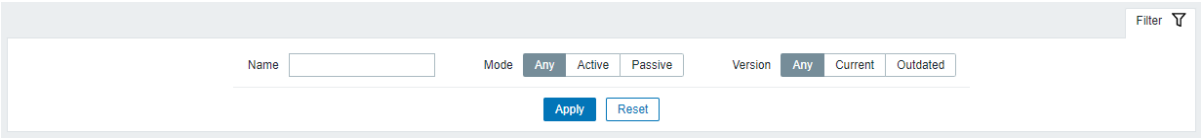
- 設定を更新 - プロキシの設定を更新します。
- ホストを有効化 - プロキシで監視されているホストのステータスを 監視中に変更します。
- ホストを無効化 - プロキシで監視されているホストのステータスを 監視対象外に変更します。
- 削除 - プロキシを削除します。

これらのオプションを使用するには、対象のプロキシの前にあるチェックボックスを選択してから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるプロキシのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決の状態で検索されます。

フィルターリンクは、プロキシの一覧の上にあります。クリックすると、プロキシを名前、モード、バージョンでフィルタリングできるフィルターが表示されます。フィルターオプションのサポート対象外では、古い（部分的にサポートされている）プロキシとサポートされていないプロキシの両方が表示されることに注意してください。



プロキシグループ

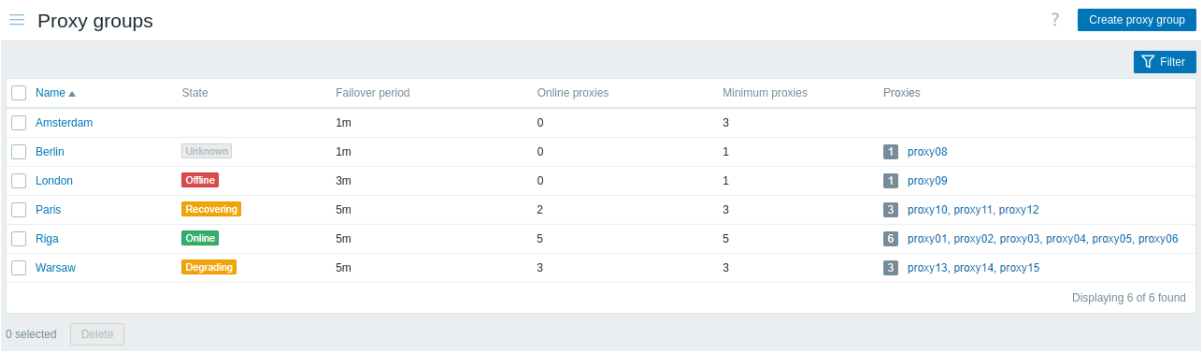
概要

管理 > プロキシグループでプロキシグループを設定できます。

プロキシグループは、プロキシ間でホストを自動的に分散し、プロキシ間の高可用性を実現するプロキシの負荷分散 で使用されます。

プロキシグループ

既存のプロキシグループとその詳細が一覧表示されます。



表示されるデータ:

列	説明
名前	プロキシグループの名前。プロキシグループ名をクリックすると、プロキシグループ設定フォームが開きます。

列	説明
状態	プロキシグループの状態（グループに少なくとも 1 台のプロキシが含まれている場合に表示）： 不明 - Zabbix サーバーの停止中にグループが作成された、またはサーバーがまだ状態を更新していない; オンライン - フェイルオーバー期間内に、プロキシの最小数が Zabbix サーバーと通信した; 劣化 - オンラインプロキシ数が最小プロキシ数のしきい値を下回ったため、グループがまもなくオフラインになる; オフライン - フェイルオーバー期間内に、最小数未満のプロキシしか Zabbix サーバーと通信していない; 復旧中 - オンラインプロキシ数が最小プロキシ数のしきい値に達したため、グループがまもなくオンラインに戻る。 プロキシグループの状態は、zabbix[proxy group,<name>,state]およびzabbix[proxy group,discovery]アイテムで監視できます。
フェイルオーバー期間	プロキシグループ内のプロキシがオンラインとみなされるために、Zabbix サーバーと通信する必要がある期間。
オンラインプロキシ	オンラインのプロキシ数（グループの最小数を下回っている場合は赤色で表示）。
最小プロキシ数	プロキシグループをオンラインに維持するために必要なオンラインプロキシの最小数。
プロキシ	グループ内のプロキシ数、およびグループ内のプロキシ一覧。各プロキシにはプロキシ設定フォームへのリンクがあります。 一覧表示されるプロキシの最大数は、テーブルセル内に表示する要素の最大数の値によって制限されます。

新しいプロキシグループを設定するには、右上隅にあるプロキシグループの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

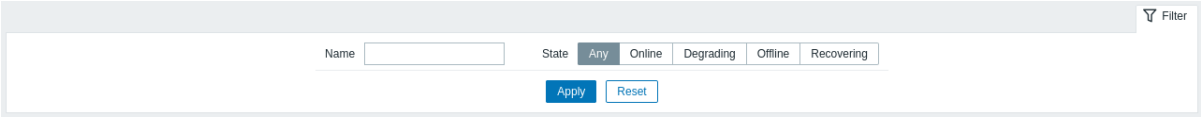
- 削除 - プロキシグループを削除します。

これらのオプションを使用するには、対象のプロキシグループの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるプロキシグループのみを表示できます。検索性能を向上させるため、データはマクロを展開せずに検索されます。

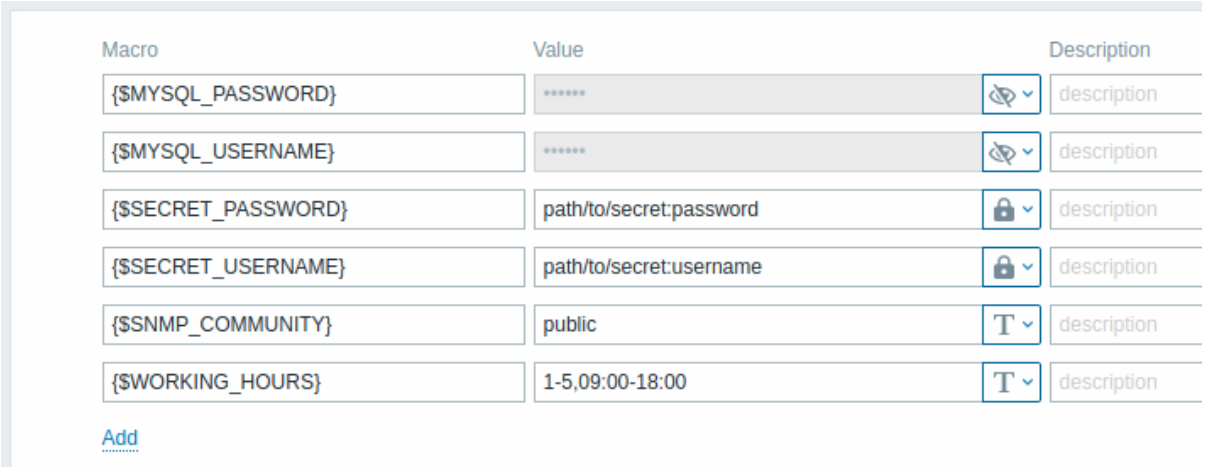
Filter リンクは、プロキシグループ一覧の上にあります。これをクリックするとフィルターが表示され、名前とステータスでプロキシグループを絞り込めます。



マクロ

概要

このセクションでは、システム全体で使用するユーザーマクロを名前と値のペアとして定義できます。マクロの値は、プレーンテキスト、シークレットテキスト、または Vault シークレットとして保存できます。説明の追加にも対応しています。



キュー

概要

管理 > キューセクションには、更新を待機しているアイテムが表示されます。

このセクションを開いたときに、キューにアイテムがないことを示す「緑色」の状態になっているのが理想的です。すべてのアイテムが遅延なく更新されていれば、待機中のアイテムはありません。しかし、サーバーのパフォーマンスが不足している場合、一部のアイテムの更新が遅延することがあり、その情報がこのセクションに表示されます。詳細については、キューセクションを参照してください。

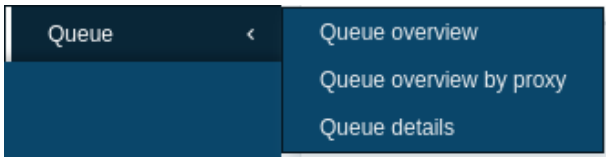
Note:
Zabbix プロキシはメンテナンス期間を認識しません。詳細については、メンテナンス中のキューの計算を参照してください。

Note:
キューは、Zabbix サーバーが実行中の場合にのみ使用できます。接続の問題やエージェントが正常に動作していないことによりアイテムのインターフェースが利用できなくなった場合、そのアイテムはキューにカウントされません。

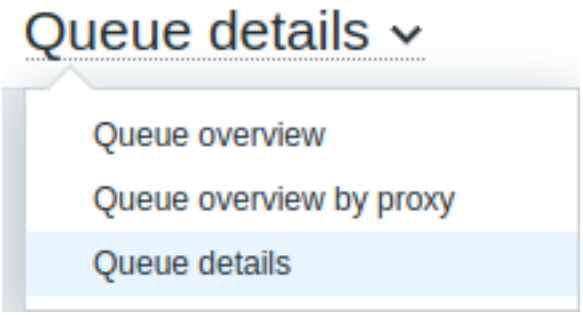
管理 > キューセクションには、次のページがあります。

- キューの概要 - アイテムのタイプ別にキューを表示します。
- プロキシ別のキューの概要 - プロキシ別にキューを表示します。
- キューの詳細 - 遅延しているアイテムの一覧を表示します。

利用可能なページの一覧は、管理メニューセクションで キューをクリックすると表示されます。左上にあるタイトルのドロップダウンを使用して、ページを切り替えることもできます。



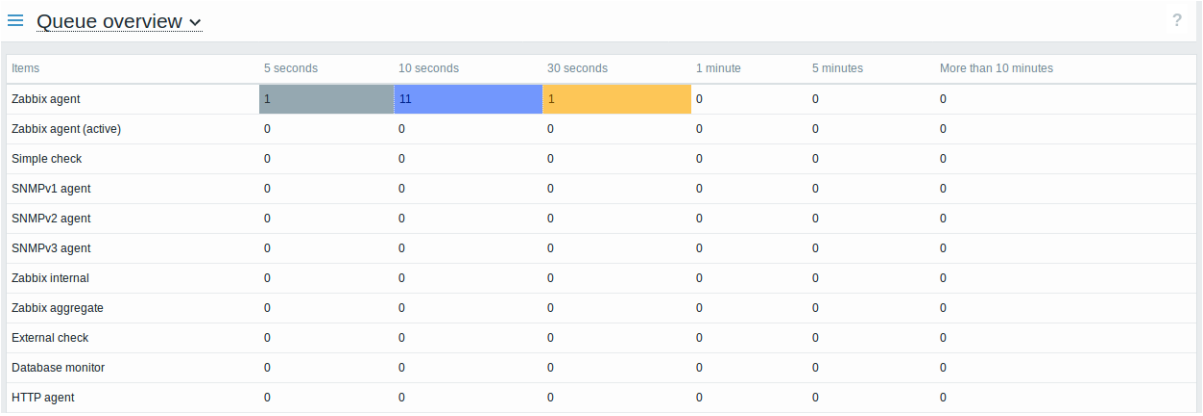
第 3 レベルのメニュー。



タイトルのドロップダウン。

キュー概要

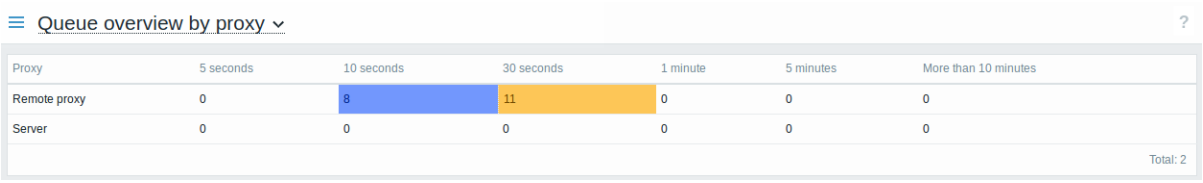
この画面では、問題が 1 つまたは複数のアイテムタイプに関連しているかどうかを簡単に特定できます。



各行にはアイテムタイプが含まれています。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5〜10 秒/10〜30 秒/30〜60 秒/1〜5 分/5〜10 分または 10 分以上です。

プロキシごとのキュー概要

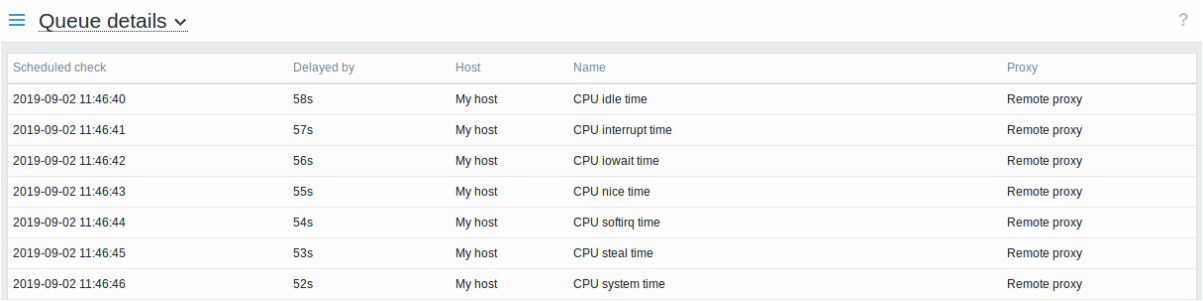
この画面では、問題がプロキシの 1 つに関連しているか、サーバーに関連しているかを簡単に特定できます。



各行にはプロキシが含まれ、サーバーはリストの最後にあります。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5〜10 秒/10〜30 秒/30〜60 秒/1〜5 分/5〜10 分または 10 分以上です。

待機中アイテムのリスト

この画面では、各待機中アイテムがリスト表示されます。



表示されるデータ:

列	説明
スケジュールされたチェック	チェックが実行される予定だった時刻が表示されます。
遅延時間	遅延の長さが表示されます。
ホスト	アイテムのホストが表示されます。
名前	待機中アイテムの名前が表示されます。
プロキシ	ホストがプロキシによって監視されている場合、プロキシ名が表示されます。

発生する可能性のあるエラーメッセージ

データが表示されず、次のエラーメッセージが表示される場合があります。



この場合のエラーメッセージは次のとおりです。

```
Cannot display item queue. Permission denied
```

これは、Web インターフェース設定ファイルの PHP 設定パラメータ $ZBX_SERVER、または $ZBX_SERVER と $ZBX_SERVER_PORT の両方が、異なるデータベースを使用する既存の Zabbix サーバーを指している場合に発生します。

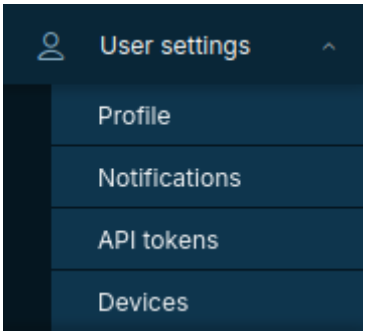
ユーザー設定

概要

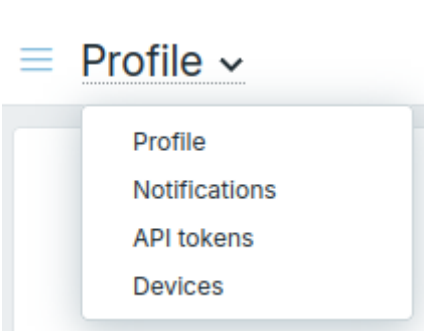
ユーザーの役割に応じて、ユーザー設定セクションには次のページが含まれる場合があります。

- プロフィールまたはユーザープロフィール - Zabbix の Web インターフェースの一部の機能をカスタマイズします。
- 通知 - 現在のユーザーの通知をカスタマイズします。
- *API* トークン - 現在のユーザーに割り当てられた API トークンを管理します。
- デバイス - 現在のユーザーが所有するモバイルデバイスを管理します。

利用可能なページの一覧は、Zabbix メニューの下部付近にある ユザーアイコンをクリックすると表示されます（*guest* ユーザーは利用できません）。左上のタイトルドロップダウンを使用して、ページを切り替えることもできます。



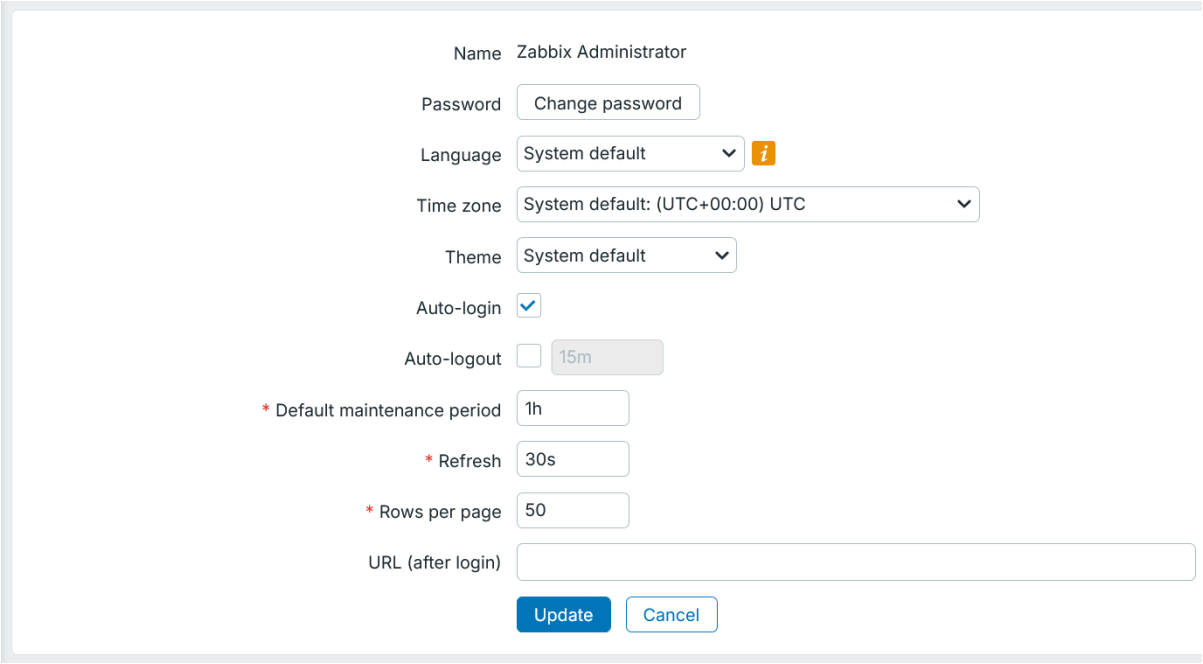
第 3 レベルメニュー。



タイトルドロップダウン。

ユーザープロファイル

ユーザープロファイルセクションでは、カスタムインターフェース言語、カラーテーマ、リストに表示する行数などを設定できます。ここで行った変更は、現在のユーザーにのみ適用されます。



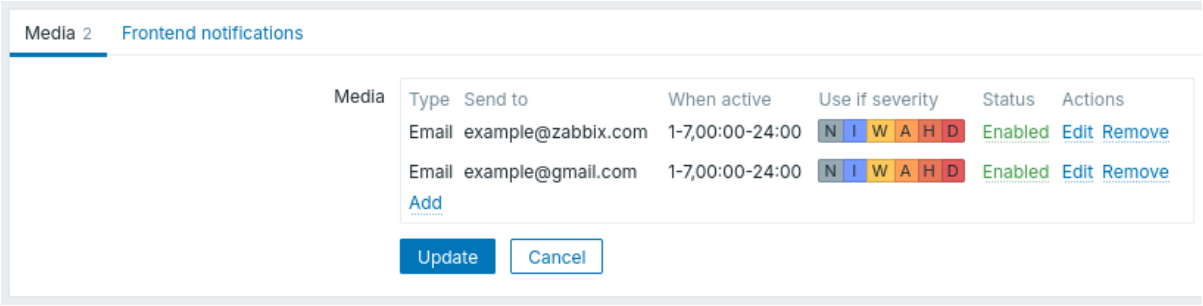
パラメーター	説明
名前	ユーザーの姓名が表示されます。 姓名がない場合は、ユーザー名が表示されます。
パスワード	パスワードの変更ボタンをクリックすると、古いパスワード、新しいパスワード、新しいパスワード（再入力）の 3 つのフィールドが表示されます。 パスワードの変更に成功すると、ユーザーはすべてのアクティブなセッションからログアウトされます。 パスワードを変更できるのは、Zabbix の内部認証を使用しているユーザーのみです。

パラメーター	説明
言語	使用するインターフェース言語を選択するか、システムのデフォルトを選択してシステムのデフォルト設定を使用します。 英語（*en_US*）を選択すると、Web インターフェースで米国式の時刻および日付形式も有効になります。 詳細については、追加の Web インターフェース言語のインストールを参照してください。
タイムゾーン	ユーザーレベルでグローバルタイムゾーンを上書きするタイムゾーンを選択するか、システムのデフォルトを選択してグローバルなタイムゾーン設定を使用します。
テーマ	プロファイル専用のカラーテーマを選択します。 システムのデフォルト - システムのデフォルト設定を使用します。 ブルー - 標準の青色テーマです。 ブルー（クラシック） - 従来のフォントを使用する標準の青色テーマです。 ダーク - 代替の暗色テーマです。 ダークブルー - 暗い青色のテーマです。 ダーク（クラシック） - 従来のフォントを使用する代替の暗色テーマです。 ハイコントラスト（ライト） - コントラストの高い明色テーマです。 ハイコントラスト（ダーク） - コントラストの高い暗色テーマです。
自動ログイン	このチェックボックスを選択すると、Zabbix にユーザー情報を記憶させ、30 日間自動的にログインできるようになります。 *30* 日間記憶するを有効にしてログインした場合： - 自動ログアウトタイマーがリセットされます（30 日間の期間が終了するまでセッションが維持されます）。 - 30 日以内の次回アクセス時に自動ログインが有効になります。 *30* 日間記憶するを無効にしてログインした場合： - 自動ログインパラメーターがクリアされます。 - 自動ログアウトは通常どおり、設定されたタイムアウトに従います。 この機能にはブラウザーの Cookie が使用されます。
自動ログアウト	このチェックボックスを選択すると、設定した秒数が経過した後に自動的にログアウトされます（最小 90 秒、最大 1 日）。 時間のサフィックスがサポートされます。例：90s、5m、2h、1d。 このオプションは、次の場合には機能しません。 * 監視メニューのページで、バックグラウンドの情報更新が実行されている場合。特定の時間間隔でデータを更新するページ（ダッシュボード、グラフ、最新データなど）を開いたままにすると、それに応じてセッションの有効期間が延長され、自動ログアウト機能が無効になります。 * *30* 日間ログイン状態を保持するオプションを選択してログインした場合。この場合、自動ログアウトタイマーは上書きされ、セッションは期間全体にわたってアクティブな状態が維持されます。 * 認証が SSO Identity Provider（IdP）を介して実行される場合。自動ログアウトで終了するのは Zabbix セッションのみです。IdP との SSO セッションはアクティブなままなので、ユーザーはパスワードを入力せずに再度ログインできる場合があります。 自動ログアウトには「0」も指定できます。これは、プロファイル設定の更新後に自動ログアウト機能が無効になることを意味します。
デフォルトのメンテナンス期間	アドホックメンテナンスに使用するデフォルトのメンテナンス期間を設定します。
更新	監視メニューのページに表示される情報を更新する頻度を設定します（最小 0 秒、最大 1 時間）。 時間のサフィックスがサポートされます。例：30s、90s、1m、1h。
1 ページあたりの行数	リストの 1 ページに表示する行数を設定します。行数（および表示するレコード数）を減らすと、読み込み時間が短縮されます。
URL（ログイン後）	ログイン後に表示する特定の URL を設定します。デフォルトのダッシュボードの代わりに、たとえば監視 > トリガーの URL を指定できます。

通知

通知セクションでは、現在のユーザー通知をカスタマイズするためのオプションを提供します。

メディアタブでは、通知の配信に使用するメディアタイプやアドレス、およびそれらを使用するタイミングなど、ユーザーのメディアの詳細を指定できます。



メディアタイプが無効になっている場合:

- 名前の後に黄色の情報アイコンが表示されます。
- ステータス列に「Disabled」と表示されます。

Note:
ユーザー自身のメディア詳細を変更する権限は、ユーザーロールに基づいて付与または取り消すことができます（自分のメディアの作成と編集オプションを参照）。Super admin ユーザーが他のユーザーのメディア詳細を変更する権限も、ユーザーロールに基づいて付与または取り消すことができます（ユーザーメディアの作成と編集オプションを参照）。

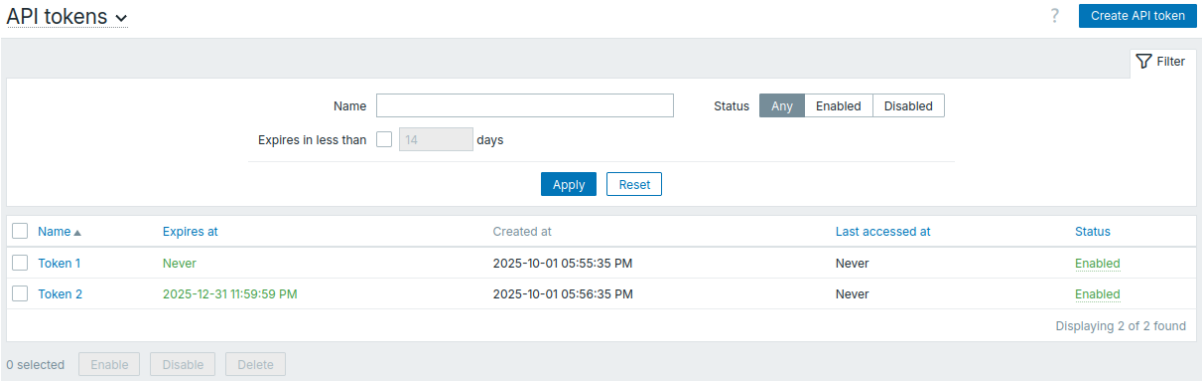
プロビジョニングされたユーザーについては、次の点に注意してください:

- プロビジョニングされたユーザーメディアは削除できません。
- プロビジョニングされたユーザーメディアは無効化/有効化できます。
- アクティブ時間, 深刻度条件、および 有効などのプロビジョニングされたユーザーメディアのフィールドは、手動で編集できます。
- プロビジョニングされたユーザーには、追加のユーザーメディア（たとえば追加のメールアドレス）を手動で追加できます。
- 手動で追加したユーザーメディアは削除できます。

Web インターフェース通知タブでは、グローバル通知を設定できます。

API トークン

API トークンセクションでは、ユーザーに割り当てられたトークンの表示、トークン詳細の編集、および新しいトークンの作成ができます。
このセクションは、ユーザー権限設定で *Manage API tokens* アクションが許可されている場合にのみ、ユーザーが利用できます。



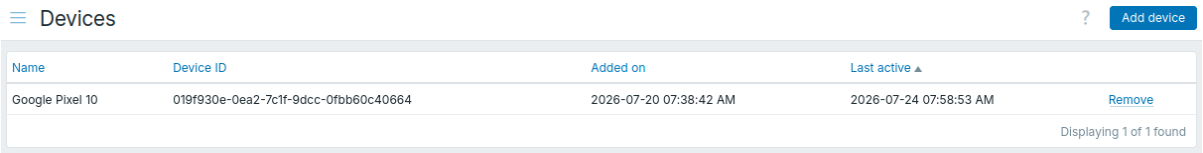
API トークンは、名前、有効期限、またはステータス（*Enabled/Disabled*）でフィルタリングできます。
一覧内のトークンのステータスをクリックすると、トークンをすばやく有効化/無効化できます。
また、一覧で複数のトークンを選択し、一覧の下にある *Enable/Disable* ボタンをクリックすることで、複数のトークンを一括で有効化/無効化することもできます。

Attention:
ユーザーは、Zabbix で自分に割り当てられたトークンの *Auth token* 値を確認できません。
Auth token 値は、トークン作成直後に一度だけ表示されます。
トークンを紛失した場合は、再生成する必要があります。

デバイス

デバイスセクションでは、Zabbix モバイルアプリで使用するモバイルデバイスを追加または削除できます。

このセクションは、`EnableMobileDevices`が有効になっている場合に表示されます。



デバイスを追加するには、次の手順に従います。

1. デバイスを追加をクリックします。
2. Zabbix モバイルアプリで QR コードをスキャンします。

Note:
デバイスを追加すると、Zabbix はプッシュ通知メディアタイプもユーザーメディアに追加します。デバイスを削除すると、そのユーザーメディアも削除されます。複数のプッシュ通知メディアタイプを設定している場合、Zabbix は有効になっていて、アルファベット順で最初に表示されるメディアタイプを追加します。

追加したデバイスは一覧に表示されます。デバイスの名前、ID、追加日時、最終アクティブ日時が表示されます。

デバイスを削除することもできます。デバイスを削除すると、そのアクセス認証情報が無効化され、Zabbix から削除されます。その後、そのデバイスは Zabbix にアクセスできなくなります。

スーパーユーザーは、ユーザー > デバイスで他のユーザーのデバイスを管理することもできます。

グローバル通知

概要

グローバル通知は、Zabbix フロントエンド内の現在の画面にリアルタイムの障害を直接表示する方法を提供します。

グローバル通知がない場合、障害やダッシュボードセクション以外で作業していると、現在の障害に関する情報を受け取ることができません。グローバル通知は、Zabbix フロントエンド内の現在の場所に関係なく、この情報が表示されることを保証します。

グローバル通知には、メッセージの表示とサウンドの再生の両方が含まれます。

Attention:
最近のブラウザバージョンでは、サウンドの自動再生が（デフォルトで）無効になっている場合があります。その場合は、この設定を手動で有効にする必要があります。

設定

グローバル通知は、通知セクションのフロントエンド通知タブでユーザーごとに有効にできます。

profile_c.png

パラメータ	説明
フロントエンド通知	チェックボックスをオンにすると、グローバル通知が有効になります。
メッセージタイムアウト	メッセージが表示される時間を設定します。デフォルトでは、メッセージは 60 秒間画面に表示されます。 時間のサフィックスがサポートされています。例: 30s, 5m, 2h, 1d。
サウンドを再生	サウンドが再生される時間を設定します。 一度だけ - サウンドが一度だけ完全に再生されます; **10** 秒 - サウンドが 10 秒間繰り返されます; メッセージタイムアウト - メッセージが表示されている間、サウンドが繰り返されます。
トリガーの深刻度	グローバル通知とサウンドを有効にするトリガーの深刻度を設定します。さまざまな深刻度に応じたサウンドも選択できます。 深刻度が 1 つも選択されていない場合、メッセージは表示されません。 また、復旧メッセージは選択された深刻度に対してのみ表示されます。たとえば、復旧と致命的が選択されている場合、致命的のトリガーの問題と復旧に対してグローバル通知が表示されます。
抑制された問題を表示	チェックボックスをオンにすると、ホストのメンテナンスによって抑制（非表示）されるはずの問題についても通知が表示されます。

グローバルメッセージの表示

メッセージが到着すると、画面右側のフローティングセクションに表示されます。このセクションは、セクションヘッダーをドラッグすることで自由に位置を変更できます。

global_messages.png

このセクションには、いくつかのコントロールがあります:

- スヌーズボタンは、現在アクティブなアラーム音を消音します。
- ミュート/ミュート解除ボタンは、アラーム音の再生/非再生を切り替えます。

ブラウザーでのサウンド

概要

サウンドはグローバル通知で使用されます。

Zabbix フロントエンドでサウンドを再生するには、ユーザープロファイルのフロントエンド通知タブでフロントエンド通知を有効にし、すべてのトリガーの深刻度にチェックを入れる必要があります。さらに、グローバル通知のポップアップウィンドウでサウンドを有効にする必要があります。

何らかの理由でデバイスでオーディオを再生できない場合、グローバル通知のポップアップウィンドウの ボタンは常に「ミュート」状態のままとなり、 ボタンにカーソルを合わせると「このデバイスでは通知オーディオをサポートできません」というメッセージが表示されます。

デフォルトのオーディオクリップを含むサウンドは、MP3 形式のみサポートされています。

Zabbix フロントエンドのサウンドは、Linux 上の最新の Firefox および Opera ブラウザ、Windows 上の Chrome、Firefox、Microsoft Edge、Opera ブラウザで正常にテストされています。

Attention:
最近のブラウザバージョンでは、サウンドの自動再生が（デフォルトで）無効になっている場合があります。その場合は、この設定を手動で有効にする必要があります。

グローバル検索

Zabbix Web インターフェースでは、ホスト、ホストグループ、テンプレート、およびテンプレートグループを検索できます。

検索入力ボックスは、メニュー内の Zabbix ロゴの下にあります。検索は、*Enter* を押すか、 検索アイコンをクリックして開始できます。



名前のどこかに入力した文字列を含むホストがある場合、ドロップダウンが表示され、そのようなホストがすべて一覧表示されます(一致した部分はオレンジ色で強調表示されます)。また、検索文字列として入力した技術名にホストの表示名が一致する場合も、そのホストがドロップダウンに表示されます。この場合、該当ホストは一覧に表示されますが、強調表示はされません。

検索可能な属性

ホストは以下のプロパティで検索できます:

- ホスト名

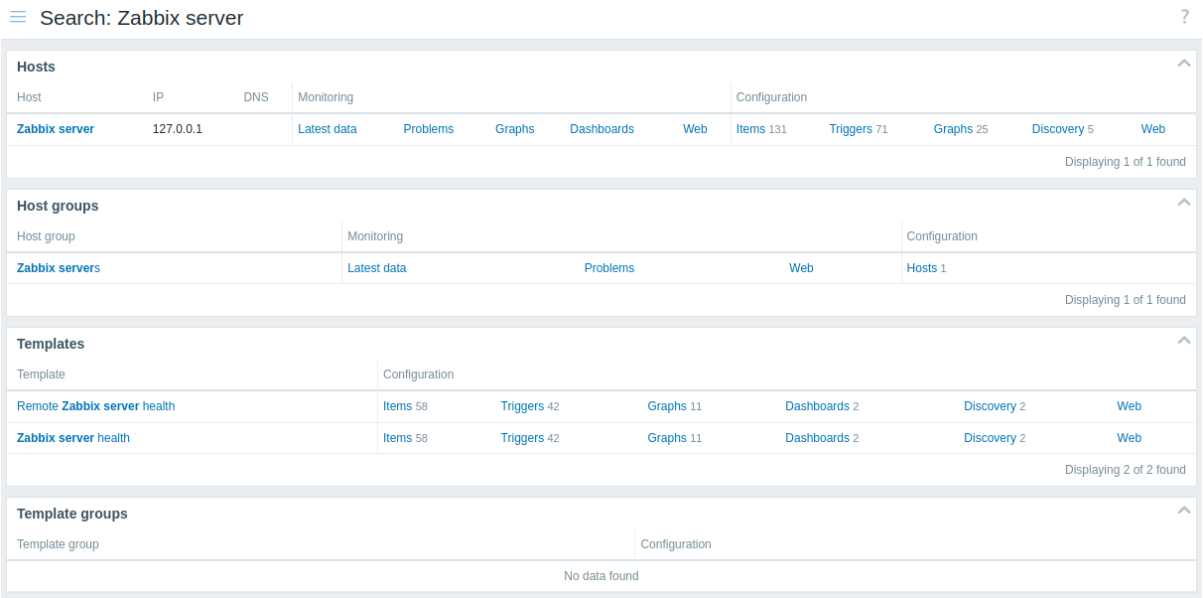
- 表示名
- IP アドレス
- DNS 名

テンプレートは名前または表示名で検索できます。表示名と異なる名前（テンプレート/ホストの）で検索した場合、検索結果では表示名の下に括弧付きで表示されます。

ホストグループおよびテンプレートグループは名前で検索できます。親グループを指定すると、すべてのネストされたグループが暗黙的に選択されます。

検索結果

検索結果は、ホスト、ホストグループ、テンプレート、テンプレートグループの 4 つの個別ブロックで構成されます。



各ブロックは個別に折りたたみ/展開できます。エントリ数は各ブロックの下部に表示されます。例えば、*13* 件中 *13* 件を表示のように表示されます。エントリがない場合、エントリ数は表示されません。1 つのブロック内に表示されるエントリ総数は 100 件に制限されています。

各エントリには、監視データおよび設定データへのリンクが用意されています。リンクの一覧については、完全なリストを参照してください。

すべての設定データ（アイテム、トリガー、グラフなど）については、見つかったエンティティ数がエンティティ名の横に灰色の数字で表示されます。エンティティ数が 0 の場合は、数字は表示されないことに注意してください。

有効なホストは青色、無効なホストは赤色で表示されます。

利用可能なリンク

各エントリについて、次のリンクを利用できます:

- Hosts
  - Monitoring
    - Latest data
    - Problems
    - Graphs
    - Host dashboards
    - Web scenarios
  - Configuration
    - Items
    - Triggers
    - Graphs
    - Discovery rules
    - Web scenarios
- Host groups
  - Monitoring
    - Latest data
    - Problems
    - Web scenarios
  - Configuration

        * Hosts
- Templates
    - Configuration
        * Items
        * Triggers
        * Graphs
        * Template dashboards
        * Discovery rules
        * Web scenarios
- Template groups
    - Configuration
        * Templates

Web インターフェースのメンテナンスモード

概要

Zabbix フロントエンドへのアクセスを制限するために、一時的に Zabbix フロントエンドを無効にすることができます。これは、ユーザーによる変更から Zabbix データベースを保護し、その整合性を維持するのに役立ちます。

Zabbix フロントエンドがメンテナンスモードの場合、データベースを安全に停止し、メンテナンス作業を実行できます。

定義された IP アドレスからのユーザーは、メンテナンスモード中も通常通りフロントエンドとやり取りできます。

設定

メンテナンスモードを有効にするには、`maintenance.inc.php` ファイル（Web サーバー上の Zabbix HTML ドキュメントディレクトリ内にある `/conf` に配置されています）を開き、次の行のコメントを解除します。

```
// メンテナンスモード。
define('ZBX_DENY_GUI_ACCESS', 1);

// Webインターフェースへの接続を許可するIPアドレスの配列（オプション）。
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// 警告画面に表示するメッセージ（オプション）。
$ZBX_GUI_ACCESS_MESSAGE = 'MySQLデータベースを15:00までアップグレードしています。しばらくお待ちください..
```

Note:
ほとんどの場合、`maintenance.inc.php` ファイルは、Web サーバー上の Zabbix HTML ドキュメントディレクトリにある `/conf` に配置されています。
ただし、一部のオペレーティングシステムや Web サーバーでは、別の場所が使用される場合があります。
たとえば、以下の場所にあります。

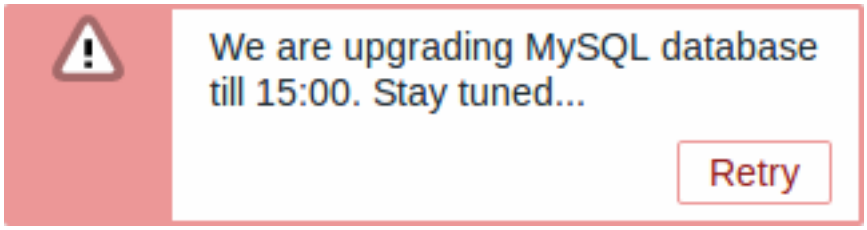
- SUSE および RedHat の場合は `/etc/zabbix/web/maintenance.inc.php`。
- Debian ベースのシステムの場合は `/usr/share/zabbix/conf/`。

PHP ファイルのコピーも参照してください。

Parameter	Details
ZBX_DENY_GUI_ACCESS	何らかの値を定義すると、メンテナンスモードが有効になります。 メンテナンスモードを無効にするには、コメントアウトするか削除してください。
ZBX_GUI_ACCESS_IP_RANGE	Web インターフェースへの接続を許可する IP アドレスの配列（オプション）。 例： `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	メンテナンスについてユーザーに通知するメッセージ（オプション）。 定義されていない場合は、デフォルトメッセージ *'Zabbix is under maintenance'* が使用されます。

表示

メンテナンスモード中に Zabbix フロントエンドにアクセスしようとすると、ユーザーには以下の画面が表示されます。メンテナンスが終了した際にユーザーの操作なしで通常の状態に戻るように、画面は 30 秒ごとに更新されます。



ZBX_GUI_ACCESS_IP_RANGE で定義された IP アドレスは、通常通りフロントエンドにアクセスできます。

ページパラメータ

概要

ほとんどの Zabbix Web インターフェースのページは、表示内容を制御するさまざまな HTTP GET パラメータをサポートしています。これらは、URL の後にパラメータ = 値のペアを指定し、URL とは疑問符（?）で区切り、パラメータ同士はアンパサンド（&）で区切ることで渡すことができます。

Monitoring > 障害

以下のパラメータがサポートされています:

Parameter	Description	Example
show	フィルターオプション *Show*。 指定可能な値: 1 - recent problems; 2 - all; 3 - in problem state.	show=1
name	フィルターオプション *Problem*: 自由形式の文字列。	name=Zabbix agent
severities	フィルターオプション *Severity*: 選択された深刻度の配列。形式は `severities[*]=*`（* を深刻度レベルに置き換えます）。 指定可能な値: 0 - not classified; 1 - information; 2 - warning; 3 - average; 4 - high; 5 - disaster.	severities[3]=3
inventory	フィルターオプション *Host inventory*: インベントリフィールドの配列 `[field],[value]`	inventory[0][field]=type&invento
evaltype	フィルターオプション *Tags*: タグの評価方法。 指定可能な値: 0 - and/or; 2 - or.	evaltype=0
tags	フィルターオプション *Tags*: 定義済みタグの配列 `[tag]`, `[operator]`, `[value]` 指定可能なoperatorの値: 0 - contains; 1 - equals; 2 - does not contain; 3 - does not equal; 4 - exists; 5 - does not exist.	tags[0][tag]=target&tags[0][ope

Parameter	Description	Example
show_tags	フィルターオプション *Show tags*。 指定可能な値: 0 - none; 1 - one; 2 - two; 3 - three.	show_tags=3
tag_name_format	フィルターオプション *Tag name*。 指定可能な値: 0 - full name; 1 - shortened name; 2 - none.	tag_name_format=1
tag_priority	フィルターオプション *Tag display priority*: タグ表示優先度をカンマ区切りで指定した文字列	tag_priority=customer, target
show_suppressed	フィルターオプション *Show suppressed problems*。 指定可能な値: 1 - show; 2 - do not show.	show_suppressed=1
acknowledgement_status	フィルターオプション *Acknowledgement status*。 指定可能な値: 0 - all; 1 - unacknowledged; 2 - acknowledged.	acknowledgement_status=0
acknowledged_by_me	フィルターオプション *By me*。`acknowledgement_status=2` の場合のみサポートされます。 指定可能な値: 0 - disabled; 1 - enabled.	acknowledged_by_me=1
compact_view	フィルターオプション *Compact view*。 指定可能な値: 0 - disabled; 1 - enabled.	compact_view=1
highlight_row	フィルターオプション *Highlight whole row*。 指定可能な値: 0 - disabled; 1 - enabled.	highlight_row=1
filter_name	フィルター属性オプション *Name*: 自由形式の文字列	filter_name=Databases
filter_show_counter	フィルター属性オプション *Show number of records*。 指定可能な値: 0 - disabled; 1 - enabled.	filter_show_counter=1
filter_custom_time	フィルター属性オプション *Set custom time period*。 指定可能な値: 0 - disabled; 1 - enabled.	filter_custom_time=1
sort	並べ替え対象の列。 指定可能な値: clock - *Time* 列で並べ替え; host - *Host* 列で並べ替え; severity - *Severity* 列で並べ替え; name - *Problem* 列で並べ替え.	sort=clock

Parameter	Description	Example
sortorder	結果の並べ替え順。 指定可能な値: DESC - 降順; ASC - 昇順.	sortorder=DESC
age_state	フィルターオプション *Age less than*。show=3 の場合のみサポートされます。 指定可能な値: 0 - age パラメータを無効化; 1 - age パラメータを有効化.	age_state=1
age	フィルターオプション *Age less than*: 整数、日数。age_state=1 かつ show=3 の場合のみサポートされます。	age=7
groupids	フィルターオプション *Host groups*: ホストグループ ID の配列。	groupids[]=4
hostids	フィルターオプション *Hosts*: ホスト ID の配列	hostids[]=10084
triggerids	フィルターオプション *Triggers*: トリガー ID の配列	triggerids[]=22382
show_timeline	フィルターオプション *Show timeline*。compact_view=1 ではサポートされません。 指定可能な値: 0 - 表示しない; 1 - 表示する.	show_timeline=1
details	フィルターオプション *Show details*。 指定可能な値: 0 - 表示しない; 1 - 表示する.	details=1
from	日付範囲の開始。相対指定も可能です（例: now-1m）。 filter_custom_time=1 の場合のみサポートされます	from=now-2h
to	日付範囲の終了。相対指定も可能です（例: now-1m）。 filter_custom_time=1 の場合のみサポートされます	to=now

関連項目: Problems ページのフィルターオプション。

キオスクモード

サポートされている Web インターフェースのページでは、URL パラメータを使用してkiosk modeを有効にできます。例えば、ダッシュボードでは次のように指定します。

- /zabbix.php?action=dashboard.view&kiosk=1 - キオスクモードを有効にする
- /zabbix.php?action=dashboard.view&kiosk=0 - 通常モードを有効にする

スライドショー

ダッシュボードでスライドショーを有効にすることができます:

- /zabbix.php?action=dashboard.view&slideshow=on - スライドショーを有効にする

定義

概要

フロントエンドの多くの項目はフロントエンド自体で設定できますが、現在一部のカスタマイズは定義ファイルを編集することでのみ可能です。

このファイルは、Zabbix HTML ドキュメントディレクトリの/include にある defines.inc.php です。

パラメータ

このファイル内でユーザーにとって関心がある可能性のあるパラメータ:

- ZBX_MIN_PERIOD

グラフの最小期間。秒単位です。デフォルトは 1 分です。

- GRAPH_YAXIS_SIDE_DEFAULT

シンプルグラフにおける Y 軸の既定の位置、およびカスタムグラフにアイテムを追加する際のドロップダウンボックスの既定値。指定可能な値: 0 - 左、1 - 右。

デフォルト: 0

- ZBX_SESSION_NAME

Zabbix Web インターフェースのセッション Cookie 名として使用される文字列。

デフォルト: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Vault response のデータキャッシュを無効化するために使用される、秒単位の TTL タイムアウト。Vault response のキャッシュを無効にするには 0 を設定します。

デフォルト: 60

- SUBFILTER_VALUES_PER_GROUP

グループごとのサブフィルター値の数（たとえば、最新データ のサブフィルター）。

デフォルト: 1000

- ZBX_MAX_WIDGET_LINES

表示するウィジェット行の最大数。

デフォルト: 1000

追加の **Web** インターフェース言語

概要　Zabbix Web インターフェースで英語以外の言語を使用するには、そのロケールを Web サーバーにインストールする必要があります。さらに、翻訳を機能させるには PHP gettext 拡張が必要です。

ロケールのインストール　インストール済みのすべての言語を一覧表示するには、次を実行します。

```
locale -a
```

必要な言語が一覧にない場合は、`/etc/locale.gen` ファイルを開き、必要なロケールのコメントアウトを解除します。Zabbix は UTF-8 エンコーディングを使用するため、UTF-8 文字セットのロケールを選択する必要があります。

次に、次を実行します。

```
locale-gen
```

Web サーバーを再起動します。Web サーバーで PHP-FPM がインストールされ使用されている場合は、Web サーバーとあわせて再起動する必要があります。

これでロケールがインストールされているはずです。新しい言語を表示するには、ブラウザで Zabbix Web インターフェースのページを Ctrl + F5 で再読み込みする必要がある場合があります。

Zabbix のインストール　Zabbix git repository から Zabbix を直接インストールする場合、翻訳ファイルは手動で生成する必要があります。
翻訳ファイルを生成するには、次を実行します。

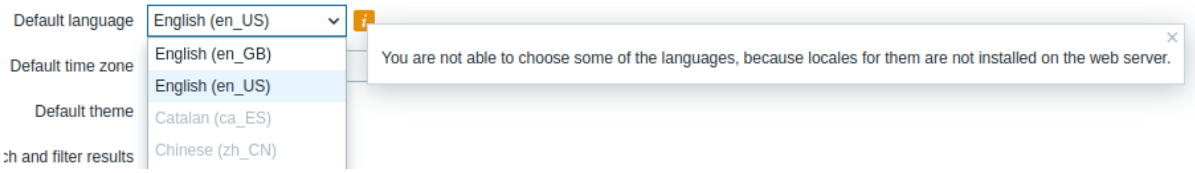
```
make gettext
locale/make_mo.sh
```

Zabbix をパッケージまたはソースの `tar.gz` ファイルからインストールする場合、この手順は不要です。

言語の選択　Zabbix Web インターフェースで言語を選択する方法はいくつかあります。

- Web インターフェースのインストール時 - Web インターフェースのインストールウィザードで選択します。選択した言語はシステムのデフォルトとして設定されます。
- インストール後は、システムのデフォルト言語を *Administration* > *General* > *GUI* で変更できます。
- 特定のユーザーの言語は、ユーザープロファイルで変更できます。

ある言語のロケールがマシンにインストールされていない場合、その言語は Zabbix の言語セレクターでグレー表示されます。少なくとも 1 つのロケールが不足している場合、言語セレクターの横にオレンジ色のアイコンが表示されます。このアイコンをクリックすると、次のメッセージが表示されます。”You are not able to choose some of the languages, because locales for them are not installed on the web server.”



独自のテーマを作成する

概要

デフォルトでは、Zabbix にはあらかじめ定義された Web インターフェースthemes がいくつか用意されています。カスタムテーマを作成するには、ここで説明する手順に従ってください。

ステップ 1: テーマを作成する

独自のテーマを作成するには:

1. CSS ファイルを新規作成するか、assets/styles/ ディレクトリから既存のテーマ（たとえば blue-theme.css、dark-theme.css、または別のテーマ）をコピーして変更します。
2. テーマ（たとえば custom-theme.css）を assets/styles/ ディレクトリに保存します。

ステップ 2: テーマ一覧でテーマを有効にする

カスタムテーマを Web インターフェースで使用できるようにするには、APP::getThemes() メソッドが返すテーマ一覧に追加する必要があります。これは、次のように APP クラス内で ZBase::getThemes() メソッドをオーバーライドして行います。

1. include/classes/core/APP.php ファイルを開き、APP クラスを見つけます。

```
class APP extends ZBase {

}
```

2. クラス内の最後の閉じ括弧 } の前に、次のメソッドを追加します。

```
public static function getThemes() {
        return array_merge(parent::getThemes(), [
            'custom-theme' => _('Custom theme')
        ]);
}
```

ここで、

- custom-theme - テーマの内部識別子。.css 拡張子を除いた CSS ファイル名と一致している必要があります。
- Custom theme - Web インターフェースに表示される表示名。

複数のテーマを追加するには、配列内に各エントリをコンマで区切って記述します。最後のエントリの後のコンマは省略できます。

例:

```
    public static function getThemes() {
        return array_merge(parent::getThemes(), [
            'custom-theme' => _('Custom theme'),
            'anothertheme' => _('Another theme'),
            'onemoretheme' => _('One more theme')
        ]);
    }
```

ステップ 3: グラフの色を変更する

グラフの色を変更するには、新しいテーマのレコードを graph_theme データベーステーブルに追加する必要があります。

MySQL/MariaDB データベースにレコードを追加する例:

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

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

ステップ 4: 新しいテーマを使用する

Zabbix の Web インターフェースでは、このテーマをデフォルトテーマとして設定するか、ユーザープロファイルで選択できます。

デバッグモード

概要

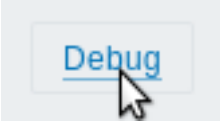
デバッグモードは、フロントエンドページの問題を診断するために使用できます。

設定

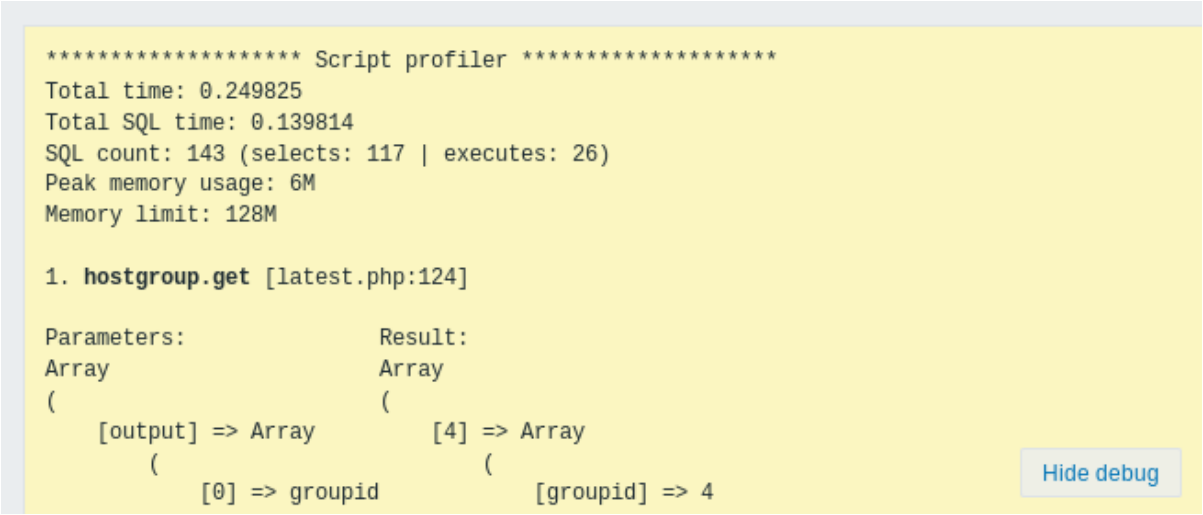
デバッグモードは、ユーザーグループに属する個々のユーザーに対して有効化できます。

- ユーザーグループ を設定する際
- 設定済みのユーザーグループ を表示する際

ユーザーグループで *Debug mode* が有効になっている場合、そのユーザーにはブラウザウィンドウの右下隅に *Debug* ボタンが表示されます。



Debug ボタンをクリックすると、ページ内容の下に新しいウィンドウが開き、ページの SQL 統計情報と、API 呼び出しおよび個々の SQL 文の一覧が表示されます。



ページのパフォーマンスに問題がある場合、このウィンドウを使用して問題の根本原因を調査できます。

Warning:
Debug mode を有効にすると、Web インターフェースのパフォーマンスに悪影響を及ぼします。

Zabbix で使用される **Cookie**

概要

このページでは、Zabbix で使用されるクッキーの一覧を提供します。

名前	説明	値	有効期限/最大年齢	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	Zabbix フロントエンドのセッションデータ。base64 でエンコードされた JSON として保存される		セッション（ブラウジングセッションが終了すると期限切れ）	+	+（Web サーバーで HTTPS が有効な場合のみ）
tab	アクティブなタブ番号。このクッキーは複数のタブを持つページ（例：ホスト、トリガー、アクションの設定ページ）でのみ使用され、ユーザーがプライマリタブから他のタブ（タグや依存関係タブなど）に移動したときに作成されます。 0 はプライマリタブを表します。	例：1	セッション（ブラウジングセッションが終了すると期限切れ）	-	-
browserwarning_ignore	古いブラウザを使用していることに関する警告を無視するかどうか。	yes	セッション（ブラウジングセッションが終了すると期限切れ）	-	-
system-message-ok	ページがリロードされた直後に表示するメッセージ。	プレーンテキストメッセージ	セッション（ブラウジングセッションが終了すると期限切れ）またはページがリロードされた直後	+	-
system-message-error	ページがリロードされた直後に表示するエラーメッセージ。	プレーンテキストメッセージ	セッション（ブラウジングセッションが終了すると期限切れ）またはページがリロードされた直後	+	-

Note:
Web サーバーのディレクティブによる Zabbix クッキーへの'HttpOnly' フラグの強制はサポートされていません。

脚注

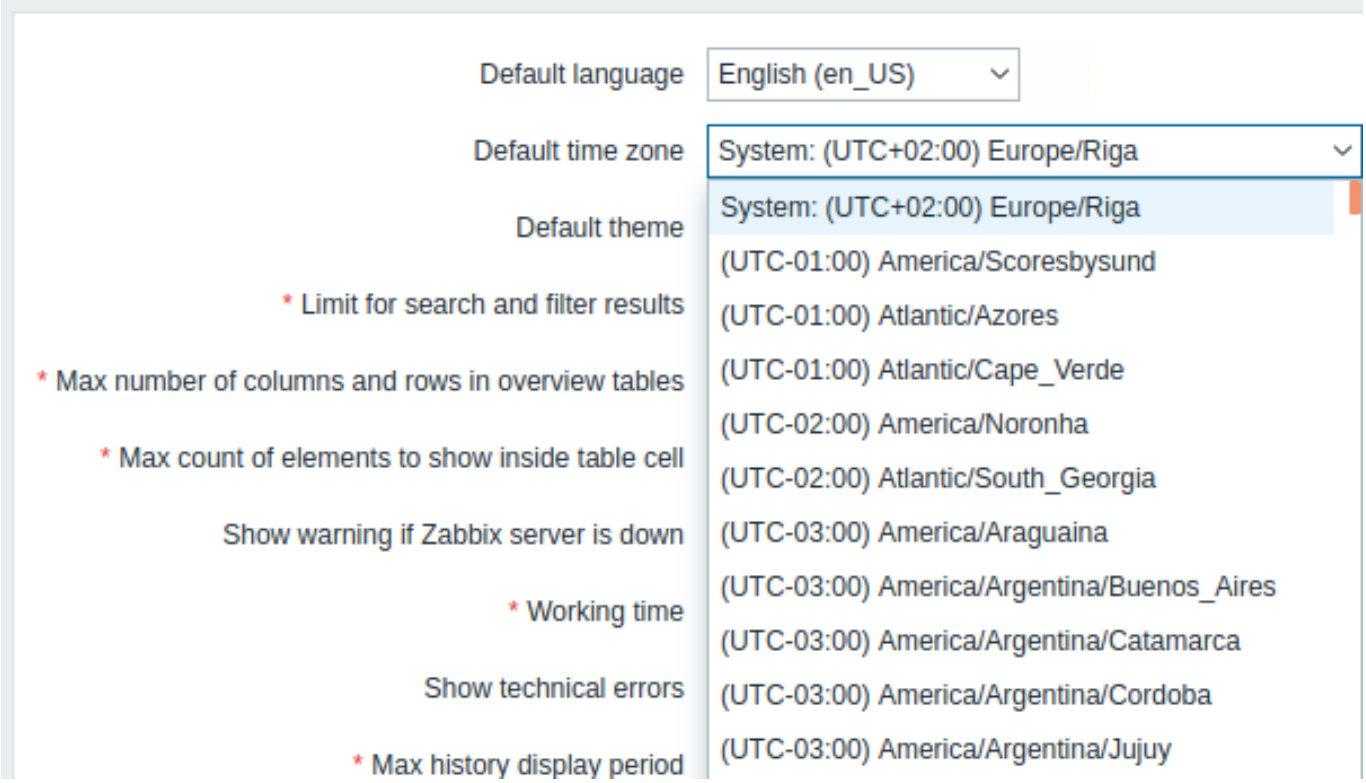
[1] `HttpOnly` が'true' の場合、クッキーは HTTP プロトコルを通じてのみアクセス可能になります。つまり、JavaScript などのスクリプト言語からはクッキーにアクセスできません。この設定は、XSS 攻撃による ID 盗難のリスクを効果的に減らすのに役立ちます（ただし、すべてのブラウザでサポートされているわけではありません）。

[2] `Secure` は、クッキーがクライアントから安全な HTTPS 接続を介してのみ送信されるべきであることを示します。'true' に設定すると、安全な接続が存在する場合にのみクッキーが設定されます。

タイムゾーン

概要

Web インターフェースのタイムゾーンは、Web インターフェース全体に対して設定でき、個々のユーザーごとに調整することもできます。



System を選択すると、Web インターフェースでは Web サーバーのタイムゾーン（`php.ini` に `date.timezone` が設定されている場合はその値を含む）が使用されます。一方、Zabbix サーバーでは、実行されているマシンのタイムゾーンが使用されます。

Note:
Zabbix サーバーは、通知内のマクロを展開するとき（例: `{EVENT.TIME}` はユーザーごとに異なるタイムゾーンに展開される場合があります）と、通知を送信する時間制限（ユーザーメディアの設定の「有効な時間帯」設定を参照）にのみ、指定されたグローバルまたはユーザーのタイムゾーンを使用します。

タイムゾーンの選択は、Web インターフェースの時刻および日付の形式には影響しません。代わりに、インターフェースの言語を変更できます（インストール時、またはユーザー設定で変更可能です）。*English (en_US)* を選択すると、Web インターフェースで米国式の時刻および日付形式も有効になります。

設定

グローバルなタイムゾーン:

- インストール時に Web インターフェースで手動設定できます。
- *Administration* > *General* > *GUI* で変更できます。

ユーザーレベルのタイムゾーン:

- ユーザーの設定/更新時に設定できます。
- 各ユーザーが自分のユーザープロファイルで設定できます。

関連情報: スケジューリング間隔を使用する際のタイムゾーンの整合。

パスワードのリセット

概要　このセクションでは、Zabbix のユーザーパスワードをリセットする手順について説明します。

手順　Zabbix のパスワードを忘れてログインできない場合は、Zabbix 管理者に問い合わせてください。

スーパーユーザーは、すべてのユーザーのパスワードをuser configuration formで変更できます。

スーパーユーザーがパスワードを忘れてログインできない場合は、次の SQL クエリを実行して、スーパーユーザーにデフォルトパスワードを適用する必要があります（'Admin'を適切なスーパーユーザー名に置き換えてください）。

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

このクエリを実行すると、ユーザーのパスワードは *zabbix* に設定されます。初回ログイン時に、必ずデフォルトパスワードを変更してください。

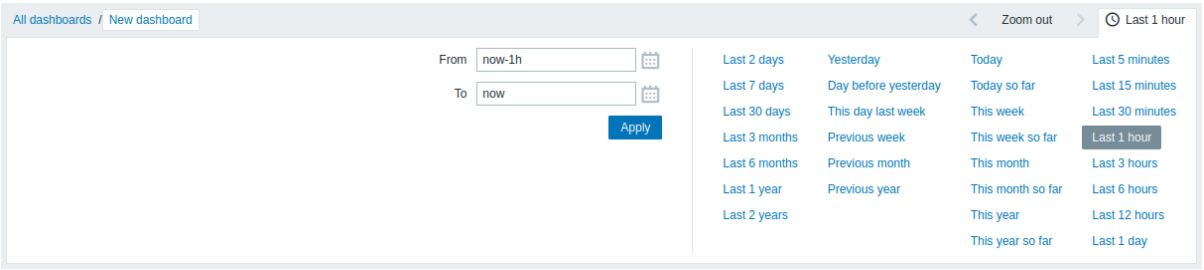
時間期間とホストセレクター

概要

このページでは、ダッシュボードで使用できる期間およびホストのセレクターについて説明します。これらのセレクターを使用すると、個々のウィジェット設定を編集せずに、互換性のあるすべてのダッシュボードウィジェットで表示されるデータを動的に調整できます。

期間セレクター

期間セレクターを使用すると、頻繁に必要となる期間をマウスのワンクリックで選択できます。ダッシュボードの右上にある期間セレクターボタンをクリックすると、展開または折りたたみができます。



Today、*This week* などのオプションでは、将来の時間や日を含む期間全体が表示されます。*Today so far*、*This week so far* などのオプションでは、経過した時間のみが表示されます。

期間を選択すると、の矢印ボタンをクリックして、時間を前後に移動できます。*Zoom out* ボタンを使用すると、期間を両方向に 50% ずつ縮小できます。

Note:
グラフ（*Custom* 期間を使用するグラフを除く）では、グラフ内の領域をマウスの左ボタンで選択することでも、表示する期間を選択できます。マウスの左ボタンを離すと、グラフが選択した領域にズームインします。グラフ内をダブルクリックしてズームアウトすることもできます。

From/*To* フィールドには、選択した期間が絶対時間構文（`Y-m-d H:i:s` 形式）または相対時間構文で表示されます。相対時間の期間には、1 つまたは複数の算術演算（- または +）を含めることができます。たとえば、`now-1d` や `now-1d-2h+5m` のように指定します。

次の相対時間の略語がサポートされています。

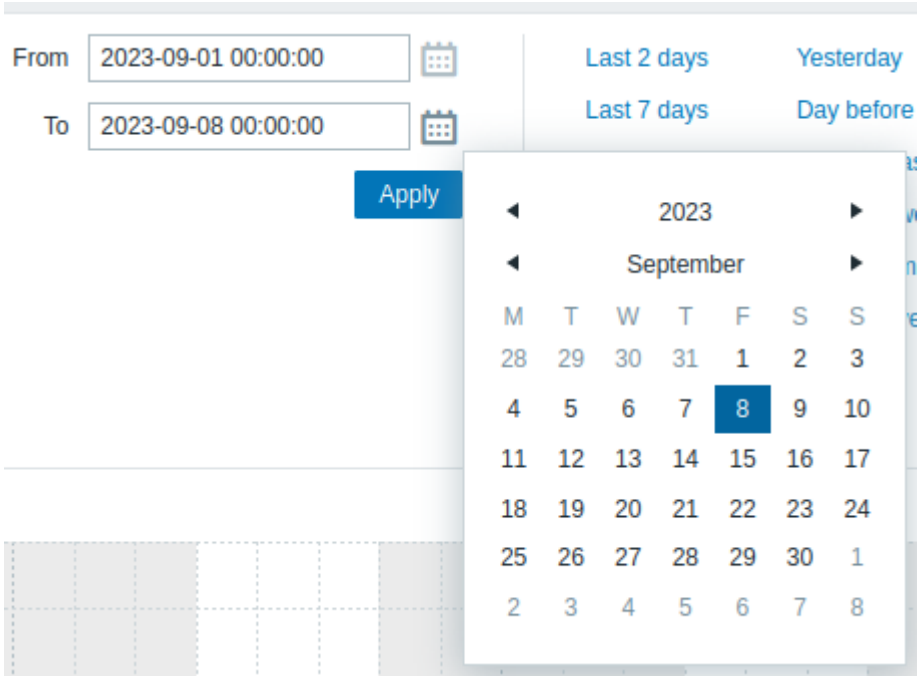
- `now`
- `s`（秒）
- `m`（分）
- `h`（時間）
- `d`（日）

- w（週）
- M（月）
- y（年）

期間セレクターでは精度もサポートされています（たとえば、now-1d/M の /M）。精度の詳細は次のとおりです。

精度	From	To
m	Y-m-d H:m:00	Y-m-d H:m:59
h	Y-m-d H:00:00	Y-m-d H:59:59
d	Y-m-d 00:00:00	Y-m-d 23:59:59
w	週の月曜日 00:00:00	週の日曜日 23:59:59
M	月の 1 日 00:00:00	月の最終日 23:59:59
y	年の 1 月 1 日 00:00:00	年の 12 月 31 日 23:59:59

日付ピッカーを使用して期間を選択することもできます。開くには、*From/To* フィールドの横にあるカレンダーアイコンをクリックします。



Note:
日付ピッカー内では、Tab、Shift+Tab、キーボードの矢印ボタンを使用して、年、月、日の間を移動できます。Enter を押すと選択が確定します。

例:

From	To	選択した期間
now/d	now/d	本日の 00:00 - 23:59
now/d	now/d+1d	本日の 00:00 - 明日の 23:59
now/w	now/w	今週の月曜日 00:00:00 - 日曜日 23:59:59
now-1y/w	now-1y/w	1 年前の週の月曜日 00:00:00 - 日曜日 23:59:59

Attention:
To パラメータに”now/M+1M” を使用すると、31 日が加算され、月の日数によっては日付が 1~3 日ずれる場合があります。たとえば、1 月に使用すると、期待される 2 月 28 日ではなく 3 月 2 日になることがあります。この問題を回避するには、”now/M-3d+1M/M” を使用してください。これにより、月の日数に正確に調整されます。同様に、*From* を過去に設定する場合は、”now/M+3d-1M/M” を使用してください。

ホストセレクター

ホストセレクターを使用すると、互換性のあるダッシュボードウィジェットにデータを表示するホストを選択できます。

ダッシュボードを編集して保存すると、少なくとも 1 つのウィジェットで、ホストの上書きパラメータがダッシュボードをデータソースとして使用するように設定されている場合に利用可能になります。



設定パラメータ**: Zabbix Web** インターフェース

概要　設定パラメータを使用すると、Zabbix の Web インターフェースの動作をカスタマイズできます。

パラメータ値は、Web インターフェース設定ファイル（zabbix.conf.php）で指定します。

このページは次のように構成されています。

- パラメータの概要（パラメータの詳細を表示するには、その名前をクリックします）
- パラメータの詳細

パラメータ	説明
$DB['TYPE']	データベースの種類。
$DB['SERVER']	データベースのホスト名または IP アドレス。
$DB['PORT']	データベースのポート。
$DB['DATABASE']	Zabbix の Web インターフェースが設定済みのデータベースへの接続に使用するデータベ
$DB['USER']	Zabbix の Web インターフェースがデータベースへの接続に使用するユーザー名。
$DB['PASSWORD']	データベースへの接続に使用するパスワード。
$DB['SCHEMA']	PostgreSQL で使用するデータベーススキーマ名。
$DB['ENCRYPTION']	Zabbix の Web インターフェースとデータベース間の接続における TLS 暗号化を有効また
$DB['KEY_FILE']	データベースとの TLS 接続に使用するクライアント秘密鍵ファイルへのパス。
$DB['CERT_FILE']	データベースとの TLS 接続に使用するクライアント証明書ファイルへのパス。
$DB['CA_FILE']	データベースとの TLS 接続に使用する CA 証明書ファイルへのパス。
$DB['VERIFY_HOST']	TLS 証明書内のデータベースサーバーのホスト名の検証を有効または無効にします。
$DB['CIPHER_LIST']	データベース接続で許可する TLS 暗号の一覧。
$DB['VAULT']	データベース認証情報の保存に使用する Vault シークレットマネージャーを指定します。
$DB['VAULT_URL']	データベース認証情報の取得に使用する Vault シークレットマネージャーの URL。
$DB['VAULT_PREFIX']	データベース認証情報の取得に使用する Vault パスまたはクエリのプレフィックス。
$DB['VAULT_DB_PATH']	Vault シークレットマネージャー内のデータベース認証情報へのパス。
$DB['VAULT_TOKEN']	Vault シークレットマネージャーへのアクセスに使用する認証トークン。
$DB['VAULT_APP_ROLE_ID']	HashiCorp Vault への認証に使用する AppRole ロール ID。
$DB['VAULT_APP_SECRET_ID']	HashiCorp Vault への認証に使用する AppRole シークレット ID。
$DB['VAULT_CERT_FILE']	Vault シークレットマネージャーへの認証に使用するクライアント証明書ファイルへのパ
$DB['VAULT_KEY_FILE']	Vault シークレットマネージャーへの認証に使用するクライアント秘密鍵ファイルへのパ
$DB['VAULT_CACHE']	Vault シークレットマネージャーから取得した認証情報のキャッシュを有効または無効にし
$ZBX_SERVER	Zabbix のホスト名/IP アドレスを上書きする値を指定します。
$ZBX_SERVER_PORT	Zabbix サーバーのポートを上書きする値を指定します。
$ZBX_SERVER_NAME	Zabbix サーバー名。
$IMAGE_FORMAT_DEFAULT	デフォルトの画像形式。
$HISTORY_PROVIDERS[]	Elasticsearch または ClickHouse の履歴ストレージプロバイダーの設定。
$SSO['SETTINGS']	SAML サインインの設定。
$SSO['CERT_STORAGE']	SAML 証明書の保存場所を指定します。
$SSO['SP_KEY']	秘密鍵をファイルシステムに保存する場合の、SAML サービスプロバイダー（SP）秘密鍵
$SSO['SP_CERT']	証明書をファイルシステムに保存する場合の、SAML サービスプロバイダー（SP）証明書
$SSO['IDP_CERT']	証明書をファイルシステムに保存する場合の、SAML アイデンティティプロバイダー証明
$ZBX_FEATURE_FLAGS['banners_enabled']	バナーのサポートを有効または無効にします。
$ZBX_FEATURE_FLAGS['http_auth_enabled']	ユーザーの HTTP 認証を有効または無効にします。
$ZBX_FEATURE_FLAGS['modules_config_enabled']	モジュールへのアクセスを有効または無効にします。
$ZBX_FEATURE_FLAGS['media_type_denylist']	事前設定された特定のメディアタイプの編集を無効にします。
$ZBX_SERVER_TLS['ACTIVE']	Zabbix サーバーへの接続で TLS が有効かどうかを指定します。
$ZBX_SERVER_TLS['CA_FILE']	サーバー証明書の検証に使用する認証局（CA）証明書ファイルへのパス。
$ZBX_SERVER_TLS['KEY_FILE']	クライアント証明書に対応するクライアント秘密鍵ファイルへのパス。
$ZBX_SERVER_TLS['CERT_FILE']	相互 TLS 認証が必要な場合のクライアント証明書ファイルへのパス。
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	サーバー証明書と照合する発行者の識別名（DN）。
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	サーバー証明書と照合するサブジェクトの識別名（DN）。

パラメータ	説明
$TELEMETRY_PROVIDERS[]	テレメトリデータソースの設定。

パラメータの詳細

$DB['TYPE']

データベースタイプ。

値: MYSQL - MySQL、POSTGRESQL - PostgreSQL。

$DB['SERVER']

データベースのホスト名または IP アドレス。

$DB['SERVER'] が定義されていない場合は、デフォルトの Unix ドメインソケットが使用されます。MySQL の場合、$DB['SERVER'] に localhost を設定すると、デフォルトの Unix ソケット経由で接続されます。
 PostgreSQL の場合、$DB['SERVER'] には、コンマで区切った複数の host:port エントリを指定できます。コンマ区切りのリスト内のホストにポートが定義されていない場合は、デフォルトのポートとして $DB['PORT'] に定義された値が使用されます。$DB['SERVER'] にソケットパス（例: /var/run/pgbouncer）を指定した場合は、その Unix ドメインソケットが使用されます。

例:

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

データベースのポート。

定義されていない場合は、値 0 が使用されます。

$DB['DATABASE']

設定されたデータベースへの接続に Web インターフェースが使用するデータベース名。

例:

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Zabbix の Web インターフェースがデータベースへの接続に使用するユーザー名。

$DB['PASSWORD']

Zabbix の Web インターフェースがデータベースへの接続に使用するパスワード。

$DB['SCHEMA']

PostgreSQL で使用するデータベーススキーマ名。

$DB['ENCRYPTION']

Zabbix の Web インターフェースとデータベース間の接続に対する暗号化を有効または無効にします。

値: true - 有効、false - 無効。

$DB['KEY_FILE']

データベースの TLS 接続に使用するクライアント秘密鍵ファイルのパス。

例:

```
// MySQL の場合:
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

データベースの TLS 接続に使用するクライアント証明書ファイルのパス。

例:

```
// MySQL の場合:
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

データベースの TLS 接続に使用する CA 証明書ファイルのパス。

例:

```
// MySQL の場合:
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
// PostgreSQL の場合:
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

TLS 証明書内のデータベースサーバーのホスト名の検証を有効または無効にします。

値: `true` - 有効、`false` - 無効。

$DB['CIPHER_LIST']

データベース接続で許可される TLS 暗号スイートのリスト。

例:

```
// MySQL の場合:
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

データベース認証情報の保存に使用する Vault シークレットマネージャーを指定します。

値: `CyberArk` または `HashiCorp`。

例:

```
// CyberArk の場合:
$DB['VAULT'] = 'CyberArk';
// HashiCorp の場合:
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

データベース認証情報の取得に使用する Vault シークレットマネージャーの URL。

例:

```
// CyberArk の場合:
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
// HashiCorp の場合:
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

データベース認証情報の取得に使用する Vault パスまたはクエリのプレフィックス。

例:

```
// HashiCorp の場合:
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Vault secrets manager 内のデータベース認証情報へのパス。

例:

```
// CyberArk の場合:
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
// HashiCorp の場合:
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Vault シークレットマネージャーへのアクセスに使用する認証トークン。

例:

```
// For HashiCorp:
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

HashiCorp Vault の認証に使用する VaultAppSecretID のロール ID。

例:

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

HashiCorp Vault への認証に使用する VaultAppSecretID。

例:

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Vault シークレットマネージャーへの認証に使用するクライアント証明書ファイルのパス。

例:

```
// CyberArk の場合:
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Vault シークレットマネージャーへの認証に使用するクライアント秘密鍵ファイルのパス。

例:

```
// CyberArk の場合:
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Vault シークレットマネージャーから取得した認証情報のキャッシュを有効または無効にします。

値: `true` - 有効、`false` - 無効。

$ZBX_SERVER

Zabbix のホスト名/IP アドレスを上書きする値を指定します。
 $ZBX_SERVER を $ZBX_SERVER_PORT なしで指定した場合、デフォルトポート（10051）が使用されます。

$ZBX_SERVER_PORT

Zabbix サーバーのポートを上書きする値を指定します。
 このパラメータは $ZBX_SERVER とともに使用します。

$ZBX_SERVER_NAME

Zabbix サーバー名。

$IMAGE_FORMAT_DEFAULT

デフォルトの画像形式。

デフォルト: IMAGE_FORMAT_PNG
 値: - IMAGE_FORMAT_PNG - PNG 形式 - IMAGE_FORMAT_JPEG - JPEG 形式 - IMAGE_FORMAT_GIF - GIF 形式

$HISTORY_PROVIDERS[]

Elasticsearch または ClickHouse の履歴ストレージプロバイダーの設定です。

サポートされている設定パラメータ: - `types` - 外部ストレージに保存するデータ型の配列です。- `provider` - 履歴プロバイダーのタイプ: `elasticsearch` または `clickhouse`。- `url` - 履歴プロバイダーの URL です。- `db` - データベース名です（ClickHouse で使用）。- `username` - データベースユーザーです（ClickHouse で使用）。- `password` - データベースパスワードです（ClickHouse で使用）。

例:

```
// ClickHouse の場合:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
```

```
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
// Elasticsearch の場合:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

SAML サインインの設定。

$SSO['SETTINGS'] 配列は、Zabbix に付属する *SAML PHP Toolkit* ライブラリ（提供）で想定されている構造に従う必要があります。

$SSO['SETTINGS'] では、次のオプションのみ指定できます。

- strict
- baseurl
- compress
- contactPerson
- organization
- sp（このリストに記載されているオプションのみ）
    - attributeConsumingService
    - x509certNew
- idp（このリストに記載されているオプションのみ）
    - singleLogoutService（オプションは 1 つのみ）
        * responseUrl
    - certFingerprint
    - certFingerprintAlgorithm
    - x509certMulti
- security（このリストに記載されているオプションのみ）
    - signMetadata
    - wantNameId
    - requestedAuthnContext
    - requestedAuthnContextComparison
    - wantXMLValidation
    - relaxDestinationValidation
    - destinationStrictlyMatches
    - rejectUnsolicitedResponsesWithInResponseTo
    - signatureAlgorithm
    - digestAlgorithm
    - lowercaseUrlencoding

その他すべてのオプションはデータベースから取得され、上書きできません。debug オプションは無視されます。

Zabbix UI がプロキシまたはロードバランサーの背後にある場合は、カスタムの use_proxy_headers オプションを使用できます。

- false（デフォルト）- X-Forwarded-* HTTP ヘッダーを無視します。
- true - X-Forwarded-* HTTP ヘッダーを使用してベース URL を構築します。

使用可能な設定オプションの詳細については、公式のライブラリドキュメントを参照してください。

例:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384',
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

SAML 証明書の保存場所を指定します。

デフォルト: `file`
 値: - `database` - 秘密鍵と証明書をデータベースに保存します。- `file` - 秘密鍵と証明書をファイルシステムに保存します。

例:

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

秘密鍵がファイルシステムに保存されている場合の、SAML サービスプロバイダー（SP）秘密鍵ファイルへのパス。

例:

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

証明書がファイルシステムに保存されている場合の、SAML サービスプロバイダー（SP）証明書ファイルへのパス。

例:

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

証明書がファイルシステムに保存されている場合の、SAML ID プロバイダー証明書ファイルへのパス。

例:

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

バナーのサポートを有効または無効にします。

デフォルト: `true`
 値: `true` - 有効、`false` - 無効。

$ZBX_FEATURE_FLAGS['http_auth_enabled']

ユーザーの HTTP 認証を有効または無効にします。

デフォルト: `true`
 値: `true` - 有効、`false` - 無効。

$ZBX_FEATURE_FLAGS['modules_config_enabled']

モジュールへのアクセスを有効または無効にします。

デフォルト: `true`
 値: `true` - 有効、`false` - 無効。

$ZBX_FEATURE_FLAGS['media_type_denylist']

事前設定された特定のメディアタイプの編集を無効にします。

値: `email`、`script`、`sms`、`webhook`、`push`。

例:

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Zabbix の Web インターフェースと Zabbix サーバー間のTLS 接続を有効にするかどうかを指定します。

デフォルト: `false`
 値: `true` - 有効、`false` - 無効。

$ZBX_SERVER_TLS['CA_FILE']

サーバー証明書の検証に使用する認証局（CA）証明書ファイルへのパス。

例:

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

相互 TLS 認証に使用するクライアント証明書に対応する、クライアント秘密鍵ファイルへのパス。

例:

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

相互 TLS 認証が必要な場合のクライアント証明書ファイルへのパス。

例:

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

サーバー証明書と照合する発行者の識別名（DN）。

例:

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

サーバー証明書と照合するサブジェクトの識別名（DN）。

例:

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

テレメトリデータソースの設定。

ClickHouse でサポートされている設定パラメータ: - `url` - テレメトリデータソースへの接続に使用する HTTP または HTTPS URL を設定します。- `db` - データベース名を設定します。- `username` - データベースへのログインに使用するユーザー名を設定します。- `password` - データベースへのログインに使用するパスワードを設定します。- `ssl_cert_file`、`ssl_key_file`、`ssl_key_password`、`ssl_verify_peer`、`ssl_verify_host`、`ssl_ca_location`、`ssl_ca_file` - データベースへのセキュアな（TLS/SSL）接続を設定するには、これらのオプションを設定します。- `vault_path` - `username` および `password` の値を保持する Vault シークレットへのパスを設定します。`username` および `password` オプションと同時に使用しないでください。

例:

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

概要　Zabbix API を使用すると、Zabbix の設定をプログラムから取得および変更でき、履歴データにもアクセスできます。
これは主に次の用途で広く使用されています。

- Zabbix と連携する新しいアプリケーションを作成する。
- Zabbix をサードパーティ製ソフトウェアに統合する。
- 定常作業を自動化する。

Zabbix API は HTTP ベースの API であり、Web インターフェースの一部として提供されます。
JSON-RPC 2.0 プロトコルを使用しており、これは次の 2 点を意味します。

- API は個別のメソッドの集合で構成される。
- クライアントと API 間のリクエストおよびレスポンスは JSON 形式でエンコードされる。

プロトコルと JSON の詳細については、JSON-RPC 2.0 specification および JSON format homepage を参照してください。

Zabbix の機能を Python アプリケーションに統合する方法の詳細については、Python library for Zabbix を参照してください。

> **Note:**
> Zabbix におけるユーザーアクセスは、設定データと履歴データの両方について、user type、割り当てられたuser role、およびuser groups に依存します。

構成　API は、名目上は個別の API にグループ化された多数のメソッドで構成されています。各メソッドは、1 つの特定のタスクを実行します。たとえば、host.create メソッドは*host* API に属し、新しいホストを作成するために使用されます。歴史的な経緯から、API は「クラス」と呼ばれることもあります。

ほとんどの API には少なくとも 4 つのメソッドがあります。get、create、update、delete で、それぞれデータの取得、作成、更新、削除を行いますが、API によってはまったく異なるメソッドセットを提供する場合もあります。

リクエストの実行　Web インターフェースをセットアップしたら、リモート HTTP リクエストを使用して API を呼び出せます。そのためには、Web インターフェースディレクトリにある api_jsonrpc.php ファイルへ HTTP POST リクエストを送信する必要があります。
 たとえば、Zabbix の Web インターフェースが https://example.com/zabbix にインストールされている場合、apiinfo.version メソッドを呼び出す HTTP リクエストは次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'
```

リクエストには、Content-Type ヘッダーを次のいずれかの値に設定する必要があります: application/json-rpc、application/json、application/jsonrequest。

リクエストオブジェクトには、次のプロパティを含める必要があります。

- jsonrpc - API が使用する JSON-RPC プロトコルのバージョン（Zabbix API は JSON-RPC バージョン 2.0 を実装しています）;
- method - 呼び出す API メソッド;
- params - API メソッドに渡されるパラメーター;
- id - リクエストの任意の識別子（省略した場合、API はこのリクエストを notification として扱います）。

リクエストが正しい場合、API が返すレスポンスは次のようになります。

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

レスポンスオブジェクトには、次のプロパティが含まれます。

- jsonrpc - JSON-RPC プロトコルのバージョン;
- result - メソッドによって返されるデータ;
- id - 対応するリクエストの識別子。

認証　Zabbix のデータにアクセスするには、次のいずれかを使用する必要があります。

- Zabbix Web インターフェースの *Users* > *API tokens* セクションで作成された既存の API トークン、またはToken API を使用して作成された API トークンを使用する。
- user.login メソッドで取得した認証トークンを使用する。

たとえば、標準の *Admin* ユーザーとしてログインして新しい認証トークンを取得したい場合、JSON リクエストは次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

認証情報が正しく指定されていれば、API から返されるレスポンスにはユーザー認証トークンが含まれます。

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

認証方法　”Authorization” ヘッダーによる認証

すべての API リクエストには認証または API トークンが必要です。リクエストの Authorization ヘッダーを使用して認証情報を指定できます:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
認証に問題がある場合は、Authorization ヘッダーの転送を参照してください。

Zabbix API はヘッダーを大文字・小文字を区別せずに受け付けます（例: authorization、Authorization、AUTHORIZATION は同じものとして扱われます）。

Authorization ヘッダーはクロスオリジンリクエスト (CORS) でもサポートされています。

Zabbix クッキー

"zbx_session" クッキーは、(モジュールまたはカスタムウィジェットから) JavaScript を使用して実行される Zabbix UI からの API リクエストを承認するために使用されます。

使用例のワークフロー　以下のセクションでは、いくつかの使用例についてより詳しく説明します。

ホストの取得　これで、有効なユーザー認証トークン（以下の例では変数として表されています）を使用して、Zabbix のデータにアクセスできます。
たとえば、host.get メソッドを使用すると、設定されているすべてのホスト の ID、ホスト名、およびインターフェースを取得できます。

リクエスト:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data @data.json
```

Note:
data.json は JSON クエリを含むファイルです。
ファイルの代わりに、--data 引数でクエリを渡すこともできます。

data.json

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

レスポンスオブジェクトには、ホストに関する要求されたデータが含まれます。

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
```

```
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
パフォーマンス上の理由から、取得したいオブジェクトプロパティを明示的に列挙することを常に推奨します。
これにより、不要なすべてのデータを取得せずに済みます。

新しいアイテムの作成　次に、前の host.get リクエストで取得したデータを使用して、ホスト"Zabbix server" に新しいアイテムを作成します。
これはitem.create メソッドを使用して実行できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

成功したレスポンスには、新しく作成されたアイテムの ID が含まれます。この ID は、以降のリクエストでアイテムを参照するために使用できます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
item.create メソッドは、他の *create methods* と同様に、オブジェクトの配列も受け付け、1 回の API 呼び出しで複数のアイテムを作成できます。

複数のトリガーの作成　したがって、*create methods* が配列を受け入れる場合、複数のトリガー を追加できます。たとえば、次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

成功した応答には、新しく作成されたトリガーの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

アイテムの更新　アイテムを有効にするには、ステータスを 0 に設定します。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

成功したレスポンスには、更新されたアイテムの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
item.update メソッドは、他の 更新メソッドと同様に、オブジェクトの配列も受け取ることができ、1 回の API 呼び出しで複数のアイテムを更新できます。

複数のトリガーの更新　複数のトリガーを有効にするには、ステータスを 0 に設定します。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.update","params":[{"triggerid":"13938","status":0},{"triggeri
```

成功したレスポンスには、更新されたトリガーの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

エラー処理　これまでのところ、試したことはすべて正常に動作していました。
しかし、API に対して誤った呼び出しを行った場合はどうなるでしょうか。
host.create を呼び出して別のホストを作成してみてください。ただし、必須の groups パラメーターを省略します。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

すると、レスポンスにはエラーメッセージが含まれます。

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

エラーが発生した場合、レスポンスオブジェクトには `result` プロパティの代わりに `error` プロパティが含まれ、次のデータが返されます。

- `code` - エラーコード
- `message` - 簡潔なエラー概要
- `data` - より詳細なエラーメッセージ

エラーは、誤った入力値の使用、セッションのタイムアウト、存在しないオブジェクトへのアクセスなど、さまざまなケースで発生します。
アプリケーションは、これらのエラーを適切に処理できる必要があります。

API バージョン　API のバージョン管理を簡単にするため、Zabbix 2.0.4 以降、API のバージョンは Zabbix 本体のバージョンと一致します。`apiinfo.version` メソッドを使用すると、利用している API のバージョンを確認できます。これは、アプリケーションをバージョン固有の機能に対応させる際に役立ちます。

Zabbix は、メジャーバージョン内での機能の後方互換性を保証します。メジャーリリース間で後方互換性のない変更を行う場合、Zabbix は通常、次のリリースで古い機能を非推奨として残し、その次のリリースでのみ削除します。場合によっては、Zabbix がメジャーリリース間で後方互換性を提供せずに機能を削除することもあります。非推奨の機能には依存せず、できるだけ早く新しい代替手段へ移行することが重要です。

> **Note:**
> API changelog で、API に加えられたすべての変更を確認できます。

さらに読む　これで、Zabbix API の利用を始めるのに十分な知識が身につきましたが、ここで終わらせないでください。さらに学習を進めるには、利用可能な API の一覧をご覧になることをお勧めします。

メソッドリファレンス

このセクションでは、Zabbix API が提供する機能の概要を説明し、利用可能なクラスとメソッドの使い方を説明します。

監視　Zabbix API では、監視中に収集した履歴やその他のデータにアクセスすることができます。

ダッシュボード

ダッシュボードを管理し、それに基づいてスケジュールされたレポートを作成します。

Dashboard API | Template dashboard API | Report API

高可用性 (HA) クラスタ

サーバノードとそのステータスの一覧を取得します。

High availability cluster API

ヒストリ

Zabbix 監視プロセスによって収集された過去の値を取得し、表示やさらなる処理を行うことができます。

History API

トレンド

Zabbix サーバーで計算されたトレンドの値を取得し、表示やさらなる処理を行うことができます。

Trend API

イベント

トリガー、ネットワークディスカバリ、その他の Zabbix システムから生成されたイベントを取得し、より柔軟な状況管理やサードパーティツールの統合を行うことができます。

Event API

障害

指定されたパラメータに従って、障害を取得します。

Problem API

マップ

マップを設定して、IT インフラストラクチャの詳細な動的表示を作成します。

Map API

タスク

Zabbix サーバーのタスクマネージャと連動し、タスクの作成とレスポンスの取得を行います。

Task API

APM　Zabbix API を使用すると、アプリケーションパフォーマンス監視（APM）用に収集されたテレメトリデータにアクセスできます。

トレース

トレースと、それを構成するスパンを取得します。

APM trace API | APM span API

メトリクス

メトリクスのデータポイントを取得します。

APM metric API

ログ

ログレコードを取得します。

APM log API

サービス　Zabbix API を使用すると、サービスの監視中に収集されたデータにアクセスできます。

サービスレベルアグリーメント (SLA)

サービスレベル目標（SLO）を定義し、サービスのパフォーマンスに関する詳細なサービスレベル指標（SLI）情報を取得します。

SLA API

サービス

サービスレベル監視用のサービスを管理し、任意のサービスに関する詳細な SLA 情報を取得します。

Service API

データ収集　Zabbix API を使用すると、監視システムの設定を管理できます。

ホストとホストグループ

ホストグループ、ホスト、およびホストインターフェース、ホストマクロ、メンテナンス期間など、それらに関連するすべてを管理します。

ホスト API | ホストグループ API | ユーザーマクロ API | 値マップ API | メンテナンス API

アイテム

監視するアイテムを定義します

Item API

トリガー

システムで発生した障害を通知するようにトリガーを設定します。トリガーの依存関係を管理します。

トリガー API

グラフ

収集したデータを見やすくするために、グラフを編集したり、グラフの item を分けたりすることができます。

Graph API | Graph item API

テンプレートとテンプレートグループ

テンプレートを管理し、ホストまたは他のテンプレートにリンクします。

Template API | Template group API | Value map API

ローレベルディスカバリ

動的なエンティティを監視するために、ローレベルディスカバリルール、item、trigger、グラフのプロトタイプを設定します。

LLD rule API | Item prototype API | Trigger prototype API | Graph prototype API | Host prototype API

イベント相関

カスタムイベントの相関ルールを作成します。

Correlation API

ネットワークディスカバリ

ネットワークレベルの検出ルールを管理し、新しいホストを自動的に検出して監視します。検出されたサービスとホストに関する情報に完全にアクセスできます。

Discovery rule API | Discovery check API | Discovered host API | Discovered service API

エクスポートとインポート

設定のバックアップ、マイグレーション、大規模な設定更新のために Zabbix の設定データをエクスポート、インポートすることができます。

Configuration API

Web 監視

Web アプリケーションやサービスを監視するための Web シナリオを設定します。

Web scenario API

アラート　Zabbix API を使用すると、監視システムのアクションとアラートを管理できます。

アクションとアラート

特定のイベントをユーザーに通知したり、リモートコマンドを自動的に実行するためのアクションと操作を定義します。生成されたアラートとその受信者に関する情報にアクセスできます。

Action API | Alert API

メディアタイプ

メディアの種類と、ユーザーがアラートを受け取る複数の方法を構成します。

Media type API

スクリプト

日々の作業を支援するスクリプトを設定・実行します。

Script API

ユーザー　Zabbix API を使用すると、監視システムのユーザーを管理できます。

ユーザーとユーザーグループ

Zabbix にアクセスするユーザを追加し、ユーザグループに割り当て、権限を付与します。ユーザーの権限を細かく管理するための役割を作成します。

User API | User group API | User directory API | User role API

API トークン

認可トークンを管理します。

Token API

認証

認証の設定オプションを変更します。

Authentication API

デバイス

Zabbix モバイルアプリで使用するモバイルデバイスを管理します。

デバイス API

管理画面　Zabbix API を使用すると、監視システムの管理設定を変更することができます。

一般

特定のグローバル設定オプションを変更します。

Autoregistration API | Icon map API | Image API | Settings API | Regular expression API | Module API | Connector API

監査ログ

各ユーザーが行った設定の変更を追跡します。

Audit log API

ハウスキーピング

ハウスキーピングの設定をします。

Housekeeping API

プロキシとプロキシグループ

分散監視設定で使用される Zabbix プロキシを管理します。

Proxy API | Proxy group API

マクロ

マクロを管理します。

User macro API

Zabbix 情報　Zabbix API のバージョン、または Zabbix サーバーから最終更新時刻を取得します。

API info API | Server info API

Alert

このクラスは、アラートを操作するためのものです。

オブジェクトリファレンス:

- Alert

利用可能なメソッド:

- alert.get - アラートを取得する

Alert オブジェクト

次のオブジェクトは、`alert`API に直接関連しています。

アラート

> **Note:**
> アラートは Zabbix サーバーによって作成され、API 経由で変更することはできません。

アラートオブジェクトには、特定のアクションオペレーションが正常に実行されたかどうかに関する情報が含まれます。次のプロパティがあります。

プロパティ	型	説明
alertid	ID	アラートの ID。
actionid	ID	アラートを生成したアクションの ID。
alerttype	integer	アラートの種類。 使用可能な値: 0 - メッセージ; 1 - リモートコマンド。
clock	timestamp	アラートが生成された時刻。
error	string	メッセージの送信またはコマンドの実行に問題がある場合のエラーテキスト。

プロパティ	型	説明
esc_step	integer	アラートが生成された時点のアクションエスカレーションステップ。
eventid	ID	アクションをトリガーしたイベントの ID。
mediatypeid	ID	メッセージの送信に使用されたメディアタイプの ID。
message	text	メッセージ本文。 プロパティの動作: - `alerttype` が”message” に設定されている場合に *supported*
retries	integer	Zabbix がメッセージ送信を試行した回数。
sendto	string	受信者のアドレス、ユーザー名、またはその他の識別子。 プロパティの動作: - `alerttype` が”message” に設定されている場合に *supported*
status	integer	アクションオペレーションが正常に実行されたかどうかを示すステータス。 `alerttype` が”message” に設定されている場合の使用可能な値: 0 - メッセージ未送信; 1 - メッセージ送信済み; 2 - 複数回の再試行後に失敗; 3 - 新しいアラートはまだアラートマネージャーによって処理されていません。 `alerttype` が”remote command” に設定されている場合の使用可能な値: 0 - コマンド未実行; 1 - コマンド実行済み; 2 - Zabbix エージェント上でコマンドの実行を試みましたが、利用できませんでした。
subject	string	メッセージ件名。 プロパティの動作: - `alerttype` が”message” に設定されている場合に *supported*
userid	ID	メッセージの送信先ユーザーの ID。
p_eventid	ID	アラートを生成した障害イベントの ID。
acknowledgeid	ID	アラートを生成した確認応答の ID。

alert.get

説明

`integer/array alert.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってアラートを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
alertids	ID/array	指定した ID のアラートのみを返します。
actionids	ID/array	指定したアクションによって生成されたアラートのみを返します。
eventids	ID/array	指定したイベントによって生成されたアラートのみを返します。
groupids	ID/array	指定したホストグループ内のオブジェクトによって生成されたアラートのみを返します。

パラメータ	型	説明
hostids	ID/array	指定したホスト上のオブジェクトによって生成されたアラートのみを返します。
mediatypeids	ID/array	指定したメディアタイプを使用したメッセージアラートのみを返します。
objectids	ID/array	指定したオブジェクトによって生成されたアラートのみを返します。
userids	ID/array	指定したユーザーに送信されたメッセージアラートのみを返します。
eventobject	integer	指定したタイプのオブジェクトに関連するイベントによって生成されたアラートのみを返します。 サポートされているオブジェクトタイプの一覧は、event `object` を参照してください。 デフォルト: 0 - トリガー。
eventsource	integer	指定したタイプのイベントによって生成されたアラートのみを返します。 サポートされているイベントタイプの一覧は、event `source` を参照してください。 デフォルト: 0 - トリガーイベント。
time_from	timestamp	指定した時刻より後に生成されたアラートのみを返します。
time_till	timestamp	指定した時刻より前に生成されたアラートのみを返します。
selectHosts	query	アクションの実行操作をトリガーしたホストのデータを含む`hosts` プロパティを返します。
selectMediatypes	query	メッセージアラートに使用されたメディアタイプの配列を含む`mediatypes` プロパティを返します。 ユーザータイプに基づく制限については、`mediatype.get` を参照してください。
selectUsers	query	メッセージの宛先となったユーザーの配列を含む`users` プロパティを返します。 ユーザータイプに基づく制限については、`user.get` を参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`。
countOutput	boolean	これらのパラメータについては、reference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

アクション ID によるアラートの取得

アクション"3" によって生成されたすべてのアラートを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

関連項目

- ホスト
- メディアタイプ
- ユーザー

ソース

CAlert::get() in *ui/include/classes/api/services/CAlert.php*.

API info

このクラスは、API に関するメタ情報を取得するためのものです。

利用可能なメソッド:

- apiinfo.version - Zabbix API のバージョンの取得

apiinfo.version

説明

`string apiinfo.version(array)`

このメソッドを使用すると、Zabbix API のバージョンを取得できます。

Attention:
このメソッドは、認証されていないユーザーのみが利用できます。

パラメーター

(array) このメソッドは空の配列を受け入れます。

戻り値

(string) Zabbix API のバージョンを返します。

Note:
Zabbix 2.0.4 以降、API のバージョンは Zabbix のバージョンと一致します。

例

API のバージョンの取得

Zabbix API のバージョンを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

ソース

CAPIInfo::version() in *ui/include/classes/api/services/CAPIInfo.php*.

APM スパン

このクラスは、アプリケーションパフォーマンス監視（APM）用に収集されたスパンを操作するために設計されています。

オブジェクト参照:

- APM スパン

利用可能なメソッド:

- apmspan.get - スパンを取得する

APM span オブジェクト

以下のオブジェクトは、apmspan API に直接関連しています。

Note:
各プロパティは、テレメトリデータソース の otel_traces テーブル内の列に対応しています。列の一覧については、サポートされているテレメトリデータの列 を参照してください。

APM スパン

APM スパンオブジェクトは、トレース内の 1 つのスパン（HTTP リクエスト、データベースクエリ、関数呼び出しなど、単一の操作）を表します。

スパンは、リクエストの最初の操作を表すルートスパンから始まる階層を形成します。その他のすべてのスパンは、その操作が開始されたスパンの子スパンです。

APM スパンオブジェクトには、次のプロパティがあります。

プロパティ	型	説明
timestamp	integer	スパンが開始された時刻の Unix 時間（ナノ秒単位）。 プロパティの動作: - 読み取り専用
traceid	string	スパンが属するトレースの ID。 プロパティの動作: - 読み取り専用
spanid	string	スパンの ID。 プロパティの動作: - 読み取り専用
parent_spanid	string	親スパンの ID。 ルートスパンの場合は空です。 プロパティの動作: - 読み取り専用
trace_state	string	トレースツールがトレースに付加する追加データ（存在する場合）（W3C Trace Contextを参照）。 プロパティの動作: - 読み取り専用
span_name	string	スパンが表す操作の名前（例: `mysqli_query`）。 プロパティの動作: - 読み取り専用
span_kind	string	スパンの役割。 指定可能な値: `Unspecified` - スパンの種類が指定されていない; `Internal` - アプリケーション内で実行された操作; `Server` - 受信リクエストの処理; `Client` - 送信リクエスト; `Producer` - 非同期操作の作成（メッセージの送信など）; `Consumer` - 非同期操作の処理。 プロパティの動作: - 読み取り専用
service_name	string	スパンを生成したサービスの名前（例: `zabbix-frontend`）。 OpenTelemetry の `service.name` リソース属性から取得されます。 プロパティの動作: - 読み取り専用
resource_attributes	object	スパンのキーと値のリソース属性（例: `host.name: eabf8c2a5ddc`）。 プロパティの動作: - 読み取り専用
scope_name	string	スパンを生成したインストルメンテーションスコープの名前。 プロパティの動作: - 読み取り専用
scope_version	string	インストルメンテーションスコープのバージョン。 プロパティの動作: - 読み取り専用
span_attributes	object	スパンのキーと値のスパン属性（例: `db.system: mysql`）。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
duration	integer	スパンの継続時間（ナノ秒）。 プロパティの動作: - 読み取り専用
status_code	string	スパンが表す操作のステータス。 指定可能な値: `Unset` - スパンに明示的なステータスが割り当てられていない; `Ok` - 操作が正常に完了した; `Error` - 操作がエラーで完了した。 プロパティの動作: - 読み取り専用
status_message	string	スパンのステータスに関する追加情報（存在する場合）。 プロパティの動作: - 読み取り専用
events	array	スパンのeventオブジェクトの配列。操作中の注目すべき時点（例外など）を表します。 プロパティの動作: - 読み取り専用
links	array	スパンのlinkオブジェクトの配列。同じトレースまたは別のトレース内の関連するスパンを指します。 プロパティの動作: - 読み取り専用

apmspan.get

説明

`integer/array apmspan.get(object parameters)`

指定したパラメータに従ってスパンを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドでは、以下のパラメータをサポートしています。

パラメータ	型	説明
time_from	timestamp	指定した時刻以降に開始したスパンのみを返します。 パラメータの動作: - 必須
time_till	timestamp	指定した時刻より前に開始したスパンのみを返します。 パラメータの動作: - 必須
traceids	string/array	指定したトレースに属するスパンのみを返します。
spanids	string/array	指定した ID を持つスパンのみを返します。
parent_spanids	string/array	指定した親スパンを持つスパンのみを返します。

パラメータ	型	説明
resource_attributes	object/array	指定したリソース属性を持つスパンのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのスパンを返します。 使用可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
resource_attributes_evaltype	integer	`resource_attributes` の条件に対する評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
span_attributes	object/array	指定したスパン属性を持つスパンのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのスパンを返します。 使用可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
span_attributes_evaltype	integer	`span_attributes` の条件に対する評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
min_duration	integer	指定した値以上の期間を持つスパンのみを返します。値の単位はナノ秒です。
max_duration	integer	指定した値未満の期間を持つスパンのみを返します。値の単位はナノ秒です。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、一致させる単一の値または値の配列を値とするオブジェクトを受け取ります。 サポートされるプロパティ: `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`。
search	object	指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 プロパティ名をキーとし、検索対象の文字列を値とするオブジェクトを受け取ります。追加のオプションを指定しない場合、`LIKE "%...%"` 検索を実行します。 サポートされるプロパティ: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 使用可能な値: `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`。

パラメータ	型	説明
offset	integer	結果を返す前にスキップするレコード数。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明しています。
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

トレースのスパン階層の取得

トレース「dee625f33cb0994e86c5cd4640e3b09c」の最初の 4 つのスパンを、開始時刻順に取得します。
parent_spanid プロパティは、スパン間の関係を示します。ルートスパンには親がなく、それ以外の各スパンは、そのスパンを開始したスパンを参照します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
```

```
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```


関連項目

- APM トレース
- APM ログ
- APM メトリクス

ソース

ui/include/classes/api/services/CApmSpan.php の CApmSpan::get()。

APM トレース

このクラスは、アプリケーションパフォーマンス監視（APM）用に収集されたトレースを操作するために設計されています。

オブジェクト参照:

- APM トレース
- イベント
- リンク

利用可能なメソッド:

- apmtrace.get - トレースを取得する

APM トレースオブジェクト

以下のオブジェクトは、apmtrace API に直接関連しています。

> **Note:**
> span_count と error_count を除く各プロパティは、テレメトリデータソースの otel_traces テーブル内の列に対応しています。ほとんどのプロパティは、同じ名前の列に対応しています（例: span_name プロパティは SpanName 列に対応します）。名前が異なる場合は、以下の表にデータソースの列を記載しています。列の一覧については、サポートされているテレメトリデータ列を参照してください。

APM トレース

APM トレースオブジェクトは、トレース（アプリケーションおよびサービスを通過するリクエストの経路）を表します。トレースは 1 つ以上のスパンで構成されます。各スパンは単一の操作を表します。

APM トレースオブジェクトは、トレースのルートスパン（トレースを開始し、親スパンを持たないスパン）に基づきます。span_count と error_count を除くすべてのプロパティには、ルートスパンのデータが含まれます。span_count と error_count は、トレースのすべてのスパンから計算されます。

APM トレースオブジェクトには、次のプロパティがあります。

Property	Type	Description
timestamp	integer	ルートスパンが開始された時刻の Unix 時間（ナノ秒単位）。 Property behavior: - *read-only*
traceid	string	トレースに属するすべてのスパンで共有される、トレースの ID。 Property behavior: - *read-only*
spanid	string	ルートスパンの ID。 Property behavior: - *read-only*
trace_state	string	トレースツールがトレースに付加する追加データ（存在する場合）（W3C Trace Contextを参照）。 Property behavior: - *read-only*
span_name	string	ルートスパンが表す操作の名前（例: `POST /zabbix.php`）。 Property behavior: - *read-only*
span_kind	string	ルートスパンの役割。 指定可能な値: `Unspecified` - スパンの種類が指定されていない; `Internal` - アプリケーション内で実行される操作; `Server` - 受信リクエストの処理; `Client` - 送信リクエスト; `Producer` - 非同期操作の作成（例: メッセージの送信）; `Consumer` - 非同期操作の処理。 Property behavior: - *read-only*
service_name	string	ルートスパンを生成したサービスの名前（例: `zabbix-nginx`）。 OpenTelemetry の `service.name` リソース属性から取得されます。 Property behavior: - *read-only*
resource_attributes	object	ルートスパンのキーと値のリソース属性（例: `host.name: eabf8c2a5ddc`）。 Property behavior: - *read-only*
scope_name	string	ルートスパンを生成したインストルメンテーションスコープの名前。 Property behavior: - *read-only*
scope_version	string	インストルメンテーションスコープのバージョン。 Property behavior: - *read-only*
span_attributes	object	ルートスパンのキーと値のスパン属性（例: `http.method: POST`）。 Property behavior: - *read-only*
duration	integer	トレースの継続時間（ナノ秒）。 Property behavior: - *read-only*

Property	Type	Description
status_code	string	ルートスパンが表す操作のステータス。 指定可能な値: `Unset` - ルートスパンに明示的なステータスが割り当てられていない; `Ok` - 操作が正常に完了した; `Error` - 操作がエラーで完了した。 Property behavior: - *read-only*
status_message	string	ルートスパンのステータスに関する追加情報（存在する場合）。 Property behavior: - *read-only*
events	array	ルートスパンのeventオブジェクトの配列。つまり、操作中の注目すべき時点（例: 例外）です。 Property behavior: - *read-only*
links	array	ルートスパンのlinkオブジェクトの配列。同じトレースまたは別のトレースにある関連スパンを指します。 Property behavior: - *read-only*
span_count	integer	トレース内のスパンの総数。 Property behavior: - *read-only*
error_count	integer	`status_code` が `Error` に設定されている、トレース内のスパン数。 Property behavior: - *read-only*

イベント

イベントオブジェクトには、次のプロパティがあります。

Property	Type	Description
timestamp	integer	イベントが発生した時刻をナノ秒単位で表す Unix 時間。 データソースの列: `Events.Timestamp`
name	string	イベントの名前。 データソースの列: `Events.Name`
attributes	object	イベントに関する追加情報がある場合に、その情報を格納するキーと値の属性。 データソースの列: `Events.Attributes`

リンク

リンクオブジェクトには、次のプロパティがあります。

Property	Type	Description
traceid	string	リンクされたスパンを含むトレースの ID。 データソースの列: `Links.TraceId`
spanid	string	リンクされたスパンの ID。 `traceid` と組み合わせることで、リンクされたスパンを一意に識別します。 データソースの列: `Links.SpanId`
trace_state	string	リンクされたスパンのトレースステート（存在する場合）（W3C Trace Contextを参照）。 データソースの列: `Links.TraceState`

Property	Type	Description
attributes	object	リンクに関する追加情報を含むキーと値の属性。 データソースの列: `Links.Attributes`

apmtrace.get

説明

`integer/array apmtrace.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってトレースを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは、以下のパラメータをサポートしています。

パラメータ	タイプ	説明
time_from	timestamp	指定した時刻以降にルートスパンが開始されたトレースのみを返します。 パラメータの動作: - 必須
time_till	timestamp	指定した時刻より前にルートスパンが開始されたトレースのみを返します。 パラメータの動作: - 必須
traceids	string/array	指定した ID を持つトレースのみを返します。
resource_attributes	object/array	指定したリソース属性をルートスパンが持つトレースのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのトレースを返します。 指定可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
resource_attributes_evaltype	integer	`resource_attributes` の条件に対する評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
span_attributes	object/array	指定したスパン属性をルートスパンが持つトレースのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのトレースを返します。 指定可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。

パラメータ	タイプ	説明
span_attributes_evaltype	integer	span_attributes の条件に対する評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
min_duration	integer	ルートスパンの duration が指定した値以上であるトレースのみを、ナノ秒単位で返します。
max_duration	integer	ルートスパンの duration が指定した値未満であるトレースのみを、ナノ秒単位で返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、一致させる単一の値または値の配列を値とするオブジェクトを受け取ります。 サポートされるプロパティ: traceid, spanid, trace_state, span_name, span_kind, service_name, scope_name, scope_version, status_code, status_message。
search	object	指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 プロパティ名をキーとし、検索対象の文字列を値とするオブジェクトを受け取ります。追加のオプションを指定しない場合、LIKE "%...%" 検索を実行します。 サポートされるプロパティ: trace_state, span_name, service_name, scope_name, scope_version, status_message。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: timestamp, traceid, spanid, trace_state, span_name, span_kind, service_name, scope_name, scope_version, duration, status_code, status_message。
offset	integer	結果を返す前にスキップするレコード数。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明しています。
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

遅いトレースの取得

zabbix-nginx サービスの中で最も遅い 2 つのトレースを取得します。
2026 年 10 月 1 日（UTC）の 10:00 から 11:00 までのトレースを検索し、所要時間が 1 秒以上のものだけを含めます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
```

```
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
        },
        {
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```

関連項目

- APM スパン
- APM メトリクス
- APM ログ

ソース

ui/include/classes/api/services/CApmTrace.php の CApmTrace::get()。

APM メトリック

このクラスは、アプリケーションパフォーマンス監視（APM）用に収集されたメトリックを扱うために設計されています。

オブジェクト参照:

- APM メトリック
- Exemplar

利用可能なメソッド:

- apmmetric.get - メトリックのデータポイントを取得します

APM メトリクスオブジェクト

以下のオブジェクトは、apmmetric API に直接関連しています。

Note:
type を除く各プロパティは、テレメトリデータソースのメトリクステーブル（otel_metrics_gauge、otel_metrics_sum、otel_metrics_histogram、otel_metrics_exponential_histogram）の列に対応しています。ほとんどのプロパティは、同じ名前の列に対応しています（例：metric_name プロパティは MetricName 列に対応します）。名前が異なる場合は、以下の表にデータソースの列名を記載しています。列の一覧については、サポートされているテレメトリデータ列を参照してください。

APM メトリック

APM メトリックオブジェクトは、メトリックのデータポイント（特定の時点における測定値または集計値）を表します。

APM メトリックオブジェクトには、次のプロパティがあります。

プロパティ	型	説明
type	integer	データポイントのタイプ。 指定可能な値: 0 - Gauge; 1 - Sum; 2 - Histogram; 3 - Exponential histogram。 タイプは、データポイントを保持するテーブル（otel_metrics_gauge、otel_metrics_sum、otel_metrics_histogram、または otel_metrics_exponential_histogram）によって決まります。 プロパティの動作: - 読み取り専用
resource_attributes	object	データポイントのキーと値のリソース属性（例: host.name: eabf8c2a5ddc）。 プロパティの動作: - 読み取り専用
resource_schema_url	string	リソース属性の OpenTelemetry スキーマの URL（存在する場合）。 プロパティの動作: - 読み取り専用
scope_name	string	データポイントを生成したインストルメンテーションスコープの名前。 プロパティの動作: - 読み取り専用
scope_version	string	インストルメンテーションスコープのバージョン（存在する場合）。 プロパティの動作: - 読み取り専用
scope_attributes	object	インストルメンテーションスコープのキーと値の属性。 プロパティの動作: - 読み取り専用
scope_schema_url	string	インストルメンテーションスコープとその属性の OpenTelemetry スキーマの URL（存在する場合）。 プロパティの動作: - 読み取り専用
service_name	string	データポイントを生成したサービスの名前（例: zabbix-frontend）。OpenTelemetry の service.name リソース属性から取得されます。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
metric_name	string	データポイントが属するメトリックの名前（例: `http.server.request.duration`）。 プロパティの動作: - 読み取り専用
metric_description	string	メトリックの説明（存在する場合）。 プロパティの動作: - 読み取り専用
metric_unit	string	メトリック値の単位（存在する場合）（例: `s` または `By`。UCUMを参照）。 プロパティの動作: - 読み取り専用
attributes	object	キーと値のデータポイント属性（例: `http.request.method: POST`）。 プロパティの動作: - 読み取り専用
start_time_unix	integer	データポイントの測定期間が開始した時点の Unix 時間（ナノ秒）。 プロパティの動作: - 読み取り専用
time_unix	integer	データポイントの Unix 時間（ナノ秒）。 *Sum*、*Histogram*、および *Exponential histogram* のデータポイントでは、測定期間の終了時点を示します。 プロパティの動作: - 読み取り専用
value	float	データポイントの値。 *Gauge* および *Sum* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
flags	integer	データポイントフラグのビットマスク。 最下位ビットが設定されている場合（値が `1` の場合）、データポイントの値は記録されていません。 プロパティの動作: - 読み取り専用
exemplars	array	exemplarオブジェクトの配列。exemplar は、記録時のトレースとスパンを含む、例として保持された個々の測定値です。 プロパティの動作: - 読み取り専用
aggregation_temporality	integer	データポイントの値を時間の経過に伴って集計する方法。 *Sum*、*Histogram*、および *Exponential histogram* のデータポイントに適用されます。 指定可能な値: 0 - *(デフォルト)* 未指定; 1 - Delta（前のデータポイント以降の期間で集計）; 2 - Cumulative（`start_time_unix` 以降の期間で集計）。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
is_monotonic	boolean	時間の経過に伴い、合計値が増加するか同じ値にとどまることのみが可能かどうか。 *Sum* のデータポイントに適用されます。 デフォルト: `false`。 プロパティの動作: - 読み取り専用
count	integer	データポイント内の測定値の数。 *Histogram* および *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
sum	float	データポイント内のすべての測定値の合計。 *Histogram* および *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
bucket_counts	array	各ヒストグラムバケット内の測定値の数を示す整数の配列。 バケットは `explicit_bounds` によって定義されます。 *Histogram* のデータポイントに適用されます。 デフォルト: `[]`。 プロパティの動作: - 読み取り専用
explicit_bounds	array	ヒストグラムバケット間の境界を示す浮動小数点数の配列。 *Histogram* のデータポイントに適用されます。 デフォルト: `[]`。 プロパティの動作: - 読み取り専用
min	float	データポイント内の測定値の最小値。 *Histogram* および *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
max	float	データポイント内の測定値の最大値。 *Histogram* および *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
scale	integer	指数ヒストグラムのスケール。バケットの境界と解像度を決定します。スケールが大きいほどバケットは狭くなり、解像度は高くなります。スケールが小さいほどバケットは広くなり、解像度は低くなります。 *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
zero_count	integer	指数ヒストグラムのゼロバケット内の測定値の数。 *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
positive_bucket_counts	array	各指数ヒストグラムバケット内の正の測定値の数を示す整数の配列。バケットの境界は `scale` によって決まり、最初のバケットのインデックスは `positive_offset` に設定されます。 *Exponential histogram* のデータポイントに適用されます。 デフォルト: []。 プロパティの動作: - 読み取り専用
positive_offset	integer	`positive_bucket_counts` 内の最初のバケットのインデックス。 *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用
negative_bucket_counts	array	各指数ヒストグラムバケット内の負の測定値の数を示す整数の配列。バケットの境界は `scale` によって決まり、最初のバケットのインデックスは `negative_offset` に設定されます。 *Exponential histogram* のデータポイントに適用されます。 デフォルト: []。 プロパティの動作: - 読み取り専用
negative_offset	integer	`negative_bucket_counts` 内の最初のバケットのインデックス。 *Exponential histogram* のデータポイントに適用されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用

Exemplar

exemplar オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
traceid	string	測定値が記録されたトレースの ID（存在する場合）。 データソースの列: `Exemplars.TraceId`。
spanid	string	測定値が記録されたスパンの ID（存在する場合）。 データソースの列: `Exemplars.SpanId`。
value	float	測定値。 データソースの列: `Exemplars.Value`。

プロパティ	型	説明
time_unix	integer	測定値が記録された時刻の Unix 時間（ナノ秒単位）。 データソースの列: `Exemplars.TimeUnix`。
filtered_attributes	object	データポイントの `attributes` に含まれていない、測定値のキーと値の属性。 データソースの列: `Exemplars.FilteredAttributes`。

apmmetric.get

説明

`integer/array apmmetric.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってメトリクスのデータポイントを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドでは、以下のパラメータをサポートしています。

パラメータ	タイプ	説明
time_from	timestamp	指定した時刻以降の時刻を持つデータポイントのみを返します。 パラメータの動作: - 必須
time_till	timestamp	指定した時刻より前の時刻を持つデータポイントのみを返します。 パラメータの動作: - 必須
types	integer/array	指定したタイプのデータポイントのみを返します。 サポートされているメトリクスタイプの一覧については、APM メトリクスオブジェクトのページを参照してください。
with_flags_on	integer	指定したフラグがすべて設定されているデータポイントのみを返します。 複数のフラグを指定するには、それらの値の合計を使用します。
with_flags_off	integer	指定したフラグがいずれも設定されていないデータポイントのみを返します。 複数のフラグを指定するには、それらの値の合計を使用します。
resource_attributes	object/array	指定したリソース属性を持つデータポイントのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのデータポイントを返します。 指定可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
resource_attributes_evaltype	integer	`resource_attributes` の条件に対する評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。

パラメータ	タイプ	説明
scope_attributes	object/array	指定したスコープ属性を持つデータポイントのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのデータポイントを返します。 指定可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
scope_attributes_evaltype	integer	`scope_attributes` の条件に対する評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
attributes	object/array	指定した属性を持つデータポイントのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのデータポイントを返します。 指定可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
attributes_evaltype	integer	`attributes` の条件に対する評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、一致させる単一の値または値の配列を値とするオブジェクトを受け取ります。 サポートされているプロパティ: `resource_schema_url`、`scope_schema_url`、`scope_name`、`scope_version`、`service_name`、`metric_name`、`metric_unit`、`flags`。
search	object	指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 プロパティ名をキーとし、検索対象の文字列を値とするオブジェクトを受け取ります。追加のオプションを指定しない場合、`LIKE "%...%"` 検索を実行します。 サポートされているプロパティ: `resource_schema_url`、`scope_schema_url`、`scope_name`、`scope_version`、`service_name`、`metric_name`、`metric_description`、`metric_unit`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `type`、`resource_schema_url`、`scope_schema_url`、`scope_name`、`scope_version`、`service_name`、`metric_name`、`metric_unit`、`start_time_unix`、`time_unix`。
offset	integer	結果を返す前にスキップするレコード数。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明しています。

パラメータ	タイプ	説明
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ヒストグラムデータポイントの取得

zabbix-frontend サービスの http.server.request.duration ヒストグラムメトリクスの最新のデータポイントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
                    "filtered_attributes": {}
```

```
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- APM トレース
- APM スパン
- APM ログ

ソース

ui/include/classes/api/services/CApmMetric.php の CApmMetric::get()。

APM ログ

このクラスは、アプリケーションパフォーマンス監視（APM）用に収集されたログを操作するために設計されています。

オブジェクトリファレンス:

- APM ログ

利用可能なメソッド:

- apmlog.get - ログレコードを取得する

APM ログオブジェクト

以下のオブジェクトは、apmlog API に直接関連しています。

Note:
各プロパティは、テレメトリデータソースの otel_logs テーブルにある同名のカラムに対応します（例：severity_text プロパティは SeverityText カラムに対応します）。カラムの一覧については、サポートされているテレメトリデータカラムを参照してください。

APM ログ

APM ログオブジェクトは、ログレコード（支払いの失敗やデータベースのタイムアウトなど、イベントをテキストで記録したもの）を表します。ログレコードには、重大度、メッセージ、属性など、特定の時刻に発生したイベントに関する情報が含まれます。ログレコードがトレースの一部である操作中に生成された場合、そのトレースおよびスパンの ID も含まれます。

APM ログオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
timestamp	integer	イベントが発生した時刻を表す Unix 時間（ナノ秒単位）。 プロパティの動作: - 読み取り専用
traceid	string	ログレコードが生成されたトレースの ID（存在する場合）。 プロパティの動作: - 読み取り専用
spanid	string	ログレコードが生成されたスパンの ID（存在する場合）。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
trace_flags	integer	ビットマスクとして表される、ログレコードのトレースコンテキストのトレースフラグ。 最下位ビットが設定されている場合（値が 1）、トレースはサンプリングされています。 プロパティの動作: - 読み取り専用
severity_text	string	アプリケーションによって提供されたログレコードの重大度レベル（例: `INFO`、`WARN`、`ERROR`）。 プロパティの動作: - 読み取り専用
severity_number	integer	数値として表されるログレコードの重大度レベル。 指定可能な値: 0 - UNSPECIFIED; 1-4 - TRACE; 5-8 - DEBUG; 9-12 - INFO; 13-16 - WARN; 17-20 - ERROR; 21-24 - FATAL。 プロパティの動作: - 読み取り専用
service_name	string	ログレコードを生成したサービスの名前（例: `zabbix-frontend`）。 OpenTelemetry の `service.name` リソース属性から取得されます。 プロパティの動作: - 読み取り専用
body	string	ログレコードのメッセージ。 プロパティの動作: - 読み取り専用
resource_schema_url	string	リソース属性の OpenTelemetry スキーマの URL（存在する場合）。 プロパティの動作: - 読み取り専用
resource_attributes	object	ログレコードのキーと値のリソース属性（例: `host.name: eabf8c2a5ddc`）。 プロパティの動作: - 読み取り専用
scope_schema_url	string	インストルメンテーションスコープとその属性の OpenTelemetry スキーマの URL（存在する場合）。 プロパティの動作: - 読み取り専用
scope_name	string	ログレコードを生成したインストルメンテーションスコープの名前。 プロパティの動作: - 読み取り専用
scope_version	string	インストルメンテーションスコープのバージョン（存在する場合）。 プロパティの動作: - 読み取り専用
scope_attributes	object	インストルメンテーションスコープのキーと値の属性。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
log_attributes	object	ログレコードのキーと値のログ属性（例: `log.file.path: /var/log/zabbix/zabbix_frontend.log`）。 プロパティの動作: - 読み取り専用
event_name	string	ログレコードが表すイベントタイプの名前（存在する場合）。 プロパティの動作: - 読み取り専用

apmlog.get

説明

`integer/array apmlog.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってログレコードを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドでは、次のパラメータをサポートしています。

パラメータ	タイプ	説明
time_from	timestamp	指定した時刻以降に発生したイベントのログレコードのみを返します。 パラメータの動作: - 必須
time_till	timestamp	指定した時刻より前に発生したイベントのログレコードのみを返します。 パラメータの動作: - 必須
traceids	string/array	指定したトレース中に生成されたログレコードのみを返します。
spanids	string/array	指定したスパン中に生成されたログレコードのみを返します。
with_flags_on	integer	指定したトレースフラグがすべて設定されているログレコードのみを返します。 複数のフラグを指定するには、それらの値の合計を使用します。
with_flags_off	integer	指定したトレースフラグがいずれも設定されていないログレコードのみを返します。 複数のフラグを指定するには、それらの値の合計を使用します。
resource_attributes	object/array	指定したリソース属性を持つログレコードのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのログレコードを返します。 使用可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。

パラメータ	タイプ	説明
resource_attributes_evaltype	integer	`resource_attributes` の条件に対する評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
scope_attributes	object/array	指定したスコープ属性を持つログレコードのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのログレコードを返します。 使用可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
scope_attributes_evaltype	integer	`scope_attributes` の条件に対する評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
log_attributes	object/array	指定したログ属性を持つログレコードのみを返します。 形式: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのログレコードを返します。 使用可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
log_attributes_evaltype	integer	`log_attributes` の条件に対する評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、一致させる単一の値または値の配列を値とするオブジェクトを受け取ります。 サポートされるプロパティ: `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`。
search	object	指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 プロパティ名をキーとし、検索対象の文字列を値とするオブジェクトを受け取ります。追加オプションを指定しない場合、`LIKE "%…%"` 検索を実行します。 サポートされるプロパティ: `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`。

パラメータ	タイプ	説明
sortfield	string/array	指定したプロパティで結果を並べ替えます。 使用可能な値: `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`。
offset	integer	結果を返す前にスキップするレコード数。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明しています。
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

トレースのエラーログの取得

トレース「`8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4`」に関連付けられた、ERROR 重大度（重大度番号 17～20）のログレコードを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
```

```
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```

ログメッセージの検索

メッセージに「timeout」を含む最新 10 件のログレコードについて、時刻、サービス、メッセージを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```

関連項目

- APM トレース
- APM スパン
- APM メトリクス

ソース

ui/include/classes/api/services/CApmLog.php の CApmLog::get()。

Authentication

このクラスは、認証設定を操作するためのものです。

オブジェクトリファレンス:

- Authentication

利用可能なメソッド:

- authentication.get - 認証を取得する
- authentication.update - 認証を更新する

Authentication オブジェクト

次のオブジェクトは、`authenticationAPI` に直接関連しています。

認証

認証オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
authentication_type	integer	デフォルトの認証方式。 指定可能な値: 0 - *(デフォルト)* 内部; 1 - LDAP。
http_auth_enabled	integer	HTTP 認証。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。 プロパティの動作: - Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['http_auth_enabled']` が `true` に設定されている場合にサポートされます。
http_login_form	integer	デフォルトのログインフォーム。 指定可能な値: 0 - *(デフォルト)* Zabbix ログインフォーム; 1 - HTTP ログインフォーム。 プロパティの動作: - Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['http_auth_enabled']` が `true` に設定されている場合にサポートされます。
http_strip_domains	string	削除するドメイン名。 プロパティの動作: - Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['http_auth_enabled']` が `true` に設定されている場合にサポートされます。

プロパティ	タイプ	説明
http_case_sensitive	integer	HTTP の大文字と小文字を区別するログイン。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。 プロパティの動作: - Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['http_auth_enabled']` が `true` に設定されている場合にサポートされます。
ldap_auth_enabled	integer	LDAP 認証。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
ldap_case_sensitive	integer	LDAP の大文字と小文字を区別するログイン。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
ldap_userdirectoryid	ID	LDAP 認証に使用するデフォルトのユーザーディレクトリの ID。 `gui_access` が LDAP またはシステムデフォルトに設定されているユーザーグループに使用されます。 プロパティの動作: - `ldap_auth_enabled` が「有効」に設定されている場合に必須
saml_auth_enabled	integer	SAML 認証。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
saml_case_sensitive	integer	SAML の大文字と小文字を区別するログイン。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
passwd_min_length	integer	パスワードの最小長要件。 指定可能な値の範囲は 1～70 です。 デフォルト: 8。
passwd_check_rules	integer	パスワードのチェックルール。 指定可能なビットマップ値: 0 - パスワードの長さをチェック; 1 - パスワードにラテン文字の大文字と小文字が使用されているかチェック; 2 - パスワードに数字が使用されているかチェック; 4 - パスワードに特殊文字が使用されているかチェック; 8 - *(デフォルト)* パスワードが一般的に使用されるパスワードのリストに含まれておらず、「Zabbix」またはユーザーの名、姓、ユーザー名の派生語が含まれていないかチェック。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、すべてのルールをチェックする場合は 15）。
ldap_jit_status	integer	LDAP プロビジョニングのステータス。 指定可能な値: 0 - 設定済みの LDAP IdP で無効; 1 - 設定済みの LDAP IdP で有効。

プロパティ	タイプ	説明
saml_jit_status	integer	SAML プロビジョニングのステータス。 指定可能な値: 0 - 設定済みの SAML IdP で無効; 1 - 設定済みの SAML IdP で有効。
jit_provision_interval	string	ログインユーザーに対する JIT プロビジョニングリクエスト間の時間間隔。 秒数（例: 3600）、または時間のサフィックスを付けた値（例: 3600s、60m、1h、1d、1M、1y）を指定できます。月および年にも対応しています。最小値: 1h。 デフォルト: 1h。
disabled_usrgrpid	ID	LDAP プロビジョニングでのみ使用できます。 プロビジョニング解除されたユーザーを割り当てるユーザーグループの ID。 このユーザーグループは無効化されている必要があり、設定後に有効化または削除することはできません。 プロパティの動作: - `ldap_jit_status` が「設定済みの LDAP IdP で有効」に設定されている場合、または `saml_jit_status` が「設定済みの SAML IdP で有効」に設定されている場合に必須
mfa_status	integer	多要素認証。 指定可能な値: 0 - 無効（設定済みのすべての MFA 方式）; 1 - 有効（設定済みのすべての MFA 方式）。
mfaid	ID	MFA が有効なユーザーグループに対するデフォルトのMFA 方式。 プロパティの動作: - `mfa_status` が「有効」に設定されている場合に必須

authentication.get

説明

`object authentication.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って認証オブジェクトを取得できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータ。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	このパラメータはリファレンスコメントで説明されています。

戻り値

`(object)` 認証オブジェクトを返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "authentication_type": "0",
        "http_auth_enabled": "0",
        "http_login_form": "0",
        "http_strip_domains": "",
        "http_case_sensitive": "1",
        "ldap_auth_enabled": "0",
        "ldap_case_sensitive": "1",
        "ldap_userdirectoryid": "0",
        "saml_auth_enabled": "0",
        "saml_case_sensitive": "0",
        "passwd_min_length": "8",
        "passwd_check_rules": "15",
        "jit_provision_interval": "1h",
        "saml_jit_status": "0",
        "ldap_jit_status": "0",
        "disabled_usrgrpid": "9",
        "mfa_status": "0",
        "mfaid": "0"
    },
    "id": 1
}
```

ソース

CAuthentication::get() in *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

説明

`object authentication.update(object authentication)`

このメソッドを使用すると、既存の認証設定を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(object) 更新する認証のプロパティ。

戻り値

(array) 更新したパラメーターの名前を配列で返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
```

```
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

ソース

CAuthentication::update() in *ui/include/classes/api/services/CAuthentication.php*.

Autoregistration

このクラスは、自動登録を操作するためのものです。

オブジェクトリファレンス:

- Autoregistration

利用可能なメソッド:

- autoregistration.get - 自動登録を取得する
- autoregistration.update - 自動登録を更新する

Autoregistration オブジェクト

次のオブジェクトは、autoregistrationAPI に直接関連しています。

自動登録

自動登録オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tls_accept	integer	自動登録で許可される受信接続の種類。 指定可能な値: 1 - 暗号化されていない接続を許可; 2 - PSK を使用した TLS を許可; 3 - 暗号化されていない接続と PSK を使用した TLS 接続の両方を許可。
tls_psk_identity	string	PSK ID。1 つの PSK に対して関連付けられる PSK ID は 1 つのみである必要があります（autoregistration、hosts、およびproxies 全体で一意）。 PSK ID は、受信側にどの PSK を使用するかを通知するためにネットワーク上を暗号化されずに送信されるため、機密情報を含めないでください。 プロパティの動作: - 書き込み専用
tls_psk	string	事前共有鍵（PSK）。32 桁以上の 16 進数である必要があります。 プロパティの動作: - 書き込み専用

autoregistration.get

説明

`object autoregistration.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って自動登録オブジェクトを取得できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	型	説明
output	query	このパラメータについては、リファレンスコメントで説明されています。

戻り値

(object) autoregistration オブジェクトを返します。

例

リクエスト:

```
{
   "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tls_accept": "3"
    },
    "id": 1
}
```

ソース

CAutoregistration::get() in *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

説明

`object autoregistration.update(object autoregistration)`

このメソッドを使用すると、既存の自動登録を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(object) 更新する自動登録のプロパティ。

戻り値

(boolean) 更新が成功すると、ブール値 true を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

ソース

CAutoregistration::update() in *ui/include/classes/api/services/CAutoregistration.php*.

Configuration

このクラスは、Zabbix の設定データのエクスポートとインポートを行うためのものです。

利用可能なメソッド:

- configuration.export - 設定のエクスポート
- configuration.import - 設定のインポート
- configuration.importcompare - インポートファイルと現在のシステムの要素の比較

configuration.export

説明

`string configuration.export(object parameters)`

このメソッドを使用すると、設定データをシリアライズされた文字列としてエクスポートできます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) エクスポートするオブジェクトと使用する形式を定義するパラメータ。

Parameter	Type	Description
format	string	データをエクスポートする形式。 指定可能な値: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - 未処理の PHP 配列。 Parameter behavior: - *required*
prettyprint	boolean	インデントを追加して、出力をより人間が読みやすい形式にします。 指定可能な値: `true` - インデントを追加する; `false` - *(default)* インデントを追加しない。
options	object	エクスポートするオブジェクト。 `options` オブジェクトには、次のパラメータがあります: `host_groups` - エクスポートするホストグループの ID の (`array`); `hosts` - エクスポートするホストの ID の (`array`); `images` - エクスポートする画像の ID の (`array`); `maps` - エクスポートするマップの ID の (`array`); `mediaTypes` - エクスポートするメディアタイプの ID の (`array`); `template_groups` - エクスポートするテンプレートグループの ID の (`array`); `templates` - エクスポートするテンプレートの ID の (`array`); `dashboards` - エクスポートするダッシュボードの ID の (`array`); `global_regexes` - エクスポートするグローバル正規表現の ID の (`array`)。 *Admin* および *User* タイプのユーザーは、*read-only* または *read-write* のpermission を持つオブジェクトと画像のみをエクスポートできます。メディアタイプはエクスポートできません。 Parameter behavior: - *required*.

戻り値

(`string`) 要求された構成データを含むシリアル化された文字列を返します。

例

テンプレートのエクスポート

テンプレート”10571” の設定を XML 文字列としてエクスポートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```


ソース

CConfiguration::export() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

説明

`boolean configuration.import(object parameters)`

このメソッドを使用すると、シリアライズされた文字列から設定データをインポートできます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`object`) インポートするデータと、そのデータをどのように処理するかのルールを含むパラメーターです。

Parameter	Type	Description
format	string	シリアライズされた文字列の形式です。 指定可能な値: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *required*
source	string	設定データを含むシリアライズされた文字列です。 Parameter behavior: - *required*
rules	object	新規オブジェクトおよび既存オブジェクトをどのようにインポートするかのルールです。 *Admin* タイプのユーザーは、*read-write* permission を持つオブジェクトとマップのみをインポートできます。たとえば、ホストとそのエンティティー（アイテム、トリガー、グラフなど）は、インポートされるホストが属するホストグループに対してユーザーのユーザーグループが権限を持っている場合にのみインポートできます。画像とメディアタイプは、*Admin* タイプのユーザーはインポートできません。 `rules` パラメーターの詳細は、以下の表で説明します。 Parameter behavior: - *required*
returnMissingObjects	boolean	レスポンスオブジェクトに不足しているオブジェクトの情報を含めるかどうかのルールです。デフォルト: `false`。

> **Attention:**
> ルールが指定されていない場合、設定は更新されません。

`rules` オブジェクトは、次のパラメーターをサポートします。

Parameter	Type	Description
dashboards	object	ダッシュボードをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、システムに存在する特定のインポート可能なダッシュボードが完全に更新されます。ページとウィジェットは新しい内容で完全に上書きされます。デフォルト: false。
discoveryRules	object	LLD ルールをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい LLD ルールが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の LLD ルールが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しない LLD ルールがデータベースから削除されます。デフォルト: false。
global_regexes	object	グローバル正規表現をどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいグローバル正規表現が作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のグローバル正規表現が更新され、そのサブ式は完全に上書きされます。デフォルト: false。
graphs	object	グラフをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいグラフが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のグラフが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないグラフがデータベースから削除されます。デフォルト: false。
host_groups	object	ホストグループをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいホストグループが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のホストグループが更新されます。デフォルト: false。
hosts	object	ホストをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいホストが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のホストが更新されます。デフォルト: false。
httptests	object	Web シナリオをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい Web シナリオが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の Web シナリオが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しない Web シナリオがデータベースから削除されます。デフォルト: false。

Parameter	Type	Description
images	object	画像をどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい画像が作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の画像が更新されます。デフォルト: false。
items	object	アイテムをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいアイテムが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のアイテムが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないアイテムがデータベースから削除されます。デフォルト: false。
maps	object	マップをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいマップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のマップが更新されます。デフォルト: false。
mediaTypes	object	メディアタイプをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいメディアタイプが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のメディアタイプが更新されます。デフォルト: false。
templateDashboards	object	テンプレートダッシュボードをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいテンプレートダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートダッシュボードが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないテンプレートダッシュボードがデータベースから削除されます。デフォルト: false。
template_groups	object	テンプレートグループをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいテンプレートグループが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートグループが更新されます。デフォルト: false。
templateLinkage	object	テンプレートリンクをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされていないが、インポートされたデータに存在するテンプレートがリンクされます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされているが、インポートされたデータに存在しないテンプレートのリンクが解除されます。このとき、リンク解除されたテンプレートから継承したエンティティー（アイテム、トリガーなど）は削除されません。デフォルト: false。

Parameter	Type	Description
templates	object	テンプレートをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいテンプレートが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートが更新されます。デフォルト: false。
triggers	object	トリガーをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいトリガーが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のトリガーが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないトリガーがデータベースから削除されます。デフォルト: false。
valueMaps	object	ホストまたはテンプレートの値マップをどのようにインポートするかのルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい値マップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の値マップが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しない値マップがデータベースから削除されます。デフォルト: false。

戻り値

(boolean) インポートに成功した場合は true を返します。

例

テンプレートのインポート

XML 文字列に含まれるテンプレート設定をインポートします。XML 文字列にアイテムやトリガーが不足している場合、それらはデータベースから削除され、それ以外は変更されません。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.import",
    "params": {
        "format": "xml",
        "rules": {
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
```

```
                "updateExisting": false
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```


ソース

CConfiguration::import() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

説明

`array configuration.importcompare(object parameters)`

このメソッドを使用すると、インポートファイルを現在のシステム要素と比較し、このインポートファイルを取り込んだ場合に何が変更されるかを表示できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`object`) インポート可能なデータと、そのデータの処理方法に関するルールを含むパラメーターです。

Parameter	Type	Description
format	string	シリアライズされた文字列の形式。 指定可能な値: `yaml` - YAML; `xml` - XML; `json` - JSON. パラメーターの動作: - *required*
source	string	設定データを含むシリアライズされた文字列。 パラメーターの動作: - *required*
rules	object	新規オブジェクトと既存オブジェクトをどのように比較するかに関するルールです。 *Admin* および *User* タイプのユーザーは、*read-only* または *read-write* 権限を持つ既存のオブジェクトとのみ、新規オブジェクトを比較できます。 `rules` パラメーターの詳細は、下の表で説明されています。 パラメーターの動作: - *required*
returnMissingObjects	boolean	レスポンスのオブジェクトに不足しているオブジェクトの情報を含めるかどうかのルールです。デフォルト: `false`。

Attention:
ルールが指定されていない場合、更新対象はなく、結果は空になります。

Note:
比較はホストグループとテンプレートに対してのみ行われます。トリガーとグラフは、インポートされたテンプレートに対してのみ比較され、それ以外はすべて「新規」と見なされます。

`rules` オブジェクトは、次のパラメーターをサポートします。

Parameter	Type	Description
dashboards	object	ダッシュボードのインポート方法に関するルールです。 サポートされるパラメーター: `createMissing` - (`boolean`) `true` に設定すると、新しいダッシュボードが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、特定のインポート可能なダッシュボードがシステムに存在する場合、それは完全に更新されます。ページとウィジェットは新しい内容で完全に上書きされます。デフォルト: `false`。
discoveryRules	object	LLD ルールのインポート方法に関するルールです。 サポートされるパラメーター: `createMissing` - (`boolean`) `true` に設定すると、新しい LLD ルールが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存の LLD ルールが更新されます。デフォルト: `false`; `deleteMissing` - (`boolean`) `true` に設定すると、インポートされたデータに存在しない LLD ルールがデータベースから削除されます。デフォルト: `false`。
global_regexes	object	グローバル正規表現のインポート方法に関するルールです。 サポートされるパラメーター: `createMissing` - (`boolean`) `true` に設定すると、新しいグローバル正規表現が作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のグローバル正規表現が更新され、その部分式は完全に上書きされます。デフォルト: `false`。
graphs	object	グラフのインポート方法に関するルールです。 サポートされるパラメーター: `createMissing` - (`boolean`) `true` に設定すると、新しいグラフが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のグラフが更新されます。デフォルト: `false`; `deleteMissing` - (`boolean`) `true` に設定すると、インポートされたデータに存在しないグラフがデータベースから削除されます。デフォルト: `false`。
host_groups	object	ホストグループのインポート方法に関するルールです。 サポートされるパラメーター: `createMissing` - (`boolean`) `true` に設定すると、新しいホストグループが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のホストグループが更新されます。デフォルト: `false`。

Parameter	Type	Description
httptests	object	Web シナリオのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい Web シナリオが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の Web シナリオが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しない Web シナリオがデータベースから削除されます。デフォルト: false。
images	object	画像のインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい画像が作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の画像が更新されます。デフォルト: false。 このパラメーターは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
items	object	アイテムのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいアイテムが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のアイテムが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないアイテムがデータベースから削除されます。デフォルト: false。
maps	object	マップのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいマップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のマップが更新されます。デフォルト: false。 このパラメーターは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
mediaTypes	object	メディアタイプのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいメディアタイプが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のメディアタイプが更新されます。デフォルト: false。 このパラメーターは出力に影響しません。configuration.import との整合性のためにのみ許可されています。
templateDashboards	object	テンプレートダッシュボードのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいテンプレートダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートダッシュボードが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないテンプレートダッシュボードがデータベースから削除されます。デフォルト: false。

Parameter	Type	Description
template_groups	object	テンプレートグループのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいテンプレートグループが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートグループが更新されます。デフォルト: false。
templateLinkage	object	テンプレートリンクのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされていないが、インポートされたデータに存在するテンプレートがリンクされます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされているが、インポートされたデータに存在しないテンプレートのリンクが解除されます。このとき、リンク解除されたテンプレートから継承したエンティティー（アイテム、トリガーなど）は削除されません。デフォルト: false。
templates	object	テンプレートのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいテンプレートが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートが更新されます。デフォルト: false。
triggers	object	トリガーのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しいトリガーが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のトリガーが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しないトリガーがデータベースから削除されます。デフォルト: false。
valueMaps	object	ホストまたはテンプレートの値マップのインポート方法に関するルールです。 サポートされるパラメーター: createMissing - (boolean) true に設定すると、新しい値マップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の値マップが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートされたデータに存在しない値マップがデータベースから削除されます。デフォルト: false。

戻り値

(array) 行われる設定の変更を含む配列を返します。

例

テンプレートのインポートの比較

XML 文字列に含まれるテンプレートを現在のシステム要素と比較し、このテンプレートをインポートした場合に何が変更されるかを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.importcompare",
    "params": {
```

```
        "format": "xml",
        "rules": {
            "discoveryRules": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "graphs": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "host_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "template_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "httptests": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "templateLinkage": {
                "createMissing": true,
                "deleteMissing": true
            },
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "templateDashboards": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><templ
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "templates": {
            "updated": [
                {
                    "before": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "after": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "items": {
                        "added": [
                            {
                                "after": {
                                    "uuid": "648006da5971424ead0c47ddbbf1ea2e",
                                    "name": "CPU utilization",
                                    "key": "system.cpu.util",
                                    "value_type": "FLOAT",
                                    "units": "%"
                                },
                                "triggers": {
                                    "added": [
                                        {
                                            "after": {
                                                "uuid": "736225012c534ec480c2a66a91322ce0",
                                                "expression": "avg(/New template/system.cpu.util,3m)>70",
                                                "name": "CPU utilization too high on 'New host' for 3 minu
                                                "priority": "WARNING"
                                            }
                                        }
                                    ]
                                }
                            }
                        ],
                        "removed": [
                            {
                                "before": {
                                    "uuid": "6805d4c39a624a8bab2cc8ab63df1ab3",
                                    "name": "CPU load",
                                    "key": "system.cpu.load",
                                    "value_type": "FLOAT"
                                },
                                "triggers": {
                                    "removed": [
                                        {
                                            "before": {
                                                "uuid": "ab4c2526c2bc42e48a633082255ebcb3",
                                                "expression": "avg(/New template/system.cpu.load,3m)>2",
```

```
                                        "name": "CPU load too high on 'New host' for 3 minutes",
                                        "priority": "WARNING"
                                    }
                                }
                            ]
                        }
                    }
                ],
                "updated": [
                    {
                        "before": {
                            "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                            "name": "Zabbix agent ping",
                            "key": "agent.ping"
                        },
                        "after": {
                            "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                            "name": "Zabbix agent ping",
                            "key": "agent.ping",
                            "delay": "3m"
                        }
                    }
                ]
            }
        }
    ]
    }
},
"id": 1
}
```


ソース

CConfiguration::importcompare() in *ui/include/classes/api/services/CConfiguration.php*.

Connector

このクラスは、コネクタオブジェクトを操作するためのものです。

オブジェクトリファレンス:

- Connector
- Tag filter

利用可能なメソッド:

- connector.create - 新しいコネクタを作成する
- connector.delete - コネクタを削除する
- connector.get - コネクタを取得する
- connector.update - コネクタを更新する

Connector オブジェクト

次のオブジェクトは、connectorAPI に直接関連しています。

コネクター

コネクターオブジェクトには、次のプロパティがあります。

Property	Type	Description
connectorid	ID	コネクターの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	コネクターの名前。 プロパティの動作: - 作成操作では必須
url	string	エンドポイント URL、つまり受信側の URL。 ユーザーマクロに対応しています。 プロパティの動作: - 作成操作では必須
protocol	integer	通信プロトコル。 指定可能な値: 0 - *(デフォルト)* Zabbix Streaming Protocol v1.0。
data_type	integer	データタイプ。 指定可能な値: 0 - *(デフォルト)* アイテムの値; 1 - イベント。
item_value_type	integer	送信するアイテムの値タイプの合計。 指定可能な値: 1 - 数値（浮動小数点）; 2 - 文字; 4 - ログ; 8 - 数値（符号なし）; 16 - テキスト; 32 - バイナリ; 64 - JSON。 デフォルト: 31 - すべてのアイテムタイプ（バイナリと JSON を除く）。 プロパティの動作: - `data_type` が「Item values」に設定されている場合にサポート
max_records	integer	1 つのメッセージ内で送信できるイベントまたはアイテムの最大数。 指定可能な値: 0-2147483647（32 ビット符号付き整数の最大値）。 デフォルト: 0 - 無制限。
max_senders	integer	このコネクターで実行する送信プロセスの数。 指定可能な値: 1-100。 デフォルト: 1。
max_attempts	integer	試行回数。 指定可能な値: 1-5。 デフォルト: 1。
attempt_interval	string	再試行間の間隔。 秒単位で指定します。 指定可能な値: 0s-10s。 デフォルト: 5s。 プロパティの動作: - `max_attempts` が 1 より大きい場合にサポート

Property	Type	Description
timeout	string	タイムアウト。 時間のサフィックスに対応しています（例: 30s、1m）。 ユーザーマクロに対応しています。 指定可能な値: 1s-60s。 デフォルト: 5s。
http_proxy	string	*[protocol]://[username[:password]@]proxy.example.com[:port]* の形式で指定する HTTP(S) プロキシ接続文字列。 ユーザーマクロに対応しています。
authtype	integer	HTTP 認証方式。 指定可能な値: 0 - *(デフォルト)* なし; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer。
username	string	ユーザー名。 ユーザーマクロに対応しています。 プロパティの動作: - `authtype` が「Basic」、「NTLM」、「Kerberos」、または「Digest」に設定されている場合にサポート
password	string	パスワード。 ユーザーマクロに対応しています。 プロパティの動作: - `authtype` が「Basic」、「NTLM」、「Kerberos」、または「Digest」に設定されている場合にサポート
token	string	Bearer トークン。 ユーザーマクロに対応しています。 プロパティの動作: - `authtype` が「Bearer」に設定されている場合に必須
verify_peer	integer	ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
verify_host	integer	接続先のホスト名がホストの証明書に記載されたものと一致することを検証するかどうか。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
ssl_cert_file	string	公開 SSL キーのファイルパス。 ユーザーマクロに対応しています。
ssl_key_file	string	秘密 SSL キーのファイルパス。 ユーザーマクロに対応しています。
ssl_key_password	string	SSL キーファイルのパスワード。 ユーザーマクロに対応しています。
description	text	コネクターの説明。
status	integer	コネクターが有効かどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

Property	Type	Description
tags_evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。

タグフィルター

タグフィルターを使用すると、一致するアイテムの値またはイベントのみをエクスポートできます。設定しない場合は、すべてがエクスポートされます。タグフィルターオブジェクトには、次のプロパティがあります。

Property	Type	Description
tag	string	タグ名。 Property behavior: *- required*
operator	integer	条件operator。 Possible values: 0 - *(default)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 12 - 存在する; 13 - 存在しない。
value	string	タグの値。

connector.create

説明

`object connector.create(object/array connectors)`

このメソッドを使用すると、新しいコネクタオブジェクトを作成できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するコネクタオブジェクト。

標準のコネクタプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
tags	array	コネクタのtag filter。

戻り値

(`object`) `connectorids` プロパティの下に作成されたコネクタの ID を含むオブジェクトを返します。返される ID の順序は、渡されたコネクタの順序と一致します。

例

コネクタの作成

タグフィルターを使用してトリガーイベントをエクスポートするためのコネクタを作成します。HTTP 認証は Bearer トークンを使用して実行されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.create",
    "params": [
        {
            "name": "Export of events",
            "data_type": 1,
            "url": "{$DATA_EXPORT_URL}",
            "authtype": 5,
            "token": "{$DATA_EXPORT_BEARER_TOKEN}",
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

ソース

CConnector::create() in *ui/include/classes/api/services/CConnector.php*.

connector.delete

説明

`object connector.delete(array connectorids)`

このメソッドを使用すると、connector エントリを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するコネクタの ID。

戻り値

(object) connectorids プロパティの下にある、削除されたコネクタの ID を含むオブジェクトを返します。

例

複数のコネクタの削除

2 つのコネクタエントリを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CConnector::delete() in *ui/include/classes/api/services/CConnector.php*.

connector.get

説明

`integer/array connector.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って connector オブジェクトを取得できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
connectorids	ID/array	指定された ID を持つコネクタのみを返します。
selectTags	query	コネクタのtag filterを含む `tags` プロパティを返します。 `count` をサポートします。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。 指定可能な値: `connectorid`, `name`, `data_type`, `status`。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

すべてのコネクタを取得する

すべてのコネクタとそのプロパティに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Export of item values",
            "protocol": "0",
            "data_type": "0",
            "url": "{$DATA_EXPORT_VALUES_URL}",
            "item_value_type": "31",
            "authtype": "4",
            "username": "{$DATA_EXPORT_VALUES_USERNAME}",
            "password": "{$DATA_EXPORT_VALUES_PASSWORD}",
            "token": "",
            "max_records": "0",
            "max_senders": "4",
            "max_attempts": "2",
            "attempt_interval": "10s",
            "timeout": "10s",
            "http_proxy": "{$DATA_EXPORT_VALUES_PROXY}",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "{$DATA_EXPORT_VALUES_SSL_CERT_FILE}",
            "ssl_key_file": "{$DATA_EXPORT_VALUES_SSL_KEY_FILE}",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "component",
                    "operator": "0",
                    "value": "memory"
                }
            ]
        },
        {
            "connectorid": "2",
            "name": "Export of events",
            "protocol": "0",
            "data_type": "1",
```

```
            "url": "{$DATA_EXPORT_EVENTS_URL}",
            "item_value_type": "31",
            "authtype": "5",
            "username": "",
            "password": "",
            "token": "{$DATA_EXPORT_EVENTS_BEARER_TOKEN}",
            "max_records": "0",
            "max_senders": "2",
            "max_attempts": "1",
            "attempt_interval": "5s",
            "timeout": "5s",
            "http_proxy": "",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "scope",
                    "operator": "0",
                    "value": "performance"
                }
            ]
        }
    ],
    "id": 1
}
```

ソース

CConnector:get() in *ui/include/classes/api/services/CConnector.php*.

connector.update

説明

`object connector.update(object/array connectors)`

このメソッドを使用すると、既存のコネクタを更新できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するコネクタのプロパティ。

各コネクタに対して connectorid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のコネクタプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
tags	array	現在のタグフィルターを置き換えるコネクタのtag filter。

戻り値

(object) connectorids プロパティの下にある、更新されたコネクタの ID を含むオブジェクトを返します。

例

HTTP 認証タイプの変更

ID ”3” のコネクタの HTTP 認証タイプをベアラーに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

タグフィルターの更新

ID ”5” のコネクタのタグフィルターを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": [
        {
            "connectorid": 5,
            "tags_evaltype": 2,
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                },
                {
                    "tag": "error",
                    "operator": 12,
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
```

```
    "id": 1
}
```

ソース

CConnector::update() in *ui/include/classes/api/services/CConnector.php*.

Graph

このクラスは、グラフを操作するためのものです。

オブジェクトリファレンス:

- Graph

利用可能なメソッド:

- graph.create - 新しいグラフの作成
- graph.delete - グラフの削除
- graph.get - グラフの取得
- graph.update - グラフの更新

Graph オブジェクト

次のオブジェクトは、graphAPI に直接関連しています。

グラフ

グラフオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
graphid	ID	グラフの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
height	integer	グラフの高さ（ピクセル単位）。 プロパティの動作: - 作成操作では 必須
name	string	グラフの名前。 プロパティの動作: - 作成操作では 必須
width	integer	グラフの幅（ピクセル単位）。 プロパティの動作: - 作成操作では 必須
flags	integer	グラフの生成元。 使用可能な値: 0 - *(デフォルト)* 通常のグラフ; 4 - プロトタイプから変換されたグラフ。 プロパティの動作: - 読み取り専用
graphtype	integer	グラフのレイアウトタイプ。 使用可能な値: 0 - *(デフォルト)* 通常; 1 - 積み上げ; 2 - 円グラフ; 3 - 分離。

プロパティ	型	説明
percent_left	float	左パーセンタイル。 デフォルト: 0。
percent_right	float	右パーセンタイル。 デフォルト: 0。
show_3d	integer	円グラフおよび分離グラフを 3D で表示するかどうか。 使用可能な値: 0 - *(デフォルト)* 2D で表示; 1 - 3D で表示。
show_legend	integer	グラフに凡例を表示するかどうか。 使用可能な値: 0 - 非表示; 1 - *(デフォルト)* 表示。
show_work_period	integer	グラフに稼働時間を表示するかどうか。 使用可能な値: 0 - 非表示; 1 - *(デフォルト)* 表示。
show_triggers	integer	グラフにトリガー線を表示するかどうか。 使用可能な値: 0 - 非表示; 1 - *(デフォルト)* 表示。
templateid	ID	親テンプレートグラフの ID。 プロパティの動作: - 読み取り専用
yaxismax	float	Y 軸の固定最大値。 デフォルト: 100。
yaxismin	float	Y 軸の固定最小値。 デフォルト: 0。
ymax_itemid	ID	Y 軸の最大値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは `ymax_type` が”calculated” に設定されているものとして描画されます。
ymax_type	integer	Y 軸の最大値の計算方法。 使用可能な値: 0 - *(デフォルト)* calculated; 1 - fixed; 2 - item。
ymin_itemid	ID	Y 軸の最小値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは `ymin_type` が”calculated” に設定されているものとして描画されます。
ymin_type	integer	Y 軸の最小値の計算方法。 使用可能な値: 0 - *(デフォルト)* calculated; 1 - fixed; 2 - item。
uuid	string	インポートされたグラフを既存のグラフに関連付けるために使用される汎用一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - グラフがテンプレートに属している場合に サポート

graph.create

説明

`object graph.create(object/array graphs)`

このメソッドを使用すると、新しいグラフを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するグラフ。

標準のグラフプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	グラフ用に作成されるグラフアイテム。 パラメータの動作: - 必須

戻り値

(object) graphids プロパティの下に作成されたグラフの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグラフの順序と一致します。

例

グラフの作成

2 つのアイテムからなるグラフを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.create",
    "params": {
        "name": "MySQL bandwidth",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
```

```
    "id": 1
}
```

関連項目

- グラフアイテム

ソース

CGraph::create() in *ui/include/classes/api/services/CGraph.php*.

graph.delete

説明

`object graph.delete(array graphIds)`

このメソッドを使用すると、グラフを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(array) 削除するグラフの ID。

戻り値

(object) graphids プロパティの下にある削除されたグラフの ID を含むオブジェクトを返します。

例

複数のグラフの削除

2 つのグラフを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

ソース

CGraph::delete() in *ui/include/classes/api/services/CGraph.php*.

graph.get

説明

integer/array graph.get(object parameters)

このメソッドを使用すると、指定したパラメーターに従ってグラフを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
graphids	ID/array	指定した ID のグラフのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループ内のホストまたはテンプレートに属するグラフのみを返します。
templateids	ID/array	指定したテンプレートに属するグラフのみを返します。
hostids	ID/array	指定したホストに属するグラフのみを返します。
itemids	ID/array	指定したアイテムを含むグラフのみを返します。
templated	boolean	true に設定した場合、テンプレートに属するグラフのみを返します。
inherited	boolean	true に設定した場合、テンプレートから継承されたグラフのみを返します。
expandName	flag	グラフ名内のマクロを展開します。
selectDiscoveryData	query	グラフディスカバリオブジェクトのデータを含む discoveryData プロパティを返します。グラフディスカバリオブジェクトは、ディスカバリされたグラフを、そのグラフの元となったグラフプロトタイプに関連付けます。 以下のプロパティを持ちます。 parent_graphid - (ID) グラフの作成元となったグラフプロトタイプの ID。 status - (int) グラフディスカバリのステータス: 0 - *(デフォルト)* グラフはディスカバリされています。 1 - グラフはすでにディスカバリされていません。 ts_delete - (timestamp) すでにディスカバリされなくなったグラフが削除される時刻。
selectDiscoveryRule	query	グラフを作成したローレベルディスカバリルールを含むdiscoveryRule プロパティを返します。
selectHostGroups	query	グラフが属するホストグループを含むhostgroups プロパティを返します。
selectTemplateGroups	query	グラフが属するテンプレートグループを含むtemplategroups プロパティを返します。
selectTemplates	query	グラフが属するテンプレートを含むtemplates プロパティを返します。
selectHosts	query	グラフが属するホストを含むhosts プロパティを返します。
selectItems	query	グラフで使用されるアイテムを含むitems プロパティを返します。
selectGraphItems	query	グラフで使用されるアイテムを含むgitems プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 text データ型 のプロパティはサポートしていません。 追加のプロパティをサポートしています。 host - グラフが属するホストの技術名。 hostid - グラフが属するホストの ID。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: graphid, name, graphtype。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
editable	boolean	

パラメータ	型	説明
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGraphDiscovery	query	グラフディスカバリオブジェクトを含む graphDiscovery プロパティを返します。グラフディスカバリオブジェクトは、グラフをその作成元となったグラフプロトタイプに関連付けます。 このクエリは 非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストからグラフの取得

ホスト"10107" から全てのグラフを取得し、名前でソートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "612",
            "name": "CPU jumps",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "439",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
```

```
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "613",
            "name": "CPU load",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "433",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "614",
            "name": "CPU utilization",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "387",
            "show_work_period": "1",
            "show_triggers": "0",
            "graphtype": "1",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "1",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "645",
            "name": "Disk space usage /",
            "width": "600",
            "height": "340",
            "yaxismin": "0",
            "yaxismax": "0",
            "templateid": "0",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
```

```
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "4"
        }
    ],
    "id": 1
}
```

関連項目

- ディスカバリルール
- グラフアイテム
- アイテム
- ホスト
- ホストグループ
- テンプレート
- テンプレートグループ

ソース

CGraph::get() in *ui/include/classes/api/services/CGraph.php*.

graph.update

説明

`object graph.update(object/array graphs)`

このメソッドを使用すると、既存のグラフを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) 更新するグラフのプロパティ。

各グラフでは graphid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準のグラフプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	既存のグラフアイテムを置き換えるグラフアイテム。グラフアイテムに gitemid プロパティが定義されている場合は更新され、そうでない場合は新しいグラフアイテムが作成されます。

戻り値

(object) graphids プロパティの下で更新されたグラフの ID を含むオブジェクトを返す。

例

Y スケールの最大値の設定

Y スケールの最大値を 100 の固定値に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

ソース

CGraph::update() in *ui/include/classes/api/services/CGraph.php*.

Housekeeping

このクラスは、ハウスキーピングを操作するためのものです。

オブジェクトリファレンス:

- Housekeeping

利用可能なメソッド:

- housekeeping.get - ハウスキーピングの取得
- housekeeping.update - ハウスキーピングの更新

Housekeeping オブジェクト

次のオブジェクトは、housekeepingAPI に直接関連しています。

ハウスキーピング

設定オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
hk_events_mode	integer	イベントおよびアラートの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_events_trigger	string	トリガーデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 365d。
hk_events_service	string	サービスデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 1d。
hk_events_internal	string	内部データの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 1d。
hk_events_discovery	string	ネットワークディスカバリデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 1d。

プロパティ	型	説明
hk_events_autoreg	string	自動登録データの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 1d。
hk_services_mode	integer	サービスの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - (デフォルト) 有効。
hk_services	string	サービスデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 365d。
hk_audit_mode	integer	監査の内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - (デフォルト) 有効。
hk_audit	string	監査データの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 31d。
hk_sessions_mode	integer	セッションの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - (デフォルト) 有効。
hk_sessions	string	セッションデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 31d。
hk_history_mode	integer	ヒストリの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - (デフォルト) 有効。
hk_history_global	integer	アイテムのヒストリ保存期間を上書きします。 指定可能な値: 0 - 上書きしない; 1 - (デフォルト) 上書きする。
hk_history	string	ヒストリデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 31d。
hk_trends_mode	integer	トレンドの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - (デフォルト) 有効。
hk_trends_global	integer	アイテムのトレンド保存期間を上書きします。 指定可能な値: 0 - 上書きしない; 1 - (デフォルト) 上書きする。
hk_trends	string	トレンドデータの保存期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 365d。

プロパティ	型	説明
db_extension	string	設定フラグ DB extension。このフラグに「timescaledb」を設定すると、サーバーはハウスキーピングおよびアイテムの削除に関する動作を変更します。 プロパティの動作: - 読み取り専用
compression_availability	integer	データベース（またはその拡張機能）がデータ圧縮をサポートしているかどうか。 指定可能な値: 0 - 利用不可; 1 - 利用可能。 プロパティの動作: - 読み取り専用
compression_status	integer	ヒストリおよびトレンドの TimescaleDB 圧縮を有効にします。 指定可能な値: 0 - *(デフォルト)* オフ; 1 - オン。
compress_older	string	指定した期間より古いヒストリおよびトレンドのレコードを圧縮します。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定します。 デフォルト: 7d。

housekeeping.get

説明

`object housekeeping.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って housekeeping オブジェクトを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	このパラメータについては、リファレンスコメントで説明されています。

戻り値

(object) housekeeping オブジェクトを返します

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "hk_events_mode": "1",
        "hk_events_trigger": "365d",
        "hk_events_service": "1d",
        "hk_events_internal": "1d",
        "hk_events_discovery": "1d",
        "hk_events_autoreg": "1d",
        "hk_services_mode": "1",
        "hk_services": "365d",
        "hk_audit_mode": "1",
        "hk_audit": "31d",
        "hk_sessions_mode": "1",
        "hk_sessions": "365d",
        "hk_history_mode": "1",
        "hk_history_global": "0",
        "hk_history": "31d",
        "hk_trends_mode": "1",
        "hk_trends_global": "0",
        "hk_trends": "365d",
        "db_extension": "",
        "compression_status": "0",
        "compress_older": "7d"
    },
    "id": 1
}
```


ソース

CHousekeeping ::get() in *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

説明

`object housekeeping.update(object housekeeping)`

このメソッドを使用すると、既存の housekeeping 設定を更新できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新するハウスキーピングプロパティ。

戻り値

(array) 更新されたパラメーター名の配列を返します。

例

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.update",
    "params": {
        "hk_events_mode": "1",
        "hk_events_trigger": "200d",
        "hk_events_internal": "2d",
        "hk_events_discovery": "2d"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

ソース

CHousekeeping::update() in *ui/include/classes/api/services/CHousekeeping.php*.

Image

このクラスは、画像を操作するためのものです。

オブジェクトリファレンス:

- Image

利用可能なメソッド:

- image.create - 新しい画像を作成する
- image.delete - 画像を削除する
- image.get - 画像を取得する
- image.update - 画像を更新する

Image オブジェクト

次のオブジェクトは、imageAPI に直接関連しています。

画像

画像オブジェクトには以下のプロパティがあります。

プロパティ	Type	説明
imageid	ID	画像の ID。 Property behavior: - *read-only* - 更新操作で *required*
name	string	画像の名前。 Property behavior: - 作成操作で *required*
imagetype	integer	画像のタイプ。 使用可能な値: 1 - *(default)* アイコン; 2 - 背景画像。 Property behavior: - *constant* - 作成操作で *required*

プロパティ	Type	説明
image	string	Base64 でエンコードされた画像。 エンコードされた画像の最大サイズは 1 MB です。最大サイズは ZBX_MAX_IMAGE_SIZE 定数の値を変更することで調整できます。 サポートされる画像形式: PNG、JPEG、GIF、WebP。 Property behavior: - 作成操作で *required*

image.create

説明

`object image.create(object/array images)`

このメソッドを使用すると、新しい画像を作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するイメージ。

このメソッドは標準イメージプロパティを持つイメージを受け付けます。

戻り値

(object) imageids プロパティの下で作成された画像の ID を含むオブジェクトを返します。返される ID の順序は、渡された画像の順序と一致します。

例

アイコン画像の作成

クラウドアイコンを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

背景画像の作成

パターンとして使用する小さな背景画像を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```


ソース

CImage::create() in *ui/include/classes/api/services/CImage.php*.

image.delete

説明

`object image.delete(array imageIds)`

このメソッドを使用すると、画像を削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除する画像の ID。

戻り値

(object) imageids プロパティの下にある削除された画像の ID を含むオブジェクトを返します。

例

複数の画像の削除

2 つの画像を削除します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

ソース

CImage::delete() in *ui/include/classes/api/services/CImage.php*.

image.get

説明

`integer/array image.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って画像を取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 希望する出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
imageids	ID/array	指定した ID を持つ画像のみを返します。
sysmapids	ID/array	指定したマップで使用されている画像を返します。
select_image	flag	Base64 でエンコードされた画像を含む `image` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `imageid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

イメージの取得

ID が”2” のイメージの全データを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

ソース

CImage::get() in *ui/include/classes/api/services/CImage.php*.

image.update

説明

`object image.update(object/array images)`

このメソッドを使用すると、既存の画像を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するイメージのプロパティ。

各イメージに対して `imageid` プロパティを定義する必要があり、他のプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準イメージプロパティを持つイメージを受け付けます。

戻り値

(`object`) `imageids` プロパティの下で更新された画像の ID を含むオブジェクトを返します。

例

イメージの名前変更

イメージの名前を「Cloud icon」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CImage::update() in *ui/include/classes/api/services/CImage.php*.

LLD ルール

このクラスは、ローレベルディスカバリルールで動作するように設計されています。

オブジェクト参照:

- LLD ルール
  - HTTP ヘッダー
  - HTTP クエリフィールド
- LLD ルールフィルター
  - LLD ルールフィルター条件
- LLD マクロパス
- LLD ルール前処理
- LLD ルールオーバーライド
  - LLD ルールオーバーライドフィルター
    - LLD ルールオーバーライドフィルター条件
  - LLD ルールオーバーライド操作
    - LLD ルールオーバーライド操作ステータス
    - LLD ルールオーバーライド操作ディスカバリ
    - LLD ルールオーバーライド操作期間
    - LLD ルールオーバーライド操作履歴
    - LLD ルールオーバーライド操作トレンド
    - LLD ルールオーバーライド操作重大度
    - LLD ルールオーバーライド操作タグ
    - LLD ルールオーバーライド操作テンプレート
    - LLD ルールオーバーライド操作インベントリ

利用可能なメソッド:

- discoveryrule.create - 新しい LLD ルールを作成
- discoveryrule.delete - LLD ルールを削除
- discoveryrule.get - LLD ルールを取得
- discoveryrule.update - LLD ルールを更新

LLD rule オブジェクト

次のオブジェクトは、`discoveryruleAPI` に直接関連しています。

LLD ルール

ローレベルディスカバリルールオブジェクトには、次のプロパティがあります。

Property	Type	Description
itemid	ID	LLD ルールの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
delay	string	LLD ルールの更新間隔。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはオプションで 1 つ以上のカスタム間隔を受け付けます。これらはすべてセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはマクロとテキストの組み合わせはサポートされません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: {$FLEX_INTERVAL}/{$FLEX_PERIOD}）。 例: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr プロパティの動作: - type が”Zabbix agent” (0)、”Simple check” (3)、”Zabbix internal” (5)、”External check” (10)、”Database monitor” (11)、”IPMI agent” (12)、”SSH agent” (13)、”TELNET agent” (14)、”JMX agent” (16)、”HTTP agent” (19)、”SNMP agent” (20)、”Script” (21)、”Browser” (22) に設定されている場合、または type が”Zabbix agent (active)” (7) に設定され、key_ に”mqtt.get” が含まれていない場合に必須
hostid	ID	LLD ルールが属するホストの ID。 プロパティの動作: - 固定 - 作成操作では必須
flags	integer	ディスカバリルールの起点。 指定可能な値: 1 - ローレベルディスカバリルール; 5 - プロトタイプから変換されたローレベルディスカバリルール。 プロパティの動作: - 読み取り専用
interfaceid	ID	LLD ルールのホストインターフェースの ID。 プロパティの動作: - LLD ルールがホストに属し、type が”Zabbix agent”、”IPMI agent”、”JMX agent”、または”SNMP agent” に設定されている場合に必須 - LLD ルールがホストに属し、type が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合にサポート
key_	string	LLD ルールのキー。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用
name	string	LLD ルールの名前。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用

Property	Type	Description
type	integer	LLD ルールのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested。このタイプは、ルールの親がテンプレート（ディスカバリ時にホストプロトタイプにリンクされることを想定）または検出されたホスト（ルートに LLD ルールがある）である場合にのみ許可されます。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が”HTTP agent” に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用
allow_traps	integer	trapper アイテムと同様に値の格納を許可します。 指定可能な値: 0 - *(デフォルト)* 受信データを受け付けない; 1 - 受信データを受け付ける。 プロパティの動作: - type が”HTTP agent” に設定されている場合にサポート
authtype	integer	認証方式。 type が”SSH agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest。 プロパティの動作: - type が”SSH agent” または”HTTP agent” に設定されている場合にサポート - type が”HTTP agent” に設定されている場合、継承されたオブジェクトでは読み取り専用
description	string	LLD ルールの説明。

Property	Type	Description
error	string	LLD ルールの値の更新時に障害が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用
follow_redirects	integer	データのポーリング中にレスポンスのリダイレクトに従います。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。
headers	array	プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用 HTTP リクエストの実行時に送信されるヘッダーの配列。
http_proxy	string	プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用 HTTP(S) プロキ接続文字列。
ipmi_sensor	string	プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用 IPMI センサー。
jmx_endpoint	string	プロパティの動作: - `type` が”IPMI agent” に設定され、`key_` が”ipmi.get” に設定されていない場合に必須 - `type` が”IPMI agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用 JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi
lifetime	string	プロパティの動作: - `type` が”JMX agent” に設定されている場合にサポート 検出されなくなったアイテムが削除されるまでの期間。秒数（例: 3600）、時間のサフィックス付きの値（例: 3600s、180m、24h、1d）、またはユーザーマクロを受け付けます。 デフォルト: 7d。
lifetime_type	integer	失われた LLD リソースを削除するシナリオ。 指定可能な値: 0 - *(デフォルト)* 存続期間のしきい値に達した後に削除; 1 - 削除しない; 2 - 直ちに削除。
enabled_lifetime	string	検出されなくなったアイテムが無効になるまでの期間。秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはユーザーマクロを受け付けます。 デフォルト: 0。
enabled_lifetime_type	integer	失われた LLD リソースを無効にするシナリオ。 指定可能な値: 0 - 存続期間のしきい値に達した後に無効化; 1 - 無効化しない; 2 - *(デフォルト)* 直ちに無効化。

Property	Type	Description
master_itemid	ID	マスターアイテムの ID。 ディスカバリルールを別のディスカバリルールのマスターアイテムにすることはできません。 プロパティの動作: - type が”Dependent item” に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データとして保存; 1 - JSON に変換。 プロパティの動作: - type が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
params	string	LLD ルールのタイプに応じた追加パラメータ: - SSH および Telnet LLD ルールで実行されるスクリプト; - データベースモニター LLD ルールの SQL クエリ; - 計算 LLD ルールの計算式; - スクリプトおよびブラウザ LLD ルールのスクリプト。 プロパティの動作: - type が”Database monitor”、”SSH agent”、”TELNET agent”、”Script”、または”Browser” に設定されている場合に必須 - type が”Script” または”Browser” に設定されている場合、継承されたオブジェクトでは読み取り専用
parameters	object/array	type が”Script” または”Browser” に設定されている場合の追加パラメータ。 name および value プロパティを持つオブジェクトの配列で、name は一意でなければなりません。 プロパティの動作: - type が”Script” または”Browser” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
password	string	認証用パスワード。 プロパティの動作: - type が”JMX agent” に設定され、username が設定されている場合に必須 - type が”Simple check”、”Database monitor”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合にサポート - type が”HTTP agent” に設定されている場合、継承されたオブジェクトでは読み取り専用
post_type	integer	posts プロパティに保存される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - type が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
posts	string	HTTP(S) リクエスト本文のデータ。 プロパティの動作: - type が”HTTP agent” に設定され、post_type が”JSON data” または”XML data” に設定されている場合に必須 - type が”HTTP agent” に設定され、post_type が”Raw data” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

Property	Type	Description
privatekey	string	秘密鍵ファイルの名前。 プロパティの動作: - `type` が”SSH agent” に設定され、`authtype` が”public key” に設定されている場合に必須
publickey	string	公開鍵ファイルの名前。 プロパティの動作: - `type` が”SSH agent” に設定され、`authtype` が”public key” に設定されている場合に必須
query_fields	array	HTTP リクエストの実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、または”PUT” に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が”HEAD” に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が”SNMP agent” に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ssl_key_password	string	SSL キーファイルのパスワード。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

Property	Type	Description
state	integer	LLD ルールの状態。 指定可能な値: 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	LLD ルールのステータス。 指定可能な値: 0 - *(デフォルト)* 有効な LLD ルール; 1 - 無効な LLD ルール。
status_codes	string	必要な HTTP ステータスコードの範囲をカンマで区切って指定します。 カンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
templateid	ID	親テンプレート LLD ルールの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m）を受け付けます。ユーザーマクロも受け付けます。 指定可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシまたはグローバル設定を使用。 プロパティの動作: - `type` が”Zabbix agent” (0)、”Simple check” (3) に設定され、`key_` が”`vmware.`” および”`icmpping`” で始まらない場合、”Zabbix agent (active)” (7)、”External check” (10)、”Database monitor” (11)、”SSH agent” (13)、”TELNET agent” (14)、”HTTP agent” (19)、”SNMP agent” (20) に設定され、`snmp_oid` が”`walk[`” または”`get[`” で始まる場合、”Script” (21)、”Browser” (22) に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用 - デフォルト値が適用され、アイテムがアクセスできないプロキシまたはプロキシグループによって監視されているホストに属する場合、グローバルタイムアウトが使用されます。
trapper_hosts	string	許可されたホスト。 プロパティの動作: - `type` が”Zabbix trapper” に設定されている場合、または `type` が”HTTP agent” に設定され、`allow_traps` が”Allow incoming data” に設定されている場合にサポート。 プロパティが省略された場合、`{$TRAPPER.ALLOWED_HOSTS}` ユーザーマクロ（デフォルト値）が自動的に設定されます。デフォルトでは、`{$TRAPPER.ALLOWED_HOSTS}` は `127.0.0.1,::1` です。 プロパティに空の文字列を設定すると、すべてのホストからの受信接続が拒否されます。

Property	Type	Description
username	string	認証用ユーザー名。 プロパティの動作: - `type` が”SSH agent”、”TELNET agent” に設定されている場合、または `type` が”JMX agent” に設定され、`password` が設定されている場合に必須 - `type` が”Simple check”、”Database monitor”、または”HTTP agent” に設定されている場合にサポート - `type` が”HTTP agent” に設定されている場合、継承されたオブジェクトでは読み取り専用
uuid	string	インポートした LLD ルールを既存のルールにリンクするために使用される一意の識別子。指定されていない場合は自動生成されます。 プロパティの動作: - LLD ルールがテンプレートに属している場合にサポート
verify_host	integer	接続先のホスト名がホストの証明書に記載された名前と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
verify_peer	integer	ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

HTTP ヘッダー

ヘッダーオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

LLD ルールフィルター

LLD ルールフィルターオブジェクトは、検出されたオブジェクトをフィルタリングするために使用できる一連の条件を定義します。以下のプロパティがあります。

プロパティ	Type	説明
conditions	object/array	結果のフィルタリングに使用するフィルター条件のセット。条件は、formula 内で配置された順序でソートされます。 Property behavior: - *required*
evaltype	integer	フィルター条件の評価方法。 使用可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - *required*
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式を持つフィルターでは `formula` の値と同じです。 Property behavior: - *read-only*
formula	string	カスタム式を持つフィルターの条件評価に使用する、ユーザー定義の式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれている必要があります。式で使用される ID は、フィルター条件で定義されたものと完全に一致している必要があります。どの条件も未使用または省略されたままにすることはできません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は *required*

LLD ルールのフィルター条件

LLD ルールのフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する対象の LLD マクロ。 Property behavior: - *required*
value	string	比較対象の値。 Property behavior: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は *required*
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際には、この ID をユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。 Property behavior: - LLD rule filter object の `evaltype` が「カスタム式」に設定されている場合は *required*

プロパティ	Type	説明
operator	integer	条件演算子。 指定可能な値: 8 - *(default)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、discoveryrule.get およびdiscoveryrule.create メソッドのページにある例を参照してください。

LLD マクロパス

LLD マクロパスには以下のプロパティがあります。

プロパティ	Type	説明
lld_macro	string	LLD マクロ。 プロパティの動作: - 必須
path	string	対応するマクロに割り当てられる値のセレクター。 プロパティの動作: - 必須

LLD ルールの前処理

LLD ルールの前処理オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
type	integer	前処理オプションのタイプ。 指定可能な値: 5 - 正規表現; 11 - XML XPath; 12 - JSONPath; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラーを確認; 17 - XML のエラーを確認; 20 - ハートビートで変更されていない値を破棄; 21 - JavaScript; 23 - Prometheus から JSON へ; 24 - CSV から JSON へ; 25 - 置換; 27 - XML から JSON へ; 28 - SNMP walk 値; 29 - SNMP walk から JSON へ; 30 - SNMP get 値。 プロパティの動作: - 必須

プロパティ	型	説明
params	string	前処理オプションで使用する追加パラメータ。複数のパラメータは改行 (\n) 文字で区切ります。 プロパティの動作: - type が「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーを確認」(16)、「XML のエラーを確認」(17)、「ハートビートで変更されていない値を破棄」(20)、「JavaScript」(21)、「Prometheus から JSON へ」(23)、「CSV から JSON へ」(24)、「置換」(25)、「SNMP walk 値」(28)、「SNMP walk から JSON へ」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler	integer	前処理ステップが失敗した場合に使用するアクションのタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定されます; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーを確認」(16)、「XML のエラーを確認」(17)、「Prometheus から JSON へ」(23)、「CSV から JSON へ」(24)、「XML から JSON へ」(27)、「SNMP walk 値」(28)、「SNMP walk から JSON へ」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler_params	string	エラーハンドラーのパラメータ。 プロパティの動作: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は 必須

各前処理タイプでサポートされるパラメータとエラーハンドラーは次のとおりです。

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
5	正規表現	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	正規表現に一致	pattern[1]			0, 1, 2, 3
15	正規表現に一致しない	pattern[1]			0, 1, 2, 3
16	JSON のエラーを確認	path[3]			0, 1, 2, 3
17	XML のエラーを確認	path[3]			0, 1, 2, 3

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
20	ハートビートで変更されていない値を破棄	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus から JSON へ	pattern^5, 6^			0, 1, 2, 3
24	CSV から JSON へ	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
27	XML から JSON へ				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON へ[7]	Field name[2]	OID prefix[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 正規表現
 [2] 文字列
 [3] JSONPath または XML XPath
 [4] 正の整数（30s、1m、2h、1d などの時間サフィックスをサポート）
 [5] ユーザーマクロ
 [6] 次の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。Prometheus パターンの各コンポーネント（メトリック、ラベル名、ラベル値、メトリック値）にはユーザーマクロを使用できます。
 [7] 改行文字で区切られた複数の「Field name,OID prefix,Format records」レコードをサポートします。

LLD ルールのオーバーライド

LLD ルールのオーバーライドオブジェクトは、異なるプロトタイプオブジェクトのプロパティを上書きするために使用される一連のルール（フィルター、条件、操作）を定義します。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
name	string	一意のオーバーライド名。 プロパティの動作: - 必須
step	integer	オーバーライドの一意の順序番号。 プロパティの動作: - 必須
stop	integer	一致した場合、後続のオーバーライドの処理を停止します。 指定可能な値: 0 - *(デフォルト)* オーバーライドの処理を停止しない; 1 - フィルターに一致した場合、オーバーライドの処理を停止する。
filter	object	オーバーライドフィルター。
operations	object/array	オーバーライド操作。

LLD ルールのオーバーライドフィルター

LLD ルールのオーバーライドフィルターオブジェクトは、検出されたオブジェクトに一致した場合にオーバーライドを適用する一連の条件を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
conditions	object/array	検出されたオブジェクトとの一致に使用するオーバーライドフィルター条件のセット。条件は、formula 内での配置順に並べ替えられます。 プロパティの動作: - 必須
evaltype	integer	オーバーライドフィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	オーバーライドフィルター条件の評価に使用される生成済みの式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID が含まれます。`eval_formula` の値は、カスタム式を使用するフィルターでは `formula` の値と同じです。 プロパティの動作: - 読み取り専用
formula	string	カスタム式を使用するオーバーライドフィルターの条件評価に使用する、ユーザー定義の式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID を含める必要があります。式で使用する ID は、オーバーライドフィルター条件で定義されたものと完全に一致していなければなりません。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「custom expression」に設定されている場合は必須

LLD ルールのオーバーライドフィルター条件

LLD ルールのオーバーライドフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する対象の LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は 必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際には、この ID をユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。 プロパティの動作: - LLD ルールのオーバーライドフィルターオブジェクト の `evaltype` が「カスタム式」に設定されている場合は 必須
operator	integer	条件演算子。 使用可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

LLD ルールのオーバーライド操作

LLD ルールのオーバーライド操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。以下のプロパティがあります。

プロパティ	Type	説明
operationobject	integer	アクションを実行する検出済みオブジェクトのタイプ。 使用可能な値: 0 - アイテムのプロトタイプ; 1 - トリガーのプロトタイプ; 2 - グラフのプロトタイプ; 3 - ホストのプロトタイプ; 4 - ディスカバリのプロトタイプ。 プロパティの動作: - 必須
operator	integer	オーバーライド条件のoperator。 使用可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 8 - 一致する; 9 - 一致しない。
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフ、またはホストのプロトタイプ名に一致させるパターン。
opstatus	object	アイテム、トリガー、およびホストのプロトタイプオブジェクト用のオーバーライド操作ステータスオブジェクト。
opdiscover	object	オーバーライド操作の検出ステータスオブジェクト（すべてのオブジェクトタイプ）。
opperiod	object	アイテムのプロトタイプオブジェクト用のオーバーライド操作期間（更新間隔）オブジェクト。
ophistory	object	アイテムのプロトタイプオブジェクト用のオーバーライド操作履歴オブジェクト。

プロパティ	Type	説明
optrends	object	アイテムのプロトタイプオブジェクト用のオーバーライド操作トレンドオブジェクト。
opseverity	object	トリガーのプロトタイプオブジェクト用のオーバーライド操作重要度オブジェクト。
optag	object/array	トリガーおよびホストのプロトタイプオブジェクト用のオーバーライド操作タグオブジェクト。
optemplate	object/array	ホストのプロトタイプオブジェクト用のオーバーライド操作テンプレートオブジェクト。
opinventory	object	ホストのプロトタイプオブジェクト用のオーバーライド操作インベントリオブジェクト。

LLD ルールのオーバーライド操作ステータス

ディスカバリされたオブジェクトに設定される LLD ルールのオーバーライド操作ステータスです。以下のプロパティがあります。

プロパティ	Type	説明
status	integer	選択したオブジェクトのステータスをオーバーライドします。 指定可能な値: 0 - 有効で作成; 1 - 無効で作成。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作 discover

LLD ルールのオーバーライド操作 discover は、検出されたオブジェクトに設定される検出ステータスです。
以下のプロパティがあります:

Property	Type	Description
discover	integer	選択したオブジェクトの検出ステータスをオーバーライドします。 指定可能な値: 0 - はい、オブジェクトの検出を継続します; 1 - いいえ、新しいオブジェクトは検出されず、既存のオブジェクトは失われたものとしてマークされます。 Property behavior: *- required*

LLD ルールのオーバーライド操作期間

LLD ルールのオーバーライド操作期間は、検出されたアイテムに設定される更新間隔の値です。以下のプロパティがあります。

Property	Type	Description
delay	string	アイテムのプロトタイプの更新間隔をオーバーライドします。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはオプションで 1 つ以上のカスタム間隔を指定できます。これらはすべてセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロまたは LLD マクロを指定できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはマクロとテキストを組み合わせた指定はサポートされていません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: {$FLEX_INTERVAL}/{$FLEX_PERIOD}）。 例: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr プロパティの動作: - *required*

LLD ルールのオーバーライド操作履歴

ディスカバリされたアイテムに設定される LLD ルールのオーバーライド操作履歴の値です。
以下のプロパティがあります:

プロパティ	Type	説明
history	string	アイテムプロトタイプの履歴を上書きします。これは履歴データをどのくらいの期間保存するかを示す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作のトレンド

LLD ルールのオーバーライド操作のトレンド値は、ディスカバリされたアイテムに設定されます。以下のプロパティがあります。

Property	Type	Description
trends	string	アイテムプロトタイプのトレンドをオーバーライドします。これは、トレンドデータをどのくらいの期間保存するかを示す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - *required*

LLD ルールのオーバーライド操作の深刻度

検出されたトリガーに設定される LLD ルールのオーバーライド操作の深刻度の値です。
以下のプロパティがあります。

Property	Type	Description
severity	integer	トリガープロトタイプの深刻度をオーバーライドします。 指定可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害. Property behavior: - *required*

LLD ルールのオーバーライド操作タグ

LLD ルールのオーバーライド操作タグオブジェクトには、検出されたオブジェクトに設定されるタグ名と値が含まれます。次のプロパティがあります。

Property	Type	Description
tag	string	新しいタグ名。ユーザーマクロおよび LLD マクロも使用できます。 Property behavior: - *required*
value	string	新しいタグ値。ユーザーマクロおよび LLD マクロも使用できます。

LLD ルールのオーバーライド操作テンプレート

検出されたホストにリンクされた LLD ルールのオーバーライド操作テンプレートオブジェクトです。
以下のプロパティがあります。

プロパティ	Type	説明
templateid	ID	ホストプロトタイプのリンク済みテンプレートのテンプレートをオーバーライドします。 プロパティの動作: - *必須*

LLD ルールのオーバーライド操作インベントリ

検出されたホストに設定される、LLD ルールのオーバーライド操作インベントリモードの値です。
以下のプロパティがあります。

Property	Type	Description
inventory_mode	integer	ホストプロトタイプのインベントリモードをオーバーライドします。 指定可能な値: -1 - 無効; 0 - *(デフォルト)* 手動; 1 - 自動。 Property behavior: - *required*

discoveryrule.create

説明

```
object discoveryrule.create(object/array lldRules)
```

このメソッドを使用すると、新しい LLD ルールを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成する LLD ルール。

標準の LLD ルールプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	LLD ルールのLLD ルールフィルター。
preprocessing	object/array	LLD ルールの前処理オプション。
lld_macro_paths	object/array	LLD ルールのlld_macro_pathオプション。
overrides	object/array	LLD ルールのオーバーライドオプション。

戻り値

(object) itemids プロパティの下で作成された LLD ルールの ID を含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールの順序と一致します。

例

LLD ルールの作成

マウントされたファイルシステムを検出するための Zabbix エージェント LLD ルールを作成します。検出されたアイテムは 30 秒ごとに更新されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Mounted filesystem discovery",
        "key_": "vfs.fs.discovery",
        "hostid": "10197",
        "type": 0,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

フィルターの使用

条件のセットを持つ LLD ルールを作成し、結果をフィルタリングします。条件は論理演算子「and」でまとめられます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
```

```
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex2",
                    "operator": "9"
                },
                {
                    "macro": "{#MACRO3}",
                    "value": "",
                    "operator": "12"
                },
                {
                    "macro": "{#MACRO4}",
                    "value": "",
                    "operator": "13"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

LLD マクロパスを持つ LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "LLD rule with LLD macro paths",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.path.1"
            },
```

```
            {
                "lld_macro": "{#MACRO2}",
                "path": "$.path.2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

条件を評価するためにカスタム式を使用するフィルターを持つ LLD ルールを作成します。LLD ルールは、”{#MACRO1}” マクロ値が正規表現”regex1” および”regex2” の両方に一致し、かつ”{#MACRO2}” の値が”regex3” または”regex4” のいずれかに一致するオブジェクトのみを検出する必要があります。式 ID”A”、”B”、”C”、”D” は任意に選択されています。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 3,
            "formula": "(A and B) and (C or D)",
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1",
                    "formulaid": "A"
                },
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex2",
                    "formulaid": "B"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex3",
                    "formulaid": "C"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex4",
                    "formulaid": "D"
                }
            ]
        }
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

カスタムクエリフィールドとヘッダーの使用

カスタムクエリフィールドとヘッダーを持つ LLD ルールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
        "name": "API HTTP agent",
        "key_": "api_discovery_rule",
        "delay": "5s",
        "url": "http://127.0.0.1?discoverer.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "elements",
                "value": "2"
            }
        ],
        "headers": [
            {
                "name": "X-Type",
                "value": "api"
            },
            {
                "name": "Authorization",
                "value": "Bearer mF_A.B5f-2.1JcM"
            }
        ],
        "allow_traps": 1,
        "trapper_hosts": "127.0.0.1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
```

```
        ]
    },
    "id": 1
}
```

前処理付き LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discovery rule with preprocessing",
        "key_": "lld.with.preprocessing",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 20,
                "params": "20",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

オーバーライド付き LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discover database host",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Discover MySQL host",
                "step": "1",
                "stop": "1",
                "filter": {
                    "evaltype": "2",
                    "conditions": [
```

```
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mysqld\\.service$"
                    },
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mariadb\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ],
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ]
                }
            ]
        },
        {
            "name": "Discover PostgreSQL host",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10263"
                        }
                    ],
                    "optag": [
```

```
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

スクリプト LLD ルールの作成

スクリプト LLD ルールを使用してシンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.lldrule",
        "hostid": "12345",
        "type": 21,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [{
            "name": "host",
            "value": "{HOST.CONN}"
        }],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

無効化期間を指定し、削除を行わない LLD ルールを作成します。

エンティティが検出されなくなった後に無効化する期間をカスタマイズし、削除を行わないように設定した LLD ルールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "lld disable after 1h",
        "key_": "lld.disable",
        "hostid": "10001",
        "type": 2,
        "lifetime_type": 1,
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "1h"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

関連項目

- LLD ルールフィルター
- LLD マクロパス
- LLD ルールの前処理

ソース

CDiscoveryRule::create() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

説明

`object discoveryrule.delete(array lldRuleIds)`

このメソッドを使用すると、LLD ルールを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 アクセスできないプロキシを使用してエンティティを検出する LLD ルールは削除できません。詳細については、ユーザーグループの設定（プロキシのアクセスリスト）を参照してください。

パラメーター

(array) 削除する LLD ルールの ID。

戻り値

(object) ruleids プロパティの下にある削除された LLD ルールの ID を含むオブジェクトを返します。

例

複数の LLD ルールの削除

2 つの LLD ルールを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRule::delete() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

説明

`integer/array discoveryrule.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って LLD ルールを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
itemids	ID/array	指定した ID の LLD ルールのみを返します。
groupids	ID/array	指定したグループのホストに属する LLD ルールのみを返します。
hostids	ID/array	指定したホストに属する LLD ルールのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承された LLD ルールのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用する LLD ルールのみを返します。
monitored	boolean	`true` に設定した場合、監視対象ホストに属する有効な LLD ルールのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属する LLD ルールのみを返します。
templateids	ID/array	指定したテンプレートに属する LLD ルールのみを返します。
selectDiscoveryData	query	LLD ルールのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。LLD ルールのディスカバリオブジェクトは、検出された LLD ルールを、その検出元となった LLD ルールのプロトタイプに関連付けます。
selectDiscoveryRulePrototypes	query	LLD ルールに属する LLD ルールのプロトタイプを含む`discoveryRulePrototypes` プロパティを返します。 `count` をサポートします。

パラメータ	型	説明
selectFilter	query	LLD ルールで使用されるフィルターのデータを含む`filter` プロパティを返します。
selectGraphs	query	LLD ルールに属するグラフのプロトタイプを含む`graphs` プロパティを返します。 `count` をサポートします。
selectHostPrototypes	query	LLD ルールに属するホストのプロトタイプを含む`hostPrototypes` プロパティを返します。 `count` をサポートします。
selectHosts	query	LLD ルールが属するホストの配列を含む`hosts` プロパティを返します。
selectItems	query	LLD ルールに属するアイテムのプロトタイプを含む`items` プロパティを返します。 `count` をサポートします。
selectTriggers	query	LLD ルールに属するトリガーのプロトタイプを含む`triggers` プロパティを返します。 `count` をサポートします。
selectLLDMacroPaths	query	各対応するマクロに割り当てられた値へのパスと LLD マクロの一覧を含む`lld_macro_paths` プロパティを返します。
selectPreprocessing	query	LLD ルールの前処理オプションを含む`preprocessing` プロパティを返します。
selectOverrides	query	プロトタイプオブジェクトに対して実行される上書きフィルター、条件、および操作の一覧を含む`lld_rule_overrides` プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしていません。 追加のプロパティをサポートします: `host` - LLD ルールが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectItems`, `selectGraphs`, `selectTriggers`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `itemid`, `name`, `key_`, `delay`, `type`, `status`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

ホストからのディスカバリルールの取得

特定のホスト ID のすべてのディスカバリルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Network interface discovery",
            "key_": "net.if.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22444",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Discovery of network interfaces as defined in global regular expression \"Netw
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
```

```
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        },
        {
            "itemid": "27426",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Mounted filesystem discovery",
            "key_": "vfs.fs.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22450",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Discovery of file systems of different types as defined in global regular expr
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


フィルター条件の取得

LLD ルール"24681" の名前とそのフィルター条件を取得します。フィルターは"and" 評価タイプを使用しているため、formula プロパティは空で、eval_formula は自動的に生成されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "24681",
            "name": "Filtered LLD rule",
            "filter": {
                "evaltype": "1",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#MACRO1}",
                        "value": "@regex1",
                        "operator": "8",
                        "formulaid": "A"
                    },
                    {
                        "macro": "{#MACRO2}",
                        "value": "@regex2",
                        "operator": "9",
                        "formulaid": "B"
                    },
                    {
                        "macro": "{#MACRO3}",
                        "value": "",
                        "operator": "12",
                        "formulaid": "C"
                    },
                    {
                        "macro": "{#MACRO4}",
                        "value": "",
                        "operator": "13",
                        "formulaid": "D"
                    }
                ],
                "eval_formula": "A and B and C and D"
            }
        }
    ],
    "id": 1
}
```

URL による LLD ルールの取得

ルールの URL フィールド値でホストの LLD ルールを取得します。LLD ルールに定義された URL 文字列の完全一致のみがサポートされます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28336",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10257",
            "name": "API HTTP agent",
            "key_": "api_discovery_rule",
            "delay": "5s",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "5",
            "description": "",
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "http://127.0.0.1/discoverer.php",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "json"
                },
                {
                    "name": "elements",
                    "value": "2"
                }
            ],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [
                {
                    "name" : "X-Type",
```

```
                    "value": "api"
                },
                {
                    "name": "Authorization",
                    "value": "Bearer mF_A.B5f-2.1JcM"
                }
            ],
            "retrieve_mode": "0",
            "request_method": "1",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```

オーバーライド付きの LLD ルールを取得

さまざまなオーバーライド設定を持つ 1 つの LLD ルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "Discover database host",
            "overrides": [
                {
                    "name": "Discover MySQL host",
                    "step": "1",
                    "stop": "1",
                    "filter": {
                        "evaltype": "2",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mysqld\\.service$",
```

```
                        "formulaid": "A"
                    },
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mariadb\\.service$",
                        "formulaid": "B"
                    }
                ],
                "eval_formula": "A or B"
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ],
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ]
                }
            ]
        },
        {
            "name": "Discover PostgreSQL host",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Database host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optag": [
                        {
                            "tag": "database",
                            "value": "postgresql"
```

```
                            }
                        ],
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ]
                    }
                ]
            }
        ]
    }
],
"id": 1
}
```


関連項目

- グラフのプロトタイプ
- ホスト
- アイテムのプロトタイプ
- LLD ルールフィルター
- トリガーのプロトタイプ

ソース

ui/include/classes/api/services/CDiscoveryRule.php の CDiscoveryRule::get()。

discoveryrule.update

説明

`object discoveryrule.update(object/array lldRules)`

このメソッドを使用すると、既存の LLD ルールを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー役割の設定で取り消すことができます。詳細はユーザー役割 を参照してください。

パラメータ

(`object/array`) 更新する LLD ルールのプロパティ。

各 LLD ルールでは `itemid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準の LLD ルールプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	既存のフィルターを置き換えるためのLLD ルールフィルター。
preprocessing	object/array	既存の前処理オプションを置き換えるためのLLD ルール前処理オプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
lld_macro_paths	object/array	既存の lld_macro_path オプションを置き換えるための、LLD ルールのlld_macro_pathオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*

パラメータ	型	説明
overrides	object/array	既存のオーバーライドオプションを置き換えるためのLLD ルールオーバーライドオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*

戻り値

(object) itemids プロパティの下で更新された LLD ルールの ID を含むオブジェクトを返します。

例

LLD ルールにフィルターを追加する

{#FSTYPE} マクロの内容が *@File systems for discovery* の正規表現に一致するように、フィルターを追加します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

LLD マクロパスの追加

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

トラッピングの無効化

ディスカバリールールの LLD トラッピングを無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

LLD ルールの前処理オプションの更新

前処理ルール「JSONPath」を使用して LLD ルールを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.path.to.json",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
```

```
            "44211"
        ]
    },
    "id": 1
}
```

LLD ルールスクリプトの更新

LLD ルールスクリプトを別のスクリプトで更新し、以前のスクリプトで使用されていた不要なパラメーターを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

LLD ルールの有効期間の更新

LLD ルールを更新し、検出されなくなったエンティティを 12 時間後に無効化し、7 日後に削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRule::update() in *ui/include/classes/api/services/CDiscoveryRule.php*.

LLD ルールのプロトタイプ

このクラスは、ローレベルディスカバリルールのプロトタイプを扱うために設計されています。

オブジェクト参照:

- LLD ルールのプロトタイプ
  - HTTP ヘッダー
  - HTTP クエリフィールド
- LLD ルールのプロトタイプフィルター
  - LLD ルールのプロトタイプフィルター条件
- LLD プロトタイプマクロパス
- LLD ルールのプロトタイプ前処理
- LLD ルールのプロトタイプ上書き
  - LLD ルールのプロトタイプ上書きフィルター
    - LLD ルールのプロトタイプ上書きフィルター条件
  - LLD ルールのプロトタイプ上書き操作
    - LLD ルールのプロトタイプ上書き操作ステータス
    - LLD ルールのプロトタイプ上書き操作ディスカバリ
    - LLD ルールのプロトタイプ上書き操作期間
    - LLD ルールのプロトタイプ上書き操作履歴
    - LLD ルールのプロトタイプ上書き操作トレンド
    - LLD ルールのプロトタイプ上書き操作深刻度
    - LLD ルールのプロトタイプ上書き操作タグ
    - LLD ルールのプロトタイプ上書き操作テンプレート
    - LLD ルールのプロトタイプ上書き操作インベントリ

利用可能なメソッド:

- discoveryruleprototype.create - 新しい LLD ルールのプロトタイプを作成
- discoveryruleprototype.delete - LLD ルールのプロトタイプを削除
- discoveryruleprototype.get - LLD ルールのプロトタイプを取得
- discoveryruleprototype.update - LLD ルールのプロトタイプを更新

LLD ルールプロトタイプオブジェクト

以下のオブジェクトは `discoveryruleprototype` API に直接関連しています。

LLD ルールのプロトタイプ

ローレベルディスカバリルールのプロトタイプオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
itemid	ID	LLD ルールのプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作で必須
ruleid	ID	親 LLD ルールまたは LLD ルールのプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 作成操作で必須

プロパティ	タイプ	説明
delay	string	LLD ルールのプロトタイプの更新間隔。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはオプションで 1 つ以上のカスタム間隔を指定できます。各値はセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロや、マクロとテキストを組み合わせた値はサポートされません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が「Zabbix agent」(0)、「Simple check」(3)、「Zabbix internal」(5)、「External check」(10)、「Database monitor」(11)、「IPMI agent」(12)、「SSH agent」(13)、「TELNET agent」(14)、「JMX agent」(16)、「HTTP agent」(19)、「SNMP agent」(20)、「Script」(21)、「Browser」(22) に設定されている場合、または `type` が「Zabbix agent (active)」(7) に設定され、`key_` に「mqtt.get」が含まれていない場合に必須
hostid	ID	LLD ルールのプロトタイプが属するホストの ID。 プロパティの動作: - 固定 - 作成操作で必須
flags	integer	LLD ルールのプロトタイプの起点。 指定可能な値: 3 - ローレベルディスカバリルールのプロトタイプ; 7 - 検出されたローレベルディスカバリルールのプロトタイプ。 プロパティの動作: - 読み取り専用
interfaceid	ID	LLD ルールのプロトタイプのホストインターフェースの ID。 プロパティの動作: - LLD ルールのプロトタイプがホストに属し、`type` が「Zabbix agent」、「IPMI agent」、「JMX agent」、または「SNMP agent」に設定されている場合に必須 - LLD ルールのプロトタイプがホストに属し、`type` が「Simple check」、「External check」、「SSH agent」、「TELNET agent」、または「HTTP agent」に設定されている場合にサポート
key_	string	LLD ルールのプロトタイプのキー。少なくとも 1 つの LLD マクロが必要です。 プロパティの動作: - 作成操作で必須 - 継承されたオブジェクトでは読み取り専用
name	string	LLD ルールのプロトタイプの名前。 プロパティの動作: - 作成操作で必須 - 継承されたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
type	integer	LLD ルールのプロトタイプのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested。 プロパティの動作: - 作成操作で必須 - 継承されたオブジェクトでは読み取り専用
url	string	URL 文字列。 {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} およびユーザーマクロをサポートします。 プロパティの動作: - type が「HTTP agent」に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用
allow_traps	integer	trapper アイテムと同様に値を格納できるようにします。 指定可能な値: 0 - *(デフォルト)* 受信データを受け付けない; 1 - 受信データを受け付ける。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート
authtype	integer	認証方式。 type が「SSH agent」に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が「HTTP agent」に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM。 プロパティの動作: - type が「SSH agent」または「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用（type が「HTTP agent」に設定されている場合）
description	string	LLD ルールのプロトタイプの説明。
error	string	LLD ルールのプロトタイプの値の更新中に障害が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
follow_redirects	integer	データのポーリング中にレスポンスのリダイレクトに従います。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
headers	array	HTTP リクエストの実行時に送信されるヘッダーの配列。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - `type` が「IPMI agent」に設定され、`key_` が「ipmi.get」に設定されていない場合に必須 - `type` が「IPMI agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - `type` が「JMX agent」に設定されている場合にサポート
lifetime	string	検出されなくなったアイテムが削除されるまでの期間。秒数（例: `1`、`60`、`300`）、時間のサフィックス付きの値（例: `30s`、`1m`、`2h`、`1d`）、またはユーザーマクロを指定できます。 デフォルト: `7d`。
lifetime_type	integer	失われた LLD リソースを削除するシナリオ。 指定可能な値: 0 - *(デフォルト)* 保存期間のしきい値に達した後に削除; 1 - 削除しない; 2 - 直ちに削除。
enabled_lifetime	string	検出されなくなったアイテムが無効になるまでの期間。秒数（例: `1`、`60`、`300`）、時間のサフィックス付きの値（例: `30s`、`1m`、`2h`、`1d`）、またはユーザーマクロを指定できます。 デフォルト: `0`。
enabled_lifetime_type	integer	失われた LLD リソースを無効にするシナリオ。 指定可能な値: 0 - 保存期間のしきい値に達した後に無効化; 1 - 無効化しない; 2 - *(デフォルト)* 直ちに無効化。
master_itemid	ID	マスターアイテムの ID。 ディスカバリルールを別のディスカバリルールのマスターアイテムにすることはできません。 プロパティの動作: - `type` が「Dependent item」に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データを保存; 1 - JSON に変換。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
params	string	LLD ルールのプロトタイプのタイプに応じた追加パラメータ: - SSH および Telnet の LLD ルールで実行されるスクリプト; - データベースモニターの LLD ルールの SQL クエリ; - 計算 LLD ルールの計算式; - スクリプトおよびブラウザの LLD ルールのスクリプト。 プロパティの動作: - `type` が「Database monitor」、「SSH agent」、「TELNET agent」、「Script」、または「Browser」に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用（`type` が「Script」または「Browser」に設定されている場合）
parameters	object/array	`type` が「Script」または「Browser」に設定されている場合の追加パラメータ。 `name` および `value` プロパティを持つオブジェクトの配列で、`name` は一意でなければなりません。 プロパティの動作: - `type` が「Script」または「Browser」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
password	string	認証用パスワード。 プロパティの動作: - `type` が「JMX agent」に設定され、`username` が設定されている場合に必須 - `type` が「Simple check」、「Database monitor」、「SSH agent」、「TELNET agent」、または「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用（`type` が「HTTP agent」に設定されている場合）
post_type	integer	`posts` プロパティに保存される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
posts	string	HTTP(S) リクエスト本文のデータ。 プロパティの動作: - `type` が「HTTP agent」に設定され、`post_type` が「JSON data」または「XML data」に設定されている場合に必須 - `type` が「HTTP agent」に設定され、`post_type` が「Raw data」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
privatekey	string	秘密鍵ファイルの名前。 プロパティの動作: - `type` が「SSH agent」に設定され、`authtype` が「public key」に設定されている場合に必須

プロパティ	タイプ	説明
publickey	string	公開鍵ファイルの名前。 プロパティの動作: - type が「SSH agent」に設定され、authtype が「public key」に設定されている場合に必須
query_fields	array	HTTP リクエストの実行時に送信されるクエリフィールドの配列。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
retrieve_mode	integer	保存するレスポンスの部分。 request_method が「GET」、「POST」、または「PUT」に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 request_method が「HEAD」に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - type が「SNMP agent」に設定されている場合に必須 - 継承されたオブジェクトでは読み取り専用
ssl_cert_file	string	公開 SSL 鍵ファイルのパス。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ssl_key_file	string	秘密 SSL 鍵ファイルのパス。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ssl_key_password	string	SSL 鍵ファイルのパスワード。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
state	integer	LLD ルールのプロトタイプの状態。 指定可能な値: 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	LLD ルールのプロトタイプのステータス。 指定可能な値: 0 - *(デフォルト)* 有効な LLD ルールのプロトタイプ; 1 - 無効な LLD ルールのプロトタイプ。
status_codes	string	必要な HTTP ステータスコードの範囲。コードはコンマで区切ります。コンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
templateid	ID	親テンプレートの LLD ルールのプロトタイプの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m）を指定できます。ユーザーマクロも使用できます。 指定可能な値の範囲: 1-600s。 デフォルト: "" - プロキシまたはグローバル設定を使用。 プロパティの動作: - type が「Zabbix agent」(0)、「Simple check」(3) で、key_ が「vmware.」および「icmpping」で始まらない場合、「Zabbix agent (active)」(7)、「External check」(10)、「Database monitor」(11)、「SSH agent」(13)、「TELNET agent」(14)、「HTTP agent」(19)、「SNMP agent」(20) で、snmp_oid が「walk[」または「get[」で始まる場合、「Script」(21)、または「Browser」(22) に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - type が「Zabbix trapper」に設定されている場合、または type が「HTTP agent」に設定され、allow_traps が「Allow incoming data」に設定されている場合にサポート。 このプロパティを省略すると、{$TRAPPER.ALLOWED_HOSTS}ユーザーマクロ（デフォルト値）が自動的に設定されます。デフォルトでは、{$TRAPPER.ALLOWED_HOSTS} は 127.0.0.1,::1 です。 プロパティに空の文字列を設定すると、すべてのホストからの受信接続が拒否されます。
username	string	認証用ユーザー名。 プロパティの動作: - type が「SSH agent」または「TELNET agent」に設定されている場合、または type が「JMX agent」に設定され、password が設定されている場合に必須 - type が「Simple check」、「Database monitor」、または「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用（type が「HTTP agent」に設定されている場合）

プロパティ	タイプ	説明
uuid	string	インポートした LLD ルールのプロトタイプを既存のプロトタイプにリンクするために使用される一意の識別子。指定されていない場合は自動生成されます。
verify_host	integer	プロパティの動作: - LLD ルールのプロトタイプがテンプレートに属している場合にサポート 接続先のホスト名がホストの証明書に記載された名前と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。
verify_peer	integer	プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用 ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

HTTP ヘッダー

ヘッダーオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

LLD ルールプロトタイプのフィルター

LLD ルールプロトタイプのフィルターオブジェクトは、検出されたオブジェクトをフィルタリングするために使用できる一連の条件を定義します。以下のプロパティがあります。

Property	Type	Description
conditions	object/array	結果のフィルタリングに使用するフィルター条件のセット。条件は、formula 内での配置順に並べ替えられます。 Property behavior: - 必須
evaltype	integer	フィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - 必須
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、各フィルター条件をその `formulaid` で参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 Property behavior: - 読み取り専用
formula	string	カスタム式を使用するフィルターの条件評価に使用する、ユーザー定義の式。この式には、各フィルター条件をその `formulaid` で参照する ID を含める必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致していなければなりません。未使用または省略された条件があってはなりません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は 必須

LLD ルールプロトタイプのフィルター条件

LLD ルールプロトタイプのフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。次のプロパティがあります。

プロパティ	型	説明
macro	string	チェックを実行する LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致する」または「正規表現に一致しない」に設定されている場合は 必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際は、この ID をユーザーが定義する必要がありますが、その後に条件をリクエストすると新たに生成されます。 プロパティの動作: - LLD ルールプロトタイプのフィルターオブジェクト の `evaltype` が「カスタム式」に設定されている場合は 必須

プロパティ	型	説明
operator	integer	条件演算子。 指定可能な値: 8 - *(デフォルト)* 正規表現に一致する; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、discoveryruleprototype.get および discoveryruleprototype.create メソッドのページにある例を参照してください。

LLD マクロパス

LLD マクロパスには、以下のプロパティがあります。

プロパティ	Type	説明
lld_macro	string	LLD マクロ。 プロパティの動作: - 必須
path	string	対応するマクロに割り当てられる値のセレクター。 プロパティの動作: - 必須

LLD ルールのプロトタイプの前処理

LLD ルールのプロトタイプの前処理オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
type	integer	前処理オプションのタイプ。 指定可能な値: 5 - 正規表現; 11 - XML XPath; 12 - JSONPath; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラーを確認; 17 - XML のエラーを確認; 20 - ハートビートで変更されていない値を破棄; 21 - JavaScript; 23 - Prometheus から JSON へ変換; 24 - CSV から JSON へ変換; 25 - 置換; 27 - XML から JSON へ変換; 28 - SNMP walk 値; 29 - SNMP walk から JSON へ変換; 30 - SNMP get 値。 プロパティの動作: - 必須

プロパティ	型	説明
params	string	前処理オプションで使用する追加パラメータ。複数のパラメータは改行 (\n) 文字で区切ります。 プロパティの動作: - type が「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーを確認」(16)、「XML のエラーを確認」(17)、「ハートビートで変更されていない値を破棄」(20)、「JavaScript」(21)、「Prometheus から JSON へ変換」(23)、「CSV から JSON へ変換」(24)、「置換」(25)、「SNMP walk 値」(28)、「SNMP walk から JSON へ変換」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler	integer	前処理ステップが失敗した場合に使用するアクションのタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定されます; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーを確認」(16)、「XML のエラーを確認」(17)、「Prometheus から JSON へ変換」(23)、「CSV から JSON へ変換」(24)、「XML から JSON へ変換」(27)、「SNMP walk 値」(28)、「SNMP walk から JSON へ変換」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler_params	string	エラーハンドラーのパラメータ。 プロパティの動作: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は 必須

各前処理タイプで、次のパラメータとエラーハンドラーがサポートされています。

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
5	正規表現	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	正規表現に一致	pattern[1]			0, 1, 2, 3
15	正規表現に一致しない	pattern[1]			0, 1, 2, 3
16	JSON のエラーを確認	path[3]			0, 1, 2, 3
17	XML のエラーを確認	path[3]			0, 1, 2, 3

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
20	ハートビートで変更されていない値を破棄	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus から JSON へ変換	pattern^5, 6^			
24	CSV から JSON へ変換	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
27	XML から JSON へ変換				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 へ 2 - Hex-STRING から MAC へ 3 - BITS から整数へ		0, 1, 2, 3
29	SNMP walk から JSON へ変換[7]	Field name[2]	OID prefix[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 へ 2 - Hex-STRING から MAC へ 3 - BITS から整数へ	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 へ 2 - Hex-STRING から MAC へ 3 - BITS から整数へ			0, 1, 2, 3

[1] 正規表現
 [2] 文字列
 [3] JSONPath または XML XPath
 [4] 正の整数（30s、1m、2h、1d などの時間のサフィックスをサポート）
 [5] ユーザーマクロ
 [6] 次の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。Prometheus パターンの各コンポーネント（メトリック、ラベル名、ラベル値、メトリック値）にはユーザーマクロを使用できます。
 [7] 改行文字で区切られた複数の「Field name,OID prefix,Format records」レコードをサポートします。

LLD ルールのプロトタイプ上書き

LLD ルールのプロトタイプ上書きオブジェクトは、異なるプロトタイプオブジェクトのプロパティを上書きするために使用される一連のルール（フィルター、条件、操作）を定義します。以下のプロパティがあります。

Property	Type	Description
name	string	一意の上書き名。 Property behavior: - *required*
step	integer	上書きの一意の順序番号。 Property behavior: - *required*
stop	integer	一致した場合、次の上書きの処理を停止します。 指定可能な値: 0 - *(default)* 上書きの処理を停止しない; 1 - フィルターに一致した場合、上書きの処理を停止する。
filter	object	上書きフィルター。
operations	object/array	上書き操作。

LLD ルールプロトタイプのオーバーライドフィルター

LLD ルールプロトタイプのオーバーライドフィルターオブジェクトは、検出されたオブジェクトに一致した場合にオーバーライドを適用する一連の条件を定義します。以下のプロパティがあります。

Property	Type	Description
conditions	object/array	検出されたオブジェクトとの一致に使用するオーバーライドフィルター条件のセット。条件は、formula 内で配置された順序でソートされます。 Property behavior: - 必須
evaltype	integer	オーバーライドフィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - 必須
eval_formula	string	オーバーライドフィルター条件の評価に使用される生成済みの式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 Property behavior: - 読み取り専用
formula	string	カスタム式を使用するオーバーライドフィルターの条件評価に使用する、ユーザー定義の式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID を含める必要があります。式で使用する ID は、オーバーライドフィルター条件で定義されたものと完全に一致していなければなりません。未使用または省略された条件があってはなりません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は必須

LLD ルールプロトタイプ上書きフィルター条件

LLD ルールプロトタイプ上書きフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。次のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は 必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件の変更時にはこの ID をユーザーが定義する必要がありますが、その後に取得をリクエストすると新たに生成されます。 プロパティの動作: - LLD ルールプロトタイプ上書きフィルターオブジェクト の `evaltype` が「カスタム式」に設定されている場合は 必須
operator	integer	条件演算子。 指定可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

LLD ルールのプロトタイプ上書き操作

LLD ルールのプロトタイプ上書き操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。以下のプロパティがあります。

プロパティ	Type	説明
operationobject	integer	アクションを実行する検出オブジェクトのタイプ。 指定可能な値: 0 - アイテムのプロトタイプ; 1 - トリガーのプロトタイプ; 2 - グラフのプロトタイプ; 3 - ホストのプロトタイプ。 プロパティの動作: - 必須
operator	integer	上書き条件のoperator。 指定可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 8 - 一致する; 9 - 一致しない。
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフ、またはホストのプロトタイプ名に一致させるパターン。
opstatus	object	アイテム、トリガー、およびホストのプロトタイプオブジェクト用の上書き操作ステータスオブジェクト。
opdiscover	object	上書き操作の検出ステータスオブジェクト（すべてのオブジェクトタイプ）。
opperiod	object	アイテムのプロトタイプオブジェクト用の上書き操作期間（更新間隔）オブジェクト。
ophistory	object	アイテムのプロトタイプオブジェクト用の上書き操作履歴オブジェクト。

プロパティ	Type	説明
optrends	object	アイテムのプロトタイプオブジェクト用の上書き操作トレンドオブジェクト。
opseverity	object	トリガーのプロトタイプオブジェクト用の上書き操作深刻度オブジェクト。
optag	object/array	トリガーおよびホストのプロトタイプオブジェクト用の上書き操作タグオブジェクト。
optemplate	object/array	ホストのプロトタイプオブジェクト用の上書き操作テンプレートオブジェクト。
opinventory	object	ホストのプロトタイプオブジェクト用の上書き操作インベントリオブジェクト。

LLD ルールのプロトタイプ上書き操作ステータス

検出されたオブジェクトに設定される LLD ルールのプロトタイプ上書き操作ステータスです。
以下のプロパティがあります。

Property	Type	Description
status	integer	選択したオブジェクトのステータスを上書きします。 指定可能な値: 0 - 有効で作成; 1 - 無効で作成。 プロパティの動作: *- required*

LLD ルールのプロトタイプ上書き操作のディスカバー

LLD ルールのプロトタイプ上書き操作のディスカバーステータスは、ディスカバー済みオブジェクトに設定されます。以下のプロパティがあります。

Property	Type	Description
discover	integer	選択したオブジェクトのディスカバーステータスを上書きします。 指定可能な値: 0 - はい、オブジェクトのディスカバリーを継続します。 1 - いいえ、新しいオブジェクトはディスカバーされず、既存のオブジェクトは失われたものとしてマークされます。 プロパティの動作: *- required*

LLD ルールプロトタイプのオーバーライド操作期間

LLD ルールプロトタイプのオーバーライド操作期間は、検出されたアイテムに設定される更新間隔の値です。以下のプロパティがあります。

Property	Type	Description
delay	string	アイテムプロトタイプの更新間隔をオーバーライドします。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはオプションで 1 つ以上のカスタム間隔を指定できます。これらはすべてセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロまたは LLD マクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはマクロとテキストを組み合わせた値はサポートされません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - 必須

LLD ルールのプロトタイプ上書き操作履歴

検出されたアイテムに設定される LLD ルールのプロトタイプ上書き操作履歴の値です。以下のプロパティがあります。

Property	Type	Description
history	string	アイテムプロトタイプの履歴を上書きします。これは履歴データをどのくらいの期間保存するかを示す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 Property behavior: - *required*

LLD ルールのプロトタイプ上書き操作のトレンド

LLD ルールのプロトタイプ上書き操作のトレンド値は、検出されたアイテムに設定されます。以下のプロパティがあります。

プロパティ	Type	説明
trends	string	アイテムプロトタイプのトレンドを上書きします。これは、トレンドデータを保存する期間を表す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - *required*

LLD ルールのプロトタイプ上書き操作の深刻度

検出されたトリガーに設定される、LLD ルールのプロトタイプ上書き操作の深刻度の値です。以下のプロパティがあります。

Property	Type	Description
severity	integer	トリガープロトタイプの深刻度を上書きします。 指定可能な値: 0 - *(default)* 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 プロパティの動作: - *required*

LLD ルールのプロトタイプ上書き操作タグ

LLD ルールのプロトタイプ上書き操作タグオブジェクトには、ディスカバリされたオブジェクトに設定されるタグ名と値が含まれます。次のプロパティがあります。

プロパティ	Type	説明
tag	string	新しいタグ名。 プロパティの動作: - 必須
value	string	新しいタグ値。

LLD ルールのプロトタイプ上書き操作テンプレート

検出されたホストにリンクされた、LLD ルールのプロトタイプ上書き操作テンプレートオブジェクトです。以下のプロパティがあります。

プロパティ	型	説明
templateid	ID	ホストプロトタイプのリンク済みテンプレートのテンプレートを上書きします。 プロパティの動作: - 必須

LLD ルールプロトタイプのオーバーライド操作インベントリ

検出されたホストに設定される、LLD ルールプロトタイプのオーバーライド操作インベントリモードの値です。以下のプロパティがあります。

Property	Type	Description
inventory_mode	integer	ホストプロトタイプのインベントリモードをオーバーライドします。 指定可能な値: -1 - 無効; 0 - *(デフォルト)* 手動; 1 - 自動。 プロパティの動作: - 必須

discoveryruleprototype.create

説明

`object discoveryruleprototype.create(object/array lldRules)`

このメソッドを使用すると、新しい LLD ルールのプロトタイプを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成する LLD ルールのプロトタイプ。

標準の LLD ルールプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	LLD ルールのLLD ルールプロトタイプフィルター。
preprocessing	object/array	LLD ルールプロトタイプの前処理オプション。
lld_macro_paths	object/array	LLD ルールプロトタイプのlld_macro_pathオプション。
overrides	object/array	LLD ルールプロトタイプのオーバーライドオプション。

戻り値

(`object`) 作成された LLD ルールプロトタイプの ID を `itemids` プロパティに含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールプロトタイプの順序と一致します。

例

LLD ルールプロトタイプの作成

データベースインスタンス内のテーブルスペースを検出するための LLD ルールプロトタイプ (Type: Nested) を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

関連項目

- LLD ルールのプロトタイプフィルター
- LLD マクロパス
- LLD ルールのプロトタイプ前処理

ソース

CDiscoveryRulePrototype::create() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*。

discoveryruleprototype.delete

説明

`object discoveryruleprototype.delete(array lldRuleIds)`

このメソッドを使用すると、LLD ルールプロトタイプを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(array) 削除する LLD ルールプロトタイプの ID。

戻り値

(object) 削除された LLD ルールプロトタイプの ID を `ruleids` プロパティで含むオブジェクトを返します。

例

複数の LLD ルールプロトタイプの削除

2 つの LLD ルールプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRulePrototype::delete() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*。

discoveryruleprototype.get

説明

`integer/array discoveryruleprototype.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って LLD ルールのプロトタイプを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。
> このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。
> 詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
discoveryids	ID/array	指定した LLD ルールまたは LLD ルールのプロトタイプに属する LLD ルールのプロトタイプのみを返します。
itemids	ID/array	指定した ID を持つ LLD ルールのプロトタイプのみを返します。
groupids	ID/array	指定したグループのホストに属する LLD ルールのプロトタイプのみを返します。
hostids	ID/array	指定したホストに属する LLD ルールのプロトタイプのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承された LLD ルールのプロトタイプのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用する LLD ルールのプロトタイプのみを返します。
monitored	boolean	`true` に設定した場合、監視中のホストに属する有効な LLD ルールのプロトタイプのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属する LLD ルールのプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属する LLD ルールのプロトタイプのみを返します。
selectDiscoveryRule	query	LLD ルールのプロトタイプの親 LLD ルールを含む`discoveryRule` プロパティを返します。 親がディスカバリルールのプロトタイプである場合は、空のレスポンスを返します。
selectDiscoveryRulePrototype	query	LLD ルールのプロトタイプの親 LLD ルールのプロトタイプを含む`discoveryRulePrototype` プロパティを返します。 親がディスカバリルール（プロトタイプではない）である場合は、空のレスポンスを返します。
selectDiscoveryRulePrototypes	query	LLD ルールのプロトタイプに属する子 LLD ルールのプロトタイプを含む`discoveryRulePrototypes` プロパティを返します。 `count` をサポートします。
selectFilter	query	LLD ルールのプロトタイプで使用されるフィルターのデータを含む`filter` プロパティを返します。
selectGraphs	query	LLD ルールのプロトタイプに属するグラフのプロトタイプを含む`graphs` プロパティを返します。 `count` をサポートします。
selectHostPrototypes	query	LLD ルールのプロトタイプに属するホストのプロトタイプを含む`hostPrototypes` プロパティを返します。 `count` をサポートします。
selectHosts	query	LLD ルールのプロトタイプが属するホストの配列を含む`hosts` プロパティを返します。
selectItems	query	LLD ルールのプロトタイプに属するアイテムのプロトタイプを含む`items` プロパティを返します。 `count` をサポートします。
selectTriggers	query	LLD ルールのプロトタイプに属するトリガーのプロトタイプを含む`triggers` プロパティを返します。 `count` をサポートします。
selectLLDMacroPaths	query	LLD マクロの一覧と、各対応するマクロに割り当てられた値へのパスを含む`lld_macro_paths` プロパティを返します。
selectPreprocessing	query	LLD ルールのプロトタイプの前処理オプションを含む`preprocessing` プロパティを返します。
selectOverrides	query	プロトタイプオブジェクトに対して実行される上書きフィルター、条件、および操作の一覧を含む`lld_rule_overrides` プロパティを返します。

パラメータ	型	説明
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしていません。 追加のプロパティをサポートします。 `host` - LLD ルールのプロトタイプが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectItems`, `selectGraphs`, `selectTriggers`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `itemid`, `name`, `key_`, `delay`, `type`, `status`。
countOutput	boolean	これらのパラメータはリファレンス解説 で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) は以下のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用された場合は、取得したオブジェクトの数

例

ホストからのディスカバリールールプロトタイプの取得

特定のホスト ID のすべてのディスカバリールールプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Discover tablespaces for {#DB}",
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
```

```
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


関連項目

- グラフのプロトタイプ
- ホスト
- アイテムのプロトタイプ
- LLD ルールのプロトタイプフィルター
- トリガーのプロトタイプ

ソース

ui/include/classes/api/services/CDiscoveryRulePrototype.php の CDiscoveryRulePrototype::get()。

discoveryruleprototype.update

説明

`object discoveryruleprototype.update(object/array lldRules)`

このメソッドを使用すると、既存の LLD ルールプロトタイプを更新できます。

すでに発見済みのプロトタイプの更新には制限があることに注意してください。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新する LLD ルールのプロトタイプのプロパティ。

各 LLD ルールのプロトタイプでは `itemid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

標準の LLD ルールプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	既存のフィルターを置き換えるためのLLD ルールプロトタイプのフィルター。
preprocessing	object/array	既存の preprocessing オプションを置き換えるためのLLD ルールプロトタイプの preprocessingオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
lld_macro_paths	object/array	既存の lld_macro_path オプションを置き換えるための、LLD ルールプロトタイプのlld_macro_pathオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
overrides	object/array	既存の overrides オプションを置き換えるためのLLD ルールプロトタイプの overridesオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*

戻り値

(`object`) `itemids` プロパティの下に、更新された LLD ルールプロトタイプの ID を含むオブジェクトを返します。

例

LLD ルールプロトタイプの事前処理オプションの更新

LLD ルールプロトタイプに JSONPath の事前処理ルールを追加します。リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRulePrototype::update() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*。

MFA

このクラスは、MFA(多要素認証) メソッドを操作するためのものです。

オブジェクトリファレンス:

- MFA

利用可能なメソッド:

- mfa.create - 新しい MFA メソッドを作成する
- mfa.delete - MFA メソッドを削除する
- mfa.get - MFA メソッドを取得する
- mfa.update - MFA メソッドを更新する

MFA オブジェクト

次のオブジェクトは、`mfa`API に直接関連しています。

MFA

MFA（多要素認証）オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
mfaid	ID	MFA 方式の ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
type	integer	MFA 方式のタイプ。 指定可能な値: 1 - TOTP（Time-based One-Time Passwords）; 2 - Duo Universal Prompt。
name	string	MFA 方式の一意な名前。 プロパティの動作: - 作成操作では 必須
hash_function	integer	TOTP コードの生成に使用するハッシュ関数のタイプ。 指定可能な値: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512。 プロパティの動作: - `type` が”TOTP” に設定されている場合は 必須

プロパティ	Type	説明
code_length	integer	認証コードの長さ。 指定可能な値: 6 - 6 桁; 8 - 8 桁。 プロパティの動作: - type が”TOTP” に設定されている場合は 必須
api_hostname	string	Duo 認証サービスによって提供される API ホスト名。 プロパティの動作: - type が”Duo Universal Prompt” に設定されている場合は 必須
clientid	string	Duo 認証サービスによって提供されるクライアント ID。 プロパティの動作: - type が”Duo Universal Prompt” に設定されている場合は 必須
client_secret	string	Duo 認証サービスによって提供されるクライアントシークレット。 プロパティの動作: - 書き込み専用 - type が”Duo Universal Prompt” に設定されている場合は 必須

mfa.create

説明

`object mfa.create(object/array MFA methods)`

このメソッドを使用すると、新しい MFA メソッドを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受けつけます。

戻り値

(`object`) `mfaids` プロパティの下で作成された MFA メソッドの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムの順序と一致します。

例

MFA メソッドの作成

時間ベースのワンタイムパスワード (TOTP) を利用した”Zabbix TOTP” MFA メソッドを作成します。TOTP コード生成用のハッシュ関数は SHA-1 に設定し、検証コードの長さは 6 桁に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Response:


```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```



```
{
"jsonrpc": "2.0",
"method": "mfa.create",
"params": {
"type": 1,
"name": "Zabbix TOTP",
"hash_function": 1,
"code_length": 6
},
"id": 1
}
```


レスポンス:


```
{
"jsonrpc": "2.0",
"result": {
"mfaids": [
"1"
]
},
"id": 1
}
```


ソース

CMfa::create() in *ui/include/classes/api/services/CMfa.php*.

mfa.delete

説明

`object mfa.delete(array mfaids)`

このメソッドを使用すると、MFA メソッドを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`array`) 削除する MFA メソッドの ID。

戻り値

(`object`) `mfaids` プロパティの下にある削除された MFA メソッドの ID を含むオブジェクトを返します。

例

MFA メソッドの削除

MFA メソッドを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CMfa::delete() in *ui/include/classes/api/services/CMfa.php*.

mfa.get

説明

`integer/array mfa.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って MFA メソッドを取得できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
mfaids	ID/array	指定された ID を持つ MFA メソッドのみを返します。
selectUsrgrps	query	MFA メソッドに関連付けられたユーザーグループ を含む usrgrps プロパティを返します。 count をサポートします。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 サポートされるプロパティ: mfaid - MFA メソッドの ID。 type - MFA メソッドのタイプ。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: name。

パラメータ	型	説明
search	object	指定されたパターンに一致する結果を返します（大文字と小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。 サポートされるプロパティ: name。
countOutput	boolean	これらのパラメータについては、リファレンスコメント を参照してください。
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

名前による MFA メソッドの検索

名前に”Zabbix” が含まれるすべての MFA メソッドを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mfaid": "1",
            "type": "1",
            "name": "Zabbix TOTP 1",
            "hash_function": "1",
            "code_length": "6",
            "api_hostname": "",
            "clientid": ""
        },
        {
            "mfaid": "2",
            "type": "1",
            "name": "Zabbix TOTP 2",
            "hash_function": "3",
```

```
            "code_length": "8",
            "api_hostname": "",
            "clientid": ""
        }
    ],
    "id": 1
}
```

ソース

CMfa::get() in *ui/include/classes/api/services/CMfa.php*.

mfa.update

説明

`object mfa.update(object/array MFA methods)`

このメソッドを使用すると、既存の MFA メソッドを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新する MFA メソッドのプロパティ

`mfaid` プロパティは各項目に対して定義する必要があります。その他のプロパティはオプションです。渡されたプロパティのみが更新され、その他のプロパティは変更されません。

このメソッドは、標準 MFA メソッドのプロパティ を持つ MFA メソッドを受けつけます。

戻り値

(object) `mfaids` プロパティの下にある更新された MFA メソッドの ID を含むオブジェクトを返します。

例

Updating method properties

メソッドプロパティの更新

時間ベースのワンタイムパスワード（TOTP）を利用した”Zabbix TOTP” MFA メソッドの TOTP コード生成用ハッシュ関数と検証コードの長さを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CMfa::update() in *ui/include/classes/api/services/CMfa.php*.

Module

このクラスは、Web コンソールのモジュールを操作するためのものです。

オブジェクトリファレンス:

- Module

利用可能なメソッド:

- module.create - 新しいモジュールをインストールする
- module.delete - モジュールをアンインストールする
- module.get - モジュールを取得する
- module.update - モジュールを更新する

Module オブジェクト

次のオブジェクトは、moduleAPI に直接関連しています。

モジュール

module オブジェクトには、次のプロパティがあります。

Property	Type	Description
moduleid	ID	データベースに保存されている module の ID。 Property behavior: - *read-only* - 更新操作では *required*
id	string	module の`manifest.json` ファイルで開発者が定義した一意の module ID。 組み込み module の可能な値については、Dashboard widget の property "type" の説明を参照してください。 Property behavior: - 作成操作では *required*
relative_path	string	Zabbix Web インターフェースのディレクトリから見た、module ディレクトリへの相対パス。 可能な値: `widgets/*` - 組み込み widget module 用; `modules/*` - サードパーティ製 module 用。 Property behavior: - 作成操作では *required*
status	integer	module が有効か無効かを示します。 可能な値: 0 - *(default)* 無効; 1 - 有効。
config	object	Module configuration。

module.create

説明

`object module.create(object/array modules)`

このメソッドを使用すると、新しい Web インターフェースモジュールをインストールできます。

Note:
このメソッドは、特権管理者ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['modules_config_enabled']` が `false` に設定されている場合、ユーザーはこのメソッドを呼び出せません。

Attention:
モジュールファイルは、モジュールの `relative_path` プロパティに対応する正しいサブディレクトリに手動で展開する必要があります。

パラメータ

(object/array) 作成するモジュール。

このメソッドは、標準モジュールプロパティを持つモジュールを受け入れます。

戻り値

(object) は、moduleids プロパティにインストールされているモジュールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたモジュールの順序と一致します。

例

モジュールのインストール

モジュールをステータス" 有効" でインストールします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

参照

- Module
- Frontend modules

ソース

CModule::create() in *ui/include/classes/api/services/CModule.php*.

module.delete

説明

`object module.delete(array moduleids)`

このメソッドを使用すると、モジュールをアンインストールできます。

Note:
このメソッドは、特権管理者ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['modules_config_enabled']` が `false` に設定されている場合、ユーザーはこのメソッドを呼び出すことができません。

Attention:
モジュールファイルは手動で削除する必要があります。

パラメーター

(array) アンインストールするモジュールの ID。

戻り値

(object) moduleids プロパティの下にあるアンインストールされたモジュールの ID を含むオブジェクトを返します。

例

複数のモジュールのアンインストール

モジュール"27" と"28" をアンインストールします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

ソース

CModule::delete() in *ui/include/classes/api/services/CModule.php*.

module.get

説明

`integer/array module.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってモジュールを取得できます。

Note:
このメソッドは スーパーユーザーユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェース設定ファイルで `$ZBX_FEATURE_FLAGS['modules_config_enabled']` が `false` に設定されている場合、ユーザーはこのメソッドを呼び出せません。

パラメータ

(object) 希望する出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
moduleids	ID/array	指定された ID を持つモジュールのみを返します。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: moduleid, relative_path。
countOutput	boolean	これらのパラメータについては、Reference commentary を参照してください。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ID でモジュールを取得する

モジュール"1"、"2"、"25" に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "moduleid": "1",
            "id": "actionlog",
            "relative_path": "widgets/actionlog",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "2",
            "id": "clock",
```

```
            "relative_path": "widgets/clock",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "25",
            "id": "example",
            "relative_path": "modules/example_module",
            "status": "1",
            "config": []
        }
    ],
    "id": 1
}
```

参照

- Module
- Dashboard widget
- Frontend modules

ソース

CModule::get() in *ui/include/classes/api/services/CModule.php*.

module.update

説明

`object module.update(object/array modules)`

このメソッドを使用すると、既存のモジュールを更新できます。

> **Note:**
> このメソッドは、特権管理者ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェースの設定ファイルで `$ZBX_FEATURE_FLAGS['modules_config_enabled']` が `false` に設定されている場合、ユーザーはこのメソッドを呼び出すことができません。

パラメータ

(object/array) 更新されるトークンプロパティ

`tokenid` プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) `roleids` プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

モジュールの無効化

モジュール"25" を無効化します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

参照

- Module
- Frontend modules

ソース

CModule::update() in *ui/include/classes/api/services/CModule.php*.

Proxy

このクラスは、プロキシを操作するためのものです。

オブジェクトリファレンス:

- Proxy

利用可能なメソッド:

- proxy.create - 新しいプロキシを作成する
- proxy.delete - プロキシを削除する
- proxy.get - プロキシを取得する
- proxy.update - プロキシを更新する

Proxy オブジェクト

次のオブジェクトは、proxyAPI に直接関連しています。

プロキシ

プロキシオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
proxyid	ID	プロキシの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	プロキシの名前。 プロパティの動作: - 作成操作では必須
proxy_groupid	ID	プロキシグループの ID。 0 - プロキシはどのプロキシグループにも割り当てられていません。
local_address	string	アクティブエージェント用のアドレス。接続先の IP アドレスまたは DNS 名。 プロパティの動作: - proxy_groupid が 0 でない場合は必須

プロパティ	タイプ	説明
local_port	string	接続先のローカルプロキシポート番号。 ユーザーマクロがサポートされています。 デフォルト: 10051。 プロパティの動作: - proxy_groupid が 0 でない場合にサポート
operating_mode	integer	プロキシのタイプ。 指定可能な値: 0 - アクティブプロキシ; 1 - パッシブプロキシ。 プロパティの動作: - 作成操作では必須
description	text	プロキシの説明。
lastaccess	timestamp	プロキシが最後にサーバーへ接続した時刻。 プロパティの動作: - 読み取り専用
address	string	接続先の IP アドレスまたは DNS 名。 ユーザーマクロがサポートされています。 プロパティの動作: - Zabbix プロキシの動作モードがパッシブの場合は必須
port	string	接続先のポート番号。 ユーザーマクロがサポートされています。 デフォルト: 10051。 プロパティの動作: - Zabbix プロキシの動作モードがパッシブの場合にサポート
allowed_addresses	string	アクティブな Zabbix プロキシの IP アドレスまたは DNS 名をコンマ区切りで指定します。
tls_connect	integer	プロキシへの接続。 指定可能な値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。
tls_accept	integer	プロキシからの接続。 指定可能なビットマップ値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。 これはビットマスクフィールドです。指定可能なビットマップ値の合計を指定できます（例: PSK と証明書の場合は 6）。
tls_issuer	string	証明書の発行者。
tls_subject	string	証明書のサブジェクト。

プロパティ	タイプ	説明
tls_psk_identity	string	PSK ID。1 つの PSK とのみ組み合わせる必要があります（自動登録、ホスト、プロキシ全体で）。 PSK ID には機密情報を含めないでください。使用する PSK を受信側に通知するため、PSK ID は暗号化されずにネットワーク経由で送信されます。 プロパティの動作: - 書き込み専用 - `tls_connect` が「PSK」に設定されている場合、または `tls_accept` に「PSK」ビットが含まれている場合は必須
tls_psk	string	事前共有鍵（PSK）。32 桁以上の 16 進数である必要があります。 プロパティの動作: - 書き込み専用 - `tls_connect` が「PSK」に設定されている場合、または `tls_accept` に「PSK」ビットが含まれている場合は必須
custom_timeouts	integer	プロキシレベルでグローバルなアイテムのタイムアウトを上書きするかどうか。 指定可能な値: 0 - *(デフォルト)* グローバル設定を使用; 1 - タイムアウトを上書き。
timeout_zabbix_agent	string	処理に `timeout_*` 秒を超える時間を費やさないようにします。 1、60、300 などの秒数、または時間のサフィックス付きの値（30s、1m など）を指定できます。ユーザーマクロも使用できます。 `custom_timeouts` が「タイムアウトを上書き」に設定されている場合にのみ使用されます。それ以外の場合、この値は無視されます。 指定可能な値の範囲: 1～600 秒。 プロパティの動作: - `custom_timeouts` が「タイムアウトを上書き」に設定されている場合は必須
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		
version	integer	プロキシのバージョン。 3 つの部分で構成される Zabbix バージョン番号。各部分には 2 桁の 10 進数を使用します。例: バージョン 6.4.1 は 60401、バージョン 7.0.2 は 70002 など。 0 - 不明なプロキシバージョン。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
compatibility	integer	Zabbix サーバーのバージョンと比較したプロキシのバージョン。 指定可能な値: 0 - 未定義; 1 - 現行バージョン（プロキシとサーバーのメジャーバージョンが同じ）; 2 - 古いバージョン（プロキシのバージョンがサーバーより古いが、一部サポートされている）; 3 - サポート対象外のバージョン（プロキシのバージョンがサーバーの以前の LTS リリースバージョンより古い、またはサーバーのメジャーバージョンがプロキシのメジャーバージョンより古い）。 プロパティの動作: - 読み取り専用
state	integer	プロキシの状態。 指定可能な値: 0 - 不明; 1 - オフライン; 2 - オンライン。 プロパティの動作: - 読み取り専用

アプリケーションパフォーマンス監視（APM）

APM オブジェクトには、次のプロパティがあります。

Property	Type	Description
data_collection_status	integer	プロキシでテレメトリデータの収集が有効かどうか。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
max_messages_per_second	integer	プロキシが 1 秒あたりに受け入れるメッセージの最大数。 指定可能な値の範囲: 0（無制限）– 2147483647。 デフォルト: 0。 プロパティの動作: - `data_collection_status` が「Enabled」に設定されている場合は *supported*
additional_resource_attributes	array	受信テレメトリデータに追加するリソース属性の配列。 プロパティの動作: - `data_collection_status` が「Enabled」に設定されている場合は *supported*

リソース属性

リソース属性オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
key	string	リソース属性名。 指定した `signal_type` に対して一意である必要があります。 プロパティの動作: - 必須

プロパティ	型	説明
value	string	リソース属性値。 空にすることができます。
signal_type	integer	属性が適用されるテレメトリデータの種類。 指定可能な値: 0 - トレース; 1 - メトリクス; 2 - ログ。 プロパティの動作: - 必須

proxy.create

説明

`object proxy.create(object/array proxies)`

このメソッドを使用すると、新しいプロキシを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するプロキシ。

標準のプロキシプロパティに加えて、このメソッドでは次のパラメータを受け入れます。

パラメータ	タイプ	説明
hosts	array	プロキシによって監視されるホスト。 ホストがすでに別のプロキシによって監視されている場合は、現在のプロキシに再割り当てされます。 ホストには `hostid` プロパティのみを定義する必要があります。
apm	object	プロキシのアプリケーションパフォーマンス監視（APM）設定。

戻り値

(object) `proxyids` プロパティの下に作成されたプロキシの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシの順序と一致します。

例

アクティブプロキシの作成

アクションプロキシ”Active proxy” を作成し、それによって監視されるホストを割り当てます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

パッシブプロキシの作成

パッシブプロキシ”Passive proxy” を作成し、それによって監視される 2 つのホストを割り当てます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Passive proxy",
        "operating_mode": "1",
        "address": "127.0.0.1",
        "port": "10051",
        "hosts": [
            {
                "hostid": "10192"
            },
            {
                "hostid": "10139"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
        ]
    },
    "id": 1
}
```

プロキシを作成して、プロキシグループへ追加

アクティブプロキシ”Active proxy” を作成し、ID”1” のプロキシグループに追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

証明書ベースの暗号化を使用したプロキシの作成

証明書ベースの暗号化を使用してアクティブプロキシを作成します。このプロキシはホスト"10785"を監視するように設定されています。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
    },
    "id": 1
}
```

APM を有効にしたプロキシの作成

APM を有効にしたアクティブプロキシ「Active proxy」を作成します。
このプロキシは、1 秒あたり最大 1000 件のメッセージを受け付け、トレースに `deployment.environment` リソース属性を追加し、メトリクスに `cloud.region` リソース属性を追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
```

```
            "max_messages_per_second": "1000",
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト
- プロキシグループ

ソース

CProxy::create() in *ui/include/classes/api/services/CProxy.php*.

proxy.delete

説明

`object proxy.delete(array proxies)`

このメソッドを使用すると、プロキシを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限ロールの設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するプロキシの ID。

戻り値

(object) 削除されたプロキシの ID を `proxyids` プロパティに含むオブジェクトを返します。

例

複数のプロキシの削除

2 つのプロキシを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

ソース

CProxy::delete() in *ui/include/classes/api/services/CProxy.php*.

proxy.get

説明

`integer/array proxy.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってプロキシを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが使用できます。メソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 ユーザーのuser groupsの *Proxy access list* で許可されているプロキシのみが返されます。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは、以下のパラメータをサポートしています。

パラメータ	型	説明
proxyids	ID/array	指定した ID を持つプロキシのみを返します。
proxy_groupids	ID/array	指定したプロキシグループに属するプロキシのみを返します。
selectApm	query	プロキシのアプリケーションパフォーマンス監視（APM）設定を含むapmプロパティを返します。
selectAssignedHosts	query	プロキシに割り当てられたホストを含むassignedHostsプロパティを返します。 count をサポートしています。
selectHosts	query	プロキシによって監視されているホストを含むhostsプロパティを返します。 count をサポートしています。
selectProxyGroup	query	プロキシグループオブジェクトを含むproxyGroupプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `proxyid`、`name`、`operating_mode`。
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明しています。
editable	boolean	
excludeSearch	boolean	

パラメータ	型	説明
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

すべてのプロキシを取得する

設定されているすべてのプロキシとそのインターフェースを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "11",
            "name": "Active proxy",
            "proxy_groupid": "0",
            "local_address": "",
            "local_port": "10051",
            "operating_mode": "0",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "0",
            "timeout_zabbix_agent": "",
            "timeout_simple_check": "",
            "timeout_snmp_agent": "",
            "timeout_external_check": "",
            "timeout_db_monitor": "",
            "timeout_http_agent": "",
            "timeout_ssh_agent": "",
            "timeout_telnet_agent": "",
            "timeout_script": "",
            "last_access": "1693391880",
```

```
            "version": "70000",
            "compatibility": "1",
            "state": "1"
        },
        {
            "proxyid": "12",
            "name": "Passive proxy",
            "proxy_groupid": "1",
            "local_address": "127.0.0.1",
            "local_port": "10051",
            "operating_mode": "1",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "1",
            "timeout_zabbix_agent": "5s",
            "timeout_simple_check": "5s",
            "timeout_snmp_agent": "5s",
            "timeout_external_check": "5s",
            "timeout_db_monitor": "5s",
            "timeout_http_agent": "5s",
            "timeout_ssh_agent": "5s",
            "timeout_telnet_agent": "5s",
            "timeout_script": "5s",
            "lastaccess": "1693391875",
            "version": "60400",
            "compatibility": "2",
            "state": "2"
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- プロキシグループ

ソース

CProxy::get() in *ui/include/classes/api/services/CProxy.php*.

proxy.update

説明

`object proxy.update(object/array proxies)`

このメソッドを使用すると、既存のプロキシを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するプロキシのプロパティ。

各プロキシに `proxyid` プロパティを定義する必要があります。その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外のプロパティは変更されません。

標準のプロキシプロパティに加えて、このメソッドでは以下のパラメータを受け付けます。

Parameter	Type	Description
hosts	array	プロキシによって監視されるホスト。 ホストがすでに別のプロキシによって監視されている場合、そのホストは現在のプロキシに再割り当てされます。 ホストには hostid プロパティのみを定義する必要があります。
apm	object	プロキシのアプリケーションパフォーマンス監視（APM）設定。

戻り値

(object) proxyids プロパティに更新されたプロキシの ID を含むオブジェクトを返します。

例

プロキシが監視するホストの変更

指定された 2 つのホストを監視するようにプロキシを更新します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            },
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

プロキシステータスの変更

プロキシをアクティブプロキシに変更し、名前を”Active proxy” に変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

プロキシをプロキシグループへ追加

ID”5” のプロキシを更新し、ID””1 のプロキシループに追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト
- プロキシグループ

ソース

CProxy::update() in *ui/include/classes/api/services/CProxy.php*.

Settings

このクラスは、一般的な管理設定を操作するためのものです。

オブジェクトリファレンス:

- Settings

利用可能なメソッド:

- settings.get - 設定を取得する
- settings.update - 設定を更新する

Settings オブジェクト

次のオブジェクトは、`settings` API に直接関連しています。

設定

settings オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
default_lang	string	デフォルトのシステム言語。 デフォルト: en_US。
default_timezone	string	デフォルトのシステムタイムゾーン。 デフォルト: system - システムのデフォルト。 サポートされているタイムゾーンの一覧については、PHP のドキュメントを参照してください。
default_theme	string	デフォルトのテーマ。 指定可能な値: blue-theme - *(デフォルト)* 青; blue-classic-theme - 青（クラシック）; dark-theme - ダーク; dark-blue-theme - ダークブルー; dark-classic-theme - ダーク（クラシック）; hc-light - ハイコントラスト（ライト）; hc-dark - ハイコントラスト（ダーク）。
search_limit	integer	検索およびフィルター結果の上限。 デフォルト: 1000。
max_overview_table_size	integer	Data overview および Trigger overview ダッシュボードウィジェットに表示する列と行の最大数。 デフォルト: 50。
max_in_table	integer	テーブルセル内に表示する要素の最大数。 デフォルト: 50。
server_check_interval	integer	Zabbix サーバーが停止している場合に警告を表示します。 指定可能な値: 0 - 警告を表示しない; 10 - *(デフォルト)* 警告を表示する。
work_period	string	稼働時間。 デフォルト: 1-5,09:00-18:00。
show_technical_errors	integer	非 Super admin ユーザー、およびデバッグモードが有効なユーザーグループに所属していないユーザーに、技術的なエラー（PHP/SQL）を表示します。 指定可能な値: 0 - *(デフォルト)* 技術的なエラーを表示しない; 1 - 技術的なエラーを表示する。
history_period	string	Latest data、Web、および Data overview ダッシュボードウィジェットに履歴データを表示する最大期間。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。 デフォルト: 24h。
period_default	string	時間フィルターのデフォルト期間。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d、1M、1y）を指定できます。月および年にも対応しています。 デフォルト: 1h。
max_period	string	時間フィルターの最大期間。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d、1M、1y）を指定できます。月および年にも対応しています。 デフォルト: 2y。

プロパティ	タイプ	説明
severity_color_0	string	”Not classified” の深刻度の色（16 進数カラーコード）。 デフォルト: 97AAB3。
severity_color_1	string	”Information” の深刻度の色（16 進数カラーコード）。 デフォルト: 7499FF。
severity_color_2	string	”Warning” の深刻度の色（16 進数カラーコード）。 デフォルト: FFC859。
severity_color_3	string	”Average” の深刻度の色（16 進数カラーコード）。 デフォルト: FFA059。
severity_color_4	string	”High” の深刻度の色（16 進数カラーコード）。 デフォルト: E97659。
severity_color_5	string	”Disaster” の深刻度の色（16 進数カラーコード）。 デフォルト: E45959。
severity_name_0	string	”Not classified” の深刻度名。 デフォルト: Not classified。
severity_name_1	string	”Information” の深刻度名。 デフォルト: Information。
severity_name_2	string	”Warning” の深刻度名。 デフォルト: Warning。
severity_name_3	string	”Average” の深刻度名。 デフォルト: Average。
severity_name_4	string	”High” の深刻度名。 デフォルト: High。
severity_name_5	string	”Disaster” の深刻度名。 デフォルト: Disaster。
custom_color	integer	カスタムイベントステータスカラーを使用します。 指定可能な値: 0 - *(デフォルト)* カスタムイベントステータスカラーを使用しない; 1 - カスタムイベントステータスカラーを使用する。
ok_period	string	OK トリガーの期間を表示します。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h）を指定できます。 デフォルト: 5m。
blink_period	string	トリガーの状態が変化したときの点滅期間。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h）を指定できます。 デフォルト: 2m。
problem_unack_color	string	未確認の PROBLEM イベントの色（16 進数カラーコード）。 デフォルト: CC0000。
problem_ack_color	string	確認済みの PROBLEM イベントの色（16 進数カラーコード）。 デフォルト: CC0000。
ok_unack_color	string	未確認の RESOLVED イベントの色（16 進数カラーコード）。 デフォルト: 009900。

プロパティ	タイプ	説明
ok_ack_color	string	確認済みの RESOLVED イベントの色（16 進数カラーコード）。 デフォルト: 009900。
problem_unack_style	integer	未確認の PROBLEM イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
problem_ack_style	integer	確認済みの PROBLEM イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
ok_unack_style	integer	未確認の RESOLVED イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
ok_ack_style	integer	確認済みの RESOLVED イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
url	string	Web インターフェースの URL。
discovery_groupid	ID	自動的に検出されたホストを配置するホストグループの ID。
default_inventory_mode	integer	デフォルトのホストインベントリモード。 指定可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
alert_usrgrpid	ID	データベース停止アラームメッセージの送信先ユーザーグループの ID。 ”0” に設定した場合、アラームメッセージは送信されません。
snmptrap_logging	integer	一致しない SNMP トラップをログに記録します。 指定可能な値: 0 - 一致しない SNMP トラップをログに記録しない; 1 - *(デフォルト)* 一致しない SNMP トラップをログに記録する。
login_attempts	integer	ログインフォームがブロックされるまでのログイン失敗回数。 デフォルト: 5。
login_block	string	login_attempts フィールドで定義されたログイン失敗回数を超えた場合に、ログインフォームがブロックされる時間。 秒数（例: 1、60）、または時間のサフィックス付きの値（例: 30s、1m）を指定できます。 デフォルト: 30s。
validate_uri_schemes	integer	URI スキームを検証します。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
uri_valid_schemes	string	有効な URI スキーム。 デフォルト: http,https,ftp,file,mailto,tel,ssh。
x_frame_options	string	X-Frame-Options HTTP ヘッダー。 デフォルト: SAMEORIGIN。

プロパティ	タイプ	説明
iframe_sandboxing_enabled	integer	iframe サンドボックスを使用します。 指定可能な値: 0 - 使用しない; 1 - (デフォルト) 使用する。
iframe_sandboxing_exceptions	string	iframe サンドボックスの例外。
connect_timeout	string	Zabbix サーバーへの接続タイムアウト。 指定可能な値の範囲: 1-30s。 デフォルト: 3s。 プロパティの動作: - 必須
socket_timeout	string	ネットワークのデフォルトタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 3s。 プロパティの動作: - 必須
media_type_test_timeout	string	メディアタイプのテストに使用するネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 65s。 プロパティの動作: - 必須
item_test_timeout	string	アイテムのテストに使用するネットワークタイムアウト。 指定可能な値の範囲: 1-600s。 デフォルト: 60s。 プロパティの動作: - 必須
script_timeout	string	スクリプト実行に使用するネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 プロパティの動作: - 必須
report_test_timeout	string	スケジュールレポートのテストに使用するネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 プロパティの動作: - 必須

プロパティ	タイプ	説明
device_link_timeout	string	Zabbix モバイルアプリで使用するモバイルデバイスの追加時のタイムアウト。Zabbix API とブリッジアダプター間の通信に適用されます。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 プロパティの動作: - 必須
auditlog_enabled	integer	監査ログを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
auditlog_mode	integer	サーバー（System ユーザー）が実行するローレベルディスカバリ、ネットワークディスカバリ、自動登録のアクティビティについて監査ログを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
ha_failover_delay	string	フェイルオーバーの遅延時間（秒）。 デフォルト: 1m。 プロパティの動作: - 読み取り専用
geomaps_tile_provider	string	ジオマップのタイルプロバイダー。 指定可能な値: `OpenStreetMap.Mapnik` - *(デフォルト)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery。
geomaps_tile_url	string	空の文字列を指定すると、`geomaps_tile_url`、`geomaps_max_zoom`、`geomaps_attribution` にカスタム値を指定できます。 ジオマップのタイル URL。 プロパティの動作: - `geomaps_tile_provider` が空の文字列に設定されている場合にサポート
geomaps_max_zoom	integer	ジオマップの最大ズームレベル。 指定可能な値: 0-30。 プロパティの動作: - `geomaps_tile_provider` が空の文字列に設定されている場合にサポート
geomaps_attribution	string	ジオマップの帰属表示テキスト。 プロパティの動作: - `geomaps_tile_provider` が空の文字列に設定されている場合にサポート
vault_provider	integer	Vault プロバイダー。 指定可能な値: 0 - *(デフォルト)* HashiCorp Vault; 1 - CyberArk Vault。

プロパティ	タイプ	説明
proxy_secrets_provider	integer	シークレットマクロの値を解決する方法: 指定可能な値: 0 - *(デフォルト)* サーバーのみ; 1 - サーバーとプロキシで個別に解決。
timeout_zabbix_agent	string	処理に使用する時間を `timeout_*` 秒以内に制限します。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m）を指定できます。ユーザーマクロも指定できます。 指定可能な値の範囲: 1-600s。 デフォルト: 3s。 `timeout_browser` のデフォルト: 60s。 プロパティの動作: - 必須
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		
apm_global_db	object	グローバルデータソースの設定。

グローバルデータソース

グローバルデータソースオブジェクトには、次のプロパティがあります。

Property	Type	Description
status	integer	グローバルデータソースのステータス。 指定可能な値: 0 - *(デフォルト)* 未設定; 1 - 設定済み。
url	string	データソースへの接続に使用する URL。 Property behavior: - ユーザーが `status` を `1`（設定済み）に変更する場合は必須。
authentication_type	integer	認証のタイプ。 指定可能な値: 0 - *(デフォルト)* ユーザー名とパスワードを使用; 1 - 認証を使用しない。 Property behavior: - ユーザーが `status` を `1`（設定済み）に変更する場合は必須。
username	string	データソースへのログインに使用するユーザー名。 Property behavior: - `status` が `1`（設定済み）に設定され、`authentication_type` が `0`（ユーザー名とパスワード）に設定されている場合にサポート。 - ユーザーが `authentication_type` を `0`（ユーザー名とパスワード）に変更する場合は必須。

Property	Type	Description
password	string	データソースへのログインに使用するパスワード。 このパラメータの値は空にできます。 Property behavior: - status が 1（設定済み）に設定され、authentication_type が 0（ユーザー名とパスワード）に設定されている場合にサポート。
db	string	データベース名。 このパラメータの値は空にできます。 Property behavior: - status が 1（設定済み）に設定されている場合にサポート。
ssl_verify_peer	integer	サーバー証明書を検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。 Property behavior: - status が 1（設定済み）に設定され、url が https://スキームを使用している場合にサポート。
ssl_verify_host	integer	ホストを検証するかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。 Property behavior: - status が 1（設定済み）に設定され、url が https://スキームを使用している場合にサポート。

settings.get

説明

`object settings.get(object parameters)`

このメソッドを使用すると、指定したパラメータに基づいて設定オブジェクトを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	このパラメータはリファレンスコメントで説明されています。

戻り値

(object) 設定オブジェクトを返します

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "alert_usrgrpid": "7",
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
```

```
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": " 未分類",
        "severity_name_1": " 情報",
        "severity_name_2": " 警告",
        "severity_name_3": " 平均",
        "severity_name_4": " 高",
        "severity_name_5": " 災害",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```


ソース

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

説明

`object settings.update(object settings)`

このメソッドを使用すると、既存の共通設定を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新する設定プロパティ。

戻り値

(array) 更新されたパラメータ名の配列を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

ソース

CSettings::update() in *ui/include/classes/api/services/CSettings.php*.

SLA

このクラスは、IT インフラストラクチャおよびビジネスサービスのパフォーマンスを評価するために使用される SLA（Service Level Agreement）オブジェクトを扱うように設計されています。

オブジェクト参照:

- SLA
- SLA スケジュール
- SLA 除外ダウンタイム
- SLA サービスタグ

利用可能なメソッド:

- sla.create - 新しい SLA を作成
- sla.delete - SLA を削除
- sla.get - SLA を取得
- sla.getsli - SLA の Service Level Indicator（SLI）データを取得
- sla.update - SLA を更新

SLA オブジェクト

次のオブジェクトは、`sla`(Service Level Agreement) API に直接関連しています。

SLA

SLA オブジェクトには以下のプロパティがあります。

プロパティ	Type	説明
slaid	ID	SLA の ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
name	string	SLA の名前。 プロパティの動作: - 作成操作で 必須

プロパティ	Type	説明
period	integer	SLA のレポート期間。 指定可能な値: 0 - 日次; 1 - 週次; 2 - 月次; 3 - 四半期ごと; 4 - 年次。 プロパティの動作: - 作成操作で 必須
slo	float	パーセントで表される、許容される最小のサービスレベル目標（SLO）。サービスレベル指標（SLI）がこれを下回ると、SLA は問題あり/未達成の状態と見なされます。 指定可能な値: 0～100（小数点以下最大 4 桁）。 プロパティの動作: - 作成操作で 必須
effective_date	integer	SLA の発効日。 指定可能な値: UTC での日時タイムスタンプ。
timezone	string	レポートのタイムゾーン（例: `Europe/London`、`UTC`）。 サポートされているタイムゾーンの完全な一覧については、PHP documentationを参照してください。 プロパティの動作: - 作成操作で 必須
status	integer	SLA のステータス。 指定可能な値: 0 - （デフォルト）無効な SLA; 1 - 有効な SLA。
description	string	SLA の説明。

SLA スケジュール

SLA スケジュールオブジェクトは、接続されたサービスが稼働状態であることが予定されている期間を定義します。以下のプロパティを持ちます。

プロパティ	Type	説明
period_from	integer	繰り返し発生する週次期間の開始時刻（含む）。 指定可能な値: 秒数（日曜日からの経過時間）。 プロパティの動作: - 必須
period_to	integer	繰り返し発生する週次期間の終了時刻（含まない）。 指定可能な値: 秒数（日曜日からの経過時間）。 プロパティの動作: - 必須

SLA excluded downtime

excluded downtime オブジェクトは、接続されたサービスが SLI に影響を与えることなく、正常に機能していないスケジュールを定義します。例えば、計画的なメンテナンスなどです。以下のプロパティがあります。

プロパティ	Type	説明
name	string	除外されるダウンタイムの名前。 プロパティの動作: - 必須
period_from	integer	除外されるダウンタイムの開始時刻（この時刻を含む）。 指定可能な値: タイムスタンプ。 プロパティの動作: - 必須
period_to	integer	除外されるダウンタイムの終了時刻（この時刻を含まない）。 指定可能な値: タイムスタンプ。 プロパティの動作: - 必須

SLA サービスタグ

SLA サービスタグオブジェクトは、SLA の計算に含めるサービスをリンクします。次のプロパティがあります。

プロパティ	型	説明
tag	string	SLA サービスタグ名。 プロパティの動作: - 必須
operator	integer	SLA サービスタグの演算子。 使用可能な値: 0 - *(デフォルト)* 等しい; 2 - 含む。
value	string	SLA サービスタグの値。

sla.create

説明

`object sla.create(object/array SLAs)`

このメソッドを使用すると、新しい SLA オブジェクトを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成する SLA オブジェクト。

標準 SLA プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
service_tags	array	SLA 用に作成するSLA サービスタグ。 パラメータの動作: - 必須
schedule	array	SLA 用に作成するSLA スケジュール。 空のパラメータを指定した場合は、24x7 スケジュールとして解釈されます。 デフォルト: 24x7 スケジュール。

パラメータ	Type	説明
excluded_downtimes	array	SLA 用に作成するSLA 除外ダウンタイム。

戻り値

(object) 作成された SLA の ID を slaids プロパティに含むオブジェクトを返します。返される ID の順序は、渡された SLA の順序と一致します。

例

SLA の作成

以下のために SLA エントリを作成するよう指示します: * SQL エンジン関連サービスの稼働時間の追跡; * 土曜日の最終時間を除くすべての平日のカスタムスケジュール; * 2022 年の最終日を有効日とする; * 7 月 4 日午前 0 時から 1 時間 15 分の計画停止; * SLA の週次レポート計算を有効にする; * 最小許容 SLO は 99.9995% とする。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Database Uptime",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Provide excellent uptime for main database engines.",
            "effective_date": 1672444800,
            "status": 1,
            "schedule": [
                {
                    "period_from": 0,
                    "period_to": 601200
                }
            ],
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla::create()。

sla.delete

説明

`object sla.delete(array slaids)`

このメソッドを使用すると、SLA エントリを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`array`) 削除する SLA の ID。

戻り値

(`object`) 削除された SLA の ID を `slaids` プロパティに含むオブジェクトを返します。

例

複数の SLA の削除

2 つの SLA エントリを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla::delete()。

sla.get

説明

`integer/array sla.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って SLA オブジェクトを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
slaids	ID/array	指定された ID を持つ SLA のみを返します。
serviceids	ID/array	特定のサービスに一致する SLA のみを返します。
selectSchedule	query	SLA スケジュールを含む `schedule` プロパティを返します。 `count` をサポートします。
selectExcludedDowntimes	query	SLA の除外ダウンタイムを含む `excluded_downtimes` プロパティを返します。 `count` をサポートします。
selectServiceTags	query	SLA サービスタグを含む `service_tags` プロパティを返します。 `count` をサポートします。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。 指定可能な値: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

すべての SLA の取得

すべての SLA とそのプロパティに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "sla.get",
    "params": {
        "output": "extend",
        "selectSchedule": ["period_from", "period_to"],
        "selectExcludedDowntimes": ["name", "period_from", "period_to"],
        "selectServiceTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "slaid": "1",
            "name": "Database Uptime",
            "period": "1",
            "slo": "99.9995",
            "effective_date": "1672444800",
            "timezone": "America/Toronto",
            "status": "1",
            "description": "Provide excellent uptime for main SQL database engines.",
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla:get()。

sla.getsli

説明

`object sla.getsli(object parameters)`

このメソッドを使用すると、Service Level Agreement (SLA) の Service Level Indicator (SLI) データを計算できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細については、User roles を参照してください。

パラメータ

(object) SLA ID、レポート期間、およびオプションで SLI の計算対象となるサービスの ID を含むパラメータ。

パラメータ	型	説明
slaid	ID	可用性情報を返す SLA の ID。 パラメータの動作: - 必須
period_from	timestamp	SLI をレポートする開始タイムスタンプ（この値を含む）。 指定可能な値: Unix タイムスタンプ。
period_to	timestamp	SLI をレポートする終了タイムスタンプ（この値を含む）。 指定可能な値: Unix タイムスタンプ。
periods	integer	レポートする期間の数。 指定可能な値: 1-100
serviceids	ID/array	SLI を返すサービスの ID。

期間の分割

次の表は、パラメータの組み合わせに基づく返される期間スライスの配置を示しています。

Note:
返される期間は、最初に利用可能な期間（SLA の有効日付に基づく）より前にはならず、現在の期間を超えることもありません。

パラメータ			返される期間
period_from	**period_to**	**periods**	
-	-	-	現在の期間を含む直近 20 期間。
-	-	N	直近 N 期間。
-	指定	-	period_to の直前の 20 期間。
-	指定	N	period_to の直前の N 期間。
指定	-	-	period_from から始まる最初の 20 期間。
指定	-	N	period_from から始まる最初の N 期間。
指定	指定	-	指定範囲内の最大 100 期間。
指定	指定	N	指定範囲内の N 期間。

戻り値

(object) 計算結果を返します。

プロパティ	Type	説明
periods	array	報告対象期間のリスト。 各報告対象期間は、以下で構成されるオブジェクトとして表されます。 - period_from - (timestamp) 報告対象期間の開始タイムスタンプ（含む）。 - period_to - (timestamp) 報告対象期間の終了タイムスタンプ（含まない）。 期間は period_from でソートされ、最も早い期間が先頭に表示されます。

プロパティ	Type	説明
serviceids	array	報告対象期間内のサービス ID のリスト。 リストの並び順は定義されていません。`serviceids` パラメータが `sla.getsli` メソッドに渡された場合でも同様です。
sli	array	各報告対象期間およびサービスの SLI データ（二次元配列として）。 `periods` プロパティのインデックスは、`sli` プロパティの第 **1** 次元として使用されます。 `serviceids` プロパティのインデックスは、`sli` プロパティの第 **2** 次元として使用されます。

SLI データ

報告された各期間およびサービスについて返される SLI データは、以下で構成されます。

Property	Type	Description
uptime	integer	スケジュールされた稼働時間中にサービスが *OK* 状態であった時間から、除外されたダウンタイムを差し引いた時間。
downtime	integer	スケジュールされた稼働時間中にサービスが *not OK* 状態であった時間から、除外されたダウンタイムを差し引いた時間。
sli	float	稼働時間と停止時間に基づく SLI（総稼働時間に対する割合）。
error_budget	integer	SLI および SLO に基づくエラーバジェット（秒単位）。
excluded_downtimes	array	この報告期間における除外されたダウンタイムの配列。 各オブジェクトには以下のパラメータが含まれます。 - `name` - (`string`) 除外されたダウンタイムの名前。 - `period_from` - (`timestamp`) 除外されたダウンタイムの開始タイムスタンプ（この時刻を含む）。 - `period_to` - (`timestamp`) 除外されたダウンタイムの終了タイムスタンプ（この時刻を含まない）。 除外されたダウンタイムは `period_from` でソートされ、最も早い期間が先に表示されます。

例

日次 SLA の SLI 計算

ID「1」の SLA にリンクされているサービス ID「1」と「4」の SLI データを取得します。「1761861599」(2025 年 10 月 30 日 23:59:59 GMT+0200）までの単一期間のデータを取得します。SLA のレポート期間が日次であるため、SLI データは「1761775200」(2025 年 10 月 30 日 00:00:00 GMT+0200）から「1761861600」(2025 年 10 月 31 日 00:00:00 GMT+0200）まで取得されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599,
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1761775200,
                "period_to": 1761861600
            }
        ],
        "serviceids": [
            1,
            4
        ],
        "sli": [
            [
                {
                    "uptime": 43843,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance OCT",
                            "period_from": 1761825600,
                            "period_to": 1761829200
                        }
                    ]
                },
                {
                    "uptime": 32225,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


月次 SLA の SLI 計算

ID が"5" の SLA にリンクされている ID が"50"、"60"、"70" のサービスの SLI データを取得します。"1635724800"（2021 年 11 月 1 日 00:00:00 UTC）から始まる 3 期間分のデータを取得します。SLA のレポート期間が月次であるため、以下の 3 か月分の SLI データが取得されます。

- "1635724800"（2021 年 11 月 1 日 00:00:00 UTC）から"1638316800"（2021 年 12 月 1 日 00:00:00 UTC）まで
- "1638316800"（2021 年 12 月 1 日 00:00:00 UTC）から"1640995200"（2022 年 1 月 1 日 00:00:00 UTC）まで
- "1640995200"（2022 年 1 月 1 日 00:00:00 UTC）から"1643673600"（2022 年 2 月 1 日 00:00:00 UTC）まで

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
            50,
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
```

```
                    "name": "Maintenance Nov 25 - Dec 01",
                    "period_from": 1637836212,
                    "period_to": 1638316800
                }
            ]
        }
    ],
    [
        {
            "uptime": 1147548,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Dec 02 - Dec 10",
                    "period_from": 1638439200,
                    "period_to": 1639109652
                }
            ]
        },
        {
            "uptime": 1147548,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Dec 02 - Dec 10",
                    "period_from": 1638439200,
                    "period_to": 1639109652
                }
            ]
        },
        {
            "uptime": 1147548,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Dec 02 - Dec 10",
                    "period_from": 1638439200,
                    "period_to": 1639109652
                }
            ]
        }
    ],
    [
        {
            "uptime": 1674000,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": []
        },
        {
            "uptime": 1674000,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": []
```

```
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```

ソース

CSla::getSli() in *ui/include/classes/api/services/CSla.php*

sla.update

説明

`object sla.update(object/array slaids)`

このメソッドを使用すると、既存の SLA エントリを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新する SLA プロパティ。

各 SLA に対して `slaid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準 SLA プロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
service_tags	array	現在の SLA サービスタグを置き換えるSLA サービスタグ。
schedule	array	現在のものを置き換えるSLA スケジュール。 パラメータに空の値を指定した場合、24x7 スケジュールとして解釈されます。
excluded_downtimes	array	現在の SLA 除外ダウンタイムを置き換えるSLA 除外ダウンタイム。

戻り値

(`object`) `slaids` プロパティに更新された SLA の ID を含むオブジェクトを返します。

例

サービスタグの更新

NoSQL 関連サービスの SLA ID "5" を月次で計算するように設定し、スケジュールや除外するダウンタイムは変更せず、SLO を 95% に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": [
        {
            "slaid": "5",
            "name": "NoSQL Database engines",
```

```
            "slo": "95",
            "period": 2,
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "redis"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mongodb"
                }
            ]
        }
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA のスケジュール変更

ID ”5” の SLA を 24 時間 365 日のスケジュールに切り替えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA の除外ダウンタイムの変更

2022 年 4 月 6 日に 4 時間の RAM アップグレードのためのダウンタイムを計画的に追加します。ただし、ID ”5” の SLA については、7 月 4 日に既に計画されているソフトウェアアップグレードは維持します（新たに定義する必要があります）

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "excluded_downtimes": [
            {
                "name": "Software version upgrade rollout",
                "period_from": "1648760400",
                "period_to": "1648764900"
            },
            {
                "name": "RAM upgrade",
                "period_from": "1649192400",
                "period_to": "1649206800"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php の CSla::update()。

Token

このクラスは、トークンを操作するためのものです。

オブジェクトリファレンス:

- Token

利用可能なメソッド:

- token.create - 新しいトークンを作成する
- token.delete - トークンを削除する
- token.get - トークンを取得する
- token.update - トークンを更新する
- token.generate - トークンを生成する

Token オブジェクト

次のオブジェクトは、tokenAPI に直接関連しています。

トークン

トークンオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tokenid	ID	トークンの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須

プロパティ	型	説明
name	string	トークンの名前。 プロパティの動作: - 作成操作で 必須
description	text	トークンの説明。
userid	ID	トークンが割り当てられているユーザーの ID。 デフォルト: 現在のユーザー。 プロパティの動作: - 定数
lastaccess	timestamp	トークンが最後に認証された日時。 トークンが一度も認証されていない場合は”0”。 プロパティの動作: - 読み取り専用
status	integer	トークンのステータス。 使用可能な値: 0 - *(デフォルト)* 有効なトークン; 1 - 無効なトークン。
expires_at	timestamp	トークンの有効期限の日時。 有効期限のないトークンの場合は”0”。
created_at	timestamp	トークンの作成日時。 プロパティの動作: - 読み取り専用
creator_userid	ID	トークンを作成したユーザーの ID。 プロパティの動作: - 読み取り専用

token.create

説明

`object token.create(object/array tokens)`

このメソッドを使用すると、新しいトークンを作成できます。

Note:
他のユーザーのトークンを管理するには、ユーザーの役割に *Manage API tokens* 権限 が必要です。

Attention:
このメソッドで作成したトークンは、使用可能になる前に生成 する必要もあります。

パラメータ

`(object/array)` トークンの作成

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

`(object)` `tokenids` プロパティの下で作成されたトークンの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトークンの順序と一致します。

例

トークンの作成

有効期限が切れることのない有効なトークンを作成し、ID2 のユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

2021 年 1 月 21 日に有効期限が切れるトークンを作成します。このトークンは現在のユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

ソース

CToken::create() in *ui/include/classes/api/services/CToken.php*.

token.delete

説明

`object token.delete(array tokenids)`

このメソッドを使用すると、トークンを削除できます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理権限 が必要です。

パラメーター

(array) 削除するトークンの ID。

戻り値

(object) tokenids プロパティの下にある削除されたトークンの ID を含むオブジェクトを返します。

例

複数のトークンの削除

2 つのトークンを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

ソース

CToken::delete() in *ui/include/classes/api/services/CToken.php*.

token.generate

説明

`object token.generate(array tokenids)`

このメソッドを使用すると、トークンを生成できます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーのロールに *Manage API tokens* 権限 が必要です。

> **Attention:**
> このメソッドでトークンを生成できるのは、トークンが作成 されている場合のみです。

パラメータ

(array) 生成するトークンの ID

戻り値

(array) 生成されたトークンの ID を tokenid プロパティに、生成された認可文字列を token プロパティに含むオブジェクトの配列を返します。

プロパティ	Type	説明
tokenid	ID	トークンの ID。
token	string	このトークン用に生成された認可文字列。

例

複数のトークンを生成する

2 つのトークンを生成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

ソース

CToken::generate() in *ui/include/classes/api/services/CToken.php*.

token.get

説明

`integer/array token.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってトークンを取得できます。

> **Note:**
> *Super admin* ユーザータイプのみ、他のユーザーのトークンを表示できます。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
tokenids	ID/array	指定した ID を持つトークンのみを返します。
userids	ID/array	指定したユーザー用に作成されたトークンのみを返します。
token	string	指定した *Auth token* 用に作成されたトークンのみを返します。
valid_at	timestamp	指定した日時において有効（期限切れでない）なトークンのみを返します。
expired_at	timestamp	指定した日時において期限切れ（無効）であるトークンのみを返します。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.

パラメータ	型	説明
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トークンの取得

ID"2" のトークンのデータをすべて取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "name": "The Token",
            "description": "",
            "userid": "1",
            "lastaccess": "0",
            "status": "0",
            "expires_at": "1609406220",
            "created_at": "1611239454",
            "creator_userid": "1"
        }
    ],
    "id": 1
}
```

ソース

CToken::get() in *ui/include/classes/api/services/CToken.php*.

token.update

説明

`object token.update(object/array tokens)`

このメソッドを使用すると、既存のトークンを更新できます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーのロールに *API* トークンの管理権限 が必要です。

パラメータ

(object/array) 更新されるトークンプロパティ

tokenid プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) tokenids プロパティの下で更新されたトークンの ID を含むオブジェクトを返します。

例

トークンの有効期限変更

トークンから有効期限を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CToken::update() in *ui/include/classes/api/services/CToken.php*.

User

このクラスは、ユーザーを操作するためのものです。

オブジェクトリファレンス:

- User
- Media

利用可能なメソッド:

- user.checkauthentication - ユーザーセッションの確認と延長をする
- user.create - 新しいユーザーを作成する
- user.delete - ユーザーを削除する
- user.get - ユーザーを取得する
- user.login - API にログインする
- user.logout - API からログアウトする
- user.provision - LDAP ユーザーのプロビジョニングをする
- user.resettotp - ユーザーの TOTP シークレットをリセットする

- user.unblock - ユーザーブロックを解除する
- user.update - ユーザーを更新する

User オブジェクト

次のオブジェクトは、userAPI に直接関連しています。

ユーザー

ユーザーオブジェクトには、次のプロパティがあります。

Property	Type	Description
userid	ID	ユーザーの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
username	string	ユーザー名。 プロパティの動作: - 作成操作では必須 - ユーザーがユーザーディレクトリにリンクされており (userdirectoryid に「0」以外の有効な値が設定されている)、ユーザーディレクトリのプロビジョニングステータスが有効で (ユーザーディレクトリオブジェクトの provision_status が「1」に設定されている)、LDAP または SAML のすべてのプロビジョニングの認証ステータスが有効な場合 (認証オブジェクトの ldap_jit_status が「設定済みの LDAP IdP で有効」または s aml_jit_status が「設定済みの SAML IdP で有効」に設定されている場合)、プロビジョニングされたユーザーでは読み取り専用
passwd	string	ユーザーのパスワード。 ユーザーがユーザーディレクトリにリンクされている場合、このパラメータの値は空の文字列にできます。 プロパティの動作: - 書き込み専用
roleid	ID	ユーザーのロールの ID。 ロールを持たないユーザーは、LDAP/SAML 情報が Zabbix で設定されたユーザーグループのマッピングと一致する場合に限り、LDAPまたはSAML認証を使用してのみ Zabbix にログインできます。
attempt_clock	timestamp	最後に失敗したログイン試行の時刻。 プロパティの動作: - 読み取り専用
attempt_failed	integer	最近のログイン失敗回数。 プロパティの動作: - 読み取り専用
attempt_ip	string	最後に失敗したログイン試行元の IP アドレス。 プロパティの動作: - 読み取り専用
autologin	integer	自動ログインを有効にするかどうか。 指定可能な値: 0 - *(デフォルト)* 自動ログイン無効; 1 - 自動ログイン有効。

Property	Type	Description
autologout	string	ユーザーセッションの有効期間。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h）を指定します。0s に設定すると、セッションは期限切れになりません。 デフォルト: 15m。
default_maintenance_period	string	ショートカットで作成したメンテナンスのデフォルトのメンテナンス期間。秒数またはサフィックス付きの時間単位（300s、5m、1h、1d、1w）を指定します。指定した値は分単位に切り捨てられます。 デフォルト: 1h。
lang	string	ユーザーの言語の言語コード（例: en_US）。 デフォルト: default - システムのデフォルト。
name	string	ユーザーの名前。
provisioned	integer	ユーザーがプロビジョニングされているかどうか。 指定可能な値: 0 - プロビジョニングされていない; 1 - プロビジョニング済み。 プロパティの動作: - 読み取り専用
refresh	string	自動更新の間隔。秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、1h）を指定します。 デフォルト: 30s。
rows_per_page	integer	1 ページに表示するオブジェクトの行数。 デフォルト: 50。
surname	string	ユーザーの姓。
theme	string	ユーザーのテーマ。 指定可能な値: default - *(デフォルト)* システムのデフォルト; blue-theme - 青; blue-classic-theme - 青（クラシック）; dark-theme - ダーク; dark-blue-theme - ダークブルー; dark-classic-theme - ダーク（クラシック）; hc-light - ハイコントラスト（ライト）; hc-dark - ハイコントラスト（ダーク）。
ts_provisioned	timestamp	最新のプロビジョニング操作が実行された時刻。 プロパティの動作: - 読み取り専用 - スーパーユーザータイプのユーザーでサポート
url	string	ログイン後にユーザーをリダイレクトするページの URL。
userdirectoryid	ID	ユーザーがリンクされているユーザーディレクトリの ID。 プロビジョニング（作成または更新）と、ユーザーディレクトリにリンクされたユーザーのログインに使用されます。 ログイン操作では、このプロパティの値が、ユーザーが所属するユーザーグループの userdirectoryid プロパティより優先されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用 - スーパーユーザータイプのユーザーでサポート

Property	Type	Description
timezone	string	ユーザーのタイムゾーン（例: `Europe/London`、`UTC`）。 デフォルト: `default` - システムのデフォルト。 サポートされているタイムゾーンの完全な一覧については、PHP ドキュメントを参照してください。

メディア

メディアオブジェクトには、次のプロパティがあります。

Property	Type	Description
mediaid	ID	ユーザーのメディアの ID。 プロパティの動作: - 読み取り専用
mediatypeid	ID	ユーザーのメディアで使用されるメディアタイプの ID。 プロパティの動作: - 必須
sendto	string/array	受信者のアドレス、ユーザー名、その他の識別子。 メディアタイプの `type` が「Email」または「Push」に設定されている場合、値は配列で表されます。「Push」の場合、配列にはワイルドカード（`*`）または 1 つ以上のモバイルデバイス ID を指定できます。 メディアタイプの `type` が「Script」、「SMS」、または「Webhook」に設定されている場合、値は文字列で表されます。 プロパティの動作: - 必須
active	integer	メディアが有効かどうか。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
severity	integer	通知を送信するトリガーの深刻度。 指定可能なビットマップ値: 1 - 未分類; 2 - 情報; 4 - 警告; 8 - 平均; 16 - 高; 32 - 災害。 これはビットマスクフィールドです。指定可能なビットマップ値の合計を指定できます（例: 平均、高、災害の場合は 48）。 デフォルト: 63。
period	string	通知を送信できる時間。時間帯またはセミコロンで区切ったユーザーマクロとして指定します。 デフォルト: 1-7,00:00-24:00。
provisioned	integer	ユーザーがプロビジョニングされているかどうか。 指定可能な値: 0 - プロビジョニングされていない; 1 - プロビジョニングされている。

Property	Type	Description
userdirectory_mediaid	ID	プロビジョニングされたメディアのユーザーディレクトリメディアマッピング ID。 プロパティの動作: - 読み取り専用 - スーパーユーザータイプのユーザーでサポート

user.checkAuthentication

説明

`object user.checkAuthentication`

このメソッドはユーザーセッションをチェックし、延長します。

Attention:
`sessionid` パラメータを使用して `user.checkAuthentication` メソッドを呼び出すと、デフォルトでユーザーセッションが延長されます。

パラメータ

このメソッドは次のパラメータを受け付けます。

Parameter	Type	Description
extend	boolean	ユーザーセッションを延長するかどうか。 デフォルト値: "true"。 値を"false" に設定すると、セッションを延長せずにユーザーセッションを確認できます。
sessionid	string	Parameter behavior: - *supported* if `sessionid` is set ユーザーの認証トークン。
token	string	Parameter behavior: - *required* if `token` is not set ユーザーのAPI トークン。
		Parameter behavior: - *required* if `sessionid` is not set

戻り値

(`object`) ユーザーに関する情報を含むオブジェクトを返します。

標準のユーザープロパティに加えて、以下の情報が返されます。

プロパティ	Type	説明
auth_type	integer	ユーザーのデフォルト認証。 指定可能な値の一覧については、Authentication object の `authentication_type` プロパティを参照してください。
debug_mode	integer	ユーザーに対してデバッグモードが有効か無効か。 指定可能な値の一覧については、User group object の `debug_mode` プロパティを参照してください。
deprovisioned	boolean	ユーザーがプロビジョニング解除済みユーザーグループに属しているかどうか。

プロパティ	Type	説明
gui_access	string	ユーザーの Web インターフェースへの認証方法。 指定可能な値の一覧については、User group object の gui_access プロパティを参照してください。
secret	string	32 文字のランダムな文字列。ユーザーログイン時に生成されます。 API トークンを使用してユーザーセッションを確認する場合、secret プロパティは返されません。
sessionid	string	認証トークン。以降の API リクエストで使用する必要があります。 API トークンを使用してユーザーセッションを確認する場合、sessionid プロパティは返されません。
type	integer	ユーザータイプ。 指定可能な値の一覧については、Role object の type プロパティを参照してください。
userip	string	ユーザーの IP アドレス。

例

認証トークンを使用し認証を確認

ユーザー認証トークンを使用してユーザーセッションを確認し、延長します。また、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "deprovisioned": false,
```

```
        "auth_type": 0,
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2",
        "secret": "0e329b933e46984e49a5c1051ecd0751"
    },
    "id": 1
}
```

API トークンを使用した認証の確認

ユーザー API トークンを使用してユーザーセッションを確認し、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919338",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "debug_mode": 0,
        "deprovisioned": false,
        "gui_access": "1",
        "mfaid": 0,
        "auth_type": 0,
        "type": 3,
        "userip": "127.0.0.1"
    },
    "id": 1
}
```

ソース

CUser::checkAuthentication() in *ui/include/classes/api/services/CUser.php*。

user.create

説明

`object user.create(object/array users)`

このメソッドを使用すると、新しいユーザーを作成できます。

Note:
このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

Note:
ユーザーパスワードの強度は、Authentication API で定義されたパスワードポリシーのルールに従って検証されます。詳細については、authentication オブジェクトを参照してください。

パラメータ

(object/array) 作成するユーザー。

標準のユーザープロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
usrgrps	array	ユーザーの追加先となるユーザーグループ。 ユーザーグループでは、usrgrpid プロパティのみを定義する必要があります。
medias	array	作成するユーザーメディア。

戻り値

(object) userids プロパティの下で作成されたユーザーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーの順序と一致します。

例

ユーザーの作成

新しいユーザーを作成し、ユーザーグループに追加して、そのユーザー用の新しいメディアを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

関連項目

- 認証
- メディア
- ユーザーグループ
- ロール

ソース

CUser::create() in *ui/include/classes/api/services/CUser.php*.

user.delete

説明

`object user.delete(array users)`

このメソッドを使用すると、ユーザーを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するユーザーの ID。

戻り値

(object) userids プロパティの下にある削除されたユーザーの ID を含むオブジェクトを返します。

例

複数のユーザーの削除

2 つのユーザーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
```

```
    "id": 1
}
```

ソース

CUser::delete() in *ui/include/classes/api/services/CUser.php*.

user.get

説明

`integer/array user.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってユーザーを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Note:
ユーザーメディア、権限、またはロール情報を要求する場合、*Admin* および *User* タイプのユーザーは、自分自身のユーザーに関するデータのみ取得できます。例については、Retrieving users as *Admin* を参照してください。

パラメータ	型	説明
mediaids	ID/array	指定されたメディアを使用しているユーザーのみを返します。
mediatypeids	ID/array	指定されたメディアタイプを使用しているユーザーのみを返します。
userids	ID/array	指定された ID を持つユーザーのみを返します。
usrgrpids	ID/array	指定されたユーザーグループに属するユーザーのみを返します。
getAccess	flag	ユーザー権限に関する追加情報を付加します。 各ユーザーに対して以下のプロパティを追加します。 `gui_access` - `(integer)` ユーザーの Web インターフェース認証方式。使用可能な値の一覧については、ユーザーグループオブジェクト の `gui_access` プロパティを参照してください。 `debug_mode` - `(integer)` ユーザーに対してデバッグが有効かどうかを示します。使用可能な値: 0 - デバッグ無効、1 - デバッグ有効。 `users_status` - `(integer)` ユーザーが無効化されているかどうかを示します。使用可能な値: 0 - ユーザー有効、1 - ユーザー無効。
selectMedias	query	ユーザーが使用しているメディアを`medias` プロパティで返します。
selectMediatypes	query	ユーザーが使用しているメディアタイプを`mediatypes` プロパティで返します。 ユーザータイプに基づく制限については、`mediatype.get` を参照してください。
selectUsrgrps	query	ユーザーが属しているユーザーグループを`usrgrps` プロパティで返します。 ユーザータイプに基づく制限については、`usergroup.get` を参照してください。
selectRole	query	ユーザーロールを`role` プロパティで返します。

パラメータ	型	説明
filter	object	指定されたフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしていません。
output	query	*Admin* および *User* タイプのユーザーが、自分のユーザーグループ内のユーザーに関するデータを要求する場合に使用可能なユーザーオブジェクト のプロパティ: `userid`, `name`, `surname`, `username`。 返されるユーザーオブジェクト のプロパティ。 *Admin* および *User* タイプのユーザーは、以下のプロパティのみ取得できます。 - 自分自身のユーザー: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`。 - 自分のユーザーグループ内のユーザー: `userid`, `name`, `surname`, `username`。 デフォルト: `extend`。
search	object	指定されたパターンに一致する結果を返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索を実行します。 `string` および `text` データ型 のプロパティのみをサポートします。 *Admin* および *User* タイプのユーザーが、自分のユーザーグループ内のユーザーに関するデータを要求する場合に使用可能なユーザーオブジェクト のプロパティ: `name`, `surname`, `username`。
sortfield	string/array	指定されたプロパティで結果をソートします。 使用可能な値: `userid`, `username`。
countOutput	boolean	これらのパラメータについては、reference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ユーザーの取得

設定されているすべてのユーザーを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```

Admin としてユーザーを取得する

Admin タイプのユーザーとして、自分自身のユーザーに関する詳細データと、所属するユーザーグループ内のユーザーに関する限定的なデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "1",
      "username": "Admin",
      "name": "Zabbix",
      "surname": "Administrator",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
      "attempt_clock": "0",
      "rows_per_page": "50",
      "timezone": "default",
      "roleid": "3",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "0",
      "usrgrps": [
        {
          "usrgrpid": "7",
          "name": "Zabbix administrators",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
```

```
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      }
    ],
    "medias": [],
    "mediatypes": [],
    "role": {
      "roleid": "3",
      "name": "Super admin role",
      "type": "3",
      "readonly": "1"
    }
  },
  {
    "userid": "2",
    "username": "guest",
    "name": "",
    "surname": "",
    "url": "",
    "autologin": "0",
    "autologout": "15m",
    "lang": "default",
    "default_maintenance_period": "1h",
    "refresh": "30s",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "",
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
    "roleid": "4",
    "userdirectoryid": "0",
    "ts_provisioned": "0",
    "provisioned": "0",
    "gui_access": "1",
    "debug_mode": "0",
    "users_status": "1",
    "usrgrps": [
      {
        "usrgrpid": "8",
        "name": "Guests",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      },
      {
        "usrgrpid": "9",
        "name": "Disabled",
        "gui_access": "0",
        "users_status": "1",
        "debug_mode": "0",
        "userdirectoryid": "0",
```

```
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


ユーザーデータの取得

ID「2」のユーザーのデータを取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
}
```


レスポンス:


```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
```

```
}
```

関連項目

- メディア
- メディアタイプ
- ユーザーグループ
- ロール

ソース

CUser::get() in *ui/include/classes/api/services/CUser.php*.

user.login

説明

`string/object user.login(object parameters)`

このメソッドを使用すると、API にログインして認証トークンを生成できます。

Warning:
このメソッドを使用する場合は、開いているセッションレコードが大量に生成されるのを防ぐために、user.logout も実行する必要があります。

Attention:
このメソッドは、マルチファクター認証が有効になっているuser group に属していない、未認証ユーザーにのみ利用できます。

パラメータ

(object) ユーザー名とパスワードを含むパラメータ。

このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
password	string	ユーザーパスワード。 パラメータの動作: - 必須
username	string	ユーザー名。 パラメータの動作: - 必須
userData	flag	認証されたユーザーの情報を返します。

戻り値

(string/object) userData パラメータが使用されている場合、認証されたユーザーに関する情報を含むオブジェクトを返します。

標準のユーザープロパティに加えて、以下の情報が返されます。

プロパティ	型	説明
auth_type	integer	ユーザーのデフォルト認証。 使用可能な値の一覧については、Authentication オブジェクトの authentication_type プロパティを参照してください。
debug_mode	integer	ユーザーに対してデバッグモードが有効か無効か。 使用可能な値の一覧については、ユーザーグループオブジェクトの debug_mode プロパティを参照してください。
deprovisioned	boolean	ユーザーがプロビジョニング解除済みユーザーグループに属しているかどうか。

プロパティ	型	説明
gui_access	string	ユーザーの Web インターフェースへの認証方式。 使用可能な値の一覧については、ユーザーグループオブジェクトの gui_access プロパティを参照してください。
mfaid	integer	ログイン時にユーザーに対して使用するMFA 方式の ID。 MFA がグローバルに無効化されている場合、またはユーザーが属するすべてのユーザーグループで無効化されている場合は”0” を返します。
secret	string	32 文字のランダムな文字列。ユーザーログイン時に生成されます。
sessionid	string	認証トークン。以降の API リクエストで使用する必要があります。
type	integer	ユーザータイプ。 使用可能な値の一覧については、Role オブジェクトの type プロパティを参照してください。
userip	string	ユーザーの IP アドレス。

Note:
ユーザーが 1 回以上の認証失敗の後に正常に認証された場合、このメソッドは attempt_clock、attempt_failed、attempt_ip プロパティの現在の値を返し、その後それらをリセットします。

userData パラメータが使用されていない場合、このメソッドは認証に必要な認証トークンを返します。

例

ユーザーの認証

ユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

認証されたユーザー情報のリクエスト

認証を行い、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "mfaid": "1",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "5b56eee8be445e98f0bd42b435736e42",
        "secret": "cd0ba923319741c6586f3d866423a8f4"
    },
    "id": 1
}
```


参照

- user.logout

ソース

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.logout

説明

`string/object user.logout(array)`

このメソッドを使用すると、API からログアウトでき、現在の認証トークンは無効になります。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(array) このメソッドは空の配列を受け入れます。

戻り値

(boolean) ユーザーが正常にログアウトされた場合は `true` を返します

例

ログアウト

API からログアウトします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

参照

- user.login

ソース

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.provision

説明

`object user.provision(object/array users)`

このメソッドを使用すると、LDAP ユーザーをプロビジョニングできます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(array) プロビジョニングするユーザーの ID

戻り値

(object) userids プロパティの下でプロビジョニングされたユーザーの ID を含むオブジェクトを返します。

例

複数ユーザーのプロビジョニング

2 つのユーザーをプロビジョニングします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
```

```
        ]
    },
    "id": 1
}
```

ソース

CUser::provision() in *ui/include/classes/api/services/CUser.php*.

user.resettotp

説明

`object user.resettotp(object/array users)`

このメソッドを使用すると、ユーザーの TOTP シークレットをリセットできます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(配列) TOTP シークレットをリセットするユーザーの ID

> **Note:**
> 指定されたユーザーのユーザーセッションも削除されます（リクエストを送信したユーザーを除く）。

戻り値

(object) TOTP シークレットがリセットされたユーザーの ID を `userids` プロパティに格納したオブジェクトを返します。

例

複数のユーザーの TOTP シークレットのリセット

2 人のユーザーの TOTP シークレットをリセットします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

参照

- MFA object

ソース

CUser::resettotp() in *ui/include/classes/api/services/CUser.php*.

user.unblock

説明

`object user.unblock(array userids)`

このメソッドを使用すると、ユーザーのブロックを解除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(array) ユーザーブロックを解除する ID

戻り値

(object) userids プロパティの下にあるブロックされていないユーザーの ID を含むオブジェクトを返します。

例

複数のユーザーのブロック解除

2 人のユーザーのブロックを解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CUser::unblock() in *ui/include/classes/api/services/CUser.php*.

user.update

説明

`object user.update(object/array users)`

このメソッドを使用すると、既存のユーザーを更新できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、User rolesを参照してください。

Note:
ユーザーパスワードの強度は、Authentication API で定義されたパスワードポリシーのルールに従って検証されます。詳細については、Authentication APIを参照してください。

パラメータ

(object/array) 更新するユーザープロパティ。

各ユーザーについて userid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のユーザープロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
current_passwd	string	ユーザーの現在のパスワード。 ユーザーがユーザーディレクトリ に関連付けられている場合、このパラメータの値は空文字列にできます。 パラメータの動作: - *write-only* - User object の passwd が設定され、ユーザーが自身のユーザーパスワードを変更する場合は *required*
usrgrps	array	既存のユーザーグループを置き換えるユーザーグループ。 ユーザーグループでは usrgrpid プロパティのみを定義する必要があります。
medias	array	既存の、プロビジョニングされていないユーザーメディア を置き換えます。メディアを更新する際、プロビジョニングされたメディアは省略できます。

戻り値

(object) userids プロパティの下で更新されたユーザーの ID を含むオブジェクトを返します。

例

ユーザー名の変更

ユーザー名を John Doe に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

ユーザーロールの変更

ユーザーのロールを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

参照

- Authentication

ソース

CUser::update() in *ui/include/classes/api/services/CUser.php*.

Web シナリオ

このクラスは、Web シナリオで動作するように設計されています。

オブジェクト参照:

- Web シナリオ
- シナリオステップ
  - HTTP フィールド
- Web シナリオタグ

利用可能なメソッド:

- httptest.create - 新しい Web シナリオを作成
- httptest.delete - Web シナリオを削除
- httptest.get - Web シナリオを取得
- httptest.update - Web シナリオを更新

Web scenario オブジェクト

次のオブジェクトは、webcheckAPI に直接関連しています。

Web シナリオ

Web シナリオオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
httptestid	ID	Web シナリオの ID。 プロパティの動作: - 読み取り専用 - 更新操作で必須

プロパティ	タイプ	説明
hostid	ID	Web シナリオが属するホストの ID。 プロパティの動作: - 固定 - 作成操作で必須
name	string	Web シナリオの名前。 プロパティの動作: - 作成操作で必須
agent	string	Web シナリオで使用するユーザーエージェント文字列。 デフォルト: Zabbix
authentication	integer	Web シナリオで使用する認証方式。 指定可能な値: 0 - *(デフォルト)* なし; 1 - 基本 HTTP 認証; 2 - NTLM 認証; 3 - Kerberos 認証; 4 - ダイジェスト認証。
delay	string	Web シナリオの実行間隔。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはユーザーマクロを指定できます。 デフォルト: 1m。
headers	array	リクエストの実行時に送信されるHTTP ヘッダー。
http_password	string	基本 HTTP、NTLM、Kerberos、またはダイジェスト認証で使用するパスワード。
http_proxy	string	Web シナリオで使用するプロキシ。 *http://[username[:password]@]proxy.example.com[:port]* の形式で指定します。
http_user	string	基本 HTTP、NTLM、Kerberos、またはダイジェスト認証で使用するユーザー名。
retries	integer	失敗する前に、Web シナリオが各ステップの実行を試行する回数。 デフォルト: 1。
ssl_cert_file	string	クライアント認証で使用する SSL 証明書ファイルの名前（PEM 形式である必要があります）。
ssl_key_file	string	クライアント認証で使用する SSL 秘密鍵ファイルの名前（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵のパスワード。
status	integer	Web シナリオが有効かどうか。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
templateid	ID	親テンプレートの Web シナリオの ID。 プロパティの動作: - 読み取り専用
variables	array	Web シナリオの変数。
verify_host	integer	接続先のホスト名が、ホストの証明書に記載されたホスト名と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* ホストの検証をスキップ; 1 - ホストを検証。

プロパティ	タイプ	説明
verify_peer	integer	ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* ピアの検証をスキップ; 1 - ピアを検証。
uuid	string	インポートした Web シナリオを既存の Web シナリオにリンクするために使用するグローバル一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - Web シナリオがテンプレートに属している場合にサポート

シナリオステップ

シナリオステップオブジェクトは、特定の Web シナリオチェックを定義します。以下のプロパティがあります。

プロパティ	タイプ	説明
name	string	シナリオステップの名前。 プロパティの動作: - 必須
no	integer	Web シナリオ内のステップのシーケンス番号。 プロパティの動作: - 必須
url	string	チェックする URL。 プロパティの動作: - 必須
follow_redirects	integer	HTTP リダイレクトに従うかどうか。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。
headers	array	リクエストの実行時に送信されるHTTP ヘッダー。シナリオステップのヘッダーは、Web シナリオに指定されたヘッダーを上書きします。
posts	string/array	文字列（raw POST データ）またはHTTP フィールドの配列（フォームフィールドデータ）として指定する HTTP POST 変数。
required	string	レスポンスに含まれている必要があるテキスト。
retrieve_mode	integer	シナリオステップが取得する必要がある HTTP レスポンスの部分。 指定可能な値: 0 - *(デフォルト)* 本文のみ; 1 - ヘッダーのみ; 2 - ヘッダーと本文。
status_codes	string	カンマで区切られた、必要な HTTP ステータスコードの範囲。
timeout	string	リクエストのタイムアウト（秒）。秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、1h）、またはユーザーマクロを指定できます。 デフォルト: 15s。最大: 1h。最小: 1s。
variables	array	シナリオステップの変数。
query_fields	array	クエリフィールド。リクエストの実行時に URL に追加されるHTTP フィールドの配列。

HTTP フィールド

HTTP フィールドオブジェクトは、Web シナリオの変数、HTTP ヘッダー、POST フィールドまたはクエリフィールドを指定するために使用される名前と値を定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	ヘッダー/変数/POST または GET フィールドの名前。 プロパティの動作: - 必須
value	string	ヘッダー/変数/POST または GET フィールドの値。 プロパティの動作: - 必須

Web シナリオタグ

Web シナリオタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	Web シナリオタグ名。 プロパティの動作: - 必須
value	string	Web シナリオタグ値。 プロパティの動作: - 読み取り専用。
object	integer	タグの継承元オブジェクトのタイプ。 使用可能な値: 0 - テンプレート; 1 - ホスト. プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

httptest.create

説明

`object httptest.create(object/array webScenarios)`

このメソッドを使用すると、新しい Web シナリオを作成できます。

Note:
Web シナリオを作成すると、Web 監視アイテム のセットが自動的に作成されます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はユーザー権限 を参照してください。

パラメータ

`(object/array)` 作成する Web シナリオ。

このメソッドは、標準の Web シナリオプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
steps	array	シナリオステップ。 パラメータの動作: - 必須
tags	array	Web シナリオタグ。

戻り値

(object) httptestids プロパティの下で作成された Web シナリオの ID を含むオブジェクトを返します。返される ID の順序は、渡された Web シナリオの順序と一致します。

例

Web シナリオの作成

会社のホームページを監視するための Web シナリオを作成します。シナリオには 2 つのステップがあり、ホームページと「About」ページをチェックし、HTTP ステータスコード 200 が返されることを確認します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.create",
    "params": {
        "name": "Homepage check",
        "hostid": "10085",
        "steps": [
            {
                "name": "Homepage",
                "url": "http://example.com",
                "status_codes": "200",
                "no": 1
            },
            {
                "name": "Homepage / About",
                "url": "http://example.com/about",
                "status_codes": "200",
                "no": 2
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- シナリオステップ

ソース

CHttpTest::create() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

説明

`object httptest.delete(array webScenarioIds)`

このメソッドを使用すると、Web シナリオを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除する Web シナリオの ID。

戻り値

(object) httptestids プロパティの下にある削除された Web シナリオの ID を含むオブジェクトを返します。

例

複数の Web シナリオの削除

2 つの Web シナリオを削除します

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

ソース

CHttpTest::delete() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

説明

`integer/array httptest.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って Web シナリオを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
groupids	ID/array	指定したホストグループに属する Web シナリオのみを返します。
hostids	ID/array	指定したホストに属する Web シナリオのみを返します。

パラメータ	型	説明
httptestids	ID/array	指定した ID を持つ Web シナリオのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承された Web シナリオのみを返します。
inheritedTags	boolean	指定した `tags` をテンプレート/ホスト/リンクされたテンプレートにも持つ Web シナリオを返します。 指定可能な値: `true` - テンプレート/ホスト/リンクされたテンプレートも指定したタグを持っている必要があります。 `false` - *(デフォルト)* テンプレート/ホスト/リンクされたテンプレートのタグは無視されます。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効な Web シナリオのみを返します。
selectInheritedTags	query	テンプレート/ホスト/リンクされたテンプレート上のタグを含む`inheritedTags` プロパティを返します。
templated	boolean	`true` に設定すると、テンプレートに属する Web シナリオのみを返します。
templateids	ID/array	指定したテンプレートに属する Web シナリオのみを返します。
expandName	flag	Web シナリオ名内のマクロを展開します。
expandStepName	flag	シナリオステップ名内のマクロを展開します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or。 2 - Or。
tags	array	指定したタグを持つ Web シナリオのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべての Web シナリオを返します。 指定可能な`operator` の値: 0 - *(デフォルト)* 含む。 1 - 等しい。 2 - 含まない。 3 - 等しくない。 4 - 存在する。 5 - 存在しない。
selectHosts	query	Web シナリオが属するホストを、`hosts` プロパティ内の配列として返します。
selectSteps	query	Web シナリオのステップを`steps` プロパティで返します。 `count` をサポートします。
selectTags	query	Web シナリオのタグを`tags` プロパティで返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `httptestid`, `name`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用された場合は、取得したオブジェクトの数

例

Web シナリオの取得

Web シナリオ"9" のすべてのデータを取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "agent": "Zabbix",
            "authentication": "0",
            "http_user": "",
            "http_password": "",
            "hostid": "10084",
            "templateid": "0",
            "http_proxy": "",
            "retries": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "uuid": "",
            "headers": [],
            "variables": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [
                        {
```

```
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
                    "name": "Homepage / About",
                    "no": "2",
                    "url": "http://example.com/about",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "post_type": "1",
                    "headers": [],
                    "variables": [],
                    "query_fields": []
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- Host
- シナリオステップ

ソース

CHttpTest::get() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

説明

`object httptest.update(object/array webScenarios)`

このメソッドを使用すると、既存の Web シナリオを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新する Web シナリオのプロパティ。

各 Web シナリオでは `httptestid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準の Web シナリオプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
steps	array	既存のステップを置き換えるシナリオステップ。
tags	array	Web シナリオタグ。

戻り値

(object) httptestid プロパティの下で更新された Web シナリオの ID を含むオブジェクトを返します。

例

Web シナリオの有効化

Web シナリオを有効にするには、status を”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- シナリオステップ

ソース

CHttpTest::update() in *ui/include/classes/api/services/CHttpTest.php*.

アイコンマップ

このクラスはアイコンマップを扱うために設計されています。

オブジェクト参照:

- アイコンマップ
- アイコンマッピング

利用可能なメソッド:

- iconmap.create - 新しいアイコンマップを作成
- iconmap.delete - アイコンマップを削除
- iconmap.get - アイコンマップを取得
- iconmap.update - アイコンマップを更新

Icon map オブジェクト

次のオブジェクトは、iconmapAPI に直接関連しています。

アイコンマップ

アイコンマップオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
iconmapid	ID	アイコンマップの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須

プロパティ	型	説明
default_iconid	ID	デフォルトアイコンの ID。 プロパティの動作: - 作成操作で 必須
name	string	アイコンマップの名前。 プロパティの動作: - 作成操作で 必須

アイコンマッピング

アイコンマッピングオブジェクトは、特定のインベントリフィールド値を持つホストに使用する特定のアイコンを定義します。以下のプロパティがあります。

プロパティ	型	説明
iconid	ID	アイコンマッピングで使用されるアイコンの ID。 プロパティの動作: - 必須
expression	string	インベントリフィールドと照合するための式。 プロパティの動作: - 必須
inventory_link	integer	ホストインベントリフィールドの ID。 サポートされているインベントリフィールドの一覧については、ホストインベントリオブジェクトを参照してください。 プロパティの動作: - 必須
sortorder	integer	アイコンマップ内でのアイコンマッピングの位置。 プロパティの動作: - 読み取り専用

iconmap.create

説明

`object iconmap.create(object/array iconMaps)`

このメソッドを使用すると、新しいアイコンマップを作成できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するアイコンマップ。

標準のアイコンマッププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
mappings	array	アイコンマップ用に作成するアイコンマッピング。 パラメータの動作: - 必須

戻り値

`(object)` `iconmapids` プロパティの下で作成されたアイコンマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイコンマップの順序と一致します。

例

アイコンマップの作成

異なるタイプのホストを表示するためのアイコンマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.create",
    "params": {
        "name": "Type icons",
        "default_iconid": "2",
        "mappings": [
            {
                "inventory_link": 1,
                "expression": "server",
                "iconid": "3"
            },
            {
                "inventory_link": 1,
                "expression": "switch",
                "iconid": "4"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- アイコンマッピング

ソース

CIconMap::create() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

説明

`object iconmap.delete(array iconMapIds)`

このメソッドを使用すると、アイコンマップを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`array`) 削除するアイコンマップの ID。

戻り値

(`object`) `iconmapids` プロパティの下にある削除されたアイコンマップの ID を含むオブジェクトを返します。

例

複数のアイコンマップの削除

2 つのアイコンマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CIconMap::delete() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

説明

`integer/array iconmap.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってアイコンマップを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
iconmapids	ID/array	指定した ID のアイコンマップのみを返します。
sysmapids	ID/array	指定したマップで使用されているアイコンマップのみを返します。
selectMappings	query	使用されているアイコンマッピングを含む`mappings` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `iconmapid`, `name`。
countOutput	boolean	これらのパラメータについては、reference commentary を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

パラメータ	Type	説明
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

アイコンマップの取得

アイコンマップ"3" のすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mappings": [
                {
                    "iconmappingid": "3",
                    "iconmapid": "3",
                    "iconid": "6",
                    "inventory_link": "1",
                    "expression": "server",
                    "sortorder": "0"
                },
                {
                    "iconmappingid": "4",
                    "iconmapid": "3",
                    "iconid": "10",
                    "inventory_link": "1",
                    "expression": "switch",
                    "sortorder": "1"
                }
            ],
            "iconmapid": "3",
            "name": "Host type icons",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

関連項目

- アイコンマッピング

ソース

CIconMap::get() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

説明

`object iconmap.update(object/array iconMaps)`

このメソッドを使用すると、既存のアイコンマップを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するアイコンマップのプロパティ。

各アイコンマップでは `iconmapid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のアイコンマッププロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
mappings	array	既存のアイコンマッピングを置き換えるアイコンマッピング。

戻り値

(object) `iconmapids` プロパティの下にある更新されたアイコンマップの ID を含むオブジェクトを返します

例

アイコンマップの名前変更

アイコンマップの名前を”OS アイコン” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- アイコンマッピング

ソース

CIconMap::update() in *ui/include/classes/api/services/CIconMap.php*.

アイテム

このクラスはアイテムを操作するために設計されています。

オブジェクト参照:

- Item
  - HTTP header
  - HTTP query field
- Item tag
- Item preprocessing

利用可能なメソッド:

- item.create - 新しいアイテムを作成
- item.delete - アイテムを削除
- item.get - アイテムを取得
- item.update - アイテムを更新

Item オブジェクト

次のオブジェクトは、`itemAPI` に直接関連しています。

アイテム

> **Note:**
> Web アイテムは、Zabbix API を使用して直接作成、更新、削除することはできません。

アイテムオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
itemid	ID	アイテムの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
delay	string	アイテムの更新間隔。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはオプションで 1 つ以上のカスタム間隔を、すべてセミコロンで区切って指定できます。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロを使用できます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはマクロとテキストを混在させた値はサポートされません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が「Zabbix agent」(0)、「Simple check」(3)、「Zabbix internal」(5)、「External check」(10)、「Database monitor」(11)、「IPMI agent」(12)、「SSH agent」(13)、「TELNET agent」(14)、「Calculated」(15)、「JMX agent」(16)、「HTTP agent」(19)、「SNMP agent」(20)、「Script」(21)、「Browser」(22)、「Telemetry query」(24) に設定されている場合、または `type` が「Zabbix agent (active)」(7) に設定され、`key_` に「mqtt.get」が含まれていない場合は必須

プロパティ	タイプ	説明
hostid	ID	アイテムが属するホストまたはテンプレートの ID。 プロパティの動作: - 固定値 - 作成操作では必須
interfaceid	ID	アイテムのホストインターフェースの ID。 プロパティの動作: - アイテムがホストに属し、type が「Zabbix agent」、「IPMI agent」、「JMX agent」、「SNMP trap」、または「SNMP agent」に設定されている場合は必須 - アイテムがホストに属し、type が「Simple check」、「External check」、「SSH agent」、「TELNET agent」、または「HTTP agent」に設定されている場合はサポート - ディスカバリされたオブジェクトでは読み取り専用
key_	string	アイテムキー。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
name	string	アイテムの名前。 ユーザーマクロをサポートします。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
name_resolved	string	ユーザーマクロを解決したアイテムの名前。 プロパティの動作: - 読み取り専用
type	integer	アイテムのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が「HTTP agent」に設定されている場合は必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
value_type	integer	アイテムの情報タイプ。 指定可能な値: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary; 6 - JSON。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
allow_traps	integer	トラッパーアイテムと同様に値を格納できるようにします。 0 - *(デフォルト)* 受信データを受け付けない; 1 - 受信データを受け付ける。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - ディスカバリされたオブジェクトでは読み取り専用
authtype	integer	認証方式。 type が「SSH agent」に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が「HTTP agent」に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest。 プロパティの動作: - type が「SSH agent」または「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクト（type が「HTTP agent」に設定されている場合）またはディスカバリされたオブジェクトでは読み取り専用
description	string	アイテムの説明。 プロパティの動作: - ディスカバリされたオブジェクトでは読み取り専用
error	string	アイテム値の更新に問題がある場合のエラーテキスト。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
flags	integer	アイテムの生成元。 指定可能な値: 0 - 通常のアイテム; 4 - プロトタイプから変換されたアイテム。 プロパティの動作: - 読み取り専用
follow_redirects	integer	データのポーリング中にレスポンスのリダイレクトに従います。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
headers	array	HTTP リクエストの実行時に送信されるヘッダーの配列。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
history	string	履歴データを保存する期間の時間単位。 ユーザーマクロも使用できます。 デフォルト: 31d。 プロパティの動作: - ディスカバリされたオブジェクトでは読み取り専用
http_proxy	string	HTTP(S) プロキ接続文字列。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
inventory_link	integer	アイテムによって値が設定されるホストインベントリフィールドの ID。 サポートされるホストインベントリフィールドとその ID の一覧については、ホストインベントリページを参照してください。 デフォルト: 0。 プロパティの動作: - value_type が「numeric float」、「character」、「numeric unsigned」、または「text」に設定されている場合はサポート - ディスカバリされたオブジェクトでは読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が「IPMI agent」に設定され、key_ が「ipmi.get」に設定されていない場合は必須 - type が「IPMI agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト値: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が「JMX agent」に設定されている場合はサポート - ディスカバリされたオブジェクトでは読み取り専用
lastclock	timestamp	アイテム値が最後に更新された時刻。 デフォルトでは、過去 24 時間以内の値のみが表示されます。 *Administration > General*メニューセクションの *Max history display period* パラメーターの値を変更すると、この期間を延長できます。 プロパティの動作: - 読み取り専用
lastns	integer	アイテム値が最後に更新された時刻のナノ秒。 デフォルトでは、過去 24 時間以内の値のみが表示されます。 *Administration > General*メニューセクションの *Max history display period* パラメーターの値を変更すると、この期間を延長できます。 プロパティの動作: - 読み取り専用
lastvalue	string	アイテムの最後の値。 デフォルトでは、過去 24 時間以内の値のみが表示されます。 *Administration > General*メニューセクションの *Max history display period* パラメーターの値を変更すると、この期間を延長できます。 プロパティの動作: - 読み取り専用
logtimefmt	string	ログエントリの時刻形式。 プロパティの動作: - value_type が「log」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
master_itemid	ID	マスターアイテムの ID。 プロパティの動作: - type が「Dependent item」に設定されている場合は必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 0 - *(デフォルト)* 生データを保存; 1 - JSON に変換。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
params	string	アイテムのタイプに応じた追加パラメーター: - SSH agent および TELNET agent アイテムで実行されるスクリプト; - データベースモニターアイテムの SQL クエリ; - 計算アイテムの数式; - スクリプトおよびブラウザーアイテムのスクリプト。 プロパティの動作: - `type` が「Database monitor」、「SSH agent」、「TELNET agent」、「Calculated」、「Script」、または「Browser」に設定されている場合は必須 - 継承されたオブジェクト（`type` が「Script」または「Browser」に設定されている場合）またはディスカバリされたオブジェクトでは読み取り専用
parameters	object/array	`type` が「Script」または「Browser」に設定されている場合の追加パラメーター。`name` および `value` プロパティを持つオブジェクトの配列で、`name` は一意でなければなりません。 プロパティの動作: - `type` が「Script」または「Browser」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
password	string	認証用パスワード。 プロパティの動作: - `type` が「JMX agent」に設定され、`username` が設定されている場合は必須 - `type` が「Simple check」、「SSH agent」、「TELNET agent」、「Database monitor」、または「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクト（`type` が「HTTP agent」に設定されている場合）またはディスカバリされたオブジェクトでは読み取り専用
post_type	integer	`posts` プロパティに保存される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
posts	string	HTTP(S) リクエスト本文のデータ。 プロパティの動作: - `type` が「HTTP agent」に設定され、`post_type` が「JSON data」または「XML data」に設定されている場合は必須 - `type` が「HTTP agent」に設定され、`post_type` が「Raw data」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
prevvalue	string	アイテムの前回の値。 デフォルトでは、過去 24 時間以内の値のみが表示されます。*Administration > General*メニューセクションの *Max history display period* パラメーターの値を変更すると、この期間を延長できます。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
privatekey	string	秘密鍵ファイルの名前。 プロパティの動作: - `type` が「SSH agent」に設定され、`authtype` が「public key」に設定されている場合は必須 - ディスカバリされたオブジェクトでは読み取り専用
publickey	string	公開鍵ファイルの名前。 プロパティの動作: - `type` が「SSH agent」に設定され、`authtype` が「public key」に設定されている場合は必須 - ディスカバリされたオブジェクトでは読み取り専用
query_fields	array	HTTP リクエストの実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
retrieve_mode	integer	保存するレスポンスの部分。 `request_method` が「GET」、「POST」、または「PUT」に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が「HEAD」に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が「SNMP agent」に設定されている場合は必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
ssl_key_password	string	SSL キーファイルのパスワード。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
state	integer	アイテムの状態。 指定可能な値: 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	アイテムのステータス。 指定可能な値: 0 - *(デフォルト)* 有効なアイテム; 1 - 無効なアイテム。
status_codes	string	カンマで区切った、必要な HTTP ステータスコードの範囲。 カンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
templateid	ID	親テンプレートアイテムの ID。 ヒント: hostid プロパティを使用して、アイテムが属するテンプレートを指定します。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 300s、1m）を指定できます。ユーザーマクロも使用できます。 指定可能な値の範囲: 1-600s。 デフォルト: "" - プロキシまたはグローバル設定を使用。 プロパティの動作: - type が「Zabbix agent」(0)、「Simple check」(3) に設定され、key_ が「vmware.」および「icmpping」で始まらない場合、「Zabbix agent (active)」(7)、「External check」(10)、「Database monitor」(11)、「SSH agent」(13)、「TELNET agent」(14)、「HTTP agent」(19)、「SNMP agent」(20) に設定され、snmp_oid が「walk[」または「get[」で始まる場合、「Script」(21)、または「Browser」(22) に設定されている場合はサポート - 継承されたオブジェクトおよびディスカバリされたオブジェクトでは読み取り専用 - デフォルト値が適用され、アイテムがアクセスできないプロキシまたはプロキシグループによって監視されるホストに属している場合は、グローバルタイムアウトが使用されます。

プロパティ	タイプ	説明
trapper_hosts	string	許可されたホスト。 プロパティの動作: - ディスカバリされたオブジェクトでは読み取り専用 - type が「Zabbix trapper」に設定されている場合、または type が「HTTP agent」に設定され、allow_traps が「Allow incoming data」に設定されている場合はサポート。 プロパティが省略された場合、{$TRAPPER.ALLOWED_HOSTS}ユーザーマクロ（デフォルト値）が自動的に設定されます。デフォルトでは、{$TRAPPER.ALLOWED_HOSTS} は 127.0.0.1,::1 です。 プロパティに空の文字列を設定すると、すべてのホストからの受信接続が拒否されます。
trends	string	トレンドデータを保存する期間の時間単位。 ユーザーマクロも使用できます。 デフォルト: 365d。 プロパティの動作: - value_type が「numeric float」または「numeric unsigned」に設定されている場合はサポート - ディスカバリされたオブジェクトでは読み取り専用
units	string	値の単位。 プロパティの動作: - value_type が「numeric float」または「numeric unsigned」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
username	string	認証用ユーザー名。 プロパティの動作: - type が「SSH agent」または「TELNET agent」に設定されている場合、または type が「JMX agent」に設定され、password が設定されている場合は必須 - type が「Simple check」、「Database monitor」、または「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクト（type が「HTTP agent」に設定されている場合）またはディスカバリされたオブジェクトでは読み取り専用
uuid	string	インポートしたアイテムを既存のアイテムにリンクするために使用する一意の識別子。指定されていない場合は自動生成されます。 プロパティの動作: - アイテムがテンプレートに属している場合はサポート
valuemapid	ID	関連付けられた値のマップの ID。 プロパティの動作: - value_type が「numeric float」、「character」、または「numeric unsigned」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
verify_host	integer	接続先のホスト名が、ホストの証明書に記載されたホスト名と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
verify_peer	integer	ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合はサポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
query	object	Telemetry queryの設定。アイテムがテレメトリデータソースから何をクエリするか、データをどのようにフィルタリングするか、結果をどのように集計するかを定義します。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合は必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは読み取り専用
time_shift	string	アイテムがクエリする時間範囲の終了時刻を、現在時刻よりどれだけ過去に設定するか。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。ユーザーマクロも使用できます。 指定可能な値の範囲: 0-86400s。 デフォルト: 15s。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合はサポート
lookback_limit	string	継続する前回のクエリがない場合（例: アイテムの初回実行時）、または前回のクエリが `delay` より古い場合（例: Zabbix サーバーが停止していた場合）に、アイテムがデータをクエリする過去の期間。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。ユーザーマクロも使用できます。 指定可能な値の範囲: 1-259200s。 デフォルト: 10m。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合はサポート
granularity	string	クエリの全時間範囲を分割する単位の長さ。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。ユーザーマクロも使用できます。 指定可能な値の範囲: 1-86400s。 デフォルト: 15s。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合はサポート

HTTP ヘッダー

ヘッダーオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

テレメトリクエリ

テレメトリクエリオブジェクトには、次のプロパティがあります。

Property	Type	Description
signal_type	integer	アイテムがクエリするシグナルのタイプ。 指定可能な値: 0 - トレース; 1 - メトリクス; 2 - ログ。 プロパティの動作: - 必須
metric_point_type	integer	アイテムがクエリするメトリックポイントのタイプ。 指定可能な値: 0 - Sum; 1 - Gauge; 2 - Histogram; 3 - Exponential histogram。 プロパティの動作: - `signal_type` が”Metrics” に設定されている場合は必須
columns	array	アイテムがクエリするテレメトリデータ列の配列。空にすることもできます。 プロパティの動作: - 必須
aggregated_columns	array	アイテムがクエリするテレメトリデータ集約列の配列、および列レコードから値を計算する方法。 プロパティの動作: - 必須
filter	object	クエリ内の列レコードに対するテレメトリクエリフィルター。 プロパティの動作: - 必須

テレメトリデータ列

テレメトリデータ列オブジェクトには、以下のプロパティがあります。

Property	Type	Description
column	string	アイテムがクエリするテレメトリデータ列の名前。 詳細は、サポートされているテレメトリデータ列を参照してください。 複数の単一値列の場合、各 `column` 値は一意である必要があります。キーと値のペアを保持する複数の列の場合、各 `column` と `attribute_key` の組み合わせは一意である必要があります。
attribute_key	string	キーと値のペアを保持する列内で、アイテムがクエリするキーの名前。 キーと値のペアを保持する複数の列の場合、各 `column` と `attribute_key` の組み合わせは一意である必要があります。 プロパティの動作: - `column` にキーと値のペアを保持する列が設定されている場合は必須

テレメトリデータ集計列

テレメトリデータ集計列オブジェクトには、次のプロパティがあります。

Property	Type	Description
column	string	アイテムがクエリするテレメトリデータ列の名前。 サポートされているテレメトリデータ列を参照してください。 複数の列を指定する場合、各 `column` 値は一意である必要があります。 `function` に「Count」を設定した場合、このプロパティはサポートされません。 プロパティの動作: - `function` に「Min」、「Max」、「Average」、「Sum」、または「Percentile」を設定した場合は必須
function	integer	アイテムが列のレコードに適用する関数。 指定可能な値: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile。 プロパティの動作: - 必須
parameters	array	アイテムが列のレコードから計算するパーセンタイル。 指定可能な値: 文字列として指定する 0 から 100 までの数値。 プロパティの動作: - `function` に「Percentile」を設定した場合は必須
alias	string	計算値を識別するためにアイテムの値で使用する、一意の名前。 各 `alias` 値は一意である必要があります。 プロパティの動作: - 必須

テレメトリデータフィルター

テレメトリデータフィルターオブジェクトには、次のプロパティがあります。

Property	Type	Description
conditions	array	テレメトリデータ列のレコードが満たす必要があるテレメトリデータフィルターの条件。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	フィルター条件の評価に使用される生成された式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 プロパティの動作: - 読み取り専用
formula	string	カスタム式を使用するフィルターの条件評価に使用する、ユーザー定義の式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれている必要があります。式で使用する ID は、フィルター条件で定義されているものと完全に一致する必要があります。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

テレメトリデータのフィルター条件

テレメトリデータのフィルター条件オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
column	string	フィルター条件を適用するテレメトリデータ列の名前。 サポートされているテレメトリデータ列を参照してください。 プロパティの動作: - 必須
attribute_key	string	キーと値のペアを保持する列内で、条件を適用するキーの名前。 プロパティの動作: - `column` にキーと値のペアを保持する列が設定されている場合は必須
operator	integer	フィルター条件の演算子。 単一の値を保持する列で使用可能な値: 0 - 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない。 キーと値のペアを保持する列で使用可能な値: 0 - 等しい; 1 - 等しくない; 12 - 存在する。 プロパティの動作: - 必須

プロパティ	型	説明
value	string	比較対象の値。 プロパティの動作: - `operator` に「等しい」、「等しくない」、「含む」または「含まない」が設定されている場合は必須
formulaid	string	カスタム式から条件を参照するために使用する任意の一意な ID。大文字のみを含めることができます。フィルター条件を変更するときはユーザーが ID を定義する必要がありますが、その後に条件をリクエストすると新しく生成されます。 プロパティの動作: - テレメトリクエリフィルターオブジェクトの `evaltype` が「カスタム式」に設定されている場合は必須

アイテムタグ

アイテムタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	アイテムタグ名。 プロパティの動作: - 必須
value	string	アイテムタグの値。 プロパティの動作: - 読み取り専用。
object	integer	タグの継承元オブジェクトのタイプ。 指定可能な値: 0 - テンプレート; 1 - ホスト。 プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

アイテムの前処理

アイテムの前処理オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
type	integer	前処理オプションのタイプ。 指定可能な値: 1 - カスタム乗数; 2 - 右側のトリミング; 3 - 左側のトリミング; 4 - トリミング; 5 - 正規表現; 6 - ブール値から 10 進数; 7 - 8 進数から 10 進数; 8 - 16 進数から 10 進数; 9 - 単純な差分; 10 - 1 秒あたりの差分; 11 - XML XPath; 12 - JSONPath; 13 - 範囲内; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラーをチェック; 17 - XML のエラーをチェック; 18 - 正規表現を使用してエラーをチェック; 19 - 変更されていない値を破棄; 20 - ハートビート付きで変更されていない値を破棄; 21 - JavaScript; 22 - Prometheus パターン; 23 - Prometheus から JSON; 24 - CSV から JSON; 25 - 置換; 26 - サポートされていないアイテムをチェック; 27 - XML から JSON; 28 - SNMP walk 値; 29 - SNMP walk から JSON; 30 - SNMP get 値。 プロパティの動作: - 必須
params	string	前処理オプションで使用する追加パラメータ。 複数のパラメータは改行（\n）文字で区切ります。 type が「サポートされていないアイテムをチェック」に設定されている場合、パラメータは <scope>[\n<pattern>] 構文に従います。ここで、*pattern* は正規表現、*scope* は次のいずれかです: -1 - 任意のエラーに一致; 0 - エラーメッセージが *pattern* に一致するかをチェック; 1 - エラーメッセージが *pattern* に一致しないかをチェック。 プロパティの動作: - type が「カスタム乗数」(1)、「右側のトリミング」(2)、「左側のトリミング」(3)、「トリミング」(4)、「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「範囲内」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーをチェック」(16)、「XML のエラーをチェック」(17)、「正規表現を使用してエラーをチェック」(18)、「ハートビート付きで変更されていない値を破棄」(20)、「JavaScript」(21)、「Prometheus パターン」(22)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「置換」(25)、「サポートされていないアイテムをチェック」(26)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は 必須

プロパティ	タイプ	説明
error_handler	integer	前処理ステップが失敗した場合に使用するアクションタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定される; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 type が「サポートされていないアイテムをチェック」に設定されている場合の指定可能な値: 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が「カスタム乗数」(1)、「正規表現」(5)、「ブール値から 10 進数」(6)、「8 進数から 10 進数」(7)、「16 進数から 10 進数」(8)、「単純な差分」(9)、「1 秒あたりの差分」(10)、「XML XPath」(11)、「JSONPath」(12)、「範囲内」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーをチェック」(16)、「XML のエラーをチェック」(17)、「正規表現を使用してエラーをチェック」(18)、「Prometheus パターン」(22)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「サポートされていないアイテムをチェック」(26)、「XML から JSON」(27)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler_params	string	エラーハンドラーのパラメータ。 プロパティの動作: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は 必須

各前処理タイプで、次のパラメータとエラーハンドラーがサポートされています。

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
1	カスタム乗数	number^1, 6^			0, 1, 2, 3
2	右側のトリミング	文字のリスト[2]			
3	左側のトリミング	文字のリスト[2]			
4	トリミング	文字のリスト[2]			
5	正規表現	パターン[3]	出力[2]		0, 1, 2, 3
6	ブール値から 10 進数				0, 1, 2, 3

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
7	8 進数から 10 進数				0, 1, 2, 3
8	16 進数から 10 進数				0, 1, 2, 3
9	単純な差分				0, 1, 2, 3
10	1 秒あたりの差分				0, 1, 2, 3
11	XML XPath	パス[4]			0, 1, 2, 3
12	JSONPath	パス[4]			0, 1, 2, 3
13	範囲内	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	正規表現に一致	パターン[3]			0, 1, 2, 3
15	正規表現に一致しない	パターン[3]			0, 1, 2, 3
16	JSON のエラーをチェック	パス[4]			0, 1, 2, 3
17	XML のエラーをチェック	パス[4]			0, 1, 2, 3
18	正規表現を使用してエラーをチェック	パターン[3]	出力[2]		0, 1, 2, 3
19	変更されていない値を破棄				

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
20	ハートビート付きで変更されていない値を破棄	秒 ^5, 6^			
21	JavaScript	スクリプト[2]			
22	Prometheus パターン	パターン ^6, 7^	`value`、`label`、`function`	出力 ^8, 9^	0, 1, 2, 3
23	Prometheus から JSON	パターン ^6, 7^			0, 1, 2, 3
24	CSV から JSON	文字[2]	文字[2]	0,1	0, 1, 2, 3
25	置換	検索文字列[2]	置換後[2]		
26	サポートされていないアイテムをチェック	scope[1]	パターン ^3, 6^		1, 2, 3
27	XML から JSON				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON[10]	フィールド名[2]	OID プレフィックス[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 浮動小数点数（整数は暗黙的に浮動小数点値に変換されます）
 [2] 文字列
 [3] 正規表現
 [4] JSONPath または XML

XPath
 [5] 正の整数（30s、1m、2h、1d などの時間サフィックスをサポート）
 [6] ユーザーマクロ
 [7] 次の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。Prometheus パターンの各コンポーネント（メトリック、ラベル名、ラベル値、メトリック値）にはユーザーマクロを使用できます。
 [8] 2 番目のパラメータで `label` が選択されている場合の、次の構文に従う Prometheus 出力: `<label name>`（ユーザーマクロを使用可能）。
 [9] 2 番目のパラメータで `function` が選択されている場合の集計関数のいずれか: `sum`、`min`、`max`、`avg`、`count`。
 [10] 改行文字で区切られた複数の「フィールド名、OID プレフィックス、形式」のレコードをサポートします。

item.create

説明

`object item.create(object/array items)`

このメソッドを使用すると、新しいアイテムを作成できます。

> **Note:**
> Web アイテムは Zabbix API 経由では作成できません。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はユーザー権限 を参照してください。

パラメータ

(`object/array`) 作成するアイテム。

標準のアイテムプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
preprocessing	array	アイテムの前処理オプション。
tags	array	アイテムタグ。

戻り値

(`object`) 作成されたアイテムの ID を `itemids` プロパティに含むオブジェクトを返します。返される ID の順序は、渡されたアイテムの順序と一致します。

例

アイテムの作成

ID "30074" のホストで空きディスク容量を監視するために、2 つのアイテムタグを持つ数値型 Zabbix エージェントアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Free disk space on /home/joe/",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "30074",
        "type": 0,
        "value_type": 3,
        "interfaceid": "30084",
        "tags": [
            {
                "tag": "component",
                "value": "storage"
            },
            {
                "tag": "equipment",
                "value": "workstation"
            }
```

```
        ],
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

ホストインベントリアイテムの作成

ホストの「OS」インベントリフィールドを埋めるために、Zabbix エージェントアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "uname",
        "key_": "system.uname",
        "hostid": "30021",
        "type": 0,
        "interfaceid": "30007",
        "value_type": 1,
        "delay": "10s",
        "inventory_link": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

前処理付きアイテムの作成

カスタム乗数を使用してアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Device uptime",
        "key_": "sysUpTime",
        "hostid": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
```

```
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

従属アイテムの作成

ID 24759 のマスターアイテムに対して従属アイテムを作成します。同じホストでのみ依存関係が許可されているため、マスターアイテムと従属アイテムは同じ hostid である必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
      "hostid": "30074",
      "name": "Dependent test item",
      "key_": "dependent.item",
      "type": 18,
      "master_itemid": "24759",
      "value_type": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムの作成

JSON レスポンスの前処理を行う POST リクエストメソッドアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
```

```
    "params": {
        "url":"http://127.0.0.1/http.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value": "100"
            }
        ],
        "interfaceid": "1",
        "type": 19,
        "hostid": "10254",
        "delay": "5s",
        "key_": "json",
        "name": "HTTP agent example JSON",
        "value_type": 0,
        "output_format": 1,
        "preprocessing": [
            {
                "type": 12,
                "params": "$.random",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

スクリプトアイテムの作成

スクリプトアイテムを使用してシンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.item",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
```

```
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItem::create() in *ui/include/classes/api/services/CItem.php*.

item.delete

説明

`object item.delete(array itemIds)`

このメソッドを使用すると、アイテムを削除できます。

Note:
Web アイテムは Zabbix API から削除できません。

Note:
このメソッドは *Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`array`) 削除するアイテムの ID。

戻り値

(`object`) `itemids` プロパティの下にある削除されたアイテムの ID を含むオブジェクトを返します。

例

複数のアイテムの削除

2 つのアイテムを削除します。
 マスターアイテムが削除されると、依存アイテムとアイテムのプロトタイプは自動的に削除されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

ソース

CItem::delete() in *ui/include/classes/api/services/CItem.php*.

item.get

説明

`integer/array item.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってアイテムを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

Parameter	Type	Description
itemids	ID/array	指定した ID のアイテムのみを返します。
groupids	ID/array	指定したグループのホストに属するアイテムのみを返します。
templateids	ID/array	指定したテンプレートに属するアイテムのみを返します。
hostids	ID/array	指定したホストに属するアイテムのみを返します。
proxyids	ID/array	指定したプロキシによって監視されているアイテムのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用するアイテムのみを返します。
graphids	ID/array	指定したグラフで使用されているアイテムのみを返します。
triggerids	ID/array	指定したトリガーで使用されているアイテムのみを返します。
webitems	flag	結果に web アイテムを含めます。
inherited	boolean	`true` に設定すると、テンプレートから継承されたアイテムのみを返します。
inheritedTags	boolean	指定した `tags` を持ち、さらにホスト/テンプレート/リンクされたテンプレートにも同じタグがあるアイテムを返します。 指定可能な値: `true` - テンプレート/ホスト/リンクされたテンプレートにも指定したタグが存在している必要があります。 `false` - *(デフォルト)* テンプレート/ホスト/リンクされたテンプレートのタグは無視されます。
templated	boolean	`true` に設定すると、テンプレートに属するアイテムのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効なアイテムのみを返します。
group	string	指定した名前のグループに属するアイテムのみを返します。
host	string	指定した名前のホストに属するアイテムのみを返します。

Parameter	Type	Description
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	array	指定したタグを持つアイテムのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのアイテムを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist。
with_triggers	boolean	`true` に設定すると、トリガーで使用されているアイテムのみを返します。
selectHosts	query	アイテムが属するホストの配列を持つ`hosts` プロパティを返します。
selectInterfaces	query	アイテムが使用するホストインターフェースの配列を持つ`interfaces` プロパティを返します。
selectTriggers	query	アイテムが使用されているトリガーを持つ`triggers` プロパティを返します。 `count` をサポートします。
selectGraphs	query	アイテムを含むグラフを持つ`graphs` プロパティを返します。 `count` をサポートします。
selectDiscoveryData	query	アイテムディスカバリオブジェクトのデータを持つ `discoveryData` プロパティを返します。アイテムディスカバリオブジェクトは、ディスカバリされたアイテムを、そのアイテムの検出元であるアイテムプロトタイプに関連付けます。 このオブジェクトは以下のプロパティを持ちます: `parent_itemid` - (`string`) アイテムの作成元となったアイテムプロトタイプの ID。 `key_` - (`string`) アイテムプロトタイプのキー。 `status` - (`int`) アイテムディスカバリのステータス: 0 - *(デフォルト)* アイテムはディスカバリされています。 1 - アイテムはすでにディスカバリされていません。 `ts_delete` - (`timestamp`) すでにディスカバリされていないアイテムが削除される時刻。 `ts_disable` - (`timestamp`) すでにディスカバリされていないアイテムが無効化される時刻。 `disable_source` - (`int`) アイテムが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(デフォルト)* 自動的に無効化。 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	アイテムを作成した LLD ルールを持つ`discoveryRule` プロパティを返します。
selectInheritedTags	query	テンプレート/ホスト/リンクされたテンプレート上のタグを持つ`inheritedTags` プロパティを返します。
selectPreprocessing	query	アイテムの前処理オプションを持つ`preprocessing` プロパティを返します。
selectTags	query	`tags` プロパティでアイテムタグを返します。
selectValueMap	query	アイテムの値マップを持つ`valuemap` プロパティを返します。

Parameter	Type	Description
filter	object	指定したフィルターに完全一致する結果のみを返します。 プロパティ名をキーとし、値として単一の値または照合対象の値の配列を指定するオブジェクトを受け付けます。 text data type のプロパティはサポートしません。 追加のプロパティをサポートします: host - アイテムが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectGraphs - 結果は name でソートされます。 selectTriggers - 結果は description でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: itemid, name, key_, delay, history, trends, type, status。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	アイテムディスカバリオブジェクトを持つ itemDiscovery プロパティを返します。アイテムディスカバリオブジェクトは、アイテムをその作成元であるアイテムプロトタイプに関連付けます。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

キーによるアイテムの検索

特定のホスト ID のトリガーで使用されているアイテムのうち、アイテムキーに「system.cpu」を含むものを取得し、名前でソートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
```

```
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "42269",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "CPU utilization",
            "key_": "system.cpu.util",
            "delay": "0",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "%",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42267",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "CPU utilization in %.",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "42264",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
```

```
            "state": "0",
            "error": "",
            "name_resolved": "CPU utilization",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42259",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (15m avg)",
            "key_": "system.cpu.load[all,avg15]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42219",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
```

```
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (15m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42249",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (1m avg)",
            "key_": "system.cpu.load[all,avg1]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42209",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
```

```
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (1m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42257",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (5m avg)",
            "key_": "system.cpu.load[all,avg5]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42217",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
```

```
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Load average (5m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42260",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Number of CPUs",
            "key_": "system.cpu.num",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
```

```
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Number of CPUs",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


キーで依存アイテムを検索する

キーに「apache」という単語が含まれている、ID「10116」のホストからすべての依存アイテムを取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25550",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Days",
            "key_": "apache.status.uptime.days",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
```

```
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Days",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "25555",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Hours",
            "key_": "apache.status.uptime.hours",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
```

```
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "Hours",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


HTTP エージェントアイテムの検索

特定のホスト ID に対して、ポストボディタイプが XML の HTTP エージェントアイテムを検索します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
```

```
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28252",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10255",
            "name": "template item",
            "key_": "ti",
            "delay": "30s",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "localhost",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "xml"
                }
            ],
            "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",
            "status_codes": "200",
            "follow_redirects": "0",
            "post_type": "3",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "1",
            "request_method": "3",
            "output_format": "0",
```

```
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "name_resolved": "template item",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
            "prevvalue": ""
        }
    ],
    "id": 1
}
```


前処理ルールを持つアイテムの取得

特定のホスト ID のすべてのアイテムとその前処理ルールを取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```


関連項目

- ディスカバリルール
- グラフ
- ホスト
- ホストインターフェース

- トリガー

ソース

CItem::get() in *ui/include/classes/api/services/CItem.php*.

item.update

説明

`object item.update(object/array items)`

このメソッドを使用すると、既存のアイテムを更新できます。

Note:
Web アイテムは Zabbix API 経由では更新できません。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するアイテムのプロパティ。

各アイテムに対して `itemid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のアイテムプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
preprocessing	array	現在の前処理オプションを置き換えるためのアイテムの前処理オプション。 パラメータの動作: - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは *read-only*
tags	array	アイテムタグ。 パラメータの動作: - ディスカバリされたオブジェクトでは *read-only*

戻り値

(`object`) `itemids` プロパティの下で更新されたアイテムの ID を含むオブジェクトを返します。

例

アイテムの有効化

アイテムを有効にします。つまり、ステータスを”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

従属アイテムの更新

従属アイテム名とマスターアイテム ID を更新します。同じホストでのみ依存関係が許可されているため、マスターアイテムと従属アイテムは同じ hostid である必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": " 従属アイテムの更新名",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムの更新

アイテム値のトラップを有効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

前処理を使用したアイテムの更新

アイテムの前処理ルール「範囲内」でアイテムを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

スクリプトアイテムの更新

異なるスクリプトでスクリプトアイテムを更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItem::update() in *ui/include/classes/api/services/CItem.php*.

アイテムプロトタイプ

このクラスはアイテムプロトタイプを操作するために設計されています。

オブジェクト参照:

- アイテムプロトタイプ
  - HTTP ヘッダー
  - HTTP クエリフィールド
- アイテムプロトタイプタグ
- アイテムプロトタイプの前処理

利用可能なメソッド:

- itemprototype.create - 新しいアイテムプロトタイプを作成
- itemprototype.delete - アイテムプロトタイプを削除
- itemprototype.get - アイテムプロトタイプを取得
- itemprototype.update - アイテムプロトタイプを更新

Item prototype オブジェクト

次のオブジェクトは、`itemprototype`API に直接関連しています。

アイテムのプロトタイプ

アイテムのプロトタイプオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
itemid	ID	アイテムのプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
delay	string	アイテムのプロトタイプの更新間隔。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはオプションで 1 つ以上のカスタム間隔を指定できます。これらはすべてセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロおよび LLD マクロを使用できます。使用する場合、値は単一のマクロである必要があります。複数のマクロ、またはマクロとテキストを組み合わせた値はサポートされません。柔軟な間隔は、スラッシュで区切った 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が「Zabbix agent」(0)、「Simple check」(3)、「Zabbix internal」(5)、「External check」(10)、「Database monitor」(11)、「IPMI agent」(12)、「SSH agent」(13)、「TELNET agent」(14)、「Calculated」(15)、「JMX agent」(16)、「HTTP agent」(19)、「SNMP agent」(20)、「Script」(21)、「Browser」(22)、「Telemetry query」(24) に設定されている場合、または `type` が「Zabbix agent (active)」(7) に設定され、`key_` に「mqtt.get」が含まれていない場合は必須
hostid	ID	アイテムのプロトタイプが属するホストの ID。 プロパティの動作: - 固定 - 作成操作では必須

プロパティ	タイプ	説明
interfaceid	ID	アイテムのプロトタイプのホストインターフェースの ID。 プロパティの動作: - アイテムのプロトタイプがホストに属し、type が「Zabbix agent」、「IPMI agent」、「JMX agent」、「SNMP trap」、または「SNMP agent」に設定されている場合は必須 - アイテムのプロトタイプがホストに属し、type が「Simple check」、「External check」、「SSH agent」、「TELNET agent」、または「HTTP agent」に設定されている場合はサポート
key_	string	アイテムのプロトタイプのキー。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用
name	string	アイテムのプロトタイプの名前。 ユーザーマクロをサポートします。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用
type	integer	アイテムのプロトタイプのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用
url	string	URL 文字列。 LLD マクロ、ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が「HTTP agent」に設定されている場合は必須 - 継承されたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
value_type	integer	アイテムのプロトタイプの情報タイプ。 指定可能な値: 0 - numeric float; 1 - character; 2 - log; 3 - numeric unsigned; 4 - text; 5 - binary; 6 - JSON。 プロパティの動作: - 作成操作では必須 - 継承されたオブジェクトでは読み取り専用
allow_traps	integer	トラッパーアイテムと同様に値を格納できるようにします。 0 - *(デフォルト)* 受信データを受け付けない; 1 - 受信データを受け付ける。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート
authtype	integer	認証方式。 type が「SSH agent」に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が「HTTP agent」に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest。 プロパティの動作: - type が「SSH agent」または「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用（type が「HTTP agent」に設定されている場合）
description	string	アイテムのプロトタイプの説明。
flags	integer	アイテムのプロトタイプの起点。 指定可能な値: 2 - アイテムのプロトタイプ; 6 - 検出されたアイテムのプロトタイプ プロパティの動作: - 読み取り専用
follow_redirects	integer	データのポーリング中にレスポンスのリダイレクトに従うかどうか。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
headers	array	HTTP リクエストの実行時に送信されるヘッダーの配列。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
history	string	履歴データを保存する期間の時間単位。 ユーザーマクロおよび LLD マクロも使用できます。 デフォルト: 31d。
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が「IPMI agent」に設定され、key_ が「ipmi.get」に設定されていない場合は必須 - type が「IPMI agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が「JMX agent」に設定されている場合にサポート
logtimefmt	string	ログエントリの時刻形式。 プロパティの動作: - value_type が「log」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
master_itemid	ID	マスターアイテムの ID。 プロパティの動作: - type が「Dependent item」に設定されている場合は必須 - 継承されたオブジェクトでは読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データとして保存; 1 - JSON に変換。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
params	string	アイテムのプロトタイプのタイプに応じた追加パラメータ: - SSH agent および TELNET agent のアイテムのプロトタイプで実行されるスクリプト; - データベースモニターのアイテムのプロトタイプで使用される SQL クエリ; - 計算アイテムのプロトタイプで使用される式; - スクリプトおよびブラウザーのアイテムのプロトタイプで使用されるスクリプト。 プロパティの動作: - type が「Database monitor」、「SSH agent」、「TELNET agent」、「Calculated」、「Script」、または「Browser」に設定されている場合は必須 - 継承されたオブジェクトでは読み取り専用（type が「Script」または「Browser」に設定されている場合）

プロパティ	タイプ	説明
parameters	object/array	type が「Script」または「Browser」に設定されている場合の追加パラメータ。name および value プロパティを持つオブジェクトの配列で、name は一意である必要があります。 プロパティの動作: - type が「Script」または「Browser」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
password	string	認証用パスワード。 プロパティの動作: - type が「JMX agent」に設定され、username が設定されている場合は必須 - type が「Simple check」、「SSH agent」、「TELNET agent」、「Database monitor」、または「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用（type が「HTTP agent」に設定されている場合）
post_type	integer	posts プロパティに保存される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ。 2 - JSON データ。 3 - XML データ。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
posts	string	HTTP(S) リクエスト本文のデータ。 プロパティの動作: - type が「HTTP agent」に設定され、post_type が「JSON data」または「XML data」に設定されている場合は必須 - type が「HTTP agent」に設定され、post_type が「Raw data」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
privatekey	string	秘密鍵ファイルの名前。 プロパティの動作: - type が「SSH agent」に設定され、authtype が「public key」に設定されている場合は必須
publickey	string	公開鍵ファイルの名前。 プロパティの動作: - type が「SSH agent」に設定され、authtype が「public key」に設定されている場合は必須
query_fields	array	HTTP リクエストの実行時に送信されるクエリフィールドの配列。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - type が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用

プロパティ	タイプ	説明
retrieve_mode	integer	保存するレスポンスの部分。 `request_method` が「GET」、「POST」、または「PUT」に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が「HEAD」に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が「SNMP agent」に設定されている場合は必須 - 継承されたオブジェクトでは読み取り専用
ssl_cert_file	string	公開 SSL キーのファイルパス。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ssl_key_file	string	秘密 SSL キーのファイルパス。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
ssl_key_password	string	SSL キーファイルのパスワード。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
status	integer	アイテムのプロトタイプのステータス。 指定可能な値: 0 - *(デフォルト)* 有効なアイテムのプロトタイプ; 1 - 無効なアイテムのプロトタイプ; 3 - サポートされていないアイテムのプロトタイプ。
status_codes	string	必要な HTTP ステータスコードの範囲。コンマで区切ります。 コンマ区切りリストの一部として、ユーザーマクロまたは LLD マクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
templateid	ID	親テンプレートのアイテムのプロトタイプの ID。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m）を指定できます。ユーザーマクロおよび LLD マクロも使用できます。 指定可能な値の範囲: 1-600s。 デフォルト: "" - プロキシまたはグローバル設定を使用。 プロパティの動作: - type が「Zabbix agent」(0)、「Simple check」(3) に設定され、key_ が「vmware.」および「icmpping」で始まらない場合、「Zabbix agent (active)」(7)、「External check」(10)、「Database monitor」(11)、「SSH agent」(13)、「TELNET agent」(14)、「HTTP agent」(19)、「SNMP agent」(20) に設定され、snmp_oid が「walk[」または「get[」で始まる場合、「Script」(21)、または「Browser」(22) に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - type が「Zabbix trapper」に設定されている場合、または type が「HTTP agent」に設定され、allow_traps が「Allow incoming data」に設定されている場合にサポート。 プロパティを省略すると、{$TRAPPER.ALLOWED_HOSTS} ユーザーマクロ（デフォルト値）が自動的に設定されます。デフォルトでは、{$TRAPPER.ALLOWED_HOSTS} は 127.0.0.1,::1 です。 プロパティに空の文字列を設定すると、すべてのホストからの受信接続が拒否されます。
trends	string	トレンドデータを保存する期間の時間単位。 ユーザーマクロおよび LLD マクロも使用できます。 デフォルト: 365d。 プロパティの動作: - value_type が「numeric float」または「numeric unsigned」に設定されている場合にサポート
units	string	値の単位。 プロパティの動作: - value_type が「numeric float」または「numeric unsigned」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
username	string	認証用ユーザー名。 プロパティの動作: - type が「SSH agent」または「TELNET agent」に設定されている場合、または type が「JMX agent」に設定され、password が設定されている場合は必須 - type が「Simple check」、「Database monitor」、または「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用（type が「HTTP agent」に設定されている場合）
uuid	string	インポートしたアイテムのプロトタイプを既存のアイテムのプロトタイプにリンクするために使用される一意の識別子。指定されていない場合は自動生成されます。 プロパティの動作: - アイテムのプロトタイプがテンプレートに属している場合にサポート

プロパティ	タイプ	説明
valuemapid	ID	関連付けられた値のマップの ID。 プロパティの動作: - `value_type` が「numeric float」、「character」、または「numeric unsigned」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
verify_host	integer	接続先のホスト名が、ホストの証明書に記載された名前と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
verify_peer	integer	ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が「HTTP agent」に設定されている場合にサポート - 継承されたオブジェクトでは読み取り専用
discover	integer	アイテムのプロトタイプの検出ステータス。 指定可能な値: 0 - *(デフォルト)* 新しいアイテムを検出する; 1 - 新しいアイテムを検出せず、既存のアイテムを失われたものとしてマークする。
query	object	Telemetry query の設定。アイテムがテレメトリデータソースに対して何をクエリするか、データをどのようにフィルタリングするか、結果をどのように集計するかを定義します。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合は必須 - 継承されたオブジェクトまたは検出されたオブジェクトでは読み取り専用
time_shift	string	アイテムがクエリする時間範囲の終了時刻を、現在時刻よりどれだけ過去にずらすか。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。ユーザーマクロおよび LLD マクロも使用できます。 指定可能な値の範囲: 0-86400s。 デフォルト: 15s。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合にサポート

プロパティ	タイプ	説明
lookback_limit	string	継続する前回のクエリがない場合（アイテムの初回実行時など）、または前回のクエリが `delay` より古い場合（Zabbix サーバーが停止していた場合など）に、アイテムがデータをクエリする過去の期間。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。ユーザーマクロおよび LLD マクロも使用できます。 指定可能な値の範囲: 1-259200s。 デフォルト: 10m。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合にサポート
granularity	string	クエリの全時間範囲を分割する各区間の長さ。 秒数（例: 1、60、300）、または時間のサフィックス付きの値（例: 30s、1m、2h、1d）を指定できます。ユーザーマクロおよび LLD マクロも使用できます。 指定可能な値の範囲: 1-86400s。 デフォルト: 15s。 プロパティの動作: - `type` が「Telemetry query」に設定されている場合にサポート

HTTP ヘッダー

ヘッダーオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

テレメトリクエリ

テレメトリクエリオブジェクトには、次のプロパティがあります。

Property	Type	Description
signal_type	integer	アイテムがクエリするシグナルのタイプ。 指定可能な値: 0 - トレース; 1 - メトリクス; 2 - ログ。 プロパティの動作: - 必須
metric_point_type	integer	アイテムがクエリするメトリクスポイントのタイプ。 指定可能な値: 0 - Sum; 1 - Gauge; 2 - Histogram; 3 - Exponential histogram。 プロパティの動作: - `signal_type` が”Metrics” に設定されている場合は必須
columns	array	アイテムがクエリするテレメトリデータ列の配列。空にすることもできます。 プロパティの動作: - 必須
aggregated_columns	array	アイテムがクエリするテレメトリデータ集約列の配列、および列レコードから値を計算する方法。 プロパティの動作: - 必須
filter	object	クエリ内の列レコードに対するテレメトリクエリフィルター。 プロパティの動作: - 必須

テレメトリデータ列

テレメトリデータ列オブジェクトには、次のプロパティがあります。

Property	Type	Description
column	string	アイテムがクエリするテレメトリデータ列の名前。 詳細は、サポートされているテレメトリデータ列を参照してください。 複数の単一値列の場合、各 `column` 値は一意である必要があります。キーと値のペアを保持する複数の列の場合、各 `column` と `attribute_key` の組み合わせは一意である必要があります。
attribute_key	string	キーと値のペアを保持する列内で、アイテムがクエリするキーの名前。 キーと値のペアを保持する複数の列の場合、各 `column` と `attribute_key` の組み合わせは一意である必要があります。 プロパティの動作: - `column` にキーと値のペアを保持する列が設定されている場合は必須

テレメトリデータ集計列

テレメトリデータ集計列オブジェクトには、次のプロパティがあります。

Property	Type	Description
column	string	アイテムがクエリするテレメトリデータ列の名前。 サポートされているテレメトリデータ列を参照してください。 複数の列を指定する場合、各 `column` 値は一意である必要があります。 `function` に「Count」が設定されている場合、このプロパティはサポートされません。 プロパティの動作: - `function` に「Min」、「Max」、「Average」、「Sum」、または「Percentile」が設定されている場合は必須
function	integer	アイテムが列のレコードに適用する関数。 指定可能な値: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile。 プロパティの動作: - 必須
parameters	array	アイテムが列のレコードから計算するパーセンタイル。 指定可能な値: 文字列として指定する 0 から 100 までの数値。 プロパティの動作: - `function` に「Percentile」が設定されている場合は必須
alias	string	計算値を識別するためにアイテムの値で使用される、一意の名前。 各 `alias` 値は一意である必要があります。 プロパティの動作: - 必須

テレメトリデータフィルター

テレメトリデータフィルターオブジェクトには、次のプロパティがあります。

Property	Type	Description
conditions	array	テレメトリデータ列のレコードが満たす必要があるテレメトリデータフィルター条件。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須

Property	Type	Description
eval_formula	string	フィルター条件の評価に使用される生成された式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 プロパティの動作: - 読み取り専用
formula	string	カスタム式を使用するフィルターの条件評価に使用する、ユーザー定義の式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれている必要があります。式で使用する ID は、フィルター条件で定義されている ID と完全に一致している必要があります。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は必須

テレメトリデータのフィルター条件

テレメトリデータのフィルター条件オブジェクトには、次のプロパティがあります。

Property	Type	Description
column	string	フィルター条件を適用するテレメトリデータ列の名前。 サポートされているテレメトリデータ列を参照してください。 プロパティの動作: - 必須
attribute_key	string	キーと値のペアを保持する列内で、条件を適用するキーの名前。 プロパティの動作: - `column` にキーと値のペアを保持する列が設定されている場合は必須
operator	integer	フィルター条件の演算子。 単一の値を持つ列で使用可能な値: 0 - 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない。 キーと値のペアを保持する列で使用可能な値: 0 - 等しい; 1 - 等しくない; 12 - 存在する。 プロパティの動作: - 必須
value	string	比較対象の値。 プロパティの動作: - `operator` に「等しい」、「等しくない」、「含む」または「含まない」が設定されている場合は必須
formulaid	string	カスタム式から条件を参照するために使用する任意の一意の ID。大文字のみを含めることができます。フィルター条件を変更するときはユーザーが ID を定義する必要がありますが、その後に条件を取得すると新たに生成されます。 プロパティの動作: - テレメトリクエリフィルターオブジェクトの `evaltype` が「カスタム式」に設定されている場合は必須

アイテムプロトタイプタグ

アイテムプロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	アイテムプロトタイプタグ名。 プロパティの動作: - 必須
value	string	アイテムプロトタイプタグの値。 プロパティの動作: - 読み取り専用。
object	integer	タグの継承元オブジェクトの種類。 指定可能な値: 0 - テンプレート; 1 - ホスト. プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

アイテムプロトタイプの前処理

アイテムプロトタイプの前処理オブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
type	integer	前処理オプションのタイプ。 指定可能な値: 1 - カスタム乗数; 2 - 右側トリム; 3 - 左側トリム; 4 - トリム; 5 - 正規表現; 6 - ブール値から 10 進数; 7 - 8 進数から 10 進数; 8 - 16 進数から 10 進数; 9 - 単純変化; 10 - 1 秒あたりの変化量; 11 - XML XPath; 12 - JSONPath; 13 - 範囲内; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON のエラーをチェック; 17 - XML のエラーをチェック; 18 - 正規表現を使用してエラーをチェック; 19 - 変更されていない値を破棄; 20 - ハートビート付きで変更されていない値を破棄; 21 - JavaScript; 22 - Prometheus パターン; 23 - Prometheus から JSON; 24 - CSV から JSON; 25 - 置換; 26 - サポートされていないアイテムをチェック; 27 - XML から JSON; 28 - SNMP walk 値; 29 - SNMP walk から JSON; 30 - SNMP get 値。 プロパティの動作: - 必須
params	string	前処理オプションで使用する追加パラメータ。 複数のパラメータは改行（\n）文字で区切ります。 type が「サポートされていないアイテムをチェック」に設定されている場合、パラメータは <scope>[\n<pattern>] 構文に従います。ここで、*pattern* は正規表現、*scope* は次のいずれかです: -1 - 任意のエラーに一致; 0 - エラーメッセージが *pattern* に一致するかチェック; 1 - エラーメッセージが *pattern* に一致しないかチェック。 プロパティの動作: - type が「カスタム乗数」(1)、「右側トリム」(2)、「左側トリム」(3)、「トリム」(4)、「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「範囲内」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーをチェック」(16)、「XML のエラーをチェック」(17)、「正規表現を使用してエラーをチェック」(18)、「ハートビート付きで変更されていない値を破棄」(20)、「JavaScript」(21)、「Prometheus パターン」(22)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「置換」(25)、「サポートされていないアイテムをチェック」(26)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は 必須

プロパティ	タイプ	説明
error_handler	integer	前処理ステップが失敗した場合に使用するアクションのタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定される; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 type が「サポートされていないアイテムをチェック」に設定されている場合の指定可能な値: 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - type が「カスタム乗数」(1)、「正規表現」(5)、「ブール値から 10 進数」(6)、「8 進数から 10 進数」(7)、「16 進数から 10 進数」(8)、「単純変化」(9)、「1 秒あたりの変化量」(10)、「XML XPath」(11)、「JSONPath」(12)、「範囲内」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON のエラーをチェック」(16)、「XML のエラーをチェック」(17)、「正規表現を使用してエラーをチェック」(18)、「Prometheus パターン」(22)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「サポートされていないアイテムをチェック」(26)、「XML から JSON」(27)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler_params	string	エラーハンドラーのパラメータ。 プロパティの動作: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は 必須

各前処理タイプでサポートされるパラメータとエラーハンドラーは次のとおりです。

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
1	カスタム乗数	number^1, 6^			0, 1, 2, 3
2	右側トリム	文字のリスト[2]			
3	左側トリム	文字のリスト[2]			
4	トリム	文字のリスト[2]			
5	正規表現	pattern[3]	output[2]		0, 1, 2, 3
6	ブール値から 10 進数				0, 1, 2, 3
7	8 進数から 10 進数				0, 1, 2, 3

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
8	16 進数から 10 進数				0, 1, 2, 3
9	単純変化				0, 1, 2, 3
10	1 秒あたりの変化量				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[4]			0, 1, 2, 3
13	範囲内	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	正規表現に一致	pattern[3]			0, 1, 2, 3
15	正規表現に一致しない	pattern[3]			0, 1, 2, 3
16	JSON のエラーをチェック	path[4]			0, 1, 2, 3
17	XML のエラーをチェック	path[4]			0, 1, 2, 3
18	正規表現を使用してエラーをチェック	pattern[3]	output[2]		0, 1, 2, 3
19	変更されていない値を破棄				

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
20	ハートビート付きで変更されていない値を破棄	seconds^5, 6^			
21	JavaScript	script[2]			
22	Prometheus パターン	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus から JSON	pattern^6, 7^			0, 1, 2, 3
24	CSV から JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
26	サポートされていないアイテムをチェック	scope[1]	pattern^3, 6^		1, 2, 3
27	XML から JSON				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON[10]	フィールド名[2]	OID プレフィックス[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 整数または浮動小数点数
 [2] 文字列
 [3] 正規表現
 [4] JSONPath または XML XPath
 [5] 正の整数（時間のサフィック

スをサポート。例: 30s、1m、2h、1d）
 [6] ユーザーマクロ、LLD マクロ
 [7] 次の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。Prometheus パターンの各コンポーネント（メトリック、ラベル名、ラベル値、メトリック値）には、ユーザーマクロまたは LLD マクロを指定できます。
 [8] 2 番目のパラメータで `label` が選択されている場合の Prometheus 出力。構文は `<label name>`（ユーザーマクロまたは LLD マクロを指定可能）。
 [9] 2 番目のパラメータで `function` が選択されている場合の集計関数のいずれか: `sum`、`min`、`max`、`avg`、`count`。
 [10] 改行文字で区切られた複数の「フィールド名、OID プレフィックス、形式レコード」をサポートします。

itemprototype.create

説明

`object itemprototype.create(object/array itemPrototypes)`

このメソッドを使用すると、新しいアイテムプロトタイプを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するアイテムのプロトタイプ。

標準のアイテムプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
ruleid	ID	このアイテムが属するLLD ルールの ID。 パラメータの動作: - 必須
preprocessing	array	アイテムプロトタイプの前処理オプション。
tags	array	アイテムプロトタイプのタグ。

戻り値

`(object)` `itemids` プロパティの下で作成されたアイテムプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムプロトタイプの順序と一致します。

例

アイテムプロトタイプの作成

検出されたファイルシステムの空きディスク容量を監視するアイテムプロトタイプを作成します。検出されたアイテムは、30 秒ごとに更新される数値型の Zabbix エージェントアイテムである必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Free disk space on {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

前処理付きアイテムプロトタイプの作成

1 秒あたりの変化量とカスタム乗数を 2 番目のステップとして使用してアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Incoming network traffic on {#IFNAME}",
        "key_": "net.if.in[{#IFNAME}]",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "delay": "60s",
        "units": "bps",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 10,
                "params": "",
                "error_handler": 0,
                "error_handler_params": ""
            },
            {
                "type": 1,
                "params": "8",
                "error_handler": 2,
                "error_handler_params": "10"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

依存アイテムプロトタイプの作成

ID 44211 のマスターアイテムプロトタイプに対して依存アイテムプロトタイプを作成します。同じホスト（テンプレート/ディスカバリールール）でのみ依存関係が許可されているため、マスターと依存アイテムは同じ hostid および ruleid を持つ必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
```

```
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Dependent test item prototype",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

HTTP エージェントのアイテムプロトタイプの作成

ユーザーマクロ、クエリフィールド、カスタムヘッダーを使用した URL でアイテムプロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "type": "19",
        "hostid": "10254",
        "ruleid": "28256",
        "interfaceid": "2",
        "name": "api item prototype example",
        "key_": "api_http_item",
        "value_type": 3,
        "url": "{$URL_PROTOTYPE}",
        "query_fields": [
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value" "100"
            }
        ],
        "headers": [
            {
                "name": "X-Source",
                "value": "api"
            }
        ],
        "delay": "35"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

スクリプトアイテムプロトタイプの作成

スクリプトアイテムプロトタイプを使用してシンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.itemprototype",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::create() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

説明

`object itemprototype.delete(array itemPrototypeIds)`

このメソッドを使用すると、アイテムプロトタイプを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するアイテムのプロトタイプの ID。

戻り値

(object) prototypeids プロパティの下にある削除されたアイテムのプロトタイプの ID を含むオブジェクトを返します。

例

複数のアイテムプロトタイプの削除

2 つのアイテムプロトタイプを削除します。
 マスターアイテムまたはアイテムプロトタイプが削除されると、依存アイテムプロトタイプは自動的に削除されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::delete() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

説明

`integer/array itemprototype.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってアイテムプロトタイプを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
discoveryids	ID/array	指定した LLD ルールに属するアイテムプロトタイプのみを返します。
graphids	ID/array	指定したグラフプロトタイプで使用されているアイテムプロトタイプのみを返します。
hostids	ID/array	指定したホストに属するアイテムプロトタイプのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承されたアイテムプロトタイプのみを返します。

Parameter	Type	Description
itemids	ID/array	指定した ID を持つアイテムプロトタイプのみを返します。
monitored	boolean	`true` に設定した場合、監視対象ホストに属する有効なアイテムプロトタイプのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属するアイテムプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するアイテムプロトタイプのみを返します。
triggerids	ID/array	指定したトリガープロトタイプで使用されているアイテムプロトタイプのみを返します。
selectDiscoveryData	query	アイテムプロトタイプのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。アイテムプロトタイプのディスカバリオブジェクトは、検出されたアイテムプロトタイプを、それが検出された元のアイテムプロトタイプに関連付けます。 以下のプロパティを持ちます。 `parent_itemid` - (`string`) アイテムプロトタイプの作成元となったアイテムプロトタイプの ID。 `key_` - (`string`) アイテムプロトタイプのキー。 `status` - (`int`) アイテムプロトタイプのディスカバリステータス: 0 - *(default)* アイテムプロトタイプは検出されています。 1 - アイテムプロトタイプはすでに検出されていません。 `ts_delete` - (`timestamp`) すでに検出されていないアイテムプロトタイプが削除される時刻。 `ts_disable` - (`timestamp`) すでに検出されていないアイテムプロトタイプが無効化される時刻。 `disable_source` - (`int`) アイテムプロトタイプが LLD ルールによって無効化されたか、手動で無効化されたかを示すインジケーター: 0 - *(default)* 自動的に無効化。 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	アイテムプロトタイプが属するローレベルディスカバリルールを含む`discoveryRule` プロパティを返します。
selectDiscoveryRulePrototype	query	アイテムプロトタイプが属する親 LLD ルールプロトタイプを含む`discoveryRulePrototype` プロパティを返します。
selectGraphs	query	アイテムプロトタイプが使用されているグラフプロトタイプを含む`graphs` プロパティを返します。 `count` をサポートします。
selectHosts	query	アイテムプロトタイプが属するホストの配列を含む`hosts` プロパティを返します。
selectInheritedTags	query	テンプレート/ホスト/リンクされたテンプレート上のタグを含む`inheritedTags` プロパティを返します。
selectTags	query	`tags` プロパティ内にアイテムプロトタイプのタグを返します。
selectTriggers	query	アイテムプロトタイプが使用されているトリガープロトタイプを含む`triggers` プロパティを返します。 `count` をサポートします。
selectPreprocessing	query	アイテムプロトタイプの前処理オプションを含む`preprocessing` プロパティを返します。
selectValueMap	query	アイテムプロトタイプの値マップを含む`valuemap` プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、値を照合対象の単一の値または値の配列とするオブジェクトを受け付けます。 `text` data type のプロパティはサポートしていません。 追加のプロパティをサポートします。 `host` - アイテムプロトタイプが属するホストの技術名。

Parameter	Type	Description
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます。 `selectGraphs` - 結果は `name` でソートされます。 `selectTriggers` - 結果は `description` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

LLD ルールからアイテムプロトタイプを取得する

特定の LLD ルール ID のすべてのアイテムプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23077",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10079",
            "name": "Incoming network traffic on en0",
            "key_": "net.if.in[en0]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "bps",
```

```
        "formula": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "2",
        "interfaceid": "0",
        "description": "",
        "inventory_link": "0",
        "lifetime": "7d",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "0",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "discover": "0",
        "uuid": "",
        "lifetime_type": "0",
        "enabled_lifetime_type": "2",
        "enabled_lifetime": "0",
        "parameters": []
    },
    {
        "itemid": "10010",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10001",
        "name": "Processor load (1 min average per core)",
        "key_": "system.cpu.load[percpu,avg1]",
        "delay": "1m",
        "history": "1w",
        "trends": "365d",
        "status": "0",
        "value_type": "0",
        "trapper_hosts": "",
        "units": "",
        "formula": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
```

```
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "The processor load is calculated as system CPU load divided by number of CPU c
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


依存アイテムの検索

特定のアイテム ID に対する依存アイテムを 1 つ検索します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```


レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25547",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Seconds",
            "key_": "apache.status.uptime.seconds",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
```

```
    "id": 1
}
```

HTTP エージェントアイテムプロトタイプの検索

特定のホスト ID に対して、リクエストメソッド HEAD の HTTP エージェントアイテムプロトタイプを検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28257",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10254",
            "name": "discovered",
            "key_": "item[{#INAME}]",
            "delay": "{#IUPDATE}",
            "history": "90d",
            "trends": "30d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "{#IURL}",
            "query_fields": [],
            "posts": "",
            "status_codes": "",
```

```
            "follow_redirects": "0",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- グラフプロトタイプ
- トリガープロトタイプ

ソース

CItemPrototype::get() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

説明

`object itemprototype.update(object/array itemPrototypes)`

このメソッドを使用すると、既存のアイテムプロトタイプを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するアイテムプロトタイプのプロパティ。

各アイテムプロトタイプでは `itemid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のアイテムプロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
preprocessing	array	現在の前処理オプションを置き換えるためのアイテムプロトタイプの前処理オプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
tags	array	アイテムプロトタイプのタグ。

戻り値

(object) itemids プロパティの下で更新されたアイテムプロトタイプの ID を含むオブジェクトを返します。

例

アイテムプロトタイプのインターフェースの変更

検出されたアイテムで使用するホストインターフェースを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

従属アイテムプロトタイプの更新

新しいマスターアイテムプロトタイプ ID で従属アイテムプロトタイプを更新します。同じホスト (テンプレート/ディスカバリールール) でのみ依存関係が許可されるため、マスターと従属アイテムは同じ hostid および ruleid を持つ必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムプロトタイプの更新

クエリフィールドを変更し、すべてのカスタムヘッダーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
```

```
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
        "headers": []
    }
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

アイテムの事前処理オプションの更新

アイテムプロトタイプをアイテム事前処理ルール「カスタム乗数」で更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

スクリプトアイテムプロトタイプの更新

異なるスクリプトでスクリプトアイテムプロトタイプを更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
```

```
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::update() in *ui/include/classes/api/services/CItemPrototype.php*.

アクション

このクラスはアクションを操作するために設計されています。

オブジェクト参照:

- アクション
- アクションの実行内容
  - アクションの実行内容メッセージ
  - アクションの実行内容の条件
- アクションのリカバリ実行内容
- アクションの更新実行内容
- アクションフィルター
  - アクションフィルターの条件

利用可能なメソッド:

- action.create - 新しいアクションを作成
- action.delete - アクションを削除
- action.get - アクションを取得
- action.update - アクションを更新

Action オブジェクト

次のオブジェクトは、actionAPI に直接関連しています。

アクション

アクションオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
actionid	ID	アクションの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須

プロパティ	タイプ	説明
esc_period	string	デフォルトの操作ステップ期間。60 秒以上である必要があります。秒数(例: 60、300)、時間のサフィックス付きの値（例: 30s、1m、2h)、またはユーザーマクロを指定できます。 プロパティの動作: - eventsource が「トリガーによって生成されたイベント」、「内部イベント」、または「サービスステータスの更新時に生成されたイベント」に設定されている場合にサポート
eventsource	integer	アクションが処理するイベントのタイプ。 サポートされるイベントタイプの一覧については、イベントの source プロパティを参照してください。 プロパティの動作: - 定数 - 作成操作では必須
name	string	アクションの名前。 プロパティの動作: - 作成操作では必須
status	integer	アクションが有効か無効か。 指定可能な値: 0 - （デフォルト）有効; 1 - 無効。
pause_symptoms	integer	イベントが症状イベントの場合にエスカレーションを一時停止するかどうか。 指定可能な値: 0 - 症状の障害に対するエスカレーションを一時停止しない; 1 - （デフォルト）症状の障害に対するエスカレーションを一時停止する。 プロパティの動作: - eventsource が「トリガーによって生成されたイベント」に設定されている場合にサポート
pause_suppressed	integer	メンテナンス期間中にエスカレーションを一時停止するかどうか。 指定可能な値: 0 - エスカレーションを一時停止しない; 1 - （デフォルト）エスカレーションを一時停止する。 プロパティの動作: - eventsource が「トリガーによって生成されたイベント」に設定されている場合にサポート
notify_if_canceled	integer	エスカレーションがキャンセルされたときに通知するかどうか。 指定可能な値: 0 - エスカレーションがキャンセルされたときに通知しない; 1 - （デフォルト）エスカレーションがキャンセルされたときに通知する。 プロパティの動作: - eventsource が「トリガーによって生成されたイベント」に設定されている場合にサポート

アクション操作

アクション操作オブジェクトは、アクションの実行時に実行される操作を定義します。次のプロパティがあります。

プロパティ	型	説明
operationtype	integer	操作のタイプ。 指定可能な値: 0 - メッセージの送信; 1 - グローバルスクリプト; 2 - ホストの追加; 3 - ホストの削除; 4 - ホストグループへの追加; 5 - ホストグループからの削除; 6 - テンプレートのリンク; 7 - テンプレートのリンク解除; 8 - ホストの有効化; 9 - ホストの無効化; 10 - ホストのインベントリモードの設定; 13 - ホストタグの追加; 14 - ホストタグの削除。 Action object の `eventsource` が「トリガーによって作成されたイベント」または「サービスステータスの更新時に作成されたイベント」に設定されている場合の指定可能な値: 0 - 「メッセージの送信」; 1 - 「グローバルスクリプト」。 Action object の `eventsource` が「内部イベント」に設定されている場合の指定可能な値: 0 - 「メッセージの送信」。 プロパティの動作: - 必須
esc_period	string	エスカレーションの 1 ステップの期間（秒）。60 秒より長くする必要があります。秒数（例: 60、300）、時間のサフィックス付きの値（例: 30s、1m、2h）、またはユーザーマクロを指定できます。0 または 0s に設定すると、デフォルトのアクションエスカレーション期間が使用されます。 デフォルト: 0s。 プロパティの動作: - Action object の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータスの更新時に作成されたイベント」に設定されている場合にサポート
esc_step_from	integer	エスカレーションを開始するステップ。 デフォルト: 1。 プロパティの動作: - Action object の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータスの更新時に作成されたイベント」に設定されている場合にサポート
esc_step_to	integer	エスカレーションを終了するステップ。 デフォルト: 1。 プロパティの動作: - Action object の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータスの更新時に作成されたイベント」に設定されている場合にサポート
evaltype	integer	操作条件の評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 1 - And; 2 - Or。

プロパティ	型	説明
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトには scriptid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定されている場合に必須
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループには groupid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定され、opcommand_hst が設定されていない場合に必須
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストには hostid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定され、opcommand_grp が設定されていない場合に必須
opconditions	array	トリガーアクションで使用する操作条件。 操作条件オブジェクトについては以下で詳しく説明します。
opgroup	array	ホストを追加するホストグループ。 ホストグループには groupid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「ホストグループへの追加」または「ホストグループからの削除」に設定されている場合に必須
opmessage	object	操作によって送信されるメッセージのデータを含むオブジェクト。 操作メッセージオブジェクトについては以下で詳しく説明します。 プロパティの動作: - operationtype が「メッセージの送信」に設定されている場合に必須
opmessage_grp	array	メッセージの送信先ユーザーグループ。 ユーザーグループには usrgrpid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージの送信」に設定され、opmessage_usr が設定されていない場合に必須
opmessage_usr	array	メッセージの送信先ユーザー。 ユーザーには userid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージの送信」に設定され、opmessage_grp が設定されていない場合に必須
optemplate	array	ホストにリンクするテンプレート。 テンプレートには templateid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「テンプレートのリンク」または「テンプレートのリンク解除」に設定されている場合に必須

プロパティ	型	説明
opinventory	object	ホストに設定するインベントリモード。 インベントリには `inventory_mode` プロパティを定義する必要があります。 プロパティの動作: - `operationtype` が「ホストのインベントリモードの設定」に設定されている場合に必須
optag	array	追加または削除するホストタグ。 タグには `tag` プロパティを定義する必要があります。 `value` プロパティは任意です。 プロパティの動作: - `operationtype` が「ホストタグの追加」または「ホストタグの削除」に設定されている場合にサポート。

アクション実行メッセージ

実行メッセージオブジェクトには、実行によって送信されるメッセージに関するデータが含まれます。以下のプロパティがあります。

プロパティ	型	説明
default_msg	integer	デフォルトのアクションメッセージの本文と件名を使用するかどうか。 指定可能な値: 0 - 実行のデータを使用; 1 - *(デフォルト)* メディアタイプのデータを使用。
mediatypeid	ID	メッセージの送信に使用されるメディアタイプの ID。 プロパティの動作: - アクション実行オブジェクト、アクション復旧実行オブジェクト、またはアクション更新実行オブジェクト の `operationtype` が「メッセージ送信」に設定されている場合、またはアクション更新実行オブジェクト の `operationtype` が「関係者全員に通知」に設定されている場合に *supported*
message	string	実行メッセージの本文。 プロパティの動作: - `default_msg` が「実行のデータを使用」に設定されている場合に *supported*
subject	string	実行メッセージの件名。 プロパティの動作: - `default_msg` が「実行のデータを使用」に設定されている場合に *supported*

アクション実行条件

アクション実行条件オブジェクトは、現在の実行を行うために満たす必要がある条件を定義します。以下のプロパティがあります。

Property	Type	Description
conditiontype	integer	条件のタイプ。 指定可能な値: 14 - イベント確認済み。 プロパティの動作: - 必須

Property	Type	Description
value	string	比較する値。 プロパティの動作: - 必須
operator	integer	条件演算子。 指定可能な値: 0 - *(デフォルト)* =

以下の演算子と値が、各実行条件タイプでサポートされています。

Condition	Condition name	Supported operators	Expected value
14	イベント確認済み	=	イベントが確認済みかどうか。 指定可能な値: 0 - 未確認; 1 - 確認済み。

アクションのリカバリー実行内容

アクションのリカバリー実行内容オブジェクトは、障害が解決されたときに実行される操作を定義します。
リカバリー実行内容が可能なのは、トリガー、内部、およびサービスアクションのみです。
このオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
operationtype	integer	操作の種類。 アクションオブジェクト の `eventsource` が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合の使用可能な値: 0 - メッセージ送信; 1 - グローバルスクリプト; 11 - 関係者全員に通知。 アクションオブジェクト の `eventsource` が「内部イベント」に設定されている場合の使用可能な値: 0 - メッセージ送信; 11 - 関係者全員に通知。 プロパティの動作: - 必須
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトでは `scriptid` プロパティを定義する必要があります。 プロパティの動作: - `operationtype` が「グローバルスクリプト」に設定されている場合は必須
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループでは `groupid` プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の `eventsource` が「トリガーによって作成されたイベント」に設定され、`operationtype` が「グローバルスクリプト」に設定され、かつ `opcommand_hst` が設定されていない場合は 必須

プロパティ	型	説明
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストでは hostid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、かつ opcommand_grp が設定されていない場合は 必須
opmessage	object	リカバリー実行内容によって送信されるメッセージに関するデータを含むオブジェクト。 実行内容メッセージオブジェクトについては、上で詳しく説明しています。 プロパティの動作: - operationtype が「メッセージ送信」に設定されている場合は 必須
opmessage_grp	array	メッセージの送信先ユーザーグループ。 ユーザーグループでは usrgrpid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定され、かつ opmessage_usr が設定されていない場合は 必須
opmessage_usr	array	メッセージの送信先ユーザー。 ユーザーでは userid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定され、かつ opmessage_grp が設定されていない場合は 必須

アクション更新操作

アクション更新操作オブジェクトは、障害が更新されたとき（コメントの追加、確認、深刻度の変更、または手動クローズ）に実行される操作を定義します。更新操作は、トリガーアクションおよびサービスアクションでのみ可能です。このオブジェクトには以下のプロパティがあります。

Property	Type	Description
operationtype	integer	操作の種類。 使用可能な値: 0 - メッセージ送信; 1 - グローバルスクリプト; 12 - 関係者全員に通知。 プロパティの動作: - 必須
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトでは scriptid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定されている場合は 必須

Property	Type	Description
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループでは groupid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定されており、かつ opcommand_hst が設定されていない場合は 必須
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストでは hostid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定されており、かつ opcommand_grp が設定されていない場合は 必須
opmessage	object	更新操作によって送信されるメッセージに関するデータを含むオブジェクト。 操作メッセージオブジェクトの詳細は、上記の説明を参照してください。
opmessage_grp	array	メッセージの送信先となるユーザーグループ。 ユーザーグループでは usrgrpid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定されており、opmessage_usr が設定されていない場合は 必須
opmessage_usr	array	メッセージの送信先となるユーザー。 ユーザーでは userid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定されており、opmessage_grp が設定されていない場合は 必須

アクションフィルター

アクションフィルターオブジェクトは、設定されたアクションの実行操作を実行するために満たす必要がある一連の条件を定義します。以下のプロパティがあります。

プロパティ	Type	説明
conditions	array	結果をフィルタリングするために使用するフィルター条件のセット。条件は、formula 内で配置された順序で並べ替えられます。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須

プロパティ	Type	説明
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、各フィルター条件をその `formulaid` によって参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 プロパティの動作: - 読み取り専用
formula	string	カスタム式を使用するフィルターの条件評価に使用する、ユーザー定義の式。この式には、各フィルター条件をその `formulaid` によって参照する ID を含める必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致している必要があります。未使用または省略された条件があってはなりません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は 必須

アクションフィルター条件

アクションフィルター条件オブジェクトは、アクション操作を実行する前に確認する必要がある特定の条件を定義します。

Property	Type	Description
conditiontype	integer	条件のタイプ。 Action object の `eventsource` が「トリガーによって生成されたイベント」に設定されている場合の使用可能な値: 0 - ホストグループ; 1 - ホスト; 2 - トリガー; 3 - イベント名; 4 - トリガーの深刻度; 6 - 期間; 13 - ホストテンプレート; 16 - 障害が抑制されているかどうか; 25 - イベントタグ; 26 - イベントタグの値。 Action object の `eventsource` が「ディスカバリールールによって生成されたイベント」に設定されている場合の使用可能な値: 7 - ホスト IP; 8 - 検出されたサービスのタイプ; 9 - 検出されたサービスのポート; 10 - ディスカバリーのステータス; 11 - 稼働時間または停止時間; 12 - 受信値; 18 - ディスカバリールール; 19 - ディスカリーチェック; 20 - プロキシ; 21 - ディスカリーオブジェクト; 29 - プロキシグループ。 Action object の `eventsource` が「アクティブエージェントの自動登録によって生成されたイベント」に設定されている場合の使用可能な値: 20 - プロキシ; 22 - ホスト名; 24 - ホストメタデータ; 29 - プロキシグループ。 Action object の `eventsource` が「内部イベント」に設定されている場合の使用可能な値: 0 - ホストグループ; 1 - ホスト; 13 - ホストテンプレート; 23 - イベントタイプ; 25 - イベントタグ; 26 - イベントタグの値。 Action object の `eventsource` が「サービスステータスの更新時に生成されたイベント」に設定されている場合の使用可能な値: 25 - イベントタグ; 26 - イベントタグの値; 27 - サービス; 28 - サービス名。 Property behavior: - *required*
value	string	比較対象の値。 Property behavior: - *required*

Property	Type	Description
value2	string	比較対象の副次的な値。 Property behavior: - Action object の `eventsource` が「トリガーによって生成されたイベント」に設定され、`conditiontype` がトリガーアクションで使用可能な値に設定され、条件のタイプ（以下を参照）が「26」の場合は *required* - Action object の `eventsource` が「内部イベント」に設定され、`conditiontype` が内部アクションで使用可能な値に設定され、条件のタイプ（以下を参照）が「26」の場合は *required* - Action object の `eventsource` が「サービスステータスの更新時に生成されたイベント」に設定され、`conditiontype` がサービスアクションで使用可能な値に設定され、条件のタイプ（以下を参照）が「26」の場合は *required*
formulaid	string	カスタム式から条件を参照するために使用する任意の一意の ID。大文字のみを含めることができます。フィルター条件を変更するときはユーザーが ID を定義する必要がありますが、その後に条件をリクエストすると新たに生成されます。
operator	integer	条件の演算子。 使用可能な値: 0 - *(default)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 4 - いずれかに含まれる; 5 - 以上; 6 - 以下; 7 - いずれにも含まれない; 8 - 一致する; 9 - 一致しない; 10 - はい; 11 - いいえ。

Note:
さまざまなタイプの式でフィルターを使用する方法をよりよく理解するには、action.get メソッドおよびaction.create メソッドのページにある例を参照してください。

各条件タイプで、次の演算子と値がサポートされています。

Condition	Condition name	Supported operators	Expected value
0	ホストグループ	等しい、 等しくない	ホストグループ ID。
1	ホスト	等しい、 等しくない	ホスト ID。
2	トリガー	等しい、 等しくない	トリガー ID。
3	イベント名	含む、 含まない	イベント名。
4	トリガーの深刻度	等しい、 等しくない、 以上、 以下	トリガーの深刻度。サポートされているトリガーの深刻度の一覧については、トリガーの `severity` プロパティを参照してください。
5	トリガーの値	等しい	トリガーの値。サポートされているトリガーの値の一覧については、トリガーの `value` プロパティを参照してください。
6	期間	いずれかに含まれる、 いずれにも含まれない	イベントがトリガーされた時刻を期間として指定します。

Condition	Condition name	Supported operators	Expected value
7	ホスト IP	等しい、 等しくない	確認する 1 つまたは複数の IP 範囲を、コンマで区切って指定します。IP 範囲でサポートされている形式の詳細については、ネットワークディスカバリーの設定セクションを参照してください。
8	検出されたサービスのタイプ	等しい、 等しくない	検出されたサービスのタイプ。サービスのタイプは、サービスの検出に使用されたディスカバリーチェックのタイプと一致します。サポートされているタイプの一覧については、ディスカバリーチェックの type プロパティを参照してください。
9	検出されたサービスのポート	等しい、 等しくない	1 つまたは複数のポート範囲を、コンマで区切って指定します。
10	ディスカバリーのステータス	等しい	検出されたオブジェクトのステータス。 使用可能な値: 0 - ホストまたはサービスが稼働中; 1 - ホストまたはサービスが停止中; 2 - ホストまたはサービスが検出された; 3 - ホストまたはサービスが失われた。
11	稼働時間または停止時間	以上、 以下	検出されたオブジェクトが現在のステータスになってからの時間を秒単位で示します。
12	受信値	等しい、 等しくない、 以上、 以下、 含む、 含まない	Zabbix エージェント、SNMPv1、SNMPv2、または SNMPv3 のディスカバリーチェックを実行したときに返される値。
13	ホストテンプレート	等しい、 等しくない	リンクされたテンプレートの ID。
16	障害が抑制されているかどうか	はい、いいえ	値は不要です。「はい」演算子を使用すると障害が抑制されている必要があり、「いいえ」を使用すると抑制されていない必要があります。
18	ディスカバリールール	等しい、 等しくない	ディスカバリールールの ID。
19	ディスカバリーチェック	等しい、 等しくない	ディスカバリーチェックの ID。
20	プロキシ	等しい、 等しくない	プロキシの ID。
21	ディスカバリーオブジェクト	等しい	ディスカバリーイベントをトリガーしたオブジェクトのタイプ。 使用可能な値: 1 - 検出されたホスト; 2 - 検出されたサービス。
22	ホスト名	含む、 含まない、 一致する、 一致しない	ホスト名。 自動登録条件では、「一致する」および「一致しない」演算子に正規表現を使用できます。
23	イベントタイプ	等しい	特定の内部イベント。 使用可能な値: 0 - アイテムが「取得不可」状態; 1 - アイテムが「正常」状態; 2 - LLD ルールが「取得不可」状態; 3 - LLD ルールが「正常」状態; 4 - トリガーが「不明」状態; 5 - トリガーが「正常」状態。
24	ホストメタデータ	含む、 含まない、 一致する、 一致しない	自動登録されたホストのメタデータ。 「一致する」および「一致しない」演算子に正規表現を使用できます。
25	タグ	等しい、 等しくない、 含む、 含まない	イベントタグ。

Condition	Condition name	Supported operators	Expected value
26	タグの値	等しい、 等しくない、 含む、 含まない	イベントタグの値。
27	サービス	等しい、 等しくない	サービス ID。
28	サービス名	等しい、 等しくない	サービス名。
29	プロキシグループ	等しい、 等しくない	プロキシグループの ID。

action.create

説明

`object action.create(object/array actions)`

このメソッドを使用すると、新しいアクションを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するアクション。

標準のアクションプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	アクション用のアクションフィルターオブジェクト。
operations	array	アクション用に作成するアクションの実行内容。
recovery_operations	array	アクション用に作成するアクションの復旧時の実行内容。
update_operations	array	アクション用に作成するアクションの更新時の実行内容。

戻り値

(`object`) `actionids` プロパティの下に作成されたアクションの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアクションの順序と一致します。

例

トリガーアクションの作成

ホスト”10084” からのトリガー (名前に”memory” が含まれる) が障害状態になると実行されるトリガーアクションを作成する。アクションには 4 つの操作が構成されます。最初の即時操作では、メディアタイプ”1” を介してユーザーグループ”7” 内のすべてのユーザーにメッセージが送信されます。イベントが 30 分以内に解決されない場合、2 番目の操作ではグループ”2” のすべてのホストでスクリプト”5”(スコープ” アクション操作” のスクリプト) が実行されます。イベントが解決された場合は、回復操作により、問題に関するメッセージを受け取ったすべてのユーザーに通知されます。イベントが更新されると、確認/更新操作により、問題に関するメッセージを受信したすべてのユーザーに (カスタムの件名とメッセージを使用して) 通知します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "30m",
        "filter": {
            "evaltype": 0,
            "conditions": [
```

```
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084"
                },
                {
                    "conditiontype": 3,
                    "operator": 2,
                    "value": "memory"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            },
            {
                "operationtype": 1,
                "esc_step_from": 2,
                "esc_step_to": 2,
                "opconditions": [
                    {
                        "conditiontype": 14,
                        "operator": 0,
                        "value": "0"
                    }
                ],
                "opcommand_grp": [
                    {
                        "groupid": "2"
                    }
                ],
                "opcommand": {
                    "scriptid": "5"
                }
            }
        ],
        "recovery_operations": [
            {
                "operationtype": "11",
                "opmessage": {
                    "default_msg": 1
                }
            }
        ],
        "update_operations": [
            {
                "operationtype": "12",
                "opmessage": {
                    "default_msg": 0,
                    "message": "Custom update operation message body",
```

```
                    "subject": "Custom update operation message subject"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

ディスカバリアクションの作成

検出されたホストにテンプレート”10001” をリンクするディスカバリアクションを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Discovery action",
        "eventsource": 1,
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 21,
                    "operator": 0,
                    "value": "1"
                },
                {
                    "conditiontype": 10,
                    "operator": 0,
                    "value": "2"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 6,
                "optemplate": [
                    {
                        "templateid": "10001"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

アクション条件を評価するためのカスタム式"A and (B or C)" を使用するトリガーアクションを作成します。ホスト"10084" またはホスト"10106" からの" 警告" 以上の深刻度を持つトリガーが障害状態になると、このアクションは、メディアタイプ"1" を介してユーザーグループ"7" のすべてのユーザーにメッセージを送信します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "15m",
        "filter": {
            "evaltype": 3,
            "formula": "A and (B or C)",
            "conditions": [
                {
                    "conditiontype": 4,
                    "operator": 5,
                    "value": "2",
                    "formulaid": "A"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084",
                    "formulaid": "B"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10106",
                    "formulaid": "C"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            }
        ]
    },
    "id": 1
```

```
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

エージェントの自動登録ルール作成

ホスト名に”SRV ” が含まれているか、メタデータに”AlmaLinux” が含まれている場合、ホストをホストグループ”2” に追加します。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers",
        "eventsource": "2",
        "filter": {
            "evaltype": "2",
            "conditions": [
                {
                    "conditiontype": "22",
                    "operator": "2",
                    "value": "SRV"
                },
                {
                    "conditiontype": "24",
                    "operator": "2",
                    "value": "AlmaLinux"
                }
            ]
        },
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

ホストタグとエージェント自動登録ルールの作成

ホストをホストグループ”2” に追加し、2 つのホストタグを追加する自動登録アクションを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers with tags",
        "eventsource": "2",
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            },
            {
                "operationtype": "13",
                "optag": [
                    {
                        "tag": "location",
                        "value": "office"
                    },
                    {
                        "tag": "city",
                        "value": "Riga"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

関連項目

- アクションフィルター
- アクション操作
- スクリプト

ソース

CAction::create() in *ui/include/classes/api/services/CAction.php*.

action.delete

説明

`object action.delete(array actionIds)`

このメソッドを使用すると、アクションを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 ユーザーは、フィルター条件にアクセスできないプロキシまたはプロキシグループが含まれているアクションを削除できません。詳細については、ユーザーグループの設定（プロキシのアクセスリスト）を参照してください。

パラメーター

(array) 削除するアクションの ID。

戻り値

(object) actionids プロパティの下にある削除されたアクションの ID を含むオブジェクトを返します。

例

複数のアクションの削除

2 つのアクションを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
        ]
    },
    "id": 1
}
```

ソース

CAction::delete() in *ui/include/classes/api/services/CAction.php*.

action.get

説明

`integer/array action.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってアクションを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
actionids	ID/array	指定した ID を持つアクションのみを返します。
groupids	ID/array	アクション条件で指定したホストグループを使用するアクションのみを返します。
hostids	ID/array	アクション条件で指定したホストを使用するアクションのみを返します。
triggerids	ID/array	アクション条件で指定したトリガーを使用するアクションのみを返します。
mediatypeids	ID/array	メッセージ送信に指定したメディアタイプを使用するアクションのみを返します。
usrgrpids	ID/array	指定したユーザーグループにメッセージを送信するよう設定されているアクションのみを返します。
userids	ID/array	指定したユーザーにメッセージを送信するよう設定されているアクションのみを返します。
scriptids	ID/array	指定したスクリプトを実行するよう設定されているアクションのみを返します。
selectFilter	query	アクション条件フィルターを含む`filter` プロパティを返します。
selectOperations	query	アクション操作を含む`operations` プロパティを返します。
selectRecoveryOperations	query	アクションのリカバリー操作を含む`recovery_operations` プロパティを返します。
selectUpdateOperations	query	アクションの更新操作を含む`update_operations` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `actionid`, `name`, `status`。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のどちらかを返します。

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合は、取得したオブジェクトの数

例

トリガーアクションの取得

アクション条件とオペレーションを含む、すべての設定済みトリガーアクションを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Report problems to Zabbix administrators",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ]
                }
            ],
            "recovery_operations": [
                {
                    "operationid": "7",
                    "actionid": "3",
                    "operationtype": "11",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                        "message": "Trigger: {TRIGGER.NAME}\r\nTrigger state: {TRIGGER.STATUS}\r\nTrigger
                        "mediatypeid": "0"
                    }
                }
            ],
            "update_operations": [
                {
```

```
                    "operationid": "31",
                    "operationtype": "12",
                    "evaltype": "0",
                    "opmessage": {
                        "default_msg": "1",
                        "subject": "",
                        "message": "",
                        "mediatypeid": "0"
                    }
                },
                {
                    "operationid": "32",
                    "operationtype": "0",
                    "evaltype": "0",
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "Updated: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} updated problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.T
                        "mediatypeid": "1"
                    },
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ],
                    "opmessage_usr": []
                },
                {
                    "operationid": "33",
                    "operationtype": "1",
                    "evaltype": "0",
                    "opcommand": {
                        "scriptid": "3"
                    },
                    "opcommand_hst": [
                        {
                            "hostid": "10084"
                        }
                    ],
                    "opcommand_grp": []
                }
            ]
        }
    ],
    "id": 1
}
```

ディスカバリアクションの取得

設定されたすべてのディスカバリアクションを、アクションの条件と操作とともに取得します。このフィルターは"and" 評価タイプを使用するため、`formula` プロパティは空で、`eval_formula` が自動的に生成されます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "2",
            "name": "Auto discovery. Linux servers.",
            "eventsource": "1",
            "status": "1",
            "esc_period": "0s",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "conditiontype": "10",
                        "operator": "0",
                        "value": "0",
                        "value2": "",
                        "formulaid": "B"
                    },
                    {
                        "conditiontype": "8",
                        "operator": "0",
                        "value": "9",
                        "value2": "",
                        "formulaid": "C"
                    },
                    {
                        "conditiontype": "12",
                        "operator": "2",
                        "value": "Linux",
                        "value2": "",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A and B and C"
            },
            "operations": [
                {
                    "operationid": "1",
                    "actionid": "2",
                    "operationtype": "6",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "optemplate": [
                        {
                            "templateid": "10001"
                        }
                    ]
                },
                {
                    "operationid": "2",
                    "actionid": "2",
```

```
                    "operationtype": "4",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opgroup": [
                        {
                            "groupid": "2"
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- アクションフィルター
- アクション操作

ソース

CAction::get() in *ui/include/classes/api/services/CAction.php*.

action.update

説明

`object action.update(object/array actions)`

このメソッドを使用すると、既存のアクションを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ使用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するアクションのプロパティ。

各アクションでは `actionid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のアクションプロパティ に加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
filter	object	現在のフィルターを置き換えるアクションフィルター オブジェクト。
operations	array	既存の実行内容を置き換えるアクションの実行内容。
recovery_operations	array	既存のリカバリー実行内容を置き換えるアクションのリカバリー実行内容。 パラメータの動作: - アクションオブジェクト の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合に *supported*
update_operations	array	既存の更新実行内容を置き換えるアクションの更新実行内容。 パラメータの動作: - アクションオブジェクト の `eventsource` が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合に *supported*

戻り値

(object) actionids プロパティの下で更新されたアクション ID を含むオブジェクトを返します。

例

アクションの無効化

ステータスを"1" に設定してアクションを無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- アクションフィルター
- アクション操作

ソース

CAction::update() in *ui/include/classes/api/services/CAction.php*.

イベント

このクラスはイベントを扱うために設計されています。

オブジェクト参照:

- イベント
- イベントタグ
- メディアタイプ URL

利用可能なメソッド:

- event.get - イベントを取得
- event.acknowledge - イベントを確認済みにする

Event オブジェクト

次のオブジェクトは、eventAPI に直接関連しています。

イベント

> **Note:**
> イベントは Zabbix サーバーによって作成され、API 経由で変更することはできません。

イベントオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
eventid	ID	イベントの ID。
source	integer	イベントの種類。 指定可能な値: 0 - トリガーによって作成されたイベント; 1 - ディスカバリルールによって作成されたイベント; 2 - アクティブエージェントの自動登録によって作成されたイベント; 3 - 内部イベント; 4 - サービスステータス更新時に作成されたイベント。
object	integer	イベントに関連するオブジェクトの種類。 `source` が「トリガーによって作成されたイベント」に設定されている場合の指定可能な値: 0 - トリガー。 `source` が「ディスカバリルールによって作成されたイベント」に設定されている場合の指定可能な値: 1 - 検出されたホスト; 2 - 検出されたサービス。 `source` が「アクティブエージェントの自動登録によって作成されたイベント」に設定されている場合の指定可能な値: 3 - 自動登録されたホスト。 `source` が「内部イベント」に設定されている場合の指定可能な値: 0 - トリガー; 4 - アイテム; 5 - LLD ルール。 `source` が「サービスステータス更新時に作成されたイベント」に設定されている場合の指定可能な値: 6 - サービス。
objectid	ID	関連オブジェクトの ID。
acknowledged	integer	イベントが確認済みかどうか。
clock	timestamp	イベントが作成された時刻。
ns	integer	イベントが作成された時点のナノ秒。
name	string	解決済みのイベント名。
value	integer	関連オブジェクトの状態。 `source` が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合の指定可能な値: 0 - OK; 1 - 障害。 `source` が「ディスカバリルールによって作成されたイベント」に設定されている場合の指定可能な値: 0 - ホストまたはサービスが稼働中; 1 - ホストまたはサービスが停止中; 2 - ホストまたはサービスが検出された; 3 - ホストまたはサービスが失われた。 `source` が「内部イベント」に設定されている場合の指定可能な値: 0 - 「正常」状態; 1 - 「不明」または「未サポート」状態。 プロパティの動作: - `source` が「トリガーによって作成されたイベント」、「ディスカバリルールによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合は *supported*

プロパティ	型	説明
severity	integer	イベントの現在の深刻度。 指定可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 致命的。
r_eventid	ID	復旧イベントの ID。
c_eventid	ID	グローバル相関ルールの下で現在のイベントを上書き（クローズ）するために使用されたイベントの ID。正確な相関ルールを特定するには `correlationid` を参照してください。 このパラメータは、イベントがグローバル相関ルールによってクローズされた場合にのみ定義されます。
cause_eventid	ID	原因イベントの ID。
correlationid	ID	障害のクローズを生成した相関ルールの ID。 このパラメータは、イベントがグローバル相関ルールによってクローズされた場合にのみ定義されます。
userid	ID	イベントをクローズしたユーザーの ID（イベントが手動でクローズされた場合）。
suppressed	integer	イベントが抑制されているかどうか。 指定可能な値: 0 - イベントは通常状態; 1 - イベントは抑制されている。
opdata	string	展開済みマクロを含む運用データ。
urls	array	アクティブなメディアタイプ URL。

イベントタグ

イベントタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	イベントタグ名。
value	string	イベントタグの値。

メディアタイプ URL

メディアタイプ URL オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	メディアタイプで定義された URL 名。
url	string	メディアタイプで定義された URL 値。

結果には、有効なイベントメニューエントリを持つアクティブなメディアタイプのエントリのみが含まれます。プロパティ内で使用されているマクロは展開されますが、いずれかのプロパティに未展開のマクロが含まれている場合、両方のプロパティが結果から除外されます。サポートされているマクロについては、サポートされているマクロを参照してください。

event.acknowledge

説明

```
object event.acknowledge(object/array parameters)
```

このメソッドを使用すると、イベントを更新できます。実行できる更新アクションは次のとおりです。

- イベントをクローズする。イベントがすでに解決済みの場合、このアクションはスキップされます。
- イベントを承認する。イベントがすでに承認済みの場合、このアクションはスキップされます。
- イベントの承認を取り消す。イベントが承認されていない場合、このアクションはスキップされます。

- メッセージを追加する。
- イベントの深刻度を変更する。イベントがすでに同じ深刻度の場合、このアクションはスキップされます。
- イベントを抑制する。イベントがすでに抑制されている場合、このアクションはスキップされます。
- イベントの抑制を解除する。イベントが抑制されていない場合、このアクションはスキップされます。
- イベントのランクを原因に変更する。イベントがすでに原因としてランク付けされている場合、このアクションはスキップされます。
- イベントのランクを症状に変更する。イベントがすでに症状としてランク付けされている場合、このアクションはスキップされます。

Attention:
更新できるのはトリガーイベントのみです。
 更新できるのは問題イベントのみです。
 イベントをクローズするか、イベントの深刻度を変更するには、トリガーに対する読み取り/書き込み権限が必要です。
 イベントをクローズするには、トリガーで手動クローズが許可されている必要があります。

Note:
このメソッドは、すべての種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーの役割設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) イベントの ID と、実行する更新操作を含むパラメータ。

Parameter	Type	Description
eventids	ID/array	承認するeventの ID。 パラメータの動作: - 必須
action	integer	イベントの更新操作。 指定可能なビットマップ値: 1 - 障害をクローズ; 2 - イベントを承認; 4 - メッセージを追加; 8 - 深刻度を変更; 16 - イベントの承認を解除; 32 - イベントを抑制; 64 - イベントの抑制を解除; 128 - イベントのランクを原因に変更; 256 - イベントのランクを症状に変更。 これはビットマスクフィールドです。指定可能なビットマップ値の合計を指定できます（たとえば、イベントを承認してメッセージを残す場合は 6、イベントを承認して抑制する場合は 34）。 パラメータの動作: - 必須
cause_eventid	ID	原因イベントの ID。 パラメータの動作: - action に「イベントのランクを症状に変更」ビットが含まれる場合に必須
message	string	メッセージのテキスト。 パラメータの動作: - action に「メッセージを追加」ビットが含まれる場合に必須

Parameter	Type	Description
severity	integer	イベントの新しい深刻度。 指定可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。 パラメータの動作: - action に「深刻度を変更」ビットが含まれる場合に必須
suppress_until	integer	イベントを抑制する期限を示す Unix タイムスタンプ。 「0」に設定すると、抑制は無期限になります。 パラメータの動作: - action に「イベントを抑制」ビットが含まれる場合に必須

戻り値

(object) 更新されたイベントの ID を eventids プロパティに含むオブジェクトを返します。

例

イベントの確認

1 つのイベントを確認し、メッセージを残します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
        "message": "Problem resolved."
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

イベントの深刻度の変更

複数のイベントの深刻度を変更し、メッセージを残します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
```

```
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
            "20428"
        ]
    },
    "id": 1
}
```

イベントを無期限に抑制する

有効期限なしで単一のイベントを抑制します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

イベントの抑制解除

以前に抑制された単一のイベントの抑制を解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

イベントを確認し、抑制する

1 つ以上のイベントを確認し、メッセージを追加して、同じリクエストでそれらを抑制します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Suppressing during planned maintenance.",
        "suppress_until": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427,
            20428
        ]
    },
    "id": 1
}
```

ソース

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

説明

`integer/array event.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってイベントを取得できます。

> **Attention:**
> このメソッドは、housekeeper によってまだ削除されていない場合、削除済みエンティティのイベントを返すことがあります。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(object) 目的の出力を定義するパラメーター。

このメソッドは次のパラメーターをサポートします。

Parameter	Type	Description
eventids	ID/array	指定した ID を持つイベントのみを返します。
groupids	ID/array	指定したホストグループに属するオブジェクトによって作成されたイベントのみを返します。
hostids	ID/array	指定したホストに属するオブジェクトによって作成されたイベントのみを返します。
objectids	ID/array	指定したオブジェクトによって作成されたイベントのみを返します。
source	integer	指定したタイプのイベントのみを返します。 サポートされているイベントタイプの一覧については、event object page を参照してください。 デフォルト: 0 - トリガーイベント。
object	integer	指定したタイプのオブジェクトによって作成されたイベントのみを返します。 サポートされているオブジェクトタイプの一覧については、event object page を参照してください。 デフォルト: 0 - トリガー。
acknowledged	boolean	`true` に設定すると、確認済みのイベントのみを返します。
action	integer	指定したevent update actionsが実行されたイベントのみを返します。複数のアクションを指定する場合は、許容されるビットマップ値の合計をビットマスクとして使用します（たとえば、確認とイベントの抑制には 34）。
action_userids	ID/array	指定した ID の、イベント更新アクションを実行したユーザーを持つイベントのみを返します。
suppressed	boolean	`true` に設定すると、抑制されたイベントのみを返します。
symptom	boolean	`true` に設定すると、症状イベントのみを返します。
severities	integer/array	指定したイベント深刻度を持つイベントのみを返します。object が trigger の場合にのみ適用されます。
trigger_severities	integer/array	指定したトリガー深刻度を持つイベントのみを返します。object が trigger の場合にのみ適用されます。
evaltype	integer	タグのevaluation method。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	array	指定したタグを持つイベントのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのイベントが返されます。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist。
eventid_from	string	ID が指定した ID 以上のイベントのみを返します。
eventid_till	string	ID が指定した ID 以下のイベントのみを返します。
time_from	timestamp	指定した時刻以降に作成されたイベントのみを返します。
time_till	timestamp	指定した時刻以前に作成されたイベントのみを返します。
problem_time_from	timestamp	現在の状態に関係なく、`problem_time_from` から障害状態にあったイベントのみを返します。source がトリガーイベントで、object が trigger の場合にのみ適用されます。`problem_time_till` が指定されている場合、このパラメーターは必須です。
problem_time_till	timestamp	現在の状態に関係なく、`problem_time_till` まで障害状態にあったイベントのみを返します。source がトリガーイベントで、object が trigger の場合にのみ適用されます。`problem_time_from` が指定されている場合、このパラメーターは必須です。

Parameter	Type	Description
value	integer/array	指定した値を持つイベントのみを返します。
selectAcknowledges	query	イベント更新を含む `acknowledges` プロパティを返します。イベント更新は新しい順に並べ替えられます。 イベント更新オブジェクトには次のプロパティがあります: `acknowledgeid` - (ID) 確認の ID; `userid` - (ID) イベントを更新したユーザーの ID; `clock` - (timestamp) イベントが更新された時刻; `message` - (string) メッセージのテキスト; `action` - (integer) 実行された更新アクション。詳細はevent.acknowledge を参照; `old_severity` - (integer) この更新アクション前のイベント深刻度; `new_severity` - (integer) この更新アクション後のイベント深刻度; `suppress_until` - (timestamp) イベントが抑制される期限; `taskid` - (ID) 現在のイベントがランク変更中の場合のタスク ID; `maintenanceid` - (ID) メンテナンスの ID; `username` - (string) イベントを更新したユーザーのユーザー名; `name` - (string) イベントを更新したユーザーの名; `surname` - (string) イベントを更新したユーザーの姓。 `count` をサポートします。
selectAlerts	query	イベントによって生成されたアラートを含む `alerts` プロパティを返します。アラートは新しい順に並べ替えられます。
selectHosts	query	イベントを作成したオブジェクトを含むホストを持つ `hosts` プロパティを返します。トリガー、アイテム、または LLD ルールによって生成されたイベントでのみサポートされます。
selectRelatedObject	query	イベントを作成したオブジェクトを含む `relatedObject` プロパティを返します。返されるオブジェクトのタイプはイベントタイプによって異なります。
selectSuppressionData	query	有効なメンテナンスと手動による抑制の一覧を含む `suppression_data` プロパティを返します: `maintenanceid` - (ID) メンテナンスの ID; `userid` - (ID) イベントを抑制したユーザーの ID; `suppress_until` - (integer) イベントが抑制されるまでの時間。
selectTags	query	イベントタグを含むtags プロパティを返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が照合対象の単一値または値の配列であるオブジェクトを受け付けます。 `text` data type のプロパティはサポートされません。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `eventid`, `objectid`, `clock`。 `groupBy` と併用した場合の指定可能な値: `objectid`。 `countOutput` と `groupBy` と併用した場合の指定可能な値: `objectid`, `rowscount`。
groupBy	string/array	指定したプロパティで結果をグループ化します。指定したプロパティは結果に返されます。 指定可能な値: `objectid`。
countOutput	boolean	これらのパラメーターはreference commentary に説明があります。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parameter	Type	Description
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されていて、groupBy パラメータが使用されていない場合は、取得したオブジェクトの件数
- groupBy パラメータが使用されている場合は、集計結果を含むオブジェクトの配列

例

トリガーイベントの取得

トリガー"22395" から最新のイベントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
```

```
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "1",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "maintenanceid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "tags": [
```

```
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

期間によるイベントの取得

2012 年 10 月 17 日から 18 日までに作成されたすべてのイベントを、逆時系列で取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Less than 25% free in the history cache",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
```

```
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper processes more than 75% busy",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


指定したユーザーによって確認済みの障害を取得する

ID=10 のユーザーによって確認済みの障害を取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
```

```
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

問題イベント数によるトップトリガーの取得

指定した期間内に「警告」、「軽度の障害」、「重度の障害」、「致命的な障害」の深刻度を持つトリガーのうち、問題イベント数が多い上位 5 件を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "countOutput": true,
        "groupBy": "objectid",
        "source": 0,
        "object": 0,
        "value": 1,
        "time_from": 1672531200,
        "time_till": 1677628800,
        "trigger_severities": [2, 3, 4, 5],
        "sortfield": ["rowscount"],
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
```

```
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "objectid": "232124",
            "rowscount": "27"
        },
        {
            "objectid": "29055",
            "rowscount": "23"
        },
        {
            "objectid": "253731",
            "rowscount": "18"
        },
        {
            "objectid": "254062",
            "rowscount": "11"
        },
        {
            "objectid": "23216",
            "rowscount": "7"
        }
    ],
    "id": 1
}
```

参照

- Alert
- Item
- Host
- LLD rule
- Service
- Trigger

ソース

CEvent::get() in *ui/include/classes/api/services/CEvent.php*.

グラフアイテム

このクラスはグラフアイテムを操作するために設計されています。

オブジェクト参照:

- グラフアイテム

利用可能なメソッド:

- graphitem.get - グラフアイテムを取得します

Graph item オブジェクト

次のオブジェクトは、graphitemAPI に直接関連しています。

グラフアイテム

> **Note:**
> グラフアイテムは graph API 経由でのみ変更できます。

グラフアイテムオブジェクトには、以下のプロパティがあります。

Property	Type	Description
gitemid	ID	グラフアイテムの ID。 Property behavior: - *read-only*
color	string	16 進数のカラーコードとして表されるグラフアイテムの描画色。 Property behavior: - 作成操作では *required*
itemid	ID	アイテムの ID。 Property behavior: - 作成操作では *required*
calc_fnc	integer	表示されるアイテムの値。 指定可能な値: 1 - 最小値; 2 - *(default)* 平均値; 4 - 最大値; 7 - すべての値; 9 - 最後の値。円グラフおよび分割円グラフでのみ使用されます。
drawtype	integer	グラフアイテムの描画スタイル。 指定可能な値: 0 - *(default)* 線; 1 - 塗りつぶし領域; 2 - 太線; 3 - 点; 4 - 破線; 5 - グラデーション線。
graphid	ID	このグラフアイテムが属するグラフの ID。
sortorder	integer	グラフ内でのアイテムの位置。 デフォルト: ”0” から始まり、エントリごとに 1 ずつ増加します。
type	integer	グラフアイテムのタイプ。 指定可能な値: 0 - *(default)* シンプル; 2 - グラフ合計。円グラフおよび分割円グラフでのみ使用されます。
yaxisside	integer	グラフアイテムの Y 軸スケールが描画されるグラフの側。 指定可能な値: 0 - *(default)* 左側; 1 - 右側。

graphitem.get

説明

`integer/array graphitem.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってグラフアイテムを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
graphids	ID/array	指定したグラフに属するグラフアイテムのみを返します。
itemids	ID/array	指定したアイテム ID を持つグラフアイテムのみを返します。
type	integer	指定したタイプのグラフアイテムのみを返します。 サポートされているグラフアイテムタイプの一覧については、グラフアイテムオブジェクトのページを参照してください。
selectGraphs	query	アイテムが属するグラフの配列を含むgraphsプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: gitemid。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

グラフからグラフアイテムの取得

グラフで使用されているすべてのグラフアイテムを、アイテムとホストに関する追加情報とともに取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "gitemid": "1242",
            "graphid": "387",
            "itemid": "22665",
            "drawtype": "1",
            "sortorder": "1",
            "color": "FF5555",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1243",
            "graphid": "387",
            "itemid": "22668",
            "drawtype": "1",
            "sortorder": "2",
```

```
            "color": "55FF55",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1244",
            "graphid": "387",
            "itemid": "22671",
            "drawtype": "1",
            "sortorder": "3",
            "color": "009999",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        }
    ],
    "id": 1
}
```


参照

- Graph

ソース

CGraphItem::get() in *ui/include/classes/api/services/CGraphItem.php*.

グラフプロトタイプ

このクラスは、グラフプロトタイプを操作するために設計されています。

オブジェクト参照:

- グラフプロトタイプ

利用可能なメソッド:

- graphprototype.create - 新しいグラフプロトタイプを作成
- graphprototype.delete - グラフプロトタイプを削除
- graphprototype.get - グラフプロトタイプを取得
- graphprototype.update - グラフプロトタイプを更新

Graph prototype オブジェクト

次のオブジェクトは、graphprototypeAPI に直接関連しています。

グラフプロトタイプ

グラフプロトタイプオブジェクトは、以下のプロパティを持ちます。

Property	Type	Description
graphid	ID	グラフプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
height	integer	グラフプロトタイプの高さ（ピクセル単位）。 プロパティの動作: - 作成操作で 必須
name	string	グラフプロトタイプの名前。 プロパティの動作: - 作成操作で 必須

Property	Type	Description
width	integer	グラフプロトタイプの幅（ピクセル単位）。 プロパティの動作: - 作成操作で 必須
flags	integer	グラフプロトタイプの生成元。 指定可能な値: 2 - グラフプロトタイプ; 6 - ディスカバリされたグラフプロトタイプ プロパティの動作: - 読み取り専用
graphtype	integer	グラフプロトタイプのレイアウトタイプ。 指定可能な値: 0 - *(デフォルト)* 通常; 1 - 積み上げ; 2 - 円グラフ; 3 - 分離。
percent_left	float	左パーセンタイル。 デフォルト: 0。
percent_right	float	右パーセンタイル。 デフォルト: 0。
show_3d	integer	ディスカバリされた円グラフおよび分離グラフを 3D で表示するかどうか。 指定可能な値: 0 - *(デフォルト)* 2D で表示; 1 - 3D で表示。
show_legend	integer	ディスカバリされたグラフに凡例を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(デフォルト)* 表示。
show_work_period	integer	ディスカバリされたグラフに稼働時間を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(デフォルト)* 表示。
templateid	ID	親テンプレートのグラフプロトタイプの ID。 プロパティの動作: - 読み取り専用
yaxismax	float	Y 軸の固定最大値。
yaxismin	float	Y 軸の固定最小値。
ymax_itemid	ID	Y 軸の最大値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは `ymax_type` が”calculated” に設定されているものとして描画されます。
ymax_type	integer	Y 軸の最大値の計算方法。 指定可能な値: 0 - *(デフォルト)* calculated; 1 - fixed; 2 - item.
ymin_itemid	ID	Y 軸の最小値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは `ymin_type` が”calculated” に設定されているものとして描画されます。

Property	Type	Description
ymin_type	integer	Y 軸の最小値の計算方法。 指定可能な値: 0 - *(デフォルト)* calculated; 1 - fixed; 2 - item.
discover	integer	グラフプロトタイプのディスカバリ状態。 指定可能な値: 0 - *(デフォルト)* 新しいグラフがディスカバリされる; 1 - 新しいグラフはディスカバリされず、既存のグラフは失われたものとしてマークされる。
uuid	string	インポートされたグラフプロトタイプを既存のものにリンクするために使用される汎用一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - グラフプロトタイプがテンプレートに属している場合に サポート

graphprototype.create

説明

`object graphprototype.create(object/array graphPrototypes)`

このメソッドを使用すると、新しいグラフプロトタイプを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するグラフプロトタイプ。

標準のグラフプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
gitems	array	グラフプロトタイプ用に作成されるグラフアイテム。グラフアイテムはアイテムとアイテムプロトタイプの両方を参照できますが、少なくとも 1 つのアイテムプロトタイプが含まれている必要があります。 パラメータの動作: - 必須

戻り値

(`object`) `graphids` プロパティの下に作成されたグラフプロトタイプの ID を含むオブジェクトを返します。
 返された ID の順番は、渡されたグラフプロトタイプの順番と一致する。

例

グラフプロトタイプの作成

2 つのアイテムを持つグラフプロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Disk space usage {#FSNAME}",
        "width": 900,
```

```
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

関連項目

- グラフアイテム

ソース

CGraphPrototype::create() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

説明

`object graphprototype.delete(array graphPrototypeIds)`

このメソッドを使用すると、グラフプロトタイプを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するグラフプロトタイプの ID。

戻り値

(object) graphids プロパティの下にある削除されたグラフプロトタイプの ID を含むオブジェクトを返します。

例

複数のグラフプロトタイプの削除

2 つのグラフプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

ソース

CGraphPrototype::delete() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

説明

`integer/array graphprototype.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってグラフプロトタイプを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
discoveryids	ID/array	指定されたディスカバリルールに属するグラフプロトタイプのみを返します。
graphids	ID/array	指定された ID を持つグラフプロトタイプのみを返します。
groupids	ID/array	指定されたホストグループまたはテンプレートグループ内のホストまたはテンプレートに属するグラフプロトタイプのみを返します。
hostids	ID/array	指定されたホストに属するグラフプロトタイプのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承されたグラフプロトタイプのみを返します。
itemids	ID/array	指定されたアイテムプロトタイプを含むグラフプロトタイプのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属するグラフプロトタイプのみを返します。
templateids	ID/array	指定されたテンプレートに属するグラフプロトタイプのみを返します。

Parameter	Type	Description
selectDiscoveryData	query	グラフプロトタイプのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。グラフプロトタイプのディスカバリオブジェクトは、検出されたグラフプロトタイプを、それが検出元となったグラフプロトタイプに関連付けます。 以下のプロパティを持ちます: `parent_graphid` - (`ID`) グラフの作成元となったグラフプロトタイプの ID; `status` - (`int`) グラフのディスカバリステータス: 0 - *(default)* グラフプロトタイプは検出されています。 1 - グラフプロトタイプはすでに検出されていません。 `ts_delete` - (`timestamp`) これ以上検出されないグラフプロトタイプが削除される時刻。
selectDiscoveryRule	query	グラフプロトタイプが属する LLD ルールを含む`discoveryRule` プロパティを返します。
selectDiscoveryRulePrototype	query	グラフプロトタイプが属する親 LLD ルールプロトタイプを含む`discoveryRulePrototype` プロパティを返します。
selectGraphItems	query	グラフプロトタイプで使用されるグラフアイテムを含む`gitems` プロパティを返します。
selectHostGroups	query	グラフプロトタイプが属するホストグループを含む`hostgroups` プロパティを返します。
selectHosts	query	グラフプロトタイプが属するホストを含む`hosts` プロパティを返します。
selectItems	query	グラフプロトタイプで使用されるアイテム およびアイテムプロトタイプを含む `items` プロパティを返します。
selectTemplateGroups	query	グラフプロトタイプが属するテンプレートグループを含む`templategroups` プロパティを返します。
selectTemplates	query	グラフプロトタイプが属するテンプレートを含む`templates` プロパティを返します。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が照合対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` data type のプロパティはサポートしていません。 追加のプロパティをサポートしています: `host` - グラフプロトタイプが属するホストの技術名; `hostid` - グラフプロトタイプが属するホストの ID。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

LLD ルールから graph prototypes を取得する

LLD ルールからすべての graph prototypes を取得します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "parent_itemid": "27426",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "discover": "0"
        }
    ],
    "id": 1
}
```

LLD ルールからグラフプロトタイプを取得する

LLD ルールからすべてのグラフプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "graphid": "1017",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "2"
            "discover": "0"
        }
    ],
    "id": 1
}
```


関連項目

- ディスカバリルール
- グラフアイテム
- アイテム
- ホスト
- ホストグループ
- テンプレート
- テンプレートグループ

ソース

CGraphPrototype::get() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

説明

`object graphprototype.update(object/array graphPrototypes)`

このメソッドを使用すると、既存のグラフプロトタイプを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するグラフプロトタイプのプロパティ。

各グラフプロトタイプでは `graphid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

標準のグラフプロトタイプのプロパティに加えて、このメソッドは次のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	既存のグラフアイテムを置き換えるグラフアイテム。グラフアイテムに `gitemid` プロパティが定義されている場合は更新され、そうでない場合は新しいグラフアイテムが作成されます。

戻り値

(object) graphids プロパティの下で更新された graph prototypes の ID を含むオブジェクトを返します。

例

グラフプロトタイプのサイズ変更

グラフプロトタイプのサイズを 1100×400 ピクセルに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

ソース

CGraphPrototype::update() in *ui/include/classes/api/services/CGraphPrototype.php*.

サーバー情報

このクラスは、Zabbix サーバーに関する情報を取得するために設計されています。

利用可能なメソッド:

- serverinfo.get - Zabbix サーバーから最終更新時刻を取得する

serverinfo.get

このメソッドを使用すると、指定した Zabbix サーバーから最終更新時刻を取得できます。

このメソッドは、*User*、*Admin*、*Super admin*、および未認証ユーザーが利用できます。

メソッドのシグネチャ: `object serverinfo.get(object parameters)`

メソッドのソース: `ui/include/classes/api/services/CServerInfo.php` の `CServerInfo::get()`

パラメータ

パラメータ	Type	説明
serverid	ID	Zabbix サーバーの ID。 パラメータの動作: - 必須
output	query	このパラメータについては、リファレンス解説で説明しています。

戻り値

Property	Type	Description
serverid	ID	Zabbix サーバーの ID。
lastaccess	timestamp	Zabbix サーバーから最後に更新された時刻。 高可用性（HA）クラスターの場合、アクティブノードの `lastaccess` の値が表示されます。

例

サーバー情報の取得

指定した Zabbix サーバーの情報を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

サービス

このクラスは、IT インフラストラクチャ/ビジネスサービスを扱うために設計されています。

オブジェクト参照:

- サービス
- ステータスルール
- サービスタグ
- サービスアラーム
- 障害タグ

利用可能なメソッド:

- service.create - 新しいサービスを作成
- service.delete - サービスを削除
- service.get - サービスを取得
- service.update - サービスを更新

Service オブジェクト

次のオブジェクトは、`service`API に直接関連しています。

サービス

サービスオブジェクトには次のプロパティがあります。

Property	Type	Description
serviceid	ID	サービスの ID。 Property behavior: - *read-only* - 更新操作では *required*
algorithm	integer	ステータス計算ルール。子サービスが存在する場合にのみ適用されます。 Possible values: 0 - ステータスを OK に設定する; 1 - すべての子に問題がある場合は最も重大なものを使用する; 2 - 子サービスの中で最も重大なものを使用する。 Property behavior: - 作成操作では *required*
name	string	サービス名。 Property behavior: - 作成操作では *required*
sortorder	integer	並べ替えに使用されるサービスの位置。 Possible values: 0-999. Property behavior: - 作成操作では *required*
weight	integer	サービスの重み。 Possible values: 0-1000000. Default: 0.
propagation_rule	integer	ステータス伝播ルール。 Possible values: 0 - *(default)* サービスのステータスをそのまま伝播する - 変更なし; 1 - 指定した `propagation_value` だけ伝播ステータスを上げる（重大度を 1 から 5 段階分）; 2 - 指定した `propagation_value` だけ伝播ステータスを下げる（重大度を 1 から 5 段階分）; 3 - このサービスを無視する - ステータスは親サービスにまったく伝播されない; 4 - 指定した `propagation_value` を使用して固定のサービスステータスを設定する。 Property behavior: - `propagation_value` が設定されている場合は *required*

Property	Type	Description
propagation_value	integer	ステータス伝播値。 Possible values if `propagation_rule` is set to "0" or "3": 0 - 未分類。 Possible values if `propagation_rule` is set to "1" or "2": 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。 Possible values if `propagation_rule` is set to "4": -1 - OK; 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。 Property behavior: - `propagation_rule` が設定されている場合は *required*
status	integer	サービスが OK 状態か問題状態かを示します。 サービスが問題状態の場合、`status` は最も重大な問題の重大度、または問題状態の子サービスの最も高いステータスのいずれかに等しくなります。 サービスが OK 状態の場合、`status` は -1 に等しくなります。 Property behavior: - *read-only*
description	string	サービスの説明。
uuid	string	インポートされたサービスを既存のサービスにリンクするために使用される、ユニバーサル一意識別子です。指定されていない場合は自動生成されます。
created_at	integer	サービスが作成された時刻の Unix タイムスタンプ。
readonly	integer	サービスへのアクセス。 Possible values: 0 - 読み書き可能; 1 - 読み取り専用。 Property behavior: - *read-only*

ステータスルール

ステータスルールオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
type	integer	(新しいステータス) ステータスを設定する条件。 使用可能な値: 0 - 少なくとも (N) 個の子サービスが (ステータス) ステータス以上である場合; 1 - 少なくとも (N%) の子サービスが (ステータス) ステータス以上である場合; 2 - (ステータス) ステータス以下の子サービスが (N) 個未満である場合; 3 - (ステータス) ステータス以下の子サービスが (N%) 未満である場合; 4 - (ステータス) ステータス以上の子サービスの重みの合計が少なくとも (W) である場合; 5 - (ステータス) ステータス以上の子サービスの重みの合計が少なくとも (N%) である場合; 6 - (ステータス) ステータス以下の子サービスの重みの合計が (W) 未満である場合; 7 - (ステータス) ステータス以下の子サービスの重みの合計が (N%) 未満である場合。 ここで: - N (W) は `limit_value`; - (ステータス) は `limit_status`; - (新しいステータス) は `new_status`。 プロパティの動作: - *required*
limit_value	integer	制限値。 使用可能な値: - N および W の場合: 1-100000; - N% の場合: 1-100。 プロパティの動作: - *required*
limit_status	integer	制限ステータス。 使用可能な値: -1 - OK; 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 プロパティの動作: - *required*
new_status	integer	新しいステータス値。 使用可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 プロパティの動作: - *required*

サービスタグ

サービスタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	サービスタグ名。 プロパティの動作: - 必須
value	string	サービスタグの値。

サービスアラーム

Note:
サービスアラームは、Zabbix API を使用して直接作成、更新、削除することはできません。

サービスアラームオブジェクトは、サービスの状態変化を表します。次のプロパティがあります。

Property	Type	Description
clock	timestamp	サービスの状態変化が発生した時刻。
value	integer	サービスのステータス。 取りうる値の一覧については、service status property を参照してください。

問題タグ

問題タグを使用すると、サービスを問題イベントに関連付けることができます。問題タグオブジェクトには、次のプロパティがあります。

Property	Type	Description
tag	string	問題タグ名。 Property behavior: - *required*
operator	integer	マッピング条件のoperator。 指定可能な値: 0 - *(default)* 等しい; 2 - 含む。
value	string	問題タグの値。

service.create

説明

`object service.create(object/array services)`

このメソッドを使用すると、新しいサービスを作成できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するサービス。

標準のサービスプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
children	array	サービスにリンクする子サービス。 子サービスでは、serviceid プロパティのみを定義する必要があります。
parents	array	サービスにリンクする親サービス。 親サービスでは、serviceid プロパティのみを定義する必要があります。
tags	array	サービス用に作成するサービスタグ。
problem_tags	array	サービス用に作成する障害タグ。
status_rules	array	サービス用に作成するステータスルール。

戻り値

(object) serviceids プロパティの下で作成されたサービスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたサービスの順序と一致します。

例

サービスの作成

少なくとも 1 つの子プロセスに問題がある場合、問題状態に切り替えるサービスを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CService::create() in *ui/include/classes/api/services/CService.php*.

service.delete

説明

`object service.delete(array serviceIds)`

このメソッドを使用すると、サービスを削除できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメーター

(array) 削除するサービスの ID。

戻り値

(object) serviceids プロパティの下にある削除されたサービスの ID を含むオブジェクトを返します。

例

複数のサービスの削除

2 つのサービスを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CService::delete() in *ui/include/classes/api/services/CService.php*.

service.get

説明

`integer/array service.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってサービスを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
serviceids	ID/array	指定した ID のサービスのみを返します。
parentids	ID/array	指定した親サービスにリンクされているサービスのみを返します。
deep_parentids	flag	すべての直接および間接の子サービスを返します。parentids と一緒に使用します。
childids	ID/array	指定した子サービスにリンクされているサービスのみを返します。

パラメータ	型	説明
evaltype	integer	タグの評価方法です。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	object/array	指定したタグを持つサービスのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのサービスを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist。
problem_tags	object/array	指定した障害タグを持つサービスのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのサービスを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist。
without_problem_tags	flag	障害タグを持たないサービスのみを返します。
slaids	ID/array	特定の SLA にリンクされているサービスのみを返します。
selectChildren	query	子サービスを含む `children` プロパティを返します。 `count` をサポートします。
selectParents	query	親サービスを含む `parents` プロパティを返します。 `count` をサポートします。
selectTags	query	サービスタグを含む`tags` プロパティを返します。 `count` をサポートします。
selectProblemEvents	query	障害イベントオブジェクトの配列を含む `problem_events` プロパティを返します。 障害イベントオブジェクトは以下のプロパティを持ちます: `eventid` - (`ID`) イベント ID; `severity` - (`string`) 現在のイベントの重大度; `name` - (`string`) 解決済みイベント名。 `count` をサポートします。
selectProblemTags	query	障害タグを含む`problem_tags` プロパティを返します。 `count` をサポートします。
selectStatusRules	query	ステータスルールを含む`status_rules` プロパティを返します。 `count` をサポートします。

パラメータ	型	説明
selectStatusTimeline	object/array	指定した期間のサービス状態の変化を含む `status_timeline` プロパティを返します。 形式 `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` は対象期間の開始日時（含む; integer timestamp）、`period_to` は対象期間の終了日時（含まない; integer timestamp）です。 指定した期間内の状態変化について、`start_value` プロパティと`alarms` 配列を含むエントリの配列を返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `serviceid`, `name`, `status`, `sortorder`, `created_at`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのサービスの取得

すべてのサービスとその依存関係に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
```

```
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
            "name": "Zabbix server node 2",
```

```
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


ソース

CService::get() in *ui/include/classes/api/services/CService.php*.

service.update

説明

`object service.update(object/array services)`

このメソッドを使用すると、既存のサービスを更新できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するサービスのプロパティ。

各サービスについて serviceid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

標準のサービスプロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
children	array	現在の子サービスを置き換える子サービス。 子サービスでは serviceid プロパティのみを定義する必要があります。
parents	array	現在の親サービスを置き換える親サービス。 親サービスでは serviceid プロパティのみを定義する必要があります。
tags	array	現在のサービスタグを置き換えるサービスタグ。
problem_tags	array	現在の障害タグを置き換える障害タグ。
status_rules	array	現在のステータスルールを置き換えるステータスルール。

戻り値

(object) serviceids プロパティの下で更新されたサービスの ID を含むオブジェクトを返します。

例

サービスの親の設定

ID ”3” のサービスを、ID ”5” のサービスの親にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

スケジュールされたダウンタイムの追加

ID ”4” のサービスに対して、毎週月曜日 22:00 から火曜日 10:00 までのダウンタイムを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

ソース

CService::update() in *ui/include/classes/api/services/CService.php*.

スクリプト

このクラスはスクリプトを扱うために設計されています。

オブジェクト参照:

- Script
  - Webhook パラメータ
- Debug
  - ログエントリ

利用可能なメソッド:

- script.create - 新しいスクリプトを作成
- script.delete - スクリプトを削除
- script.execute - スクリプトを実行
- script.get - スクリプトを取得
- script.getscriptsbyevents - イベントのスクリプトを取得
- script.getscriptsbyhosts - ホストのスクリプトを取得
- script.update - スクリプトを更新

Script オブジェクト

次のオブジェクトは、scriptAPI に直接関連しています。

スクリプト

スクリプトオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
scriptid	ID	スクリプトの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	スクリプトの名前。 プロパティの動作: - 作成操作では必須
type	integer	スクリプトのタイプ。 scope に「action operation」が設定されている場合の使用可能な値: 0 - スクリプト; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - webhook。 scope に「manual host action」または「manual event action」が設定されている場合の使用可能な値: 6 - URL。 プロパティの動作: - 作成操作では必須
command	string	実行するコマンド。 プロパティの動作: - type に「Script」、「IPMI」、「SSH」、「TELNET」、または「Webhook」が設定されている場合は必須

プロパティ	タイプ	説明
scope	integer	スクリプトのスコープ。 使用可能な値: 1 - アクション操作; 2 - 手動ホストアクション; 4 - 手動イベントアクション。 プロパティの動作: - 作成操作では必須
execute_on	integer	スクリプトを実行する場所。 使用可能な値: 0 - Zabbix エージェントで実行; 1 - Zabbix サーバーで実行。Zabbix サーバーでグローバルスクリプトの実行が有効になっている場合のみサポートされます; 2 - *(デフォルト)* Zabbix サーバーまたはプロキシで実行。 プロパティの動作: - type に「Script」が設定されている場合はサポート
menu_path	string	ホストまたはイベントをクリックしたときに Web インターフェースに表示される、メニューのようなナビゲーションを構成するスラッシュ区切りのフォルダー。 プロパティの動作: - scope に「manual host action」または「manual event action」が設定されている場合はサポート
authtype	integer	SSH スクリプトタイプで使用する認証方式。 使用可能な値: 0 - パスワード; 1 - 公開鍵。 プロパティの動作: - type に「SSH」が設定されている場合はサポート
username	string	認証に使用するユーザー名。 プロパティの動作: - type に「SSH」または「TELNET」が設定されている場合は必須
password	string	パスワード認証を使用する SSH スクリプトおよび TELNET スクリプトのパスワード。 プロパティの動作: - type に「SSH」が設定され、authtype に「password」が設定されている場合、または type に「TELNET」が設定されている場合はサポート
publickey	string	公開鍵認証を使用する SSH スクリプトで使用する公開鍵ファイルの名前。 プロパティの動作: - type に「SSH」が設定され、authtype に「public key」が設定されている場合は必須
privatekey	string	公開鍵認証を使用する SSH スクリプトで使用する秘密鍵ファイルの名前。 プロパティの動作: - type に「SSH」が設定され、authtype に「public key」が設定されている場合は必須
port	string	SSH スクリプトおよび TELNET スクリプトで使用するポート番号。 プロパティの動作: - type に「SSH」または「TELNET」が設定されている場合はサポート

プロパティ	タイプ	説明
groupid	ID	スクリプトを実行できるホストグループの ID。 「0」に設定すると、すべてのホストグループでスクリプトを使用できます。 デフォルト: 0。
usrgrpid	ID	スクリプトの実行を許可するユーザーグループの ID。 「0」に設定すると、すべてのユーザーグループでスクリプトを使用できます。 デフォルト: 0。
host_access	integer	プロパティの動作: - scope に「manual host action」または「manual event action」が設定されている場合はサポート スクリプトの実行に必要なホスト権限。 使用可能な値: 2 - *(デフォルト)* 読み取り; 3 - 書き込み。
confirmation	string	プロパティの動作: - scope に「manual host action」または「manual event action」が設定されている場合はサポート 確認ポップアップのテキスト。 Zabbix の Web インターフェースからスクリプトを実行しようとすると、ポップアップが表示されます。
timeout	string	プロパティの動作: - scope に「manual host action」または「manual event action」が設定されている場合はサポート webhook スクリプトの実行タイムアウト（秒）。時間のサフィックスがサポートされています（例: 30s、1m）。 使用可能な値: 1～60 秒。 デフォルト: 30 秒。
parameters	array	プロパティの動作: - type に「Webhook」が設定されている場合は必須 webhook 入力パラメータの配列。
description	string	プロパティの動作: - type に「Webhook」が設定されている場合はサポート スクリプトの説明。
url	string	ユーザー定義の URL。 プロパティの動作: - type に「URL」が設定されている場合は必須
new_window	integer	URL を新しいウィンドウで開くかどうか。 使用可能な値: 0 - いいえ; 1 - *(デフォルト)* はい。 プロパティの動作: - type に「URL」が設定されている場合はサポート

プロパティ	タイプ	説明
manualinput	integer	スクリプトがユーザー入力を受け付けるかどうかを示します。 使用可能な値: 0 - *(デフォルト)* 無効; 1 - 有効; プロパティの動作: - `scope` に「manual host action」または「manual event action」が設定されている場合はサポート
manualinput_prompt	string	手動入力プロンプトのテキスト。 プロパティの動作: - `manualinput` に「Enabled」が設定されている場合は必須
manualinput_validator	string	ユーザーが入力した値の検証に使用する文字列フィールド。文字列は、正規表現またはカンマ区切りの値のセットで構成されます。 プロパティの動作: - `manualinput` に「Enabled」が設定されている場合は必須
manualinput_validator_type	integer	想定されるユーザー入力のタイプを決定します。 使用可能な値: 0 - *(デフォルト)* 文字列。manualinput_validator を正規表現として扱うことを示します; 1 - リスト。manualinput_validator を、入力可能な値のカンマ区切りリストとして扱うことを示します。 プロパティの動作: - `manualinput` に「Enabled」が設定されている場合はサポート
manualinput_default_value	string	ユーザー入力を自動入力する際のデフォルト値。 プロパティの動作: - `manualinput_validator_type` に「String」が設定されている場合はサポート

webhook パラメータ

webhook スクリプトが呼び出されたときに渡されるパラメータには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータ名。 プロパティの動作: - 必須
value	string	パラメータ値。マクロをサポートします。

デバッグ

実行された webhook スクリプトのデバッグ情報です。debug オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
logs	array	ログエントリの配列。
ms	string	ミリ秒単位のスクリプト実行時間。

ログエントリ

ログエントリオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
level	integer	ログレベル。

プロパティ	型	説明
ms	string	スクリプトの実行開始からログエントリが追加されるまでに経過した時間（ミリ秒）。
message	string	ログメッセージ。

script.create

説明

`object script.create(object/array scripts)`

このメソッドを使用すると、新しいスクリプトを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受け入れます

戻り値

(object) scriptids プロパティの下で作成されたスクリプトの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスクリプトの順序と一致します。

例

webhook スクリプトの作成

外部サービスに HTTP リクエストを送信する webhook スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Webhook script",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "scope": 1,
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

SSH スクリプトの作成

ホスト上で実行でき、コンテキストメニューを持つ公開鍵認証付きの SSH スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "SSH script",
        "command": "my script command",
        "type": 2,
        "authtype": 1,
        "username": "John",
        "publickey": "pub.key",
        "privatekey": "priv.key",
        "password": "secret",
        "port": "12345",
        "scope": 2,
        "menu_path": "All scripts/SSH",
        "usrgrpid": "7",
        "groupid": "4"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

カスタムスクリプトの作成

サーバーを再起動するカスタムスクリプトを作成します。このスクリプトはホストへの書き込み権限を必要とし、ユーザーに手動入力を促します。入力が正常に送信されると、スクリプトはフロントエンドに確認メッセージを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Reboot server",
        "command": "reboot server {MANUALINPUT}",
        "type": 0,
        "scope": 2,
        "confirmation": "Are you sure you would like to reboot the server {MANUALINPUT}?",
        "manualinput": 1,
        "manualinput_prompt": "Which server you want to reboot?",
        "manualinput_validator": "[1-9]",
        "manualinput_validator_type": 0,
        "manualinput_default_value": "1"
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

URL タイプのスクリプト作成

同じウィンドウに表示され、確認テキストを含む、ホストスコープ用の URL タイプのスクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

手動入力が可能な URL 型スクリプト作成

新しいウィンドウで開き、手動入力が可能なイベントスコープ用の URL 型スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script with manual input",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Select a page to open:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

ソース

CScript::create() in *ui/include/classes/api/services/CScript.php*.

script.delete

説明

`object script.delete(array scriptIds)`

このメソッドを使用すると、スクリプトを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するスクリプトの ID。

戻り値

(object) scriptids プロパティの下にある削除されたスクリプトの ID を含むオブジェクトを返します。

例

複数のスクリプトの削除

2 つのスクリプトを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

ソース

CScript::delete() in *ui/include/classes/api/services/CScript.php*.

script.execute

説明

`object script.execute(object parameters)`

このメソッドを使用すると、ホストまたはイベントでスクリプトを実行できます。ただし、URL タイプのスクリプトは除きます。これらは実行できません。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 実行するスクリプトの ID、およびホストの ID またはイベントの ID と manualinput 値を含むパラメータ。

パラメータ	Type	説明
scriptid	ID	実行するscriptの ID。 Parameter behavior: - *required*
hostid	ID	スクリプトを実行するhostの ID。 Parameter behavior: - `eventid` が設定されていない場合は *required*
eventid	ID	スクリプトを実行するeventの ID。 Parameter behavior: - `hostid` が設定されていない場合は *required*
manualinput	string	{MANUALINPUT} マクロを置換してスクリプトを実行するための、ユーザー指定の値。

戻り値

(`object`) スクリプト実行の結果を返します。

プロパティ	Type	説明
response	string	スクリプトが正常に実行されたかどうか。 指定可能な値 - `success`。
value	string	スクリプトの出力。
debug	object	webhook スクリプトが実行された場合は、debug オブジェクトを含みます。その他のスクリプトタイプでは、空のオブジェクトを含みます。

例

Webhook スクリプトの実行

外部サービスに HTTP リクエストを送信する Webhook スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "{\"status\":\"sent\",\"timestamp\":\"1611235391\"}",
        "debug": {
            "logs": [
                {
                    "level": 3,
                    "ms": 480,
                    "message": "[Webhook Script] HTTP status: 200."
                }
            ],
            "ms": 495
        }
    },
    "id": 1
}
```

カスタムスクリプトの実行

ホストで"ping" スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_req=1 tt
        "debug": []
    },
    "id": 1
}
```

手動入力でカスタムスクリプトを実行する

ホスト上で、コマンド `ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac` を使用して"ping" スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_seq=1 tt
        "debug": []
    },
    "id": 1
}
```

ソース

CScript::execute() in *ui/include/classes/api/services/CScript.php*.

script.get

説明

`integer/array script.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってスクリプトを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
groupids	ID/array	指定したホストグループで実行可能なスクリプトのみを返します。
hostids	ID/array	指定したホストで実行可能なスクリプトのみを返します。
scriptids	ID/array	指定した ID を持つスクリプトのみを返します。
usrgrpids	ID/array	指定したユーザーグループのユーザーが実行可能なスクリプトのみを返します。
selectHostGroups	query	スクリプトを実行可能なホストグループを含む`hostgroups` プロパティを返します。
selectHosts	query	スクリプトを実行可能なホストを含む`hosts` プロパティを返します。
selectActions	query	スクリプトに関連付けられているアクションを含む`actions` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `scriptid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列

- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのスクリプトを取得

設定されているすべてのスクリプトを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
```

```
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Webhook",
            "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "5",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
```

```
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": [
                {
                    "name": "token",
                    "value": "{$WEBHOOK.TOKEN}"
                },
                {
                    "name": "host",
                    "value": "{HOST.HOST}"
                },
                {
                    "name": "v",
                    "value": "2.2"
                }
            ]
        },
        {
            "scriptid": "5",
            "name": "URL",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Go to {HOST.NAME}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action=latest.view&hostids[]={HOST.ID}",
            "new_window": "0",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "6",
            "name": "URL with user input",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
```

```
            "confirmation": "Open zabbix page {MANUALINPUT}?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
            "new_window": "0",
            "manualinput": "1",
            "manualinput_prompt": "Select a page to open:",
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```

関連項目

- ホスト
- ホストグループ

ソース

CScript::get() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

説明

`object script.getscriptsbyevents(object parameters)`

このメソッドを使用すると、指定したイベントで利用可能なすべてのスクリプト、またはスクリプト ID が指定されている場合は特定のスクリプトを取得できます。manualinput が指定されている場合、{MANUALINPUT} マクロは指定された値に置き換えられます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーの役割設定で取り消すことができます。詳細はユーザーの役割 を参照してください。

パラメータ

(object/array) このメソッドは、以下のパラメータを持つオブジェクト、またはオブジェクトの配列を受け付けます。

パラメータ	型	説明
eventid	ID	スクリプトを返す対象のイベントの ID。 一意である必要があります。 パラメータの動作: - 必須
scriptid	ID	返すスクリプトの ID。
manualinput	string	ユーザーが指定した {MANUALINPUT} マクロの値。

戻り値

(object) は、イベント ID をプロパティとして、利用可能なスクリプトの配列を値として持つオブジェクトを返します。スクリプト ID が指定されている場合、関連付けられた値は特定のスクリプトを含む配列です。

Note:
このメソッドは、confirmation テキスト、manualinput prompt テキスト、および url 内のマクロを自動的に展開します。
manualinput パラメータが指定されている場合、{MANUALINPUT} マクロは指定された値に解決されます。

例

イベント ID でスクリプトを取得する

イベント"632" および"614" に使用できるすべてのスクリプトを取得します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
```

```
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
```

```
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "614": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "1",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
```

```
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
```

```
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```


manualinput 値を持つ特定のスクリプトの情報を取得します。

イベント”632” の ID”4”、manualinput 値”dashboard.view” を持つスクリプトの情報を取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```


Response:


```
{
    "jsonrpc": "2.0",
    "result": {
      "632": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
```

```
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        }
      ]
    },
    "id": 1
}
```

ソース

CScript::getScriptsByEvents() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

説明

`object script.getscriptsbyhosts(object parameters)`

このメソッドを使用すると、指定したホストで利用可能なすべてのスクリプト、または script ID が指定されている場合は特定のスクリプトを取得できます。manualinput が指定されている場合、{MANUALINPUT} マクロは指定された値に置き換えられます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) このメソッドは、以下のパラメータを持つオブジェクト、またはオブジェクトの配列を受け付けます。

パラメータ	Type	説明
hostid	ID	スクリプトの返却対象となるホストの ID。 一意である必要があります。 Parameter behavior: - 必須
scriptid	ID	返却するスクリプトの ID。
manualinput	string	ユーザーが指定した {MANUALINPUT} マクロの値。

戻り値

(object) ホスト ID をプロパティ、利用可能なスクリプトの配列を値とするオブジェクトを返します。スクリプト ID が指定された場合、関連する値は特定のスクリプトを含む配列となります。

> **Note:**
> このメソッドは、confirmation テキスト、manualinput prompt テキスト、url 内のマクロを自動的に展開します。manualinput パラメータが指定された場合、{MANUALINPUT} マクロは指定された値に解決されます。

例

ホスト ID によるスクリプトの取得

ホスト"30079" と"30073" で利用可能なすべてのスクリプトを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
```

```
    "params": [
        {
            "hostid":  "30079"
        },
        {
            "hostid":  "30073"
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
```

```
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "0",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
```

```
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "30073": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
```

```
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
```

```
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

manualinput 値を持つ特定のスクリプトの情報を取得します。

ホスト”30079” 上の ID ”4”、manualinput 値”dashboard.view” を持つスクリプトの情報を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "30079": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "2",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
    },
    "id": 1
}
```

ソース

CScript::getScriptsByHosts() in *ui/include/classes/api/services/CScript.php*.

script.update

説明

`object script.update(object/array scripts)`

このメソッドを使用すると、既存のスクリプトを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) スクリプトのプロパティが更新されます。

`scriptid` プロパティはスクリプトごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。例外は `type` プロパティが 5（Webhook）からその他に変更された場合です。その場合は `parameters` プロパティはクリーンアップされます。

戻り値

(object) `scriptids` プロパティの下で更新されたスクリプトの ID を含むオブジェクトを返します。

例

スクリプトコマンドの変更

スクリプトのコマンドを”/bin/ping -c 10 {HOST.CONN} 2>&1” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

スクリプトコマンドを変更して、手動入力を追加します

スクリプトのコマンドを”/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
```

```
        "manualinput_prompt": "Specify the number of ICMP packets to send with the ping command",
        "manualinput_validator": "^(?:[1-9]|10)$",
        "manualinput_validator_type": "0",
        "manualinput_default_value": "10"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CScript::update() in *ui/include/classes/api/services/CScript.php*.

タスク

このクラスは、タスク（設定の再読み込みを行わずにアイテムやローレベルディスカバリルールをチェックするなど）を扱うために設計されています。

オブジェクト参照:

- タスク
  - '今すぐ実行' リクエストオブジェクト
  - 'プロキシ設定の更新' リクエストオブジェクト
  - '診断情報' リクエストオブジェクト
    * 統計リクエストオブジェクト
    * 統計結果オブジェクト

利用可能なメソッド:

- task.create - 新しいタスクを作成
- task.get - タスクを取得

タスクオブジェクト

以下のオブジェクトは task API に直接関連しています。

タスク

タスクオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
taskid	ID	タスクの ID。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
type	integer	タスクのタイプ。 指定可能な値: 1 - 診断情報; 2 - プロキシ設定の更新; 6 - 今すぐ実行。 *Admin* および *User* タイプのユーザーは、「今すぐ実行」タスクのみ作成できます。 `task.get` は常に"7"（タスク実行サマリー）を返すことに注意してください。 プロパティの動作: - 必須
status	integer	タスクのステータス。 指定可能な値: 1 - 新規タスク; 2 - 進行中のタスク; 3 - タスクは完了している; 4 - タスクの有効期限切れ。 プロパティの動作: - 読み取り専用
clock	timestamp	タスクが作成された時刻。 プロパティの動作: - 読み取り専用
ttl	integer	タスクの有効期限が切れるまでの秒数。 プロパティの動作: - 読み取り専用
proxyid	ID	診断情報の統計情報が収集される対象のプロキシの ID。 プロパティの動作: - `type` が「診断情報」または「プロキシ設定の更新」に設定されている場合に サポート
request	object	タスクタイプに応じたタスクリクエストオブジェクト: 「今すぐ実行」タスクのオブジェクトは、以下で詳しく説明されています。 「プロキシ設定の更新」タスクのオブジェクトは、以下で詳しく説明されています。 「診断情報」タスクのオブジェクトは、以下で詳しく説明されています。 プロパティの動作: - 必須
result	object	診断情報タスクの結果オブジェクト。 結果の準備がまだできていない場合は NULL を含むことがあります。 結果オブジェクトは、以下で詳しく説明されています。 プロパティの動作: - 読み取り専用

「今すぐ実行」リクエストオブジェクト

「今すぐ実行」タスクリクエストオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	アイテムおよびローレベルディスカバリルールの ID。 *Admin* および *User* タイプのユーザーは、*read-write* permission を持つホスト上のアイテム、または *read* permission を持ち、かつそのロールで `invoke_execute_now` action が有効になっている場合に、「今すぐ実行」を実行できます。ローレベルディスカバリ（LLD）ルールについても、*Admin* タイプのユーザーに同様に適用されます。

「プロキシ設定のリフレッシュ」リクエストオブジェクト

「プロキシ設定のリフレッシュ」タスクリクエストオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
proxyids	array	プロキシ ID。

「Diagnostic information」リクエストオブジェクト

診断情報タスクリクエストオブジェクトには、以下のプロパティがあります。すべての種類のプロパティに対する統計リクエストオブジェクトについては、以下で詳しく説明しています。

プロパティ	型	説明
historycache	object	履歴キャッシュの統計リクエスト。サーバーおよびプロキシで使用できます。
valuecache	object	アイテムキャッシュの統計リクエスト。サーバーで使用できます。
preprocessing	object	前処理マネージャーの統計リクエスト。サーバーおよびプロキシで使用できます。
alerting	object	アラートマネージャーの統計リクエスト。サーバーで使用できます。
lld	object	LLD マネージャーの統計リクエスト。サーバーで使用できます。

統計リクエストオブジェクト

統計リクエストオブジェクトは、サーバー/プロキシの内部プロセスについてどの種類の情報を収集するかを定義するために使用されます。以下のプロパティがあります。

Property	Type	Description
stats	query	返される統計オブジェクトのプロパティ。 診断情報統計の各タイプで利用可能なフィールドの一覧は、以下で詳しく説明されています。 デフォルト: `extend` は利用可能なすべての統計フィールドを返します。
top	object	返される統計値をソートおよび制限するためのオブジェクト。 診断情報統計の各タイプで利用可能なフィールドの一覧は、以下で詳しく説明されています。 例: { “source.alerts”: 10 }

各種診断情報リクエストタイプで利用可能な統計フィールドの一覧

各種診断情報リクエストプロパティでリクエスト可能な統計フィールドは以下の通りです。

診断タイプ	利用可能なフィールド	説明
historycache	items	キャッシュされたアイテム数
	values	キャッシュされた値の数
	memory	共有メモリの統計情報 (空き領域、使用中チャンク数、空きチャンク数、最大空きチャンクサイズ)
	memory.data	ヒストリデータキャッシュ共有メモリの統計情報
	memory.index	ヒストリインデックスキャッシュ共有メモリの統計情報
valuecache	items	キャッシュされたアイテム数
	values	キャッシュされた値の数
	memory	共有メモリの統計情報 (空き領域、使用中チャンク数、空きチャンク数、最大空きチャンクサイズ)

診断タイプ	利用可能なフィールド	説明
	mode	値キャッシュモード
preprocessing	values	キューイングされた値の数
	preproc.values	前処理ステップを持つキューイングされた値の数
alerting	alerts	キューイングされたアラートの数
lld	rules	キューイングされたルールの数
	values	キューイングされた値の数

各種診断情報リクエストで利用可能なソートフィールドの一覧

以下の統計フィールドを使用して、要求された情報のソートおよび制限を行うことができます。

診断タイプ	利用可能なフィールド	Type
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

統計結果オブジェクト

統計結果オブジェクトは、タスクオブジェクトの `result` フィールドで取得されます。

プロパティ	Type	説明
status	integer	タスク結果のステータス。 指定可能な値: -1 - タスクの実行中にエラーが発生した。 0 - タスク結果が作成された。 プロパティの動作: - 読み取り専用
data	string/object	特定の診断情報タスクの統計リクエストオブジェクトに応じた結果。 タスクの実行中にエラーが発生した場合は、エラーメッセージ文字列が含まれます。

task.create

説明

`object task.create(object/array tasks)`

このメソッドを使用すると、タスクを作成できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するタスク。

このメソッドは標準タスクプロパティを持つタスクを受け付けます。

「今すぐ実行」タスクは、以下のタイプのアイテム/ディスカバリールールに対してのみ作成できます:

- Zabbix エージェント（パッシブ）
- シンプルチェック
- SNMP エージェント（v1/v2/v3）
- Zabbix 内部

- 外部チェック
- データベース監視
- HTTP エージェント
- IPMI エージェント
- SSH エージェント
- TELNET エージェント
- JMX エージェント
- 計算
- 従属アイテム
- スクリプト
- ブラウザ

アイテム/ディスカバリールールが「従属アイテム」タイプの場合、そのマスターアイテムも上記のいずれかのタイプである必要があります。

戻り値

(object) taskids プロパティの下で作成されたタスクの ID を含むオブジェクトを返します。アイテムおよびローレベルディスカバリルールごとに 1 つのタスクが作成されます。返される ID の順序は、渡された itemids の順序と一致します。

例

タスクの作成

アイテムとローレベルディスカバリールールの「今すぐ実行」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

2 つのプロキシの「プロキシ設定のリフレッシュ」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
```

```
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

「診断情報」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 1,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "proxyid": 0
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

参考

- Task
- ' 今すぐ実行' リクエストオブジェクト
- ' プロキシ設定のリフレッシュ' リクエストオブジェクト
- ' 診断情報' リクエストオブジェクト

ソース

CTask::create() in *ui/include/classes/api/services/CTask.php*.

task.get

説明

`integer/array task.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってタスクを取得できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	Type	説明
taskids	ID/array	指定された ID を持つタスクのみを返します。
output	query	これらのパラメータはリファレンスコメントで説明されています。
preservekeys	boolean	

戻り値

(integer/array) オブジェクトの配列を返します

例

ID でタスクを取得

ID が"1" のタスクに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "taskid": "1",
            "type": "7",
            "status": "3",
            "clock": "1601039076",
            "ttl": "3600",
            "proxyid": null,
```

```
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "result": {
                "data": {
                    "alerting": {
                        "alerts": 0,
                        "top": {
                            "media.alerts": []
                        },
                        "time": 0.000663
                    },
                    "lld": {
                        "rules": 0,
                        "values": 0,
                        "top": {
                            "values": []
                        },
                        "time": 0.000442
                    }
                },
                "status": "0"
            }
        }
    ],
    "id": 1
}
```


参考

- Task
- ' 今すぐ実行' リクエストオブジェクト
- ' プロキシ設定のリフレッシュ' リクエストオブジェクト
- ' 診断情報' リクエストオブジェクト

ソース

CTask::get() in *ui/include/classes/api/services/CTask.php*.

ダッシュボード

このクラスはダッシュボードを操作するために設計されています。

オブジェクト参照:

- ダッシュボード
- ダッシュボードページ
    - ダッシュボードウィジェット
        * ダッシュボードウィジェットフィールド
- ダッシュボードユーザーグループ
- ダッシュボードユーザー

利用可能なメソッド:

- dashboard.create - 新しいダッシュボードを作成
- dashboard.delete - ダッシュボードを削除
- dashboard.get - ダッシュボードを取得
- dashboard.update - ダッシュボードを更新

Dashboard オブジェクト

次のオブジェクトは、dashboardAPI に直接関連しています。

ダッシュボード

dashboard オブジェクトには次のプロパティがあります。

Property	Type	Description
dashboardid	ID	ダッシュボードの ID。 Property behavior: - *read-only* - 更新操作では *required*
name	string	ダッシュボードの名前。 Property behavior: - 作成操作では *required*
userid	ID	ダッシュボードの所有者であるユーザーの ID。
private	integer	ダッシュボード共有の種類。 Possible values: 0 - public dashboard; 1 - *(default)* private dashboard.
display_period	integer	デフォルトのスライドショー間隔（秒）。 Possible values: 10, 30, 60, 120, 600, 1800, 3600. Default: 30.
auto_start	integer	スライドショーを自動開始するかどうか。 Possible values: 0 - do not auto start slideshow; 1 - *(default)* auto start slideshow.

ダッシュボードページ

ダッシュボードページオブジェクトには、次のプロパティがあります。

Property	Type	Description
dashboard_pageid	ID	ダッシュボードページの ID。 Property behavior: - *read-only*
name	string	ダッシュボードページ名。 Default: empty string.
display_period	integer	ダッシュボードのスライドショー間隔（秒）。 Possible values: 0, 10, 30, 60, 120, 600, 1800, 3600. Default: 0 (will use the default slideshow interval).
widgets	array	ダッシュボードウィジェットオブジェクトの配列。

ダッシュボードウィジェット

ダッシュボードウィジェットオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
widgetid	ID	ダッシュボードウィジェットの ID。 プロパティの動作: - 読み取り専用
type	string	ダッシュボードウィジェットのタイプ。 指定可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリーステータス; favgraphs - お気に入りグラフ; favmaps - お気に入りマップ; gauge - ゲージ; geomap - ジオマップ; graph - グラフ（クラシック）; graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホストの可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemhistory - アイテム履歴; itemnavigator - アイテムナビゲーター; item - アイテムの値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; problemhosts - 障害ホスト; problems - 障害; problemsbysv - 深刻度別の障害; scatterplot - 散布図; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - 上位ホスト; topitems - 上位アイテム; toptriggers - 上位トリガー; trigover - トリガー概要; url - URL; web - Web 監視。 プロパティの動作: - 必須
name	string	カスタムウィジェット名。
x	integer	ダッシュボードの左側からの水平方向の位置。 指定可能な値の範囲は 0 から 71 です。
y	integer	ダッシュボードの上側からの垂直方向の位置。 指定可能な値の範囲は 0 から 63 です。
width	integer	ウィジェットの幅。 指定可能な値の範囲は 1 から 72 です。
height	integer	ウィジェットの高さ。 指定可能な値の範囲は 1 から 64 です。

プロパティ	型	説明
view_mode	integer	ウィジェットの表示モード。 指定可能な値: 0 - *(デフォルト)* デフォルトのウィジェット表示; 1 - ヘッダーを非表示;
fields	array	ダッシュボードウィジェットフィールドオブジェクトの配列。 プロパティの動作: - ダッシュボードウィジェットフィールド内の各ウィジェットを参照

ダッシュボードウィジェットフィールド

ダッシュボードウィジェットフィールドオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
type	integer	ウィジェットフィールドのタイプ。 使用可能な値: 0 - 整数; 1 - 文字列; 2 - ホストグループ; 3 - ホスト; 4 - アイテム; 5 - アイテムのプロトタイプ; 6 - グラフ; 7 - グラフのプロトタイプ; 8 - マップ; 9 - サービス; 10 - SLA; 11 - ユーザー; 12 - アクション; 13 - メディアタイプ。 プロパティの動作: - 必須
name	string	ウィジェットフィールド名。 使用可能な値: ダッシュボードウィジェットフィールドを参照してください。 プロパティの動作: - 必須
value	mixed	タイプに応じたウィジェットフィールドの値。 使用可能な値: ダッシュボードウィジェットフィールドを参照してください。 プロパティの動作: - 必須

ダッシュボードユーザーグループ

ユーザーグループに基づくダッシュボード権限の一覧です。以下のプロパティがあります。

プロパティ	Type	説明
usrgrpid	ID	ユーザーグループの ID。 Property behavior: - *required*

プロパティ	Type	説明
permission	integer	権限レベルの種類。 使用可能な値: 2 - 読み取り専用; 3 - 読み書き可能。 Property behavior: - *required*

ダッシュボードユーザー

ユーザーに基づくダッシュボード権限の一覧です。以下のプロパティがあります。

プロパティ	Type	説明
userid	ID	ユーザーの ID。 Property behavior: - *required*
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き可能。 Property behavior: - *required*

dashboard.create

説明

`object dashboard.create(object/array dashboards)`

このメソッドを使用すると、新しいダッシュボードを作成できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するダッシュボード。

標準のダッシュボードプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
pages	array	ダッシュボード用に作成するダッシュボードページ。ダッシュボードページは、指定された順序と同じ順序で並べられます。 パラメータの動作: - 必須
users	array	ダッシュボード上に作成するダッシュボードユーザー共有。
userGroups	array	ダッシュボード上に作成するダッシュボードユーザーグループ共有。

戻り値

(object) dashboardids プロパティの下に作成されたダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたダッシュボードの順序と一致します。

例

ダッシュボードの作成

単一のダッシュボードページで、タグ付きの障害ウィジェットと 2 種類の共有 (ユーザーグループとユーザー) を持つ、”My dashboard” という名前のダッシュボードを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "service"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "zabbix_server"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- ダッシュボードページ
- ダッシュボードウィジェット
- ダッシュボードウィジェットフィールド
- ダッシュボードユーザー
- ダッシュボードユーザーグループ

ソース

CDashboard::create() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

説明

`object dashboard.delete(array dashboardids)`

このメソッドを使用すると、ダッシュボードを削除できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメーター

(array) 削除するダッシュボードの ID。

戻り値

(object) dashboardids プロパティの下にある削除されたダッシュボードの ID を含むオブジェクトを返します。

例

複数のダッシュボードの削除

2 つのダッシュボードを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2",
            "3"
        ]
    },
```

```
    "id": 1
}
```

ソース

CDashboard::delete() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

説明

`integer/array dashboard.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってダッシュボードを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 希望する出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dashboardids	ID/array	指定した ID を持つダッシュボードのみを返します。
selectPages	query	正しい順序で並べられたダッシュボードページを含む`pages` プロパティを返します。
selectUsers	query	ダッシュボードが共有されているユーザーを含む`users` プロパティを返します。
selectUserGroups	query	ダッシュボードが共有されているユーザーグループを含む`userGroups` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dashboardid`。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ID によるダッシュボードの取得

ダッシュボード"1" と"2" に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
```

```
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "1",
            "name": "Dashboard",
            "userid": "1",
            "private": "0",
            "display_period": "30",
            "auto_start": "1",
            "users": [],
            "userGroups": [],
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "9",
                            "type": "systeminfo",
                            "name": "",
                            "x": "12",
                            "y": "8",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "8",
                            "type": "problemsbysv",
                            "name": "",
                            "x": "12",
                            "y": "4",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "7",
                            "type": "problemhosts",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
```

```
                {
                    "widgetid": "6",
                    "type": "discovery",
                    "name": "",
                    "x": "6",
                    "y": "9",
                    "width": "18",
                    "height": "4",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "5",
                    "type": "web",
                    "name": "",
                    "x": "0",
                    "y": "9",
                    "width": "18",
                    "height": "4",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "4",
                    "type": "problems",
                    "name": "",
                    "x": "0",
                    "y": "3",
                    "width": "12",
                    "height": "6",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "3",
                    "type": "favmaps",
                    "name": "",
                    "x": "8",
                    "y": "0",
                    "width": "12",
                    "height": "3",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "1",
                    "type": "favgraphs",
                    "name": "",
                    "x": "0",
                    "y": "0",
                    "width": "12",
                    "height": "3",
                    "view_mode": "0",
                    "fields": []
                }
            ]
        },
        {
            "dashboard_pageid": "2",
            "name": "",
            "display_period": "0",
            "widgets": []
```

```
                },
                {
                    "dashboard_pageid": "3",
                    "name": "Custom page name",
                    "display_period": "60",
                    "widgets": []
                }
            ]
        },
        {
            "dashboardid": "2",
            "name": "My dashboard",
            "userid": "1",
            "private": "1",
            "display_period": "60",
            "auto_start": "1",
            "users": [
                {
                    "userid": "4",
                    "permission": "3"
                }
            ],
            "userGroups": [
                {
                    "usrgrpid": "7",
                    "permission": "2"
                }
            ],
            "pages": [
                {
                    "dashboard_pageid": "4",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "10",
                            "type": "problems",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "2",
                                    "name": "groupids",
                                    "value": "4"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- ダッシュボードページ
- ダッシュボードウィジェット

- ダッシュボードウィジェットフィールド
- ダッシュボードユーザー
- ダッシュボードユーザーグループ

ソース

CDashboard::get() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

説明

`object dashboard.update(object/array dashboards)`

このメソッドを使用すると、既存のダッシュボードを更新できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するダッシュボードのプロパティ。

各ダッシュボードについて dashboardid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

標準のダッシュボードプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
pages	array	既存のダッシュボードページを置き換えるダッシュボードページ。 ダッシュボードページは dashboard_pageid プロパティによって更新されます。dashboard_pageid プロパティを持たないオブジェクトについては新しいダッシュボードページが作成され、再利用されなかった既存のダッシュボードページは削除されます。ダッシュボードページは指定された順序どおりに並べられます。ダッシュボードページのうち、指定されたプロパティのみが更新されます。
users	array	既存の要素を置き換えるダッシュボードユーザーの共有設定。
userGroups	array	既存の要素を置き換えるダッシュボードユーザーグループの共有設定。

戻り値

(object) dashboardids プロパティの下にある更新されたダッシュボードの ID を含むオブジェクトを返します。

例

ダッシュボードの名前変更

ダッシュボードの名前を”SQL server status” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

ダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換えて、3 番目のページとして新しいページを追加します。他のすべてのダッシュボードページを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

ダッシュボードの所有者の変更

Admin と super admin のみが使用できます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
```

```
        "userid": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- ダッシュボードページ
- ダッシュボードウィジェット
- ダッシュボードウィジェットフィールド
- ダッシュボードユーザー
- ダッシュボードユーザーグループ

ソース

CDashboard::update() in *ui/include/classes/api/services/CDashboard.php*.

Dashboard widget fields

このページには、ダッシュボードウィジェットのパラメータと、対応するダッシュボードウィジェットフィールドオブジェクトの可能なプロパティ値へのナビゲーションリンクが含まれています。

各ウィジェットのパラメータとプロパティ値を確認するには、サイドバーを使用して該当するウィジェットのページに移動してください。

1 Action log

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、`dashboard.create` および `dashboard.update` メソッドでアクションログ ウィジェットを設定できます。

> **Attention:**
> ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。アクションログウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

Action log ウィジェットでは、次のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
受信者	11	userids.0	ユーザー ID。 注: 複数のユーザーを設定するには、プロパティ名の番号を増分したダッシュボードウィジェットフィールドオブジェクトをユーザーごとに作成します。
アクション	12	actionids.0	アクション ID。 注: 複数のアクションを設定するには、プロパティ名の番号を増分したダッシュボードウィジェットフィールドオブジェクトをアクションごとに作成します。
メディアタイプ	13	mediatypeids.0	メディアタイプ ID。 注: 複数のメディアタイプを設定するには、プロパティ名の番号を増分したダッシュボードウィジェットフィールドオブジェクトをメディアタイプごとに作成します。
ステータス	0	statuses.0	0 - 進行中; 1 - 送信済み/実行済み; 2 - 失敗。 注: 複数の値を設定するには、プロパティ名の番号を増分したダッシュボードウィジェットフィールドオブジェクトを値ごとに作成します。
検索文字列	1	message	任意の文字列値。
期間	1	time_period._reference	`DASHBOARD._timeperiod` - ダッシュボードの期間セレクターをデータソースとして設定します; `ABCDE._timeperiod` - 互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）をデータソースとして設定します。 デフォルト: `DASHBOARD._timeperiod` または、開始および終了パラメータで期間のみを設定できます。
開始	1	time_period.from	絶対時間形式（`YYYY-MM-DD hh:mm:ss`）または相対時間形式（`now`、`now/d`、`now/w-1w` など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポートされます - `time_period.to` が設定されている場合は必須です
終了	1	time_period.to	絶対時間形式（`YYYY-MM-DD hh:mm:ss`）または相対時間形式（`now`、`now/d`、`now/w-1w` など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポートされます - `time_period.from` が設定されている場合は必須です

パラメータ	type	name	value
エントリの並べ替え基準	0	sort_triggers	3 - 時刻（昇順）； 4 - *(デフォルト)* 時刻（降順）； 5 - タイプ（昇順）； 6 - タイプ（降順）； 7 - ステータス（昇順）； 8 - ステータス（降順）； 11 - 受信者（昇順）； 12 - 受信者（降順）。
表示行数	0	show_lines	指定可能な値の範囲は 1～100 です。 デフォルト: 25。

例

次の例は、アクションログウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アクションログウィジェットの設定

時間でソート (昇順) したアクション操作の詳細の 10 のエントリを表示するアクションログウィジェットを設定します。さらに、ユーザー"1" に電子メールを送信しようとしたが失敗したアクション操作のみの詳細を表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "actionlog",
                        "name": "Action log",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 10
                            },
                            {
                                "type": 0,
                                "name": "sort_triggers",
                                "value": 3
                            },
                            {
                                "type": 11,
```

```
                                "name": "userids.0",
                                "value": 1
                            },
                            {
                                "type": 13,
                                "name": "mediatypeids.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "statuses.0",
                                "value": 2
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

2 Clock

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*Clock*ウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。[*Clock*] ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

時計ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分
時間の形式	0	time_type	0 - *(デフォルト)* ローカル時間 1 - サーバー時間 2 - ホスト時間
時計のタイプ	0	clock_type	0 - *(デフォルト)* アナログ 1 - デジタル

時間の形式が” ホスト時間” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
アイテム	4	itemid.0	アイテムID。 パラメーターの動作: - 必須

時計のタイプが” デジタル” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
表示	0	show.0	1 - 日付 2 - *(デフォルト)* 時刻 3 - タイムゾーン 注意: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。

詳細設定

時計のタイプが” デジタル” に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
背景色	1	bg_color	16 進数の色コード (例: FF0000)。 デフォルト: "" (空)。

日付

時計のタイプが” デジタル” に設定され、表示が” 日付” に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	date_bold	0 - *(デフォルト)* 無効 1 - 有効

パラメーター	type	name	value
色	1	date_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。

時間

時計のタイプが" デジタル" に設定され、表示が" 時刻" に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	time_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	time_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。
秒	0	time_sec	0 - 無効 1 - *(デフォルト)* 有効
形式	0	time_format	0 - *(デフォルト)* 24 時間 1 - 12 時間

タイムゾーン

時計のタイプが" デジタル" に設定され、表示が" タイムゾーン" に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	tzone_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	tzone_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。
タイムゾーン	1	tzone_timezone	有効なタイムゾーン文字列 (例: Europe/Riga、system、UTC など)。サポートされているタイムゾーンの完全なリストについては、PHP ドキュメントを参照してください。 デフォルト: local。
フォーマット	0	tzone_format	パラメーターの動作: - 時間の形式が" ローカル時間" または" サーバー時間" に設定されている場合はサポート 0 - *(デフォルト)* 短縮形式 1 - 完全形式 パラメーターの動作: - 時間の形式が" ローカル時間" または" サーバー時間" に設定されている場合はサポート

例

次の例は、時計ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

時計ウィジェットの設定

カスタマイズされたデジタル時計にローカルの日付、時間、およびタイムゾーンを表示する時計ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
```

```
    "auto_start": 1,
    "pages": [
        {
            "widgets": [
                {
                    "type": "clock",
                    "name": "Clock",
                    "x": 0,
                    "y": 0,
                    "width": 12,
                    "height": 3,
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 0,
                            "name": "clock_type",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "show.0",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "show.1",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "show.2",
                            "value": 3
                        },
                        {
                            "type": 1,
                            "name": "date_color",
                            "value": "E1E1E1"
                        },
                        {
                            "type": 0,
                            "name": "time_bold",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "tzone_color",
                            "value": "E1E1E1"
                        },
                        {
                            "type": 1,
                            "name": "tzone_timezone",
                            "value": "Europe/Riga"
                        },
                        {
                            "type": 0,
                            "name": "tzone_format",
                            "value": 1
                        }
                    ]
                }
            ]
        }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

3 Discovery status

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで指定可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでディスカバリーステータス ウィジェットを設定できます。

パラメーター

ディスカバリのステータスウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

例

次の例は、ディスカバリのステータスウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ディスカバリのステータスウィジェットの設定

ディスカバリのステータスウィジェットを更新間隔を 15 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "discovery",
                        "name": "Discovery status",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 900
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

4 Favorite graphs

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドのお気に入りのグラフ ウィジェットを設定できます。

パラメーター

お気に入りのグラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分

例

次の例は、お気に入りのグラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

お気に入りのグラフウィジェットの設定

お気に入りのグラフウィジェットを更新間隔を 10 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favgraphs",
                        "name": "Favorite graphs",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
```

```
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

5 Favorite maps

説明

これらのパラメータと、対応するダッシュボードウィジェットフィールドオブジェクトで指定可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドのお気に入りのマップ ウィジェットを設定できます。

パラメーター

お気に入りのマップウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分

例

次の例は、お気に入りのマップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

お気に入りのマップウィジェットの設定

お気に入りのマップウィジェットを更新間隔を 10 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
```

```
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favmaps",
                        "name": "Favorite maps",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

6 Gauge

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、

dashboard.create および dashboard.update メソッドでゲージ ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。ゲージウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

ゲージウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
アイテム		4	itemid.0	アイテムの ID。 パラメーターの動作: - アイテム (ウィジェット) が設定されていない場合は必須
	アイテム (ウィジェット)	1	itemid._reference	アイテムの ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（参照パラメーターを"ABCDE" に設定）をアイテムのデータソースとして設定します。 パラメーターの動作: - アイテムが設定されていない場合は必須
最小値		1	min	任意の数値。サフィックス（例："1d"、"2w"、"4K"、"8G"）がサポートされています。 デフォルト: "0"。
最大値		1	max	任意の数値。サフィックス（例："1d"、"2w"、"4K"、"8G"）がサポートされています。 デフォルト: "100"。
値アークアーク背景		1	value_arc_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
		1	empty_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
背景		1	bg_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。

パラメーター	type	name	value
表示	0	show.0	1 - 説明; 2 - 値; 3 - 針; 4 - 目盛り; 5 - 値アーク。 注: 複数の値を設定するには、プロパティ名の番号を増やして各値ごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 デフォルト: 1, 2, 4, 5。 以下の両方の場合、「針」と「目盛り」の値はサポートされません: - 値「値アーク」の表示用のダッシュボードウィジェットフィールドオブジェクトが設定されていない; - アーク表示詳細設定パラメーターが「無効」に設定されている。 表示オプションの詳細設定パラメーターは、該当する値のダッシュボードウィジェットフィールドオブジェクトが設定されていない場合はサポートされません。
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメーターを"ABCDE"に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

高度な設定

Gauge ウィジェットでは、次の高度な設定パラメータがサポートされています。

Note:
Thresholds プロパティ名の番号（例: thresholds.0.color）は、昇順に並べたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix の Web インターフェースでウィジェット設定を更新した後、値は昇順に並べ替えられます（例: `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`、`"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`）。

パラメータ	type	name	value
Angle	0	angle	指定可能な値: 180 *(デフォルト)* または 270。
Description			
Description	1	description	マクロを含む任意の文字列値。 サポートされるマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ。 デフォルト: {ITEM.NAME}。
Size	0	desc_size	1～100 の値を指定できます。 デフォルト: 15。
Vertical position	0	desc_v_pos	0 - 上; 1 - *(デフォルト)* 下。
Bold	0	desc_bold	0 - *(デフォルト)* 無効; 1 - 有効。

パラメータ	type	name	value
Color	1	desc_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
Value			
Decimal places	0	decimal_places	1～10 の値を指定できます。 デフォルト: 2。
Size	0	value_size	1～100 の値を指定できます。 デフォルト: 25。
Bold	0	value_bold	0 - *(デフォルト)* 無効; 1 - 有効。
Color	1	value_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
Units			
Units (checkbox)	0	units_show	0 - 無効; 1 - *(デフォルト)* 有効。
Units (value)	1	units	任意の文字列値。 パラメータの動作: - *Units*（チェックボックス）が「有効」に設定されている場合にサポート
Size	0	units_size	1～100 の値を指定できます。 デフォルト: 25。 パラメータの動作: - *Units*（チェックボックス）が「有効」に設定されている場合にサポート
Bold	0	units_bold	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - *Units*（チェックボックス）が「有効」に設定されている場合にサポート
Position	0	units_pos	0 - 値の前; 1 - 値の上; 2 - *(デフォルト)* 値の後; 3 - 値の下。 パラメータの動作: - *Units*（チェックボックス）が「有効」に設定されている場合にサポート このパラメータに次のいずれかの時間関連の単位が設定されている場合、無視されます: unixtime、uptime、s。
Color	1	units_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。
Value arc			
Arc size	0	value_arc_size	1～100 の値を指定できます。 デフォルト: 20。
Needle			
Color	1	needle_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。 パラメータの動作: - *Show* に「Value arc」が設定されたダッシュボードウィジェットフィールドオブジェクトが設定されている場合、または *Show arc* が「有効」に設定されている場合にサポート
Scale			

パラメータ	type	name	value
Show units	0	scale_show_units	0 - 無効; 1 - *(デフォルト)* 有効。 パラメータの動作: - *Units*（チェックボックス）が「有効」に設定され、かつ *Show* に「Value arc」が設定されたダッシュボードウィジェットフィールドオブジェクトが設定されているか、*Show arc* が「有効」に設定されている場合にサポート
Size	0	scale_size	1～100 の値を指定できます。 デフォルト: 15。 パラメータの動作: - *Show* に「Value arc」が設定されたダッシュボードウィジェットフィールドオブジェクトが設定されている場合、または *Show arc* が「有効」に設定されている場合にサポート
Decimal places	0	scale_decimal_places	1～10 の値を指定できます。 デフォルト: 0。 パラメータの動作: - *Show* に「Value arc」が設定されたダッシュボードウィジェットフィールドオブジェクトが設定されている場合、または *Show arc* が「有効」に設定されている場合にサポート
Thresholds			
Color	1	thresholds.0.color	16 進数のカラーコード（例: FF0000）。
Threshold	1	thresholds.0.threshold	任意の数値。サフィックス（例: 「1d」、「2w」、「4K」、「8G」）がサポートされています。
Show labels	0	th_show_labels	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - *Thresholds* が設定され、かつ *Show* に「Value arc」が設定されたダッシュボードウィジェットフィールドオブジェクトが設定されているか、*Show arc* が「有効」に設定されている場合にサポート
Show arc	0	th_show_arc	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - *Thresholds* が設定されている場合にサポート
Arc size	0	th_arc_size	1～100 の値を指定できます。 デフォルト: 5。 パラメータの動作: - *Show arc* が「有効」に設定されている場合にサポート

例

次の例は、ゲージウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

ゲージウィジェットの設定

アイテム”44474” (インターフェース enp0s3: 送信ビット数) の値を表示するゲージウィジェットを設定します。さらに、しきい値などの複数の詳細オプションを使用して、ウィジェットを視覚的に微調整します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
```

```
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "gauge",
                        "name": "Gauge",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 44474
                            },
                            {
                                "type": 1,
                                "name": "min",
                                "value": "100000"
                            },
                            {
                                "type": 1,
                                "name": "max",
                                "value": "1000000"
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "show.4",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "show.5",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "angle",
                                "value": 270
                            },
                            {
                                "type": 0,
```

```
            "name": "desc_size",
            "value": 10
        },
        {
            "type": 0,
            "name": "desc_bold",
            "value": 1
        },
        {
            "type": 0,
            "name": "decimal_places",
            "value": 0
        },
        {
            "type": 0,
            "name": "value_bold",
            "value": 1
        },
        {
            "type": 0,
            "name": "units_size",
            "value": 15
        },
        {
            "type": 0,
            "name": "units_pos",
            "value": 3
        },
        {
            "type": 1,
            "name": "needle_color",
            "value": "3C3C3C"
        },
        {
            "type": 1,
            "name": "thresholds.0.color",
            "value": "FF465C"
        },
        {
            "type": 1,
            "name": "thresholds.0.threshold",
            "value": "700000"
        },
        {
            "type": 1,
            "name": "thresholds.1.color",
            "value": "FFD54F"
        },
        {
            "type": 1,
            "name": "thresholds.1.threshold",
            "value": "500000"
        },
        {
            "type": 1,
            "name": "thresholds.2.color",
            "value": "0EC9AC"
        },
        {
            "type": 1,
            "name": "thresholds.2.threshold",
            "value": "100000"
```

```
                            },
                            {
                                "type": 0,
                                "name": "th_show_labels",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "th_show_arc",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "th_arc_size",
                                "value": 15
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

7 Geomap

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで指定可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*Geomap* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*Geomap* ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

Geomap ウィジェットでは、次のパラメータがサポートされています。

パラメータ		タイプ	名前	値
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成し、プロパティ名の番号を 1 つずつ増やします。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID の代わりに指定します: `ABCDE._hostgroupids` - 互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）をホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成し、プロパティ名の番号を 1 つずつ増やします。複数のホストを設定する場合、ホストグループパラメータはまったく設定しないか、設定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホスト ID の代わりに指定します: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - 互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）をホストのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		タイプ	名前	値
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - タグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - タグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグの値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - タグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
初期表示クラスタリング		1	default_view	カンマ区切りの緯度、経度、ズームレベル（任意、指定可能な値の範囲は 0～30）。 例: 40.6892494,-74.0466891,10。
	クラスタリングモード	0	clustering_mode	0 - *(デフォルト)* 自動; 1 - ズームレベル。
	クラスタリングズームレベル	0	clustering_zoom_level	指定可能な値の範囲は 0～30 です。 デフォルト: 0。 パラメータの動作: - クラスタリングモードが「ズームレベル」に設定されている場合にサポート

パラメータ	タイプ	名前	値
Reference	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、地理マップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

地理マップウィジェットの設定

以下のタグ設定に基づいて、ホストグループ”2” と”22” のホストを表示する地理マップウィジェットを設定します。”component” という名前のタグに”node” という値が含まれる、または”location” という名前のタグに”New York” という値と等しいもの。さらに、マップの初期表示を座標”40.6892494”(緯度)、”-74.0466891”(経度)、ズームレベル”10” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "geomap",
                        "name": "Geomap",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 22
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "default_view",
                                "value": "40.6892494,-74.0466891,10"
                            },
                            {
                                "type": 0,
                                "name": "evaltype",
                                "value": 2
                            },
```

```
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "node"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "location"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "New York"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

8 Graph

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでグラフ ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。グラフウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

グラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分
参照	1	reference	5 文字で構成される文字列値 (例: ABCDE または JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

データセット

データセットの設定では、次のパラメータがサポートされています。

Note:
プロパティ名の最初の番号（例: ds.0.hosts.0、ds.0.items.0）は特定のデータセットを表し、2 番目の番号がある場合は、設定されたホストまたはアイテムを表します。

パラメータ	タイプ	名前	値
データセットタイプ	0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。

パラメータ	タイプ	名前	値
アイテム	4	ds.0.itemids.0	アイテム ID。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されたアイテムのみを指定してください。 注: 複数のアイテムを設定するには、プロパティ名の番号を増分したダッシュボードウィジェットフィールドオブジェクトをアイテムごとに作成します。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテム（ウィジェット）が設定されていない場合は必須
アイテム（ウィジェット）	1	ds.0.itemids.0._reference	アイテム ID の代わりに指定します: `ABCDE._itemid` - アイテムのデータソースとして、互換性のあるウィジェット（参照パラメータを「ABCDE」に設定）を指定します。 注: 複数のウィジェットを設定するには、プロパティ名の番号を増分したダッシュボードウィジェットフィールドオブジェクトをウィジェットごとに作成します。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテムが設定されていない場合は必須
色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合は必須
ホストパターン	1	ds.0.hosts.0	ホスト名またはパターン（例: ”Zabbix*”）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
アイテムパターン	1	ds.0.items.0	アイテム名またはパターン（例: ”*: 1 秒あたりの処理済み * 値の数”）。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されたアイテムのパターンのみを指定してください。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合にサポート
カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 指定可能な値は 0～11 です。 デフォルト: 0。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合にサポート

パラメータ	タイプ	名前	値
描画	0	ds.0.type	0 - *(デフォルト)* 線; 1 - 点; 2 - 階段状; 3 - 棒。
積み上げ	0	ds.0.stacked	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 描画が「線」、「階段状」、または「棒」に設定されている場合にサポート
幅	0	ds.0.width	指定可能な値は 1～10 です。 デフォルト: 1。 パラメータの動作: - 描画が「線」または「階段状」に設定されている場合にサポート
点のサイズ	0	ds.0.pointsize	指定可能な値は 1～10 です。 デフォルト: 3。 パラメータの動作: - 描画が「点」に設定されている場合にサポート
透明度	0	ds.0.transparency	指定可能な値は 1～10 です。 デフォルト: 5。
塗りつぶし	0	ds.0.fill	指定可能な値は 1～10 です。 デフォルト: 3。 パラメータの動作: - 描画が「線」または「階段状」に設定されている場合にサポート
欠損データ	0	ds.0.missingdatafunc	0 - *(デフォルト)* なし; 1 - 接続; 2 - 0 として扱う; 3 - 最後の既知の値。 パラメータの動作: - 描画が「線」または「階段状」に設定されている場合にサポート
ホストの上書き	1	ds.0.override_hostid.reference	ABCDE - ホストのデータソースとして、互換性のあるウィジェット（「参照」パラメータを「ABCDE」に設定）を指定します。 DASHBOARD._hostid - ホストのデータソースとして、ダッシュボードのホストセレクターを指定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
Y 軸値を反転	0	ds.0.axisy	0 - *(デフォルト)* 左; 1 - 右。
時間シフト	0	ds.0.invert_values	0 - (デフォルト) 無効; 1 - 有効。
	1	ds.0.timeshift	有効な時間文字列（例: 3600、1h など）。 時間サフィックスを使用できます。負の値も使用できます。 デフォルト: ""（空）。

パラメータ		タイプ	名前	値
集計関数		0	ds.0.aggregate_function	0 - *(デフォルト)* 使用しない; 1 - 最小; 2 - 最大; 3 - 平均; 4 - 件数; 5 - 合計; 6 - 最初; 7 - 最後。
集計間隔		1	ds.0.aggregate_interval	有効な時間文字列（例: 3600、1h など）。 時間サフィックスを使用できます。 デフォルト: 1h。
集計		0	ds.0.aggregate_grouping	0 - *(デフォルト)* 各アイテム; 1 - データセット。 パラメータの動作: - 集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、または「最後」に設定されている場合にサポート
近似		0	ds.0.approximation	1 - 最小; 2 - *(デフォルト)* 平均; 4 - 最大; 7 - すべて。
データセットラベル		1	ds.0.data_set_label	任意の文字列値。 デフォルト: ""（空）。
アイテムタグ	評価タイプ	0	ds.0.item_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	ds.0.item_tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須
	演算子	0	ds.0.item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須

パラメータ		タイプ	名前	値
	タグ値	1	ds.0.item_tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須

表示オプション

表示オプションの設定では、次のパラメータがサポートされています。

パラメータ	タイプ	名前	値
履歴データの選択	0	source	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。
シンプルなトリガー	0	simple_triggers	0 - *(デフォルト)* 無効; 1 - 有効。
稼働時間	0	working_time	0 - *(デフォルト)* 無効; 1 - 有効。
ラベル内のホスト名	0	show_hostnames	0 - *(デフォルト)* 自動; 1 - 表示; 2 - 非表示。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		タイプ	名前	値
パーセンタイルライン(左)				
	ステータス	0	percentile_left	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - データセットの設定で *Y* 軸が「左」に設定されている場合にサポート
	値	0	percentile_left_value	指定可能な値の範囲は 1～100 です。 パラメータの動作: - データセットの設定で *Y* 軸が「左」に設定されている場合にサポート
パーセンタイルライン(右)				
	ステータス	0	percentile_right	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - データセットの設定で *Y* 軸が「右」に設定されている場合にサポート
	値	0	percentile_right_value	指定可能な値の範囲は 1～100 です。 パラメータの動作: - データセットの設定で *Y* 軸が「右」に設定されている場合にサポート

期間

期間の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	値
期間	1	time_period._referenceDASHBOARD._timeperiod	- ダッシュボードの期間セレクタをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータが"ABCDE"に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.to が設定されている場合は必須
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.from が設定されている場合は必須

Axes

軸の設定には、以下のパラメーターがサポートされています。

パラメーター	type	name	value
左 *Y* 軸	0	lefty	0 - 無効 1 - *(デフォルト)* 有効 パラメーターの動作: - *Y* 軸 (データセット 設定内) が" 左" に設定されている場合に サポートされます。
右 *Y*	0	righty	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - *Y* 軸 (データセット 設定内) が" 右" に設定されている場合に サポートされます。
スケール	0	lefty_scale righty_scale	0 - *(デフォルト)* 線形; 1 - 対数
最小値	1	lefty_min righty_min	任意の数値 デフォルト: "" (空)
最大値	1	lefty_max righty_max	任意の数値 デフォルト: "" (空)
単位 (タイプ)	0	lefty_units righty_units	0 - *(デフォルト)* 自動; 1 - 静的
単位 (値)	1	lefty_static_units righty_static_units	任意の文字列値。 デフォルト: "" (空)

パラメーター	type	name	value
X 軸	0	xaxis	0 - 無効; 1 - *(デフォルト)* 有効

凡例

凡例の設定では、次のパラメータがサポートされています。

パラメータ	type	name	value
凡例を表示	0	legend	0 - 無効; 1 - *(デフォルト)* 有効。
最小値/平均値/最大値を表示	0	legend_statistic	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
集計関数を表示	0	legend_aggregation	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
行	0	legend_lines_mode	0 - *(デフォルト)* 固定; 1 - 可変。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
行数/最大行数	0	legend_lines	指定可能な値の範囲は 1～10 です。 デフォルト: 1。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
列数	0	legend_columns	指定可能な値の範囲は 1～4 です。 デフォルト: 4。 パラメータの動作: - 凡例を表示が「有効」に設定され、最小値/平均値/最大値を表示が「無効」に設定されている場合にサポート

Problems

問題の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
問題を表示	0	show_problems	0 - *(デフォルト)* 無効; 1 - *(デフォルト)* 有効

パラメータ		type	name	value
選択された項目のみ		0	graph_item_problems	0 - (デフォルト) 有効
問題のあるホスト		1	problemhosts.0	ホスト 名 注: プロパティ名の数字は、設定されたホストを参照します。 複数のホストを設定するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを設定する場合、サポートされません。
重大度		0	severities.0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効) 注: 複数の値を設定するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトを値ごとに作成します。
問題		1	problem_name	問題イベント名 (大文字と小文字は区別されません。完全な名前またはその一部)
問題タグ				
	評価タイプ	0	evaltype	0 - (デフォルト) And/Or; 2 - Or
	タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 問題タグを構成する場合は 必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 問題のあるタグを構成する場合は必須

パラメータ	type	name	value
タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 問題のあるタグを構成する場合は必須

オーバーライド

オーバーライドの設定では、以下のパラメータがサポートされています。

Note:
プロパティ名の最初の番号（例: or.0.hosts.0、or.0.items.0）は特定のデータセットを表し、2 番目の番号がある場合は、設定されたホストまたはアイテムを表します。

パラメータ	type	name	value
ホストパターン	1	or.0.hosts.0	ホスト名またはパターン（例: Zabbix*）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。 パラメータの動作: - オーバーライドを設定する場合は必須
アイテムパターン	1	or.0.items.0	アイテム名またはパターン（例: *: Number of processed *values per second）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのパターンのみを設定してください。 パラメータの動作: - オーバーライドを設定する場合は必須
基本色	1	or.0.color	16 進数のカラーコード（例: FF0000）。
幅	0	or.0.width	1 から 10 までの値を指定できます。
描画	0	or.0.type	0 - 線; 1 - ポイント; 2 - 階段状; 3 - 棒。
透明度	0	or.0.transparency	1 から 10 までの値を指定できます。
塗りつぶし	0	or.0.fill	1 から 10 までの値を指定できます。
ポイントサイズ	0	or.0.pointsize	1 から 10 までの値を指定できます。
欠損データ	0	or.0.missingdatafunc	0 - なし; 1 - 接続; 2 - 0 として扱う; 3 - 最後に確認された値。
Y 軸	0	or.0.axisy	0 - 左; 1 - 右。
値を反転	0	or.0.invert_values	0 - （デフォルト）無効; 1 - 有効。
時間シフト	1	or.0.timeshift	有効な時間文字列（例: 3600、1h など）。 時間のサフィックスを使用できます。負の値も使用できます。

例

次の例は、グラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフウィジェットの設定

グラフウィジェットは、以下のように設定します。

- 1 つのホストに合計 9 つのアイテムを持つ 2 つのデータセット。
- 最初のデータセットは「アイテムリスト」タイプで、3 つのアイテムで構成され、それぞれ異なる色の線で表されますが、幅、透明度、塗りつぶしは同じです。
- 2 番目のデータセットは「アイテムパターン」タイプで、6 つのアイテムで構成され、集計が設定されており、カスタムの色、幅、透明度、塗りつぶしの線で表されます。
- 2 番目のデータセットにはカスタムのデータセットラベルもあります。
- グラフのデータは直近 3 時間の期間で表示されます。
- グラフの問題は、設定されたアイテムに対してのみ表示されます。
- グラフには 2 つの Y 軸があり、右側の Y 軸は 2 番目のデータセットの値のみを表示します。
- グラフの凡例には、4 行で設定されたアイテムと、データセットの最小値、最大値、平均値が表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23269
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "BF00FF"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23257
                            },
```

```
                    {
                        "type": 1,
                        "name": "ds.0.color.2",
                        "value": "0040FF"
                    },
                    {
                        "type": 0,
                        "name": "ds.0.width",
                        "value": 3
                    },
                    {
                        "type": 0,
                        "name": "ds.0.transparency",
                        "value": 3
                    },
                    {
                        "type": 0,
                        "name": "ds.0.fill",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "ds.1.hosts.0",
                        "value": "Zabbix server"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.items.0",
                        "value": "*: Number of processed *values per second"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.color",
                        "value": "000000"
                    },
                    {
                        "type": 0,
                        "name": "ds.1.transparency",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "ds.1.fill",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "ds.1.axisy",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "ds.1.aggregate_function",
                        "value": 3
                    },
                    {
                        "type": 1,
                        "name": "ds.1.aggregate_interval",
                        "value": "1m"
                    },
                    {
                        "type": 0,
```

```
                            "name": "ds.1.aggregate_grouping",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "ds.1.data_set_label",
                            "value": "Number of processed values per second"
                        },
                        {
                            "type": 0,
                            "name": "graph_time",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "time_period.from",
                            "value": "now-3h"
                        },
                        {
                            "type": 1,
                            "name": "time_period.to",
                            "value": "now"
                        },
                        {
                            "type": 0,
                            "name": "legend_statistic",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "legend_lines",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "show_problems",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "reference",
                            "value": "YZABC"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
        }
    ]
},
"id": 1
```

```
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

9 Graph (classic)

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでグラフ（クラシック） ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。グラフ（クラシック）ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

グラフ（クラシック）ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	値
リフレッシュ間隔	0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ソース	0	source_type	0 - *(デフォルト)* グラフ; 1 - シンプルグラフ。
グラフ	6	graphid.0	グラフの ID。 パラメーターの動作: - 必須（ソースが「グラフ」に設定されている場合）

パラメーター		type	name	値
	グラフ（ウィジェット）	1	graphid._reference	グラフの ID の代わりに： ABCDE._graphid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をグラフのデータソースとして設定します。 パラメーターの動作: - 必須（ソースが「シンプルグラフ」に設定され、グラフが設定されていない場合）
アイテム		4	itemid.0	アイテムの ID。 パラメーターの動作: - 必須（ソースが「シンプルグラフ」に設定され、アイテム（ウィジェット）が設定されていない場合）
	アイテム（ウィジェット）	1	itemid._reference	アイテムの ID の代わりに： ABCDE._itemid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をアイテムのデータソースとして設定します。 パラメーターの動作: - 必須（ソースが「シンプルグラフ」に設定され、アイテムが設定されていない場合）
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をデータソースとして設定します。 デフォルト：DASHBOARD._timeperiod または、*From* および *To* パラメーターで期間のみを設定することもできます。
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメーターの動作: - サポート（期間が設定されていない場合） - time_period.to が設定されている場合は必須
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメーターの動作: - サポート（期間が設定されていない場合） - time_period.from が設定されている場合は必須
凡例を表示		0	show_legend	0 - 無効; 1 - *(デフォルト)* 有効。
ホストを上書き		1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメーターが"ABCDE" に設定されているもの）をホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター	type	name	値
参照	1	reference	任意の 5 文字の文字列（例：ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

例

次の例は、グラフ（クラシック）ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフ（クラシック）ウィジェットの設定

アイテム"42269" のシンプルなグラフを表示するグラフ（クラシック）ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "name": "Graph (classic)",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 1
                            },
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42269
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "RSTUV"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
```

```
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

10 Graph prototype

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでグラフのプロトタイプ ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。グラフのプロトタイプウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

グラフのプロトタイプウィジェットでは、次のパラメータがサポートされています。

パラメータ	タイプ	名前	値
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。

パラメータ		タイプ	名前	値
ソースグラフのプロトタイプアイテムのプロトタイプ期間		0	source_type	2 - *(デフォルト)* グラフのプロトタイプ; 3 - シンプルグラフのプロトタイプ。
		7	graphid.0	グラフのプロトタイプ ID。 パラメータの動作: - ソースが「グラフのプロトタイプ」に設定されている場合は必須
		5	itemid.0	アイテムのプロトタイプ ID。 パラメータの動作: - ソースが「シンプルグラフのプロトタイプ」に設定されている場合は必須
		1	time_period._referenceDASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータに「ABCDE」が設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始日時および終了日時パラメータでのみ期間を設定できます。
	開始日時	1	time_period.from	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間形式（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポートされます - time_period.to が設定されている場合は必須
	終了日時	1	time_period.to	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間形式（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポートされます - time_period.from が設定されている場合は必須
凡例を表示		0	show_legend	0 - 無効; 1 - *(デフォルト)* 有効。

パラメータ	タイプ	名前	値
ホストを上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメータに「ABCDE」が設定されているもの）をホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
列	0	columns	指定可能な値は 1 から 24 です。 デフォルト: 2。
行	0	rows	指定可能な値は 1 から 16 です。 デフォルト: 1。
参照	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、グラフのプロトタイプウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフのプロトタイプウィジェットの設定

ローレベルディスカバリによってアイテムのプロトタイプ (ID: ”42316”) から作成された 3 つのグラフ (3 列、1 行) のグリッドを表示するグラフのプロトタイプウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graphprototype",
                        "name": "Graph prototype",
                        "x": 0,
                        "y": 0,
                        "width": 48,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 3
                            },
                            {
                                "type": 5,
```

```
                                "name": "itemid.0",
                                "value": 42316
                            },
                            {
                                "type": 0,
                                "name": "columns",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "OPQWX"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

11 Honeycomb

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*Honeycomb* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*Honeycomb* ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメーター

Honeycomb ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
リフレッシュ間隔		0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、プロパティ名の番号を増やして各ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストタグ				

パラメーター		type	name	value
	評価タイプ	0	evaltype_host	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（ホストタグを設定する場合） このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（ホストタグを設定する場合） このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（ホストタグを設定する場合） このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
アイテムパターンアイテムタグ		1	items.0	アイテム名またはパターン。 注: 複数のアイテムパターンを設定するには、プロパティ名の番号を増やして各アイテムパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメーターの動作: - 必須
	評価タイプ	0	evaltype_item	0 - *(デフォルト)* And/Or; 2 - Or。

パラメーター		type	name	value
	タグ名	1	item_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（アイテムタグを設定する場合）
	演算子	0	item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（アイテムタグを設定する場合）
	タグ値	1	item_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（アイテムタグを設定する場合）
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。
表示		0	show.0	1 - プライマリラベル; 2 - セカンダリラベル。 注: 複数の値を設定するには、プロパティ名の番号を増やして各値ごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 デフォルト: 1, 2。
Reference		1	reference	5 文字からなる任意の文字列値（例: `ABCDE` や `JBPNL`）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

高度な設定

以下の高度な設定パラメータは、*Honeycomb* ウィジェットでサポートされています。

Note:
しきい値プロパティ名の番号（例: thresholds.0.color）は、昇順に並べ替えられたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix の Web インターフェースでウィジェット設定を更新した後、値は昇順に並べ替えられます（例: `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`、`"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`）。

パラメータ	type	name	value
プライマリラベルタイプ	0	primary_label_type	0 - *(デフォルト)* テキスト; 1 - 値。
テキスト	1	primary_label	マクロを含む任意の文字列値。 サポートされるマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ。 デフォルト: {HOST.NAME} パラメータの動作: - タイプが「テキスト」に設定されている場合にサポート
小数点以下の桁数	0	primary_label_decimal_places	0 から 6 までの値を指定できます。 デフォルト: 2。 パラメータの動作: - タイプが「値」に設定されている場合にサポート
サイズ（タイプ）	0	primary_label_size_type	0 - *(デフォルト)* 自動; 1 - カスタム。
サイズ	0	primary_label_size	1 から 100 までの値を指定できます。 デフォルト: 20。 パラメータの動作: - サイズ（タイプ）が「カスタム」に設定されている場合にサポート
太字	0	primary_label_bold	0 - *(デフォルト)* 無効; 1 - 有効。
色	1	primary_label_color	16 進数のカラーコード（例: FF0000）。 デフォルト: Settings オブジェクトおよびUser オブジェクトの theme に基づく: 「blue-theme」、「blue-classic-theme」、または「hc-light」の場合は 1F2C33; 「dark-theme」、「dark-classic-theme」、または「hc-dark」の場合は EEEEEE; 「dark-blue-theme」の場合は E5F1FF。
単位（チェックボックス）	0	primary_label_units_show	0 - 無効; 1 - *(デフォルト)* 有効。 パラメータの動作: - タイプが「値」に設定されている場合にサポート
単位（値）	1	primary_label_units	任意の文字列値。 ""（空） パラメータの動作: - タイプが「値」に設定され、単位（チェックボックス）が「有効」に設定されている場合にサポート

パラメータ	type	name	value
位置	0	primary_label_units_pos	0 - 値の前; 1 - *(デフォルト)* 値の後。 パラメータの動作: - タイプが「値」に設定され、単位（チェックボックス）が「有効」に設定されている場合にサポート このパラメータに次の時間関連の単位のいずれかが設定されている場合、無視されます: unixtime、uptime、s。
セカンダリラベルタイプ	0	secondary_label_type	0 - テキスト; 1 - *(デフォルト)* 値。
テキスト	1	secondary_label	マクロを含む任意の文字列値。 サポートされるマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ。 デフォルト: {{ITEM.LASTVALUE}.fmtnum(2)} パラメータの動作: - タイプが「テキスト」に設定されている場合にサポート
小数点以下の桁数	0	secondary_label_decimal_places	0から10までの値を指定できます。 デフォルト: 2。 パラメータの動作: - タイプが「値」に設定されている場合にサポート
サイズ（タイプ）	0	secondary_label_size_type	0 - *(デフォルト)* 自動; 1 - カスタム。
サイズ	0	secondary_label_size	1から100までの値を指定できます。 デフォルト: 30。 パラメータの動作: - サイズ（タイプ）が「カスタム」に設定されている場合にサポート
太字	0	secondary_label_bold	0 - 無効; 1 - *(デフォルト)* 有効。
色	1	secondary_label_color	16進数のカラーコード（例: FF0000）。 デフォルト: Settingsオブジェクトおよび User オブジェクトの theme に基づく: 「blue-theme」、「blue-classic-theme」、または「hc-light」の場合は 1F2C33; 「dark-theme」、「dark-classic-theme」、または「hc-dark」の場合は EEEEEE; 「dark-blue-theme」の場合は E5F1FF。
単位（チェックボックス）	0	secondary_label_units_show	0 - 無効; 1 - *(デフォルト)* 有効。 パラメータの動作: - タイプが「値」に設定されている場合にサポート
単位（値）	1	secondary_label_units	任意の文字列値。 ""（空） パラメータの動作: - タイプが「値」に設定され、単位（チェックボックス）が「有効」に設定されている場合にサポート

パラメータ	type	name	value
位置	0	secondary_label_position	0 - 値の前; 1 - *(デフォルト)* 値の後。 パラメータの動作: - タイプが「値」に設定され、単位（チェックボックス）が「有効」に設定されている場合にサポート このパラメータに次の時間関連の単位のいずれかが設定されている場合、無視されます: unixtime、uptime、s。
背景色			
背景色	1	bg_color	16 進数のカラーコード（例: FF0000）。 デフォルト: Settings オブジェクトおよびUser オブジェクトの `theme` に基づく: 「blue-theme」および「blue-classic-theme」の場合は `D9E7ED`; 「dark-theme」および「dark-classic-theme」の場合は `3D5059`; 「dark-blue-theme」の場合は `004FA8`; 「hc-light」の場合は `AAD7E9`; 「hc-dark」の場合は `335463`。
しきい値			
色の補間	0	interpolation	0 - 無効; 1 - *(デフォルト)* 有効。
色	1	thresholds.0.color	16 進数のカラーコード（例: FF0000）。
しきい値	1	thresholds.0.threshold	任意の数値。サフィックス（例: 「1d」、「2w」、「4K」、「8G」）がサポートされています。

例

次の例は、ハニカムウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

ハニカムウィジェットの設定

Zabbix サーバープロセスの使用率を表示するハニカムウィジェットを設定します。さらに、ハニカムセルのプライマリラベルを変更し、しきい値を設定してウィジェットを視覚的に微調整します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "honeycomb",
                        "name": "Honeycomb",
                        "x": "0",
                        "y": "0",
                        "width": "24",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
```

```
                            "type": 3,
                            "name": "hostids.0",
                            "value": 10084
                        },
                        {
                            "type": 1,
                            "name": "items.0",
                            "value": "Zabbix server: Utilization*"
                        },
                        {
                            "type": 1,
                            "name": "primary_label",
                            "value": "{ITEM.NAME}"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.color",
                            "value": "0EC9AC"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.0.threshold",
                            "value": "0"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.color",
                            "value": "FFD54F"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.1.threshold",
                            "value": "70"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.color",
                            "value": "FF465C"
                        },
                        {
                            "type": 1,
                            "name": "thresholds.2.threshold",
                            "value": "90"
                        },
                        {
                            "type": 1,
                            "name": "reference",
                            "value": "KSTMQ"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
```

```
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

12 Host availability

説明

これらのパラメータと、対応するダッシュボードウィジェットフィールドオブジェクトで指定可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドのホストの可用性 ウィジェットを設定できます。

Parameters

ホスト可用性ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - (デフォルト) 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメーター		type	name	value
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: `ABCDE._hostgroupids` の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 テンプレートダッシュボード でウィジェットを構成する場合、このパラメータはサポートされません。
インターフェースタイプ		0	interface_type.0	0 - なし; 1 - Zabbix エージェント (パッシブチェック); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix エージェント (アクティブチェック) デフォルト: 1、2、3、4、5 (すべて有効) 注: 複数の値を設定するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。
レイアウト		0	layout	0 - *(デフォルト)* 水平; 1 - 垂直
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効
合計のみを表示		0	only_totals	0 - *(デフォルト)* 無効; 1 - 有効

例

次の例は、ホスト稼働状況ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

ホスト稼働状況ウィジェットの設定

"Zabbix エージェント" および"SNMP" インターフェースが設定されているホストグループ"4" のホストの可用性情報を (縦型レイアウトで) 表示するホスト稼働状況ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostavail",
                        "name": "Host availability",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

13 Host card

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでホストカード ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。ホストカードウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

ホストカードウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホスト	3	hostid.0	ホストの ID。 パラメータの動作: - 必須（ホスト（ウィジェット/ダッシュボード）が設定されていない場合） このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
	ホスト（ウィジェット/ダッシュボード）	1	hostid._reference	ホストの ID の代わりに： DASHBOARD.hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定; ABCDE._hostid - 互換性のあるウィジェット（参照パラメータが”ABCDE” に設定されているもの）をホストのデータソースとして設定。 パラメータの動作: - 必須（ホストが設定されていない場合） このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
抑制された障害を表示		0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
表示		0	sections.0	0 - ホストグループ; 1 - 説明; 2 - 監視; 3 - 可用性; 4 - 監視サーバー; 5 - テンプレート; 6 - インベントリ; 7 - タグ。 注：プロパティ名の数字はセクションリスト内のセクション順序を参照します。複数のセクションを設定するには、各セクションごとにプロパティ名の数字を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。

表示が「インベントリ」に設定されている場合、以下のパラメータがサポートされます。

パラメータ	type	name	value
インベントリフィールド	0	inventory.0	インベントリの ID。 注：複数のインベントリフィールドを設定するには、各インベントリフィールドごとにプロパティ名の数字を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。

例

次の例は、*Host card* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

Configuring a *Host card* widget

”監視”、”可用性”、”監視元”、”インベントリ”、”タグ” の各セクションを表示する ホストカードウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostcard",
                        "name": "Host card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostid.0",
                                "value": 10084
                            },
                            {
                                "type": 0,
                                "name": "show_suppressed",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 6
                            },
                            {
                                "type": 0,
                                "name": "sections.4",
                                "value": 7
                            },
                            {
                                "type": 0,
                                "name": "inventory.0",
                                "value": 25
                            },
                            {
                                "type": 0,
                                "name": "inventory.1",
                                "value": 26
                            }
```

```
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

14 Host navigator

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでホストナビゲーター ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。ホストナビゲーターウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

以下のパラメータは、ホストナビゲーターウィジェットでサポートされています。

パラメータ		タイプ	名前	値
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成し、プロパティ名の番号を 1 つずつ増やします。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホストグループ (ウィジェット)	1	groupids._reference	ホストグループ ID の代わりに指定します: ABCDE._hostgroupids - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットを、ホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストパターン		1	hosts.0	ホスト名またはパターン。 注: 複数のホストパターンを設定するには、ホストパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成し、プロパティ名の番号を 1 つずつ増やします。複数のホストパターンを使用する場合、ホストグループパラメータは未設定にするか、設定したホストパターンに一致するホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストのステータス		0	status	-1 - *(デフォルト)* すべて; 0 - 有効; 1 - 無効。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストタグ	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		タイプ	名前	値
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグの値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
深刻度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、値ごとにダッシュボードウィジェットフィールドオブジェクトを作成し、プロパティ名の番号を 1 つずつ増やします。

パラメータ		タイプ	名前	値
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。
障害を表示		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制されていない障害; 2 - なし。
グループ化	属性	0	group_by.0.attribute	0 - ホストグループ; 1 - タグの値; 2 - 深刻度。 注: プロパティ名の番号は、グループ化属性リスト内の属性の順序を示します。 パラメータの動作: - グループ化を設定する場合は必須
	値	1	group_by.0.tag_name	任意の文字列値。 注: プロパティ名の番号は、属性パラメータで設定されたグループ化属性を示します。 パラメータの動作: - グループ化を設定し、属性を「タグの値」に設定する場合は必須
ホスト数の上限		0	show_lines	指定可能な値の範囲は 1～9999 です。 デフォルト: 100。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
Reference		1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、ホストナビゲーターウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目

的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ホストナビゲーターウィジェットの設定

ホストグループ別にグループ化されたホストと、"city" タグの値別にグループ化されたホストを表示するホストナビゲーターウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostnavigator",
                        "name": "Host navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "city"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "SWKLB"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
```

```
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

15 アイテムカード

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでアイテムカード ウィジェットを設定できます。

Attention:
ダッシュボードの作成時または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。アイテムカードウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

アイテムカードウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。

パラメーター		type	name	value
アイテム		4	itemid.0	アイテムの ID。 パラメーターの動作: - *Item (Widget)* が設定されていない場合は必須
	Item (Widget)	1	itemid._reference	アイテムの ID の代わりに: `ABCDE._itemid` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をアイテムのデータソースとして設定します。
表示		0	sections.0	パラメーターの動作: - *Item* が設定されていない場合は必須 0 - 説明; 1 - エラーテキスト; 2 - メトリクス; 3 - 最新データ; 4 - 情報の種類; 5 - トリガー; 6 - ホストインターフェース; 7 - タイプ; 8 - ホストインベントリ; 9 - タグ。 注: プロパティ名の数字はセクションリスト内のセクション順序を参照します。複数のセクションを設定するには、プロパティ名の数字を増やして各セクションごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
ホストの上書き		1	override_hostid._reference	`ABCDE._hostid` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します; `DASHBOARD._hostid` - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 デフォルト: "" (空) このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

スパークライン

表示が「最新データ」に設定されている場合、次のパラメータがサポートされます。

パラメータ	タイプ	名前	値
幅	0	sparkline.width	指定可能な値は 0～10 です。 デフォルト: 1。
塗りつぶし色	0	sparkline.fill	指定可能な値は 0～10 です。 デフォルト: 3。
	1	sparkline.color	16 進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。

パラメータ		タイプ	名前	値
期間		1	sparkline.time_period.from	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - reference パラメータが ABCDE に等しい互換性のあるウィジェットをデータソースとして設定します。 デフォルト: ""（空） また、開始および終了パラメータでのみ期間を設定することもできます。
	開始	1	sparkline.time_period.from	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間形式（now、now/d、now/w-1w など）の有効な時間文字列。 デフォルト: now-1h。 パラメータの動作: - 期間が設定されていない場合にサポートされます - sparkline.time_period.to が設定されている場合に必須です
	終了	1	sparkline.time_period.to	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間形式（now、now/d、now/w-1w など）の有効な時間文字列。 デフォルト: now。 パラメータの動作: - 期間が設定されていない場合にサポートされます - time_period.to が設定されている場合に必須です
履歴データ		0	sparkline.history	0 - （デフォルト）自動; 1 - 履歴; 2 - トレンド。

例

以下の例は、アイテムカードウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定のみを説明することを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムカードウィジェットの設定

これらのセクションを表示するアイテムカードウィジェットを設定します: ” 説明”、” 最新データ”、” トリガー”、” タグ”。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemcard",
                        "name": "Item card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
```

```
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42257
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 9
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

こちらもご参照ください

- ダッシュボードウィジェットフィールド
- dashboard.create

- dashboard.update

16 アイテムの履歴

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで指定可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでアイテムの履歴 ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。アイテムの履歴ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

以下のパラメータは、アイテムの履歴ウィジェットでサポートされています。

パラメータ	タイプ	名前	値
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
レイアウト	0	layout	0 - *(デフォルト)* 水平; 1 - 垂直。
列（下記を参照）			
行を表示	0	show_lines	指定できる値の範囲は 1～100 です。 デフォルト: 25。
ホストを上書き	1	override_hostid._reference	ABCDE._hostid - ホストのデータソースとして、互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）を指定します。 DASHBOARD._hostid - ホストのデータソースとして、ダッシュボードのホストセレクターを指定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
高度な設定（下記を参照）			
Reference	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

列

列には、アイテムパラメーターの設定に応じて、共通パラメーターと追加パラメーターがあります。

Note:
列に関連するすべてのパラメーターにおいて、プロパティ名に含まれる番号 (例: columns.0.name) は、パラメーターが設定されている列を参照します。

以下のパラメータはすべての列でサポートされています。

パラメータ	type	name	value
Name	1	columns.0.name	任意の文字列値 パラメータの動作: - 必須
項目	4	columns.0.itemid	項目 ID テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートで設定されている項目のみを設定する必要があります。 パラメーターの動作: - 必須
基本色	1	columns.0.base_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

設定された *Item* が数値型の項目の場合、以下の列パラメータがサポートされます。

Parameter	type	name	value
Display	0	columns.0.display	1 - *(default)* そのまま; 2 - Bar; 3 - Indicators.
Min	1	columns.0.min	任意の数値 Parameter behavior: - *Display* が”Bar” または”Indicators” に設定されている場合に サポートされます。
最大	1	columns.0.max	任意の数値 パラメータの動作: - 表示が” バー” または” インジケーター” に設定されている場合にサポートされます。
しきい値			
色	1	columns.0.thresholds.color	16 進数のカラーコード（例：FF0000）
しきい値	1	columns.0.thresholds.threshold	任意の数値 シンタックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています
履歴データ	0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - 傾向

設定されたアイテムが文字、テキスト、またはログタイプのアイテムの場合、次の列パラメータがサポートされます。

Parameter	type	name	value
ハイライト			
ハイライト	1	columns.0.highlights.color	16 進数のカラーコード（例: FF0000）。
しきい値	1	columns.0.highlights.pattern	任意の正規表現。
表示	0	columns.0.display	1 - *(デフォルト)* そのまま表示; 4 - HTML; 5 - 1 行。

Parameter	type	name	value
1 行	0	columns.0.max_length	指定可能な値の範囲は 1～500 です。 デフォルト: 100。 パラメータの動作: - 表示が「1 行」に設定されている場合にサポート
等幅フォントを使用	0	columns.0.monospace_font	0 - *(デフォルト)* デフォルトのフォントを使用; 1 - 等幅フォントを使用。
現地時間を表示	0	columns.0.local_time	0 - *(デフォルト)* タイムスタンプを表示; 1 - 現地時間を表示。 パラメータの動作: - アイテムがログタイプのアイテムに設定され、タイムスタンプを表示が「有効」に設定されている場合にサポート

設定されたアイテムがバイナリタイプのアイテムの場合、以下の列パラメータがサポートされます。

パラメータ	type	名前	値
サムネイルを表示	1	columns.0.show_thumbnail	0 - *(デフォルト)* 無効; 1 - 有効

詳細設定

アイテム履歴ウィジェットでは、以下の詳細設定パラメータがサポートされています。

パラメータ	type	name	値
新しい値タイムスタンプを表示	0	sortorder	0 - *(デフォルト)* 上部; 1 - 下部。
	0	show_timestamp	0 - *(デフォルト)* 無効; 1 - 有効。

パラメータ	type	name	値
列ヘッダーを表示	0	show_column_header	0 - オフ; 1 - 横; 2 - *(デフォルト)* 縦。
期間	1	time_period._referenceDASHBOARD	._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をデータソースとして設定。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータのみで期間を設定することもできます。
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.to が設定されている場合は必須
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.from が設定されている場合は必須

例

次の例は、アイテムヒストリウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムヒストリウィジェットの設定

2 つの数値アイテム"42269" と"42270" の最新データを表示するアイテムヒストリウィジェットを設定します。さらに、アイテム列を縦に、列名を横に表示するように設定します。表示データは 15 行に制限し、タイムスタンプ列を別に用意します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemhistory",
                        "name": "Item history",
                        "x": "0",
                        "y": "0",
                        "width": "18",
                        "height": "6",
```

```
                    "view_mode": "0",
                    "fields": [
                        {
                            "type": "0",
                            "name": "layout",
                            "value": "1"
                        },
                        {
                            "type": "1",
                            "name": "columns.0.name",
                            "value": "CPU utilization"
                        },
                        {
                            "type": "4",
                            "name": "columns.0.itemid",
                            "value": "42269"
                        },
                        {
                            "type": "1",
                            "name": "columns.1.name",
                            "value": "Memory utilization"
                        },
                        {
                            "type": "4",
                            "name": "columns.1.itemid",
                            "value": "42270"
                        },
                        {
                            "type": "0",
                            "name": "show_lines",
                            "value": "15"
                        },
                        {
                            "type": "0",
                            "name": "show_timestamp",
                            "value": "1"
                        },
                        {
                            "type": "0",
                            "name": "show_column_header",
                            "value": "1"
                        },
                        {
                            "type": "1",
                            "name": "reference",
                            "value": "KIVKD"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
    ],
    "users": [
        {
            "userid": 1,
            "permission": 3
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

17 アイテムナビゲーター

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでアイテムナビゲーター ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。アイテムナビゲーターウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

アイテムナビゲーターウィジェットでは、次のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、各ホストグループに対して、プロパティ名の番号を増加させたダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		type	name	value
	ホストグループ（ウィジェット）	1	groupids._reference	ホストグループ ID の代わりに指定: ABCDE._hostgroupids - 互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）をホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、各ホストに対して、プロパティ名の番号を増加させたダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストの場合、ホストグループパラメータは未設定にするか、設定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホスト（ウィジェット/ダッシュボード）	1	hostids._reference	ホスト ID の代わりに指定: DASHBOARD.hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 ABCDE._hostid - 互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）をホストのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストタグ				
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		type	name	value
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
アイテムパターンアイテムタグ		1	items.0	アイテムの名前またはパターン。 注: 複数のアイテムパターンを設定するには、各アイテムパターンに対して、プロパティ名の番号を増加させたダッシュボードウィジェットフィールドオブジェクトを作成します。
	評価タイプ	0	item_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	item_tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - アイテムタグを設定する場合は必須

パラメータ		type	name	value
	演算子	0	item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - アイテムタグを設定する場合は必須
	タグ値	1	item_tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - アイテムタグを設定する場合は必須
状態		0	state	-1 - *(デフォルト)* すべて; 0 - 正常; 1 - サポートされていない。
障害を表示		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制されていない障害; 2 - なし。
グループ化	属性	0	group_by.0.attribute	0 - ホストグループ; 1 - ホスト名; 2 - ホストタグの値; 3 - アイテムタグの値。 注: プロパティ名の番号は、グループ化属性リスト内の属性の順序を示します。 パラメータの動作: - グループ化を設定する場合は必須
	値	1	group_by.0.tag_name	任意の文字列値。 注: プロパティ名の番号は、属性パラメータで設定されたグループ化属性を示します。 パラメータの動作: - グループ化を設定し、属性を「ホストタグの値」または「アイテムタグの値」に設定する場合は必須
アイテムの上限		0	show_lines	指定可能な値の範囲は 1～9999 です。 デフォルト: 100。

パラメータ	type	name	value
Reference	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、アイテムナビゲーターウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムナビゲーターウィジェットの設定

最大 1000 個のアイテムをホスト別、さらに” コンポーネント” アイテムタグ値別にグループ化して表示するアイテムナビゲーターウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemnavigator",
                        "name": "Item navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 1000
                            },
                            {
```

```
                                "type": 1,
                                "name": "reference",
                                "value": "DFNLK"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

18 アイテム値

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでアイテムの値 ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。アイテムの値ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメーター

アイテム値ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
アイテム		4	itemid.0	アイテムの ID。 パラメーターの動作: - *Item (Widget)* が設定されていない場合は必須
	Item (Widget)	1	itemid._reference	アイテムの ID の代わりに: `ABCDE._itemid` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をアイテムのデータソースとして設定します。 パラメーターの動作: - *Item* が設定されていない場合は必須
表示		0	show.0	1 - 説明; 2 - 値; 3 - 時刻; 4 - 変化インジケーター; 5 - スパークライン。 デフォルト: 1, 2, 3, 4。 注: 複数の値を設定するには、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。
ホストの上書き		1	override_hostid._reference	`ABCDE._hostid` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します; `DASHBOARD._hostid` - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

高度な設定

アイテム値ウィジェットでは、以下の高度な設定パラメータがサポートされています。

Note:
Thresholds プロパティ名の数字（例：thresholds.0.color）は、昇順で並べ替えられたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix Web インターフェースでウィジェット設定を更新した後、値は昇順に並べ替えられます（例："thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5"）。

パラメータ	type	name	value
背景色	1	bg_color	16 進数のカラーコード（例：FF0000）。 デフォルト："" （空）。

パラメータ		type	name	value
しきい値				
	色	1	thresholds.0.color	16 進数のカラーコード（例：FF0000）。
	しきい値	1	thresholds.0.threshold	任意の文字列値。
集計関数		0	aggregate_function	0 - *(デフォルト)* 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクタをデータソースとして設定; ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE のもの）をデータソースとして設定。 デフォルト：DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。 パラメータの動作: - 集計関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート
	From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート - time_period.to が設定されている場合は必須
	To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が「min」「max」「avg」「count」「sum」「first」「last」に設定されている場合にサポート - time_period.from が設定されている場合は必須
履歴データ		0	history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

説明

表示が「説明」に設定されている場合、次の高度な設定パラメータがサポートされます。

パラメータ	type	name	value
説明	1	description	マクロを含む任意の文字列値。 サポートされるマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ。 デフォルト: {ITEM.NAME}。
水平位置	0	desc_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右。 2 つ以上の要素（説明、値、時刻）に同じ水平位置と垂直位置を指定することはできません。
垂直位置	0	desc_v_pos	0 - 上; 1 - 中央; 2 - *(デフォルト)* 下。 2 つ以上の要素（説明、値、時刻）に同じ水平位置と垂直位置を指定することはできません。
サイズ	0	desc_size	指定可能な値の範囲は 1～100 です。 デフォルト: 15。
太字	0	desc_bold	0 - *(デフォルト)* 無効; 1 - 有効。
色	1	desc_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。

値

表示が「値」に設定されている場合、次の高度な設定パラメータがサポートされます。

パラメータ		type	name	value
小数点以下の桁数				
	小数点以下の桁数	0	decimal_places	指定可能な値の範囲は 1～10 です。 デフォルト: 2。
	サイズ	0	decimal_size	指定可能な値の範囲は 1～100 です。 デフォルト: 35。
位置				
	水平方向の位置	0	value_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右。 2 つ以上の要素（説明、値、時刻）に同じ水平方向の位置と垂直方向の位置を指定することはできません。

パラメータ		type	name	value
	垂直方向の位置	0	value_v_pos	0 - 上; 1 - *(デフォルト)* 中央; 2 - 下。 2 つ以上の要素（説明、値、時刻）に同じ水平方向の位置と垂直方向の位置を指定することはできません。
	サイズ	0	value_size	指定可能な値の範囲は 1～100 です。 デフォルト: 45。
	太字	0	value_bold	0 - 無効; 1 - *(デフォルト)* 有効。
	色	1	value_color	16 進数の色コード（例: FF0000）。 デフォルト: ""（空）。
単位				
	単位（チェックボックス）	0	units_show	0 - 無効; 1 - *(デフォルト)* 有効。
	単位（値）	1	units	任意の文字列値。
	位置	0	units_pos	0 - 値の前; 1 - 値の上; 2 - *(デフォルト)* 値の後; 3 - 値の下。
	サイズ	0	units_size	指定可能な値の範囲は 1～100 です。 デフォルト: 35。
	太字	0	units_bold	0 - 無効; 1 - *(デフォルト)* 有効。
	色	1	units_color	16 進数の色コード（例: FF0000）。 デフォルト: ""（空）。

時刻

表示が「時刻」に設定されている場合、次の高度な設定パラメータがサポートされます。

パラメータ	type	name	value
水平位置	0	time_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右。 2 つ以上の要素（説明、値、時刻）に同じ水平位置と垂直位置を指定することはできません。
垂直位置	0	time_v_pos	0 - *(デフォルト)* 上; 1 - 中央; 2 - 下。 2 つ以上の要素（説明、値、時刻）に同じ水平位置と垂直位置を指定することはできません。
サイズ	0	time_size	指定可能な値の範囲は 1～100 です。 デフォルト: 15。
太字	0	time_bold	0 - *(デフォルト)* 無効; 1 - 有効。

パラメータ	type	name	value
色	1	time_color	16 進数のカラーコード（例: FF0000）。 デフォルト: ""（空）。

変更インジケーター

表示が" 変更インジケーター" に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
変更インジケーター ↑ の色	1	up_color	16 進数カラーコード（例：FF0000） デフォルト："（空）
変更インジケーター ↓ の色	1	down_color	16 進数カラーコード（例：FF0000） デフォルト："（空）
インジケーターの色変更	1	updown_color	16 進数カラーコード（例：FF0000） デフォルト："（空）

スパークライン

表示を「スパークライン」に設定した場合、次の高度な設定パラメータがサポートされます。

パラメータ	タイプ	名前	値
幅	0	sparkline.width	指定可能な値は 0～10 です。 デフォルト: 1。
塗りつぶし	0	sparkline.fill	指定可能な値は 0～10 です。 デフォルト: 3。
色	1	sparkline.color	16 進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。
期間	1	sparkline.time_period.reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメータでのみ期間を設定することもできます。
開始	1	sparkline.time_period.from	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間形式（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - sparkline.time_period.to が設定されている場合は必須
終了	1	sparkline.time_period.to	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間形式（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - sparkline.time_period.from が設定されている場合は必須

パラメータ	タイプ	名前	値
履歴データ	0	sparkline.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

例

次の例は、アイテムの値ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムの値ウィジェットの設定

アイテム"42266"(Zabbix エージェントの可用性) のアイテム値を表示するアイテムの値ウィジェットを設定します。さらにウィジェットを複数の詳細オプションを使用して、Zabbix エージェントの可用性ステータスに基づいて変更される動的な背景色といった視覚的な調整をします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "item",
                        "name": "Item value",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42266
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
```

```
            "type": 1,
            "name": "description",
            "value": "Agent status"
        },
        {
            "type": 0,
            "name": "desc_h_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "desc_v_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "desc_bold",
            "value": 1
        },
        {
            "type": 1,
            "name": "desc_color",
            "value": "F06291"
        },
        {
            "type": 0,
            "name": "value_h_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "value_size",
            "value": 25
        },
        {
            "type": 1,
            "name": "value_color",
            "value": "FFFF00"
        },
        {
            "type": 0,
            "name": "units_show",
            "value": 0
        },
        {
            "type": 0,
            "name": "time_h_pos",
            "value": 2
        },
        {
            "type": 0,
            "name": "time_v_pos",
            "value": 2
        },
        {
            "type": 0,
            "name": "time_size",
            "value": 10
        },
        {
            "type": 0,
            "name": "time_bold",
```

```
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_color",
                                "value": "9FA8DA"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "0"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "D1C4E9"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "1"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create

- dashboard.update

説明

これらのパラメータと、対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでマップ ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。マップウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメーター

マップウィジェットでは、以下のパラメーターがサポートされています。

パラメータ	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - (デフォルト) 15 分
マップ	8	sysmapid.0	マップID。 パラメーターの動作: - マップ (ウィジェット) が設定されていない場合、必須
マップ (ウィジェット)	1	sysmapid._reference	ABCDE._mapid - マップのデータソースとして、マップナビゲーションウィジェット (参照パラメーターが"ABCDE" に設定されている) を設定します。 パラメーターの動作: - マップが設定されていない場合、必須
参照	1	reference	5 文字からなる任意の文字列値 (例: ABCDE や JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

例

次の例は、マップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

マップウィジェットの設定

マップ"1" を表示するマップウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
```

```
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 0,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 8,
                                "name": "sysmapid.0",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

リンクされたマップウィジェットの設定

マップナビゲーションウィジェットにリンクされたマップウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
```

```
            "type": "map",
            "name": "Map",
            "x": 0,
            "y": 5,
            "width": 54,
            "height": 5,
            "view_mode": 0,
            "fields": [
                {
                    "type": 1,
                    "name": "sysmapid._reference",
                    "value": "ABCDE._mapid"
                }
            ]
        },
        {
            "type": "navtree",
            "name": "Map navigation tree",
            "x": 0,
            "y": 0,
            "width": 18,
            "height": 5,
            "view_mode": 0,
            "fields": [
                {
                    "type": 1,
                    "name": "navtree.1.name",
                    "value": "Element A"
                },
                {
                    "type": 1,
                    "name": "navtree.2.name",
                    "value": "Element B"
                },
                {
                    "type": 1,
                    "name": "navtree.3.name",
                    "value": "Element C"
                },
                {
                    "type": 1,
                    "name": "navtree.4.name",
                    "value": "Element A1"
                },
                {
                    "type": 1,
                    "name": "navtree.5.name",
                    "value": "Element A2"
                },
                {
                    "type": 1,
                    "name": "navtree.6.name",
                    "value": "Element B1"
                },
                {
                    "type": 1,
                    "name": "navtree.7.name",
                    "value": "Element B2"
                },
                {
                    "type": 0,
                    "name": "navtree.4.parent",
```

```
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.6.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.7.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.1.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.2.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.3.order",
            "value": 3
        },
        {
            "type": 0,
            "name": "navtree.4.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.6.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.7.order",
            "value": 2
        },
        {
            "type": 8,
            "name": "navtree.6.sysmapid",
            "value": 1
        },
        {
            "type": 1,
            "name": "reference",
            "value": "ABCDE"
        }
```

```
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update
- Map navigation tree

20 マップナビゲーションツリー

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでマップナビゲーションツリー ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。マップナビゲーションツリーウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

Parameters

マップナビゲーションツリーウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(デフォルト)* 15 分
利用できないマップを表示	1	show_unavailable	0 - *(デフォルト)* 無効; 1 - 有効
参照	1	reference	5 文字の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

マップナビゲーションツリー要素の設定には、以下のパラメータがサポートされています。

パラメータ	タイプ	名前	値
名前	1	navtree.1.name	任意の文字列値 注: プロパティ名に含まれる数字は要素番号を設定します。
リンクされたマップ	8	navtree.1.sysmapid	マップ ID 注: プロパティ名に含まれる数字は、マップがリンクされている要素を参照します。
要素階層を作成するためのパラメーター	0	navtree.1.parent	親要素番号 注: プロパティ名に含まれる数字は、子要素を参照します。プロパティ値は親要素を参照します。
	0	navtree.1.order	マップナビゲーションツリー内の要素の位置 注: プロパティ名に含まれる数字は、要素番号を参照します。プロパティ値は、マップナビゲーションツリー内の要素の位置を参照します。親要素の位置は、マップナビゲーションツリー全体内で決定されます。子要素の位置は、親要素内で決定されます。

例

次の例は、マップナビゲーションウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

マップナビゲーションウィジェットの設定

以下のマップナビゲーションツリーを表示するマップナビゲーションウィジェットを設定します:

- 要素 A
  - 要素 A1
  - 要素 A2
- 要素 B
  - 要素 B1 (リンクされたマップウィジェットに表示できるリンクされたマップ"1" を含む)
  - 要素 B2
- 要素 C

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
```

```
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
                                "type": 1,
                                "name": "navtree.2.name",
                                "value": "Element B"
                            },
                            {
                                "type": 1,
                                "name": "navtree.3.name",
                                "value": "Element C"
                            },
                            {
                                "type": 1,
                                "name": "navtree.4.name",
                                "value": "Element A1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.5.name",
                                "value": "Element A2"
                            },
                            {
                                "type": 1,
                                "name": "navtree.6.name",
                                "value": "Element B1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.7.name",
                                "value": "Element B2"
                            },
                            {
                                "type": 0,
                                "name": "navtree.4.parent",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.5.parent",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.6.parent",
```

```
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "navtree.7.parent",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "navtree.1.order",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "navtree.2.order",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "navtree.3.order",
                        "value": 3
                    },
                    {
                        "type": 0,
                        "name": "navtree.4.order",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "navtree.5.order",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "navtree.6.order",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "navtree.7.order",
                        "value": 2
                    },
                    {
                        "type": 8,
                        "name": "navtree.6.sysmapid",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "reference",
                        "value": "HJQXF"
                    }
                ]
            }
        ]
    }
],
"userGroups": [
    {
        "usrgrpid": 7,
        "permission": 2
    }
```

```
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update
- Map

21 パイチャート

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで円グラフ ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時に検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。円グラフウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

円グラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

データセット

データセットの設定では、以下のパラメータがサポートされています。

Note:
プロパティ名の最初の番号（例: ds.0.hosts.0、ds.0.items.0）は特定のデータセットを表し、2 番目の番号がある場合は、設定されたホストまたはアイテムを表します。

パラメータ		タイプ	名前	値
データセットタイプ		0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。
	アイテム	4	ds.0.itemids.0	アイテム ID。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されたアイテムのみを設定してください。 注: 複数のアイテムを設定するには、プロパティ名の番号を増分して、アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテム（ウィジェット）が設定されていない場合は必須
	アイテム（ウィジェット）	1	ds.0.itemids.0._reference	アイテム ID の代わりに指定します: `ABCDE._itemid` - アイテムのデータソースとして、互換性のあるウィジェット（*Reference* パラメータを”ABCDE” に設定）を設定します。 注: 複数のウィジェットを設定するには、プロパティ名の番号を増分して、ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、アイテムが設定されていない場合は必須
	色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合は必須
	アイテムタイプ	0	ds.0.type.0	0 - *(デフォルト)* 通常; 1 - 合計。 「合計」は、チャート全体で 1 つのアイテムにのみ設定できます。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合にサポート
	ホストパターン	1	ds.0.hosts.0	ホスト名またはパターン（例: ”Zabbix*”）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		タイプ	名前	値
	アイテムパターン	1	ds.0.items.0	アイテム名またはパターン（例: ”*: 1 秒あたりに処理された * 値の数”）。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されたアイテムのパターンのみを設定してください。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
	色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合にサポート
	カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 指定可能な値の範囲は 0～11 です。 デフォルト: 0。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合にサポート
集計関数		0	ds.0.aggregate_function	1 - 最小; 2 - 最大; 3 - 平均; 4 - カウント; 5 - 合計; 6 - 最初; 7 - *(デフォルト)* 最後。
データセットの集計		0	ds.0.dataset_aggregation	0 - *(デフォルト)* なし; 1 - 最小; 2 - 最大; 3 - 平均; 4 - カウント; 5 - 合計。 パラメータの動作: - アイテムタイプが「合計」に設定されている場合にサポート
データセットラベル		1	ds.0.data_set_label	任意の文字列値。 デフォルト: ""（空）。
アイテムタグ	評価タイプ	0	ds.0.item_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。

パラメータ	タイプ	名前	値
タグ名	1	ds.0.item_tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須
演算子	0	ds.0.item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須
タグの値	1	ds.0.item_tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須

表示オプション

表示オプションの設定では、次のパラメータがサポートされています。

パラメータ	タイプ	名前	値
履歴データの選択	0	source	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。
描画	0	draw_type	0 - *(デフォルト)* 円グラフ; 1 - ドーナツ。
幅	0	width	20 - 半径の 20%; 30 - 半径の 30%; 40 - 半径の 40%; 50 - *(デフォルト)* 半径の 50%。 パラメータの動作: - 描画が「ドーナツ」に設定されている場合にサポート

パラメータ		タイプ	名前	値
	線の幅	0	stroke	指定可能な値の範囲は 0～10 です。 デフォルト: 0。
	合計値を表示	0	total_show	パラメータの動作: - 描画が「ドーナツ」に設定されている場合にサポート 0 - *(デフォルト)* 無効; 1 - 有効。
	サイズ	0	value_size_type	パラメータの動作: - 描画が「ドーナツ」に設定されている場合にサポート 0 - *(デフォルト)* 自動; 1 - カスタム。
	サイズ（カスタムサイズの値）	0	value_size	パラメータの動作: - 合計値を表示が「有効」に設定されている場合にサポート 指定可能な値の範囲は 1～100 です。 デフォルト: 20。
	小数点以下の桁数	0	decimal_places	パラメータの動作: - 合計値を表示が「有効」に設定されている場合にサポート 指定可能な値の範囲は 0～6 です。 デフォルト: 2。
	単位（チェックボックス）	0	units_show	パラメータの動作: - 合計値を表示が「有効」に設定されている場合にサポート 0 - *(デフォルト)* 無効; 1 - 有効。
	単位（値）	1	units	パラメータの動作: - 合計値を表示が「有効」に設定されている場合にサポート 任意の文字列値。
	太字	0	value_bold	パラメータの動作: - 単位（チェックボックス）が「有効」に設定されている場合にサポート 0 - *(デフォルト)* 無効; 1 - 有効。
	色	1	value_color	パラメータの動作: - 合計値を表示が「有効」に設定されている場合にサポート 16 進数のカラーコード（例: FF0000）。
セクター間の間隔		0	space	パラメータの動作: - 合計値を表示が「有効」に設定されている場合にサポート 指定可能な値の範囲は 0～10 です。 デフォルト: 1。

パラメータ	タイプ	名前	値
N% 未満のセクターを統合 (チェックボックス)	0	merge	0 - *(デフォルト)* 無効; 1 - 有効。
N% 未満のセクターを統合 (値)	0	merge_percent	指定可能な値の範囲は 1～10 です。 デフォルト: 1。 パラメータの動作: - *N*% 未満のセクターを統合（チェックボックス）が「有効」に設定されている場合にサポート
N% 未満のセクターを統合 (色)	1	merge_color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - *N*% 未満のセクターを統合（チェックボックス）が「有効」に設定されている場合にサポート

期間

期間の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
期間	1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.to が設定されている場合は必須
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）での有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.from が設定されている場合は必須

凡例

Legend の設定では、次のパラメータがサポートされています。

パラメータ	type	name	value
凡例を表示	0	legend	0 - 無効; 1 - *(デフォルト)* 有効。
値を表示	0	legend_value	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合に サポート
集計関数を表示	0	legend_aggregation	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合に サポート
行	0	legend_lines_mode	0 - *(デフォルト)* 固定; 1 - 可変。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合に サポート
行数/ 行の最大数	0	legend_lines	指定可能な値の範囲は 1～10 です。 デフォルト: 1。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合に サポート

パラメータ	type	name	value
列数	0	legend_columns	指定可能な値の範囲は 1～4 です。 デフォルト: 4。 パラメータの動作: - 凡例を表示が「有効」に設定され、値を表示が「無効」に設定されている場合に サポート

例

次の例は、円グラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

円グラフウィジェットの設定

円グラフウィジェットは次のように設定します。

- 1 つのホストに合計 9 つのアイテムの 2 つのデータセット。
- 最初のデータセットは「アイテムリスト」タイプで、すべて「通常」タイプで異なる色で表現される 3 つのアイテムで構成されます。
- 2 番目のデータセットは「アイテムパターン」タイプで、6 つのアイテムで構成され、各アイテムに集計が設定され、カスタムカラーで表現されます。
- 2 番目のデータセットにはカスタムデータセットラベルもあります。
- 円グラフのデータは、カスタム幅と中央に単位付きの合計値を持つドーナツグラフとして表示されます。
- 円グラフのデータは、直近 3 時間のカスタム期間で表示および集計されます。
- 円グラフの凡例には、設定されたアイテムが 4 行で表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "piechart",
                        "name": "Pie chart",
                        "x": 0,
                        "y": 0,
                        "width": 24,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
                            {
```

```
                "type": 0,
                "name": "ds.0.type.0",
                "value": 0
            },
            {
                "type": 4,
                "name": "ds.0.itemids.1",
                "value": 23269
            },
            {
                "type": 1,
                "name": "ds.0.color.1",
                "value": "BF00FF"
            },
            {
              "type": 0,
              "name": "ds.0.type.1",
              "value": 0
            },
            {
                "type": 4,
                "name": "ds.0.itemids.2",
                "value": 23257
            },
            {
                "type": 1,
                "name": "ds.0.color.2",
                "value": "0040FF"
            },
            {
              "type": 0,
              "name": "ds.0.type.2",
              "value": 0
            },
            {
                "type": 1,
                "name": "ds.1.hosts.0",
                "value": "Zabbix server"
            },
            {
                "type": 1,
                "name": "ds.1.items.0",
                "value": "*: Number of processed *values per second"
            },
            {
                "type": 1,
                "name": "ds.1.color",
                "value": "000000"
            },
            {
                "type": 0,
                "name": "ds.1.aggregate_function",
                "value": 3
            },
            {
                "type": 1,
                "name": "ds.1.data_set_label",
                "value": "Number of processed values per second"
            },
            {
                "type": 0,
                "name": "draw_type",
```

```
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "width",
                                "value": 30
                            },
                            {
                                "type": 0,
                                "name": "total_show",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "units_show",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "graph_time",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_period.from",
                                "value": "now-3h"
                            },
                            {
                                "type": 1,
                                "name": "time_period.to",
                                "value": "now"
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

22 問題のあるホスト

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで障害ホスト ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。障害ホストウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

障害ホストウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - *Reference* パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
除外するホストグループホスト		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定であるか、指定したホストが属するホストグループが少なくとも 1 つ設定されている必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - *Reference* パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害		1	problem	障害イベント名（大文字小文字を区別しない、完全一致または部分一致）。
深刻度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（障害タグを設定する場合）

パラメーター		type	name	value
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（障害タグを設定する場合）
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（障害タグを設定する場合）
抑止中の障害を表示		0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
障害のないグループを非表示		0	hide_empty_groups	0 - *(デフォルト)* 無効; 1 - 有効。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害の表示		0	ext_ack	0 - *(デフォルト)* すべて; 1 - 未確認のみ; 2 - 分離。
Reference		1	reference	5 文字からなる任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

例

次の例は、障害中のホストウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

障害中のホストウィジェットの設定

ホストグループ”2” と”4” に属する、名前に”CPU” という文字列が含まれ、深刻度が” 警告”、” 軽度の障害”、” 重度の障害”、” 致命的な障害” の障害が発生しているホストを表示する障害中のホストウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemhosts",
                        "name": "Problem hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "problem",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            }
                        ]
                    }
                ]
```

```
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

23 問題

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで障害 ウィジェットを設定できます。

> **Attention:**
> ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。障害ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

障害ウィジェットでは、次のパラメータがサポートされています。

パラメータ		タイプ	名前	値
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
表示		0	show	1 - *(デフォルト)* 最近の障害; 2 - 履歴; 3 - 障害。
ホストグループ		2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増分させて、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID の代わりに指定します: `ABCDE._hostgroupids` - *Reference* パラメータに"ABCDE" を設定した互換性のあるウィジェットを、ホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストグループを除外		2	exclude_groupids.0	ホストグループ ID。 注: 複数のホストグループを除外するには、プロパティ名の番号を増分させて、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、プロパティ名の番号を増分させて、ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを設定する場合、ホストグループパラメータは未設定にするか、設定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホスト ID の代わりに指定します: `DASHBOARD.hostids` - ダッシュボードのホストセレクターを、ホストのデータソースとして設定します; `ABCDE._hostids` - *Reference* パラメータに"ABCDE" を設定した互換性のあるウィジェットを、ホストのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
障害		1	problem	障害のイベント名（大文字と小文字を区別しない、完全な名前またはその一部）。

パラメータ		タイプ	名前	値
深刻度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、プロパティ名の番号を増分させて、値ごとにダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* AND/OR; 2 - OR。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 障害タグを設定する場合は必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 障害タグを設定する場合は必須
	タグの値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 障害タグを設定する場合は必須
タグを表示		0	show_tags	0 - *(デフォルト)* なし; 1 - 1; 2 - 2; 3 - 3。
タグ名(形式)		0	tag_name_format	0 - *(デフォルト)* 完全; 1 - 短縮; 2 - なし。 パラメータの動作: - タグを表示が「1」、「2」、または「3」に設定されている場合にサポート

パラメータ	タイプ	名前	値
タグ表示の優先順位	1	tag_priority	カンマ区切りのタグ一覧。 パラメータの動作: - タグを表示が「1」、「2」、または「3」に設定されている場合にサポート
運用データを表示	0	show_opdata	0 - *(デフォルト)* なし; 1 - 個別に表示; 2 - 障害名とともに表示。
抑制された障害を表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
確認ステータス	0	acknowledgement_status	0 - *(デフォルト)* すべて; 1 - 未確認; 2 - 確認済み。
自分によるもの	0	acknowledged_by_me	0 - *(デフォルト)* 無効; 1 - 有効。
エントリのソート基準	0	sort_triggers	1 - 深刻度（降順）; 2 - ホスト（昇順）; 3 - 時刻（昇順）; 4 - *(デフォルト)* 時刻（降順）; 13 - 深刻度（昇順）; 14 - ホスト（降順）; 15 - 障害（昇順）; 16 - 障害（降順）。 「時刻（降順）」および「時刻（昇順）」以外のすべての値では、タイムラインを表示パラメータを「無効」に設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、「ホスト(昇順)」および「ホスト（降順）」はサポートされません。

パラメータ	タイプ	名前	値
タイムラインを表示	0	show_timeline	0 - 無効; 1 - *(デフォルト)* 有効。 パラメータの動作: - エントリのソート基準が「時刻（降順）」または「時刻（昇順）」に設定されている場合にサポート
行全体を強調表示	0	highlight_row	0 - *(デフォルト)* 無効; 1 - 有効。 このパラメータはハイコントラストテーマではサポートされません。
表示行数	0	show_lines	指定可能な値の範囲は 1～100 です。 デフォルト: 25。
Reference	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、障害ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

障害ウィジェットの設定

ホストグループ”4” で、以下の条件を満たす障害を表示する障害ウィジェットを設定します:

- ”scope” という名前のタグが付いており、そのタグの値が”performance”、”availability” または”capacity” である障害。
- 深刻度が” 警告”、” 軽度の障害”、” 重度の障害”、” 致命的な障害” である障害。

さらに、タグと運用データを表示するようにウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "name": "Problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
```

```
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 2,
                        "name": "groupids.0",
                        "value": 4
                    },
                    {
                        "type": 1,
                        "name": "tags.0.tag",
                        "value": "scope"
                    },
                    {
                        "type": 0,
                        "name": "tags.0.operator",
                        "value": 0
                    },
                    {
                        "type": 1,
                        "name": "tags.0.value",
                        "value": "performance"
                    },
                    {
                        "type": 1,
                        "name": "tags.1.tag",
                        "value": "scope"
                    },
                    {
                        "type": 0,
                        "name": "tags.1.operator",
                        "value": 0
                    },
                    {
                        "type": 1,
                        "name": "tags.1.value",
                        "value": "availability"
                    },
                    {
                        "type": 1,
                        "name": "tags.2.tag",
                        "value": "scope"
                    },
                    {
                        "type": 0,
                        "name": "tags.2.operator",
                        "value": 0
                    },
                    {
                        "type": 1,
                        "name": "tags.2.value",
                        "value": "capacity"
                    },
                    {
                        "type": 0,
                        "name": "severities.0",
                        "value": 2
                    },
                    {
                        "type": 0,
                        "name": "severities.1",
                        "value": 3
```

```
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "show_tags",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show_opdata",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

24 問題の深刻度別

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで深刻度別の障害 ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。深刻度別の障害ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメーター

重大度別障害ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新しない; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: ABCDE._hostgroupids - 参照パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
除外するホストグループ		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
	ホスト (ウィジェット/ダッシュボード)	1	hostids._reference	ホストの ID の代わりに: DASHBOARD.hostids - ダッシュボードのホストセレクターをホストのデータソースとして設定します; ABCDE._hostids - 参照パラメーターを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害		1	problem	障害のイベント名（大文字小文字を区別しない、完全一致または部分一致）。
重大度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度障害; 4 - 障害; 5 - 重度障害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、各値ごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ				
	評価タイプ	0	evaltype	0 - (デフォルト) And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須 (障害タグを設定する場合)

パラメーター	type	name	value
表示	0	show_type	0 - *(デフォルト)* ホストグループ; 1 - 合計。 このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされず、デフォルトで「合計」に設定されます。
レイアウト	0	layout	0 - *(デフォルト)* 横並び; 1 - 縦並び。 パラメーターの動作: - 表示が「合計」の場合のみサポート
運用データの表示	0	show_opdata	0 - *(デフォルト)* なし; 1 - 個別に; 2 - 障害名と一緒に。
抑制された障害の表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
障害のないグループを非表示	0	hide_empty_groups	0 - *(デフォルト)* 無効; 1 - 有効。 パラメーターの動作: - 表示が「ホストグループ」の場合のみサポート このパラメーターはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
障害の表示	0	ext_ack	0 - *(デフォルト)* すべて; 1 - 未確認のみ; 2 - 分離。
タイムラインの表示	0	show_timeline	0 - 無効; 1 - *(デフォルト)* 有効。

パラメーター	type	name	value
参照	1	reference	5 文字からなる任意の文字列値（例: ABCDE または JBPNL）。この値はウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、深刻度ごとの障害数ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

深刻度ごとの障害数ウィジェットの設定

すべてのホストグループの障害の合計数を表示する深刻度ごとの障害数ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemsbysv",
                        "name": "Problems by severity",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_type",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

25 散布図

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで散布図 ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。散布図ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

散布図ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター	type	name	値
リフレッシュ間隔	0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
参照	1	reference	5 文字からなる任意の文字列値（例：ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

データセット

データセットの設定には、以下のパラメータがサポートされています。

Note:
プロパティ名の最初の数字（例: ds.0.hosts.0, ds.0.items.0）は特定のデータセットを表し、2 つ目の数字がある場合は設定されたホストまたはアイテムを表します。

パラメータ	type	name	value
データセットタイプ: アイテムリスト	0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。
色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定されている場合は必須
X 軸	4	ds.0.x_axis_itemids.0	アイテムID。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみを設定する必要があります。 注: 複数のアイテムを設定するには、プロパティ名の数字を増やして各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*X* 軸（ウィジェット）が設定されていない場合は必須
X 軸（ウィジェット）	1	ds.0.x_axis_itemids.0_reference	アイテムIDの代わりに: `ABCDE._itemid` - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をアイテムのデータソースとして設定します。 注: 複数のウィジェットを設定するには、プロパティ名の数字を増やして各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*X* 軸が設定されていない場合は必須

パラメータ	type	name	value
Y 軸	4	ds.0.y_axis_itemids.0	アイテムID。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのみを設定する必要があります。 注: 複数のアイテムを設定するには、プロパティ名の数字を増やして各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*Y* 軸（ウィジェット）が設定されていない場合は必須
Y 軸（ウィジェット）	1	ds.0.y_axis_itemids.0_reference	アイテムIDの代わりに: ABCDE._itemid - 互換性のあるウィジェット（参照パラメータを"ABCDE"に設定）をアイテムのデータソースとして設定します。 注: 複数のウィジェットを設定するには、プロパティ名の数字を増やして各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 パラメータの動作: - データセットタイプが「アイテムリスト」に設定され、*Y* 軸が設定されていない場合は必須
ホストの上書きデータセット: アイテムパターン	1	ds.0.override_hostid.reference	ABCDE._hostid - 互換性のあるウィジェット（" 参照" パラメータを"ABCDE"に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合はサポート
カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 有効な値の範囲: 0-11。 デフォルト: 0。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合はサポート

パラメータ	type	name	value
ホストパターン	1	ds.0.hosts.0	ホスト名またはパターン（例: ”Zabbix*”）。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
X 軸アイテムパターン	1	ds.0.x_axis_items.0	アイテム名またはパターン（例: ”*: Number of processed *values per second”）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのパターンのみを設定する必要があります。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
Y 軸アイテムパターン	1	ds.0.y_axis_items.0	アイテム名またはパターン（例: ”*: Number of processed *values per second”）。 テンプレートダッシュボードでウィジェットを設定する場合、テンプレートで設定されたアイテムのパターンのみを設定する必要があります。 パラメータの動作: - データセットタイプが「アイテムパターン」に設定されている場合は必須
ホストグループ	2	groupids.0	ホストグループID。 注: 複数のホストグループを設定するには、プロパティ名の数字を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストグループ (ウィジェット)	1	groupids._reference	ホストグループID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（参照パラメータを”ABCDE” に設定）をホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストの上書き	1	ds.0.override_hostid._reference	`ABCDE._hostid` - 互換性のあるウィジェット（” 参照” パラメータを”ABCDE” に設定）をホストのデータソースとして設定; `DASHBOARD._hostid` - ダッシュボードのホストセレクタをホストのデータソースとして設定。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		type	name	value
ホストタグ				
	評価タイプ	0	evaltype_host	0 - *(デフォルト)* And/Or; 2 - Or。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の数字はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の数字はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の数字はタグ評価リスト内のタグの順序を参照します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
マーカー		0	ds.0.marker	0 - *(デフォルト)* 楕円; 1 - 四角; 2 - 三角; 3 - ひし形; 4 - 星; 5 - 十字。
マーカーサイズ		0	ds.0.marker_size	0 - *(デフォルト)* 小（6px）; 1 - 中（9px）; 2 - 大（15px）。

パラメータ	type	name	value
タイムシフト	1	ds.0.timeshift	有効な時間文字列（例: 3600, 1h など）。 時間のサフィックスを使用できます。負の値も許可されます。 デフォルト: ""（空）。
集約間隔	1	ds.0.aggregate_interval	有効な時間文字列（例: 3600, 1h など）。 時間のサフィックスを使用できます。 デフォルト: 15m。
集約関数	0	ds.0.aggregate_function	0 - 使用しない; 1 - min; 2 - max; 3 - *(デフォルト)* avg; 4 - count; 5 - sum; 6 - first; 7 - last。

表示オプション

表示オプションの設定には、以下のパラメータがサポートされています。

パラメータ	type	name	値
履歴データの選択	0	source	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。
ラベルにホスト名を表示	0	show_hostnames	0 - *(デフォルト)* 自動; 1 - 表示; 2 - 非表示。 このパラメータは、テンプレートダッシュボード上のウィジェットを設定する場合はサポートされません。

期間

期間の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
期間	1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータが"ABCDE"に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータで期間のみを設定することもできます。

パラメータ	type	name	value
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.to が設定されている場合は必須
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.from が設定されている場合は必須

軸

軸の設定には、以下のパラメータがサポートされています。

パラメータ	タイプ	名前	値
X 軸	0	x_axis	0 - 無効; 1 - *(デフォルト)* 有効。
Y 軸	0	y_axis	0 - *(デフォルト)* 無効; 1 - 有効。
最小	1	x_axis_min y_axis_min	任意の数値。 デフォルト: "" (空)。
最大	1	x_axis_max y_axis_max	任意の数値。 デフォルト: "" (空)。
単位 (タイプ)	0	x_axis_units y_axis_units	0 - *(デフォルト)* 自動; 1 - 静的。
単位 (値)	1	x_axis_static_units y_axis_static_units	任意の文字列。 デフォルト: "" (空)。

凡例

凡例の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	値
凡例を表示	0	legend	0 - 無効; 1 - *(デフォルト)* 有効。
集約関数を表示	0	legend_aggregation	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
行数	0	legend_lines_mode	0 - *(デフォルト)* 固定; 1 - 可変。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート

パラメータ	type	name	値
行数/ 最大行数	0	legend_lines	可能な値の範囲: 1-10。 デフォルト: 1。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート
列数	0	legend_columns	可能な値の範囲: 1-4。 デフォルト: 4。 パラメータの動作: - 凡例を表示が「有効」に設定されている場合にサポート

閾値

閾値の設定には、以下のパラメータがサポートされています。

Note:
閾値プロパティ名の数字 (例: thresholds.0.color) は、昇順にソートされたリスト内の閾値の位置を参照します。ただし、閾値が異なる順序で設定されている場合、Zabbix フロントエンドでウィジェット設定を更新した後に値が昇順にソートされます (例: `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`)。

パラメータ	型	名前	値
色の補間	0	interpolation	0 - *(デフォルト)* 無効; 1 - 有効。
色	1	thresholds.0.color	16 進数のカラーコード（例: FF0000）。
X 軸	1	thresholds.0.x_axis_threshold	任意の数値。サフィックス（例: "1d", "2w", "4K", "8G"）がサポートされています。
Y 軸	1	thresholds.0.y_axis_threshold	任意の数値。サフィックス（例: "1d", "2w", "4K", "8G"）がサポートされています。

例

次の例は、散布図ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定のみを説明することを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

散布図ウィジェットの設定

散布図ウィジェットは、以下のように設定します。

- データセットタイプ「アイテムパターン」のデータセットを 5 つ、それぞれデフォルトの集計 (平均、15 分) と異なるマーカーで設定
- 凡例は、設定したアイテムを 1 行 2 列で表示
- しきい値を設定し、いずれかの軸で値が 80 に達した場合にマーカーを赤色で表示

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "scatterplot",
                        "name": "Scatter plot",
                        "x": 0,
```

```
                "y": 0,
                "width": 36,
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 1,
                        "name": "ds.0.color",
                        "value": "0C5A87"
                    },
                    {
                        "type": 1,
                        "name": "ds.0.hosts.0",
                        "value": "de-frankfurt*"
                    },
                    {
                        "type": 1,
                        "name": "ds.0.x_axis_items.0",
                        "value": "Memory utilization"
                    },
                    {
                        "type": 1,
                        "name": "ds.0.y_axis_items.0",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "ds.0.marker",
                        "value": 0
                    },
                    {
                        "type": 0,
                        "name": "ds.0.marker_size",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "ds.1.color",
                        "value": "66B0D9"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.hosts.0",
                        "value": "fr-paris*"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.x_axis_items.0",
                        "value": "Memory utilization"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.y_axis_items.0",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "ds.1.marker",
                        "value": 1
                    },
                    {
                        "type": 0,
```

```
            "name": "ds.1.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.2.color",
            "value": "0A466A"
        },
        {
            "type": 1,
            "name": "ds.2.hosts.0",
            "value": "lv-riga*"
        },
        {
            "type": 1,
            "name": "ds.2.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.2.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.2.marker",
            "value": 2
        },
        {
            "type": 0,
            "name": "ds.2.marker_size",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.3.color",
            "value": "3394C3"
        },
        {
            "type": 1,
            "name": "ds.3.hosts.0",
            "value": "pl-warsaw*"
        },
        {
            "type": 1,
            "name": "ds.3.x_axis_items.0",
            "value": "Memory utilization"
        },
        {
            "type": 1,
            "name": "ds.3.y_axis_items.0",
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "ds.3.marker",
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.3.marker_size",
            "value": 1
```

```
            },
            {
                "type": 1,
                "name": "ds.4.color",
                "value": "1492C8"
            },
            {
                "type": 1,
                "name": "ds.4.hosts.0",
                "value": "se-stockholm*"
            },
            {
                "type": 1,
                "name": "ds.4.x_axis_items.0",
                "value": "Memory utilization"
            },
            {
                "type": 1,
                "name": "ds.4.y_axis_items.0",
                "value": "CPU utilization"
            },
            {
                "type": 0,
                "name": "ds.4.marker",
                "value": 4
            },
            {
                "type": 0,
                "name": "ds.4.marker_size",
                "value": 1
            },
            {
                "type": 0,
                "name": "legend_columns",
                "value": 2
            },
            {
                "type": 0,
                "name": "interpolation",
                "value": 1
            },
            {
                "type": 1,
                "name": "thresholds.0.color",
                "value": "D40000"
            },
            {
                "type": 1,
                "name": "thresholds.0.x_axis_threshold",
                "value": "80"
            },
            {
                "type": 1,
                "name": "thresholds.0.y_axis_threshold",
                "value": ""
            },
            {
                "type": 1,
                "name": "thresholds.1.color",
                "value": "D40000"
            },
            {
```

```
                                "type": 1,
                                "name": "thresholds.1.x_axis_threshold",
                                "value": ""
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.y_axis_threshold",
                                "value": "80"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "JQISY"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- ダッシュボードウィジェットフィールド
- dashboard.create
- dashboard.update

26 SLA レポート

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*SLA レポート* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*SLA* レポートウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

SLA レポートウィジェットでは、次のパラメータがサポートされています。

パラメータ	タイプ	名前	値
更新間隔	0	rf_rate	0 - *(デフォルト)* 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
SLA	10	slaid.0	SLA ID。 パラメータの動作: - 必須
サービス	9	serviceid.0	サービス ID。
期間の表示	0	show_periods	指定できる値の範囲は 1～100 です。 デフォルト: 20。
開始	1	date_from	YYYY-MM-DD 形式の有効な日付文字列。 d、w、M、y 修飾子を使用した相対日付（例: now、now/d、now/w-1w など）がサポートされています。
終了	1	date_to	YYYY-MM-DD 形式の有効な日付文字列。 d、w、M、y 修飾子を使用した相対日付（例: now、now/d、now/w-1w など）がサポートされています。

例

次の例は、*SLA* レポートウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

SLA レポートウィジェットの設定

SLA "4"、サービス"2" の過去 30 日間の SLA レポートを表示する *SLA* レポートウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "slareport",
                        "name": "SLA report",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
```

```
                        "fields": [
                            {
                                "type": 10,
                                "name": "slaid.0",
                                "value": 4
                            },
                            {
                                "type": 9,
                                "name": "serviceid.0",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "date_from",
                                "value": "now-30d"
                            },
                            {
                                "type": 1,
                                "name": "date_to",
                                "value": "now"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

27 システム情報

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでシステム情報 ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。システム情報ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

System Information ウィジェットでは、以下のパラメータがサポートされています。

Parameter	type	name	value
Refresh interval	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(default)* 15 分。
Show	0	info_type	0 - *(default)* システム統計; 1 - 高可用性ノード。
Show software update check details	0	show_software_update_check_details	0 - *(default)* 無効; 1 - 有効。 Parameter behavior: - Zabbix サーバー設定でAllowSoftwareUpdateCheck が有効で、かつ *Show* が”System stats” に設定されている場合に *supported*

例

次の例は、システム情報ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

システム情報ウィジェットの設定

10 分間隔で更新し、ソフトウェアアップデートのチェックを有効にした状態でシステム統計情報を表示するシステム情報ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "systeminfo",
                        "name": "System information",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
```

```
                                "value": 600
                            },
                            {
                                "type": 0,
                                "name": "show_software_update_check_details",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

28 トップホスト

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、`dashboard.create` および `dashboard.update` メソッドで上位のホスト ウィジェットを設定できます。

> **Attention:**
> ウィジェットの `fields` プロパティは、ダッシュボードの作成時または更新時に検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。上位のホストウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

以下のパラメータは、上位のホストウィジェットでサポートされています。

パラメータ		タイプ	名前	値
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、各ホストグループに対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホストグループ（ウィジェット）	1	groupids._reference	ホストグループ ID の代わりに指定します: `ABCDE._hostgroupids` - *Reference* パラメータに"ABCDE" を設定した互換性のあるウィジェットを、ホストグループのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、各ホストに対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメータは未設定にするか、設定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	ホスト（ウィジェット/ダッシュボード）	1	hostids._reference	ホスト ID の代わりに指定します: `DASHBOARD.hostids` - ダッシュボードのホストセレクターを、ホストのデータソースとして設定します。 `ABCDE._hostids` - *Reference* パラメータに"ABCDE" を設定した互換性のあるウィジェットを、ホストのデータソースとして設定します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
ホストタグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		タイプ	名前	値
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
	タグの値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを設定する場合は必須 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
メンテナンス中のホストを表示列(下記参照)		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。

パラメータ	タイプ	名前	値
並べ替え基準	0	column	設定した列の数値。
順序	0	order	2 - (デフォルト) 上位 N 件; 3 - 下位 N 件。
ホスト数の上限	0	show_lines	1 から 1000 までの値を指定できます。 デフォルト: 10。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

列

列には、データパラメーターの設定に応じて、共通パラメーターと追加パラメーターがあります。

Note:
列に関連するすべてのパラメーターにおいて、プロパティ名に含まれる番号 (例: columns.0.name) は、パラメーターが設定されている列を参照します。

以下のパラメーターはすべての列でサポートされています。

パラメーター	type	name	value
Name	1	columns.0.name	任意の文字列値 パラメーターの動作: - 必須
Data	0	columns.0.data	1 - 項目値 2 - ホスト名 3 - テキスト パラメーターの動作: - 必須
基本色	1	columns.0.base_color	16 進数カラーコード（例：FF0000） パラメータの動作: - 必須

アイテムの値

データに「アイテムの値」を設定した場合、次のパラメータがサポートされます。

Note:
しきい値プロパティ名の最初の番号（例: columnsthresholds.0.color.0）は、しきい値が設定されている列を参照し、2 番目の番号は昇順に並べられたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix の Web インターフェースでウィジェット設定を更新した後、値は昇順に並べ替えられます (例: `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`、`"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`)。

パラメータ	タイプ	名前	値
アイテム	1	columns.0.item	有効なアイテム名。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されているアイテムのみを設定してください。
アイテムの値の表示形式	0	columns.0.display_value_as	0 - *(デフォルト)* 数値; 1 - テキスト; 2 - バイナリ。
表示	0	columns.0.display	1 - *(デフォルト)* そのまま; 2 - バー; 3 - インジケーター; 6 - スパークライン。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定されている場合にサポート
最小	1	columns.0.min	任意の数値。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「バー」または「インジケーター」に設定されている場合にサポート
最大	1	columns.0.max	任意の数値。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「バー」または「インジケーター」に設定されている場合にサポート
小数点以下の桁数	0	columns.0.decimal_places	0-10 の値を指定できます。 デフォルト: 2。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定されている場合にサポート
スパークライン 幅	0	columns.0.sparkline.width	1-10 の値を指定できます。 デフォルト: 1。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート

パラメータ		タイプ	名前	値
	塗りつぶし	0	columns.0.sparkline.fill	1～10 の値を指定できます。 デフォルト: 3。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	色	1	columns.0.sparkline.color	16進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	期間	1	columns.0.sparkline.time_period	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメータでのみ期間を設定できます。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
	開始	1	columns.0.sparkline.time_period.from	絶対形式（YYYY-MM-DD hh:mm:ss）または相対形式（now、now/d、now/w-1w など）の有効な時刻文字列。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート - columns.0.sparkline.time_period.to が設定されている場合に必須
	終了	1	columns.0.sparkline.time_period.to	絶対形式（YYYY-MM-DD hh:mm:ss）または相対形式（now、now/d、now/w-1w など）の有効な時刻文字列。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート - columns.0.sparkline.time_period.from が設定されている場合に必須
	履歴データ	0	columns.0.sparkline.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定され、表示が「スパークライン」に設定されている場合にサポート
しきい値				
	色	1	columnsthresholds.0.color	16進数のカラーコード（例: FF0000）。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定されている場合にサポート

パラメータ		タイプ	名前	値
	しきい値	1	columnsthresholds.0.threshold	任意の文字列値。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定されている場合にサポート
ハイライト	色	1	columns.0.highlights.0.color	16進数のカラーコード（例: FF0000）。 パラメータの動作: - アイテムの値の表示形式が「テキスト」に設定されている場合にサポート
	パターン	1	columns.0.highlights.0.pattern	任意の文字列値。 パラメータの動作: - アイテムの値の表示形式が「テキスト」に設定されている場合にサポート
サムネイルを表示		0	columns.0.show_thumbnail	0 - (デフォルト) 無効; 1 - 有効。 パラメータの動作: - アイテムの値の表示形式が「バイナリ」に設定されている場合にサポート
集計関数		0	columns.0.aggregate_function	0 - (デフォルト) 使用しない; 1 - 最小; 2 - 最大; 3 - 平均; 4 - 件数; 5 - 合計; 6 - 最初; 7 - 最後。
期間		1	columns.0.time_period.reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメータでのみ期間を設定できます。 パラメータの動作: - 集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、「最後」に設定されている場合にサポート
	開始	1	columns.0.time_period.from	絶対形式（YYYY-MM-DD hh:mm:ss）または相対形式（now、now/d、now/w-1w など）の有効な時刻文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、「最後」に設定されている場合にサポート - columns.0.time_period.to が設定されている場合に必須

パラメータ	タイプ	名前	値
終了	1	columns.0.time_period.to	絶対形式（YYYY-MM-DD hh:mm:ss）または相対形式（now、now/d、now/w-1w など）の有効な時刻文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、「最後」に設定されている場合にサポート - columns.0.time_period.from が設定されている場合に必須
履歴データ	0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - アイテムの値の表示形式が「数値」に設定されている場合にサポート
参照	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

テキスト

Data が”Text” に設定されている場合、以下のパラメーターがサポートされます。

Parameter	type	name	value
Text	1	columns.0.text	任意の文字列値。マクロを含みます。 サポートされるマクロ: {HOST.*}, {INVENTORY.*}。 パラメーターの動作: - *required* if *Data* is set to ”Text”

例

次の例は、上位ホストウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

上位ホストウィジェットの設定

ホストグループ”4” で CPU 使用率上位のホストを表示する上位ホストウィジェットを構成します。さらに、次のカスタム列を構成します: ”Host name”, ”CPU utilization in %”, ”1m avg”, ”5m avg”, ”15m avg”, ”Processes”。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "tophosts",
                        "name": "Top hosts",
                        "x": 0,
                        "y": 0,
```

```
                "width": 36,
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 2,
                        "name": "groupids.0",
                        "value": 4
                    },
                    {
                        "type": 1,
                        "name": "columns.0.name",
                        "value": "Host"
                    },
                    {
                        "type": 0,
                        "name": "columns.0.data",
                        "value": 2
                    },
                    {
                        "type": 1,
                        "name": "columns.0.base_color",
                        "value": "FFFFFF"
                    },
                    {
                        "type": 1,
                        "name": "columns.1.name",
                        "value": "CPU utilization in %"
                    },
                    {
                        "type": 0,
                        "name": "columns.1.data",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "columns.1.base_color",
                        "value": "4CAF50"
                    },
                    {
                        "type": 1,
                        "name": "columns.1.item",
                        "value": "CPU utilization"
                    },
                    {
                        "type": 0,
                        "name": "columns.1.display",
                        "value": 3
                    },
                    {
                        "type": 1,
                        "name": "columns.1.min",
                        "value": "0"
                    },
                    {
                        "type": 1,
                        "name": "columns.1.max",
                        "value": "100"
                    },
                    {
                        "type": 1,
                        "name": "columnsthresholds.1.color.0",
```

```
            "value": "FFFF00"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.0",
            "value": "50"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.1",
            "value": "FF8000"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.1",
            "value": "80"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.2",
            "value": "FF4000"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.2",
            "value": "90"
        },
        {
            "type": 1,
            "name": "columns.2.name",
            "value": "1m avg"
        },
        {
            "type": 0,
            "name": "columns.2.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.2.base_color",
            "value": "FFFFFF"
        },
        {
            "type": 1,
            "name": "columns.2.item",
            "value": "Load average (1m avg)"
        },
        {
            "type": 1,
            "name": "columns.3.name",
            "value": "5m avg"
        },
        {
            "type": 0,
            "name": "columns.3.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.3.base_color",
            "value": "FFFFFF"
        },
```

```
                        {
                            "type": 1,
                            "name": "columns.3.item",
                            "value": "Load average (5m avg)"
                        },
                        {
                            "type": 1,
                            "name": "columns.4.name",
                            "value": "15m avg"
                        },
                        {
                            "type": 0,
                            "name": "columns.4.data",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "columns.4.base_color",
                            "value": "FFFFFF"
                        },
                        {
                            "type": 1,
                            "name": "columns.4.item",
                            "value": "Load average (15m avg)"
                        },
                        {
                            "type": 1,
                            "name": "columns.5.name",
                            "value": "Processes"
                        },
                        {
                            "type": 0,
                            "name": "columns.5.data",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "columns.5.base_color",
                            "value": "FFFFFF"
                        },
                        {
                            "type": 1,
                            "name": "columns.5.item",
                            "value": "Number of processes"
                        },
                        {
                            "type": 0,
                            "name": "columns.5.decimal_places",
                            "value": 0
                        },
                        {
                            "type": 0,
                            "name": "column",
                            "value": 1
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
```

```
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

29 トップアイテム

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*Top items* ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成時または更新時には検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*Top items* ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

Top items ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - (デフォルト) 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。

パラメータ		type	name	value
ホストグループ		2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID の代わりに: `ABCDE._hostgroupids` - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを指定する場合、ホストグループパラメータは未設定であるか、指定したホストが属するホストグループが少なくとも 1 つ設定されている必要があります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホスト ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクタをホストのデータソースとして設定します; `ABCDE._hostids` - *Reference* パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストタグ				
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順を参照します。 パラメータの動作: - 必須 (*Host tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグ順を参照します。 パラメータの動作: - 必須 (*Host tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順を参照します。 パラメータの動作: - 必須 (*Host tags* を設定する場合) このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
レイアウト		0	layout	0 - *(デフォルト)* 横方向; 1 - 縦方向。
障害を表示		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制されていない; 2 - なし。
アイテム (下記参照)				

高度な設定

以下の高度な設定パラメータは、上位アイテムウィジェットでサポートされています。

パラメータ		タイプ	名前	値
ホストの並べ替え基準		0	host_ordering_order_by	0 - *(デフォルト)* ホスト名; 3 - アイテムの値。
	アイテムのパターン	1	host_ordering_item.0	アイテム名またはパターン（例: "*: 1 秒あたりに処理された * 値の数"）。 注: 複数のアイテムパターンを設定するには、各アイテムパターンに対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されたアイテムのパターンのみを設定してください。 パラメータの動作: - 並べ替え基準が「アイテムの値」に設定されている場合は必須
	順序	0	host_ordering_order	2 - *(デフォルト)* 上位 N 件; 3 - 下位 N 件。
	制限	0	host_ordering_limit	指定可能な値の範囲は 1～100 です。 デフォルト: 10。
アイテムの並べ替え基準		0	item_ordering_order_by	1 - ホスト; 2 - アイテム名; 3 - *(デフォルト)* アイテムの値。
	ホストのパターン	1	item_ordering_host.0	ホスト名またはパターン。 注: 複数のホストパターンを設定するには、各ホストパターンに対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメータの動作: - 並べ替え基準が「ホスト」に設定されている場合は必須
	順序	0	item_ordering_order	2 - *(デフォルト)* 上位 N 件; 3 - 下位 N 件。

パラメーター	タイプ	名前	値
制限	0	item_ordering_limit	指定可能な値の範囲は 1～100 です。 デフォルト: 10。

列

Note:
プロパティ名の最初の番号（例: columns.0.items.0、columns.0.item_tags_evaltype）は特定の列を表し、2 番目の番号がある場合は、設定されたエンティティ（例: アイテムパターン、タグ）を表します。

パラメーター	タイプ	名前	値
アイテムパターン	1	columns.0.items.0	アイテム の名前またはパターン（例: ”*: 1 秒あたりに処理された * 値の数”）。 注: 複数のアイテムパターンを設定するには、各アイテムパターンに対して、プロパティ名の 2 番目の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートに設定されたアイテムのパターンのみを設定してください。 パラメーターの動作: - 必須
アイテムタグ			
評価タイプ	0	columns.0.item_tags_evaltype	0 - (デフォルト) And/Or; 2 - Or。
タグ名	1	columns.0.item_tags.0.tag	任意の文字列値。 注: プロパティ名の 2 番目の番号は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - アイテムタグを設定する場合は必須
演算子	0	columns.0.item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の 2 番目の番号は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - アイテムタグを設定する場合は必須

パラメーター		タイプ	名前	値
	タグの値	1	columns.0.item_tags.0.value	任意の文字列値。 注: プロパティ名の 2 番目の番号は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - アイテムタグを設定する場合は必須
基本色		1	columns.0.base_color	16 進数のカラーコード（例: FF0000）。
値の表示形式		0	columns.0.display_value_as	1 - (デフォルト) 数値; 2 - テキスト。
表示		0	columns.0.display	1 - (デフォルト) そのまま; 2 - バー; 3 - インジケーター; 6 - スパークライン。
最小		1	columns.0.min	任意の数値。サフィックス（例: ”1d”、”2w”、”4K”、”8G”）がサポートされています。 パラメーターの動作: - 値の表示形式が「数値」に設定され、表示が「バー」または「インジケーター」に設定されている場合にサポート
最大		1	columns.0.max	任意の数値。サフィックス（例: ”1d”、”2w”、”4K”、”8G”）がサポートされています。 パラメーターの動作: - 値の表示形式が「数値」に設定され、表示が「バー」または「インジケーター」に設定されている場合にサポート
スパークライン	幅	0	columns.0.sparkline.width	指定可能な値は 0～10 です。 デフォルト: 1。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート
	塗りつぶし	0	columns.0.sparkline.fill	指定可能な値は 0～10 です。 デフォルト: 3。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート
	色	1	columns.0.sparkline.color	16 進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート

パラメーター		タイプ	名前	値
	期間	1	columns.0.sparkline.time_period	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメーターが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメーターで期間のみを設定できます。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート
	開始	1	columns.0.sparkline.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）による有効な時間文字列。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート - columns.0.sparkline.time_period.to が設定されている場合は必須
	終了	1	columns.0.sparkline.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）による有効な時間文字列。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート - columns.0.sparkline.time_period.from が設定されている場合は必須
	履歴データ	0	columns.0.sparkline.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメーターの動作: - 表示が「スパークライン」に設定されている場合にサポート
しきい値	色	1	columns.0.thresholds.color	16 進数のカラーコード（例: FF0000）。 パラメーターの動作: - 値の表示形式が「数値」に設定されている場合にサポート
	しきい値	1	columns.0.thresholds.threshold	任意の数値。 10、20、30、0.5、0.75 サフィックス（例: ”1d”、”2w”、”4K”、”8G”）がサポートされています。 パラメーターの動作: - 値の表示形式が「数値」に設定されている場合にサポート
ハイライト	ハイライト	1	columns.0.highlights.color	16 進数のカラーコード（例: FF0000）。 パラメーターの動作: - 値の表示形式が「テキスト」に設定されている場合にサポート

パラメーター		タイプ	名前	値
	しきい値	1	columns.0.highlights.0.pattern	任意の正規表現。 パラメーターの動作: - 値の表示形式が「テキスト」に設定されている場合にサポート
小数点以下の桁数詳細設定		0	columns.0.decimal_places	指定可能な値は 0～10 です。 デフォルト: 2。
集計関数		0	columns.0.aggregate_function	値の表示形式が「数値」に設定されている場合: 0 - *(デフォルト)* 使用しない; 1 - 最小; 2 - 最大; 3 - 平均; 4 - 件数; 5 - 合計; 6 - 最初; 7 - 最後。 値の表示形式が「テキスト」に設定されている場合: 0 - *(デフォルト)* 使用しない; 4 - 件数; 6 - 最初; 7 - 最後。
期間		1	columns.0.time_period.reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメーターが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメーターで期間のみを設定できます。 パラメーターの動作: - 集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、「最後」に設定されている場合にサポート
	開始	1	columns.0.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）による有効な時間文字列。 パラメーターの動作: - 期間が設定されておらず、集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、「最後」に設定されている場合にサポート - columns.0.time_period.to が設定されている場合は必須

パラメーター	タイプ	名前	値
終了	1	columns.0.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）による有効な時間文字列。 パラメーターの動作: - 期間が設定されておらず、集計関数が「最小」、「最大」、「平均」、「件数」、「合計」、「最初」、「最後」に設定されている場合にサポート - columns.0.time_period.from が設定されている場合は必須
履歴データ	0	columns.0.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメーターの動作: - 値の表示形式が「数値」に設定されている場合にサポート
集計	0	columns.0.aggregate_function	0 - (デフォルト) 各アイテム 1 - 結合。 パラメーターの動作: - 値の表示形式が「数値」に設定され、表示が「そのまま」、「バー」または「インジケーター」に設定されている場合にサポート
結合集計関数	0	columns.0.column_aggregate_function	1 - 最小; 2 - 最大; 3 - 平均; 4 - 件数; 5 - 合計 (デフォルト)。 パラメーターの動作: - 集計を設定する場合は必須
結合列名	1	columns.0.combined_column_name	任意の文字列値。 パラメーターの動作: - 集計を設定する場合は必須

例

次の例は、*Top items* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

トップアイテムウィジェットの設定

ホスト”10084” のデータと、名前が”component” のタグに値が”cpu” を含むアイテムのみを表示する トップアイテムウィジェットを設定します。さらに、ホストを上位に表示し、セルを表すために色付きのゲージバーを使用します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "topitems",
                        "name": "Top items",
                        "x": 0,
```

```
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "columns.0.items.0",
                                "value": "*"
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.item_tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.display",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

30 のトップトリガー

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*Top triggers* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*Top triggers* ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

Top triggers ウィジェットでは、次のパラメータがサポートされています。

パラメータ	タイプ	名前	値
更新間隔	0	rf_rate	0 - *(デフォルト)* 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ	2	groupids.0	ホストグループ ID。 注: 複数のホストグループを設定するには、各ホストグループに対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。

パラメータ		タイプ	名前	値
ホスト		3	hostids.0	ホスト ID。 注: 複数のホストを設定するには、各ホストに対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストを設定する場合、ホストグループパラメータは未設定にするか、設定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータはサポートされません。
障害		1	problem	障害のイベント名（大文字と小文字を区別しない、完全な名前またはその一部）。
深刻度		0	severities.0	トリガーの深刻度。 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 災害。 デフォルト: 空（すべて有効）。 注: 複数の値を設定するには、各値に対して、プロパティ名の番号を 1 つずつ増やしたダッシュボードウィジェットフィールドオブジェクトを作成します。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 障害タグを設定する場合は必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 障害タグを設定する場合は必須
	タグの値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 障害タグを設定する場合は必須

パラメータ		タイプ	名前	値
期間		1	time_period._reference	DASHBOARD._timeperiod - ダッシュボードの期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（*Reference* パラメータに"ABCDE" を設定）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定できます。
	From	1	time_period.from	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポート - time_period.to が設定されている場合は必須
	To	1	time_period.to	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合にサポート - time_period.from が設定されている場合は必須
トリガー数の上限		0	show_lines	指定可能な値の範囲は 1～100 です。 デフォルト: 10。

例

次の例は、上位トリガーウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

Top triggers ウィジェットの設定

ホストグループ「4」のトリガーのうち、各トリガーのすべての障害の数をカウントして上位 5 件を表示する *Top triggers* ウィジェットを設定します。このウィジェットは、「Warning」「Average」「High」「Disaster」の深刻度を持つトリガーのみを表示し、タグ名「scope」に「performance」「availability」「capacity」のいずれかの値を含むタグを持つ障害のみを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "toptriggers",
                        "name": "Top triggers",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
```

```
                    "view_mode": 0,
                    "fields": [
                        {
                            "type": 2,
                            "name": "groupids.0",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "severities.0",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "severities.1",
                            "value": 3
                        },
                        {
                            "type": 0,
                            "name": "severities.2",
                            "value": 4
                        },
                        {
                            "type": 0,
                            "name": "severities.3",
                            "value": 5
                        },
                        {
                            "type": 1,
                            "name": "tags.0.tag",
                            "value": "scope"
                        },
                        {
                            "type": 0,
                            "name": "tags.0.operator",
                            "value": 0
                        },
                        {
                            "type": 1,
                            "name": "tags.0.value",
                            "value": "performance"
                        },
                        {
                            "type": 1,
                            "name": "tags.1.tag",
                            "value": "scope"
                        },
                        {
                            "type": 0,
                            "name": "tags.1.operator",
                            "value": 0
                        },
                        {
                            "type": 1,
                            "name": "tags.1.value",
                            "value": "availability"
                        },
                        {
                            "type": 1,
                            "name": "tags.2.tag",
                            "value": "scope"
                        },
```

```
                            {
                                "type": 0,
                                "name": "tags.2.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.2.value",
                                "value": "capacity"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

31 トリガーの概要

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドでトリガーの概要 ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。トリガーの概要ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメータ

トリガー概要ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ		type	name	value
リフレッシュ間隔		0	rf_rate	0 - リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
表示		0	show	1 - *(デフォルト)* 最近の障害; 2 - すべて; 3 - 障害。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、プロパティ名の番号を増やして各ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: `ABCDE._hostgroupids` - 参照パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、プロパティ名の番号を増やして各ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを設定する場合、ホストグループパラメータは未設定であるか、設定したホストが属する少なくとも 1 つのホストグループで設定する必要があります。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクタをホストのデータソースとして設定します; `ABCDE._hostids` - 参照パラメータを"ABCDE" に設定した互換性のあるウィジェットをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順序を参照します。 パラメータの動作: - 障害タグを設定する場合は必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグ順序を参照します。 パラメータの動作: - 障害タグを設定する場合は必須
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグ順序を参照します。 パラメータの動作: - 障害タグを設定する場合は必須
抑制された障害を表示		0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
レイアウト		0	layout	0 - *(デフォルト)* 水平; 1 - 垂直。

例

次の例は、トリガーの概要ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

トリガーの概要ウィジェットの設定

”scope” という名前で値に”availability” を含むタグでトリガーされ、すべてのホストグループのトリガー状態を表示するトリガー概要ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Trigger overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "availability"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
```

```
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

32 URL

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドで*URL* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*URL* ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメータの動作を参照してください。

パラメータ

URL ウィジェットでサポートされているパラメータは以下の通りです。

パラメータ	タイプ	名前	値
リフレッシュ間隔	0	rf_rate	0 - *(デフォルト)* リフレッシュなし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
URL	1	url	有効な URL 文字列。 パラメータの動作: - 必須
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメータを"ABCDE" に設定）をホストのデータソースとして設定; DASHBOARD._hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定。 このパラメータはテンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

例

次の例は、*URL* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

URL ウィジェットの設定

Zabbix マニュアルのホームページを表示する *URL* ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
```

```
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "url",
                        "name": "URL",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "url",
                                "value": "https://www.zabbix.com/documentation/8.0/en"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

33 Web 監視

説明

これらのパラメータと、対応するダッシュボードウィジェットのフィールドオブジェクトで使用可能なプロパティ値を使用して、dashboard.create および dashboard.update メソッドの*Web 監視* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、ウィジェットを誤って作成または更新するリスクも生じます。*Web* 監視ウィジェットを正常に作成または更新するには、以下の表に示すパラメータの動作を参照してください。

パラメーター

Web 監視ウィジェットでサポートされているパラメーターは以下の通りです。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループの ID。 注: 複数のホストグループを設定するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループの ID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストグループのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホストグループの除外		2	exclude_groupids.0	ホストグループの ID。 注: 複数のホストグループを除外するには、各ホストグループごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホストの ID。 注: 複数のホストを設定するには、各ホストごとにプロパティ名の番号を増やしてダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを指定する場合、ホストグループパラメーターは未設定か、指定したホストが所属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメーター		type	name	value
	ホスト (ウィジェット/ダッシュボード)	1	hostids._reference	ホストの ID の代わりに: `DASHBOARD.hostids` - ダッシュボードのホストセレクターをホストのデータソースとして設定します; `ABCDE._hostids` - 互換性のあるウィジェット（*Reference* パラメーターを"ABCDE" に設定）をホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
シナリオタグ	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（シナリオタグを設定する場合）
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（シナリオタグを設定する場合）
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の番号はタグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 必須（シナリオタグを設定する場合）
メンテナンス中のホストを表示		0	maintenance	0 - 無効; 1 - *(デフォルト)* 有効。

パラメーター	type	name	value
Reference	1	reference	5 文字からなる任意の文字列値（例: ABCDE や JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

例

次の例は、*Web* 監視ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

Web 監視ウィジェットの設定

ホストグループ"4" のアクティブな Web 監視シナリオのステータス概要を表示する *Web* 監視ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "web",
                        "name": "Web monitoring",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

テンプレート

このクラスはテンプレートで動作するように設計されています。

オブジェクト参照:

- テンプレート
- テンプレートタグ

利用可能なメソッド:

- template.create - 新しいテンプレートを作成
- template.delete - テンプレートを削除
- template.get - テンプレートを取得
- template.massadd - 関連オブジェクトをテンプレートに追加
- template.massremove - 関連オブジェクトをテンプレートから削除
- template.update - テンプレートを更新

Template オブジェクト

次のオブジェクトは、templateAPI に直接関連しています。

テンプレート

テンプレートオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
templateid	ID	テンプレートの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
description	text	テンプレートの説明。
host	string	テンプレートの技術名。 プロパティの動作: - 作成操作では 必須
name	string	テンプレートの表示名。 デフォルト: host プロパティの値。
readme	text	ホストウィザード に表示する、テンプレート固有の設定手順です。 Markdown 形式をサポートします。

プロパティ	型	説明
uuid	string	インポートしたテンプレートを既存のテンプレートにリンクするために使用される、汎用一意識別子。指定されていない場合は自動生成されます。
vendor_name	string	テンプレートのベンダー名。 作成操作では、`vendor_name` と `vendor_version` の両方を設定するか、両方とも空のままにする必要があります。更新操作では、データベース内に値がある場合、`vendor_version` は空のままにできます。
vendor_version	string	テンプレートのベンダーバージョン。 作成操作では、`vendor_name` と `vendor_version` の両方を設定するか、両方とも空のままにする必要があります。更新操作では、データベース内に値がある場合、`vendor_name` は空のままにできます。
wizard_ready	integer	テンプレートをホストウィザードで選択可能にするかどうか。 使用可能な値: 0 - *(デフォルト)* 使用不可。 1 - 使用可能。

テンプレートタグ

テンプレートタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	テンプレートタグ名。 プロパティの動作: - 必須
value	string	テンプレートタグの値。 プロパティの動作: - 読み取り専用。
object	integer	タグの継承元オブジェクトのタイプ。 指定可能な値: 0 - テンプレート。 プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

template.create

説明

`object template.create(object/array templates)`

このメソッドを使用すると、新しいテンプレートを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するテンプレート。

標準のテンプレートプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
groups	object/array	テンプレートを追加するテンプレートグループ。 テンプレートグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
tags	object/array	テンプレートタグ。
templates	object/array	テンプレートにリンクするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。
macros	object/array	テンプレート用に作成するユーザーマクロ。

戻り値

(object) templateids プロパティの下で作成されたテンプレートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートの順序と一致します。

例

テンプレートの作成

タグ付きのテンプレートを作成し、このテンプレートに 2 つのテンプレートをリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.create",
    "params": {
        "host": "Linux template",
        "groups": {
            "groupid": 1
        },
        "templates": [
            {
                "templateid": "11115"
            },
            {
                "templateid": "11116"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::create() in *ui/include/classes/api/services/CTemplate.php*.

template.delete

説明

`object template.delete(array templateIds)`

このメソッドを使用すると、テンプレートを削除できます。

テンプレートを削除すると、そのテンプレートに含まれるすべてのエンティティ（アイテム、トリガー、グラフなど）も、リンクされているホストまたはテンプレートから削除されます。
これらのエンティティを保持するには、まず`host.update`、`host.massremove`、`template.update`、または`template.massremove`メソッドを使用して、関連するホストまたはテンプレートからテンプレートのリンクを解除してください。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。
> このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。
> 詳細はUser roles を参照してください。

パラメーター

(`array`) 削除するテンプレートの ID。

戻り値

(`object`) `templateids` プロパティの下にある削除されたテンプレートの ID を含むオブジェクトを返します。

例

複数のテンプレートの削除

2 つのテンプレートを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::delete() in *ui/include/classes/api/services/CTemplate.php*.

template.get

説明

`integer/array template.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従ってテンプレートを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
templateids	ID/array	指定したテンプレート ID を持つテンプレートのみを返します。
groupids	ID/array	指定したテンプレートグループに属するテンプレートのみを返します。
parentTemplateids	ID/array	指定したテンプレートがリンクしているテンプレートのみを返します。
hostids	ID/array	指定したホスト/テンプレートにリンクされているテンプレートのみを返します。
graphids	ID/array	指定したグラフを含むテンプレートのみを返します。
itemids	ID/array	指定したアイテムを含むテンプレートのみを返します。
triggerids	ID/array	指定したトリガーを含むテンプレートのみを返します。
with_items	flag	アイテムを持つテンプレートのみを返します。
with_triggers	flag	トリガーを持つテンプレートのみを返します。
with_graphs	flag	グラフを持つテンプレートのみを返します。
with_httptests	flag	Web シナリオを持つテンプレートのみを返します。
evaltype	integer	タグの評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	object/array	指定したタグを持つテンプレートのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのテンプレートを返します。 使用可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
inheritedTags	boolean	指定した `tags` を、リンクされたテンプレートにも持つテンプレートを返します。 使用可能な値: `true` - リンクされたテンプレートも指定したタグを持っている必要があります; `false` - *(デフォルト)* リンクされたテンプレートのタグは無視されます。
selectInheritedTags	query	リンクされたテンプレート上のタグを含む`inheritedTags`プロパティを返します。
selectTags	query	テンプレートタグを`tags`プロパティで返します。
selectDiscoveryRules	query	テンプレートの LLD ルールを含む`discoveryRules`プロパティを返します。 `count` をサポートします。
selectHosts	query	テンプレートにリンクされているホストを`hosts`プロパティで返します。 `count` をサポートします。
selectTemplateGroups	query	テンプレートが属するテンプレートグループを`templategroups`プロパティで返します。

パラメータ	型	説明
selectTemplates	query	指定したテンプレートがリンクしているテンプレートを`templates`プロパティで返します。 `count` をサポートします。
selectParentTemplates	query	指定したテンプレートにリンクされているテンプレートを`parentTemplates` プロパティで返します。 `count` をサポートします。
selectHttpTests	query	テンプレートの Web シナリオを`httpTests`プロパティで返します。 `count` をサポートします。
selectItems	query	テンプレートのアイテムを`items`プロパティで返します。 `count` をサポートします。
selectTriggers	query	テンプレートのトリガーを`triggers`プロパティで返します。 `count` をサポートします。
selectGraphs	query	テンプレートのグラフを`graphs`プロパティで返します。 `count` をサポートします。
selectMacros	query	テンプレートのマクロを`macros`プロパティで返します。
selectDashboards	query	テンプレートのダッシュボードを`dashboards`プロパティで返します。 `count` をサポートします。
selectValueMaps	query	テンプレートの値マップを含む`valuemaps`プロパティを返します。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectTemplates` - 結果は `name` でソートされます; `selectHosts` - `host` でソートされます; `selectParentTemplates` - `host` でソートされます; `selectItems` - `name` でソートされます; `selectDiscoveryRules` - `name` でソートされます; `selectTriggers` - `description` でソートされます; `selectGraphs` - `name` でソートされます; `selectDashboards` - `name` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 使用可能な値: `hostid`, `host`, `name`, `status`.
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	テンプレートの低レベルディスカバリを `discoveries` プロパティで返します。 `count` をサポートします。 このクエリは非推奨です。代わりに `selectDiscoveryRules` を使用してください。

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

テンプレート名によるテンプレートの取得

"Linux by Zabbix agent" と"Windows by Zabbix agent" という 2 つのテンプレートに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux by Zabbix agent",
            "flags": "0",
            "templateid": "10001",
            "description": " これは公式の Linux テンプレートです。Zabbix agent 8.0 以降が必要です。\r\n\r\nファイル
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "f8f7908280354f2abeed07dc788c3747",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## 概要\r\n\r\nこれは公式の Linux テンプレートです。Zabbix agent 8.0 以降が必要です。\r\n\r\
        },
        {
            "proxyid": "0",
            "host": "Windows by Zabbix agent",
```

```
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Windows by Zabbix agent",
            "flags": "0",
            "templateid": "10081",
            "description": " これは公式の Windows テンプレートです。Zabbix agent 8.0 以降が必要です。\r\n\r\nこのテ
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "13b06904a6bf41cbb795e3193d896340",
            "vendor_name": "Zabbix",
            "vendor_version": "8.0-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## 概要\r\n\r\nこれは公式の Windows テンプレートです。Zabbix agent 8.0 以降が必要です。"
        }
    ],
    "id": 1
}
```


テンプレートグループの取得

テンプレート”Linux by Zabbix エージェント” が所属するテンプレートグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Response:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
```

```
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートによるホストの取得

”10001”(*Linux by Zabbix* エージェント) テンプレートがリンクされているホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートタグによる検索

タグ”host-name” が”{HOST.NAME}” と等しいテンプレートを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
```

```
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- テンプレートグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース

ソース

CTemplate::get() in *ui/include/classes/api/services/CTemplate.php*.

template.massadd

説明

`object template.massadd(object parameters)`

このメソッドを使用すると、指定したテンプレートに関連する複数のオブジェクトを同時に追加できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 更新するテンプレートの ID と、テンプレートに追加するオブジェクトを含むパラメータ。

このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
templates	object/array	更新対象のテンプレート。 テンプレートでは、`templateid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	指定したテンプレートに追加するテンプレートグループ。 テンプレートグループでは、`groupid` プロパティのみを定義する必要があります。

パラメータ	型	説明
macros	object/array	指定したテンプレート用に作成するユーザーマクロ。
templates_link	object/array	指定したテンプレートにリンクするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートにグループをリンクする

テンプレートグループ"2" を 2 つのテンプレートに追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

2 つのテンプレートを 1 つのテンプレートにリンクする

テンプレート"10106" と"10104" をテンプレート"10073" にリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
```

```
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

関連項目

- template.update
- ホスト
- テンプレートグループ
- ユーザーマクロ

ソース

CTemplate::massAdd() in *ui/include/classes/api/services/CTemplate.php*.

template.massremove

説明

`object template.massremove(object parameters)`

このメソッドを使用すると、複数のテンプレートから関連オブジェクトを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新するテンプレートの ID と、削除するオブジェクトを含むパラメータ。

パラメータ	型	説明
templateids	ID/array	更新するテンプレートの ID。 パラメータの動作: - 必須
groupids	ID/array	指定したテンプレートの削除元となるテンプレートグループの ID。
macros	string/array	指定したテンプレートから削除するユーザーマクロの ID。
templateids_clear	ID/array	指定したテンプレートからリンク解除してクリアするテンプレートの ID（上流）。
templateids_link	ID/array	指定したテンプレートからリンク解除するテンプレートの ID（上流）。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートをグループから削除する

2 つのテンプレートをグループ”2” から削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

ホストからテンプレートのリンクを解除

テンプレート”10085” からテンプレート”10106” と”10104” のリンクを解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

関連項目

- template.update
- ユーザーマクロ

ソース

CTemplate::massRemove() in *ui/include/classes/api/services/CTemplate.php*.

template.update

説明

`object template.update(object/array templates)`

このメソッドを使用すると、既存のテンプレートを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するテンプレートのプロパティ。

各テンプレートでは `templateid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、その他は変更されません。

標準のテンプレートプロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	テンプレートが所属する現在のテンプレートグループを置き換えるテンプレートグループ。 テンプレートグループでは `groupid` プロパティのみを定義する必要があります。
tags	object/array	現在のテンプレートタグを置き換えるテンプレートタグ。
macros	object/array	指定したテンプレート上の現在のユーザーマクロを置き換えるユーザーマクロ。
templates	object/array	現在リンクされているテンプレートを置き換えるテンプレート。渡されなかったテンプレートはリンク解除のみされます。 テンプレートでは `templateid` プロパティのみを定義する必要があります。
templates_clear	object/array	指定したテンプレートからリンク解除およびクリアするテンプレート。 テンプレートでは `templateid` プロパティのみを定義する必要があります。

戻り値

(`object`) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

標準テンプレートプロパティの変更

テンプレートの技術名を「Linux by Zabbix agent Custom」、表示名を「My template」、テンプレートの説明を更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "host": "Linux by Zabbix agent Custom",
        "name": "My template",
        "description": "This is a custom Linux template."
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートグループの更新

指定したテンプレートのすべてのテンプレートグループを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

複数のテンプレートグループの更新

複数のテンプレートのすべてのテンプレートグループを異なるものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10086",
            "groups": [
                {
                    "groupid": "24"
                }
            ]
        },
        {
            "templateid": "10087",
            "groups": [
                {
                    "groupid": "1"
                },
```

```
                {
                    "groupid": "12"
                }
            ]
        },
        {
            "templateid": "10088",
            "groups": [
                {
                    "groupid": "1"
                },
                {
                    "groupid": "12"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

テンプレートタグの更新

すべてのテンプレートタグを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートマクロの更新

すべてのテンプレートマクロを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

複数のテンプレートマクロの更新

複数のテンプレートで、すべてのユーザーマクロを指定したユーザーマクロに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Timeout after which agent is considered unavailable. Works only fo
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Timeout after which agent is considered unavailable. Works only fo
                    }
                ]
            }
        ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

テンプレートのリンクテンプレートの更新

指定したテンプレートからすべてのテンプレートのリンクを解除し (クリアせず)、別のテンプレートをリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10001"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

複数のテンプレートからリンクされたテンプレートをクリアする

指定したテンプレートから特定のリンクされたテンプレートのリンクを解除し、クリアします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
```

```
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::update() in *ui/include/classes/api/services/CTemplate.php*.

テンプレートグループ

このクラスはテンプレートグループを操作するために設計されています。

オブジェクト参照:

- テンプレートグループ

利用可能なメソッド:

- templategroup.create - 新しいテンプレートグループを作成
- templategroup.delete - テンプレートグループを削除
- templategroup.get - テンプレートグループを取得
- templategroup.massadd - 関連オブジェクトをテンプレートグループに追加
- templategroup.massremove - 関連オブジェクトをテンプレートグループから削除
- templategroup.propagate - 権限をテンプレートグループのサブグループに伝播
- templategroup.update - テンプレートグループを更新

Template group オブジェクト

次のオブジェクトは、`templategroup`API に直接関連しています。

テンプレートグループ

テンプレートグループオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
groupid	ID	テンプレートグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	テンプレートグループの名前。 プロパティの動作: - 作成操作では必須
uuid	string	汎用一意識別子。インポートされたテンプレートグループを、すでに存在するテンプレートグループにリンクするために使用されます。指定されていない場合は自動生成されます。

templategroup.create

説明

`object templategroup.create(object/array templateGroups)`

このメソッドを使用すると、新しいテンプレートグループを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するテンプレートグループ。このメソッドは、標準のテンプレートグループプロパティを持つテンプレートグループを受け付けます。

戻り値

(object) は、groupids プロパティに指定された、作成されたテンプレートグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートグループの順序と一致します。

例

テンプレートグループの作成

"Templates/Databases" という名前のテンプレートグループを作成します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::create() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

説明

`object templategroup.delete(array templateGroupIds)`

このメソッドを使用すると、テンプレートグループを削除できます。

テンプレートグループに、そのグループにのみ属するテンプレートが含まれている場合、そのテンプレートグループは削除できません。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するテンプレートグループの ID

戻り値

(object) groupids プロパティの下に削除されたテンプレートグループの ID を含むオブジェクトを返します。

例

複数のテンプレートグループの削除

二つのテンプレートグループを削除する。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::delete() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

説明

`integer/array templategroup.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従ってテンプレートグループを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細については、User rolesを参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
graphids	ID/array	指定したグラフを含むテンプレートを持つテンプレートグループのみを返します。
groupids	ID/array	指定したテンプレートグループ ID を持つテンプレートグループのみを返します。
templateids	ID/array	指定したテンプレートを含むテンプレートグループのみを返します。
triggerids	ID/array	指定したトリガーを含むテンプレートを持つテンプレートグループのみを返します。

Parameter	Type	Description
with_graphs	boolean	グラフを含むテンプレートを持つテンプレートグループのみを返します。
with_graph_prototypes	boolean	グラフプロトタイプを含むテンプレートを持つテンプレートグループのみを返します。
with_httptests	boolean	Web 監視を含むテンプレートを持つテンプレートグループのみを返します。
with_items	boolean	アイテムを含むテンプレートを持つテンプレートグループのみを返します。 `with_simple_graph_items` パラメータより優先されます。
with_item_prototypes	boolean	アイテムプロトタイプを含むテンプレートを持つテンプレートグループのみを返します。 `with_simple_graph_item_prototypes` パラメータより優先されます。
with_simple_graph_item_prototypes	boolean	作成が有効で、かつ数値型の情報タイプを持つアイテムプロトタイプを含むテンプレートを持つテンプレートグループのみを返します。
with_simple_graph_items	boolean	数値アイテムを含むテンプレートを持つテンプレートグループのみを返します。
with_templates	boolean	テンプレートを含むテンプレートグループのみを返します。
with_triggers	boolean	トリガーを含むテンプレートを持つテンプレートグループのみを返します。
selectTemplates	query	テンプレートグループに属するテンプレートを含む`templates` プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectTemplates` - 結果は `template` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `groupid`, `name`。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用された場合は、取得したオブジェクトの数

例

名前によるデータの取得

"Templates/Databases" と "Templates/Modules" という名前の二つのテンプレートグループについて、全てのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
```

```
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

参照

- Template

ソース

CTemplateGroup::get() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

説明

`object templategroup.massadd(object parameters)`

このメソッドを使用すると、指定したすべてのテンプレートグループに、関連する複数のオブジェクトを同時に追加できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 更新するテンプレートグループの ID と、すべてのテンプレートグループに追加するオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
groups	object/array	更新するテンプレートグループ。 テンプレートグループでは、`groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須

パラメータ	Type	説明
templates	object/array	すべてのテンプレートグループに追加するテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。 パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループへテンプレートの追加

ID12 と 13 のテンプレートグループへ二つのテンプレートを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

参照

- Template

ソース

CTemplateGroup::massAdd() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

説明

`object templategroup.massremove(object parameters)`

このメソッドを使用すると、複数のテンプレートグループから関連オブジェクトを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新するテンプレートグループの ID と、削除する必要があるオブジェクトを含むパラメータ。

パラメータ	型	説明
groupids	ID/array	更新するテンプレートグループの ID。 パラメータの動作: - 必須
templateids	ID/array	すべてのテンプレートグループから削除するテンプレートの ID。 パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループからテンプレートの削除

指定されたテンプレートグループから 2 つのテンプレートを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::massRemove() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

説明

`object templategroup.propagate(object parameters)`

このメソッドを使用すると、すべてのテンプレートグループのサブグループに権限を適用できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 希望する出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
groups	object/array	伝播するテンプレートグループ。 テンプレートグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
permissions	boolean	権限を伝播する必要がある場合は、true に設定します。 パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下に伝搬されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループの権限をそのサブグループへ伝搬する

テンプレートグループの権限をそのサブグループへ伝搬します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
```

```
            "15",
        ]
    },
    "id": 1
}
```

参照

- templategroup.update
- templategroup.massadd
- Template

ソース

CTemplateGroup::propagate() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.update

説明

`object templategroup.update(object/array templateGroups)`

このメソッドを使用すると、既存のテンプレートグループを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

更新する（object/array）テンプレートグループのプロパティ。

各テンプレートグループでは groupid プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、それ以外は変更されません。

戻り値

（object）groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループの名前の変更

テンプレートグループの名前を”Templates/Databases” へ変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::update() in *ui/include/classes/api/services/CTemplateGroup.php*.

テンプレートダッシュボード

このクラスは、テンプレートダッシュボードで動作するように設計されています。

オブジェクト参照:

- テンプレートダッシュボード
- テンプレートダッシュボードページ
    - テンプレートダッシュボードウィジェット
        * テンプレートダッシュボードウィジェットフィールド

利用可能なメソッド:

- templatedashboard.create - 新しいテンプレートダッシュボードを作成
- templatedashboard.delete - テンプレートダッシュボードを削除
- templatedashboard.get - テンプレートダッシュボードを取得
- templatedashboard.update - テンプレートダッシュボードを更新

Template dashboard オブジェクト

次のオブジェクトは、`templatedashboard`API に直接関連しています。

テンプレートダッシュボード

テンプレートダッシュボードオブジェクトには、次のプロパティがあります。

Property	Type	Description
dashboardid	ID	テンプレートダッシュボードの ID。 Property behavior: - *read-only* - 更新操作では *required*
name	string	テンプレートダッシュボードの名前。 Property behavior: - 作成操作では *required*
templateid	ID	ダッシュボードが属するテンプレートの ID。 Property behavior: - *constant* - 作成操作では *required*
display_period	integer	デフォルトのスライドショー間隔（秒）。 指定可能な値: 10, 30, 60, 120, 600, 1800, 3600. デフォルト: 30.
auto_start	integer	スライドショーを自動開始します。 指定可能な値: 0 - スライドショーを自動開始しない; 1 - *(default)* スライドショーを自動開始する。
uuid	string	インポートされたテンプレートダッシュボードを既存のものにリンクするために使用される、ユニバーサル一意識別子。指定されていない場合は自動生成されます。

テンプレートダッシュボードページ

テンプレートダッシュボードページオブジェクトには、次のプロパティがあります。

Property	Type	Description
dashboard_pageid	ID	ダッシュボードページの ID。 Property behavior: - *read-only*
name	string	ダッシュボードページ名。 Default: empty string.
display_period	integer	ダッシュボードのスライドショー間隔（秒）。 Possible values: 0, 10, 30, 60, 120, 600, 1800, 3600. Default: 0 (will use the default slideshow interval).
widgets	array	テンプレートダッシュボードウィジェットオブジェクトの配列。

テンプレートダッシュボードウィジェット

テンプレートダッシュボードウィジェットオブジェクトには、以下のプロパティがあります。

Property	Type	Description
widgetid	ID	ダッシュボードウィジェットの ID。 プロパティの動作: - 読み取り専用
type	string	ダッシュボードウィジェットのタイプ。 指定可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリのステータス; favgraphs - お気に入りのグラフ; favmaps - お気に入りのマップ; gauge - ゲージ; graph - グラフ（クラシック）; graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホストの可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemnavigator - アイテムナビゲーター; item - アイテムの値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; plaintext - プレーンテキスト; problemhosts - 障害ホスト; problems - 障害; problemsbysv - 深刻度ごとの障害; scatterplot - 散布図; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - 上位ホスト; topitems - 上位アイテム; toptriggers - 上位トリガー; trigover - トリガーの概要; url - URL; web - Web 監視。 プロパティの動作: - 必須

Property	Type	Description
name	string	カスタムウィジェット名。
x	integer	ダッシュボードの左端からの水平方向の位置。 指定可能な値の範囲は 0 から 71 です。
y	integer	ダッシュボードの上端からの垂直方向の位置。 指定可能な値の範囲は 0 から 63 です。
width	integer	ウィジェットの幅。 指定可能な値の範囲は 1 から 72 です。
height	integer	ウィジェットの高さ。 指定可能な値の範囲は 1 から 64 です。
view_mode	integer	ウィジェットの表示モード。 指定可能な値: 0 - *(デフォルト)* デフォルトのウィジェット表示; 1 - ヘッダーを非表示;
fields	array	テンプレートダッシュボードウィジェットフィールドオブジェクトの配列。

テンプレートダッシュボードウィジェットフィールド

テンプレートダッシュボードウィジェットフィールドオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
type	integer	ウィジェットフィールドのタイプ。 使用可能な値: 0 - 整数; 1 - 文字列; 4 - アイテム; 5 - アイテムプロトタイプ; 6 - グラフ; 7 - グラフプロトタイプ; 8 - マップ; 9 - サービス; 10 - SLA; 11 - ユーザー; 12 - アクション; 13 - メディアタイプ。 プロパティの動作: - 必須
name	string	ウィジェットフィールド名。 使用可能な値: ダッシュボードウィジェットフィールドを参照してください。なお、一部のホスト関連パラメータ（例: 障害ウィジェットの ホストグループ、除外するホストグループ、ホスト、ホストの可用性ウィジェットの ホストグループなど）は、テンプレートダッシュボード上でウィジェットを設定する場合には使用できません。これは、テンプレートダッシュボードが、そのテンプレートにリンクされているホストのデータのみを表示するためです。 プロパティの動作: - 必須

プロパティ	型	説明
value	mixed	タイプに応じたウィジェットフィールドの値。 使用可能な値: ダッシュボードウィジェットフィールドを参照してください。なお、一部のホスト関連パラメータ（例: 障害ウィジェットの ホストグループ、除外するホストグループ、ホスト、ホストの可用性ウィジェットの ホストグループなど）は、テンプレートダッシュボード上でウィジェットを設定する場合には使用できません。これは、テンプレートダッシュボードが、そのテンプレートにリンクされているホストのデータのみを表示するためです。 プロパティの動作: - 必須

templatedashboard.create

説明

`object templatedashboard.create(object/array templateDashboards)`

このメソッドを使用すると、新しいテンプレートダッシュボードを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するテンプレートダッシュボード。

標準のテンプレートダッシュボードのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
pages	array	ダッシュボード用に作成するテンプレートダッシュボードページ。ダッシュボードページは、指定された順序と同じ順序で並べられます。 パラメータの動作: - 必須

戻り値

(object) dashboardids プロパティの下に作成されたテンプレートダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートダッシュボードの順序と一致します。

例

テンプレートダッシュボードの作成

1 つのダッシュボードページに 1 つのグラフウィジェットを持つ「Graphs」という名前のテンプレートダッシュボードを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.create",
    "params": {
        "templateid": "10318",
        "name": "Graphs",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "x": 0,
```

```
                        "y": 0,
                        "width": 12,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 6,
                                "name": "graphid",
                                "value": "1123"
                            }
                        ]
                    }
                ]

            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

関連項目

- テンプレートダッシュボードページ
- テンプレートダッシュボードウィジェット
- テンプレートダッシュボードウィジェットフィールド

ソース

CTemplateDashboard::create() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

説明

`object templatedashboard.delete(array templateDashboardIds)`

このメソッドを使用すると、テンプレートダッシュボードを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するテンプレートダッシュボードの ID。

戻り値

(object) dashboardids プロパティの下にある削除されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

複数のテンプレートダッシュボードの削除

2 つのテンプレートダッシュボードを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

ソース

CTemplateDashboard::delete() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

説明

`integer/array templatedashboard.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってテンプレートダッシュボードを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 希望する出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dashboardids	ID/array	指定した ID を持つテンプレートダッシュボードのみを返します。
templateids	ID/array	指定したテンプレートに属するテンプレートダッシュボードのみを返します。
selectPages	query	正しい順序で並べられたテンプレートダッシュボードページを含むpages プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dashboardid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

パラメータ	型	説明
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

テンプレートダッシュボードの取得

指定したテンプレートのウィジェット付きテンプレートダッシュボードをすべて取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "23",
            "name": "Docker overview",
            "templateid": "10001",
            "display_period": "30",
            "auto_start": "1",
            "uuid": "6dfcbe0bc5ad400ea9c1c2dd7649282f",
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "220",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1125"
                                }
                            ]
                        },
                        {
```

```
                "widgetid": "221",
                "type": "graph",
                "name": "",
                "x": "12",
                "y": "0",
                "width": "36",
                "height": "5",
                "view_mode": "0",
                "fields": [
                    {
                        "type": "6",
                        "name": "graphid",
                        "value": "1129"
                    }
                ]
            },
            {
                "widgetid": "222",
                "type": "graph",
                "name": "",
                "x": "0",
                "y": "5",
                "width": "36",
                "height": "5",
                "view_mode": "0",
                "fields": [
                    {
                        "type": "6",
                        "name": "graphid",
                        "value": "1128"
                    }
                ]
            },
            {
                "widgetid": "223",
                "type": "graph",
                "name": "",
                "x": "12",
                "y": "5",
                "width": "36",
                "height": "5",
                "view_mode": "0",
                "fields": [
                    {
                        "type": "6",
                        "name": "graphid",
                        "value": "1126"
                    }
                ]
            },
            {
                "widgetid": "224",
                "type": "graph",
                "name": "",
                "x": "0",
                "y": "10",
                "width": "36",
                "height": "5",
                "view_mode": "0",
                "fields": [
                    {
                        "type": "6",
```

```
                                    "name": "graphid",
                                    "value": "1127"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- テンプレートダッシュボードページ
- テンプレートダッシュボードウィジェット
- テンプレートダッシュボードウィジェットフィールド

ソース

CTemplateDashboard::get() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

説明

`object templatedashboard.update(object/array templateDashboards)`

このメソッドを使用すると、既存のテンプレートダッシュボードを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するテンプレートダッシュボードのプロパティ。

各ダッシュボードに対して `dashboardid` プロパティを指定する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新されます。

また、このメソッドは標準のテンプレートダッシュボードのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
pages	array	既存のダッシュボードページを置き換えるテンプレートダッシュボードページ。 ダッシュボードページは `dashboard_pageid` プロパティによって更新されます。`dashboard_pageid` プロパティを持たないオブジェクトについては新しいダッシュボードページが作成され、再利用されなかった既存のダッシュボードページは削除されます。ダッシュボードページは指定された順序と同じ順序で並べられます。ダッシュボードページの指定されたプロパティのみが更新されます。pages プロパティには少なくとも 1 つのダッシュボードページオブジェクトが必要です。

戻り値

(`object`) `dashboardids` プロパティの下で更新されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

テンプレートダッシュボードの名前変更

テンプレートダッシュボードの名前を「Performance graphs」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

テンプレートダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換え、3 番目のページとして新しいページを追加します。他のすべてのダッシュボードページを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
```

```
        ]
    },
    "id": 1
}
```

関連項目

- テンプレートダッシュボードウィジェット
- テンプレートダッシュボードウィジェットフィールド

ソース

CTemplateDashboard::update() in *ui/include/classes/api/services/CTemplateDashboard.php*.

ディスカバリチェック

このクラスは、ディスカバリチェックで動作するように設計されています。

オブジェクト参照:

- ディスカバリチェック

利用可能なメソッド:

- dcheck.get - ディスカバリチェックを取得

Discovery check オブジェクト

次のオブジェクトは、dcheckAPI に直接関連するものです。

ディスカバリチェック

ディスカバリチェックオブジェクトは、ネットワークディスカバリルールによって実行される特定のチェックを定義します。以下のプロパティがあります。

Property	Type	Description
dcheckid	ID	ディスカバリチェックの ID。
druleid	ID	このチェックが属するディスカバリルールの ID。
key_	string	アイテムキー（type が”Zabbix agent” に設定されている場合）または SNMP OID（type が”SNMPv1 agent”、”SNMPv2 agent”、または”SNMPv3 agent” に設定されている場合）。 プロパティの動作: - type が”Zabbix agent”、”SNMPv1 agent”、”SNMPv2 agent”、または”SNMPv3 agent” に設定されている場合は 必須
ports	string	チェックする 1 つまたは複数のポート範囲。カンマで区切ります。 デフォルト: 0。 プロパティの動作: - type が”SSH” (0)、”LDAP” (1)、”SMTP” (2)、”FTP” (3)、”HTTP” (4)、”POP” (5)、”NNTP” (6)、”IMAP” (7)、”TCP” (8)、”Zabbix agent” (9)、”SNMPv1 agent” (10)、”SNMPv2 agent” (11)、”SNMPv3 agent” (13)、”HTTPS” (14)、または”Telnet” (15) に設定されている場合に サポート
snmp_community	string	SNMP コミュニティ。 プロパティの動作: - type が”SNMPv1 agent” または”SNMPv2 agent” に設定されている場合は 必須
snmpv3_authpassphrase	string	認証パスフレーズ。 プロパティの動作: - type が”SNMPv3 agent” に設定され、snmpv3_securitylevel が”authNoPriv” または”authPriv” に設定されている場合に サポート

Property	Type	Description
snmpv3_authprotocol	integer	認証プロトコル。 使用可能な値: 0 - *(デフォルト)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512。 プロパティの動作: - `type` が”SNMPv3 agent” に設定され、`snmpv3_securitylevel` が”authNoPriv” または”authPriv” に設定されている場合に サポート
snmpv3_contextname	string	SNMPv3 コンテキスト名。 プロパティの動作: - `type` が”SNMPv3 agent” に設定されている場合に サポート
snmpv3_privpassphrase	string	プライバシーパスフレーズ。 プロパティの動作: - `type` が”SNMPv3 agent” に設定され、`snmpv3_securitylevel` が”authPriv” に設定されている場合に サポート
snmpv3_privprotocol	integer	プライバシープロトコル。 使用可能な値: 0 - *(デフォルト)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C。 プロパティの動作: - `type` が”SNMPv3 agent” に設定され、`snmpv3_securitylevel` が”authPriv” に設定されている場合に サポート
snmpv3_securitylevel	string	セキュリティレベル。 使用可能な値: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv。 プロパティの動作: - `type` が”SNMPv3 agent” に設定されている場合に サポート
snmpv3_securityname	string	セキュリティ名。 プロパティの動作: - `type` が”SNMPv3 agent” に設定されている場合に サポート

Property	Type	Description
type	integer	チェックのタイプ。 使用可能な値: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet。 プロパティの動作: - 必須
uniq	integer	このチェックをデバイス一意性の判定基準として使用するかどうか。ディスカバリルールごとに設定できる一意チェックは 1 つだけです。 使用可能な値: 0 - *(デフォルト)* このチェックを一意性の判定基準として使用しない; 1 - このチェックを一意性の判定基準として使用する。 プロパティの動作: - type が”Zabbix agent”、”SNMPv1 agent”、”SNMPv2 agent”、または”SNMPv3 agent” に設定されている場合に サポート
host_source	integer	ホスト名のソース。 使用可能な値: 1 - *(デフォルト)* DNS; 2 - IP; 3 - このチェックのディスカバリ値。
name_source	integer	表示名のソース。 使用可能な値: 0 - *(デフォルト)* 指定なし; 1 - DNS; 2 - IP; 3 - このチェックのディスカバリ値。
allow_redirect	integer	ICMP ping の対象が別の IP アドレスから応答する状況を許可します。 使用可能な値: 0 - *(デフォルト)* リダイレクトされた応答を、対象ホストがダウンしている場合として扱う（失敗）; 1 - リダイレクトされた応答を、対象ホストが稼働している場合として扱う（成功）。 プロパティの動作: - type が”ICMP ping” に設定されている場合に サポート

dcheck.get

説明

`integer/array dcheck.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってディスカバリチェックを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dcheckids	ID/array	指定された ID を持つディスカバリチェックのみを返します。
druleids	ID/array	指定されたディスカバリルールに属するディスカバリチェックのみを返します。
dserviceids	ID/array	指定された検出済みサービスを検出したディスカバリチェックのみを返します。
selectDRules	query	ディスカバリチェックに関連するディスカバリルールを返します。
sortfield	string/array	指定されたプロパティで結果を並べ替えます。 指定可能な値: dcheckid, druleid。
countOutput	boolean	これらのパラメータについては、リファレンスコメントを参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ディスカバリルールのディスカバリチェックを取得

ディスカバリルール"6" によって使用されるすべてのディスカバリチェックを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "dcheckid": "6",
            "druleid": "4",
            "type": "3",
            "key_": "",
            "snmp_community": "",
            "ports": "21",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "snmpv3_contextname": "",
            "host_source": "1",
            "name_source": "0",
            "allow_redirect": "0"
        }
    ],
    "id": 1
}
```


ソース

CDCheck::get() in *ui/include/classes/api/services/CDCheck.php*.

ディスカバリルール

このクラスは、ネットワークディスカバリルールで動作するように設計されています。

> **Note:**
> この API は、ネットワークディスカバリルールで動作することを目的としています。
> ローレベルディスカバリルールについては、LLD rule API を参照してください。

オブジェクト参照:

- ディスカバリルール

利用可能なメソッド:

- drule.create - 新しいネットワークディスカバリルールを作成
- drule.delete - ネットワークディスカバリルールを削除
- drule.get - ネットワークディスカバリルールを取得
- drule.update - ネットワークディスカバリルールを更新

Discovery rule オブジェクト

次のオブジェクトは、druleAPI に直接関連しています。

ディスカバリルール

ディスカバリルールオブジェクトは、ネットワークディスカバリルールを定義します。以下のプロパティがあります。

Property	Type	Description
druleid	ID	ディスカバリルールの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須

Property	Type	Description
iprange	string	チェックする 1 つ以上の IP 範囲。複数指定する場合はコンマで区切ります。 IP 範囲でサポートされる形式の詳細については、ネットワークディスカバリの設定セクションを参照してください。 プロパティの動作: - 作成操作では必須
name	string	ディスカバリルールの名前。 プロパティの動作: - 作成操作では必須
delay	string	ディスカバリルールの実行間隔。 秒数（例: 1、60、300）、時間のサフィックス付きの値（例: 30s、1m、2h、1d）、またはユーザーマクロを指定できます。 デフォルト: 1h。
proxyid	ID	ディスカバリに使用するプロキシの ID。
status	integer	ディスカバリルールが有効かどうか。 指定可能な値: 0 - （デフォルト）有効; 1 - 無効。
concurrency_max	integer	ディスカバリルールごとの同時チェック数の最大値。 指定可能な値: 0 - （デフォルト）チェック数は無制限; 1 - 1 回のチェック; 2-999 - カスタムのチェック数。
error	string	ディスカバリルールの実行時に障害が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用

drule.create

説明

`object drule.create(object/array discoveryRules)`

このメソッドを使用すると、新しいディスカバリルールを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するディスカバリルール。

標準のディスカバリルールのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
dchecks	array	ディスカバリルール用に作成するディスカバリチェック。 パラメータの動作: - 必須

戻り値

(object) druleids プロパティの下に作成されたディスカバリルールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたディスカバリルールの順序と一致します。

例

ディスカバリルールの作成

ローカルネットワークで Zabbix エージェントを実行しているマシンを見つけるためのディスカバリルールを作成します。ルールはポート 10050 で単一の Zabbix エージェントチェックを使用する必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.create",
    "params": {
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "concurrency_max": "10",
        "dchecks": [
            {
                "type": "9",
                "key_": "system.uname",
                "ports": "10050",
                "uniq": "0"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ディスカバリチェック

ソース

CDRule::create() in *ui/include/classes/api/services/CDRule.php*.

drule.delete

説明

`object drule.delete(array discoveryRuleIds)`

このメソッドを使用すると、ディスカバリルールを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するディスカバリルールの ID。

戻り値

(object) druleids プロパティの下にある削除されたディスカバリルールの ID を含むオブジェクトを返します。

例

複数のディスカバリルールの削除

2 つのディスカバリルールを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CDRule::delete() in *ui/include/classes/api/services/CDRule.php*.

drule.get

説明

`integer/array drule.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従ってディスカバリルールを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dhostids	ID/array	指定した検出済みホストを作成したディスカバリルールのみを返します。
druleids	ID/array	指定した ID を持つディスカバリルールのみを返します。
dserviceids	ID/array	指定した検出済みサービスを作成したディスカバリルールのみを返します。
selectDChecks	query	ディスカバリルールで使用されるディスカバリチェックを含む`dchecks`プロパティを返します。 `count` をサポートします。
selectDHosts	query	ディスカバリルールによって作成された検出済みホストを含む`dhosts`プロパティを返します。 `count` をサポートします。

パラメータ	型	説明
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectDChecks - 結果は dcheckid でソートされます。 selectDHosts - 結果は dhostsid でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: druleid, name。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

すべてのディスカバリルールの取得

設定されたすべてのディスカバリルールと、それらが使用するディスカバリチェックを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "druleid": "2",
            "proxyid": "0",
            "name": "Local network",
            "iprange": "192.168.3.1-255",
            "delay": "5s",
            "status": "0",
            "concurrency_max": "0",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "7",
                    "druleid": "2",
                    "type": "3",
```

```
                "key_": "",
                "snmp_community": "",
                "ports": "21",
                "snmpv3_securityname": "",
                "snmpv3_securitylevel": "0",
                "snmpv3_authpassphrase": "",
                "snmpv3_privpassphrase": "",
                "uniq": "0",
                "snmpv3_authprotocol": "0",
                "snmpv3_privprotocol": "0",
                "snmpv3_contextname": "",
                "host_source": "1",
                "name_source": "0",
                "allow_redirect": "0"
            },
            {
                "dcheckid": "8",
                "druleid": "2",
                "type": "4",
                "key_": "",
                "snmp_community": "",
                "ports": "80",
                "snmpv3_securityname": "",
                "snmpv3_securitylevel": "0",
                "snmpv3_authpassphrase": "",
                "snmpv3_privpassphrase": "",
                "uniq": "0",
                "snmpv3_authprotocol": "0",
                "snmpv3_privprotocol": "0",
                "snmpv3_contextname": "",
                "host_source": "1",
                "name_source": "0",
                "allow_redirect": "0"
            }
        ]
    },
    {
        "druleid": "6",
        "proxyid": "0",
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "delay": "1h",
        "status": "0",
        "concurrency_max": "10",
        "error": "",
        "dchecks": [
            {
                "dcheckid": "10",
                "druleid": "6",
                "type": "9",
                "key_": "system.uname",
                "snmp_community": "",
                "ports": "10050",
                "snmpv3_securityname": "",
                "snmpv3_securitylevel": "0",
                "snmpv3_authpassphrase": "",
                "snmpv3_privpassphrase": "",
                "uniq": "0",
                "snmpv3_authprotocol": "0",
                "snmpv3_privprotocol": "0",
                "snmpv3_contextname": "",
                "host_source": "2",
```

```
                    "name_source": "3",
                    "allow_redirect": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- 検出されたホスト
- ディスカバリチェック

ソース

CDRule::get() in *ui/include/classes/api/services/CDRule.php*.

drule.update

説明

`object drule.update(object/array discoveryRules)`

このメソッドを使用すると、既存のディスカバリルールを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するディスカバリルールのプロパティ。

各ディスカバリルールでは druleid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のディスカバリルールプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
dchecks	array	既存のチェックを置き換えるディスカバリチェック。

戻り値

(object) druleids プロパティの下で更新されたディスカバリルール ID を含むオブジェクトを返します。

例

ディスカバリルールの IP アドレス範囲の変更

ディスカバリルールの IP アドレス範囲を”192.168.2.1-255” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ディスカバリチェック

ソース

CDRule::update() in *ui/include/classes/api/services/CDRule.php*.

デバイス

このクラスは、Zabbix モバイルアプリで使用するモバイルデバイスに対応するよう設計されています。

オブジェクトリファレンス:

- Device

利用可能なメソッド:

- device.get - デバイスを取得
- device.offboard - デバイスを削除（オフボード）

デバイスオブジェクト

Device オブジェクトは、Zabbix モバイルアプリで使用するモバイルデバイスを表します。

Property	Type	Description
deviceid	ID	デバイス ID（内部 ID、例: 1、2）。 プロパティの動作: - 読み取り専用
userid	ID	デバイスユーザーの ID。 プロパティの動作: - 固定
uuid	string	デバイスの汎用一意識別子（UUIDv7）（外部 ID、例: `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`）。 プロパティの動作: - 読み取り専用
name	string	デバイス名。 プロパティの動作: - 読み取り専用
status	integer	デバイスのステータス。 指定可能な値: 0 - 新規 1 - アクティブ 2 - 未割り当て プロパティの動作: - 読み取り専用

Property	Type	Description
activated_at	timestamp	モバイルアプリにデバイスが追加された日時。まだ追加されていない場合は 0。 プロパティの動作: - 読み取り専用
lastaccess	timestamp	モバイルアプリでデバイスが最後に認証された日時。まだ認証されていない場合は 0。 プロパティの動作: - 読み取り専用

device.get

このメソッドを使用すると、Zabbix モバイルアプリで使用するモバイルデバイスを取得できます。

このメソッドは、EnableMobileDevicesが有効で、ユーザーがデバイスへのアクセス権を持っている場合、*User*、*Admin*、および *Super admin* ユーザーが使用できます。API メソッドへのアクセスは、ユーザーロールの設定で取り消すことができます。

User および *Admin* ユーザーは、自分のデバイスのみ取得できます。

メソッドのシグネチャ: `integer/array device.get(object parameters)`

メソッドのソース: `ui/include/classes/api/services/CDevice.php` の `CDevice::get()`

パラメータ

Parameter	Type	Description
deviceids	ID/array	指定したデバイス ID を持つデバイスのみを返します。
userids	ID/array	指定したユーザー ID を持つユーザーに属するデバイスのみを返します。
roleids	ID/array	指定したユーザーロール ID を持つユーザーに属するデバイスのみを返します。
usrgrpids	ID/array	指定したユーザーグループ ID を持つユーザーに属するデバイスのみを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `deviceid`、`name`。
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明しています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

オブジェクトの配列、または取得したオブジェクト数（`countOutput` を使用した場合）を返します。

例

デバイスの取得

次のユーザー ID を持つユーザーに属するデバイスを取得します: 1、2、3。
 ユーザー 2（ゲスト）にはデバイスがなく、デバイス 3 はまだ Zabbix モバイルアプリに追加または認証されていないことに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
        },
        {
            "deviceid": "2",
            "userid": "1",
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```

device.offboard

このメソッドを使用すると、Zabbix モバイルアプリで使用するモバイルデバイスを削除（オフボード）できます。

このメソッドは、EnableMobileDevicesが有効で、ユーザーがデバイスへのアクセス権を持っている場合、*User*、*Admin*、および *Super admin* ユーザーが使用できます。API メソッドへのアクセスは、ユーザーロールの設定で取り消すことができます。

User および *Admin* ユーザーは、自分のデバイスのみ削除できます。

メソッドのシグネチャ: `object device.offboard(object parameters)`

メソッドのソース: `ui/include/classes/api/services/CDevice.php` の `CDevice::offboard()`

パラメータ

パラメータ	型	説明
uuid	string	デバイスの汎用一意識別子（UUIDv7）。`deviceid` と一致します。
force	boolean	「デバイスが見つかりません」エラーを無視してデバイスを強制的に削除するには、`true` に設定します。 デフォルト: `false`

戻り値

削除されたデバイスの UUID を含むオブジェクトを返します。

例

デバイスの削除

指定した UUID のデバイスを削除し、「Device not found」エラーを無視します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

トリガー

このクラスはトリガーを操作するために設計されています。

オブジェクト参照:

- Trigger
- Trigger tag

利用可能なメソッド:

- trigger.create - 新しいトリガーを作成
- trigger.delete - トリガーを削除
- trigger.get - トリガーを取得
- trigger.update - トリガーを更新

Trigger オブジェクト

次のオブジェクトは、`trigger`API に直接関連しています。

トリガー

トリガーオブジェクトには、以下のプロパティがあります。

Property	Type	Description
triggerid	ID	トリガーの ID。 Property behavior: - *read-only* - 更新操作で *required*
description	string	トリガーの名前。 Property behavior: - 作成操作で *required*

Property	Type	Description
expression	string	トリガー条件式。 `value_type` が 5（バイナリ）または 6（JSON）に設定されているアイテムは、トリガー条件式では使用できません。 Property behavior: - 作成操作で *required*
event_name	string	トリガーによって生成されるイベント名。
opdata	string	運用データ。
comments	string	トリガーの追加説明。
error	string	トリガーの状態更新時に問題が発生した場合のエラーテキスト。 Property behavior: - *read-only*
flags	integer	トリガーの生成元。 使用可能な値: 0 - *(default)* 通常のトリガー; 4 - プロトタイプから変換されたトリガー。 Property behavior: - *read-only*
lastchange	timestamp	トリガーの状態が最後に変更された時刻。 Property behavior: - *read-only*
priority	integer	トリガーの深刻度。 使用可能な値: 0 - *(default)* 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。
state	integer	トリガー条件式の状態。 使用可能な値: 0 - *(default)* トリガーの状態は最新; 1 - 現在のトリガーの状態は不明。 Property behavior: - *read-only*
status	integer	トリガーが有効か無効か。 使用可能な値: 0 - *(default)* 有効; 1 - 無効。
templateid	ID	親テンプレートのトリガー ID。 Property behavior: - *read-only*
type	integer	トリガーが複数の障害イベントを生成できるかどうか。 使用可能な値: 0 - *(default)* 複数イベントを生成しない; 1 - 複数イベントを生成する。
url	string	トリガーに関連付けられた URL。
url_name	string	トリガーに関連付けられた URL のラベル。

Property	Type	Description
value	integer	トリガーが OK 状態か障害状態か。 使用可能な値: 0 - *(default)* OK; 1 - 障害。 Property behavior: - *read-only*
recovery_mode	integer	OK イベントの生成モード。 使用可能な値: 0 - *(default)* 条件式; 1 - 復旧条件式; 2 - なし。
recovery_expression	string	トリガーの復旧条件式。 `value_type` が 5（バイナリ）または 6（JSON）に設定されているアイテムは、トリガーの復旧条件式では使用できません。
correlation_mode	integer	OK イベントのクローズ方法。 使用可能な値: 0 - *(default)* すべての障害; 1 - タグ値が一致する場合はすべての障害。
correlation_tag	string	照合に使用するタグ。
manual_close	integer	手動クローズを許可するかどうか。 使用可能な値: 0 - *(default)* いいえ; 1 - はい。
uuid	string	インポートしたトリガーを既存のトリガーにリンクするために使用される汎用一意識別子。指定されていない場合は自動生成されます。 Property behavior: - トリガーがテンプレートに属している場合に *supported*

トリガータグ

トリガータグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	トリガータグ名。 プロパティの動作: - 必須
value	string	トリガータグの値。 プロパティの動作: - 読み取り専用。
object	integer	タグの継承元オブジェクトのタイプ。 使用可能な値: 0 - テンプレート; 1 - ホスト; 3 - アイテム。 プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

プロパティ	型	説明
automatic	integer	トリガータグのタイプ: 使用可能な値: 0 - *(デフォルト)* ユーザーが作成したタグ; 1 - LLD によって作成されたタグ;プロパティの動作: - 読み取り専用。

trigger.create

説明

`object trigger.create(object/array triggers)`

このメソッドを使用すると、新しいトリガーを作成できます。

> **Note:**
> このメソッドは *Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するトリガー。

標準のトリガープロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
dependencies	array	トリガーが依存するトリガー。 トリガーには `triggerid` プロパティのみを定義する必要があります。
tags	array	トリガータグ。

戻り値

(`object`) `triggerids` プロパティの下で作成されたトリガーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガーの順序と一致します。

例

トリガーの作成

2 つのトリガーを作成し、それぞれが他の 1 つのトリガーに依存するようにします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Processor load is too high on {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Service has stopped",
            "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
            "dependencies": [
                {
                    "triggerid": "17368"
```

```
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::create() in *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

説明

`object trigger.delete(array triggerIds)`

このメソッドを使用すると、トリガーを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`array`) 削除するトリガーの ID。

戻り値

(`object`) `triggerids` プロパティの下にある削除されたトリガーの ID を含むオブジェクトを返します。

例

複数のトリガーの削除

2 つのトリガーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::delete() in *ui/include/classes/api/services/CTrigger.php*.

trigger.get

説明

`integer/array trigger.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってトリガーを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドでは、以下のパラメータをサポートしています。

パラメータ	型	説明
triggerids	ID/array	指定した ID を持つトリガーのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループのホストまたはテンプレートに属するトリガーのみを返します。
templateids	ID/array	指定したテンプレートに属するトリガーのみを返します。
hostids	ID/array	指定したホストに属するトリガーのみを返します。
itemids	ID/array	指定したアイテムを含むトリガーのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるトリガーのみを返します。
functions	string/array	指定した関数を使用するトリガーのみを返します。 サポートされている関数の一覧については、サポートされている関数のページを参照してください。
group	string	指定した名前のホストグループまたはテンプレートグループに属するホストまたはテンプレートのトリガーのみを返します。
host	string	指定した技術名のホストに属するトリガーのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたトリガーのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するトリガーのみを返します。
dependent	boolean	`true` に設定すると、依存関係を持つトリガーのみを返します。`false` に設定すると、依存関係を持たないトリガーのみを返します。
monitored	flag	有効なホストに属し、有効なアイテムのみを含む、有効なトリガーのみを返します。
active	flag	監視中のホストに属する有効なトリガーのみを返します。
maintenance	boolean	`true` に設定すると、メンテナンス中のホストに属する有効なトリガーのみを返します。
withUnacknowledgedEvents	flag	未確認イベントを持つトリガーのみを返します。

パラメータ	型	説明
withAcknowledgedEvents	flag	すべてのイベントが確認済みのトリガーのみを返します。
withLastEventUnacknowledged	flag	最後のイベントが未確認のトリガーのみを返します。
skipDependent	flag	他のトリガーに依存する障害状態のトリガーを除外します。他のトリガーが無効、アイテムが無効、またはアイテムのホストが無効の場合、それらのトリガーは無視されます。
lastChangeSince	timestamp	指定した時刻より後に状態が変更されたトリガーのみを返します。
lastChangeTill	timestamp	指定した時刻より前に状態が変更されたトリガーのみを返します。
only_true	flag	最近（ok_period以内）障害状態になったトリガーのみを返します。
min_severity	integer	指定した重要度以上のトリガーのみを返します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	array	指定したタグを持つトリガーのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのトリガーを返します。 指定可能な演算子の値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
expandComment	flag	トリガーの説明内のマクロを展開します。
expandDescription	flag	トリガー名内のマクロを展開します。
expandExpression	flag	トリガーの条件式内の関数とマクロを展開します。
selectHostGroups	query	トリガーが属するホストグループをhostgroupsプロパティで返します。
selectHosts	query	トリガーが属するホストをhostsプロパティで返します。
selectItems	query	トリガーに含まれるアイテムをitemsプロパティで返します。
selectFunctions	query	トリガーで使用されている関数を `functions` プロパティで返します。 関数オブジェクトはトリガーの条件式で使用されている関数を表し、以下のプロパティを持ちます: `functionid` - (ID) 関数の ID; `itemid` - (ID) 関数で使用されるアイテムの ID; `function` - (string) 関数名; `parameter` - (string) 関数に渡されるパラメータ。返される文字列では、クエリパラメータが `$` 記号に置き換えられます。
selectDependencies	query	トリガーが依存するトリガーを `dependencies` プロパティで返します。

パラメータ	型	説明
selectDiscoveryData	query	トリガーのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。トリガーのディスカバリオブジェクトは、検出されたトリガーを、そのトリガーの検出元となったトリガープロトタイプに関連付けます。 以下のプロパティを持ちます: `parent_triggerid` - (ID) トリガーの作成元となったトリガープロトタイプの ID; `status` - (int) トリガーのディスカバリステータス: 0 - *(デフォルト)* トリガーは検出済み; 1 - トリガーは検出されなくなった; `ts_delete` - (timestamp) 検出されなくなったトリガーが削除される時刻; `ts_disable` - (timestamp) 検出されなくなったトリガーが無効になる時刻; `disable_source` - (int) トリガーが LLD ルールによって無効にされたか、手動で無効にされたかを示す値: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	トリガーを作成したローレベルディスカバリルールを `discoveryRule` プロパティで返します。
selectLastEvent	query	重要なトリガーイベントのうち、最後のものを`lastEvent`プロパティで返します。
selectTags	query	トリガーのタグを`tags`プロパティで返します。
inheritedTags	boolean	指定した `tags` をテンプレート、ホスト、またはリンクされたテンプレートにも持つトリガーを返します。 指定可能な値: `true` - テンプレート、ホスト、またはリンクされたテンプレートにも指定したタグが必要です; `false` - *(デフォルト)* テンプレート、ホスト、またはリンクされたテンプレートのタグを無視します。
selectInheritedTags	query	テンプレート、リンクされたテンプレート、ホスト、およびトリガーまたはリカバリ条件式で参照されるアイテムから継承されたタグを含む`inheritedTags`プロパティを返します。
selectTemplateGroups	query	トリガーが属するテンプレートグループを`templategroups`プロパティで返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 プロパティ名をキーとし、一致させる単一の値または値の配列を値とするオブジェクトを指定します。 `text`データ型のプロパティはサポートしていません。 以下の追加プロパティをサポートしています: `host` - トリガーが属するホストの技術名; `hostid` - トリガーが属するホストの ID。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `host` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `triggerid`、`description`、`status`、`priority`、`lastchange`、`hostname`。
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明しています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	

パラメータ	型	説明
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	トリガーのディスカバリオブジェクトを triggerDiscovery プロパティで返します。トリガーのディスカバリオブジェクトは、トリガーを作成したトリガープロトタイプにトリガーを関連付けます。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

トリガー ID によるデータの取得

トリガー"14062" で使用されているすべてのデータと関数を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "14062",
            "expression": "{13513}<10m",
            "description": "{HOST.NAME} has been restarted (uptime < 10m)",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "The host uptime is less than 10 minutes",
            "error": "",
            "templateid": "10016",
            "type": "0",
            "state": "0",
            "flags": "0",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "event_name": "",
```

```
            "uuid": "",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13513",
                    "itemid": "24350",
                    "triggerid": "14062",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

障害状態のトリガーの取得

障害状態にあるすべてのトリガーの ID、名前、深刻度を取得し、深刻度の降順で並べ替えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```

タグ付きの特定のトリガーの取得

タグ付きの特定のトリガーを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17370",
            "description": "Service status",
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}",
                    "automatic": "1"
                },
                {
                    "tag": "error",
                    "value": "",
                    "automatic": "1"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- ディスカバリルール
- アイテム
- ホスト
- ホストグループ
- テンプレートグループ

ソース

CTrigger::get() in *ui/include/classes/api/services/CTrigger.php*.

trigger.update

説明

`object trigger.update(object/array triggers)`

このメソッドを使用すると、既存のトリガーを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するトリガーのプロパティ。

各トリガーでは triggerid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

標準のトリガープロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
dependencies	array	トリガーが依存しているトリガー。
tags	array	トリガーでは triggerid プロパティのみを定義する必要があります。 トリガータグ。

戻り値

(object) triggerids プロパティの下で更新されたトリガーの ID を含むオブジェクトを返します。

例

トリガーの有効化

トリガーを有効にします。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

トリガータグの置換

トリガーのタグを置換します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "tags": [
            {
                "tag": "service",
                "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
            },
            {
                "tag": "error",
                "value": ""
            }
        ]
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

依存関係の置き換え

トリガーの依存関係を置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::update() in *ui/include/classes/api/services/CTrigger.php*.

トリガープロトタイプ

このクラスは、トリガープロトタイプを操作するために設計されています。

オブジェクト参照:

- トリガープロトタイプ
- トリガープロトタイプタグ

利用可能なメソッド:

- triggerprototype.create - 新しいトリガープロトタイプを作成
- triggerprototype.delete - トリガープロトタイプを削除
- triggerprototype.get - トリガープロトタイプを取得

- triggerprototype.update - トリガープロトタイプを更新

トリガープロトタイプオブジェクト

以下のオブジェクトは `triggerprototype` API に直接関連しています。

トリガープロトタイプ

トリガープロトタイプオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
triggerid	ID	トリガープロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
description	string	トリガープロトタイプの名前。 プロパティの動作: - 作成操作で 必須
expression	string	トリガー条件式。 少なくとも 1 つのアイテムプロトタイプを含める必要があります。 `value_type` が 5（バイナリ）または 6（JSON）に設定されているアイテムプロトタイプは、トリガー条件式では使用できません。 プロパティの動作: - 作成操作で 必須
event_name	string	トリガーによって生成されるイベント名。
opdata	string	運用データ。
comments	string	トリガープロトタイプに対する追加コメント。
priority	integer	トリガープロトタイプの深刻度。 指定可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。
status	integer	トリガープロトタイプが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
flags	integer	トリガープロトタイプの生成元。 指定可能な値: 2 - トリガープロトタイプ; 6 - ディスカバリされたトリガープロトタイプ。 プロパティの動作: - 読み取り専用
templateid	ID	親テンプレートのトリガープロトタイプの ID。 プロパティの動作: - 読み取り専用
type	integer	トリガープロトタイプが複数の障害イベントを生成できるかどうか。 指定可能な値: 0 - *(デフォルト)* 複数イベントを生成しない; 1 - 複数イベントを生成する。
url	string	トリガープロトタイプに関連付けられた URL。

プロパティ	型	説明
url_name	string	トリガープロトタイプに関連付けられた URL のラベル。
recovery_mode	integer	OK イベントの生成モード。 指定可能な値: 0 - *(デフォルト)* 条件式; 1 - 復旧条件式; 2 - なし。
recovery_expression	string	トリガーの復旧条件式。 少なくとも 1 つのアイテムプロトタイプを含める必要があります。 `value_type` が 5（バイナリ）または 6（JSON）に設定されているアイテムプロトタイプは、トリガーの復旧条件式では使用できません。
correlation_mode	integer	OK イベントのクローズ対象。 指定可能な値: 0 - *(デフォルト)* すべての障害; 1 - タグの値が一致する場合はすべての障害。
correlation_tag	string	照合に使用するタグ。
manual_close	integer	手動クローズを許可するかどうか。 指定可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。
discover	integer	トリガープロトタイプのディスカバリ状態。 指定可能な値: 0 - *(デフォルト)* 新しいトリガーがディスカバリされる; 1 - 新しいトリガーはディスカバリされず、既存のトリガーは消失としてマークされる。
uuid	string	既存のトリガープロトタイプにインポートしたトリガープロトタイプを関連付けるために使用される、汎用一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - トリガープロトタイプがテンプレートに属している場合に サポート

トリガープロトタイプタグ

トリガープロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	トリガープロトタイプタグ名。 Property behavior: - *required*
value	string	トリガープロトタイプタグの値。 Property behavior: - *read-only*.
object	integer	タグの継承元オブジェクトのタイプ。 指定可能な値: 0 - テンプレート; 1 - ホスト; 3 - アイテム; 4 - アイテムプロトタイプ。 Property behavior: - *read-only*.

プロパティ	Type	説明
objectid	ID	タグの継承元オブジェクトの ID。 Property behavior: - *read-only*.

triggerprototype.create

説明

`object triggerprototype.create(object/array triggerPrototypes)`

このメソッドを使用すると、新しいトリガープロトタイプを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するトリガープロトタイプ。

このメソッドは、標準のトリガープロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
dependencies	array	トリガープロトタイプが依存するトリガーおよびトリガープロトタイプ。 トリガーでは、`triggerid` プロパティのみを定義する必要があります。
tags	array	トリガープロトタイプタグ。

戻り値

(`object`) `triggerids` プロパティの下に作成されたトリガープロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガープロトタイプの順序と一致します。

例

トリガープロトタイプの作成

ファイルシステムの空きディスク容量が 20% 未満になった場合に検出するトリガープロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "Free disk space is less than 20% on volume {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::create() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

説明

`object triggerprototype.delete(array triggerPrototypeIds)`

このメソッドを使用すると、トリガープロトタイプを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するトリガープロトタイプの ID。

戻り値

(object) triggerids プロパティの下にある削除されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

複数のトリガープロトタイプの削除

2 つのトリガープロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

説明

`integer/array triggerprototype.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってトリガープロトタイプを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
active	flag	監視対象のホストに属する、有効なトリガープロトタイプのみを返します。
discoveryids	ID/array	指定した LLD ルールに属するトリガープロトタイプのみを返します。
functions	string/array	指定した関数を使用するトリガーのみを返します。 サポートされている関数の一覧については、Supported functionsページを参照してください。
group	string	指定した名前のホストグループまたはテンプレートグループに含まれるホストまたはテンプレートに属するトリガープロトタイプのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに含まれるホストまたはテンプレートに属するトリガープロトタイプのみを返します。
host	string	指定した名前のホストに属するトリガープロトタイプのみを返します。
hostids	ID/array	指定したホストに属するトリガープロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたトリガープロトタイプのみを返します。
maintenance	boolean	`true` に設定すると、メンテナンス中のホストに属する有効なトリガープロトタイプのみを返します。
min_severity	integer	指定した深刻度以上のトリガープロトタイプのみを返します。
monitored	flag	監視対象のホストに属し、かつ有効なアイテムのみを含む、有効なトリガープロトタイプのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するトリガープロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するトリガープロトタイプのみを返します。
triggerids	ID/array	指定した ID を持つトリガープロトタイプのみを返します。
expandExpression	flag	トリガー条件式内の関数およびマクロを展開します。
selectDependencies	query	トリガープロトタイプが依存しているトリガープロトタイプおよびトリガーを、`dependencies` プロパティで返します。

Parameter	Type	Description
selectDiscoveryData	query	トリガープロトタイプのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。トリガープロトタイプのディスカバリオブジェクトは、検出されたトリガープロトタイプを、その検出元となったトリガープロトタイプに関連付けます。 このオブジェクトは以下のプロパティを持ちます。 `parent_triggerid` - (ID) トリガープロトタイプの作成元となったトリガープロトタイプの ID。 `status` - (int) トリガープロトタイプのディスカバリステータス: 0 - *(default)* トリガープロトタイプは検出されています。 1 - トリガープロトタイプはすでに検出されていません。 `ts_delete` - (timestamp) すでに検出されていないトリガープロトタイプが削除される時刻。 `ts_disable` - (timestamp) すでに検出されていないトリガープロトタイプが無効化される時刻。 `disable_source` - (int) トリガープロトタイプが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(default)* 自動的に無効化。 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	トリガープロトタイプが属するLLD ルールを `discoveryRule` プロパティで返します。
selectDiscoveryRulePrototype	query	トリガープロトタイプが属する親 LLD ルールプロトタイプを含む`discoveryRulePrototype` プロパティを返します。
selectFunctions	query	トリガープロトタイプで使用されている関数を `functions` プロパティで返します。 関数オブジェクトはトリガー条件式で使用される関数を表し、以下のプロパティを持ちます。 `functionid` - (ID) 関数の ID。 `itemid` - (ID) 関数で使用されるアイテムの ID。 `function` - (string) 関数名。 `parameter` - (string) 関数に渡されるパラメータ。クエリパラメータは、返される文字列内で $ 記号に置き換えられます。
selectHostGroups	query	トリガープロトタイプが属するホストグループを`hostgroups` プロパティで返します。
selectHosts	query	トリガープロトタイプが属するホストを`hosts` プロパティで返します。
selectInheritedTags	query	テンプレート、リンクされたテンプレート、ホストから継承されたタグ、およびトリガー条件式またはリカバリ条件式で参照されるアイテムプロトタイプから継承されたタグを含む`inheritedTags` プロパティを返します。
selectItems	query	トリガープロトタイプで使用されるアイテムおよびアイテムプロトタイプを`items` プロパティで返します。
selectTags	query	トリガープロトタイプのタグを`tags` プロパティで返します。
selectTemplateGroups	query	トリガープロトタイプが属するテンプレートグループを`templategroups` プロパティで返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、値を一致対象の単一の値または値の配列とするオブジェクトを受け付けます。 `text` data type のプロパティはサポートしていません。 追加のプロパティをサポートしています。 `host` - トリガープロトタイプが属するホストの技術名。 `hostid` - トリガープロトタイプが属するホストの ID。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます。 `selectHosts` - 結果は `host` でソートされます。

Parameter	Type	Description
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	これらのパラメータはreference commentaryで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

LLD ルールからトリガープロトタイプを取得する

LLD ルールからすべてのトリガープロトタイプとその関数を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
```

```
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "triggerid": "13272",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


タグで特定のトリガープロトタイプを取得

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
```

```
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- ディスカバリルール
- アイテム
- ホスト
- ホストグループ
- テンプレートグループ

ソース

ui/include/classes/api/services/CTriggerPrototype.php の CTriggerPrototype::get()。

triggerprototype.update

説明

`object triggerprototype.update(object/array triggerPrototypes)`

このメソッドを使用すると、既存のトリガープロトタイプを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細については、User roles を参照してください。

パラメータ

(`object/array`) 更新するトリガープロトタイプのプロパティ。

各トリガープロトタイプでは `triggerid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のトリガープロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
dependencies	array	トリガープロトタイプが依存するトリガーおよびトリガープロトタイプ。 トリガーでは triggerid プロパティのみを定義する必要があります。
tags	array	トリガープロトタイプタグ。

戻り値

(object) triggerids プロパティの下で更新されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

トリガープロトタイプの有効化

トリガープロトタイプを有効にします。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

トリガープロトタイプタグの置換

1 つのトリガープロトタイプのタグを置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```


ソース

CTriggerPrototype::update() in *ui/include/classes/api/services/CTriggerPrototype.php*.

トレンド

このクラスはトレンドデータを扱うために設計されています。

オブジェクト参照:

- 浮動小数点トレンド
- 整数トレンド

利用可能なメソッド:

- trend.get - トレンドを取得します

Trend オブジェクト

次のオブジェクトは、`trend`API に直接関連しています。

> **Note:**
> トレンドオブジェクトは、アイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API によって変更することはできません。

浮動小数点トレンド

浮動小数点トレンドオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	値が計算された時間のタイムスタンプ。たとえば、”04:00:00” のタイムスタンプは、”04:00:00-04:59:59” の期間について計算された値を意味します。
itemid	ID	関連するアイテムの ID。
num	integer	その時間に利用可能だった値の数。
value_min	float	1 時間あたりの最小値。
value_avg	float	1 時間あたりの平均値。
value_max	float	1 時間あたりの最大値。

整数トレンド

整数トレンドオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	値が計算された 1 時間のタイムスタンプ。たとえば、”04:00:00” のタイムスタンプは、”04:00:00-04:59:59” の期間について計算された値を意味します。
itemid	ID	関連するアイテムの ID。
num	integer	その 1 時間で利用可能だった値の数。
value_min	integer	1 時間あたりの最小値。
value_avg	integer	1 時間あたりの平均値。
value_max	integer	1 時間あたりの最大値。

trend.get

説明

`integer/array trend.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従ってトレンドデータを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。
> このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。
> 詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
itemids	ID/array	指定したアイテム ID を持つトレンドのみを返します。
time_from	timestamp	指定した時刻以降、またはその時刻に収集された値のみを返します。
time_till	timestamp	指定した時刻以前、またはその時刻に収集された値のみを返します。
countOutput	boolean	取得したオブジェクトの数をカウントします。
limit	integer	取得するオブジェクトの数を制限します。
output	query	返却するTrend object のプロパティを設定します。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

アイテムのトレンドデータの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
```

```
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```


ソース

CTrend::get() in *ui/include/classes/api/services/CTrend.php*.

プロキシグループ

このクラスは、プロキシグループを操作するために設計されています。

オブジェクト参照:

- プロキシグループ

利用可能なメソッド:

- proxygroup.create - 新しいプロキシグループを作成
- proxygroup.delete - プロキシグループを削除
- proxygroup.get - プロキシグループを取得
- proxygroup.update - プロキシグループを更新

Proxy group オブジェクト

次のオブジェクトは、proxygroupAPI に直接関連しています。

プロキシグループ

プロキシグループオブジェクトには、次のプロパティがあります。

Property	Type	Description
proxy_groupid	ID	プロキシグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	プロキシグループの名前。 プロパティの動作: - 作成操作では必須
description	text	プロキシグループの説明。
failover_delay	string	プロキシグループ内のプロキシがオンラインとみなされるために、Zabbix サーバーと通信する必要がある期間。 時間のサフィックスがサポートされています（例: 30s、1m）。 ユーザーマクロがサポートされています。 指定可能な値: 10s-15m。 デフォルト: 1m。
min_online	string	プロキシグループをオンラインに維持するために必要な、オンライン状態のプロキシの最小数。 ユーザーマクロがサポートされています。 指定可能な値の範囲: 1-1000。 デフォルト: 1。

Property	Type	Description
state	integer	プロキシグループの状態。 指定可能な値: 0 - 不明; 1 - オフライン; 2 - 復旧中; 3 - オンライン; 4 - 低下中。 プロパティの動作: - 読み取り専用

proxygroup.create

説明

`object proxygroup.create(object/array proxyGroups)`

このメソッドを使用すると、新しいプロキシグループを作成できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー役割の設定で取り消すことができます。詳細はUser rolesを参照してください。

パラメータ

(`object/array`) 作成するプロキシグループ。

このメソッドは、標準のプロキシグループプロパティを持つプロキシグループを受け付けます。

戻り値

(`object`) は、`proxy_groupids` プロパティに作成されたプロキシグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシグループの順序と一致します。

例

プロキシグループを作成する

カスタム設定でプロキシグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::create() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

説明

`object proxygroup.delete(array proxyGroupIds)`

このメソッドを使用すると、プロキシグループを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー役割の設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(array) 消去するプロキシグループの ID

戻り値

(object) proxy_groupids プロパティの下にある、削除されたプロキシグループの ID を含むオブジェクトを返す。

例

複数のプロキシグループを削除する

2 つのプロキシグループを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::delete() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

説明

`integer/array proxygroup.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってプロキシグループを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。ユーザーのユーザーグループのプロキシアクセスリストで許可されているプロキシグループのみが返されます。

パラメータ

(object) 希望する出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
proxy_groupids	ID/array	指定した ID を持つプロキシグループのみを返します。
proxyids	ID/array	指定したプロキシを含むプロキシグループのみを返します。
selectProxies	query	プロキシグループに属するプロキシを含む`proxies` プロパティを返します。 `count` をサポートします。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `proxy_groupid`, `name`。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

全てのプロキシグループの取得

すべての構成済みのプロキシとプロキシグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxy_groupid": "1",
```

```
            "name": "Proxy group 1",
            "failover_delay": "1m",
            "min_online": "3",
            "description": "",
            "state": "1",
            "proxies": [
                {
                    "proxyid": "1",
                    "name": "proxy 1"
                },
                {
                    "proxyid": "2",
                    "name": "proxy 2"
                }
            ]
        },
        {
            "proxy_groupid": "2",
            "name": "Proxy group 2",
            "failover_delay": "10m",
            "min_online": "3",
            "description": "",
            "state": "3",
            "proxies": [
                {
                    "proxyid": "3",
                    "name": "proxy 3"
                },
                {
                    "proxyid": "4",
                    "name": "proxy 4"
                },
                {
                    "proxyid": "5",
                    "name": "proxy 5"
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Proxy

ソース

CProxyGroup::get() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

説明

`object proxygroup.update(object/array proxyGroups)`

このメソッドを使用すると、既存のプロキシグループを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するプロキシグループのプロパティ。

各プロキシグループでは proxy_groupid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のプロキシグループプロパティ を持つプロキシグループを受け付けます。

戻り値

(object) proxy_groupids プロパティの下にある、更新されたプロキシグループの ID を含むオブジェクトを返す。

例

オンラインプロキシの最小数を変更する

グループがオンラインになるために必要なオンラインプロキシの最小数を変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::update() in *ui/include/classes/api/services/CProxyGroup.php*.

ホスト

このクラスはホストを操作するために設計されています。

オブジェクト参照:

- Host
- Host inventory
- Host tag

利用可能なメソッド:

- host.create - 新しいホストを作成
- host.delete - ホストを削除
- host.get - ホストを取得
- host.massadd - ホストに関連オブジェクトを追加
- host.massremove - ホストから関連オブジェクトを削除
- host.update - ホストを更新

Host オブジェクト

次のオブジェクトは、hostAPI に直接関連しています。

ホスト

ホストオブジェクトには、次のプロパティがあります。

プロパティ	型	説明
hostid	ID	ホストの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
host	string	ホストの技術的な名前。 プロパティの動作: - 作成操作では必須
description	text	ホストの説明。
flags	integer	ホストの起点。 指定可能な値: 0 - 通常のホスト; 4 - プロトタイプから変換されたホスト。 プロパティの動作: - 読み取り専用
inventory_mode	integer	ホストインベントリの登録モード。 指定可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
ipmi_authtype	integer	IPMI 認証アルゴリズム。 指定可能な値: -1 - *(デフォルト)* デフォルト; 0 - なし; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+。
ipmi_password	string	IPMI パスワード。
ipmi_privilege	integer	IPMI 権限レベル。 指定可能な値: 1 - コールバック; 2 - *(デフォルト)* ユーザー; 3 - オペレーター; 4 - 管理者; 5 - OEM。
ipmi_username	string	IPMI ユーザー名。
maintenance_from	timestamp	有効なメンテナンスの開始時刻。 プロパティの動作: - 読み取り専用
maintenance_status	integer	有効なメンテナンスのステータス。 指定可能な値: 0 - *(デフォルト)* メンテナンスなし; 1 - メンテナンス中。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
maintenance_type	integer	有効なメンテナンスのタイプ。 指定可能な値: 0 - *(デフォルト)* データ収集ありのメンテナンス; 1 - データ収集なしのメンテナンス。 プロパティの動作: - 読み取り専用
maintenanceid	ID	現在ホストに適用されているメンテナンスの ID。 プロパティの動作: - 読み取り専用
name	string	ホストの表示名。 デフォルト: `host` プロパティの値。
monitored_by	integer	ホストの監視に使用されるソース。 指定可能な値: 0 - *(デフォルト)* Zabbix サーバー; 1 - プロキシ; 2 - プロキシグループ。 プロパティの動作: - `select_server_for_monitoring` が「無効」に設定されている場合は必須。
proxyid	ID	ホストの監視に使用されるプロキシの ID。 プロパティの動作: - `monitored_by` が「プロキシ」に設定されている場合は必須
proxy_groupid	ID	ホストの監視に使用されるプロキシグループの ID。 プロパティの動作: - `monitored_by` が「プロキシグループ」に設定されている場合は必須
status	integer	ホストのステータスと機能。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
tls_connect	integer	ホストへの接続。 指定可能な値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。
tls_accept	integer	ホストからの接続。 指定可能なビットマップ値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。 これはビットマスクフィールドです。指定可能なビットマップ値の合計を使用できます（たとえば、PSK と証明書の場合は 6）。
tls_issuer	string	証明書の発行者。
tls_subject	string	証明書のサブジェクト。

プロパティ	型	説明
tls_psk_identity	string	PSK ID。自動登録、ホスト、プロキシ全体で、1 つの PSK とのみ組み合わせる必要があります。 PSK ID には機密情報を含めないでください。使用する PSK を受信側に通知するため、PSK ID は暗号化されずにネットワーク経由で送信されます。 プロパティの動作: - 書き込み専用 - `tls_connect` が「PSK」に設定されている場合、または `tls_accept` に「PSK」ビットが含まれている場合は必須
tls_psk	string	事前共有鍵（PSK）。32 桁以上の 16 進数である必要があります。 プロパティの動作: - 書き込み専用 - `tls_connect` が「PSK」に設定されている場合、または `tls_accept` に「PSK」ビットが含まれている場合は必須
active_available	integer	ホストのアクティブインターフェースの可用性ステータス。 指定可能な値: 0 - インターフェースのステータスが不明; 1 - インターフェースが使用可能; 2 - インターフェースが使用不可。 プロパティの動作: - 読み取り専用
assigned_proxyid	ID	ホストがプロキシグループによって監視されている場合に、Zabbix サーバーによって割り当てられたプロキシの ID。 プロパティの動作: - 読み取り専用

ホストインベントリ

ホストインベントリオブジェクトには、以下のプロパティがあります。

Note:
各プロパティには固有の ID 番号があり、ホストインベントリフィールドをアイテムに関連付けるために使用されます。

ID	プロパティ	Type	説明	最大長
4	alias	string	別名。	128 文字
11	asset_tag	string	資産タグ。	64 文字
28	chassis	string	シャーシ。	64 文字
23	contact	string	連絡担当者。	SQL データベースでは 65535 文字
32	contract_number	string	契約番号。	64 文字
47	date_hw_decomm	string	HW 廃止日。	64 文字
46	date_hw_expiry	string	HW 保守期限日。	64 文字
45	date_hw_install	string	HW 設置日。	64 文字
44	date_hw_purchase	string	HW 購入日。	64 文字
34	deployment_status	string	展開ステータス。	64 文字
14	hardware	string	ハードウェア。	255 文字
15	hardware_full	string	ハードウェア詳細。	SQL データベースでは 65535 文字
39	host_netmask	string	ホストサブネットマスク。	39 文字
38	host_networks	string	ホストネットワーク。	SQL データベースでは 65535 文字
40	host_router	string	ホストルーター。	39 文字
30	hw_arch	string	HW アーキテクチャ。	32 文字
33	installer_name	string	設置担当者名。	64 文字
24	location	string	場所。	SQL データベースでは 65535 文字
25	location_lat	string	場所の緯度。	16 文字
26	location_lon	string	場所の経度。	16 文字
12	macaddress_a	string	MAC アドレス A。	64 文字

ID	プロパティ	Type	説明	最大長
13	macaddress_b	string	MAC アドレス B。	64 文字
29	model	string	モデル。	64 文字
3	name	string	名前。	128 文字
27	notes	string	メモ。	SQL データベースでは 65535 文字
41	oob_ip	string	OOB IP アドレス。	39 文字
42	oob_netmask	string	OOB ホストサブネットマスク。	39 文字
43	oob_router	string	OOB ルーター。	39 文字
5	os	string	OS 名。	128 文字
6	os_full	string	詳細 OS 名。	255 文字
7	os_short	string	短縮 OS 名。	128 文字
61	poc_1_cell	string	主 POC 携帯電話番号。	64 文字
58	poc_1_email	string	主メールアドレス。	128 文字
57	poc_1_name	string	主 POC 名。	128 文字
63	poc_1_notes	string	主 POC メモ。	SQL データベースでは 65535 文字
59	poc_1_phone_a	string	主 POC 電話 A。	64 文字
60	poc_1_phone_b	string	主 POC 電話 B。	64 文字
62	poc_1_screen	string	主 POC スクリーン名。	64 文字
68	poc_2_cell	string	副 POC 携帯電話番号。	64 文字
65	poc_2_email	string	副 POC メールアドレス。	128 文字
64	poc_2_name	string	副 POC 名。	128 文字
70	poc_2_notes	string	副 POC メモ。	SQL データベースでは 65535 文字
66	poc_2_phone_a	string	副 POC 電話 A。	64 文字
67	poc_2_phone_b	string	副 POC 電話 B。	64 文字
69	poc_2_screen	string	副 POC スクリーン名。	64 文字
8	serialno_a	string	シリアル番号 A。	64 文字
9	serialno_b	string	シリアル番号 B。	64 文字
48	site_address_a	string	サイト住所 A。	128 文字
49	site_address_b	string	サイト住所 B。	128 文字
50	site_address_c	string	サイト住所 C。	128 文字
51	site_city	string	サイト市区町村。	128 文字
53	site_country	string	サイト国。	64 文字
56	site_notes	string	サイトメモ。	SQL データベースでは 65535 文字
55	site_rack	string	サイトラック位置。	128 文字
52	site_state	string	サイト都道府県/州。	64 文字
54	site_zip	string	サイト ZIP/郵便番号。	64 文字
16	software	string	ソフトウェア。	255 文字
18	software_app_a	string	ソフトウェアアプリケーション A。	64 文字
19	software_app_b	string	ソフトウェアアプリケーション B。	64 文字
20	software_app_c	string	ソフトウェアアプリケーション C。	64 文字
21	software_app_d	string	ソフトウェアアプリケーション D。	64 文字
22	software_app_e	string	ソフトウェアアプリケーション E。	64 文字
17	software_full	string	ソフトウェア詳細。	SQL データベースでは 65535 文字
10	tag	string	タグ。	64 文字
1	type	string	タイプ。	64 文字
2	type_full	string	タイプ詳細。	64 文字
35	url_a	string	URL A。	2048 文字
36	url_b	string	URL B。	2048 文字
37	url_c	string	URL C。	2048 文字
31	vendor	string	ベンダー。	64 文字

ホストタグ

ホストタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	ホストタグ名。 プロパティの動作: - 必須

プロパティ	型	説明
value	string	ホストタグの値。 プロパティの動作: - 読み取り専用。
automatic	integer	ホストタグの種類。 指定可能な値: 0 - *(デフォルト)* 手動（ユーザーが作成したタグ）; 1 - 自動（ローレベルディスカバリによって作成されたタグ） プロパティの動作: - 読み取り専用
object	integer	タグの継承元オブジェクトの種類。 指定可能な値: 0 - テンプレート。 プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

host.create

説明

`object host.create(object/array hosts)`

このメソッドを使用すると、新しいホストを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するホスト。

このメソッドは、標準のホストプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
groups	object/array	ホストを追加するホストグループ。 ホストグループでは、`groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
interfaces	object/array	ホスト用に作成されるインターフェース。
tags	object/array	ホストタグ。
templates	object/array	ホストにリンクするテンプレート。 テンプレートでは、`templateid` プロパティのみを定義する必要があります。
macros	object/array	ホスト用に作成されるユーザーマクロ。
inventory	object	ホストインベントリのプロパティ。

戻り値

(object) hostids プロパティの下に生成されたホストの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストの順序と一致します。

例

ホストの作成

IP インターフェースとタグを持つ「Linux server」というホストを作成し、グループに追加し、テンプレートをリンクし、ホストインベントリに MAC アドレスを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
                "value": "123321"
            },
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

SNMP インターフェースを持つホストの作成

詳細付きの SNMPv3 インターフェースを持つ”SNMP host” というホストを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "SNMP host",
        "interfaces": [
            {
                "type": 2,
                "main": 1,
                "useip": 1,
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": 3,
                    "bulk": 0,
                    "securityname": "mysecurityname",
                    "contextname": "",
                    "securitylevel": 1
                }
            }
        ],
        "groups": [
            {
                "groupid": "4"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

PSK 暗号化を使用したホストの作成

PSK 暗号化を設定した「PSK host」というホストを作成します。
ホストは事前に PSK を使用するよう設定されている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "PSK host",
        "interfaces": [
            {
                "type": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050",
                "useip": 1,
                "main": 1
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "tls_accept": 2,
        "tls_connect": 2,
        "tls_psk_identity": "PSK 001",
        "tls_psk": "1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

プロキシで監視するホスト情報の作成

ID "1" のプロキシで監視するホスト情報を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

プロキシグループで監視するホストの作成

ID "1" のプロキシグループで監視するホストを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

```
{
"jsonrpc": "2.0",
"method": "host.create",
"params": {
"host": " プロキシグループで監視するホスト",
"groups": [
{
"groupid": "2"
}
],
"monitored_by": 2,
"proxy_groupid": 1
},
"id": 1
}
```

レスポンス:

```
{
"jsonrpc": "2.0",
"result": {
"hostids": [
"10592"
]
```

```
},
"id": 1
}
```

関連項目

- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース
- ホストインベントリ
- ホストタグ
- プロキシ
- プロキシグループ

ソース

CHost::create() in *ui/include/classes/api/services/CHost.php*.

host.delete

説明

`object host.delete(array hosts)`

このメソッドを使用すると、ホストを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するホストの ID。

戻り値

(object) hostids プロパティの下にある削除されたホストの ID を含むオブジェクトを返します。

例

複数のホストの削除

2 つのホストを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

ソース

CHost::delete() in *ui/include/classes/api/services/CHost.php*.

host.get

説明

`integer/array host.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってホストを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
groupids	ID/array	指定したグループに属するホストのみを返します。
dserviceids	ID/array	指定したディスカバリサービスに関連するホストのみを返します。
graphids	ID/array	指定したグラフを持つホストのみを返します。
hostids	ID/array	指定したホスト ID を持つホストのみを返します。
httptestids	ID/array	指定した Web 監視を持つホストのみを返します。
interfaceids	ID/array	指定したインターフェースを使用するホストのみを返します。
itemids	ID/array	指定したアイテムを持つホストのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるホストのみを返します。
monitored_hosts	flag	監視対象のホストのみを返します。
proxyids	ID/array	指定したプロキシによって監視されているホストのみを返します。
proxy_groupids	ID/array	指定したプロキシグループによって監視されているホストのみを返します。
templated_hosts	flag	ホストとテンプレートの両方を返します。
templateids	ID/array	指定したテンプレートにリンクされているホストのみを返します。
triggerids	ID/array	指定したトリガーを持つホストのみを返します。
with_items	flag	アイテムを持つホストのみを返します。 `with_monitored_items` および `with_simple_graph_items` パラメータを上書きします。
with_item_prototypes	flag	アイテムのプロトタイプを持つホストのみを返します。 `with_simple_graph_item_prototypes` パラメータを上書きします。
with_simple_graph_item_prototypes	flag	作成が有効で、かつ数値型の情報タイプを持つアイテムのプロトタイプを持つホストのみを返します。
with_graphs	flag	グラフを持つホストのみを返します。
with_graph_prototypes	flag	グラフのプロトタイプを持つホストのみを返します。
with_httptests	flag	Web 監視を持つホストのみを返します。 `with_monitored_httptests` パラメータを上書きします。
with_monitored_httptests	flag	有効な Web 監視を持つホストのみを返します。
with_monitored_items	flag	有効なアイテムを持つホストのみを返します。 `with_simple_graph_items` パラメータを上書きします。
with_monitored_triggers	flag	有効なトリガーを持つホストのみを返します。トリガーで使用されるすべてのアイテムも有効である必要があります。
with_simple_graph_items	flag	数値型の情報タイプを持つアイテムを持つホストのみを返します。
with_triggers	flag	トリガーを持つホストのみを返します。 `with_monitored_triggers` パラメータを上書きします。
withProblemsSuppressed	boolean	`true` に設定すると、抑制された障害を持つホストのみを返します。

Parameter	Type	Description
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
severities	integer/array	指定した深刻度の障害のみを持つホストを返します。problem オブジェクトがトリガーである場合にのみ適用されます。
tags	object/array	指定したタグを持つホストのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列はすべてのホストを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
inheritedTags	boolean	指定した `tags` を、リンクされているすべてのテンプレートにも持つホストを返します。 指定可能な値: `true` - リンクされたテンプレートも指定した `tags` を持っている必要があります; `false` - *(デフォルト)* リンクされたテンプレートのタグは無視されます。
selectDiscoveryData	query	ホストディスカバリオブジェクトのデータを含む `discoveryData` プロパティを返します。ホストディスカバリオブジェクトは、ディスカバリされたホストを、そのホストの検出元であるホストのプロトタイプに関連付けます。 以下のプロパティを持ちます: `host` - (`string`) ホストの ID; `parent_hostid` - (`string`) ホストの作成元となったホストのプロトタイプの ID; `status` - (`int`) ホストディスカバリのステータス: 0 - *(デフォルト)* ホストはディスカバリされている, 1 - ホストはもうディスカバリされていない; `ts_delete` - (`timestamp`) すでにディスカバリされなくなったホストが削除される時刻; `ts_disable` - (`timestamp`) すでにディスカバリされなくなったホストが無効化される時刻; `disable_source` - (`int`) ホストが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(デフォルト)* 自動的に無効化, 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	ホストを作成したローレベルディスカバリルールを含む`discoveryRule` プロパティを返します（VMware 監視ではホストのプロトタイプから作成）。
selectDiscoveryRules	query	ホストの LLD ルールを含む`discoveryRules` プロパティを返します。 `count` をサポートします。
selectGraphs	query	ホストのグラフを含む`graphs` プロパティを返します。 `count` をサポートします。
selectHostGroups	query	ホストが属するホストグループのデータを含む`hostgroups` プロパティを返します。
selectHttpTests	query	ホストの Web シナリオを含む`httpTests` プロパティを返します。 `count` をサポートします。

Parameter	Type	Description
selectInterfaces	query	ホストインターフェースを含むinterfaces プロパティを返します。 count をサポートします。
selectInventory	query	ホストインベントリデータを含むinventory プロパティを返します。
selectItems	query	ホストのアイテムを含むitems プロパティを返します。 count をサポートします。
selectMacros	query	ホストマクロを含むmacros プロパティを返します。
selectParentTemplates	query	ホストにリンクされているテンプレートを含む parentTemplates プロパティを返します。 Template オブジェクトのフィールドに加えて、link_type - (integer) テンプレートがホストにリンクされる方法も含まれます。 指定可能な値: 0 - *(デフォルト)* 手動でリンク; 1 - LLD によって自動的にリンク。 count をサポートします。
selectDashboards	query	dashboards プロパティを返します。 count をサポートします。
selectTags	query	ホストタグを含むtags プロパティを返します。
selectInheritedTags	query	ホストにリンクされているすべてのテンプレート上のタグを含むinheritedTags プロパティを返します。
selectTriggers	query	ホストのトリガーを含むtriggers プロパティを返します。 count をサポートします。
selectValueMaps	query	ホストの値マップを含むvaluemaps プロパティを返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 text data type のプロパティはサポートしません。 追加のプロパティをサポートします: ホストインターフェース のプロパティ。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectParentTemplates - 結果は host でソートされます; selectInterfaces; selectItems - name でソートされます; selectDiscoveryRules - name でソートされます; selectTriggers - description でソートされます; selectGraphs - name でソートされます; selectDashboards - name でソートされます。
search	object	指定したパターンに一致する結果を返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。 string および text data type のプロパティのみをサポートします。 追加のプロパティをサポートします: ホストインターフェース のプロパティ。

Parameter	Type	Description
searchInventory	object	指定したパターンに一致するインベントリデータを持つホストを返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。
sortfield	string/array	string および text data type のプロパティのみをサポートします。 指定したプロパティで結果をソートします。 指定可能な値: hostid, host, name, status。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	ホストのローレベルディスカバリルールを含むdiscoveries プロパティを返します。 count をサポートします。 このクエリは非推奨です。代わりに selectDiscoveryRules を使用してください。
selectHostDiscovery	query	ホストディスカバリオブジェクトのデータを含む hostDiscovery プロパティを返します。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

名前によるデータの取得

"Zabbix server" と "Linux server" という名前の 2 つのホストに関するすべてのデータを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "proxyid": "0",
            "host": "Zabbix server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Zabbix server",
            "flags": "0",
            "description": "The Zabbix monitoring server.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```


ホストグループの取得

ホスト”Zabbix server” が所属するホストグループ取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "hostgroups": [
                {
                    "groupid": "2",
                    "name": "Linux servers",
                    "flags": "0",
                    "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
                },
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ]
        }
    ],
    "id": 1
}
```

リンク先テンプレートの取得

ホスト”10084” にリンクしているテンプレートの ID および名前を取得します。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートでホストの取得

テンプレート"10001"(*Linux by Zabbix agent*) がリンクされているホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

ホストインベントリのデータで検索

ホストインベントリの"OS" フィールドに"Linux" が含まれるホストを検索します

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

ホストタグによる検索

タグ”host-name” が”linux-server” に等しいホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

ホストレベルだけでなく、リンクされた親テンプレートにもこれらのタグを持つホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

タグ”component” を持つホストを除外してホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
```

```
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "tags": [
                {
                    "tag": "class",
                    "value": "os",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "linux",
                    "automatic": "0"
                }
            ]
        },
        {
            "hostid": "10629",
            "tags": [
                {
                    "tag": "class",
                    "value": "application",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "browser",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

タグとテンプレートタグでホストの検索

タグと親テンプレートにリンクされているすべてのタグでホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
```

```
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10502",
            "name": "Desktop",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ],
            "inheritedTags": [
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ]
        }
    ],
    "id": 1
}
```

障害の深刻度でホストの検索

”Disaster” の障害を持つホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
```

```
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

”Average” と”High” の障害を持つホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

ホストのアイテムの取得

”Zabbix server” ホストのアイテムを取得します。リクエストは、アイテムのプロパティ itemid、name、status に限定されています。レスポンスサイズが大きいため、この例ではアイテムの一部のみを示しています。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
```

```
        {
          "itemid": "42227",
          "name": "Host name of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix agent ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Maximum number of open file descriptors",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maximum number of processes",
          "status": "0"
        },
        {
          "itemid": "69869",
          "name": "Interface enp0s3: Inbound packets discarded",
          "status": "0"
        },
        {
          "itemid": "69870",
          "name": "Interface enp0s3: Inbound packets with errors",
          "status": "0"
        },
        {
          "itemid": "69871",
          "name": "Interface enp0s3: Bits received",
          "status": "0"
        },
        {
          "itemid": "69872",
          "name": "Interface enp0s3: Outbound packets discarded",
          "status": "0"
        },
        {
          "itemid": "69873",
          "name": "Interface enp0s3: Outbound packets with errors",
          "status": "0"
        },
        {
          "itemid": "69874",
          "name": "Interface enp0s3: Bits sent",
          "status": "0"
        },
        {
          "itemid": "42253",
          "name": "Number of processes",
          "status": "0"
        },
        {
          "itemid": "42254",
```

```
            "name": "Number of running processes",
            "status": "0"
          },
          {
            "itemid": "42255",
            "name": "System boot time",
            "status": "0"
          }
        ]
      }
    ],
    "id": 1
}
```

関連項目

- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース
- プロキシ
- プロキシグループ

ソース

CHost::get() in *ui/include/classes/api/services/CHost.php*.

host.massadd

説明

`object host.massadd(object parameters)`

このメソッドを使用すると、指定したすべてのホストに対して、関連する複数のオブジェクトを同時に追加できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 更新するホストの ID と、すべてのホストに追加するオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
hosts	object/array	更新するホストです。 ホストでは `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	指定したホストに追加するホストグループです。 ホストグループでは `groupid` プロパティのみを定義する必要があります。
interfaces	object/array	指定したホスト用に作成するホストインターフェースです。
macros	object/array	指定したホスト用に作成するユーザーマクロです。
templates	object/array	指定したホストにリンクするテンプレートです。 テンプレートでは `templateid` プロパティのみを定義する必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

マクロの追加

2 つのホストに 2 つの新しいマクロを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Test description"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

関連項目

- host.update
- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース

ソース

CHost::massAdd() in *ui/include/classes/api/services/CHost.php*.

host.massremove

説明

`object host.massremove(object parameters)`

このメソッドを使用すると、複数のホストから関連オブジェクトを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新するホストの ID と、削除するオブジェクトを含むパラメータ。

パラメータ	型	説明
hostids	ID/array	更新するホストの ID。 パラメータの動作: - 必須
groupids	ID/array	指定したホストを削除するホストグループの ID。
interfaces	object/array	指定したホストから削除するホストインターフェース。 ホストインターフェースオブジェクトでは、ip、dns、port プロパティのみを定義する必要があります。
macros	string/array	指定したホストから削除するユーザーマクロ。
templateids	ID/array	指定したホストからリンク解除するテンプレートの ID。
templateids_clear	ID/array	指定したホストからリンク解除してクリアするテンプレートの ID。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

テンプレートのリンク解除

2 つのホストからテンプレートのリンクを解除し、テンプレート化されたすべてのエンティティを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

関連項目

- host.update
- ユーザーマクロ
- ホストインターフェース

ソース

CHost::massRemove() in *ui/include/classes/api/services/CHost.php*.

host.update

説明

`object host.update(object/array hosts)`

このメソッドを使用すると、既存のホストを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するホストのプロパティ。

各ホストに対して `hostid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、それ以外は変更されません。

ただし、ホストの技術名を更新すると、その値を使用してホストの表示名も更新されます（別途指定されていない場合）。

このメソッドは、標準のホストプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	ホストが所属する現在のホストグループを置き換えるためのホストグループ。 リクエストに含まれていないすべてのホストグループはリンク解除されます。 ホストグループでは `groupid` プロパティのみを定義する必要があります。
interfaces	object/array	現在のホストインターフェースを置き換えるためのホストインターフェース。 リクエストに含まれていないすべてのインターフェースは削除されます。
tags	object/array	現在のホストタグを置き換えるためのホストタグ。 リクエストに含まれていないすべてのタグは削除されます。
inventory	object	ホストインベントリ のプロパティ。
macros	object/array	現在のユーザーマクロを置き換えるためのユーザーマクロ。 リクエストに含まれていないすべてのマクロは削除されます。
templates	object/array	現在リンクされているテンプレートを置き換えるためのテンプレート。 リクエストに含まれていないすべてのテンプレートはリンク解除のみ行われます。 テンプレートでは `templateid` プロパティのみを定義する必要があります。
templates_clear	object/array	ホストからリンク解除してクリアするためのテンプレート。 テンプレートでは `templateid` プロパティのみを定義する必要があります。

戻り値

(`object`) `hostids` プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

ホストの有効化

ホストの監視を有効にする、つまりステータスを 0 にします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
```

```
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

複数ホストの有効化

2 つのホストの監視を有効にします。つまり、ステータスを”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "id": 1
}
```

テンプレートのリンク削除

ホストから 2 つのテンプレートのリンクを削除してクリアします。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
```

```
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

ホストマクロの更新

すべてのホストマクロを新しい 2 つのマクロに置き換えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Updated description"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

ホストインベントリの更新

インベントリモードの変更とロケーションの追加をします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
```

```
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

ホストタグの更新

すべてのホストタグを新しいものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

検出されたホストマクロの更新

作成した検出ルールの” 自動” マクロを” 手動” に変換し、その値を”new-value” に変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

ホストの暗号化の更新

ホスト”10590” を更新して、ホストから Zabbix サーバーへの接続に対してのみ PSK 暗号化を使用するようにし、PSK ID と PSK キーを変更します。
なお、このホストはPSK を使用するよう事前設定 されている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10590",
        "tls_connect": 1,
        "tls_accept": 2,
        "tls_psk_identity": "PSK 002",
        "tls_psk": "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

関連項目

- host.massadd
- host.massremove
- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース
- ホストインベントリ
- ホストタグ
- プロキシ
- プロキシグループ

ソース

CHost::update() in *ui/include/classes/api/services/CHost.php*.

ホストインターフェース

このクラスでは、ホストインターフェースにアクセスできます。

オブジェクトリファレンス:

- ホストインターフェース
  - 詳細

利用可能なメソッド:

- hostinterface.get - ホストインターフェースを取得する

ホストインターフェースの作成、更新、削除には、ホストメソッドを使用します。

Host interface オブジェクト

次のオブジェクトは、`hostinterface`API に直接関連しています。

ホストインターフェース

ホストインターフェースオブジェクトには、次のプロパティがあります。

プロパティ	型	説明
interfaceid	ID	インターフェースの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
available	integer	ホストインターフェースの可用性。 指定可能な値: 0 - *(デフォルト)* 不明; 1 - 使用可能; 2 - 使用不可。 プロパティの動作: - 読み取り専用
hostid	ID	インターフェースが属するホストの ID。 プロパティの動作: - 固定 - 作成操作では必須
type	integer	インターフェースのタイプ。 指定可能な値: 1 - エージェント; 2 - SNMP; 3 - IPMI; 4 - JMX。 プロパティの動作: - 作成操作では必須
ip	string	インターフェースで使用する IP アドレス。 DNS 経由で接続する場合は空にできます。 プロパティの動作: - 作成操作では必須
dns	string	インターフェースで使用する DNS 名。 IP 経由で接続する場合は空にできます。 プロパティの動作: - 作成操作では必須

プロパティ	型	説明
port	string	インターフェースで使用するポート番号。 ユーザーマクロを含めることができます。 プロパティの動作: - 作成操作では必須
useip	integer	IP 経由で接続するかどうか。 指定可能な値: 0 - ホストの DNS 名を使用して接続; 1 - ホストの IP アドレスを使用して接続。 プロパティの動作: - 作成操作では必須
main	integer	ホスト上でインターフェースをデフォルトとして使用するかどうか。同じタイプのインターフェースは、ホスト上で 1 つだけデフォルトに設定できます。 指定可能な値: 0 - デフォルトではない; 1 - デフォルト。 プロパティの動作: - 作成操作では必須
details	object	インターフェースの追加の詳細オブジェクト。 プロパティの動作: - type が「SNMP」に設定されている場合は必須
disable_until	timestamp	使用できないホストインターフェースの次回のポーリング時刻。 プロパティの動作: - 読み取り専用
error	string	ホストインターフェースが使用できない場合のエラーテキスト。 プロパティの動作: - 読み取り専用
errors_from	timestamp	ホストインターフェースが使用できなくなった時刻。 プロパティの動作: - 読み取り専用

詳細

details オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
version	integer	SNMP インターフェースのバージョン。 指定可能な値: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. プロパティの動作: - 必須
bulk	integer	bulk SNMP リクエストを使用するかどうか。 指定可能な値: 0 - bulk リクエストを使用しない; 1 - (デフォルト) - bulk リクエストを使用する。

プロパティ	Type	説明
community	string	SNMP コミュニティ。SNMPv1 および SNMPv2 インターフェースでのみ使用されます。 プロパティの動作: - `version` が”SNMPv1” または”SNMPv2c” に設定されている場合は 必須
max_repetitions	integer	ネイティブ SNMP bulk リクエスト (GetBulkRequest-PDUs) の最大繰り返し値。 SNMPv2 および v3 の `discovery[]` アイテムと `walk[]` アイテムでのみ使用されます。 デフォルト: 10.
securityname	string	SNMPv3 セキュリティ名。SNMPv3 インターフェースでのみ使用されます。
securitylevel	integer	SNMPv3 セキュリティレベル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - (デフォルト) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	SNMPv3 認証パスフレーズ。SNMPv3 インターフェースでのみ使用されます。
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。SNMPv3 インターフェースでのみ使用されます。
authprotocol	integer	SNMPv3 認証プロトコル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - (デフォルト) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - (デフォルト) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	SNMPv3 コンテキスト名。SNMPv3 インターフェースでのみ使用されます。

hostinterface.get

説明

`integer/array hostinterface.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってホストインターフェースを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	Type	説明
hostids	ID/array	指定したホストで使用されているホストインターフェースのみを返します。
interfaceids	ID/array	指定した ID を持つホストインターフェースのみを返します。
itemids	ID/array	指定したアイテムで使用されているホストインターフェースのみを返します。
triggerids	ID/array	指定したトリガー内のアイテムで使用されているホストインターフェースのみを返します。
selectItems	query	インターフェースを使用しているアイテムを含む`items` プロパティを返します。 `count` をサポートします。
selectHosts	query	インターフェースを使用しているホストの配列を含む`hosts` プロパティを返します。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectItems`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `interfaceid`, `dns`, `ip`。
countOutput	boolean	これらのパラメータについては、reference commentary を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストインターフェースの取得

ホスト"30057" で使用されているインターフェースに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "interfaceid": "50039",
            "hostid": "30057",
            "main": "1",
            "type": "1",
            "useip": "0",
            "ip": "",
            "dns": "localhost",
            "port": "10050",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": []
        },
        {
            "interfaceid": "55082",
            "hostid": "30057",
            "main": "1",
            "type": "2",
            "useip": "1",
            "ip": "127.0.0.1",
            "dns": "",
            "port": "161",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": {
                "version": "2",
                "bulk": "0",
                "community": "{$SNMP_COMMUNITY}",
                "max_repetitions": "10"
            }
        }
    ],
    "id": 1
}
```


参照

- Host
- Item

ソース

CHostInterface::get() in *ui/include/classes/api/services/CHostInterface.php*.

ホストグループ

このクラスはホストグループを操作するために設計されています。

オブジェクト参照:

- ホストグループ

利用可能なメソッド:

- hostgroup.create - 新しいホストグループを作成
- hostgroup.delete - ホストグループを削除
- hostgroup.get - ホストグループを取得
- hostgroup.massadd - 関連オブジェクトをホストグループに追加
- hostgroup.massremove - 関連オブジェクトをホストグループから削除

- hostgroup.propagate - 権限とタグフィルターをホストグループのサブグループに伝播
- hostgroup.update - ホストグループを更新

Host group オブジェクト

次のオブジェクトは、hostgroupAPI に直接関連しています。

ホストグループ

ホストグループオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
groupid	ID	ホストグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ホストグループの名前。 プロパティの動作: - 作成操作では必須
flags	integer	ホストグループの生成元。 使用可能な値: 0 - 通常のホストグループ。 4 - ディスカバリされたホストグループ。 プロパティの動作: - 読み取り専用
uuid	string	既存のホストグループにインポートしたホストグループを関連付けるために使用される汎用一意識別子。指定されていない場合は自動生成されます。

hostgroup.create

説明

`object hostgroup.create(object/array hostGroups)`

このメソッドを使用すると、新しいホストグループを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するホストグループ。

このメソッドは、標準のホストグループプロパティを持つホストグループを受け付けます。

戻り値

(`object`) `groupids` プロパティの下に作成されたホストグループの ID を含むオブジェクトを返します。
 返される ID の順序は、渡されたホストグループの順序と一致します。

例

ホストグループの作成

”Linux servers” というホストグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107819"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::create() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

説明

`object hostgroup.delete(array hostGroupIds)`

このメソッドを使用すると、ホストグループを削除できます。

次の場合、ホストグループは削除できません。

- そのホストグループにのみ属するホストが含まれている場合;
- internal としてマークされている場合;
- ホストプロトタイプで使用されている場合;
- グローバルスクリプトで使用されている場合;
- 相関条件で使用されている場合。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(`array`) 削除するホストグループの ID。

戻り値

(`object`) `groupids` プロパティの下にある削除されたホストグループの ID を含むオブジェクトを返します。

例

複数のホストグループの削除

2 つのホストグループを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::delete() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

説明

`integer/array hostgroup.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってホストグループを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
graphids	ID/array	指定したグラフを持つホストを含むホストグループのみを返します。
groupids	ID/array	指定したホストグループ ID を持つホストグループのみを返します。
hostids	ID/array	指定したホストを含むホストグループのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるホストグループのみを返します。
triggerids	ID/array	指定したトリガーを持つホストを含むホストグループのみを返します。
with_graphs	boolean	グラフを持つホストを含むホストグループのみを返します。
with_graph_prototypes	boolean	グラフのプロトタイプを持つホストを含むホストグループのみを返します。
with_hosts	boolean	ホストを含むホストグループのみを返します。
with_httptests	boolean	Web 監視を持つホストを含むホストグループのみを返します。 `with_monitored_httptests` パラメータを上書きします。
with_items	boolean	アイテムを持つホストを含むホストグループのみを返します。 `with_monitored_items` および `with_simple_graph_items` パラメータを上書きします。
with_item_prototypes	boolean	アイテムのプロトタイプを持つホストを含むホストグループのみを返します。 `with_simple_graph_item_prototypes` パラメータを上書きします。
with_simple_graph_item_prototypes	boolean	作成が有効で、かつ数値型の情報を持つアイテムのプロトタイプを持つホストを含むホストグループのみを返します。
with_monitored_httptests	boolean	有効な Web 監視を持つホストを含むホストグループのみを返します。
with_monitored_hosts	boolean	監視対象のホストを含むホストグループのみを返します。

パラメータ	型	説明
with_monitored_items	boolean	有効なアイテムを持つホストを含むホストグループのみを返します。 with_simple_graph_items パラメータを上書きします。
with_monitored_triggers	boolean	有効なトリガーを持つホストを含むホストグループのみを返します。トリガーで使用されるすべてのアイテムも有効である必要があります。
with_simple_graph_items	boolean	数値型のアイテムを持つホストを含むホストグループのみを返します。
with_triggers	boolean	トリガーを持つホストを含むホストグループのみを返します。 with_monitored_triggers パラメータを上書きします。
selectDiscoveryRules	query	ホストグループを検出した LLD ルールを含むdiscoveryRules プロパティを返します。
selectDiscoveryData	query	ホストグループ検出オブジェクトを含む discoveryData プロパティを返します。 各ホストグループ検出オブジェクトは、検出されたホストグループにリンクされたホストグループのプロトタイプであり、以下のプロパティを持ちます。 parent_group_prototypeid - (ID) ホストグループの検出元となったホストグループのプロトタイプの ID。 name - (string) ホストグループのプロトタイプ名。 ts_delete - (timestamp) これ以上検出されなくなったホストグループが削除される時刻。 status - (int) ホストグループの検出ステータス: 0 - *(デフォルト)* ホストグループは検出されています。 1 - ホストグループはもう検出されていません。
selectHostPrototypes	query	このホストグループを検出したホストのプロトタイプを含むhostPrototypes プロパティを返します。
selectHosts	query	ホストグループに属するホストを含むhosts プロパティを返します。 count をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます。 selectHosts - 結果は host でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: groupid, name。
countOutput	boolean	これらのパラメータはリファレンス解説 で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroupDiscoveries	query	ホストグループ検出オブジェクトを含む groupDiscoveries プロパティを返します。 各ホストグループ検出オブジェクトは、検出されたホストグループにリンクされたホストグループのプロトタイプです。 このクエリは 非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用された場合は、取得したオブジェクトの数

例

名前によるデータの取得

”Zabbix servers” と”Linux servers” という名前の 2 つのホストグループに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

関連項目

- ホスト

ソース

CHostGroup::get() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

説明

`object hostgroup.massadd(object parameters)`

このメソッドを使用すると、指定したすべてのホストグループに対して、関連する複数のオブジェクトを同時に追加できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

更新するホストグループの ID と、すべてのホストグループに追加するオブジェクトを含む（object）パラメータです。

このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
groups	object/array	更新対象のホストグループ。 ホストグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
hosts	object/array	すべてのホストグループに追加するホスト。 ホストでは、hostid プロパティのみを定義する必要があります。

戻り値

（object）groupids プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストをホストグループに追加する

2 つのホストを ID が 5 と 6 のホストグループに追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト

ソース

CHostGroup::massAdd() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

説明

`object hostgroup.massremove(object parameters)`

このメソッドを使用すると、複数のホストグループから関連オブジェクトを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 更新するホストグループの ID と、削除するオブジェクトを含むパラメータです。

パラメータ	Type	説明
groupids	ID/array	更新するホストグループの ID。 パラメータの動作: - 必須
hostids	ID/array	すべてのホストグループから削除するホストの ID。

戻り値

(`object`) `groupids` プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストグループからホストを削除する

指定したホストグループから 2 つのホストを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
```

```
    "id": 1
}
```

ソース

CHostGroup::massRemove() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

説明

`object hostgroup.propagate(object parameters)`

このメソッドを使用すると、ホストグループのすべてのサブグループに権限とタグフィルターを適用できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、User rolesを参照してください。

パラメータ

(`object`) 希望する出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
groups	object/array	伝播するホストグループ。 ホストグループでは `groupid` プロパティが定義されている必要があります。 パラメータの動作: - 必須
permissions	boolean	権限を伝播するには”true” に設定します。 パラメータの動作: - `tag_filters` が設定されていない場合は 必須
tag_filters	boolean	タグフィルターを伝播するには”true” に設定します。 パラメータの動作: - `permissions` が設定されていない場合は 必須

戻り値

(`object`) は、`groupids` プロパティの下で、伝播されたホストグループの ID を含むオブジェクトを返します。

例

ホストグループの権限とタグフィルターをそのサブグループに伝達します。

ホストグループの権限とタグフィルターをそのサブグループに伝達します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

参照

- hostgroup.update
- hostgroup.massadd
- Host

ソース

CHostGroup::propagate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

説明

`object hostgroup.update(object/array hostGroups)`

このメソッドを使用すると、既存のホストグループを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するホストグループのプロパティ。

各ホストグループでは groupid プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、それ以外は変更されません。

戻り値

(object) groupids プロパティの下で更新された host gtoup の ID を含むオブジェクトを返します。

例

ホストグループの名前の変更

ホストグループの名前を”Linux hosts” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

複数のホストグループの名前変更

2 つのホストグループの名前にリージョン情報を含めるように名前を変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::update() in *ui/include/classes/api/services/CHostGroup.php*.

ホストプロトタイプ

このクラスは、ホストプロトタイプを操作するために設計されています。

オブジェクト参照:

- ホストプロトタイプ
- グループリンク
- グループプロトタイプ
- ホストプロトタイプタグ
- カスタムインターフェース
    - カスタムインターフェースの詳細

利用可能なメソッド:

- hostprototype.create - 新しいホストプロトタイプを作成
- hostprototype.delete - ホストプロトタイプを削除
- hostprototype.get - ホストプロトタイプを取得
- hostprototype.update - ホストプロトタイプを更新

Host prototype オブジェクト

次のオブジェクトは、`hostprototype`API に直接関連しています。

ホストプロトタイプ

ホストプロトタイプオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
hostid	ID	ホストプロトタイプの ID。 Property behavior: - *read-only* - 更新操作では *required*
host	string	ホストプロトタイプの技術名。 Property behavior: - 作成操作では *required* - 継承オブジェクトでは *read-only*
name	string	ホストプロトタイプの表示名。 デフォルト: `host` プロパティの値。 Property behavior: - 継承オブジェクトでは *read-only*
status	integer	ホストプロトタイプのステータス。 使用可能な値: 0 - *(default)* 監視対象ホスト; 1 - 非監視ホスト。
flags	integer	ホストプロトタイプの生成元。 使用可能な値: 2 - ホストプロトタイプ; 6 - ディスカバリされたホストプロトタイプ Property behavior: - *read-only*
inventory_mode	integer	ホストインベントリの設定モード。 使用可能な値: -1 - *(default)* 無効; 0 - 手動; 1 - 自動。
templateid	ID	親テンプレートのホストプロトタイプの ID。 Property behavior: - *read-only*
discover	integer	ホストプロトタイプのディスカバリステータス。 使用可能な値: 0 - *(default)* 新しいホストがディスカバリされる; 1 - 新しいホストはディスカバリされず、既存のホストは失われたものとしてマークされる。
custom_interfaces	integer	ホストプロトタイプによって作成されるホストのカスタムインターフェースのソース。 使用可能な値: 0 - *(default)* 親ホストからインターフェースを継承する; 1 - ホストプロトタイプのカスタムインターフェースを使用する。 Property behavior: - 継承オブジェクトでは *read-only*

プロパティ	Type	説明
uuid	string	インポートされたホストプロトタイプを既存のものにリンクするために使用される汎用一意識別子。指定されていない場合は自動生成されます。 Property behavior: - ホストプロトタイプがテンプレートに属している場合に *supported*

グループリンク

グループリンクオブジェクトは、ホストプロトタイプをホストグループに関連付けます。以下のプロパティがあります。

プロパティ	型	説明
groupid	ID	ホストグループの ID。 プロパティの動作: - 必須

グループプロトタイプ

グループプロトタイプオブジェクトは、ディスカバリされたホスト用に作成されるグループを定義します。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
group_prototypeid	ID	グループプロトタイプの ID。 Property behavior: - *read-only*
name	string	グループプロトタイプの名前。 Property behavior: - 作成操作では *required*

ホストプロトタイプタグ

ホストプロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	ホストプロトタイプタグ名。 プロパティの動作: - 必須
value	string	ホストプロトタイプタグの値。 プロパティの動作: - 読み取り専用。
object	integer	タグの継承元オブジェクトのタイプ。 指定可能な値: 0 - テンプレート。 プロパティの動作: - 読み取り専用。
objectid	ID	タグの継承元オブジェクトの ID。 プロパティの動作: - 読み取り専用。

カスタムインターフェース

Host prototype object の `custom_interfaces` が「ホストプロトタイプのカスタムインターフェースを使用」に設定されている場合、カスタムインターフェースがサポートされます。カスタムインターフェースオブジェクトには、以下のプロパティがあります。

Property	Type	Description
type	integer	インターフェースタイプ。 指定可能な値: 1 - エージェント; 2 - SNMP; 3 - IPMI; 4 - JMX. Property behavior: - 必須
useip	integer	IP 経由で接続するかどうか。 指定可能な値: 0 - ホストの DNS 名を使用して接続; 1 - ホストの IP アドレスを使用して接続。 Property behavior: - 必須
ip	string	インターフェースで使用される IP アドレス。 マクロを含めることができます。 Property behavior: - `useip` が「ホストの IP アドレスを使用して接続」に設定されている場合は 必須
dns	string	インターフェースで使用される DNS 名。 マクロを含めることができます。 Property behavior: - `useip` が「ホストの DNS 名を使用して接続」に設定されている場合は 必須
port	string	インターフェースで使用されるポート番号。 ユーザーマクロおよび LLD マクロを含めることができます。 Property behavior: - 必須
main	integer	このインターフェースをホスト上のデフォルトとして使用するかどうか。 ホスト上では、あるタイプのインターフェースのうち 1 つだけをデフォルトとして設定できます。 指定可能な値: 0 - デフォルトではない; 1 - デフォルト。 Property behavior: - 必須
details	object	カスタムインターフェースの詳細 のための追加オブジェクト。 Property behavior: - `type` が「SNMP」に設定されている場合は 必須

カスタムインターフェースの詳細

details オブジェクトには、以下のプロパティがあります。

Property	Type	Description
version	integer	SNMP インターフェースのバージョン。 指定可能な値: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. プロパティの動作: - 必須
bulk	integer	bulk SNMP リクエストを使用するかどうか。 指定可能な値: 0 - bulk リクエストを使用しない; 1 - *(デフォルト)* - bulk リクエストを使用する。
community	string	SNMP コミュニティ。 プロパティの動作: - `version` が”SNMPv1” または”SNMPv2c” に設定されている場合は 必須
max_repetitions	integer	ネイティブ SNMP bulk リクエスト (GetBulkRequest-PDUs) の最大繰り返し値。 `discovery[]` および `walk[]` アイテムで、SNMPv2 と v3 の場合にのみ使用されます。 デフォルト: 10。
securityname	string	SNMPv3 セキュリティ名。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合に サポート
securitylevel	integer	SNMPv3 セキュリティレベル。 指定可能な値: 0 - *(デフォルト)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. プロパティの動作: - `version` が”SNMPv3” に設定されている場合に サポート
authpassphrase	string	SNMPv3 認証パスフレーズ。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authNoPriv” または”authPriv” に設定されている場合に サポート
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authPriv” に設定されている場合に サポート
authprotocol	integer	SNMPv3 認証プロトコル。 指定可能な値: 0 - *(デフォルト)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authNoPriv” または”authPriv” に設定されている場合に サポート

Property	Type	Description
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - *(デフォルト)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authPriv” に設定されている場合に サポート
contextname	string	SNMPv3 コンテキスト名。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合に サポート

hostprototype.create

説明

`object hostprototype.create(object/array hostPrototypes)`

このメソッドを使用すると、新しいホストプロトタイプを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するホストプロトタイプ。

このメソッドは、標準のホストプロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
groupLinks	array	ホストプロトタイプ用に作成するグループリンク。 パラメータの動作: - 必須
ruleid	ID	このホストプロトタイプが属する LLD ルールの ID。 パラメータの動作: - 必須
groupPrototypes	array	ホストプロトタイプ用に作成するグループプロトタイプ。
macros	object/array	ホストプロトタイプ用に作成するユーザーマクロ。
tags	object/array	ホストプロトタイプタグ。
interfaces	object/array	ホストプロトタイプのカスタムインターフェース。
templates	object/array	ホストプロトタイプにリンクするテンプレート。 テンプレートでは、`templateid` プロパティのみを定義する必要があります。

戻り値

`(object)` `hostids` プロパティの下で作成されたホストプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストプロトタイプの順序と一致します。

例

ホストプロトタイプの作成

グループプロトタイプ”{#HV.NAME}”、タグペア”datacenter”: ”{#DATACENTER.NAME}”、カスタム SNMPv2 インターフェース 127.0.0.1:161（コミュニティ {$SNMP_COMMUNITY}）を持つ LLD ルール”23542” 上にホストプロトタイプ”{#VM.NAME}” を作成します。ホストグループ”2” にリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.create",
    "params": {
        "host": "{#VM.NAME}",
        "ruleid": "23542",
        "custom_interfaces": "1",
        "groupLinks": [
            {
                "groupid": "2"
            }
        ],
        "groupPrototypes": [
            {
                "name": "{#HV.NAME}"
            }
        ],
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            }
        ],
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

関連項目

- グループリンク
- グループプロトタイプ

- ホストプロトタイプタグ
- カスタムインターフェース
- ユーザーマクロ

ソース

CHostPrototype::create() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

説明

`object hostprototype.delete(array hostPrototypeIds)`

このメソッドを使用すると、ホストプロトタイプを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するホストのプロトタイプの ID。

戻り値

(object) hostids プロパティの下にある削除されたホストのプロトタイプの ID を含むオブジェクトを返します。

例

複数のホストのプロトタイプの削除

2 つのホストのプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

ソース

CHostPrototype::delete() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

説明

`integer/array hostprototype.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってホストプロトタイプを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
hostids	ID/array	指定した ID を持つホストプロトタイプのみを返します。
discoveryids	ID/array	指定した LLD ルールに属するホストプロトタイプのみを返します。
inherited	boolean	true に設定した場合、テンプレートから継承されたアイテムのみを返します。
selectDiscoveryData	query	ホストプロトタイプのディスカバリオブジェクトデータを含む discoveryData プロパティを返します。ホストプロトタイプのディスカバリオブジェクトは、検出されたホストプロトタイプを、それが検出された元のホストプロトタイプに関連付けます。 以下のプロパティがあります。 host - (string) ホストの ID。 parent_hostid - (string) ホストプロトタイプの作成元となったホストプロトタイプの ID。 status - (int) ホストプロトタイプのディスカバリステータス。 0 - *(デフォルト)* ホストプロトタイプは検出されています。 1 - ホストプロトタイプはすでに検出されていません。 ts_delete - (timestamp) すでに検出されなくなったホストプロトタイプが削除される時刻。 ts_disable - (timestamp) すでに検出されなくなったホストプロトタイプが無効化される時刻。 disable_source - (int) ホストプロトタイプが LLD ルールによって無効化されたのか、手動で無効化されたのかを示す指標。 0 - *(デフォルト)* 自動的に無効化。 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	ホストプロトタイプが属する LLD ルールを含むdiscoveryRule プロパティを返します。
selectDiscoveryRulePrototype	query	ホストプロトタイプが属する親 LLD ルールプロトタイプを含むdiscoveryRulePrototype プロパティを返します。
selectInterfaces	query	ホストプロトタイプのカスタムインターフェースを含むinterfaces プロパティを返します。
selectGroupLinks	query	ホストプロトタイプのグループリンクを含むgroupLinks プロパティを返します。
selectGroupPrototypes	query	ホストプロトタイプのグループプロトタイプを含むgroupPrototypes プロパティを返します。
selectInheritedTags	query	リンクされたテンプレート上のタグを含むinheritedTags プロパティを返します。
selectMacros	query	ホストプロトタイプのマクロを含むmacros プロパティを返します。
selectParentHost	query	ホストプロトタイプが属するホストを含むparentHost プロパティを返します。
selectTags	query	ホストプロトタイプのタグを含むtags プロパティを返します。
selectTemplates	query	ホストプロトタイプにリンクされたテンプレートを含むtemplates プロパティを返します。 count をサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: hostid, host, name, status, discovered。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

パラメータ	型	説明
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合は取得したオブジェクトの数

例

LLD ルールからのホストプロトタイプの取得

LLD ルールから、すべてのホストプロトタイプ、そのグループリンク、グループプロトタイプ、およびタグを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "discover": "0",
            "custom_interfaces": "1",
            "uuid": "051a1469d4d045cbbf818fcc843a352e",
            "flags": "2",
            "inventory_mode": "-1",
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "groupLinks": [
                {
```

```
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
                }
            ],
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "available": "0",
                    "error": "",
                    "errors_from": "0",
                    "disable_until": "0",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- グループリンク
- グループプロトタイプ
- ユーザーマクロ

ソース

ui/include/classes/api/services/CHostPrototype.php の CHostPrototype::get()。

hostprototype.update

説明

`object hostprototype.update(object/array hostPrototypes)`

このメソッドを使用すると、既存のホストプロトタイプを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するホストプロトタイプのプロパティ。

各ホストプロトタイプには hostid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

標準のホストプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groupLinks	array	ホストプロトタイプ上の現在のグループリンクを置き換えるグループリンク。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
groupPrototypes	array	ホストプロトタイプ上の既存のグループプロトタイプを置き換えるグループプロトタイプ。 リクエストに含まれていないすべてのグループプロトタイプは削除されます。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
macros	object/array	現在のユーザーマクロを置き換えるユーザーマクロ。 リクエストに含まれていないすべてのマクロは削除されます。
tags	object/array	現在のタグを置き換えるホストプロトタイプタグ。 リクエストに含まれていないすべてのタグは削除されます。
interfaces	object/array	パラメータの動作: - 継承されたオブジェクトでは *read-only* 現在のインターフェースを置き換える、ホストプロトタイプのカスタムインターフェース。 カスタムインターフェースオブジェクトには、そのすべてのパラメータを含める必要があります。 リクエストに含まれていないすべてのインターフェースは削除されます。
templates	object/array	パラメータの動作: - Host prototype object の custom_interfaces が”use host prototypes custom interfaces” に設定されている場合に *supported* - 継承されたオブジェクトでは *read-only* 現在リンクされているテンプレートを置き換えるテンプレート。 テンプレートでは templateid プロパティのみを定義する必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストプロトタイプの ID を含むオブジェクトを返します。

例

ホストプロトタイプの無効化

ホストプロトタイプを無効にします。つまり、ステータスを”1” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

ホストプロトタイプタグの更新

ホストプロトタイプタグを新しいものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

ホストプロトタイプのカスタムインターフェースの更新

継承されたインターフェースをホストプロトタイプのカスタムインターフェースに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "custom_interfaces": "1",
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
```

```
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

関連項目

- グループリンク
- グループプロトタイプ
- ホストプロトタイプタグ
- カスタムインターフェース
- ユーザーマクロ

ソース

CHostPrototype::update() in *ui/include/classes/api/services/CHostPrototype.php*.

マップ

このクラスは、マップを操作するために設計されています。

オブジェクト参照:

- マップ
- マップ要素
    - マップ要素ホスト
    - マップ要素ホストグループ
    - マップ要素マップ
    - マップ要素トリガー
    - マップ要素タグ
    - マップ要素 URL
- マップリンク
    - マップリンクトリガー
    - マップリンクインジケーター
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

利用可能なメソッド:

- map.create - 新しいマップを作成
- map.delete - マップを削除
- map.get - マップを取得
- map.update - マップを更新

Map オブジェクト

次のオブジェクトは、mapAPI に直接関連しています。

マップ

マップオブジェクトには以下のプロパティがあります。

Property	Type	Description
sysmapid	ID	マップの ID。 プロパティの動作: - 読み取り専用 - 更新操作で必須
height	integer	マップの高さ（ピクセル単位）。 プロパティの動作: - 作成操作で必須
name	string	マップ名。 プロパティの動作: - 作成操作で必須
width	integer	マップの幅（ピクセル単位）。 プロパティの動作: - 作成操作で必須
backgroundid	ID	マップの背景として使用される画像の ID。
background_scale	integer	背景画像の比率を維持したスケーリングを有効にするかどうか。 指定可能な値: 0 - スケーリングを無効にする; 1 - *(デフォルト)* 画像のスケーリングを有効にする。
expand_macros	integer	マップ設定時にラベル内のマクロを展開するかどうか。 指定可能な値: 0 - *(デフォルト)* マクロを展開しない; 1 - マクロを展開する。
expandproblem	integer	問題が 1 件だけある要素について、問題のトリガーを表示するかどうか。 指定可能な値: 0 - 常に問題数を表示する; 1 - *(デフォルト)* 問題が 1 件のみの場合は問題のトリガーを表示する。
grid_align	integer	グリッドへの整列を有効にするかどうか。 指定可能な値: 0 - グリッド整列を無効にする; 1 - *(デフォルト)* グリッド整列を有効にする。
grid_show	integer	マップ上にグリッドを表示するかどうか。 指定可能な値: 0 - グリッドを表示しない; 1 - *(デフォルト)* グリッドを表示する。
grid_size	integer	マップグリッドのサイズ（ピクセル単位）。 サポートされる値: 20、40、50、75、100。 デフォルト: 50。
highlight	integer	アイコンのハイライトを有効にするかどうか。 指定可能な値: 0 - ハイライト無効; 1 - *(デフォルト)* ハイライト有効。
iconmapid	ID	マップで使用されるアイコンマップの ID。

Property	Type	Description
label_format	integer	高度なラベルを有効にするかどうか。 指定可能な値: 0 - *(デフォルト)* 高度なラベルを無効にする; 1 - 高度なラベルを有効にする。
label_location	integer	マップ要素ラベルの位置。 指定可能な値: 0 - *(デフォルト)* 下; 1 - 左; 2 - 右; 3 - 上。
label_string_host	string	ホスト要素用のカスタムラベル。 プロパティの動作: - `label_type_host` が”custom” に設定されている場合は必須
label_string_hostgroup	string	ホストグループ要素用のカスタムラベル。 プロパティの動作: - `label_type_hostgroup` が”custom” に設定されている場合は必須
label_string_image	string	画像要素用のカスタムラベル。 プロパティの動作: - `label_type_image` が”custom” に設定されている場合は必須
label_string_map	string	マップ要素用のカスタムラベル。 プロパティの動作: - `label_type_map` が”custom” に設定されている場合は必須
label_string_trigger	string	トリガー要素用のカスタムラベル。 プロパティの動作: - `label_type_trigger` が”custom” に設定されている場合は必須
label_type	integer	マップ要素ラベルのタイプ。 指定可能な値: 0 - ラベル; 1 - IP アドレス; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし。
label_type_host	integer	ホスト要素のラベルタイプ。 指定可能な値: 0 - ラベル; 1 - IP アドレス; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。
label_type_hostgroup	integer	ホストグループ要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。

Property	Type	Description
label_type_image	integer	ホストグループ要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 4 - なし; 5 - カスタム。
label_type_map	integer	マップ要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。
label_type_trigger	integer	トリガー要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。
markelements	integer	最近ステータスが変更されたマップ要素をハイライトするかどうか。 指定可能な値: 0 - *(デフォルト)* 要素をハイライトしない; 1 - 要素をハイライトする。
severity_min	integer	マップに表示されるトリガーの最小深刻度。 サポートされるトリガー深刻度の一覧については、トリガーの `severity` プロパティを参照してください。
show_element_label	integer	デフォルトで要素ラベルをどのように表示するか。 指定可能な値: 0 - 常に表示; 1 - *(デフォルト)* 自動的に非表示。
show_link_label	integer	デフォルトでリンクラベルをどのように表示するか。 指定可能な値: 0 - 常に表示; 1 - *(デフォルト)* 自動的に非表示。
show_unack	integer	問題をどのように表示するか。 指定可能な値: 0 - *(デフォルト)* すべての問題数を表示する; 1 - 未確認の問題数のみを表示する; 2 - 確認済みと未確認の問題数を別々に表示する。
userid	ID	マップの所有者であるユーザーの ID。
private	integer	マップ共有のタイプ。 指定可能な値: 0 - 公開マップ; 1 - *(デフォルト)* 非公開マップ。
show_suppressed	integer	抑制された問題を表示するかどうか。 指定可能な値: 0 - *(デフォルト)* 抑制された問題を非表示にする; 1 - 抑制された問題を表示する。

マップ要素

マップ要素オブジェクトは、マップ上に表示されるオブジェクトを定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
selementid	ID	マップ要素の ID。 プロパティの動作: - 読み取り専用
elements	array	要素データオブジェクト。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合は 必須
elementtype	integer	マップ要素のタイプ。 指定可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - イメージ。 プロパティの動作: - 必須
iconid_off	ID	デフォルト状態で要素を表示するために使用されるイメージの ID。 プロパティの動作: - 必須
areatype	integer	ホストグループ内の個々のホストをどのように表示するか。 指定可能な値: 0 - *(デフォルト)* ホストグループ要素がマップ全体を占有する; 1 - ホストグループ要素は固定サイズになる。
elementsubtype	integer	ホストグループ要素をマップ上でどのように表示するか。 指定可能な値: 0 - *(デフォルト)* ホストグループを単一の要素として表示する; 1 - グループ内の各ホストを個別に表示する。
evaltype	integer	マップ要素タグのフィルタリング条件の評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
height	integer	固定サイズのホストグループ要素の高さ（ピクセル単位）。 デフォルト: 200。
iconid_disabled	ID	無効なマップ要素を表示するために使用されるイメージの ID。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合に サポート
iconid_maintenance	ID	メンテナンス中のマップ要素を表示するために使用されるイメージの ID。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合に サポート
iconid_on	ID	障害のあるマップ要素を表示するために使用されるイメージの ID。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合に サポート
label	string	要素のラベル。

プロパティ	型	説明
label_location	integer	マップ要素ラベルの位置。 指定可能な値: -1 - *(デフォルト)* デフォルトの位置; 0 - 下; 1 - 左; 2 - 右; 3 - 上。
permission	integer	権限レベルのタイプ。 指定可能な値: -1 - なし; 2 - 読み取り専用; 3 - 読み書き。
show_label	integer	要素ラベルの表示方法。 指定可能な値: -1 - *(デフォルト)* マップのデフォルト; 0 - 常に表示; 1 - 自動的に非表示。
sysmapid	ID	この要素が属するマップの ID。 プロパティの動作: - 読み取り専用
urls	array	マップ要素の URL。 マップ要素 URL オブジェクトについては、以下で詳しく説明します。
use_iconmap	integer	ホスト要素に対してアイコンマッピングを使用する必要があるかどうか。 指定可能な値: 0 - アイコンマッピングを使用しない; 1 - *(デフォルト)* アイコンマッピングを使用する。
viewtype	integer	ホストグループ要素の配置アルゴリズム。 指定可能な値: 0 - *(デフォルト)* グリッド。
width	integer	固定サイズのホストグループ要素の幅（ピクセル単位）。 デフォルト: 200。
x	integer	要素の X 座標（ピクセル単位）。 デフォルト: 0。
y	integer	要素の Y 座標（ピクセル単位）。 デフォルト: 0。
zindex	integer	マップ要素の順序付けに使用される値（z-index）。 デフォルト: 0。

マップ要素ホスト

マップ要素のホストオブジェクトは、1 つのホスト要素を定義します。

プロパティ	Type	説明
hostid	ID	ホストの ID。

マップ要素ホストグループ

マップ要素のホストグループオブジェクトは、1 つのホストグループ要素を定義します。

プロパティ	Type	説明
groupid	ID	ホストグループの ID。

マップ要素 Map

マップ要素の Map オブジェクトは、1 つのマップ要素を定義します。

プロパティ	型	説明
sysmapid	ID	マップの ID。

マップ要素トリガー

マップ要素 Trigger オブジェクトは、1 つ以上のトリガー要素を定義します。

プロパティ	Type	説明
triggerid	ID	トリガーの ID。

マップ要素タグ

マップ要素のタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	マップ要素のタグ名。 プロパティの動作: - 必須
operator	integer	マップ要素タグ条件の演算子。 使用可能な値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
value	string	マップ要素のタグ値。

マップ要素 URL

マップ要素 URL オブジェクトは、特定のマップ要素で利用可能なクリック可能なリンクを定義します。以下のプロパティがあります。

プロパティ	型	説明
sysmapelementurlid	ID	マップ要素 URL の ID。 プロパティの動作: - 読み取り専用
name	string	リンクのキャプション。 プロパティの動作: - 必須
url	string	リンク URL。 プロパティの動作: - 必須
selementid	ID	URL が属するマップ要素の ID。

マップリンク

マップリンクオブジェクトは、2 つのマップ要素間のリンクを定義します。
以下のプロパティがあります。

プロパティ	型	説明
linkid	ID	マップリンクの ID。 プロパティの動作: - 読み取り専用
sysmapid	ID	このリンクが属するマップの ID。
selementid1	ID	一方の端でリンクされる最初のマップ要素の ID。 プロパティの動作: - 必須
selementid2	ID	もう一方の端でリンクされる最初のマップ要素の ID。 プロパティの動作: - 必須
drawtype	integer	リンク線の描画スタイル。 指定可能な値: 0 - *(デフォルト)* 実線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	16 進数カラーコードで表した線の色。 デフォルト: 000000。
label	string	リンクラベル。
show_label	integer	リンクラベルの表示方法。 指定可能な値: -1 - *(デフォルト)* マップのデフォルト; 0 - 常に表示; 1 - 自動的に非表示。
indicator_type	integer	リンクインジケーターのタイプを選択します。 指定可能な値: 0 - *(デフォルト)* 静的リンク; 1 - トリガー; 2 - アイテムの値。
linktriggers	array	リンク状態インジケーターとして使用するマップリンクトリガー。 マップリンクトリガーオブジェクトの詳細は以下を参照してください。 プロパティの動作: - `indicator_type` が”trigger” (1) に設定されている場合は 必須。
itemid	ID	アイテムの ID。 プロパティの動作: - `indicator_type` が”item value” (2) に設定されている場合は 必須。
highlights	array	リンク状態インジケーターとして使用するマップリンクのハイライト。 マップリンクインジケーターオブジェクトの詳細は以下を参照してください。 プロパティの動作: - `indicator_type` が”item value” (2) に設定されている場合は 必須。
thresholds	array	リンク状態インジケーターとして使用するマップリンクしきい値。 マップリンクインジケーターオブジェクトの詳細は以下を参照してください。 プロパティの動作: - `indicator_type` が”item value” (2) に設定されている場合は 必須。

プロパティ	型	説明
permission	integer	権限レベルのタイプ。 指定可能な値: -1 - なし; 2 - 読み取り専用; 3 - 読み書き。

マップリンクトリガー

マップリンクトリガーオブジェクトは、トリガーの状態に基づくマップリンクのステータスインジケーターを定義します。次のプロパティがあります。

プロパティ	型	説明
triggerid	ID	リンクインジケーターとして使用されるトリガーの ID。 プロパティの動作: - 必須
color	string	16 進数のカラーコードによるインジケーターの色。 デフォルト: DD0000。
drawtype	integer	インジケーターの描画スタイル。 指定可能な値: 0 - *(デフォルト)* 実線; 2 - 太線; 3 - 点線; 4 - 破線。

マップリンクインジケーター

マップリンクインジケーターオブジェクトは、アイテムの値に基づくマップリンクのステータスインジケーターを定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
pattern	string	比較対象の正規表現です。ハイライトでのみ使用できます。
sortorder	integer	ハイライトのソート順を定義するために使用されます。ハイライトでのみ使用できます。
threshold	string	比較対象の数値です。しきい値でのみ使用できます。
drawtype	integer	インジケーターの描画スタイルです。 使用可能な値: 0 - *(デフォルト)* 線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	16 進数のカラーコードで指定するインジケーターの色です。 デフォルト: DD0000。 プロパティの動作: - 必須

マップ URL

マップ URL オブジェクトは、マップ上の特定のタイプのすべての要素で利用可能なクリック可能なリンクを定義します。次のプロパティがあります。

プロパティ	Type	説明
sysmapurlid	ID	マップ URL の ID。 Property behavior: - *read-only*
name	string	リンクのキャプション。 Property behavior: - *required*
url	string	リンク URL。 Property behavior: - *required*
elementtype	integer	URL が利用可能になるマップ要素のタイプ。 サポートされているタイプの一覧については、マップ要素の `type` プロパティを参照してください。 デフォルト: 0。
sysmapid	ID	URL が属するマップの ID。

マップユーザー

ユーザーに基づくマップ権限の一覧です。以下のプロパティがあります。

プロパティ	型	説明
sysmapuserid	ID	マップユーザーの ID。 プロパティの動作: - 読み取り専用
userid	ID	ユーザーの ID。 プロパティの動作: - 必須
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

マップユーザーグループ

ユーザーグループに基づくマップ権限の一覧です。以下のプロパティがあります。

プロパティ	型	説明
sysmapusrgrpid	ID	マップユーザーグループの ID。 プロパティの動作: - 読み取り専用
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 必須

プロパティ	型	説明
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

マップ図形

マップ図形オブジェクトは、マップ上に表示される幾何学図形（テキストありまたはなし）を定義します。このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
sysmap_shapeid	ID	マップ図形要素の ID。 Property behavior: - *read-only*
type	integer	マップ図形要素のタイプ。 指定可能な値: 0 - 長方形; 1 - 楕円。 新しい図形を作成する場合、このプロパティは必須です。 Property behavior: - *required*
x	integer	図形の X 座標（ピクセル単位）。 デフォルト: 0。
y	integer	図形の Y 座標（ピクセル単位）。 デフォルト: 0。
width	integer	図形の幅（ピクセル単位）。 デフォルト: 200。
height	integer	図形の高さ（ピクセル単位）。 デフォルト: 200。
text	string	図形のテキスト。
font	integer	図形内のテキストのフォント。 指定可能な値: 0 - Georgia, serif 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif 2 - ”Times New Roman”, Times, serif 3 - Arial, Helvetica, sans-serif 4 - ”Arial Black”, Gadget, sans-serif 5 - ”Comic Sans MS”, cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif 8 - Tahoma, Geneva, sans-serif 9 - ”Trebuchet MS”, Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - ”Courier New”, Courier, monospace 12 - ”Lucida Console”, Monaco, monospace デフォルト: 9。

Property	Type	Description
font_size	integer	フォントサイズ（ピクセル単位）。 デフォルト: 11。
font_color	string	フォントの色。 デフォルト: 000000。
text_halign	integer	テキストの水平方向の配置。 指定可能な値: 0 - 中央; 1 - 左; 2 - 右。 デフォルト: 0。
text_valign	integer	テキストの垂直方向の配置。 指定可能な値: 0 - 中央; 1 - 上; 2 - 下。 デフォルト: 0。
border_type	integer	枠線のタイプ。 指定可能な値: 0 - なし; 1 - ──────; 2 - ┅; 3 - – – –。 デフォルト: 0。
border_width	integer	枠線の幅（ピクセル単位）。 デフォルト: 0。
border_color	string	枠線の色。 デフォルト: 000000。
background_color	string	背景色（塗りつぶし色）。 デフォルト: (empty)。
zindex	integer	すべての図形および線の順序付けに使用される値（z-index）。 デフォルト: 0。

マップの線

マップの線オブジェクトは、マップ上に表示される線を定義します。
以下のプロパティがあります。

プロパティ	型	説明
sysmap_shapeid	ID	マップ図形要素の ID。 プロパティの動作: - 読み取り専用
x1	integer	線のポイント 1 の X 座標（ピクセル単位）。 デフォルト: 0。
y1	integer	線のポイント 1 の Y 座標（ピクセル単位）。 デフォルト: 0。

プロパティ	型	説明
x2	integer	線のポイント 2 の X 座標（ピクセル単位）。 デフォルト: 200。
y2	integer	線のポイント 2 の Y 座標（ピクセル単位）。 デフォルト: 200。
line_type	integer	線の種類。 使用可能な値: 0 - なし; 1 - ————; 2 - ┅; 3 - – – –. デフォルト: 0。
line_width	integer	線の幅（ピクセル単位）。 デフォルト: 0。
line_color	string	線の色。 デフォルト: 000000。
zindex	integer	すべての図形および線の順序付けに使用される値（z-index）。 デフォルト: 0。

map.create

説明

`object map.create(object/array maps)`

このメソッドを使用すると、新しいマップを作成できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するマップ。

標準のマッププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
links	array	マップ上に作成するマップリンク。
selements	array	マップ上に作成するマップ要素。
urls	array	マップ上に作成するマップ URL。
users	array	マップ上に作成するマップユーザー共有。
userGroups	array	マップ上に作成するマップユーザーグループ共有。
shapes	array	マップ上に作成するマップ図形。
lines	array	マップ上に作成するマップ線。

Note:
マップリンクを作成するには、マップ要素の `selementid` に任意の値を設定し、その値を使用してリンクの `selementid1` または `selementid2` プロパティでこの要素を参照する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。例を参照してください。

戻り値

(`object`) `sysmapids` プロパティの下で作成されたマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたマップの順序と一致します。

例

空のマップを作成する

要素のないマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

ホストマップの作成

2 つのホスト要素とそれらの間のリンクを持つマップを作成します。マップリンクオブジェクトでマップ要素を参照するために、一時的な”selementid1” および”selementid2” の値を使用していることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "selementid": "1",
                "elements": [
                    {"hostid": "1033"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            },

            {
                "selementid": "2",
                "elements": [
                    {"hostid": "1037"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            }
        ],
        "links": [
            {
                "selementid1": "1",
                "selementid2": "2"
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

トリガーマップの作成

2 つのトリガーを含むトリガー要素でマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Trigger map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "elements": [
                    {"triggerid": "12345"},
                    {"triggerid": "67890"}
                ],
                "elementtype": 2,
                "iconid_off": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

マップの共有

2 種類の共有 (ユーザーとユーザーグループ) でマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map sharing",
```

```
        "width": 600,
        "height": 600,
        "users": [
            {
                "userid": "4",
                "permission": "3"
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

マップシェイプ

マップ名のタイトルを持つマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "shapes": [
            {
                "type": 0,
                "x": 0,
                "y": 0,
                "width": 600,
                "height": 11,
                "text": "{MAP.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
```

```
    "id": 1
}
```

マップライン

マップラインを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map API lines",
        "width": 500,
        "height": 500,
        "lines": [
            {
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

関連項目

- マップ要素
- マップリンク
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

ソース

CMap::create() in *ui/include/classes/api/services/CMap.php*.

map.delete

説明

`object map.delete(array mapIds)`

このメソッドを使用すると、マップを削除できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するマップの ID。

戻り値

(object) sysmapids プロパティの下にある削除されたマップの ID を含むオブジェクトを返します。

例

複数のマップの削除

2 つのマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

ソース

CMap::delete() in *ui/include/classes/api/services/CMap.php*.

map.get

説明

`integer/array map.get(object parameters)`

このメソッドを使用すると、指定されたパラメータに従ってマップを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
sysmapids	ID/array	指定した ID のマップのみを返します。
userids	ID/array	指定したユーザー ID に属するマップのみを返します。

パラメータ	型	説明
expandUrls	flag	対応するマップ要素にグローバルマップ URL を追加し、すべてのマップ要素 URL 内のマクロを展開します。
selectIconMap	query	マップで使用されているアイコンマップを含む`iconmap` プロパティを返します。
selectLinks	query	要素間のマップリンクを含む`links` プロパティを返します。
selectSelements	query	マップ要素を含む`selements` プロパティを返します。
selectUrls	query	マップ URL を含む`urls` プロパティを返します。
selectUsers	query	そのマップが共有されているユーザーを含む`users` プロパティを返します。
selectUserGroups	query	そのマップが共有されているユーザーグループを含む`userGroups` プロパティを返します。
selectShapes	query	マップ図形を含む`shapes` プロパティを返します。
selectLines	query	マップ線を含む`lines` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `name`, `width`, `height`。
countOutput	boolean	これらのパラメータはリファレンス解説 で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

マップの取得

マップ"3" のすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.get",
    "params": {
        "output": "extend",
        "selectSelements": "extend",
        "selectLinks": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "selectShapes": "extend",
        "selectLines": "extend",
        "sysmapids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix server",
                    "label_location": "3",
                    "x": "11",
                    "y": "141",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "-1",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ]
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
                    "x": "211",
                    "y": "191",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "0",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": []
                },
                {
                    "selementid": "12",
                    "sysmapid": "3",
                    "elementtype": "0",
```

```
                "evaltype": "0",
                "iconid_off": "185",
                "iconid_on": "0",
                "label": "{HOST.NAME}\r\n{HOST.CONN}",
                "label_location": "0",
                "x": "111",
                "y": "61",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "0",
                "show_label": "1",
                "zindex":"0",
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ],
                "urls": [],
                "tags": []
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "show_label": "1",
                "indicator_type": "0",
                "itemid": "0",
                "linktriggers": [],
                "thresholds": [],
                "highlights": []
            }
        ],
        "users": [
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
```

```
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "background_scale": "1",
        "label_type": "2",
        "label_location": "3",
        "show_element_label": "0",
        "show_link_label": "1",
        "highlight": "1",
        "expandproblem": "1",
        "markelements": "0",
        "show_unack": "0",
        "grid_size": "50",
        "grid_show": "1",
        "grid_align": "1",
        "label_format": "0",
        "label_type_host": "2",
        "label_type_hostgroup": "2",
        "label_type_trigger": "2",
        "label_type_map": "2",
        "label_type_image": "2",
        "label_string_host": "",
        "label_string_hostgroup": "",
        "label_string_trigger": "",
        "label_string_map": "",
        "label_string_image": "",
        "iconmapid": "0",
        "expand_macros": "0",
        "severity_min": "0",
        "userid": "1",
        "private": "1",
```

```
            "show_suppressed": "1"
        }
    ],
    "id": 1
}
```

関連項目

- アイコンマップ
- マップ要素
- マップリンク
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

ソース

CMap::get() in *ui/include/classes/api/services/CMap.php*.

map.update

説明

`object map.update(object/array maps)`

このメソッドを使用すると、既存のマップを更新できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するマップのプロパティ。

各マップでは mapid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のマッププロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
links	array	既存のリンクを置き換えるマップリンク。
selements	array	既存の要素を置き換えるマップ要素。
urls	array	既存の URL を置き換えるマップ URL。
users	array	既存の要素を置き換えるマップユーザー 共有。
userGroups	array	既存の要素を置き換えるマップユーザーグループ 共有。
shapes	array	既存の図形を置き換えるマップ図形。
lines	array	既存の線を置き換えるマップ線。

> **Note:**
> 新しいマップ要素間にマップリンクを作成するには、要素の selementid に任意の値を設定し、その値を使用してリンクの selementid1 または selementid2 プロパティでこの要素を参照する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。map.create の例を参照してください。

戻り値

(object) sysmapids プロパティの下で更新されたマップの ID を含むオブジェクトを返します。

例

マップのサイズ変更

マップのサイズを 1200x1200 ピクセルに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

マップの所有者を変更

管理者とスーパ管理者のみ利用可能です。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

関連項目

- マップ要素
- マップリンク
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

ソース

CMap::update() in *ui/include/classes/api/services/CMap.php*.

メディアタイプ

このクラスはメディアタイプを操作するために設計されています。

オブジェクト参照:

- メディアタイプ
  - webhook パラメータ
  - スクリプトパラメータ
- メッセージテンプレート

利用可能なメソッド:

- mediatype.create - 新しいメディアタイプを作成
- mediatype.delete - メディアタイプを削除
- mediatype.get - メディアタイプを取得
- mediatype.update - メディアタイプを更新

Media type オブジェクト

次のオブジェクトは、`mediatype`API に直接関連しています。

メディアタイプ

メディアタイプオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
mediatypeid	ID	メディアタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	メディアタイプの名前。 プロパティの動作: - 作成操作では必須
type	integer	メディアタイプで使用する転送方式。 指定可能な値: 0 - Email; 1 - Script; 2 - SMS; 4 - Webhook; 5 - Push. プロパティの動作: - 作成操作では必須
exec_path	string	AlertScriptsPath サーバー設定パラメータで指定されたディレクトリに配置されているスクリプトファイルの名前 (notification.sh など)。 プロパティの動作: - `type` が”Script” に設定されている場合は必須
gsm_modem	string	GSM モデムのシリアルデバイス名。 プロパティの動作: - `type` が”SMS” に設定されている場合は必須
passwd	string	認証パスワード。 プロパティの動作: - `smtp_authentication` が”Normal password” に設定されている場合にサポート

プロパティ	型	説明
provider	integer	メールプロバイダー。 指定可能な値: 0 - *(デフォルト)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.
smtp_email	string	通知の送信元となるメールアドレス。 プロパティの動作: - `type` が”Email” に設定されている場合は必須
smtp_helo	string	SMTP HELO。 プロパティの動作: - `type` が”Email” に設定されている場合にサポート
smtp_server	string	SMTP サーバー。 プロパティの動作: - `type` が”Email” に設定されている場合は必須
smtp_port	integer	接続先 SMTP サーバーのポート。 デフォルト: 25。 プロパティの動作: - `type` が”Email” に設定されている場合にサポート
smtp_security	integer	使用する SMTP 接続のセキュリティレベル。 指定可能な値: 0 - *(デフォルト)* None; 1 - STARTTLS; 2 - SSL/TLS. プロパティの動作: - `type` が”Email” に設定されている場合にサポート
smtp_verify_host	integer	SMTP の SSL ホスト検証。 指定可能な値: 0 - *(デフォルト)* No; 1 - Yes. プロパティの動作: - `smtp_security` が”STARTTLS” または”SSL/TLS” に設定されている場合にサポート
smtp_verify_peer	integer	SMTP の SSL ピア検証。 指定可能な値: 0 - *(デフォルト)* No; 1 - Yes. プロパティの動作: - `smtp_security` が”STARTTLS” または”SSL/TLS” に設定されている場合にサポート

プロパティ	型	説明
smtp_authentication	integer	使用する SMTP 認証方式。 指定可能な値: 0 - *(デフォルト)* None; 1 - Normal password; 2 - OAuth token. *Office365 relay* メールプロバイダーでは OAuth 認証は許可されません。 プロパティの動作: - type が”Email” に設定されている場合にサポート
redirection_url	string	OAuth 認証に戻るための Zabbix Web インターフェース URL。 デフォルト: zabbix.php?action=oauth.authorize の部分を含む API 設定プロパティ url の値 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合は必須
client_id	string	OAuth 認証サーバーに登録されたクライアント識別子。 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合は必須
client_secret	string	OAuth 認証サーバーに登録されたクライアントシークレット。Super Admin タイプのユーザーのみアクセスできます。 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合は必須
authorization_url	string	アクセストークンおよびリフレッシュトークンを取得するためのパラメータ付き OAuth URL。 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合は必須
token_url	string	認証トークンをアクセストークンおよびリフレッシュトークンと交換するための OAuth URL。 この URL は、無効なアクセストークンをサーバーが更新する際にも使用されます。 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合は必須
tokens_status	integer	トークンのステータスを示すビットマスク。 指定可能な値: 0 - (デフォルト) 両方のトークンに無効な値が含まれる 1 - アクセストークンに有効な値が含まれる 2 - リフレッシュトークンに有効な値が含まれる 3 - 両方のトークンに有効な値が含まれる。 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合にサポート
access_token	string	OAuth アクセストークンの値。 プロパティの動作: - smtp_authentication が”OAuth token” に設定されている場合は必須

プロパティ	型	説明
access_token_updated	timestamp	`refresh_token` を使用した更新時、またはトークン変更時に API によってサーバーが `access_token` を最後に変更した時刻のタイムスタンプ。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合にサポート
access_expires_in	integer	`access_token` が期限切れとなり、`refresh_url` へのリクエストが必要になるまでの秒数。 アクセストークンの更新時に Zabbix サーバーによって設定されるか、トークン変更時に API によって設定されます。 タイムスタンプは `access_token_updated` の値にこの値を加算して計算されます。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合にサポート
refresh_token	string	OAuth リフレッシュトークンの値。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は必須
status	integer	メディアタイプが有効かどうか。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
username	string	ユーザー名。 プロパティの動作: - `smtp_authentication` が”Normal password” に設定されている場合にサポート
maxsessions	integer	並列処理できるアラートの最大数。 `type` が”SMS” に設定されている場合の指定可能な値: 1。 `type` が”Email”、”Script”、または”Webhook” に設定されている場合の指定可能な値: 0-100。 デフォルト: 1。
maxattempts	integer	アラート送信を試行する最大回数。 指定可能な値: 1-100。 デフォルト: 3。
attempt_interval	string	再試行間隔。 秒数 (例: 1、30、60) または時間のサフィックス付きの値 (例: 30s、1m、1h) を指定できます。 指定可能な値: 0-1h。 デフォルト: 10s。
message_format	integer	メッセージ形式。 指定可能な値: 0 - プレーンテキスト; 1 - *(デフォルト)* HTML。 プロパティの動作: - `type` が”Email” に設定されている場合にサポート

プロパティ	型	説明
script	text	webhook スクリプト本文 (JavaScript)。
timeout	string	プロパティの動作: - `type` が”Webhook” に設定されている場合は必須 webhook スクリプトのタイムアウト。 秒数 (例: 1、60) または時間のサフィックス付きの値 (例: 30s、1m) を指定できます。 指定可能な値: 1-60s。 デフォルト: 30s。
process_tags	integer	プロパティの動作: - `type` が”Webhook” に設定されている場合にサポート webhook スクリプトのレスポンスに含まれる JSON プロパティの値をタグとして処理します。これらのタグは既存の障害タグに追加されます。 指定可能な値: 0 - *(デフォルト)* webhook スクリプトのレスポンスを無視; 1 - webhook スクリプトのレスポンスをタグとして処理。
show_event_menu	integer	プロパティの動作: - `type` が”Webhook” に設定されている場合にサポート カスタム URL へのリンクを含む項目をevent menuに追加します。また、problem.getおよびevent.getの出力に `urls` プロパティを追加します。 指定可能な値: 0 - *(デフォルト)* event menu 項目または `urls` プロパティを含めない; 1 - event menu 項目および `urls` プロパティを含める。
event_menu_url	string	プロパティの動作: - `type` が”Webhook” に設定されている場合にサポート event menu項目およびproblem.getとevent.getが返す `urls` プロパティで使用する URL。
event_menu_name	string	プロパティの動作: - `show_event_menu` が”Include event menu entry and `urls` property” に設定されている場合にサポート event menu項目およびproblem.getとevent.getが返す `urls` プロパティで使用する名前。
parameters	array	プロパティの動作: - `show_event_menu` が”Include event menu entry and `urls` property” に設定されている場合にサポート Webhookまたはスクリプトのパラメータ。
description	text	プロパティの動作: - `type` が”Webhook” または”Script” に設定されている場合にサポート メディアタイプの説明。

webhook パラメータ

webhook パラメータには以下のプロパティがあります。

プロパティ	Type	説明
name	string	パラメータ名。 プロパティの動作: - 必須

プロパティ	Type	説明
value	string	パラメータ値。マクロをサポートします。 サポートされるマクロについては、サポートされるマクロページを参照してください。

スクリプトパラメータ

スクリプトパラメータには以下のプロパティがあります。

プロパティ	型	説明
sortorder	integer	パラメータ値がコマンドライン引数としてスクリプトに渡される順序です。最初の引数は 0 から始まります。 プロパティの動作: - 必須
value	string	パラメータ値です。マクロをサポートします。 サポートされるマクロについては、サポートされるマクロページを参照してください。

メッセージテンプレート

メッセージテンプレートオブジェクトは、通知を送信するためのアクションの実行時にデフォルトメッセージとして使用されるテンプレートを定義します。以下のプロパティがあります。

プロパティ	Type	説明
eventsource	integer	イベントソース。 使用可能な値: 0 - トリガー; 1 - ディスカバリ; 2 - 自動登録; 3 - 内部; 4 - サービス。 プロパティの動作: - 必須
recovery	integer	実行モード。 使用可能な値: 0 - 実行; 1 - 復旧時の実行; 2 - 更新時の実行。 プロパティの動作: - 必須
subject	string	メッセージの件名。
message	string	メッセージ本文。

mediatype.create

説明

`object mediatype.create(object/array mediaTypes)`

このメソッドを使用すると、新しいメディアタイプを作成できます。

Note:
このメソッドは、特権管理者ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェースの設定ファイルの `$ZBX_FEATURE_FLAGS['media_type_denylist']` 機能フラグで配信方法が指定されている場合、ユーザーは該当するメディアタイプに対してこのメソッドを呼び出すことができません。

パラメータ

(object/array) 作成するメディアタイプ。

標準のメディアタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
message_templates	array	メディアタイプ用に作成するメッセージテンプレート。

戻り値

(object) mediatypeids プロパティの下に作成されたメディアタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメディアタイプの順序と一致します。

例

メールメディアタイプの作成

カスタム SMTP ポートとメッセージテンプレートを使用して新しいメールメディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "smtp_port": "587",
        "message_format": "1",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problem: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" started at {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Resolved in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" has been resolved at {EVENT.R
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem \"{EVENT.NAME}\" on host \"{HOST
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
```

```
    "id": 1
}
```

スクリプトメディアタイプの作成

試行回数と試行間隔にカスタム値を設定した新しいスクリプトメディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push notifications",
        "exec_path": "push-notification.sh",
        "maxattempts": "5",
        "attempt_interval": "11s",
        "parameters": [
            {
                "sortorder": "0",
                "value": "{ALERT.SENDTO}"
            },
            {
                "sortorder": "1",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "sortorder": "2",
                "value": "{ALERT.MESSAGE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Webhook メディアタイプの作成

新しい Webhook メディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "4",
        "name": "Webhook",
        "script": "var Webhook = {\r\n    token: null,\r\n    to: null,\r\n    subject: null,\r\n    messa
        "parameters": [
            {
                "name": "Message",
                "value": "{ALERT.MESSAGE}"
            },
            {
```

```
                "name": "Subject",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "name": "To",
                "value": "{ALERT.SENDTO}"
            },
            {
                "name": "Token",
                "value": "<Token>"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::create() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

説明

`object mediatype.delete(array mediaTypeIds)`

このメソッドを使用すると、メディアタイプを削除できます。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェースの設定ファイルの `$ZBX_FEATURE_FLAGS['media_type_denylist']` 機能フラグに配信方法が指定されている場合、ユーザーは該当するメディアタイプに対してこのメソッドを呼び出すことができません。

パラメーター

(`array`) 削除するメディアタイプの ID。

戻り値

(`object`) `mediatypeids` プロパティの下で削除されたメディアタイプの ID を含むオブジェクトを返します。

例

複数のメディアタイプの削除

2 つのメディアタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::delete() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

説明

`integer/array mediatype.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってメディアタイプを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェース設定ファイルの `$ZBX_FEATURE_FLAGS['media_type_denylist']` 機能フラグに配信方法が指定されている場合、ユーザーは該当するメディアタイプに対してこのメソッドを呼び出すことができません。

パラメータ

(`object`) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Note:
メディアタイプのユーザー関連情報を要求する場合、*Admin* および *User* タイプのユーザーは、自分自身のユーザーに関するデータのみ取得できます。例については、Retrieving media types as *Admin* を参照してください。

パラメータ	型	説明
mediatypeids	ID/array	指定した ID を持つメディアタイプのみを返します。
mediaids	ID/array	指定したmediaで使用されているメディアタイプのみを返します。
userids	ID/array	指定したユーザーで使用されているメディアタイプのみを返します。
selectActions	query	メディアタイプを使用しているアクションを含む`actions` プロパティを返します。
selectMessageTemplates	query	メディアタイプメッセージの配列を含む`message_templates` プロパティを返します。 パラメータの動作: - *Super admin* タイプのユーザーで *supported*
selectUsers	query	メディアタイプを使用しているユーザーを含む`users` プロパティを返します。 ユーザータイプに基づく制限については、`user.get` を参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `mediatypeid`。

パラメータ	型	説明
filter	object	指定したフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしていません。
output	query	*Admin* および *User* タイプのユーザーで指定可能なMedia type object プロパティ: `mediatypeid`, `name`, `type`, `status`, `maxattempts`。 返されるMedia type object のプロパティ。 *Admin* および *User* タイプのユーザーは、次のプロパティのみ取得できます: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`。例については、Retrieving media types as *Admin* を参照してください。
search	object	デフォルト: `extend`。 指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索を実行します。 `string` および `text` データ型 のプロパティのみをサポートします。 *Admin* および *User* タイプのユーザーで指定可能なMedia type object プロパティ: `name`, `description`。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

メディアタイプの取得

すべての設定済みメディアタイプを取得します。次の例では、2 つのメディアタイプが返されます:

- メールのメディアタイプ
- SMS のメディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "smtp_server": "mail.example.com",
            "smtp_helo": "example.com",
            "smtp_email": "zabbix@example.com",
            "exec_path": "",
            "gsm_modem": "",
            "username": "",
            "passwd": "",
            "status": "0",
            "smtp_port": "25",
            "smtp_security": "0",
            "smtp_verify_peer": "0",
            "smtp_verify_host": "0",
            "smtp_authentication": "0",
            "maxsessions": "1",
            "maxattempts": "3",
            "attempt_interval": "10s",
            "message_format": "0",
            "script": "",
            "timeout": "30s",
            "process_tags": "0",
            "show_event_menu": "1",
            "event_menu_url": "",
            "event_menu_name": "",
            "description": "",
            "provider": "0",
            "message_templates": [
                {
                    "eventsource": "0",
                    "recovery": "0",
                    "subject": "Problem: {EVENT.NAME}",
                    "message": "Problem started at {EVENT.TIME} on {EVENT.DATE}\r\nProblem name: {EVENT.NA
                },
                {
                    "eventsource": "0",
                    "recovery": "1",
                    "subject": "Resolved: {EVENT.NAME}",
                    "message": "Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE
                },
                {
                    "eventsource": "0",
                    "recovery": "2",
                    "subject": "Updated problem: {EVENT.NAME}",
                    "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
                },
                {
                    "eventsource": "1",
                    "recovery": "0",
                    "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                    "message": "Discovery rule: {DISCOVERY.RULE.NAME}\r\n\r\nDevice IP: {DISCOVERY.DEVICE.
                },
                {
                    "eventsource": "2",
                    "recovery": "0",
                    "subject": "Autoregistration: {HOST.HOST}",
```

```
                "message": "Host name: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.PORT}"
            }
        ],
        "parameters": []
    },
    {
        "mediatypeid": "3",
        "type": "2",
        "name": "SMS",
        "smtp_server": "",
        "smtp_helo": "",
        "smtp_email": "",
        "exec_path": "",
        "gsm_modem": "/dev/ttyS0",
        "username": "",
        "passwd": "",
        "status": "0",
        "smtp_port": "25",
        "smtp_security": "0",
        "smtp_verify_peer": "0",
        "smtp_verify_host": "0",
        "smtp_authentication": "0",
        "maxsessions": "1",
        "maxattempts": "3",
        "attempt_interval": "10s",
        "message_format": "1",
        "script": "",
        "timeout": "30s",
        "process_tags": "0",
        "show_event_menu": "1",
        "event_menu_url": "",
        "event_menu_name": "",
        "description": "",
        "provider": "0",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "",
                "message": "{EVENT.SEVERITY}: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "",
                "message": "RESOLVED: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "",
                "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
            },
            {
                "eventsource": "2",
                "recovery": "0",
```

```
                    "subject": "",
                    "message": "Autoregistration: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.P
                }
            ],
            "parameters": []
        }
    ],
    "id": 1
}
```

Admin としてメディアタイプを取得

Admin タイプのユーザーとして、有効になっているすべてのメディアタイプと、それらのメディアタイプを使用するユーザーを取得します。次の例では、2 つのメディアタイプが返されます。

- ユーザーが 1 人いるメールメディアタイプ（*Admin* タイプのユーザー自身のユーザーのみ）
- ユーザーがいない SMS メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": [
                {
                    "userid": "3",
                    "username": "database-admin",
                    "name": "John",
                    "surname": "Doe",
                    "url": "",
                    "autologin": "0",
                    "autologout": "0",
                    "lang": "default",
                    "refresh": "30s",
                    "theme": "default",
                    "attempt_failed": "0",
                    "attempt_ip": "",
                    "attempt_clock": "0",
                    "rows_per_page": "50",
                    "timezone": "default",
                    "roleid": "2",
                    "provisioned": "0"
                }
            ]
```

```
        },
        {
            "mediatypeid": "3",
            "type": "2",
            "name": "SMS",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": []
        }
    ],
    "id": 1
}
```

スクリプトと Webhook のメディアタイプ取得

次の例では、3 つのメディアタイプが返されます。

- パラメータ付きのスクリプトメディアタイプ
- パラメータなしのスクリプトメディアタイプ
- パラメータ付きの Webhook メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
```

```
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```

参照

- User

ソース

CMediaType::get() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

説明

`object mediatype.update(object/array mediaTypes)`

このメソッドを使用すると、既存のメディアタイプを更新できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。
 Web インターフェース設定ファイルの $ZBX_FEATURE_FLAGS['media_type_denylist'] 機能フラグに配信方法が指定されている場合、ユーザーは該当するメディアタイプに対してこのメソッドを呼び出すことができません。

パラメータ

(object/array) 更新するメディアタイプのプロパティ。

各メディアタイプに対して mediatypeid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

さらに、標準のメディアタイププロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
message_templates	array	現在のメッセージテンプレートを置き換えるメッセージテンプレート。

戻り値

(object) mediatypeids プロパティの下で更新されたメディアタイプの ID を含むオブジェクトを返します。

例

メディアタイプの有効化

メディアタイプを有効にします。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::update() in *ui/include/classes/api/services/CMediaType.php*.

メンテナンス

このクラスはメンテナンスを扱うために設計されています。

オブジェクト参照:

- メンテナンス
- 期間
- 障害タグ

利用可能なメソッド:

- maintenance.create - 新しいメンテナンスを作成
- maintenance.delete - メンテナンスを削除
- maintenance.get - メンテナンスを取得
- maintenance.update - メンテナンスを更新

Maintenance オブジェクト

次のオブジェクトは、maintenanceAPI に直接関連しています。

メンテナンス

メンテナンスオブジェクトには次のプロパティがあります。

Property	Type	Description
maintenanceid	ID	メンテナンスの ID。 Property behavior: - *read-only* - 更新操作では 必須
name	string	メンテナンスの名前。 Property behavior: - 作成操作では 必須
active_since	timestamp	メンテナンスが有効になる時刻（含む）。 指定された値は分単位に切り捨てられます。 Property behavior: - 作成操作では 必須
active_till	timestamp	メンテナンスが無効になる時刻（含まない）。 指定された値は分単位に切り捨てられます。 Property behavior: - 作成操作では 必須
description	string	メンテナンスの説明。
maintenance_type	integer	メンテナンスの種類。 指定可能な値: 0 - *(default)* データ収集あり; 1 - データ収集なし。
tags_evaltype	integer	障害タグの評価方法。 指定可能な値: 0 - *(default)* And/Or; 2 - Or.

時間帯

時間帯オブジェクトは、メンテナンスを有効にする期間を定義するために使用されます。このオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
period	integer	メンテナンス期間の長さ（秒単位）。 指定された値は分単位に切り捨てられます。 指定可能な値の範囲: 300 - 86399940。 デフォルト: 3600。
timeperiod_type	integer	時間帯のタイプ。 指定可能な値: 0 - *(デフォルト)* 一度のみ; 2 - 毎日; 3 - 毎週; 4 - 毎月。
start_date	timestamp	メンテナンス期間を有効にする日時。 指定された値は分単位に切り捨てられます。 デフォルト: 現在日時。 プロパティの動作: - `timeperiod_type` が「一度のみ」に設定されている場合に サポート

プロパティ	型	説明
start_time	integer	メンテナンス開始時刻の時刻部分（秒単位）。 指定された値は分単位に切り捨てられます。 デフォルト: 0。 プロパティの動作: - timeperiod_type が「毎日」、「毎週」、または「毎月」に設定されている場合に サポート
every	integer	毎日および毎週の期間では、every はメンテナンスを有効にする日または週の間隔を定義します。 timeperiod_type が「毎日」または「毎週」に設定されている場合のデフォルト値: 1。 毎月の期間で day が設定されている場合、every プロパティはメンテナンスを有効にする月の日を定義します。 timeperiod_type が「毎月」に設定され、day が設定されている場合のデフォルト値: 1。 毎月の期間で dayofweek が設定されている場合、every プロパティはメンテナンスを有効にする月の週を定義します。 timeperiod_type が「毎月」に設定され、dayofweek が設定されている場合の指定可能な値: 1 - *(デフォルト)* 第 1 週; 2 - 第 2 週; 3 - 第 3 週; 4 - 第 4 週; 5 - 最終週。 プロパティの動作: - timeperiod_type が「毎日」、「毎週」、または「毎月」に設定されている場合に サポート
dayofweek	integer	メンテナンスを有効にする曜日。 指定可能なビットマップ値: 1 - 月曜日; 2 - 火曜日; 4 - 水曜日; 8 - 木曜日; 16 - 金曜日; 32 - 土曜日; 64 - 日曜日。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、月曜日・水曜日・金曜日の場合は 21）。 プロパティの動作: - timeperiod_type が「毎週」に設定されている場合、または timeperiod_type が「毎月」に設定されていて day が設定されていない場合に 必須
day	integer	メンテナンスを有効にする月の日。 プロパティの動作: - timeperiod_type が「毎月」に設定されていて dayofweek が設定されていない場合に 必須

プロパティ	型	説明
month	integer	メンテナンスを有効にする月。 指定可能なビットマップ値: 1 - 1 月; 2 - 2 月; 4 - 3 月; 8 - 4 月; 16 - 5 月; 32 - 6 月; 64 - 7 月; 128 - 8 月; 256 - 9 月; 512 - 10 月; 1024 - 11 月; 2048 - 12 月。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、1 月・4 月・7 月・10 月の場合は 585）。 プロパティの動作: - `timeperiod_type` が「毎月」に設定されている場合に 必須

イベント名

イベント名オブジェクトは、メンテナンスの開始時に抑制する障害を定義するために使用します。イベント名は、Maintenance オブジェクトの `maintenance_type` が「データ収集あり」に設定されている場合にのみ指定できます。次のプロパティがあります。

Property	Type	Description
operator	integer	条件の演算子。 指定可能な値: 2 - *(デフォルト)* 含む; 3 - 含まない。
value	string	イベント名の値。 プロパティの動作: - 必須

障害タグ

障害タグオブジェクトは、メンテナンスの開始時に抑制する必要がある障害を定義するために使用します。タグは、Maintenance object の `maintenance_type` が「with data collection」に設定されている場合にのみ指定できます。以下のプロパティがあります。

Property	Type	Description
tag	string	障害タグ名。 Property behavior: *- required*
operator	integer	条件のoperator。 指定可能な値: 0 - 等しい; 1 - 等しくない; 2 - *(default)* 含む; 3 - 含まない。
value	string	障害タグの値。

maintenance.create

説明

`object maintenance.create(object/array maintenances)`

このメソッドを使用すると、新しいメンテナンスを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 作成するメンテナンス。

標準のメンテナンスプロパティに加えて、このメソッドでは以下のパラメータを受け入れます。

パラメータ	型	説明
groups	object/array	メンテナンスの対象にするホストグループ。 ホストグループには、`groupid` プロパティのみを定義する必要があります。 パラメータの動作: - `hosts` が設定されていない場合は必須
hosts	object/array	メンテナンスの対象にするホスト。 ホストには、`hostid` プロパティのみを定義する必要があります。 パラメータの動作: - `groups` が設定されていない場合は必須
triggers	object/array	メンテナンスの対象にするトリガー。 トリガーには、`triggerid` プロパティのみを定義する必要があります。 パラメータの動作: - `groups` と `hosts` が設定されていない場合は必須 - maintenance_type=0（データ収集あり）の場合にサポート
event_names	object/array	メンテナンスの対象にする障害を定義するイベント名。 パラメータの動作: - maintenance_type=0（データ収集あり）の場合にサポート
tags	object/array	メンテナンスの対象にする障害を定義する障害タグ。 パラメータの動作: - メンテナンスオブジェクトの `maintenance_type` が「データ収集あり」に設定されている場合にサポート
timeperiods	object/array	メンテナンスの期間。 パラメータの動作: - 必須

戻り値

(`object`) `maintenanceids` プロパティの下で作成されたメンテナンスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメンテナンスの順序と一致します。

例

メンテナンスの作成

ID「20」のホストグループに対してデータ収集を行い、障害タグ **service:mysqld** および **error** を指定したメンテナンスを作成します。
2026 年 3 月 17 日から 2027 年 3 月 17 日まで有効で、毎週日曜日の 18:00 に開始し、1 時間継続する必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

アドホックメンテナンスの作成

ID が「4」のホストグループ、ID が「12345」のトリガー、およびイベント名 **Printer: ink low** を対象に、データ収集を行う 1 時間のアドホックメンテナンスを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Ad-hoc maintenance",
        "active_since": 1792057740,
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
```

```
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "4"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト
- ホストグループ
- トリガー
- 障害名
- 障害タグ
- 期間

ソース

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

説明

`object maintenance.delete(array maintenanceIds)`

このメソッドを使用すると、メンテナンス期間を削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するメンテナンス期間の ID。

戻り値

(object) maintenanceids プロパティの下にある削除されたメンテナンス期間の ID を含むオブジェクトを返します。

例

複数のメンテナンス期間の削除

2 つのメンテナンス期間を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

ソース

CMaintenance::delete() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

説明

`integer/array maintenance.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってメンテナンスを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。このメソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは、以下のパラメータをサポートしています。

パラメータ	型	説明
groupids	ID/array	指定したホストグループに割り当てられているメンテナンスのみを返します。
hostids	ID/array	指定したホストに割り当てられているメンテナンスのみを返します。
triggerids	ID/array	指定したトリガーに割り当てられているメンテナンスのみを返します。
maintenanceids	ID/array	指定した ID を持つメンテナンスのみを返します。
selectHostGroups	query	メンテナンスに割り当てられているホストグループを含むhostgroupsプロパティを返します。
selectHosts	query	メンテナンスに割り当てられているホストを含むhostsプロパティを返します。
selectTriggers	query	メンテナンスに割り当てられているトリガーを含むtriggersプロパティを返します。
selectEventNames	query	メンテナンスに割り当てられているイベント名を含むevent_namesプロパティを返します。
selectTags	query	メンテナンスの障害タグを含むtagsプロパティを返します。
selectTimeperiods	query	メンテナンスの期間を含むtimeperiodsプロパティを返します。

パラメータ	型	説明
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `maintenanceid`、`name`、`maintenance_type`、`active_since`、`active_till`。
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明しています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合は、取得したオブジェクトの数

例

メンテナンスの取得

設定されているすべてのメンテナンス、割り当てられたメンテナンス範囲（ホストグループ、ホスト、トリガー、イベント名、タグ）に関するデータ、および定義された期間を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
      "name": "Regular DB maintenance",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1791936000",
      "active_till": "2107555200",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "20",
          "name": "Databases",
```

```
          "flags": "0",
          "uuid": "748ad4d098d447d492bb935c907f652f"
        }
      ],
      "triggers": [],
      "event_names": [],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "4",
          "every": "1",
          "month": "4095",
          "dayofweek": "0",
          "day": "1",
          "start_time": "64800",
          "period": "3600",
          "start_date": "0"
        }
      ]
    },
    {
      "maintenanceid": "2",
      "name": "Ad-hoc: Printer: ink OUT",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1792057740",
      "active_till": "1792061340",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "4",
          "name": "Zabbix servers",
          "flags": "0",
          "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
      ],
      "triggers": [],
      "event_names": [
        {
          "operator": "2",
          "value": "Printer: ink OUT"
        }
      ],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "0",
          "every": "1",
          "month": "0",
          "dayofweek": "0",
          "day": "0",
          "start_time": "0",
          "period": "3600",
          "start_date": "1792057740"
        }
      ]
    }
  ],
  "id": 1
}
```

関連項目

- ホスト
- ホストグループ
- トリガー
- 障害名
- 障害タグ
- 期間

ソース

CMaintenance::get() in *ui/include/classes/api/services/CMaintenance.php*。

maintenance.update

説明

`object maintenance.update(object/array maintenances)`

このメソッドを使用すると、既存のメンテナンスを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するメンテナンスのプロパティ。

各メンテナンスに対して `maintenanceid` プロパティを定義する必要があります。それ以外のプロパティは任意です。渡されたプロパティのみが更新され、それ以外のプロパティは変更されません。

標準のメンテナンスプロパティに加えて、このメソッドでは以下のパラメータを受け付けます。

Parameter	Type	Description
groups	object/array	現在のホストグループを置き換えるホストグループ。 ホストグループには `groupid` プロパティのみを定義する必要があります。 パラメータの動作: - `hosts` が設定されていない場合は必須
hosts	object/array	現在のホストを置き換えるホスト。 ホストには `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - `groups` が設定されていない場合は必須
triggers	object/array	現在のトリガーを置き換えるトリガー。 トリガーには `triggerid` プロパティのみを定義する必要があります。 パラメータの動作: - `groups` と `hosts` が設定されていない場合は必須 - maintenance_type=0（データ収集あり）の場合にサポート
event_names	object/array	現在のイベント名を置き換えるイベント名。 パラメータの動作: - maintenance_type=0（データ収集あり）の場合にサポート
tags	object/array	現在のタグを置き換える障害タグ。 パラメータの動作: - Maintenance オブジェクトの `maintenance_type` が「データ収集あり」に設定されている場合にサポート
timeperiods	object/array	現在の期間を置き換えるメンテナンスのタイム期間。

戻り値

(object) maintenanceids プロパティの下で更新されたメンテナンスの ID を含むオブジェクトを返します。

例

異なるホストの割り当て

現在メンテナンスに割り当てられているホストを 2 つの異なるホストに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

関連項目

- 期間

ソース

CMaintenance::update() in *ui/include/classes/api/services/CMaintenance.php*.

ユーザーグループ

このクラスはユーザーグループを操作するために設計されています。

オブジェクト参照:

- ユーザーグループ
- 権限
- タグベースの権限

利用可能なメソッド:

- usergroup.create - 新しいユーザーグループを作成
- usergroup.delete - ユーザーグループを削除
- usergroup.get - ユーザーグループを取得
- usergroup.update - ユーザーグループを更新

User group オブジェクト

次のオブジェクトは、usergroupAPI に直接関連しています。

ユーザーグループ

ユーザーグループオブジェクトには、次のプロパティがあります。

Property	Type	Description
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ユーザーグループの名前。 プロパティの動作: - 作成操作では必須
debug_mode	integer	デバッグモードが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
gui_access	integer	グループ内のユーザーの Web インターフェース認証方式。 指定可能な値: 0 - *(デフォルト)* システムのデフォルト認証方式を使用; 1 - 内部認証を使用; 2 - LDAP 認証を使用; 3 - Web インターフェースへのアクセスを無効化。
mfa_status	integer	グループ内のユーザーに対して MFA が有効か無効か。 指定可能な値: 0 - 無効（設定されているすべての MFA 方式に対して）; 1 - 有効（設定されているすべての MFA 方式に対して）。
mfaid	ID	グループ内のユーザーに使用されるMFA 方式。 プロパティの動作: - *Super admin* タイプのユーザーでサポート - Authentication オブジェクトの `mfa_status` が「Enabled」に設定されている場合にサポート
proxy_mode	integer	プロキシグループに含まれていないプロキシへのアクセスを処理するモード。 指定可能な値: 0 - 拒否リスト; 1 - *(デフォルト)* 許可リスト。
proxy_group_mode	integer	プロキシグループ（およびそのプロキシ）へのアクセスを処理するモード。 プロパティの動作: - *Super admin* タイプのユーザーでサポート。 指定可能な値: 0 - 拒否リスト; 1 - *(デフォルト)* 許可リスト。
users_status	integer	ユーザーグループが有効か無効か。 プロビジョニング解除済みのユーザーの場合、ユーザーグループを有効にすることはできません。 プロパティの動作: - *Super admin* タイプのユーザーでサポート。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。

Property	Type	Description
userdirectoryid	ID	認証に使用されるユーザーディレクトリの ID。 プロパティの動作: - *Super admin* タイプのユーザーでサポート - `gui_access` が「システムのデフォルト認証方式を使用」または「LDAP 認証を使用」に設定されている場合にサポート

権限

permission オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
id	ID	権限を追加するホストグループまたはテンプレートグループの ID。 プロパティの動作: - 必須
permission	integer	ホストグループまたはテンプレートグループに対するアクセスレベル。 使用可能な値: 0 - アクセス拒否; 2 - 読み取り専用アクセス; 3 - 読み書きアクセス。 プロパティの動作: - create 操作で 必須

タグベースの権限

タグベースの権限オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
groupid	ID	権限を追加するホストグループの ID。 Property behavior: - *required*
tag	string	タグ名。
value	string	タグの値。

usergroup.create

説明

`object usergroup.create(object/array userGroups)`

このメソッドを使用すると、新しいユーザーグループを作成できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するユーザーグループ。

標準のユーザーグループプロパティに加えて、このメソッドでは以下のパラメータを受け付けます。

Parameter	Type	Description
hostgroup_rights	object/array	ユーザーグループに割り当てるホストグループの権限。
proxies	object/array	ユーザーグループにリンクするプロキシ。 プロキシには `proxyid` プロパティのみを定義する必要があります。

Parameter	Type	Description
proxy_groups	object/array	ユーザーグループにリンクするプロキシグループ。 プロキシグループには proxy_groupid プロパティのみを定義する必要があります。
templategroup_rights	object/array	ユーザーグループに割り当てるテンプレートグループの権限。
tag_filters	array	ユーザーグループに割り当てるタグベースの権限。
users	object/array	ユーザーグループに追加するユーザー。 ユーザーには userid プロパティのみを定義する必要があります。

戻り値

(object) 作成されたユーザーグループの ID を usrgrpids プロパティに含むオブジェクトを返します。返される ID の順序は、渡されたユーザーグループの順序と一致します。

例

ユーザーグループの作成

ホストグループ"2" へのアクセスが拒否されたユーザーグループ *Operation managers* を作成し、ユーザーを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

参照

- Permission

ソース

CUserGroup::create() in *ui/include/classes/api/services/CUserGroup.php*。

usergroup.delete

説明

`object usergroup.delete(array userGroupIds)`

このメソッドを使用すると、ユーザーグループを削除できます。

Attention:
Deprovisioned ユーザーグループ (Authentication の disabled_usrgrpid に指定されたユーザーグループ) は削除できません。

Note:
このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー役割の設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するユーザーグループの ID。

戻り値

(object) usrgrpids プロパティの下にある削除されたユーザーグループの ID を含むオブジェクトを返します。

例

複数のユーザーグループの削除

2 つのユーザーグループを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

ソース

CUserGroup::delete() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

説明

`integer/array usergroup.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってユーザーグループを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは、以下のパラメータをサポートしています。

パラメータ	型	説明
mfaids	ID/array	指定した MFA 方式を持つユーザーグループのみを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
mfa_status	integer	指定した MFA ステータスを持つユーザーグループのみを返します。 サポートされているステータスの一覧については、ユーザーグループのページを参照してください。
proxyids	ID/array	指定したプロキシを含むユーザーグループのみを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
proxy_groupids	ID/array	指定したプロキシグループを含むユーザーグループのみを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
selectProxies	ID/array	ユーザーグループのプロキシを含む `proxies` プロパティを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
selectProxyGroups	ID/array	ユーザーグループのプロキシグループを含む `proxy_groups` プロパティを返します。 パラメータの動作: - *Super admin* タイプのユーザーでサポート
status	integer	指定したステータスを持つユーザーグループのみを返します。 サポートされているステータスの一覧については、ユーザーグループのページを参照してください。
userids	ID/array	指定したユーザーを含むユーザーグループのみを返します。
usrgrpids	ID/array	指定した ID を持つユーザーグループのみを返します。
selectTagFilters	query	ユーザーグループのタグベースの権限を`tag_filters` プロパティで返します。
selectUsers	query	ユーザーグループのユーザーを`users` プロパティで返します。 ユーザータイプに基づく制限については、`user.get`を参照してください。
selectHostGroupRights	query	ユーザーグループのホストグループ権限を `hostgroup_rights` プロパティで返します。 ホストグループへのアクセスレベルの一覧については、ユーザーグループのページを参照してください。
selectTemplateGroupRights	query	ユーザーグループのテンプレートグループ権限を `templategroup_rights` プロパティで返します。 テンプレートグループへのアクセスレベルの一覧については、ユーザーグループのページを参照してください。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。
output	query	返されるユーザーグループオブジェクトのプロパティ。 *Admin* および *User* タイプのユーザーは、以下のプロパティのみ取得できます: `usrgrpid`、`name`、`gui_access`、`users_status`、`debug_mode`、`mfa_status`。 デフォルト: `extend`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `usrgrpid`、`name`。
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明しています。
editable	boolean	
excludeSearch	boolean	

パラメータ	型	説明
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

有効なユーザーグループの取得

すべての有効なユーザーグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix administrators",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Guests",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Enabled debug mode",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
```

```
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "No access to the frontend",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Read only",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Deny",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        }
    ],
    "id": 1
}
```


参照

- User

ソース

CUserGroup::get() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

説明

`object usergroup.update(object/array userGroups)`

このメソッドを使用すると、既存のユーザーグループを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新するユーザーグループのプロパティ。

各ユーザーグループに対して `usrgrpid` プロパティを定義する必要があります。それ以外のすべてのプロパティはオプションです。渡されたプロパティのみが更新され、それ以外のプロパティは変更されません。

標準のユーザーグループプロパティに加えて、このメソッドでは以下のパラメータを受け付けます。

Parameter	Type	Description
hostgroup_rights	object/array	ユーザーグループに割り当てられている現在の権限を置き換えるホストグループの権限。
proxies	object/array	ユーザーグループにリンクするプロキシ。 プロキシには proxyid プロパティのみを定義する必要があります。
proxy_groups	object/array	ユーザーグループにリンクするプロキシグループ。 プロキシグループには proxy_groupid プロパティのみを定義する必要があります。
templategroup_rights	object/array	ユーザーグループに割り当てられている現在の権限を置き換えるテンプレートグループの権限。
tag_filters	array	ユーザーグループに割り当てられている現在の権限を置き換えるタグベースの権限。
users	object/array	ユーザーグループに割り当てられている現在のユーザーを置き換えるユーザー。 ユーザーには userid プロパティのみを定義する必要があります。

戻り値

(object) usrgrpids プロパティの下で更新されたユーザーグループの ID を含むオブジェクトを返します。

例

ユーザーグループの有効化と権限の更新

ユーザーグループを有効にし、ホストグループ"2" および"4" への読み書きアクセス権を付与します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

参照

- Permission

ソース

CUserGroup::update() in *ui/include/classes/api/services/CUserGroup.php*.

ユーザーディレクトリ

このクラスは、ユーザーディレクトリを操作するために設計されています。

オブジェクト参照:

- ユーザーディレクトリ
  - メディアタイプのマッピング
  - プロビジョニンググループのマッピング

利用可能なメソッド:

- userdirectory.create - 新しいユーザーディレクトリを作成
- userdirectory.delete - ユーザーディレクトリを削除
- userdirectory.get - ユーザーディレクトリを取得
- userdirectory.update - ユーザーディレクトリを更新
- userdirectory.test - ユーザーディレクトリ接続をテスト

User directory オブジェクト

次のオブジェクトは、`userdirectoryAPI` に直接関連しています。

ユーザーディレクトリ

ユーザーディレクトリオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
userdirectoryid	ID	ユーザーディレクトリの ID。 ユーザーディレクトリが削除されると、削除されたユーザーディレクトリにリンクされているすべてのユーザーについて、User オブジェクトの `userdirectoryid` プロパティの値が「0」に設定されます。 プロパティの動作: - 読み取り専用 - 更新操作では必須
idp_type	integer	ユーザーディレクトリの ID プロバイダーで使用される認証プロトコルのタイプ。 タイプ SAML のユーザーディレクトリは 1 つだけ存在できることに注意してください。 指定可能な値: 1 - LDAP タイプのユーザーディレクトリ; 2 - SAML タイプのユーザーディレクトリ。 プロパティの動作: - 作成操作では必須
group_name	string	LDAP/SAML ユーザーディレクトリと Zabbix 間でグループをマッピングするために使用する、グループ名を含む LDAP/SAML ユーザーディレクトリ属性。 例: *cn* プロパティの動作: - `provision_status` が「有効」に設定され、Authentication オブジェクトの `saml_jit_status` が「設定済みの SAML IdP で有効」に設定されている場合は必須

プロパティ	タイプ	説明
user_username	string	プロビジョニング時にUser オブジェクトの `name` プロパティの値として使用されるユーザー名を含む LDAP/SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性も含む）。 例: *cn*、*commonName*、*displayName*、*name*
user_lastname	string	プロビジョニング時にUser オブジェクトの `surname` プロパティの値として使用されるユーザーの姓を含む LDAP/SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性も含む）。 例: *sn*、*surname*、*lastName*
provision_status	integer	ユーザーディレクトリのプロビジョニングステータス。 指定可能な値: 0 - *(デフォルト)* 無効（このユーザーディレクトリによって作成されたユーザーのプロビジョニングが無効）； 1 - 有効（このユーザーディレクトリによって作成されたユーザーのプロビジョニングが有効。さらに、LDAP または SAML プロビジョニング（Authentication オブジェクトの `ldap_jit_status` または `saml_jit_status`）も有効にする必要があります）。
provision_groups	array	LDAP/SAML ユーザーグループのパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングするためのプロビジョニンググループマッピングオブジェクトの配列。 プロパティの動作: - `provision_status` が「有効」に設定されている場合は必須
provision_media	array	ユーザーの LDAP/SAML メディア属性（メールアドレスなど）を、通知の送信に使用する Zabbix ユーザーメディアにマッピングするためのメディアタイプマッピングオブジェクトの配列。
LDAP固有のプロパティ**:**		
name	string	ユーザーディレクトリの一意の名前。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合は必須
host	string	LDAP サーバーのホスト名、IP アドレス、または URI。 URI にはスキーマ（`ldap://`または `ldaps://`）、ホスト、およびポート（任意）を含める必要があります。 例: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合は必須
port	integer	LDAP サーバーのポート。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合は必須

プロパティ	タイプ	説明
base_dn	string	ユーザーアカウントへの LDAP ユーザーディレクトリのベースパス。 例: *ou=Users,dc=example,dc=org* *ou=Users,ou=system*（OpenLDAP の場合） *DC=company,DC=com*（Microsoft Active Directory の場合） *uid=%{user},dc=example,dc=com*（直接ユーザーバインディングの場合。プレースホルダー「*%{user}*」は必須） プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合は必須
search_attribute	string	ログイン要求で提供された情報からユーザーアカウントを識別するための LDAP ユーザーディレクトリ属性。 例: *uid*（OpenLDAP の場合） *sAMAccountName*（Microsoft Active Directory の場合） プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合は必須
bind_dn	string	LDAP サーバーへのバインドおよび検索に使用する LDAP サーバーアカウント。 直接ユーザーバインディングおよび匿名バインディングでは、`bind_dn` を空にする必要があります。 例: *uid=ldap_search,ou=system*（OpenLDAP の場合） *CN=ldap_search,OU=user_group,DC=company,DC=com*（Microsoft Active Directory の場合） *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
bind_password	string	LDAP サーバーへのバインドおよび検索に使用するアカウントの LDAP パスワード。 直接ユーザーバインディングおよび匿名バインディングでは、`bind_password` を空にする必要があります。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
description	string	ユーザーディレクトリの説明。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
group_basedn	string	グループへの LDAP ユーザーディレクトリのベースパス。LDAP ユーザーディレクトリでユーザーの所属確認を設定するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 例: *ou=Groups,dc=example,dc=com* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート

プロパティ	タイプ	説明
group_filter	string	ユーザーが所属する LDAP ユーザーディレクトリグループを取得するためのフィルター文字列。LDAP ユーザーディレクトリでユーザーの所属確認を設定するために使用されます。 group_membership が設定されている場合、ユーザーのプロビジョニング時には無視されます。 サポートされる group_filter プレースホルダー: *%{attr}* - 検索属性（search_attribute プロパティの値に置換）; *%{groupattr}* - グループ属性（group_member プロパティの値に置換）; *%{host}* - LDAP サーバーのホスト名、IP アドレス、または URI（host プロパティの値に置換）; *%{user}* - Zabbix ユーザーのユーザー名。 デフォルト: *(%{groupattr}=%{user})* 例: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* は、LDAP グループオブジェクトに値「*uid=User1,ou=Users,dc=example,dc=com*」の「*member*」属性が含まれている場合に「User1」と一致し、「User1」が所属するグループを返します; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* は、LDAP グループオブジェクトに group_member プロパティで指定された属性が含まれ、その値が「*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*」である場合に「User1」と一致し、「User1」が所属するグループを返します。 プロパティの動作: - idp_type が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
group_member	string	グループメンバーに関する情報を含む LDAP ユーザーディレクトリ属性。LDAP ユーザーディレクトリでユーザーの所属確認を設定するために使用されます。 group_membership が設定されている場合、ユーザーのプロビジョニング時には無視されます。 プロパティの動作: - idp_type が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
group_membership	string	ユーザーが所属するグループに関する情報を含む LDAP ユーザーディレクトリ属性。 例: *memberOf* プロパティの動作: - idp_type が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
search_filter	string	ログイン要求で提供された情報に基づいて、LDAP ユーザーディレクトリ内のユーザーを検索および認証するために使用するカスタムフィルター文字列。 サポートされる search_filter プレースホルダー: *%{attr}* - 検索属性名（例: *uid*、*sAMAccountName*）; *%{user}* - Zabbix ユーザーのユーザー名。 デフォルト: *(%{attr}=%{user})* プロパティの動作: - idp_type が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート

プロパティ	タイプ	説明
start_tls	integer	LDAP サーバーとの通信を Transport Layer Security（TLS）で保護できる LDAP サーバー設定オプション。 `ldaps://`プロトコルを使用するホストでは、`start_tls` を「無効」に設定する必要があることに注意してください。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。 プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
user_ref_attr	string	ユーザーオブジェクトを参照するために使用する LDAP ユーザーディレクトリ属性。`user_ref_attr` の値は、ユーザーディレクトリ内の指定された属性から値を取得し、`group_filter` 文字列内の*%{ref}* プレースホルダーの代わりに挿入するために使用されます。 例: *cn*、*uid*、*member*、*uniqueMember* プロパティの動作: - `idp_type` が「LDAP タイプのユーザーディレクトリ」に設定されている場合にサポート
SAML固有のプロパティ:		
idp_entityid	string	ID プロバイダーを識別し、SAML メッセージで ID プロバイダーとの通信に使用する URI。 例: *https://idp.example.com/idp* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合は必須
sp_entityid	string	ID プロバイダーのサービスプロバイダーを識別する URL または任意の文字列。 例: *https://idp.example.com/sp* *zabbix* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合は必須
username_attribute	string	認証時にUser オブジェクトの `username` プロパティの値と比較されるユーザー名を含む SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性も含む）。 例: *uid*、*userprincipalname*、*samaccountname*、*username*、*userusername*、*urn:oid:0.9.2342.19200300.100.1.1*、*urn:oid:1.3.6.1.4.1.5923.1.1.1.13*、*urn:oid:0.9.2342.19200300.100.1.44* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合は必須

プロパティ	タイプ	説明
sso_url	string	ID プロバイダーの SAML シングルサインオンサービスの URL。Zabbix はこの URL に SAML 認証要求を送信します。 例: *http://idp.example.com/idp/sso/saml* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合は必須
slo_url	string	ID プロバイダーの SAML シングルログアウトサービスの URL。Zabbix はこの URL に SAML ログアウト要求を送信します。 例: *https://idp.example.com/idp/slo/saml* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
encrypt_nameid	integer	SAML name ID を暗号化するかどうか。 指定可能な値: 0 - *(デフォルト)* name ID を暗号化しない; 1 - name ID を暗号化する。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
encrypt_assertions	integer	SAML アサーションを暗号化するかどうか。 指定可能な値: 0 - *(デフォルト)* アサーションを暗号化しない; 1 - アサーションを暗号化する。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
nameid_format	string	SAML ID プロバイダーのサービスプロバイダーの name ID 形式。 例: *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
scim_status	integer	SAML の SCIM プロビジョニングを有効にするか無効にするか。 指定可能な値: 0 - *(デフォルト)* SCIM プロビジョニングは無効; 1 - SCIM プロビジョニングは有効。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート

プロパティ	タイプ	説明
sign_assertions	integer	SAML アサーションに SAML 署名を付与するかどうか。 指定可能な値: 0 - アサーションに署名しない; 1 - *(デフォルト)* アサーションに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_authn_requests	integer	SAML AuthN 要求に SAML 署名を付与するかどうか。 指定可能な値: 0 - AuthN 要求に署名しない; 1 - *(デフォルト)* AuthN 要求に署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_messages	integer	SAML メッセージに SAML 署名を付与するかどうか。 指定可能な値: 0 - メッセージに署名しない; 1 - *(デフォルト)* メッセージに署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_logout_requests	integer	SAML ログアウト要求に SAML 署名を付与するかどうか。 指定可能な値: 0 - ログアウト要求に署名しない; 1 - *(デフォルト)* ログアウト要求に署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
sign_logout_responses	integer	SAML ログアウト応答に SAML 署名を付与するかどうか。 指定可能な値: 0 - ログアウト応答に署名しない; 1 - *(デフォルト)* ログアウト応答に署名する。 プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定されている場合にサポート
idp_certificate	string	SAML シングルサインオン（SSO）サービス設定用のサービスプロバイダー（SP）証明書の内容。 例: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* プロパティの動作: - `idp_type` が「SAML タイプのユーザーディレクトリ」に設定され、Web インターフェース設定ファイルで `$SSO['CERT_STORAGE']` が `database` に設定されている場合にサポート - 書き込み専用

プロパティ	タイプ	説明
sp_private_key	string	SAML シングルサインオン（SSO）サービス設定用のサービスプロバイダー（SP）秘密鍵の内容。 ID プロバイダー（IdP）との安全な認証およびデータ交換を提供します。 例: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、Web インターフェース設定ファイルで $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 書き込み専用
sp_certificate	string	SAML シングルサインオン（SSO）サービス設定用のサービスプロバイダー（SP）証明書の内容。 例: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、Web インターフェース設定ファイルで $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 書き込み専用
idp_certificate_hash	string	idp_certificate 値の md5 ハッシュ。空の idp_certificate の場合は空の文字列を返します。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、Web インターフェース設定ファイルで $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 読み取り専用
sp_private_key_hash	string	sp_private_key 値の md5 ハッシュ。空の sp_private_key の場合は空の文字列を返します。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、Web インターフェース設定ファイルで $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 読み取り専用
sp_certificate_hash	string	sp_certificate 値の md5 ハッシュ。空の sp_certificate の場合は空の文字列を返します。 プロパティの動作: - idp_type が「SAML タイプのユーザーディレクトリ」に設定され、Web インターフェース設定ファイルで $SSO['CERT_STORAGE'] が database に設定されている場合にサポート - 読み取り専用

メディアタイプマッピング

メディアタイプマッピングオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
userdirectory_mediaid	ID	メディアタイプマッピング ID。 プロパティの動作: - 読み取り専用
name	string	メディアタイプマッピングの一覧に表示される名前。 プロパティの動作: - 必須
mediatypeid	ID	作成されるメディアタイプの ID。メディアオブジェクト のプロパティ `mediatypeid` の値として使用されます。 プロパティの動作: - 必須
attribute	string	ユーザーのメディア（例: *user@example.com*）を含む LDAP/SAML ユーザーディレクトリ属性（`scim_status` が「SCIM プロビジョニングが有効」に設定されている場合は SCIM 属性でも可）。この値は、メディアオブジェクト のプロパティ `sendto` の値として使用されます。 LDAP/SAML アイデンティティプロバイダーから受信したデータ内にこの属性が存在し、かつ値が空でない場合、プロビジョニングされたユーザーのメディア作成がトリガーされます。 プロパティの動作: - 必須
active	integer	プロビジョニングされたユーザーに対してメディアが作成される際の、ユーザーメディア `active` プロパティの値。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
severity	integer	プロビジョニングされたユーザーに対してメディアが作成される際の、ユーザーメディア `severity` プロパティの値。 デフォルト: 63。
period	string	プロビジョニングされたユーザーに対してメディアが作成される際の、ユーザーメディア `period` プロパティの値。 デフォルト: 1-7,00:00-24:00。

プロビジョニンググループマッピング

プロビジョニンググループマッピングには、以下のプロパティがあります。

プロパティ	型	説明
name	string	LDAP/SAML ユーザーディレクトリ内のグループのフルネーム（例: *Zabbix administrators*）です（`scim_status` が「SCIM provisioning is enabled」に設定されている場合は、SCIM も含みます）。ワイルドカード文字「*」をサポートします。 すべてのプロビジョニンググループマッピング内で一意である必要があります。 プロパティの動作: - 必須
roleid	ID	ユーザーに割り当てるユーザーロールの ID です。 複数のプロビジョニンググループマッピングが一致した場合、最も高いユーザータイプ（*User*、*Admin*、または *Super admin*）のロールがユーザーに割り当てられます。同じユーザータイプのロールが複数ある場合は、最初のロール（アルファベット順でソート）がユーザーに割り当てられます。 プロパティの動作: - 必須

プロパティ	型	説明
user_groups	array	Zabbix ユーザーグループ ID オブジェクトの配列です。各オブジェクトには以下のプロパティがあります。 usrgrpid - (ID) ユーザーに割り当てる Zabbix ユーザーグループの ID。 複数のプロビジョニンググループマッピングが一致した場合、一致したすべてのマッピングの Zabbix ユーザーグループがユーザーに割り当てられます。 プロパティの動作: - 必須

userdirectory.create

説明

`object userdirectory.create(object/array userDirectory)`

このメソッドを使用すると、新しいユーザーディレクトリを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。

パラメータ

(object/array) 作成するユーザーディレクトリ。

このメソッドは、標準ユーザーディレクトリプロパティを持つユーザーディレクトリを受け入れます。

戻り値

(object) は、userdirectoryids プロパティに作成されたユーザーディレクトリの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーディレクトリの順序と一致します。

例

ユーザーディレクトリの作成

StartTLS over LDAP を使用してユーザーを認証するためのユーザーディレクトリを作成します。LDAP 経由でユーザーを認証するには、LDAP 認証 を有効にする必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
        "idp_type": "1",
        "name": "LDAP API server #1",
        "host": "ldap://local.ldap",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "ldapsecretpassword",
        "search_attribute": "uid",
        "start_tls": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
```

```
    },
    "id": 1
}
```

ユーザーディレクトリの作成（JIT プロビジョニング有効）

LDAP 経由でユーザーを認証するためのユーザーディレクトリを作成します。（JIT プロビジョニング有効）LDAP 経由でユーザーを認証するには、LDAP 認証 が有効になっている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
```

```
        },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CUserDirectory::create() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

説明

`object userdirectory.delete(array userDirectoryIds)`

このメソッドを使用すると、ユーザーディレクトリを削除できます。ユーザーディレクトリは、少なくとも 1 つのユーザーグループで直接使用されている場合は削除できません。
 デフォルトの LDAP ユーザーディレクトリは、`authentication.ldap_configured` が 1 に設定されている場合、または他のユーザーディレクトリが残っている場合は削除できません。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ利用できます。

パラメータ

(`array`) 削除するユーザーディレクトリの ID。

戻り値

(`object`) `userdirectoryids` プロパティの下にある削除されたユーザーディレクトリの ID を含むオブジェクトを返します。

例

複数のユーザーディレクトリの削除

2 つのユーザーディレクトリを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
```

```
    },
    "id": 1
}
```

ソース

CUserDirectory::delete() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

説明

`integer/array userdirectory.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってユーザーディレクトリを取得できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
userdirectoryids	ID/array	指定した ID を持つユーザーディレクトリのみを返します。
selectUsrgrps	query	ユーザーディレクトリに関連付けられたユーザーグループを含む usrgrps プロパティを返します。 count をサポートします。
selectProvisionMedia	query	ユーザーディレクトリに関連付けられたメディアタイプマッピングを含む provision_media プロパティを返します。
selectProvisionGroups	query	ユーザーディレクトリに関連付けられたプロビジョニンググループマッピングを含む provision_groups プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: name。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、値を単一の値または値の配列とするオブジェクトを受け付けます。 サポートされるプロパティ: userdirectoryid, idp_type, provision_status。
search	object	指定したパターンに一致する結果を返します（大文字・小文字を区別しません）。 プロパティ名をキー、検索する文字列を値とするオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索が実行されます。 サポートされるプロパティ: name, description。 SAML タイプのユーザーディレクトリでは、name と description の両方のプロパティは空の値になります。両方のプロパティは userdirectory.update 操作で変更できます。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

パラメータ	型	説明
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

ユーザーディレクトリの取得

各ユーザーディレクトリに関連付けられたメディアタイプマッピングとプロビジョニンググループマッピングを表示する追加プロパティを含むすべてのユーザーディレクトリを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
       {
            "userdirectoryid": "1",
            "idp_type": "2",
            "name": "",
            "provision_status": "1",
            "description": "",
            "group_name": "groups",
            "user_username": "",
            "user_lastname": "",
            "idp_entityid": "http://example.com/simplesaml/saml2/idp/metadata.php",
            "sso_url": "http://example.com/simplesaml/saml2/idp/SSOService.php",
            "slo_url": "",
            "username_attribute": "uid",
            "sp_entityid": "zabbix",
            "nameid_format": "",
            "sign_messages": "0",
            "sign_assertions": "0",
            "sign_authn_requests": "0",
            "sign_logout_requests": "0",
            "sign_logout_responses": "0",
            "encrypt_nameid": "0",
            "encrypt_assertions": "0",
            "scim_status": "1",
            "provision_media": [
                {
                    "userdirectory_mediaid": "1",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
```

```
                "severity": "63",
                "period": "1-7,00:00-24:00"
            }
        ],
        "provision_groups": [
            {
                "name": "*",
                "roleid": "1",
                "user_groups": [
                    {
                        "usrgrpid": "13"
                    }
                ]
            }
        ]
    },
    {
        "userdirectoryid": "2",
        "idp_type": "1",
        "name": "AD server",
        "provision_status": "1",
        "description": "",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "start_tls": "0",
        "search_filter": "",
        "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
        "group_name": "CN",
        "group_member": "member",
        "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
        "group_membership": "",
        "user_username": "givenName",
        "user_lastname": "sn",
        "user_ref_attr": "CN",
        "provision_media": [
            {
                "userdirectory_mediaid": "2",
                "name": "example.com",
                "mediatypeid": "1",
                "attribute": "user@example.com",
                "active": "0",
                "severity": "63",
                "period": "1-7,00:00-24:00"
            }
        ],
        "provision_groups": [
            {
                "name": "*",
                "roleid": "4",
                "user_groups": [
                    {
                        "usrgrpid": "8"
                    }
                ]
            },
            {
                "name": "Zabbix administrators",
                "roleid": "2",
                "user_groups": [
```

```
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API server #1",
            "provision_status": "0",
            "description": "",
            "host": "ldap://local.ldap",
            "port": "389",
            "base_dn": "ou=Users,dc=example,dc=org",
            "search_attribute": "uid",
            "bind_dn": "cn=ldap_search,dc=example,dc=org",
            "start_tls": "1",
            "search_filter": "",
            "group_basedn": "",
            "group_name": "",
            "group_member": "",
            "group_filter": "",
            "group_membership": "",
            "user_username": "",
            "user_lastname": "",
            "user_ref_attr": "",
            "provision_media": [],
            "provision_groups": []
        }
    ],
    "id": 1
}
```


参照

- User group

ソース

CUserDirectory::get() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

説明

`object userdirectory.test(array userDirectory)`

このメソッドを使用すると、ユーザーディレクトリの接続設定をテストできます。

Note:
このメソッドでは、ユーザープロビジョニング用のユーザーディレクトリ設定に対して、構成済みデータがどのように一致するかもテストできます (たとえば、どのユーザーロール、ユーザーグループ、ユーザーメディアがユーザーに割り当てられるかを確認できます)。この種類のテストでは、`provision_status` が有効に設定されたuser directory に対して API リクエストを送信する必要があります。

Note:
このメソッドは *Super admin* ユーザータイプでのみ利用できます。

パラメータ

(object) ユーザーディレクトリのプロパティ。

userdirectory.get API は bind_password フィールドを返さないため、userdirectoryid および/または bind_password を指定する必要があります。
 また、標準のユーザーディレクトリプロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
test_username	string	ユーザーディレクトリでテストするユーザー名。
test_password	string	ユーザーディレクトリでテストするユーザー名に対応するパスワード。

戻り値

(bool) 成功した場合は true を返します。

例

既存のユーザーのユーザーディレクトリのテスト

"user1" のユーザーディレクトリ"3" をテストします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "password",
        "test_username": "user1",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

存在しないユーザーのユーザーディレクトリのテスト

存在しない"user2" でユーザーディレクトリ"3" のテストをします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "test_username": "user2",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

ユーザープロビジョニングのユーザーディレクトリのテスト

"user3" プロビジョニングのユーザーディレクトリ設定と一致する構成されたデータのユーザーディレクトリ"3" をテストします (例えば、何かユーザーの役割、ユーザーグループ、ユーザーメディアがユーザーに割り当てられます)。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "2",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "test_username": "user3",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

ソース

CUserDirectory::test() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

説明

`object userdirectory.update(object/array userDirectory)`

このメソッドを使用すると、既存のユーザーディレクトリを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。

パラメーター

(object/array) 更新するユーザーディレクトリのプロパティ。

userdirectoryid プロパティは各ユーザーディレクトリに対して定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

戻り値

(object) userdirectoryids プロパティの下にある更新されたユーザーディレクトリの ID を含むオブジェクトを返します。

例

ユーザーディレクトリのバインドパスワードの更新

ユーザーディレクトリの新しいバインドパスワードを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

ユーザーディレクトリのマッピングの更新

ユーザーディレクトリ"2" のプロビジョニンググループマッピングとメディアタイプマッピングを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "2",
        "provision_media": [
                {
                    "userdirectory_mediaid": "2"
                }
            ],
            "provision_groups": [
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
```

```
                        {
                            "usrgrpid": "8"
                        },
                        {
                            "usrgrpid": "11"
                        }
                    ]
                }
            ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CUserDirectory::update() in *ui/include/classes/api/services/CUserDirectory.php*.

ユーザーマクロ

このクラスは、ホストレベルおよびグローバルのユーザーマクロを扱うために設計されています。

オブジェクト参照:

- グローバルマクロ
- ホストマクロ
    - マクロ設定

利用可能なメソッド:

- usermacro.create - 新しいホストマクロを作成
- usermacro.createglobal - 新しいグローバルマクロを作成
- usermacro.delete - ホストマクロを削除
- usermacro.deleteglobal - グローバルマクロを削除
- usermacro.get - ホストマクロおよびグローバルマクロを取得
- usermacro.update - ホストマクロを更新
- usermacro.updateglobal - グローバルマクロを更新

User macro オブジェクト

次のオブジェクトは、usermacroAPI に直接関連しています。

グローバルマクロ

グローバルマクロオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
globalmacroid	ID	グローバルマクロの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
description	string	マクロの説明。

プロパティ	型	説明
macro	string	マクロ文字列。 プロパティの動作: - 作成操作では必須
type	integer	マクロのタイプ。 指定可能な値: 0 - *(デフォルト)* テキストマクロ; 1 - シークレットマクロ; 2 - Vault シークレット。
value	string	マクロの値。 プロパティの動作: - type が「シークレットマクロ」に設定されている場合は 書き込み専用 - 作成操作では必須

ホストマクロ

ホストマクロオブジェクトは、ホスト、ホストプロトタイプ、またはテンプレートで使用可能なマクロを定義します。以下のプロパティがあります。

Property	Type	Description
hostmacroid	ID	マクロの ID。 Property behavior: - 読み取り専用 - 更新操作で 必須
automatic	integer	マクロがディスカバリルールによって制御されるかどうかを定義します。 使用可能な値: 0 - *(デフォルト)* マクロはユーザーによって管理されます。 1 - マクロはディスカバリルールによって管理されます。 ユーザーは自動マクロを作成できません。 自動マクロを更新するには、手動に変換する必要があります。
config	object/array	ホストウィザードでマクロをどのように表示するかを制御する#macro-configurationマクロ設定。
description	string	マクロの説明。
hostid	ID	このマクロが属するホスト、ホストプロトタイプ、またはテンプレートの ID。 Property behavior: - 定数 - 作成操作で 必須
macro	string	マクロ文字列。 Property behavior: - 作成操作で 必須
type	integer	マクロのタイプ。 使用可能な値: 0 - *(デフォルト)* テキストマクロ。 1 - シークレットマクロ。 2 - Vault シークレット。
value	string	マクロの値。 Property behavior: - type が”Secret macro” に設定されている場合は 書き込み専用 - 作成操作で 必須

マクロ設定

マクロ設定オブジェクトは、ホストウィザードでマクロがどのように表示されるかを定義します。

プロパティ	型	説明
type	integer	マクロ入力フィールドのタイプ。 使用可能な値: 0 - マクロはホストウィザードで使用されない; 1 - テキストボックス; 2 - リスト; 3 - チェックボックス。 プロパティの動作: - 必須
label	string	マクロ入力フィールドのラベル。 プロパティの動作: - type が”Textbox”、”List”、または”Checkbox” に設定されている場合は必須
description	text	マクロ入力フィールドの横に表示されるヘルプテキスト。Markdown 形式をサポートします。
priority	integer	マクロリスト内でのマクロ入力フィールドの位置。 priority が設定されていない場合、マクロはグループ化されていないマクロリストの末尾に追加されます。
required	integer	マクロを必須としてマークします。 使用可能な値: 0 - 必須ではない; 1 - 必須。 プロパティの動作: - type が”Textbox” または”List” に設定されている場合に サポート
regex	string	テキストボックスフィールド内のユーザー入力を検証するための正規表現。 プロパティの動作: - type が”Textbox” に設定されている場合に サポート
section_name	string	マクロがグループ化される折りたたみ可能なセクションのラベル。 section_name が設定されていない場合、マクロはグループ化されません。グループ化されていないマクロが最初に表示され、グループ化されたマクロはその下に、各折りたたみ可能セクション内で priority 順に表示されます。
options	text	リスト項目またはチェックボックスの値を定義する JSON 文字列。 リストの場合: value および text プロパティを持つオブジェクトの配列。 例: `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` チェックボックスの場合: checked および unchecked プロパティを持つオブジェクト。 例: `{"checked": true, "unchecked": false}` プロパティの動作: - type が”List” または”Checkbox” に設定されている場合は 必須

usermacro.create

説明

`object usermacro.create(object/array hostMacros)`

このメソッドを使用すると、新しいホストマクロを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するホストマクロ。

このメソッドは、標準のホストマクロプロパティを持つホストマクロを受け付けます。

戻り値

(object) hostmacroids プロパティの下で作成されたホストマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストマクロの順序と一致します。

例

ホストマクロの作成

ホスト"10198" に値"public" のホストマクロ {$SNMP_COMMUNITY} を作成します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::create() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

説明

`object usermacro.createglobal(object/array globalMacros)`

このメソッドを使用すると、新しいグローバルマクロを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成するグローバルマクロ。

このメソッドは、標準のグローバルマクロのプロパティを持つグローバルマクロを受け付けます。

戻り値

(object) globalmacroids プロパティの下で作成されたグローバルマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグローバルマクロの順序と一致します。

例

グローバルマクロの作成

値が"public" のグローバルマクロ {$SNMP_COMMUNITY} を作成します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::createGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

説明

`object usermacro.delete(array hostMacroIds)`

このメソッドを使用すると、ホストマクロを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するホストマクロの ID。

戻り値

(object) hostmacroids プロパティの下にある削除されたホストマクロの ID を含むオブジェクトを返します。

例

複数のホストマクロの削除

2 つのホストマクロを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
```

```
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::delete() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

説明

`object usermacro.deleteglobal(array globalMacroIds)`

このメソッドを使用すると、グローバルマクロを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するグローバルマクロの ID。

戻り値

(object) globalmacroids プロパティの下にある削除されたグローバルマクロの ID を含むオブジェクトを返します。

例

複数のグローバルマクロの削除

2 つのグローバルマクロを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
```

```
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::deleteGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

説明

`integer/array usermacro.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってホストマクロおよびグローバルマクロを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 希望する出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
globalmacro	flag	ホストマクロの代わりにグローバルマクロを返します。
globalmacroids	ID/array	指定された ID を持つグローバルマクロのみを返します。
groupids	ID/array	指定されたホストグループまたはテンプレートグループのホストまたはテンプレートに属するホストマクロのみを返します。
hostids	ID/array	指定されたホストまたはテンプレートに属するマクロのみを返します。
hostmacroids	ID/array	指定された ID を持つホストマクロのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたホストプロトタイプユーザーマクロのみを返します。
selectHostGroups	query	ホストマクロが属するホストグループを`hostgroups` プロパティで返します。 ホストマクロを取得する場合にのみ使用されます。
selectHosts	query	ホストマクロが属するホストを`hosts` プロパティで返します。 ホストマクロを取得する場合にのみ使用されます。
selectTemplateGroups	query	テンプレートマクロが属するテンプレートグループを`templategroups` プロパティで返します。 テンプレートマクロを取得する場合にのみ使用されます。
selectTemplates	query	ホストマクロが属するテンプレートを`templates` プロパティで返します。 ホストマクロを取得する場合にのみ使用されます。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `macro`。
countOutput	boolean	これらのパラメータはリファレンス解説 で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

パラメータ	型	説明
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ホストのホストマクロの取得

ホスト"10198" に定義されているすべてのホストマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "9",
            "hostid": "10198",
            "macro": "{$INTERFACE}",
            "value": "eth0",
            "description": "",
            "type": "0",
            "automatic": "0"
        },
        {
            "hostmacroid": "11",
            "hostid": "10198",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0",
            "automatic": "0"
        }
    ],
    "id": 1
}
```

テンプレートのホストマクロの取得

マクロ名に"STATUS" を含むテンプレート"10265" に定義されたホストマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
```

```
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "The hostname or IP address of the Apache status page host.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "1",
                "section_name": "",
                "label": "Apache status host",
                "description": "Hostname or IP address of the Apache status page host.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "The URL path of the Apache status page.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Apache status page path",
                "description": "URL path of the Apache status page.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "The port of the Apache status page.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
```

```
                "label": "Apache status page port",
                "description": "In the range from 1 to 65535 inclusive.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "The request scheme, which may be either HTTP or HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Request scheme",
                "description": "Request scheme, which may be either HTTP or HTTPS.",
                "required": "0",
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```


グローバルマクロの取得

すべてのグローバルマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
```

```
            "type": "0"
        }
    ],
    "id": 1
}
```

ソース

CUserMacro::get() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

説明

`object usermacro.update(object/array hostMacros)`

このメソッドを使用すると、既存のホストマクロを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー役割の設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

更新する（object/array）のホストマクロのプロパティ。

各ホストマクロには hostmacroid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

戻り値

（object）hostmacroids プロパティの下で更新されたホストマクロの ID を含むオブジェクトを返します。

例

ホストマクロの値の変更

ホストマクロの値を”public” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

検出ルールによって作成されたマクロの値を変更します。

検出ルールによって作成された” 自動” マクロを” 手動” マクロに変換し、その値を” 新しい値” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::update() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

説明

`object usermacro.updateglobal(object/array globalMacros)`

このメソッドを使用すると、既存のグローバルマクロを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

更新する（object/array）グローバルマクロのプロパティ。

各グローバルマクロでは globalmacroid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

戻り値

（object）globalmacroids プロパティの下で更新されたグローバルマクロの ID を含むオブジェクトを返します。

例

グローバルマクロの値の変更

グローバルマクロの値を”public” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::updateGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

レポート

このクラスは、スケジュールされたレポートを扱うために設計されています。

オブジェクト参照:

- レポート
- ユーザー
- ユーザーグループ

利用可能なメソッド:

- report.create - 新しいスケジュールされたレポートを作成
- report.delete - スケジュールされたレポートを削除
- report.get - スケジュールされたレポートを取得
- report.update - スケジュールされたレポートを更新

Report オブジェクト

次のオブジェクトは、reportAPI に直接関連しています。

レポート

レポートオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
reportid	ID	レポートの ID。 プロパティの動作: - 読み取り専用 - 更新操作で必須
userid	ID	レポートを作成したユーザーの ID。 プロパティの動作: - 作成操作で必須
name	string	一意のレポート名。 プロパティの動作: - 作成操作で必須
dashboardid	ID	レポートの基となるダッシュボードの ID。 プロパティの動作: - 作成操作で必須
period	integer	レポートを作成する対象期間。 指定可能な値: 0 - *(デフォルト)* 前日; 1 - 前週; 2 - 前月; 3 - 前年。

プロパティ	タイプ	説明
cycle	integer	期間の繰り返しスケジュール。 指定可能な値: 0 - *(デフォルト)* 毎日; 1 - 毎週; 2 - 毎月; 3 - 毎年。
start_time	integer	レポートを作成して送信する時刻。1 日の開始からの秒数で指定します。 デフォルト: 0。
weekdays	integer	レポートを送信する曜日。 指定可能なビットマップ値: 1 - 月曜日; 2 - 火曜日; 4 - 水曜日; 8 - 木曜日; 16 - 金曜日; 32 - 土曜日; 64 - 日曜日。 これはビットマスクフィールドです。指定可能なビットマップ値の合計を指定できます（例: 月曜日、水曜日、金曜日の場合は 21）。 デフォルト: 0。 プロパティの動作: - `cycle` が”weekly” に設定されている場合に必須
active_since	string	開始日。 指定可能な値: 空の文字列 - *(デフォルト)* 未指定（0 として保存）; YYYY-MM-DD 形式の特定の日付（その日の開始時刻（00:00:00）のタイムスタンプとして保存）。
active_till	string	終了日。 指定可能な値: 空の文字列 - *(デフォルト)* 未指定（0 として保存）; YYYY-MM-DD 形式の特定の日付（その日の終了時刻（23:59:59）のタイムスタンプとして保存）。
subject	string	レポートメッセージの件名。
message	string	レポートメッセージの本文。
status	integer	レポートが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
description	text	レポートの説明。
state	integer	レポートの状態。 指定可能な値: 0 - *(デフォルト)* レポートがまだ処理されていない; 1 - レポートが生成され、すべての受信者に正常に送信された; 2 - レポートの生成に失敗した。”info” にエラー情報が含まれる; 3 - レポートは生成されたが、一部またはすべての受信者への送信に失敗した。”info” にエラー情報が含まれる。 プロパティの動作: - 読み取り専用

プロパティ	タイプ	説明
lastsent	timestamp	最後に正常に送信されたレポートの Unix タイムスタンプ。 プロパティの動作: - 読み取り専用
info	string	エラーの説明または追加情報。 プロパティの動作: - 読み取り専用

ユーザー

users オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
userid	ID	レポートの送信先ユーザーの ID。 プロパティの動作: - 必須
access_userid	ID	レポートが代理で生成されるユーザーの ID。 0 - *(デフォルト)* 受信者によってレポートを生成。
exclude	integer	ユーザーをメーリングリストから除外するかどうか。 使用可能な値: 0 - *(デフォルト)* 含める; 1 - 除外。

ユーザーグループ

ユーザーグループオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
usrgrpid	ID	レポートの送信先となるユーザーグループの ID。 プロパティの動作: - 必須
access_userid	ID	レポートの生成時に代理として使用されるユーザーの ID。 0 - *(デフォルト)* 受信者としてレポートを生成。

report.create

説明

`object report.create(object/array reports)`

このメソッドを使用すると、新しいスケジュールレポートを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザー役割の設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するスケジュールレポート。

このメソッドは、標準のスケジュールレポートのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
users	object/array	レポートの送信先となるユーザー。 パラメータの動作: - `user_groups` が設定されていない場合は必須
user_groups	object/array	レポートの送信先となるユーザーグループ。 パラメータの動作: - `users` が設定されていない場合は必須

戻り値

(`object`) `reportids` プロパティの下で作成されたスケジュールされたレポートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスケジュール済みレポートの順序と一致します。

例

スケジュールレポートの作成

2021-04-01 から 2021-08-31 まで、毎週月曜日から金曜日の 12:00 に前週分の週次レポートを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.create",
    "params": {
        "userid": "1",
        "name": "Weekly report",
        "dashboardid": "1",
        "period": "1",
        "cycle": "1",
        "start_time": "43200",
        "weekdays": "31",
        "active_since": "2021-04-01",
        "active_till": "2021-08-31",
        "subject": "Weekly report",
        "message": "Report accompanying text",
        "status": "1",
        "description": "Report description",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            },
            {
                "userid": "2",
                "access_userid": "0",
                "exclude": "1"
            }
        ],
        "user_groups": [
            {
                "usrgrpid": "7",
                "access_userid": "0"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- ユーザー
- ユーザーグループ

ソース

CReport::create() in *ui/include/classes/api/services/CReport.php*.

report.delete

説明

`object report.delete(array reportids)`

このメソッドを使用すると、スケジュールされたレポートを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除する定期レポートの ID。

戻り値

(object) reportids プロパティの下にある削除された定期レポートの ID を含むオブジェクトを返します。

例

複数の定期レポートの削除

2 つの定期レポートの削除

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CReport::delete() in *ui/include/classes/api/services/CReport.php*.

report.get

説明

`integer/array report.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従ってスケジュールされたレポートを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
reportids	ID/array	指定されたレポート ID を持つ定期レポートのみを返します。
expired	boolean	`true` に設定すると、期限切れの定期レポートのみを返します。
selectUsers	query	レポートの送信先として設定されているユーザーを含む`users` プロパティを返します。
selectUserGroups	query	レポートの送信先として設定されているユーザーグループを含む`user_groups` プロパティを返します。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `reportid`, `name`, `status`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント を参照してください。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

レポートデータの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": {
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "Weekly report",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Weekly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "Report description",
            "state": "1",
            "lastsent": "1613563219",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                },
                {
                    "userid": "2",
                    "access_userid": "0",
                    "exclude": "1"
                }
            ],
            "user_groups": [
                {
                    "usrgrpid": "7",
                    "access_userid": "0"
                }
            ]
        },
        {
            "reportid": "2",
            "userid": "1",
            "name": "Monthly report",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Monthly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
```

```
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```

関連項目

- ユーザー
- ユーザーグループ

ソース

CReport::get() in *ui/include/classes/api/services/CReport.php*.

report.update

説明

`object report.update(object/array reports)`

このメソッドを使用すると、既存の定期レポートを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新する定期レポートのプロパティ。

各定期レポートでは `reportid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準の定期レポートのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
users	object/array	定期レポートに現在割り当てられているユーザーを置き換えるためのユーザー。 パラメータの動作: - `user_groups` が設定されていない場合は 必須
user_groups	object/array	定期レポートに現在割り当てられているユーザーグループを置き換えるためのユーザーグループ。 パラメータの動作: - `users` が設定されていない場合は 必須

戻り値

(`object`) `reportids` プロパティの下で更新されたスケジュールされたレポートの ID を含むオブジェクトを返します。

例

スケジュールレポートの無効化

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- ユーザー
- ユーザーグループ

ソース

CReport::update() in *ui/include/classes/api/services/CReport.php*.

ロール

このクラスはユーザーロールを扱うために設計されています。

オブジェクト参照:

- ロール
- ロールルール
    - UI 要素
    - サービス
    - サービスタグ
    - モジュール
    - アクション

利用可能なメソッド:

- role.create - 新しいユーザーロールを作成
- role.delete - ユーザーロールを削除
- role.get - ユーザーロールを取得
- role.update - ユーザーロールを更新

Role オブジェクト

次のオブジェクトは、`roleAPI` に直接関連しています。

ロール

ロールオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
roleid	ID	ロールの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須

プロパティ	型	説明
name	string	ロールの名前。 プロパティの動作: - 作成操作で 必須
type	integer	ユーザータイプ。 指定可能な値: 1 - *(デフォルト)* ユーザー; 2 - 管理者; 3 - 特権管理者。 プロパティの動作: - 作成操作で 必須
readonly	integer	ロールが読み取り専用かどうか。 指定可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。 プロパティの動作: - 読み取り専用

ロールルール

ロールルールオブジェクトには、次のプロパティがあります。

プロパティ	型	説明
ui	array	UI 要素オブジェクトの配列。
ui.default_access	integer	新しい UI 要素へのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
services.read.mode	integer	サービスへの読み取り専用アクセス。 指定可能な値: 0 - `services.read.list` プロパティで指定されたサービス、または `services.read.tag` プロパティに一致するサービスへの読み取り専用アクセス; 1 - *(デフォルト)* すべてのサービスへの読み取り専用アクセス。
services.read.list	array	サービスオブジェクトの配列。 指定されたサービスとその子サービスへの読み取り専用アクセスが、ユーザーロールに付与されます。読み取り専用アクセスによって、サービスへの読み取り/書き込みアクセスが上書きされることはありません。 プロパティの動作: - `services.read.mode` が「0」に設定されている場合にサポート
services.read.tag	object	サービスタグオブジェクトの配列。 タグに一致するサービスとその子サービスへの読み取り専用アクセスが、ユーザーロールに付与されます。読み取り専用アクセスによって、サービスへの読み取り/書き込みアクセスが上書きされることはありません。 プロパティの動作: - `services.read.mode` が「0」に設定されている場合にサポート

プロパティ	型	説明
services.write.mode	integer	サービスへの読み取り/書き込みアクセス。 指定可能な値: 0 - *(デフォルト)* services.write.list プロパティで指定されたサービス、または services.write.tag プロパティに一致するサービスへの読み取り/書き込みアクセス; 1 - すべてのサービスへの読み取り/書き込みアクセス。
services.write.list	array	サービスオブジェクトの配列。 指定されたサービスとその子サービスへの読み取り/書き込みアクセスが、ユーザーロールに付与されます。読み取り/書き込みアクセスによって、サービスへの読み取り専用アクセスが上書きされます。 プロパティの動作: - services.write.mode が「0」に設定されている場合にサポート
services.write.tag	object	サービスタグオブジェクトの配列。 タグに一致するサービスとその子サービスへの読み取り/書き込みアクセスが、ユーザーロールに付与されます。読み取り/書き込みアクセスによって、サービスへの読み取り専用アクセスが上書きされます。 プロパティの動作: - services.write.mode が「0」に設定されている場合にサポート
modules	array	モジュールオブジェクトの配列。
modules.default_access	integer	新しいモジュールへのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
api.access	integer	API へのアクセスを有効にするかどうか。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
api.mode	integer	api プロパティに一覧表示された API メソッドの扱い方。 指定可能な値: 0 - *(デフォルト)* 拒否リスト; 1 - 許可リスト。
api	array	API メソッドの配列。
actions	array	アクションオブジェクトの配列。
actions.default_access	integer	新しいアクションへのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
devices.access	integer	Zabbix モバイルアプリで使用する、ユーザー自身のモバイルデバイスへのアクセスを有効にするかどうか。 指定可能な値: 0 - (ユーザーおよび管理者のデフォルト) 無効; 1 - (スーパーユーザーのデフォルト) 有効。 プロパティの動作: - EnableMobileDevicesが有効な場合にサポート
devices.actions	array	デバイスアクションオブジェクトの配列。 プロパティの動作: - devices.access が「1」に設定されている場合にサポート

プロパティ	型	説明
devices.actions.default_access	integer	新しいデバイスアクションへのアクセスを有効にするかどうか。 指定可能な値: 0 - (ユーザーおよび管理者のデフォルト) 無効; 1 - (スーパーユーザーのデフォルト) 有効。 プロパティの動作: - EnableMobileDevicesが有効な場合にサポート

UI 要素

UI 要素オブジェクトには、次のプロパティがあります。

Property	Type	Description
name	string	UI 要素の名前。 Roleオブジェクトの type が「User」、「Admin」、または「Super admin」に設定されている場合の使用可能な値: monitoring.dashboard - ダッシュボード; monitoring.problems - 監視 > 障害; monitoring.hosts - 監視 > ホスト; monitoring.latest_data - 監視 > 最新データ; monitoring.maps - 監視 > マップ; services.services - サービス > サービス; services.sla_report - サービス > *SLA* レポート; inventory.overview - インベントリ > 概要; inventory.hosts - インベントリ > ホスト; reports.availability_report - レポート > 可用性レポート; reports.top_triggers - レポート > トリガー上位 *100* 件。 Roleオブジェクトの type が「Admin」または「Super admin」に設定されている場合の使用可能な値: monitoring.discovery - 監視 > ディスカバリ; services.sla - サービス > *SLA*; reports.scheduled_reports - レポート > スケジュールレポート; reports.notifications - レポート > 通知; configuration.template_groups - データ収集 > テンプレートグループ; configuration.host_groups - データ収集 > ホストグループ; configuration.templates - データ収集 > テンプレート; configuration.hosts - データ収集 > ホスト; configuration.maintenance - データ収集 > メンテナンス; configuration.discovery - データ収集 > ディスカバリ; configuration.trigger_actions - アラート > アクション > トリガーアクション; configuration.service_actions - アラート > アクション > サービスアクション; configuration.discovery_actions - アラート > アクション > ディスカバリアクション; configuration.autoregistration_actions - アラート > アクション > 自動登録アクション; configuration.internal_actions - アラート > アクション > 内部アクション。 Roleオブジェクトの type が「Super admin」に設定されている場合の使用可能な値: reports.system_info - レポート > システム情報; reports.audit - レポート > 監査ログ; reports.action_log - レポート > アクションログ; configuration.event_correlation - データ収集 > イベント相関; administration.media_types - アラート > メディアタイプ; administration.scripts - アラート > スクリプト; administration.user_groups - ユーザー > ユーザーグループ; administration.user_roles - ユーザー > ユーザーロール; administration.users - ユーザー > ユーザー; administration.api_tokens - ユーザー > *API* トークン; administration.authentication - ユーザー > 認証; administration.linked_devices - ユーザー > デバイス; administration.general - 管理 > 一般設定; administration.audit_log - 管理 > 監査ログ; administration.housekeeping - 管理 > ハウスキーピング; administration.proxy_groups - 管理 > プロキシグループ; administration.proxies - 管理 > プロキシ; administration.macros - 管理 > マクロ; administration.queue - 管理 > キュー。 Property behavior: - *required*

Property	Type	Description
status	integer	UI 要素へのアクセスが有効かどうか。 使用可能な値: 0 - 無効; 1 - *(default)* 有効。

サービス

プロパティ	型	説明
serviceid	ID	サービスの ID。 プロパティの動作: - 必須

サービスタグ

プロパティ	Type	説明
tag	string	タグ名。 空文字列が指定された場合、サービスタグはサービスのマッチングに使用されません。 プロパティの動作: - 必須
value	string	タグ値。 値が指定されていない場合、または空文字列が指定された場合、サービスのマッチングにはタグ名のみが使用されます。

モジュール

モジュールオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
moduleid	ID	モジュールの ID。 プロパティの動作: - 必須
status	integer	モジュールへのアクセスが有効かどうか。 使用可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

アクション

アクションオブジェクトには、次のプロパティがあります。

Property	Type	Description
name	string	アクションの名前。 Role オブジェクトの type が「User」、「Admin」、または「Super admin」に設定されている場合の使用可能な値: edit_dashboards - ダッシュボードの作成および編集; edit_maps - マップの作成および編集; add_problem_comments - 障害コメントの追加; change_severity - 障害の深刻度の変更; acknowledge_problems - 障害の認識; suppress_problems - 障害の抑制; close_problems - 障害のクローズ; execute_scripts - スクリプトの実行; manage_api_tokens - API トークンの管理; change_problem_ranking - 障害のランキングを原因から症状へ、またはその逆に変更; edit_own_media - ユーザー自身のメディアの作成および編集を許可。 Role オブジェクトの type が「Admin」または「Super admin」に設定されている場合の使用可能な値: edit_maintenance - メンテナンスの作成および編集; manage_scheduled_reports - スケジュールレポートの管理、 manage_sla - SLA の管理、 select_server_for_monitoring - サーバーおよびサーバーが実行するネットワークディスカバリールールによって監視されるホストの作成をユーザーに許可。 Role オブジェクトの type が「User」または「Admin」に設定されている場合の使用可能な値: invoke_execute_now - ホストに対する読み取り権限のみを持つユーザーがアイテムのチェックを実行できるようにする。 Role オブジェクトの type が「Super admin」に設定されている場合の使用可能な値: edit_user_media - 他のユーザーのメディアを作成および編集することをユーザーに許可。 Property behavior: - *required*
status	integer	アクションを実行するアクセスが有効かどうか。 使用可能な値: 0 - 無効; 1 - *(default)* 有効。

デバイスアクション

デバイスアクションオブジェクトには、次のプロパティがあります。

プロパティ	タイプ	説明
name	string	デバイスアクションの名前。 Roleオブジェクトの `type` が「User」、「Admin」、または「Super admin」に設定されている場合の使用可能な値: `manage_own` - ユーザーが自分のモバイルデバイスを追加または削除できるようにします。 Roleオブジェクトの `type` が「Super admin」に設定されている場合の使用可能な値: `manage_user` - ユーザーがすべてのユーザーのモバイルデバイスを追加または削除できるようにします。 プロパティの動作: - 必須
status	integer	アクションを実行するためのアクセスが有効かどうか。 使用可能な値: 0 - 無効; 1 - (デフォルト) 有効。

role.create

説明

`object role.create(object/array roles)`

このメソッドを使用すると、新しいロールを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object/array)` 作成するロール。

このメソッドは、標準のロールプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
rules	array	ロール用に作成するロールルール。

戻り値

`(object)` `roleids` プロパティの下に作成されたロールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたロールの順序と一致します。

例

ロールの作成

タイプ「User」で 2 つの UI 要素へのアクセスを拒否したロールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.create",
    "params": {
        "name": "Operator",
        "type": "1",
        "rules": {
            "ui": [
                {
```

```
                    "name": "monitoring.hosts",
                    "status": "0"
                },
                {
                    "name": "monitoring.maps",
                    "status": "0"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- ロールルール
- UI 要素
- モジュール
- アクション

ソース

CRole::create() in *ui/include/classes/api/services/CRole.php*.

role.delete

説明

`object role.delete(array roleids)`

このメソッドを使用すると、ロールを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメーター

(array) 削除するユーザーの役割の ID。

戻り値

(object) roleids プロパティの下にある削除されたユーザーの役割の ID を含むオブジェクトを返します。

例

複数のユーザーの役割の削除

2 つのユーザーの役割を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
```

```
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CRole::delete() in *ui/include/classes/api/services/CRole.php*.

role.get

説明

`integer/array role.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従ってロールを取得できます。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータ。

このメソッドは、以下のパラメータをサポートしています。

パラメータ	型	説明
roleids	ID/array	指定した ID を持つロールのみを返します。
current	boolean	`true` に設定すると、現在のユーザーのロールのみを返します。
selectRules	query	ロールルールを含む`rules` プロパティを返します。
selectUsers	query	ロールが割り当てられているユーザーを含む`users` プロパティを返します。 ユーザータイプに基づく制限については、`user.get` を参照してください。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `roleid`、`name`。
countOutput	boolean	これらのパラメータについては、リファレンス解説で説明しています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ロールデータの取得

”Super admin role” ロールデータとそのアクセスルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.problems",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.hosts",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.latest_data",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.maps",
                        "status": "1"
                    },
                    {
                        "name": "services.services",
                        "status": "1"
                    },
                    {
                        "name": "services.sla_report",
                        "status": "1"
                    },
                    {
```

```
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
        },
        {
            "name": "configuration.service_actions",
            "status": "1"
```

```
            },
            {
                "name": "configuration.discovery_actions",
                "status": "1"
            },
            {
                "name": "configuration.autoregistration_actions",
                "status": "1"
            },
            {
                "name": "configuration.internal_actions",
                "status": "1"
            },
            {
                "name": "reports.system_info",
                "status": "1"
            },
            {
                "name": "reports.audit",
                "status": "1"
            },
            {
                "name": "reports.action_log",
                "status": "1"
            },
            {
                "name": "configuration.event_correlation",
                "status": "1"
            },
            {
                "name": "administration.media_types",
                "status": "1"
            },
            {
                "name": "administration.scripts",
                "status": "1"
            },
            {
                "name": "administration.user_groups",
                "status": "1"
            },
            {
                "name": "administration.user_roles",
                "status": "1"
            },
            {
                "name": "administration.users",
                "status": "1"
            },
            {
                "name": "administration.api_tokens",
                "status": "1"
            },
            {
                "name": "administration.authentication",
                "status": "1"
            },
            {
                "name": "administration.general",
                "status": "1"
            },
            {
```

```
                    "name": "administration.audit_log",
                    "status": "1"
                },
                {
                    "name": "administration.housekeeping",
                    "status": "1"
                },
                {
                    "name": "administration.proxy_groups",
                    "status": "1"
                },
                {
                    "name": "administration.proxies",
                    "status": "1"
                },
                {
                    "name": "administration.macros",
                    "status": "1"
                },
                {
                    "name": "administration.queue",
                    "status": "1"
                }
            ],
            "ui.default_access": "1",
            "services.read.mode": "1",
            "services.read.list": [],
            "services.read.tag": {
                "tag": "",
                "value": ""
            },
            "services.write.mode": "1",
            "services.write.list": [],
            "services.write.tag": {
                "tag": "",
                "value": ""
            },
            "modules": [
                {
                    "moduleid": 1,
                    "status": "1"
                },
                {
                    "moduleid": 2,
                    "status": "1"
                },
                {
                    "moduleid": 3,
                    "status": "1"
                },
                {
                    "moduleid": 4,
                    "status": "1"
                },
                {
                    "moduleid": 5,
                    "status": "1"
                },
                {
                    "moduleid": 6,
                    "status": "1"
                },
```

```
                {
                    "moduleid": 7,
                    "status": "1"
                },
                {
                    "moduleid": 8,
                    "status": "1"
                },
                {
                    "moduleid": 9,
                    "status": "1"
                },
                {
                    "moduleid": 10,
                    "status": "1"
                },
                {
                    "moduleid": 11,
                    "status": "1"
                },
                {
                    "moduleid": 12,
                    "status": "1"
                },
                {
                    "moduleid": 13,
                    "status": "1"
                },
                {
                    "moduleid": 14,
                    "status": "1"
                },
                {
                    "moduleid": 15,
                    "status": "1"
                },
                {
                    "moduleid": 16,
                    "status": "1"
                },
                {
                    "moduleid": 17,
                    "status": "1"
                },
                {
                    "moduleid": 18,
                    "status": "1"
                },
                {
                    "moduleid": 19,
                    "status": "1"
                },
                {
                    "moduleid": 20,
                    "status": "1"
                },
                {
                    "moduleid": 21,
                    "status": "1"
                },
                {
                    "moduleid": 22,
```

```
                "status": "1"
            },
            {
                "moduleid": 23,
                "status": "1"
            },
            {
                "moduleid": 24,
                "status": "1"
            },
            {
                "moduleid": 25,
                "status": "1"
            },
            {
                "moduleid": 26,
                "status": "1"
            },
            {
                "moduleid": 27,
                "status": "1"
            },
            {
                "moduleid": 28,
                "status": "1"
            },
            {
                "moduleid": 29,
                "status": "1"
            },
            {
                "moduleid": 30,
                "status": "1"
            },
            {
                "moduleid": 31,
                "status": "1"
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
            {
                "name": "edit_maps",
                "status": "1"
            },
            {
                "name": "acknowledge_problems",
```

```
                        "status": "1"
                    },
                    {
                        "name": "suppress_problems",
                        "status": "1"
                    },
                    {
                        "name": "close_problems",
                        "status": "1"
                    },
                    {
                        "name": "change_severity",
                        "status": "1"
                    },
                    {
                        "name": "add_problem_comments",
                        "status": "1"
                    },
                    {
                        "name": "execute_scripts",
                        "status": "1"
                    },
                    {
                        "name": "manage_api_tokens",
                        "status": "1"
                    },
                    {
                        "name": "edit_maintenance",
                        "status": "1"
                    },
                    {
                        "name": "manage_scheduled_reports",
                        "status": "1"
                    },
                    {
                        "name": "manage_sla",
                        "status": "1"
                    },
                    {
                        "name": "invoke_execute_now",
                        "status": "1"
                    },
                    {
                        "name": "change_problem_ranking",
                        "status": "1"
                    },
                    {
                        "name": "edit_own_media",
                        "status": "1"
                    },
                    {
                        "name": "edit_user_media",
                        "status": "1"
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```


関連項目

- ロールルール
- ユーザー

ソース

CRole::get() in *ui/include/classes/api/services/CRole.php*.

role.update

説明

`object role.update(object/array roles)`

このメソッドを使用すると、既存のロールを更新できます。

Note:
このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新するロールのプロパティ。

各ロールでは `roleid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準のロールプロパティ に加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
rules	array	ロールに対して更新するアクセスrules。

戻り値

(`object`) `roleids` プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

スクリプト実行機能の無効化

ID "5" のロールを更新し、スクリプト実行機能を無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
```

```
            "5"
        ]
    },
    "id": 1
}
```

API へのアクセス制限

ID が”5” のロールを更新し、すべての”create”、”update”、”delete” メソッドの呼び出しを拒否します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CRole::update() in *ui/include/classes/api/services/CRole.php*.

値マップ

このクラスは値マップを操作するために設計されています。

オブジェクト参照:

- 値マップ
    - 値マッピング

利用可能なメソッド:

- valuemap.create - 新しい値マップを作成
- valuemap.delete - 値マップを削除
- valuemap.get - 値マップを取得
- valuemap.update - 値マップを更新

Value map オブジェクト

次のオブジェクトは、valuemapAPI に直接関連しています。

値マップ

値マップオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
valuemapid	ID	値マップの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
hostid	ID	値マップが属するホストまたはテンプレートの ID。 プロパティの動作: - 固定 - 作成操作で 必須
name	string	値マップの名前。 プロパティの動作: - 作成操作で 必須
mappings	array	現在の値マップの値マッピング。マッピングオブジェクトの詳細は以下を参照してください。 プロパティの動作: - 作成操作で 必須
uuid	string	既存の値マップにインポートした値マップを関連付けるために使用される汎用一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - 値マップがテンプレートに属する場合に サポート

値のマッピング

値のマッピングオブジェクトは、値マップの値のマッピングを定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
type	integer	マッピング一致タイプ。 使用可能な値: 0 - *(デフォルト)* 値が等しい場合にマッピングが適用されます。 1 - 値が以上の場合にマッピングが適用されます[1]。 2 - 値が以下の場合にマッピングが適用されます[1]。 3 - 値が範囲内にある場合にマッピングが適用されます（範囲には端点を含みます。カンマ文字で区切って複数の範囲を定義できます）[1]。 4 - 値が正規表現に一致する場合にマッピングが適用されます[2]。 5 - 一致が見つからない場合、マッピングは適用されず、デフォルト値が使用されます。 `type` が”0”、”1”、”2”、”3”、”4” に設定されている場合、`value` は空にできません。
value	string	`type` が”5” に設定されている場合、`value` は空でなければなりません。 元の値。 プロパティの動作: - `type` が”1”、”2”、”3”、”4” に設定されている場合は 必須 - `type` が”5” に設定されている場合は サポート
newvalue	string	元の値がマッピングされる先の値。 プロパティの動作: - 必須

[1] 値の型が”numeric unsigned”、”numeric float” のアイテムでのみサポートされます。
 [2] 値の型が”character” のアイテムでのみサポートされます。

valuemap.create

説明

`object valuemap.create(object/array valuemaps)`

このメソッドを使用すると、新しい値マップを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成する値のマッピング。

このメソッドは、標準の値のマッピングプロパティを持つ値のマッピングを受け付けます。

戻り値

(object) valuemapids プロパティの下に作成されたバリューマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたバリューマップの順序と一致します。

例

バリューマップの作成

2 つのマッピングを使用して 1 つのバリューマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
        "hostid": "50009",
        "name": "Service state",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Up"
            },
            {
                "type": "5",
                "newvalue": "Down"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::create() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

説明

`object valuemap.delete(array valuemapids)`

このメソッドを使用すると、値マップを削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除するバリューマップの ID。

戻り値

(object) valuemapids プロパティの下にある削除されたバリューマップの ID を含むオブジェクトを返します。

例

複数のバリューマップの削除

2 つのバリューマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::delete() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

説明

`integer/array valuemap.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って値マップを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
valuemapids	ID/array	指定した ID を持つ値マップのみを返します。
selectMappings	query	現在の値マップの値マッピングを**mappings** プロパティで返します。 count をサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: valuemapid, name。
countOutput	boolean	これらのパラメータについては、リファレンスコメント を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

値のマップの取得

すべての設定済みの値のマップを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status"
        },
        {
            "valuemapid": "5",
            "name": "APC Battery Status"
        },
        {
            "valuemapid": "7",
            "name": "Dell Open Manage System Status"
```

```
        }
    ],
    "id": 1
}
```

マッピング付きで 1 つの値のマップを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status",
            "mappings": [
                {
                    "type": "0",
                    "value": "1",
                    "newvalue": "unknown"
                },
                {
                    "type": "0",
                    "value": "2",
                    "newvalue": "notInstalled"
                },
                {
                    "type": "0",
                    "value": "3",
                    "newvalue": "ok"
                },
                {
                    "type": "0",
                    "value": "4",
                    "newvalue": "failed"
                },
                {
                    "type": "0",
                    "value": "5",
                    "newvalue": "highTemperature"
                },
                {
                    "type": "0",
                    "value": "6",
                    "newvalue": "replaceImmediately"
                },
                {
                    "type": "0",
                    "value": "7",
                    "newvalue": "lowCapacity"
                }
            ]
```

```
        }
    ],
    "id": 1
}
```

ソース

CValueMap::get() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

説明

`object valuemap.update(object/array valuemaps)`

このメソッドを使用すると、既存の値マップを更新できます。

Note:
このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新する値マップのプロパティ。

各値マップでは `valuemapid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外のプロパティは変更されません。

戻り値

(object) `valuemapids` プロパティの下にある更新されたバリューマップの ID を含むオブジェクトを返します。

例

値のマップ名の変更

値のマップ名を「Device status」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

1 つの値のマッピングを変更する。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
```

```
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::update() in *ui/include/classes/api/services/CValueMap.php*.

履歴

このクラスは履歴データを扱うために設計されています。

オブジェクト参照:

- 浮動小数点履歴
- 整数履歴
- 文字列履歴
- テキスト履歴
- ログ履歴

利用可能なメソッド:

- history.clear - 履歴データをクリア
- history.get - 履歴データを取得
- history.push - 履歴データを Zabbix サーバーに送信

History オブジェクト

次のオブジェクトは、`history`API に直接関連しています。

> **Note:**
> History オブジェクトはアイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API から変更することはできません。

浮動小数点数の履歴

浮動小数点数の履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	float	受信した値。

整数ヒストリ

整数ヒストリオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	integer	受信した値。

文字列履歴

文字列履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	string	受信した値。

テキスト履歴

テキスト履歴オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
id	ID	履歴エントリの ID。
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	text	受信した値。

ログ履歴

ログ履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
id	ID	履歴エントリの ID。
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
logeventid	integer	Windows イベントログエントリの ID。
ns	integer	値を受信した時点のナノ秒。
severity	integer	Windows イベントログエントリのレベル。
source	string	Windows イベントログエントリのソース。
timestamp	timestamp	Windows イベントログエントリの時刻。
value	text	受信した値。

バイナリ履歴

バイナリ履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	text	受信した値。

JSON 履歴

JSON 履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	text	受信した値。

history.clear

説明

`object history.clear(array itemids)`

このメソッドを使用すると、アイテムの履歴を削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) ヒストリを消去するアイテムの ID。

戻り値

(object) itemids プロパティの下にあるヒストリが消去されたアイテムの ID を含むオブジェクトを返します。

例

ヒストリの消去

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

ソース

CHistory::clear() in *ui/include/classes/api/services/CHistory.php*.

history.get

説明

`integer/array history.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って履歴データを取得できます。

Attention:
このメソッドは、housekeeper によってまだ削除されていない場合、削除済みのエンティティの履歴データを返すことがあります。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

`(object)` 目的の出力を定義するパラメーター。

このメソッドは次のパラメーターをサポートします。

Parameter	Type	Description
history	integer	返す履歴オブジェクトの種類。 指定可能な値: 0 - 数値浮動小数点; 1 - 文字; 2 - ログ; 3 - *(default)* 数値符号なし; 4 - テキスト; 5 - バイナリ; 6 - JSON.
hostids	ID/array	指定したホストの履歴のみを返します。
itemids	ID/array	指定したアイテムの履歴のみを返します。
maxValueSize	integer	`value` プロパティで返す最大バイト数。 指定可能な範囲: 1-134217728 (1 byte から 128MiB)。 `null` に設定すると、制限は適用されません。 デフォルト: 65536 (64KiB)。 Parameter behavior: - `history` が”binary” または”JSON” に設定されている場合は *supported*
time_from	timestamp	指定した時刻以降に受信した値のみを返します。
time_till	timestamp	指定した時刻以前に受信した値のみを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `itemid`, `clock`, `ns`. 履歴データがClickHouse に保存されている場合、`ns` によるソートはサポートされません。

Parameter	Type	Description
search	object	指定したパターンに一致する結果を返します（大文字小文字を区別しません）。 キーがプロパティ名、値が検索文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。 history が 6 (JSON) に設定されている場合、value による検索はサポートされません。
countOutput	boolean	これらのパラメーターはreference commentary に説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

アイテム履歴データの取得

数値 (浮動小数点) アイテムから受信した最新の 10 個の値を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23296",
            "clock": "1351090996",
            "value": "0.085",
            "ns": "563157632"
        },
        {
            "itemid": "23296",
            "clock": "1351090936",
            "value": "0.16",
            "ns": "549216402"
```

```
        },
        {
            "itemid": "23296",
            "clock": "1351090876",
            "value": "0.18",
            "ns": "537418114"
        },
        {
            "itemid": "23296",
            "clock": "1351090816",
            "value": "0.21",
            "ns": "522659528"
        },
        {
            "itemid": "23296",
            "clock": "1351090756",
            "value": "0.215",
            "ns": "507809457"
        },
        {
            "itemid": "23296",
            "clock": "1351090696",
            "value": "0.255",
            "ns": "495509699"
        },
        {
            "itemid": "23296",
            "clock": "1351090636",
            "value": "0.36",
            "ns": "477708209"
        },
        {
            "itemid": "23296",
            "clock": "1351090576",
            "value": "0.375",
            "ns": "463251343"
        },
        {
            "itemid": "23296",
            "clock": "1351090516",
            "value": "0.315",
            "ns": "447947017"
        },
        {
            "itemid": "23296",
            "clock": "1351090456",
            "value": "0.275",
            "ns": "435307141"
        }
    ],
    "id": 1
}
```


ソース

CHistory::get() in *ui/include/classes/api/services/CHistory.php*.

history.push

説明

`object history.push(object/array itemHistoryData)`

このメソッドは、アイテムヒストリデータを Zabbix サーバーに送信できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 送信するアイテムの履歴データ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
itemid	ID	関連するアイテムの ID。
host	string	Parameter behavior: - host と key が設定されていない場合は *required* ホストの技術名。
key	string	Parameter behavior: - itemid が設定されていない場合は *required* アイテムキー。
value	mixed	Parameter behavior: - itemid が設定されていない場合は *required* アイテムの値。
clock	timestamp	Parameter behavior: - *required* 値を受信した時刻。
ns	integer	値を受信した時点のナノ秒。

戻り値

(object) データ送信操作の結果を返します。

例

アイテムヒストリデータの送信

Zabbix サーバーへ、アイテム"10600"、"10601"、"999999" のヒストリデータを送信します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.push",
    "params": [
        {
            "itemid": 10600,
            "value": 0.5,
            "clock": 1690891294,
            "ns": 45440940
        },
        {
            "itemid": 10600,
            "value": 0.6,
            "clock": 1690891295,
            "ns": 312431
        },
        {
            "itemid": 10601,
            "value": "[Tue Aug 01 15:01:35 2023] [error] [client 1.2.3.4] File does not exist: /var/www/ht
        },
        {
            "itemid": 999999,
            "value": 123
```

```
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

参照

- Trapper items
- HTTP agent items
- Host
- Item

ソース

CHistory::push() in *ui/include/classes/api/services/CHistory.php*.

検出されたサービス

このクラスは、検出されたサービスを操作するために設計されています。

オブジェクト参照:

- 検出されたサービス

利用可能なメソッド:

- dservice.get - 検出されたサービスを取得します

Discovered service オブジェクト

次のオブジェクトは、dserviceAPI に直接関連しています。

検出されたサービス

> **Note:**
> 検出されたサービスは Zabbix サーバーによって作成され、API 経由で変更することはできません。

検出されたサービスオブジェクトには、ホスト上のネットワークディスカバリルールによって検出されたサービスに関する情報が含まれます。次のプロパティがあります。

プロパティ	型	説明
dserviceid	ID	検出されたサービスの ID。
dcheckid	ID	サービスの検出に使用されたディスカバリチェックの ID。
dhostid	ID	サービスを実行している検出されたホストの ID。
dns	string	サービスを実行しているホストの DNS。
ip	string	サービスを実行しているホストの IP アドレス。
lastdown	timestamp	検出されたサービスが最後に停止した時刻。
lastup	timestamp	検出されたサービスが最後に稼働した時刻。
port	integer	サービスポート番号。
status	integer	サービスのステータス。 使用可能な値: 0 - サービス稼働; 1 - サービス停止。
value	string	Zabbix エージェント、SNMPv1、SNMPv2、または SNMPv3 のディスカバリチェックの実行時にサービスから返された値。

dservice.get

説明

`integer/array dservice.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って検出されたサービスを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dserviceids	ID/array	指定した ID を持つディスカバリされたサービスのみを返します。
dhostids	ID/array	指定したディスカバリされたホストに属するディスカバリされたサービスのみを返します。
dcheckids	ID/array	指定したディスカバリチェックによって検出されたディスカバリされたサービスのみを返します。
druleids	ID/array	指定したディスカバリルールによって検出されたディスカバリされたサービスのみを返します。
selectDRules	query	サービスを検出したディスカバリルールの配列を含む`drules` プロパティを返します。
selectDHosts	query	サービスが属するディスカバリされたホストの配列を含む`dhosts` プロパティを返します。
selectHosts	query	サービスと同じ IP アドレスおよびプロキシを持つホストを含む`hosts` プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトによって返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `hostid` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dserviceid`, `dhostid`, `ip`。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	

パラメータ	型	説明
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストで検出されたサービスの取得

検出されたホスト"11" で検出されたすべてサービスを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dserviceid": "12",
            "dhostid": "11",
            "value": "",
            "port": "80",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650607",
            "dcheckid": "5",
            "ip": "192.168.1.134",
            "dns": "john.local"
        },
        {
            "dserviceid": "13",
            "dhostid": "11",
            "value": "",
            "port": "21",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650610",
            "dcheckid": "6",
            "ip": "192.168.1.134",
            "dns": "john.local"
        }
    ],
```

```
    "id": 1
}
```

関連項目

- 検出されたホスト
- ディスカバリチェック
- ホスト

ソース

CDService::get() in *ui/include/classes/api/services/CDService.php*.

検出されたホスト

このクラスは、検出されたホストを操作するために設計されています。

オブジェクト参照:

- 検出されたホスト

利用可能なメソッド:

- dhost.get - 検出されたホストを取得します

Discovered host オブジェクト

次のオブジェクトは、dhostAPI に直接関連しています。

検出されたホスト

> **Note:**
> 検出されたホストは Zabbix サーバーによって作成され、API 経由で変更することはできません。

検出されたホストオブジェクトには、ネットワークディスカバリルールによって検出されたホストに関する情報が含まれます。次のプロパティがあります。

プロパティ	Type	説明
dhostid	ID	検出されたホストの ID。
druleid	ID	ホストを検出したディスカバリルールの ID。
lastdown	timestamp	検出されたホストが最後にダウンした時刻。
lastup	timestamp	検出されたホストが最後にアップした時刻。
status	integer	検出されたホストがアップしているかダウンしているか。ホストは、少なくとも 1 つのアクティブな検出済みサービスがある場合にアップとみなされます。 使用可能な値: 0 - ホストがアップ; 1 - ホストがダウン。

dhost.get

説明

`integer/array dhost.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従って検出されたホストを取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
dhostids	ID/array	指定した ID を持つ検出済みホストのみを返します。
druleids	ID/array	指定したディスカバリルールによって作成された検出済みホストのみを返します。
dserviceids	ID/array	指定したサービスを実行している検出済みホストのみを返します。
selectDRules	query	ホストを検出したディスカバリルールの配列を含む`drules` プロパティを返します。
selectDServices	query	ホスト上で実行されている検出済みサービスを含む`dservices` プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectDServices` - 結果は `dserviceid` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dhostid`, `druleid`。
countOutput	boolean	これらのパラメータについては、reference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ディスカバリルールによって検出されたホストの取得

ディスカバリルール"4" によって検出されたすべてのホストと実行中の発見されたサービスを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "dservices": [
                {
                    "dserviceid": "1",
                    "dhostid": "1",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697227",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.1",
                    "dns": "station.company.lan"
                }
            ],
            "dhostid": "1",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697227",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "2",
                    "dhostid": "2",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.4",
                    "dns": "john.company.lan"
                }
            ],
            "dhostid": "2",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "3",
                    "dhostid": "3",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.26",
                    "dns": "printer.company.lan"
                }
            ],
```

```
            "dhostid": "3",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "4",
                    "dhostid": "4",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.7",
                    "dns": "mail.company.lan"
                }
            ],
            "dhostid": "4",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        }
    ],
    "id": 1
}
```


関連項目

- 検出されたサービス
- ディスカバリルール

ソース

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

正規表現

このクラスはグローバル正規表現を扱うために設計されています。

オブジェクト参照:

- 正規表現
- 式

利用可能なメソッド:

- regexp.create - 新しい正規表現を作成
- regexp.delete - 正規表現を削除
- regexp.get - 正規表現を取得
- regexp.update - 正規表現を更新

Regular expression オブジェクト

次のオブジェクトは、regexpAPI に直接関連しています。

正規表現

グローバル正規表現オブジェクトには、次のプロパティがあります。

Property	Type	Description
regexpid	ID	正規表現の ID です。 Property behavior: - *read-only* - 更新操作では *required* です。
name	string	正規表現の名前です。 Property behavior: - 作成操作では *required* です。
description	text	正規表現の説明です。
test_string	string	テスト文字列。

式

expressions オブジェクトには、次のプロパティがあります。

Property	Type	Description
expression	string	正規表現。 Property behavior: - *required*
expression_type	integer	正規表現の種類。 Possible values: 0 - 文字列を含む; 1 - リスト内の任意の部分文字列を含む; 2 - 文字列を含まない; 3 - 正規表現に一致する; 4 - 正規表現に一致しない。 Property behavior: - *required*
exp_delimiter	string	式の区切り文字。 Default value: `","`. Possible values: `","` or `"."`, or `"/"`. Property behavior: - *supported* if `expression_type` is set to *Contains any substring from list*.
case_sensitive	integer	大文字と小文字の区別。 Default value: 0. Possible values: 0 - 大文字と小文字を区別しない; 1 - 大文字と小文字を区別する.

regexp.create

説明

`object regexp.create(object/array regularExpressions)`

このメソッドを使用すると、新しいグローバル正規表現を作成できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 作成する正規表現。

標準プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
expressions	array	Expressions オプション。 パラメータの動作: - 必須

戻り値

(object) regexpids プロパティの下で作成された正規表現の ID を含むオブジェクトを返します。

例

新しいグローバル正規表現の作成

Request:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "Global regular expression used to discover and identify memory resources on the mo
        "test_string": "/boot",
        "expressions": [
            {
                "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$"
                "expression_type": "4",
                "case_sensitive": "1"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "9"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

説明

`object regexp.delete(array regexpids)`

このメソッドを使用すると、グローバル正規表現を削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除する正規表現の ID。

戻り値

(object) regexpids プロパティの下にある削除された正規表現の ID を含むオブジェクトを返します。

例

複数のグローバル正規表現の削除

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::delete() in *ui/include/classes/api/services/CRegexp.php*.

regexp.get

説明

`integer/array regexp.get(object parameters)`

このメソッドを使用すると、指定したパラメーターに従ってグローバル正規表現を取得できます。

> **Note:**
> このメソッドは *Super Admin* のみに利用可能です。メソッドの呼び出し権限はユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
regexpids	ID/array	指定した ID を持つ正規表現のみを返します。
selectExpressions	query	`expressions` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `regexpid`, `name`。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

パラメータ	型	説明
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

グローバル正規表現の取得。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid","name","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "regexpid": "1",
            "name": "File systems for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Network interfaces for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
```

```
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
    },
    "id": 1
}
```

ソース

CRegexp::get() in *ui/include/classes/api/services/CRegexp.php*.

regexp.update

説明

`object regexp.update(object/array regularExpressions)`

このメソッドを使用すると、既存のグローバル正規表現を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object/array) 更新する正規表現のプロパティ。

各オブジェクトに対して regexpid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

標準プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
expressions	array	式 オプション。

戻り値

(object) regexpids プロパティの下にある更新された正規表現の ID を含むオブジェクトを返します。

例

ファイルシステム検出用のグローバル正規表現の更新。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
        "regexpid": "1",
        "name": " 検出用ファイルシステム",
        "description": " 監視対象ホスト上のファイルシステムを検出し識別するために使用されるグローバル正規表現",
        "test_string": "",
```

```
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::update() in *ui/include/classes/api/services/CRegexp.php*.

監査ログ

このクラスは監査ログを扱うために設計されています。

オブジェクト参照:

- 監査ログ

利用可能なメソッド:

- auditlog.get - 監査ログのレコードを取得します

Audit log オブジェクト

次のオブジェクトは、auditlogAPI に直接関連しています。

監査ログ

監査ログオブジェクトには、ユーザーの操作に関する情報が含まれます。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
auditid	ID	監査ログエントリの ID。CUID アルゴリズムを使用して生成されます。
userid	ID	監査ログエントリ作成者の userid。
username	string	監査ログエントリ作成者のユーザー名。
clock	timestamp	監査ログエントリ作成時のタイムスタンプ。
ip	string	監査ログエントリ作成者の IP アドレス。

Property	Type	Description
action	integer	監査ログエントリのアクション。 使用可能な値: 0 - 追加; 1 - 更新; 2 - 削除; 4 - ログアウト; 7 - 実行; 8 - ログイン; 9 - ログイン失敗; 10 - 履歴のクリア; 11 - 設定のリフレッシュ; 12 - プッシュ。

Property	Type	Description
resourcetype	integer	監査ログエントリのリソースタイプ。 使用可能な値: 0 - ユーザー; 3 - メディアタイプ; 4 - ホスト; 5 - アクション; 6 - グラフ; 11 - ユーザーグループ; 13 - トリガー; 14 - ホストグループ; 15 - アイテム; 16 - 画像; 17 - 値のマッピング; 18 - サービス; 19 - マップ; 22 - Web シナリオ; 23 - ディスカバリルール; 25 - スクリプト; 26 - プロキシ; 27 - メンテナンス; 28 - 正規表現; 29 - マクロ; 30 - テンプレート; 31 - トリガープロトタイプ; 32 - アイコンマッピング; 33 - ダッシュボード; 34 - イベント相関関係; 35 - グラフプロトタイプ; 36 - アイテムプロトタイプ; 37 - ホストプロトタイプ; 38 - 自動登録; 39 - モジュール; 40 - 設定; 41 - ハウスキーピング; 42 - 認証; 43 - テンプレートダッシュボード; 44 - ユーザーロール; 45 - API トークン; 46 - 定期レポート; 47 - 高可用性ノード; 48 - SLA; 49 - ユーザーディレクトリ; 50 - テンプレートグループ; 51 - コネクタ; 52 - LLD ルール; 53 - 履歴; 54 - 多要素認証; 55 - プロキシグループ; 56 - LLD ルールプロトタイプ。
resourceid	ID	監査ログエントリのリソース識別子。
resource_cuid	ID	CUID アルゴリズムを使用して生成された、監査ログエントリのリソース一意識別子。
resourcename	string	監査ログエントリの人が判読可能なリソース名。
recordsetid	ID	監査ログエントリのレコードセット ID。同じ操作中に作成された監査ログレコードは、同じレコードセット ID を持ちます。CUID アルゴリズムを使用して生成されます。

Property	Type	Description
details	text	監査ログエントリの詳細。詳細は JSON オブジェクトとして保存され、各プロパティ名は変更が発生したプロパティまたはネストされたオブジェクトへのパスを表し、各値にはそのプロパティまたはネストされたオブジェクトにおける変更に関するデータが（配列形式で）含まれます。 使用可能な値の形式: ["add"] - ネストされたオブジェクトが追加された; ["add", "<value>"] - 追加されたオブジェクトのプロパティが <value> に等しい; ["update"] - ネストされたオブジェクトが更新された; ["update", "<new value>", "<old value>"] - 更新されたオブジェクトのプロパティが <old value> から <new value> に変更された; ["delete"] - ネストされたオブジェクトが削除された。

auditlog.get

説明

`integer/array auditlog.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って監査ログレコードを取得できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
auditids	ID/array	指定した ID の監査ログのみを返します。
userids	ID/array	指定したユーザーによって作成された監査ログのみを返します。
time_from	timestamp	指定した時刻以降に作成された監査ログエントリのみを返します。
time_till	timestamp	指定した時刻以前に作成された監査ログエントリのみを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `auditid`, `userid`, `clock`。 これらのパラメータについては、リファレンスコメントを参照してください。
countOutput	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用された場合は、取得したオブジェクトの数

例

監査ログの取得

最新の監査ログレコード 2 件を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "auditid": "cksstgfam0001yhdcc41y20q2",
            "userid": "1",
            "username": "Admin",
            "clock": "1629975715",
            "ip": "127.0.0.1",
            "action": "1",
            "resourcetype": "0",
            "resourceid": "0",
            "resource_cuid": "0",
            "resourcename": "Jim",
            "recordsetid": "cksstgfal0000yhdcso67ondl",
            "details": "{\"user.name\":[\"update\",\"Jim\",\"\"],\"user.medias[37]\":[\"add\"],\"user.medi
        },
        {
            "auditid": "ckssofl0p0001yhdcqxclsg8r",
            "userid": "1",
            "username": "Admin",
            "clock": "1629967278",
            "ip": "127.0.0.1",
            "action": "0",
            "resourcetype": "0",
            "resourceid": "20",
            "resource_cuid": "0",
            "resourcename": "John",
            "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
            "details": "{\"user.username\":[\"add\",\"John\"], \"user.userid:\":[\"add\",\"20\"],\"user.us
        }
    ],
    "id": 1
}
```

参照

- Audit log object

ソース

CAuditLog::get() in *ui/include/classes/api/services/CAuditLog.php*.

相関関係

このクラスは相関関係を扱うために設計されています。

オブジェクト参照:

- 相関関係
- 相関関係の操作
- 相関関係フィルター
  - 相関関係フィルター条件

利用可能なメソッド:

- correlation.create - 新しい相関関係を作成
- correlation.delete - 相関関係を削除
- correlation.get - 相関関係を取得
- correlation.update - 相関関係を更新

Correlation オブジェクト

次のオブジェクトは、`correlation`API に直接関連しています。

相関関係

相関関係オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
correlationid	ID	相関関係の ID。 プロパティの動作: - 読み取り専用 - 更新操作で必須
name	string	相関関係の名前。 プロパティの動作: - 作成操作で必須
description	string	相関関係の説明。
status	integer	相関関係が有効か無効か。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。

相関関係の操作

相関関係の操作オブジェクトは、相関関係が実行されたときに実行される操作を定義します。以下のプロパティがあります。

Property	Type	Description
type	integer	操作のタイプ。 使用可能な値: 0 - 古いイベントをクローズ; 1 - 新しいイベントをクローズ。 プロパティの動作: - 必須

相関フィルター

相関フィルターオブジェクトは、設定された相関操作を実行するために満たす必要がある条件のセットを定義します。次のプロパティがあります。

Property	Type	Description
conditions	array	結果のフィルタリングに使用するフィルター条件のセット。条件は、式内での配置順に並べ替えられます。 Property behavior: - *required*

Property	Type	Description
evaltype	integer	フィルター条件の評価方法。 可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - *required*
eval_formula	string	フィルター条件の評価に使用される生成済みの式。式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれます。`eval_formula` の値は、カスタム式を持つフィルターでは `formula` の値と等しくなります。 Property behavior: - *read-only*
formula	string	カスタム式を持つフィルターの条件評価に使用する、ユーザー定義の式。式には、`formulaid` によって特定のフィルター条件を参照する ID を含める必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致していなければなりません。条件が未使用のまま残ったり、省略されたりしてはいけません。 Property behavior: - *required* if `evaltype` is set to "custom expression"

相関フィルター条件

相関フィルター条件オブジェクトは、相関操作を実行する前に確認する必要がある特定の条件を定義します。

プロパティ	Type	説明
type	integer	条件のタイプ。 使用可能な値: 0 - 古いイベントタグ; 1 - 新しいイベントタグ; 2 - 新しいイベントのホストグループ; 3 - イベントタグのペア; 4 - 古いイベントタグ値; 5 - 新しいイベントタグ値。 Property behavior: - *required*
tag	string	イベントタグ（古いまたは新しい）。 Property behavior: - `type` が「古いイベントタグ」、「新しいイベントタグ」、「古いイベントタグ値」、または「新しいイベントタグ値」に設定されている場合は *required*
groupid	ID	ホストグループの ID。 Property behavior: - `type` が「新しいイベントのホストグループ」に設定されている場合は *required*

プロパティ	Type	説明
oldtag	string	古いイベントタグ。 Property behavior: - type が「イベントタグのペア」に設定されている場合は *required*
newtag	string	古いイベントタグ。 Property behavior: - type が「イベントタグのペア」に設定されている場合は *required*
value	string	イベントタグ値（古いまたは新しい）。 Property behavior: - type が「古いイベントタグ値」または「新しいイベントタグ値」に設定されている場合は *required*
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。この ID は、フィルター条件を変更する際にユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。
operator	integer	条件演算子。 Property behavior: - type が「新しいイベントのホストグループ」、「古いイベントタグ値」、または「新しいイベントタグ値」に設定されている場合は *required*

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、correlation.get およびcorrelation.create メソッドのページにある例を参照してください。

以下の演算子と値が、各条件タイプでサポートされています。

条件	条件名	サポートされる演算子	想定される値
2	ホストグループ	=, <>	ホストグループ ID。
4	古いイベントタグ値	=, <>, like, not like	string
5	新しいイベントタグ値	=, <>, like, not like	string

correlation.create

説明

`object correlation.create(object/array correlations)`

このメソッドを使用すると、新しい相関を作成できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(`object/array`) 作成する相関関係。

このメソッドは、標準の相関関係プロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
operations	array	相関関係用に作成する相関関係の操作。 パラメータの動作: - 必須

パラメータ	型	説明
filter	object	相関関係の相関関係フィルターオブジェクト。 パラメータの動作: - 必須

戻り値

(object) correlationids プロパティの下に作成された相関関係の ID を含むオブジェクトを返します。返す ID の順序は、渡された相関関係の順序と一致します。

例

新しいイベントタグ相関の作成

1 つの条件と 1 つの演算による評価方法 AND/OR を使用して相関関係を作成します。デフォルトでは相関関係は有効になります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new event tag correlation",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "type": 1,
                    "tag": "ok"
                }
            ]
        },
        "operations": [
            {
                "type": 0
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

カスタムフィルター条件を使用する相関関係を作成します。式 ID、”A” または”B” は任意に選択されています。条件タイプは、演算子”<>” を使用した” ホストグループ” になります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new host group correlation",
        "description": "a custom description",
        "status": 0,
```

```
        "filter": {
            "evaltype": 3,
            "formula": "A or B",
            "conditions": [
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "A"
                },
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "B"
                }
            ]
        },
        "operations": [
            {
                "type": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- 相関フィルター
- 相関操作

ソース

CCorrelation::create() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

説明

`object correlation.delete(array correlationids)`

このメソッドを使用すると、相関を削除できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメーター

(array) 削除する相関関係の ID。

戻り値

(object) correlationids プロパティの下にある削除された相関関係の ID を含むオブジェクトを返します。

例

複数の相関関係の削除

2 つの相関関係を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CCorrelation::delete() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

説明

`integer/array correlation.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って相関関係を取得できます。

> **Note:**
> このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーのロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
correlationids	ID/array	指定した ID を持つ相関関係のみを返します。
selectFilter	query	相関条件を含む`filter` プロパティを返します。
selectOperations	query	相関操作を含む`operations` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `correlationid`, `name`, `status`。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

パラメータ	型	説明
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

相関関係の取得

設定されたすべての相関を、相関条件および操作とともに取得します。フィルターは"and/or" 評価タイプを使用するため、formula プロパティは空で、eval_formula が自動的に生成されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "correlationid": "1",
            "name": "Correlation 1",
            "description": "",
            "status": "0",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "type": "3",
                        "oldtag": "error",
                        "newtag": "ok",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
            },
            "operations": [
                {
                    "type": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- 相関フィルター
- 相関操作

ソース

CCorrelation::get() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

説明

`object correlation.update(object/array correlations)`

このメソッドを使用すると、既存の相関関係を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザー権限設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(`object/array`) 更新する相関関係のプロパティ。

各相関関係について `correlationid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準の相関関係プロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	現在のフィルターを置き換えるための相関関係フィルターオブジェクト。
operations	array	既存の操作を置き換える相関関係の操作。

戻り値

(`object`) `correlationids`'プロパティの下で更新された相関関係 ID を含むオブジェクトを返します。

例

相関関係の無効化

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

評価方法を維持したままで条件の置き換え

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- 相関フィルター
- 相関操作

ソース

CCorrelation::update() in *ui/include/classes/api/services/CCorrelation.php*.

障害

このクラスは障害を扱うために設計されています。

オブジェクト参照:

- Problem
  - メディアタイプ URL
- 障害タグ

利用可能なメソッド:

- problem.get - 障害を取得します

Problem オブジェクト

次のオブジェクトは、`problem` API に直接関連しています。

障害

> **Note:**
> 障害は Zabbix サーバーによって作成され、API 経由では変更できません。

障害オブジェクトには以下のプロパティがあります。

Property	Type	Description
eventid	ID	障害イベントの ID。
source	integer	障害イベントの種類。 使用可能な値: 0 - トリガーによって作成されたイベント; 3 - 内部イベント; 4 - サービスステータス更新時に作成されたイベント。
object	integer	障害イベントに関連するオブジェクトの種類。 `source` が「トリガーによって作成されたイベント」に設定されている場合の使用可能な値: 0 - トリガー。 `source` が「内部イベント」に設定されている場合の使用可能な値: 0 - トリガー; 4 - アイテム; 5 - LLD ルール。 `source` が「サービスステータス更新時に作成されたイベント」に設定されている場合の使用可能な値: 6 - サービス。
objectid	ID	関連オブジェクトの ID。
clock	timestamp	障害イベントが作成された時刻。
ns	integer	障害イベントが作成された時点のナノ秒。
r_eventid	ID	復旧イベントの ID。
r_clock	timestamp	復旧イベントが作成された時刻。
r_ns	integer	復旧イベントが作成された時点のナノ秒。
cause_eventid	ID	原因イベントの ID。
correlationid	ID	このイベントがグローバル相関ルールによって復旧された場合の相関ルールの ID。
userid	ID	障害をクローズしたユーザーの ID（障害が手動でクローズされた場合）。
name	string	解決済みの障害名。
acknowledged	integer	障害の確認状態。 使用可能な値: 0 - 未確認; 1 - 確認済み。
severity	integer	障害の現在の深刻度。 使用可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 致命的。
suppressed	integer	障害が抑制されているかどうか。 使用可能な値: 0 - 障害は通常状態; 1 - 障害は抑制されている。
opdata	string	展開済みマクロを含む運用データ。
urls	array	アクティブなメディアタイプ URL。

メディアタイプ URL

メディアタイプ URL オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	メディアタイプで定義された URL 名。

プロパティ	Type	説明
url	string	メディアタイプで定義された URL 値。

結果には、有効なイベントメニューエントリを持つアクティブなメディアタイプのエントリのみが含まれます。
プロパティで使用されているマクロは展開されますが、いずれかのプロパティに未展開のマクロが含まれている場合、両方のプロパティが結果から除外されます。
サポートされているマクロについては、サポートされているマクロを参照してください。

障害タグ

障害タグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	障害タグ名。
value	string	障害タグの値。

problem.get

説明

`integer/array problem.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って障害を取得できます。

このメソッドは、未解決の障害を取得するためのものです。必要に応じて、最近解決された障害も追加で取得できます。「最近」とみなす期間は、管理 > 一般 で定義されます。その期間より前に解決された障害は、障害テーブルには保持されません。さらに過去に解決された障害を取得するには、`event.get` メソッドを使用してください。

Attention:
このメソッドは、削除されたエンティティに関する障害が、housekeeper によってまだ削除されていない場合、それらの障害を返すことがあります。

Note:
このメソッドは、あらゆる種類のユーザーが利用できます。メソッドの呼び出し権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータ。

このメソッドは次のパラメータをサポートします。

Parameter	Type	Description
eventids	ID/array	指定した ID を持つ障害のみを返します。
groupids	ID/array	指定したホストグループに属するオブジェクトによって作成された障害のみを返します。
hostids	ID/array	指定したホストに属するオブジェクトによって作成された障害のみを返します。
objectids	ID/array	指定したオブジェクトによって作成された障害のみを返します。
source	integer	指定したタイプの障害のみを返します。 サポートされるイベントタイプの一覧については、problem event object page を参照してください。 デフォルト: 0 - トリガーによって作成された障害。
object	integer	指定したタイプのオブジェクトによって作成された障害のみを返します。 サポートされるオブジェクトタイプの一覧については、problem event object page を参照してください。 デフォルト: 0 - トリガー。

Parameter	Type	Description
acknowledged	boolean	`true` に設定すると、確認済みの障害のみを返します。
action	integer	指定したevent update actions が実行された障害のみを返します。複数のアクションを指定する場合は、許容される任意のビットマップ値の合計をビットマスクとして使用します（たとえば、確認とイベントの抑制には 34）。
action_userids	ID/array	障害イベント更新アクションを実行したユーザーの、指定した ID を持つ障害のみを返します。
suppressed	boolean	`true` に設定すると、抑制中の障害のみを返します。
symptom	boolean	`true` に設定すると、症状の障害イベントのみを返します。
severities	integer/array	指定したイベント深刻度を持つ障害のみを返します。object が trigger の場合にのみ適用されます。
evaltype	integer	タグのevaluation method。 指定可能な値: 0 - *(default)* And/Or; 2 - Or.
tags	array	指定したタグを持つ障害のみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべての障害を返します。 指定可能なoperator の値: 0 - *(default)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
recent	boolean	`true` に設定すると、`ok_period` 内に解決されたものを含む最近の障害と、未解決の障害を返します。 `false` に設定すると、未解決の障害のみを返します。
eventid_from	string	指定した ID 以上の ID を持つ障害のみを返します。
eventid_till	string	指定した ID 以下の ID を持つ障害のみを返します。
time_from	timestamp	指定した時刻以降に作成された障害のみを返します。
time_till	timestamp	指定した時刻以前に作成された障害のみを返します。
selectAcknowledges	query	障害の更新情報を含む `acknowledges` プロパティを返します。障害の更新は新しい順に並べ替えられます。 障害更新オブジェクトには次のプロパティがあります: `acknowledgeid` - (ID) 更新の ID; `userid` - (ID) イベントを更新したユーザーの ID; `eventid` - (ID) 更新されたイベントの ID; `clock` - (timestamp) イベントが更新された時刻; `message` - (string) メッセージのテキスト; `action` - (integer) 更新アクションの種類（`event.acknowledge` を参照）; `old_severity` - (integer) この更新アクション前のイベント深刻度; `new_severity` - (integer) この更新アクション後のイベント深刻度; `suppress_until` - (timestamp) イベントが抑制される期限; `taskid` - (ID) 現在のイベントがランク変更中の場合のタスクの ID; `maintenanceid` - (ID) メンテナンスの ID。 `count` をサポートします。
selectTags	query	障害タグを含むtags プロパティを返します。出力形式: `[{"tag": "<tag>", "value": "<value>"}, ...]`。
selectSuppressionData	query	アクティブなメンテナンスと手動による抑制の一覧を含む `suppression_data` プロパティを返します: `maintenanceid` - (ID) メンテナンスの ID; `userid` - (ID) 障害を抑制したユーザーの ID; `suppress_until` - (integer) 障害が抑制されるまでの時間。

Parameter	Type	Description
filter	object	指定したフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が照合対象の単一値または値の配列であるオブジェクトを受け付けます。 text data type のプロパティはサポートされません。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: eventid.
countOutput	boolean	これらのパラメータはreference commentary に説明があります。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

返り値

(integer/array) 次のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

トリガーの障害イベントを取得する

トリガー"15112" から最近のイベントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectTags": "extend",
        "selectSuppressionData": "extend",
        "objectids": "15112",
        "recent": true,
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
```

```
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "1",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0",
                    "maintenanceid": "0"
                }
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


指定ユーザーによって確認済みの障害の取得

ID=10 のユーザーによって確認済みの障害の取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```


関連項目

- アラート
- アイテム
- ホスト
- LLD ルール
- トリガー

ソース

CEvent::get() in *ui/include/classes/api/services/CProblem.php*.

高可用性ノード

このクラスは、高可用性クラスターの一部であるサーバーノード、またはスタンドアロンサーバーインスタンスで動作するように設計されています。

オブジェクト参照:

- 高可用性ノード

利用可能なメソッド:

- hanode.get - ノードを取得

High availability node オブジェクト

次のオブジェクトは、Zabbix サーバーの HA (高可用性) クラスタの運用に関連しています。

高可用性ノード

Note:
ノードは Zabbix サーバーによって作成され、API 経由で変更することはできません。

高可用性ノードオブジェクトには、次のプロパティがあります。

Property	Type	Description
ha_nodeid	ID	ノードの ID。
name	string	`zabbix_server.conf` の HANodeName 設定項目を使用してノードに割り当てられた名前。スタンドアロンモードで動作しているサーバーでは空です。
address	string	ノードが接続元として使用する IP または DNS 名。
port	integer	ノードが稼働しているポート。
lastaccess	integer	ハートビート時刻、つまりノードからの最後の更新時刻。UTC タイムスタンプ。
status	integer	ノードの状態。 可能な値: 0 - standby; 1 - 手動で停止; 2 - 利用不可; 3 - active.

hanode.get

説明

`integer/array hanode.get(object parameters)`

このメソッドを使用すると、指定したパラメータに従って高可用性クラスタノードの一覧を取得できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。詳細については、User rolesを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
ha_nodeids	ID/array	指定したノード ID を持つノードのみを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 サポートするプロパティ: `name`, `address`, `status`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `name`, `lastaccess`, `status`。
countOutput	flag	これらのパラメータについては、リファレンスコメントを参照してください。
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列

- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ステータスで並べ替えたノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "ckuo7i1nw000h0sajj3l3hh8u": {
      "ha_nodeid": "ckuo7i1nw000h0sajj3l3hh8u",
      "name": "node-active",
      "address": "192.168.1.13",
      "port": "10051",
      "lastaccess": "1635335704",
      "status": "3"
    },
    "ckuo7i1nw000e0sajwfttc1mp": {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    },
    "ckuo7i1nv000c0sajz85xcrtt": {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    "ckuo7i1nv000a0saj1fcdkeu4": {
      "ha_nodeid": "ckuo7i1nv000a0saj1fcdkeu4",
      "name": "node2",
      "address": "192.168.1.6",
      "port": "10051",
      "lastaccess": "1635335705",
      "status": "0"
    }
  },
  "id": 1
}
```

ID で特定のノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
```

```
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "name": "node4",
            "address": "192.168.1.8",
            "port": "10052",
            "lastaccess": "1635334214",
            "status": "1"
        },
        {
            "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
            "name": "node6",
            "address": "192.168.1.10",
            "port": "10053",
            "lastaccess": "1635332902",
            "status": "2"
        }
    ],
    "id": 1
}
```

停止したノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nw000g0sajjsjre7e3",
            "address": "192.168.1.12",
            "port": "10051"
        },
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "address": "192.168.1.8",
            "port": "10052"
        },
        {
            "ha_nodeid": "ckuo7i1nv000d0sajd95y1b6x",
            "address": "192.168.1.9",
```

```
            "port": "10053"
        }
    ],
    "id": 1
}
```

スタンバイノードの数を取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

特定の IP アドレスでノードのステータスを確認する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

ソース

CHaNode::get() in *ui/include/classes/api/services/CHaNode.php*.

表記法　メソッドシグネチャ

各 API メソッドのページには、メソッドシグネチャが記載されています。

例: `object action.create(object/array actions)`

シグネチャ	説明
`object`	メソッドが返すデータの型。
`action.create`	メソッドの完全な名前。
`object/array`	メソッドが受け取るデータの型。
`actions`	メソッドが受け取るデータの概要。

データ型

Zabbix API は、以下のデータ型を入力としてサポートしています。

型	説明
ID	エンティティを参照するために使用される一意の識別子。
boolean	ブール値 (`true` または `false`)。
flag	渡された値は、`null` と等しくない場合に `true` とみなされ、そうでない場合は `false` とみなされます。
integer	整数。
float	浮動小数点数。
string	テキスト文字列。
text	より長いテキスト文字列。
timestamp	Unix タイムスタンプ。
array	順序付けられた値のシーケンス (単純な配列)。
object	連想配列。
query	返されるデータを定義する値。(特定のプロパティのみを返す) プロパティ名の配列として定義するか、定義済みの値のいずれかとして定義できます。 `extend` - すべてのオブジェクトプロパティを返します。 `count` - 取得したレコードの数を返します。特定のサブセレクトでのみサポートされます。

Attention:
Zabbix API は常に文字列または配列としてのみ値を返します。

プロパティの動作

一部のオブジェクトプロパティには、その動作を説明する短いラベルが付けられています。使用されるラベルは次のとおりです:

- 読み取り専用 - プロパティの値は自動的に設定され、特定の状況下でもユーザーが定義または変更することはできません (例: 継承オブジェクトまたは検出されたオブジェクトの場合は読み取り専用)。
- 書き込み専用 - プロパティの値は設定できますが、作成後にアクセスすることはできません。
- 定数 - プロパティの値はオブジェクトの作成時に設定できますが、作成後は変更できません。
- サポート - プロパティの値の設定は必須ではありませんが、特定の状況下では設定が許可されます (例: `type` が” シンプルチェック”、” 外部チェック”、”SSH エージェント”、”TELNET エージェント”、または”HTTP エージェント” に設定されている場合、サポート)。ただし、サポートのプロパティは、状況に関係なくデフォルト値に設定される場合があることに注意してください。
- 必須 - プロパティの値は、すべての操作 (取得操作を除く) または特定の条件 (例: 作成操作の場合は必須、`operationtype` が” グローバルスクリプト” に設定され、`opcommand_hst` が設定されていない場合は必須) で設定する必要があります。

Note:
更新操作の場合、プロパティは更新操作中に設定されると” 設定済み” とみなされます。

ラベルが付いていないプロパティはオプションです。

パラメーターの動作

一部の操作パラメーターには、操作における動作を説明する短いラベルが付けられています。使用されるラベルは次のとおりです:

- 読み取り専用 - パラメーターの値は自動的に設定され、特定の条件下でもユーザーが定義または変更することはできません (例: 継承されたオブジェクトまたは検出されたオブジェクトの場合は読み取り専用)。
- 書き込み専用 - パラメーターの値は設定できますが、設定後はアクセスできません。
- サポート - パラメーターの値の設定は必須ではありませんが、特定の条件下では設定が許可されます (例: Proxy オブジェクトの `operating_mode` が" パッシブプロキシ" に設定されている場合はサポート)。ただし、サポートされたパラメーターは、条件に関わらずデフォルト値に設定される場合があります。
- 必須 - パラメーターの値の設定は必須です。

ラベルが付いていないパラメーターはオプションです。

予約 **ID** 値**"0"**　予約 ID 値"0" は、要素をフィルタリングしたり、参照されているオブジェクトを削除したりするために使用できます。たとえば、ホストから参照されているプロキシを削除するには、proxyid を 0 に設定する必要があります ("proxyid": "0")。また、サーバーオプション proxyids によって監視されるホストをフィルタリングするには、proxyids を 0 に設定する必要があります ("proxyids": "0")。

一般的な**"get"** メソッドのパラメータ　以下のパラメータは、すべての `get` メソッドでサポートされています。

パラメータ	型	説明
countOutput	boolean	実際のデータの代わりに、結果内のレコード数を返します。
editable	boolean	`true` に設定すると、ユーザーが書き込み権限を持つオブジェクトのみを返します。 デフォルト: `false`。
excludeSearch	boolean	`search` パラメータで指定された条件に一致しない結果を返します。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名 (例: `host.get` のホストオブジェクトのプロパティ、`item.get` のアイテムオブジェクトのプロパティなど)、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型のプロパティはサポートしていません。 一部のメソッドでは、このパラメータに対して固有の機能があり、メソッドページで説明されています (例: host.get の `filter` パラメータはホストインターフェースのプロパティもサポートしています)。
limit	integer	返されるレコード数を制限します。
output	query	返されるオブジェクトのプロパティ。 オブジェクト ID (つまり `hostid`、`itemid` など) は、`output` パラメータで指定されていない場合でも、常にレスポンスに含まれることに注意してください。 デフォルト: `extend`。
preservekeys	boolean	結果の配列で ID をキーとして使用します。
search	object	指定されたパターンに一致する結果を返します (大文字と小文字は区別されません)。 キーがプロパティ名 (例: `host.get` のホストオブジェクトのプロパティ、`item.get` のアイテムオブジェクトのプロパティなど)、値が検索対象の文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索が実行されます。 `string` および `text` データ型のプロパティのみをサポートします。 一部のメソッドでは、このパラメータに対して固有の機能があり、メソッドページで説明されています (例: host.get の `search` パラメータはホストインターフェースのプロパティもサポートしています)。
searchByAny	boolean	`true` に設定すると、`filter` または `search` パラメータで指定されたすべての条件ではなく、いずれかの条件に一致する結果を返します。 デフォルト: `false`。
searchWildcardsEnabled	boolean	`true` に設定すると、`search` パラメータでワイルドカード文字として"*" を使用できるようになります。 デフォルト: `false`。

パラメータ	型	説明
sortfield	string/array	指定されたプロパティで結果を並べ替えます。並べ替えに使用できるプロパティの一覧については、各 API get メソッドの説明を参照してください。マクロは並べ替え前に展開されません。 値が指定されていない場合、データは並べ替えられずに返されます。
sortorder	string/array	並べ替え順。配列が渡された場合、各値は `sortfield` パラメータで指定された対応するプロパティに適用されます。 指定可能な値: ASC - (デフォルト) 昇順; DESC - 降順。
startSearch	boolean	`search` パラメータはフィールドの先頭を比較し、代わりに `LIKE "…%"` 検索を実行します。 `searchWildcardsEnabled` が `true` に設定されている場合は無視されます。

エンティティの起源フラグ　Get メソッドは、低レベルディスカバリ（LLD ルール/LLD ルールプロトタイプ、アイテム/アイテムプロトタイプなど）に関連するエンティティに対して `flags` プロパティを返します。このプロパティは、エンティティがディスカバリされたものかどうかを示すのに便利です。なぜなら、ディスカバリされたエンティティの編集は制限されるためです。

`flags` プロパティは、これらの値の組み合わせ（"+" 演算）に基づいて結果を返します。

値	説明
0	基本エンティティ（アイテム、トリガー、グラフ、ホスト）
1	低レベルディスカバリルール
2	任意のプロトタイプ（アイテムプロトタイプ、トリガープロトタイプ、LLD ルールプロトタイプなど）
4	ディスカバリされたエンティティ（ディスカバリされたアイテム、トリガー、グラフ、ホスト、LLD ルール）

`flags` プロパティによって返される組み合わせ値は次のとおりです。

値	組み合わせ	説明
0	0	通常のエンティティ（アイテム、トリガー、グラフ、ホスト）。
2	2	エンティティプロトタイプ（アイテムプロトタイプ、トリガープロトタイプなど）。
6	2+4	ディスカバリされたアイテム、トリガー、グラフ、ホスト（プロトタイプから変換）。
1	1	低レベルディスカバリルール。
3	1+2	低レベルディスカバリルールプロトタイプ。
5	1+4	ディスカバリされた低レベルディスカバリルール（プロトタイプから変換）。
7	1+2+4	ディスカバリされた低レベルディスカバリルールプロトタイプ。

例　ユーザー権限のチェック

ユーザーは、名前が"MySQL" または"Linux" で始まるホストに書き込む権限を持っていますか？

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
"result" が"0" の場合は、読み取り/書き込み権限を持つホストが存在しないことを意味します。

不一致を数える

名前に部分文字列"ubuntu" が含まれていないホストの数を数えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

ワイルドカードを使用したホストの検索

名前に"server" という単語が含まれ、インターフェースポートが"10050" または"10071" であるホストを検索します。結果をホスト名で降順に並べ替え、5 つに限定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    ],
    "id": 1
}
```


”preservekeys” を含むワイルドカードを使用したホストの検索

前のリクエストにパラメーター”preservekeys” を追加すると、結果はオブジェクトの ID をキーとする連想配列として返されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
```

```
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5,
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "50003": {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50005": {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50004": {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "99032": {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        "99061": {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
```

```
                }
            ]
        }
    },
    "id": 1
}
```

付録 **2. 7.4** から **8.0** への変更点

互換性のない変更点　history

ZBXNEXT-10069 history.get メソッドに新しいパラメーター maxValueSize が追加されました。バイナリ値を持つアイテム (history=5) のデータを要求する場合は、maxValueSize を null に設定してください。そうしないと、history.get はアイテムの値を 64KiB に切り詰めて返します。

ホスト

ZBXNEXT-9185 host オブジェクトの monitored_by プロパティに、動作の説明を追加しました。
ZBXNEXT-10387 host.massupdate: メソッドを削除しました。
代わりに host.update を使用してください。

ホストグループ

ZBXNEXT-10387 hostgroup.massupdate: メソッドが削除されました。代わりに hostgroup.update、hostgroup.massadd、または hostgroup.massremove を使用してください。

hostinterface

ZBXNEXT-10814 hostinterface.create、hostinterface.update、hostinterface.delete、hostinterface.massadd、hostinterface.massremove メソッドは、hostinterface API から削除されました。代わりに、host オブジェクトのメソッドを使用してください。

ロール

ZBXNEXT-9185 ロールのaction オブジェクトの name プロパティに、新しい select_server_for_monitoring 値を追加しました。

テンプレート

ZBXNEXT-10387 template.massupdate: メソッドが削除されました。代わりに template.update、template.massadd、または template_massremove を使用してください。

テンプレートグループ

ZBXNEXT-10387 templategroup.massupdate: メソッドは削除されました。代わりに templategroup.update、templategroup.massadd、または templategroup.massremove を使用してください。

ユーザーグループ

ZBXNEXT-9185 usergroup.create メソッドおよびusergroup.update メソッドに、新しいパラメータ proxies と proxy_groups を追加しました。
 usergroup.get メソッドに、新しいパラメータ proxyids、proxy_groupids、selectProxies、selectProxyGroups を追加しました。
 ユーザーグループ オブジェクトに、新しいプロパティ proxy_mode と proxy_group_mode を追加しました。

その他の変更点およびバグフィックス　アクション

ZBXNEXT-9399 イベントソースがディスカバリまたは自動登録の場合、conditiontype アクション フィルターパラメータに新しい値 29（プロキシグループ）を追加しました。

apmlog

ZBXNEXT-10083 新しいAPM log API を追加しました。

apmmetric

ZBXNEXT-10083 新しいAPM メトリクス API を追加しました。

apmspan

ZBXNEXT-10083 新しいAPM span API を追加しました。

apmtrace

ZBXNEXT-10083 新しいAPM trace API を追加しました。

設定

ZBXNEXT-5419 configuration.importcompare メソッドに新しいパラメーター returnMissingObjects を追加しました。
 ZBXNEXT-5419 rules オブジェクト（configuration.importcompare メソッド）に新しいパラメーター dashboards を追加しました。
 ZBXNEXT-5419 configuration.import メソッドに新しいパラメーター returnMissingObjects を追加しました。
 ZBXNEXT-5419 rules オブジェクト（configuration.import メソッド）に新しいパラメーター dashboards を追加しました。ZBXNEXT-10390 configuration.export メソッドの options オブジェクトに新しい global_regexes パラメーターを追加しました。ZBXNEXT-10390 configuration.importcompare およびconfiguration.import メソッドの rules オブジェクトに新しい global_regexes パラメーターを追加しました。

コネクタ

ZBXNEXT-10069 connector オブジェクトに、新しい item_value_type の値（64 - JSON）が追加されました。

ダッシュボード

ZBXNEXT-4769 graphウィジェットに、新しいダッシュボードウィジェットフィールド show_hostnames を追加しました。
 ZBXNEXT-9546 svggraphおよびpiechartウィジェットに、新しいダッシュボードウィジェットフィールド ds.0.item_tags_evaltype、ds.0.item_tags.0.tag、ds.0.item_tags.0.operator、ds.0.item_tags.0.value を追加しました。
 ZBXNEXT-10158 topitemsウィジェットに、新しいダッシュボードウィジェットフィールド columns.0.aggregate_columns、columns.0.combined_column_name、columns.0.column_aggregate_function を追加しました。
 ZBXNEXT-9424、ZBXNEXT-4769 新しいダッシュボードウィジェットタイプscatterplotを追加しました。
 ZBXNEXT-8160 geomapウィジェットに、新しいダッシュボードウィジェットフィールド clustering_mode および clustering_zoom_level を追加しました。
 ZBXNEXT-943 graphウィジェットに、新しいダッシュボードウィジェットフィールド ds.0.invert_values を追加しました。
 ZBXNEXT-10657 honeycombウィジェットの Color および Background color パラメータに、新しい「dark-blue-theme」テーマのデフォルト値を追加しました。

drule

ZBXNEXT-9185 ディスカバリルール オブジェクトの timeout プロパティの動作を更新しました。

デバイス

ZBXNEXT-10357 新しいDevice APIを追加しました。メソッドは device.get および device.offboard です。

event

ZBXNEXT-1150 event.get メソッドの selectAcknowledges パラメータに maintenanceid プロパティを追加しました。

履歴

ZBXNEXT-10069 history.get メソッドに、新しい history の値（6 - JSON）を追加しました。
 ZBXNEXT-10069 新しい JSON history オブジェクトを追加しました。

hostprototype

ZBXNEXT-9546 hostprototype.get メソッドに新しいパラメータ selectInheritedTags を追加しました。

httptest

ZBXNEXT-9546 httptest.get メソッドに新しいパラメータ inheritedTags および selectInheritedTags を追加しました。

アイテム

ZBXNEXT-10249 item オブジェクトに新しい type 値（24 - Telemetry query）を追加しました。
 ZBXNEXT-10249 item オブジェクトに新しいプロパティ query、time_shift、lookback_limit、granularity を追加しました。
 ZBXNEXT-9185 item オブジェクトの timeout プロパティの動作を更新しました。
 ZBXNEXT-10069 item オブジェクトに新しい value_type 値（6 - JSON）を追加しました。
 ZBXNEXT-9546 item.get メソッドに新しいパラメータ inheritedTags と selectInheritedTags を追加しました。

アイテムプロトタイプ

ZBXNEXT-10249 アイテムプロトタイプ オブジェクトに新しい type 値（24 - Telemetry query）を追加しました。
 ZBXNEXT-10249 アイテムプロトタイプ オブジェクトに新しいプロパティ query、time_shift、lookback_limit、granularity を追加しました。
 ZBXNEXT-10069 itemprototype オブジェクトに新しい value_type 値（6 - JSON）を追加しました。
 ZBXNEXT-9546 itemprototype.get メソッドに新しいパラメータ selectInheritedTags を追加しました。

メンテナンス

ZBXNEXT-8339 maintenance.create メソッドとmaintenance.update メソッドに triggers および event_names プロパティを追加しました。problem_tags プロパティに「等しくない」および「含まない」演算子を追加しました。
 ZBXNEXT-8339 maintenance.get メソッドに triggerids、selectTriggers、selectEventNames パラメータを追加しました。

メディアタイプ

ZBXNEXT-10357 メディアタイプ オブジェクトに新しい type 値（5 - Push）を追加しました。

障害

ZBXNEXT-1150 problem.get メソッドの selectAcknowledges パラメータに maintenanceid プロパティを追加しました。

プロキシ

ZBXNEXT-10084 proxy.get メソッドに新しいパラメータ selectApm を追加しました。
 ZBXNEXT-10084 proxy.create およびproxy.update メソッドに新しいパラメータ apm を追加しました。
 ZBXNEXT-10084 新しいアプリケーションパフォーマンス監視（APM） オブジェクトを追加しました。
 ZBXNEXT-10249 proxy オブジェクトに新しいプロパティ timeout_telemetry_query を追加しました。

regexp

ZBXNEXT-10390 正規表現オブジェクトに新しい description プロパティが追加されました。

ロール

ZBXNEXT-10469 ロールルール オブジェクトの api.access プロパティのデフォルト値を変更しました。デフォルト値は 0（無効）になりました。

serverinfo

ZBXNEXT-10357 新しいサーバー情報 APIとメソッド serverinfo.get を追加しました。

設定

ZBXNEXT-10249 settings オブジェクトに新しいプロパティ timeout_telemetry_query を追加しました。
 ZBXNEXT-10680 settings オブジェクトに新しいプロパティ apm_global_db を追加しました。
 ZBXNEXT-10357 settings オブジェクトに新しいプロパティ device_link_timeout を追加しました。
 ZBXNEXT-10657 settings オブジェクトの default_theme プロパティに dark-blue-theme の値を追加しました。

テンプレート

ZBXNEXT-9546 template.get メソッドに新しいパラメータ inheritedTags および selectInheritedTags を追加しました。

templatedashboard

ZBXNEXT-9424 新しいテンプレートダッシュボードウィジェットタイプ散布図を追加しました。

トリガー

ZBXNEXT-9546 trigger.get メソッドに新しいパラメータ inheritedTags および selectInheritedTags を追加しました。
 ZBXNEXT-8339 trigger.get メソッドに新しいプロパティ maintenanceids を追加しました。

triggerprototype

ZBXNEXT-9546 triggerprototype.get メソッドに新しいパラメータ selectInheritedTags を追加しました。

ユーザー

ZBXNEXT-10657 user オブジェクトの theme プロパティに dark-blue-theme 値を追加しました。
 ZBXNEXT-8339 user.create、user.get、およびuser.update メソッドに新しい default_maintenance_period プロパティを追加しました。

userdirectory

ZBXNEXT-9834 ユーザーディレクトリオブジェクトに、idp_certificate、sp_private_key、sp_certificate、idp_certificate_hash、sp_private_key_hash、sp_certificate_hash プロパティを追加しました。
 DEV-5022 ユーザーディレクトリオブジェクトの sign_assertions、sign_authn_requests、sign_messages、sign_logout_requests、sign_logout_responses プロパティのデフォルト値を変更しました。デフォルト値は 1（有効）になりました。

8.0.1 このバージョンはまだリリースされていません。

概要　Zabbix は多くの機能を提供していますが、追加機能の余地は常にあります。拡張機能は、Zabbix のソースコードを変更することなく、監視機能を修正・強化する便利な方法です。

Zabbix の機能は、組み込みの拡張オプション（トラッパーアイテム、ユーザーパラメータなど）を使用するか、カスタム拡張機能（ロード可能モジュール、プラグインなど）を使用または作成することで拡張できます。

このセクションでは、Zabbix を拡張するためのすべてのオプションの概要と参照を提供します。

カスタムコマンドによるデータ収集　トラッパーアイテム

トラッパーアイテムは、データを取得する代わりに受信するアイテムです。トラッパーアイテムは、Zabbix サーバーまたはプロキシに特定のデータを送信する場合に便利です。たとえば、長時間実行されるユーザースクリプトの場合の定期的な可用性データやパフォーマンスデータです。Zabbix サーバーまたはプロキシへのデータ送信は、Zabbix senderユーティリティまたは Zabbix sender プロトコルを使用して行えます。Zabbix サーバーへのデータ送信は、`history.push` API メソッドを使用して行うこともできます。

外部チェック

外部チェックは、実行可能ファイル (例えば、シェルスクリプトやバイナリ) を実行してチェックを行うアイテムです。

外部チェックは Zabbix サーバーまたはプロキシ (ホストがプロキシによって監視されている場合) によって実行され、監視対象ホストでエージェントを実行する必要はありません。

ユーザーパラメータ

ユーザーパラメータは、ユーザー定義のキーに関連付けられたユーザー定義のコマンドであり、実行時に Zabbix エージェントが稼働しているホストから必要なデータを取得できます。ユーザーパラメータは、Zabbix で事前定義されていないエージェントまたはエージェント 2 のアイテムを設定する際に便利です。

`system.run[]` Zabbix エージェントのアイテム

`system.run[]` Zabbix エージェントのアイテムは、ユーザー定義コマンド用のアイテムです（定義済みキー `system.run[]` に関連付けられています。例: `system.run[myscript.sh]`）。Zabbix エージェントが実行されているホスト上で実行できます。

Note:
`system.run[]` アイテムはデフォルトで無効になっています。使用する場合は、有効化（許可）し、Zabbix エージェントまたは Zabbix エージェント 2 の設定ファイルで定義する必要があります（`AllowKey` または`AllowKeyRegexp` 設定パラメータ）。

Attention:
外部チェック、ユーザーパラメーター、`system.run[]` Zabbix エージェントアイテムなどのアイテムでのユーザー定義コマンドは、Zabbix コンポーネントの実行に使用される OS ユーザーから実行されます。これらのコマンドを実行するには、このユーザーに必要な権限が必要です。

HTTP エージェントアイテム

HTTP エージェントアイテムは、HTTP/HTTPS 経由でデータリクエストを実行するためのアイテムです。HTTP エージェントアイテムは、*Elasticsearch* や *OpenWeatherMap* などのサービスからデータを取得するために HTTP エンドポイントへリクエストを送信したり、Zabbix API、Apache または NGINX Web サーバーのステータスを確認したりする場合に便利です。HTTP エージェントアイテム（トラッピングを有効にした場合）は、トラッパーアイテムとしても機能します。

スクリプトアイテム

スクリプトアイテムは、HTTP/HTTPS 経由でデータを取得するユーザー定義の JavaScript コードを実行するためのアイテムです。スクリプトアイテムは、HTTP エージェントアイテムで提供される機能だけでは十分でない場合に便利です。例えば、複数のステップや複雑なロジックを必要とする高度なデータ収集シナリオでは、スクリプトアイテムを設定して HTTP コールを実行し、受信したデータを処理し、変換した値を 2 回目の HTTP コールに渡すことができます。

Note:
HTTP エージェントアイテムとスクリプトアイテムは Zabbix サーバーおよびプロキシでサポートされており、監視対象ホストでエージェントを実行する必要はありません。

高度な拡張機能　ロード可能モジュール

C 言語で記述されたロード可能モジュールは、UNIX プラットフォーム上で Zabbix コンポーネント（サーバー、プロキシ、エージェント）の機能を拡張するための多用途かつパフォーマンス重視のオプションです。ロード可能モジュールは基本的に Zabbix デーモンによって使用され、起動時にロードされる共有ライブラリです。ライブラリには特定の関数を含める必要があり、Zabbix プロセスがそのファイルが実際にロードして動作できるモジュールであることを検出できるようにします。

ロード可能モジュールには、新しいメトリックの追加やその他のロジックの実装（例：Zabbix の履歴データエクスポート）などの機能拡張、優れたパフォーマンス、提供される機能の開発・利用・共有のオプションなど、多くの利点があります。これにより、メンテナンスが容易になり、Zabbix のコアコードベースとは独立して新機能を簡単に提供できるようになります。

ロード可能モジュールは、複雑な監視構成で特に有用です。組み込みシステムの監視、大量の監視パラメータ、複雑なロジックや起動時間の長い重いスクリプトを使用する場合、ユーザーパラメータ、`system.run[]` Zabbix エージェントアイテム、外部チェックなどの拡張機能はパフォーマンスに影響を与えます。ロード可能モジュールは、パフォーマンスを犠牲にすることなく Zabbix の機能を拡張する方法を提供します。

プラグイン

プラグインは、Zabbix エージェント 2 の監視機能を拡張します。
Go プログラミング言語で記述されており、ロード可能モジュール（C で記述）に代わる選択肢を提供します。

Zabbix エージェント 2 向けに独自のプラグイン (独立したデータコレクター) を開発することも可能です。詳細はDeveloper centerを参照してください。

アラートのカスタマイズ　Webhook

Webhookは、Zabbix のメディアタイプであり、Zabbix のアラート機能をヘルプデスクシステム、チャット、メッセンジャーなどの外部ソフトウェアに拡張するオプションを提供します。スクリプトアイテムと同様に、Webhook はカスタム JavaScript コードを使用して HTTP コールを行うのに便利であり、たとえば Microsoft Teams、Discord、Jira などのさまざまなプラットフォームに通知をプッシュすることができます。また、作成されたヘルプデスクチケットに関する情報などのデータを返し、それを Zabbix に表示することも可能です。

既存の Webhook は、Zabbix のGit リポジトリで利用できます。カスタム Webhook の開発については、Webhook 開発ガイドラインを参照してください。

アラートスクリプト

アラートスクリプトは、Zabbix のメディアタイプの一つで、Zabbix アラートを処理するための独自の方法（スクリプト）を作成するオプションを提供します。アラートスクリプトは、Zabbix でアラートを送信する既存のメディアタイプに満足できない場合に便利です。

フロントエンドのカスタマイズ　カスタムテーマ

カスタムテーマを使用して、Zabbix フロントエンドの外観を変更することができます。独自のテーマの作成と適用方法については、手順を参照してください。

フロントエンドモジュール

フロントエンドモジュールは、サードパーティ製モジュールの追加や独自開発により、Zabbix フロントエンドの機能を拡張するオプションを提供します。フロントエンドモジュールを使用すると、新しいメニュー項目やそれに対応するビュー、アクションなどを追加できます。

グローバルスクリプト　グローバルスクリプトは、設定されたスコープとユーザー権限に応じて、監視対象で実行できるユーザー定義のコマンドセットです（シェル（/bin/sh）インタープリターによる）。グローバルスクリプトは、以下のアクションに設定できます。

- アクションオペレーション
- 手動ホストアクション
- 手動イベントアクション

グローバルスクリプトは多くのケースで有用です。例えば、アクションオペレーションや手動ホストアクションに設定した場合、グローバルスクリプトを使用して、アプリケーション（Web サーバー、ミドルウェア、CRM など）の再起動やディスクスペースの解放 (古いファイルの削除、`/tmp` のクリーンアップなど）などのリモートコマンドを自動または手動で実行できます。また、手動イベントアクションに設定した場合は、外部システムでの問題チケットの管理などにも利用できます。

グローバルスクリプトは、Zabbix サーバー、プロキシ、エージェントによって実行できます。

> **Attention:**
> ユーザー定義コマンドは、Zabbix コンポーネントの実行に使用される OS ユーザーから実行されます。これらのコマンドを実行するには、このユーザーに必要な権限が必要です。

Zabbix API Zabbix APIは、Zabbix フロントエンドの一部である HTTP ベースの API です。Zabbix API を使用すると、以下の操作をプログラムで実行できます。

- Zabbix の設定をプログラムで取得・変更する。
- Zabbix の設定をインポートおよびエクスポートする。
- Zabbix の履歴データやトレンドデータにアクセスする。
- アプリケーションを Zabbix と連携するように設定する。
- Zabbix をサードパーティ製ソフトウェアと統合する。
- 定型作業を自動化する。

Zabbix API は、多数のメソッドで構成されており、名目上は個別の API にグループ化されています。各メソッドは特定のタスクを実行します。利用可能なメソッドや、Zabbix API が提供する機能の概要については、Zabbix API のメソッドリファレンスを参照してください。

ロード可能なモジュール

概要

ロード可能なモジュールは、Zabbix の機能を拡張するためのパフォーマンス重視のオプションを提供します。

Zabbix の機能は、ユーザーパラメータ、外部チェック、system.run Zabbix エージェントアイテムなど、さまざまな方法で拡張できます。これらは非常にうまく機能しますが、主な欠点は fork() です。Zabbix はユーザーメトリックを処理するたびに新しいプロセスを fork する必要があり、これはパフォーマンスにとって良くありません。通常は大きな問題ではありませんが、組み込みシステムの監視や、多数の監視パラメータ、複雑なロジックや起動時間の長い重いスクリプトを使用する場合には深刻な問題となる可能性があります。

ロード可能なモジュールのサポートにより、パフォーマンスを犠牲にすることなく、Zabbix エージェント、サーバー、プロキシを拡張する方法が提供されます。

ロード可能なモジュールは基本的に Zabbix デーモンによって使用され、起動時にロードされる共有ライブラリです。このライブラリには特定の関数が含まれている必要があり、Zabbix プロセスがそのファイルが実際にロードして動作できるモジュールであることを検出できるようになっています。

ロード可能なモジュールには多くの利点があります。優れたパフォーマンスとあらゆるロジックを実装できる能力は非常に重要ですが、おそらく最も重要な利点は、Zabbix モジュールを開発、使用、共有できることです。これにより、トラブルのないメンテナンスが可能になり、Zabbix のコアコードベースとは独立して新しい機能をより簡単に提供できるようになります。

モジュールのライセンスおよびバイナリ形式での配布は AGPL-3.0 ライセンスによって管理されます（モジュールは実行時に Zabbix とリンクし、Zabbix ヘッダーを使用します。Zabbix の全コードは Zabbix 7.0 以降 AGPL-3.0 ライセンスでライセンスされています）。バイナリ互換性は Zabbix によって保証されません。

モジュール API の安定性は、1 つの Zabbix LTS（長期サポート）リリースサイクルの間保証されます。Zabbix API の安定性は保証されません（技術的にはモジュールから Zabbix 内部関数を呼び出すことは可能ですが、そのようなモジュールが動作する保証はありません）。

モジュール API

共有ライブラリが Zabbix モジュールとして扱われるためには、いくつかの関数を実装しエクスポートする必要があります。現在、Zabbix モジュール API には 6 つの関数があり、そのうち 1 つだけが必須で、他の 5 つはオプションです。

必須インターフェース

必須の関数は zbx_module_api_version() のみです:

```
int zbx_module_api_version(void);
```

この関数は、このモジュールで実装されている API バージョンを返す必要があります。また、モジュールを読み込むには、このバージョンが Zabbix でサポートされているモジュール API バージョンと一致していなければなりません。Zabbix がサポートするモジュール API のバージョンは ZBX_MODULE_API_VERSION です。したがって、この関数はこの定数を返す必要があります。この目的で使用されていた旧定数 ZBX_MODULE_API_VERSION_ONE は、ソース互換性を維持するために ZBX_MODULE_API_VERSION と等しい値として定義されていますが、使用は推奨されません。

オプションのインターフェース

オプションの関数は zbx_module_init(), zbx_module_item_list(), zbx_module_item_timeout(), zbx_module_history_write および zbx_module_uninit() です。

```
int zbx_module_init(void);
```

この関数は、モジュールに必要な初期化処理（ある場合）を実行する必要があります。
成功した場合は ZBX_MODULE_OK を返します。
それ以外の場合は ZBX_MODULE_FAIL を返します。
後者の場合、Zabbix は起動しません。

```
ZBX_METRIC  *zbx_module_item_list(void);
```

この関数は、モジュールがサポートするアイテムの一覧を返す必要があります。
各アイテムは ZBX_METRIC 構造体で定義されます。詳細は以下のセクションを参照してください。
一覧の終端は、key フィールドが NULL の ZBX_METRIC 構造体です。

```
void    zbx_module_item_timeout(int timeout);
```

モジュールが zbx_module_item_list() を公開している場合、この関数は、モジュールによって実装されたアイテムチェックが従うべき Zabbix 設定ファイル内のタイムアウト設定を Zabbix が指定するために使用されます。
ここで、timeout パラメータは秒単位です。

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

この関数は、Zabbix サーバーがさまざまなデータ型の履歴をエクスポートするために使用するコールバック関数を返す必要があります。
コールバック関数は ZBX_HISTORY_WRITE_CBS 構造体のフィールドとして提供され、モジュールが特定の型の履歴に関心がない場合、そのフィールドは NULL にできます。

```
int zbx_module_uninit(void);
```

この関数は、割り当て済みリソースの解放、ファイルディスクリプタのクローズなど、必要な終了処理（ある場合）を実行する必要があります。

すべての関数は、モジュールがロードされたときに Zabbix の起動時に 1 回呼び出されます。ただし、zbx_module_uninit() は例外で、モジュールがアンロードされたときに Zabbix の終了時に 1 回呼び出されます。

アイテムの定義

各アイテムは ZBX_METRIC 構造体で定義されます。

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

ここで、**key** はアイテムキーです（例: ”dummy.random”）、**flags** は CF_HAVEPARAMS または 0 のいずれかです（アイテムがパラメータを受け付けるかどうかによります）、**function** はアイテムを実装する C 関数です（例: zbx_module_dummy_random）、**test_param** は Zabbix エージェントが -p フラグ付きで起動されたときに使用されるパラメータ一覧です（例: ”1,1000”、NULL も可）。定義例は次のようになります。

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

アイテムを実装する各関数は 2 つのポインターパラメータを受け取る必要があります。1 つ目は AGENT_REQUEST 型、2 つ目は AGENT_RESULT 型です。

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

これらの関数は、アイテム値の取得に成功した場合は SYSINFO_RET_OK を返す必要があります。それ以外の場合は SYSINFO_RET_FAIL を返します。AGENT_REQUEST から情報を取得する方法と AGENT_RESULT に情報を設定する方法につ

いては、以下の"dummy" モジュールの例を参照してください。

履歴エクスポートコールバックの提供

> **Attention:**
> モジュールによる履歴のエクスポートは、Zabbix プロキシではサポートされなくなりました。

モジュールは、履歴データを種類別にエクスポートする関数を指定できます。種類は、Numeric (float)、Numeric (unsigned)、Character、Text、Log です。

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CBS;
```

それぞれの関数は、引数として history_num 個の要素を持つ history 配列を受け取る必要があります。
エクスポートする履歴データ型に応じて、history はそれぞれ以下の構造体の配列になります。

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    double      value;
}
ZBX_HISTORY_FLOAT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    zbx_uint64_t    value;
}
ZBX_HISTORY_INTEGER;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_STRING;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_TEXT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
```

```
    const char  *source;
    int     timestamp;
    int     logeventid;
    int     severity;
}
ZBX_HISTORY_LOG;
```

コールバックは、Zabbix サーバーの履歴同期プロセスによって、履歴同期処理の最後に使用されます。これは、データが Zabbix データベースに書き込まれ、value cache に保存された後です。

Attention:
履歴エクスポートモジュールで内部エラーが発生した場合は、復旧するまで監視全体を停止させるのではなく、データを破棄して Zabbix サーバーの実行を継続できるような実装にすることを推奨します。

モジュールのビルド

モジュールは現在、Zabbix のソースツリー内でビルドすることを想定しています。これは、モジュール API が Zabbix のヘッダーで定義されているいくつかのデータ構造に依存しているためです。

ロード可能なモジュールにとって最も重要なヘッダーは `include/module.h` であり、ここでこれらのデータ構造が定義されています。
`include/module.h` を正しく動作させるために必要なその他のシステムヘッダーは `stdlib.h` と `stdint.h` です。

この情報を踏まえると、モジュールをビルドする準備は整っています。
モジュールには `stdlib.h`、`stdint.h`、および `module.h` を含める必要があり、ビルドスクリプトではこれらのファイルが include パスに含まれていることを確認する必要があります。
詳細は、以下の「dummy」モジュールの例を参照してください。

もう 1 つ便利なヘッダーは `include/zbxcommon.h` で、ここには `zabbix_log()` 関数が定義されています。この関数は、ログ記録やデバッグの目的で使用できます。

設定パラメータ

Zabbix エージェント、サーバー、プロキシは、モジュールを処理するために 2 つのパラメータをサポートしています。

- `LoadModulePath` - ロード可能なモジュールの場所への絶対パス
- `LoadModule` - 起動時にロードするモジュール

モジュールは、`LoadModulePath` で指定したディレクトリに配置する必要があります。または、モジュール名の前にパスを指定する必要があります。指定したパスが絶対パス（`/` で始まるパス）の場合、`LoadModulePath` は無視されます。`LoadModule` パラメータは複数指定できます。

たとえば、Zabbix エージェントを拡張するには、次のパラメータを追加します。

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

エージェントの起動時に、`/usr/local/lib/zabbix/agent` ディレクトリから mariadb.so、apache.so、kernel.so モジュールがロードされ、dummy.so は `/usr/local/lib/zabbix` からロードされます。モジュールが存在しない場合、権限に問題がある場合、または共有ライブラリが Zabbix モジュールでない場合、エージェントは起動に失敗します。

Web インターフェースの設定

ロード可能なモジュールは、Zabbix エージェント、サーバー、およびプロキシでサポートされています。
そのため、Zabbix Web インターフェースでのアイテムタイプは、モジュールがどこにロードされるかによって異なります。
モジュールがエージェントにロードされる場合、アイテムタイプは”Zabbix agent” または”Zabbix agent (active)” である必要があります。
モジュールがサーバーまたはプロキシにロードされる場合、アイテムタイプは”Simple check” である必要があります。

Zabbix モジュールによる履歴エクスポートには、Web インターフェースの設定は不要です。
モジュールがサーバーによって正常にロードされ、少なくとも 1 つの非 NULL のコールバック関数を返す `zbx_module_history_write_cbs()` 関数を提供している場合、履歴エクスポートは自動的に有効になります。

Dummy モジュール

Zabbix には、C 言語で書かれたサンプルモジュールが含まれています。
このモジュールは `src/modules/dummy` にあります:

```
alex@alex:~trunk/src/modules/dummy$ ls -l
-rw-rw-r-- 1 alex alex 9019 Apr 24 17:54 dummy.c
-rw-rw-r-- 1 alex alex   67 Apr 24 17:54 Makefile
-rw-rw-r-- 1 alex alex  245 Apr 24 17:54 README
```

このモジュールは十分にドキュメント化されており、独自のモジュールを作成する際のテンプレートとして使用できます。

上記の説明に従って Zabbix ソースツリーのルートで `./configure` を実行した後、`dummy.so` をビルドするには `make` を実行するだけです。

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* 変数はアイテム処理のタイムアウト設定を保持します */
static int  item_timeout = 0;

/* モジュールは内部関数を static として定義し、Zabbix の内部シンボル */
/* (zbx_*) およびロード可能モジュール API 関数 (zbx_module_*) とは異なる命名規則を使用して、競合を避ける必要があります */
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
```

```
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
```

```
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* 簡潔さのため、パラメータと型の厳密な検証は行っていません */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
```

```
 *               ZBX_MODULE_FAIL - module initialization failed                *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* dummy.random の初期化 */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... を使って何か処理する */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... を使って何か処理する */
    }
```

```
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... を使って何か処理する */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... を使って何か処理する */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... を使って何か処理する */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

このモジュールは 3 つの新しいアイテムを公開します:

- dummy.ping - 常に'1' を返します
- dummy.echo[param1] - 最初のパラメータをそのまま返します。たとえば、dummy.echo[ABC] は ABC を返します
- dummy.random[param1, param2] - param1-param2 の範囲内のランダムな数値を返します。たとえば、dummy.random[1,1000000]

制限事項

ロード可能なモジュールのサポートは Unix プラットフォームでのみ実装されています。つまり、Windows エージェントでは動作しません。

場合によっては、モジュールが zabbix_agentd.conf からモジュール関連の設定パラメータを読み取る必要があることがあります。これは現在サポートされていません。モジュールでいくつかの設定パラメータを使用する必要がある場合は、モジュール固有の設定ファイルの解析を実装する必要があるでしょう。

Web インターフェースモジュール

概要

Zabbix のソースコードを変更することなく、サードパーティ製モジュールを追加したり、独自のモジュールを開発したりすることで、Zabbix Web インターフェースの機能を拡張できます。

モジュールのコードは、Zabbix のソースコードと同じ権限で実行されることに注意してください。これは次を意味します。

- サードパーティ製モジュールは有害である可能性があります。インストールするモジュールは信頼できるものでなければなりません。
- サードパーティ製モジュールのコードにエラーがあると、Web インターフェースがクラッシュする場合があります。その場合は、Web インターフェースからモジュールのコードを削除してください。Zabbix Web インターフェースを再読み込みすると、いくつかのモジュールが存在しないことを示すメッセージが表示されます。モジュール管理（*Administration* > *General* > *Modules*）に移動し、*Scan directory* をもう一度クリックして、存在しないモジュールをデータベースから削除してください。

インストール

特定のモジュールについては、必ずインストールマニュアルをお読みください。失敗を簡単に検出できるよう、新しいモジュールは 1 つずつインストールすることを推奨します。

モジュールをインストールする直前に、次の点を確認してください。

- モジュールを信頼できるソースからダウンロードしたことを確認してください。悪意のあるコードをインストールすると、データ損失などの結果を招く可能性があります。
- 同じモジュールの異なるバージョン（同じ ID）は並行してインストールできますが、有効化できるのは一度に 1 つのバージョンだけです。

モジュールをインストールする手順:

- モジュールを、Zabbix Web インターフェースのインストール先にある modules ディレクトリ内の専用フォルダに展開します（例: zabbix/ui/modules）。
- モジュールフォルダに少なくとも manifest.json ファイルが含まれていることを確認します。
- モジュール管理 に移動し、ディレクトリをスキャンボタンをクリックします。
- 新しいモジュールが、バージョン、作成者、説明、ステータスとともに一覧に表示されます。
- ステータスをクリックしてモジュールを有効化します。

トラブルシューティング:

問題	解決策
モジュールが一覧に表示されない	モジュールが一覧に表示されない場合は、manifest.json が modules/your-module/ に存在することを確認してください。存在しない場合、モジュールを誤ったディレクトリに展開した可能性があります。存在する場合は、そのモジュールが使用中の Zabbix バージョンと互換性がない可能性があります。また、Web サーバーのユーザーに、モジュールフォルダおよびすべてのサブディレクトリに対する少なくとも読み取りおよび検索権限 (r-x) があり、フォルダ内のすべてのファイルに対する読み取り権限 (r--) があることも確認してください。
Web インターフェースがクラッシュした	モジュールのコードが現在の Zabbix バージョンまたはサーバー設定と互換性がありません。モジュールファイルを削除して、Web インターフェースを再読み込みしてください。いくつかのモジュールが存在しないという通知が表示されます。モジュール管理 に移動し、もう一度 ディレクトリをスキャンをクリックして、存在しないモジュールをデータベースから削除してください。
同一の *namespace*、*ID*、または *actions* に関するエラーメッセージが表示される	新しいモジュールが、他の有効なモジュールによってすでに登録されている namespace、ID、または actions を登録しようとしました。新しいモジュールを有効化する前に、競合しているモジュール（エラーメッセージに記載）を無効化してください。

問題	解決策
技術的なエラーメッセージが表示される	エラーをモジュールの開発者に報告してください。

モジュールの開発

カスタムモジュールの開発の詳細については、開発者センターを参照してください。

付録

概要　このセクションには、メインマニュアルを補足するリファレンス資料、設定パラメータ、技術仕様、および手順ガイドが含まれています。

基本リファレンス

- インストールとセットアップ - データベースの作成、サーバーおよびエージェントのインストールと設定。
- プロセス設定 - Zabbix サーバー、プロキシ、エージェントなどの設定パラメータ。
- アイテム - アイテムキーのリファレンスとサポートされているデータ型。
- マクロ - 組み込みマクロおよびユーザーマクロの構文と使用方法。

インストールとセットアップ

このセクションでは、データベースの作成、セキュリティ設定、プラットフォーム固有のセットアップ手順、および機能の設定について説明します。

サイドバーを使用して、お使いの環境に関連する手順を確認してください。

データベースの作成

概要

Zabbix サーバーまたはプロキシのインストール時に、Zabbix データベースを作成する必要があります。

このセクションでは、Zabbix データベースを作成するための手順を説明します。サポートされている各データベースごとに、個別の手順が用意されています。

> **Note:**
> データベースロール/ユーザーを最小権限で作成してデータベースのセキュリティを向上させるには、各サポート対象データベースのデータベース作成のベストプラクティスを参照してください:

> - MySQL/MariaDB
> - PostgreSQL/TimescaleDB
>
> 安全な TLS 接続の設定については、データベースへの安全な接続を参照してください。

UTF-8 は、Zabbix でサポートされる唯一のエンコーディングです。これは、セキュリティ上の欠陥なく動作することが確認されています。他の一部のエンコーディングを使用すると、既知のセキュリティ上の問題があることに注意してください。UTF-8 への切り替えについては、Zabbix データベースの文字セットと照合順序の修復を参照してください。また、utf8mb4 照合順序でのフィルタリングの制限も参照してください。

> **Note:**
> Zabbix Git リポジトリからインストールする場合は、次の手順に進む前に以下のコマンドを実行する必要があります。

 `make dbschema`

MySQL/MariaDB

utf8mb4_bin 照合順序を使用する utf8mb4 文字セットは、Zabbix サーバー/プロキシが MySQL/MariaDB データベースで正常に動作するためにサポートされています。
 将来の問題を回避するため、新規インストールと既存のインストールの両方で、utf8mb4_bin 照合順序を使用する utf8mb4 文字セットを使用することを強く推奨します。utf8mb3 文字セットと utf8mb3_bin 照合順序を使用しているデータベースを、utf8mb4 文字セットと utf8mb4_bin 照合順序に変換するには、Zabbix データベースの文字セットと照合順序の修復の手順に従ってください。

スキーマのインポート中に、決定論的なトリガーを作成する必要があります。MySQL および MariaDB では、バイナリログが有効で、スーパーユーザー権限がなく、MySQL 設定ファイルで log_bin_trust_function_creators = 1 が設定されていない場合、GLOBAL log_bin_trust_function_creators = 1 を設定する必要があります。

Zabbix をパッケージまたはコンテナからインストールする場合は、それぞれの手順を参照してください。

Zabbix をソースからインストールする場合:

- データベースとユーザーを作成して設定します。セキュリティ上の理由から、Zabbix データベースユーザーには強力で一意のパスワードを使用してください。

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- データをデータベースにインポートし、utf8mb4 文字セットをデフォルトとして設定します（Zabbix ソースのルートディレクトリにいることを前提とします）。Zabbix プロキシデータベースの場合は、schema.sql のみをインポートしてください（images.sql および data.sql はインポートしません）。

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### Zabbix プロキシ用のデータベースを作成する場合は、ここで停止します
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

スキーマが正常にインポートされた後、log_bin_trust_function_creators を無効にできます。

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

データベースオブジェクトを作成する権限を持つデータベースユーザーが必要です。

Zabbix をパッケージまたはコンテナからインストールする場合は、それぞれの手順を参照してください。

Zabbix をソースからインストールする場合:

- データベースユーザーを作成します。

次のシェルコマンドでユーザー zabbix を作成します。プロンプトが表示されたらパスワードを入力し、同じパスワードを再入力してください（最初に sudo のパスワードを求められる場合があります）。

```
sudo -u postgres createuser --pwprompt zabbix
```

- データベースを作成します。

次のシェルコマンドは、先ほど作成したユーザーを所有者（-O zabbix）として、データベース zabbix（最後のパラメータ）を作成します。

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- 初期スキーマとデータをインポートします（Zabbix ソースのルートディレクトリにいることを前提とします）。Zabbix プロキシのデータベースでは、schema.sql のみをインポートしてください（images.sql や data.sql は不要です）。

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### Zabbix プロキシ用のデータベースを作成している場合はここで停止してください
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
上記のコマンドは、ほとんどの GNU/Linux 環境で動作する例として示しています。システムやデータベースの構成に応じて、別のコマンドを使用できます。たとえば、次のようになります。

 psql -U <username>

 データベースの設定で問題がある場合は、データベース管理者に相談してください。

TimescaleDB

TimescaleDB の作成と設定の手順は、別のセクションで説明しています。

Elasticsearch と ClickHouse

Zabbix は、履歴データをElasticsearch またはClickHouse に保存することもできます。セットアップ手順については、リンク先のページを参照してください。

SQLite

Zabbix プロキシでのみ SQLite の使用がサポートされています。

データベースが存在しない場合は自動的に作成されます。

インストールセクションに戻る。

Zabbix データベースの文字セットと照合順序の修復

MySQL/MariaDB

歴史的に、MySQL とその派生製品では'utf8' を utf8mb3 の別名として使用していました。utf8mb3 は、標準 UTF8 の MySQL 独自の 3 バイト実装であり、UTF8 自体は 4 バイトです。MySQL 8.0.28 および MariaDB 10.6.1 以降では、'utf8mb3' 文字セットは非推奨となっており、いずれサポートが削除される一方で、'utf8' は'utf8mb4' を参照するようになります。Zabbix 6.0 以降では'utf8mb4' がサポートされています。将来の問題を避けるため、'utf8mb4' の使用を強く推奨します。'utf8mb4' に切り替えるもう 1 つの利点は、補助 Unicode 文字をサポートできることです。

Warning:
Zabbix 6.0 より前のバージョンは utf8mb4 を認識しないため、utf8mb4 への変換を実行する前に、必ず先に Zabbix サーバーと DB スキーマを 6.0.x 以降へアップグレードしてください。

1. データベースの文字セットと照合順序を確認します。

例:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2_general_ci    |
+--------------------------+----------------------+
```

または:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

ご覧のとおり、ここでの文字セットは'utf8mb4' ではなく、照合順序も'utf8mb4_bin' ではないため、修正する必要があります。

2. Zabbix を停止します。

3. データベースのバックアップを作成します。

4. データベースレベルで文字セットと照合順序を修正します。

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

修正後の値:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. テーブルおよびカラムレベルで文字セットと照合順序を修正するために、スクリプト を読み込みます。

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. スクリプトを実行します。

```
                SET @ZABBIX_DATABASE = '<your DB name>';
MariaDB の場合 →  set innodb_strict_mode = OFF;
                CALL zbx_convert_utf8();
MariaDB の場合 →  set innodb_strict_mode = ON;
                drop procedure zbx_convert_utf8;
```

'utf8mb4' は、ディスク容量をわずかに多く消費することが想定される点にご注意ください。

7. エラーがなければ、修正後のデータベースでバックアップを作成してもよいでしょう。

8. Zabbix を起動します。

データベースのプライマリキーへのアップグレード

概要

このセクションでは、既存のインストール環境のテーブルを主キーに手動でアップグレードする手順を説明します。

主キーへのアップグレードにより、データのインデックス作成とアクセス方法が最適化され、クエリの高速化と容量の節約につながる場合があります。また、クラスタ構成におけるデータ管理と同期も改善され、スケーリングの支援や、一部のサーバーが障害を起こしてもシステムの信頼性を維持しやすくなります。

> **Attention:**
> このページに記載されている手順は上級ユーザー向けに設計されており、環境に応じて調整が必要になる場合があります。主キーへのアップグレードには時間とリソースがかかることがあります。十分な空きディスク容量があることを確認してください。データベースのサイズと保存されているデータによっては、この処理に history テーブルで現在使用している容量の最大 2.5 倍が必要になる場合があります。

Zabbix 6.0 以降では、新規インストール時のすべてのテーブルで主キーが使用されます。

主キーへの自動データベースアップグレードはありませんが、既存のインストール環境は Zabbix サーバーを 6.0 以降にアップグレードした後であれば、手動でアップグレードできます。

> **Attention:**
> Zabbix 7.0 以降では、テーブルを主キーにアップグレードすると、テーブルは倍精度データ型も使用するようになります。

 Zabbix 7.0（またはそれ以降）を使用している場合、テーブルはすでに倍精度を使用しています。ただし、このページの手順は、すでに倍精度を使用しているテーブルに影響を与えることなく、テーブルを主キーにアップグレードするために引き続き実行できます。

 Zabbix 6.4（またはそれ以前）を使用している場合は、先にテーブルを倍精度にアップグレードすることを検討してください。詳細については、Zabbix 7.0 ドキュメントの 拡張範囲の数値へのアップグレード を参照してください。

以下の手順が利用できます。

- MariaDB/MySQL
- PostgreSQL
- TimescaleDB

重要な注意事項

データベースのアップグレードを実行するには:

1. Zabbix サーバーを停止します。

アップグレード中は Zabbix サーバーを停止することを強く推奨します。ただし、どうしても必要な場合は、サーバーが稼働している間にアップグレードを実行することもできます（MySQL、MariaDB、および TimescaleDB なしの PostgreSQL のみ）。

2. データベースのバックアップを取ります。
3. Zabbix のバージョンに対応した最新の zabbix-sql-scripts パッケージをインストールします(例：RHEL の場合：`dnf install zabbix-sql-scripts`)。
4. データベース用のスクリプトを実行します。
5. Zabbix サーバーを起動します。

> **Warning:**
> サーバーデータベースのみに対してスクリプトを実行してください。プロキシにはこのアップグレードの恩恵はありません。

データベースでパーティションを使用している場合は、DB 管理者または Zabbix サポートにお問い合わせください。

CSV ファイルは、主キーへのアップグレードが正常に完了した後に削除できます。

オプションで、Zabbix フロントエンドをメンテナンスモードに切り替えることができます。

MariaDB/MySQL

エクスポートとインポートは、セッションが切断されないように tmux/screen で実行する必要があります。

参照: 重要な注意事項

MySQL 8.4+ と mysqlsh

この方法は稼働中の Zabbix サーバーでも使用できますが、アップグレード中はサーバーを停止することを推奨します。
MySQL Shell (*mysqlsh*) は インストール されており、DB に接続できる必要があります。

- root（推奨）または FILE 権限を持つ任意のユーザーとして MySQL コンソールにログインします。
- local_infile 変数を有効にして MySQL を起動します。
- `history_upgrade_prepare.sql` を実行して、古いテーブルの名前を変更し、新しいテーブルを作成します。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- データをエクスポートおよびインポートします。

mysqlsh 経由で接続します。ソケット接続を使用する場合は、パスの指定が必要になることがあります。

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

次の方法で JavaScript モードに切り替えます。

```
\js
```

その後、以下のコードを実行します（CSVPATH は必要に応じて変更できます）。

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

”JavaScript is not supported” というメッセージが表示される場合、MySQL Shell のインストールには JS サポートが含まれていません。
その場合は、Oracle から公式の MySQL Shell パッケージ をインストールするか（またはソースからビルドして）、JavaScript モードが有効になるようにしてください。

- 古いテーブルを削除するには、移行後の手順 に従ってください。

MariaDB/MySQL 8.4+ で mysqlsh を使用しない場合

このアップグレード方法は時間がかかるため、*mysqlsh* を使用したアップグレードができない場合にのみ使用してください。

テーブルのアップグレード

- MySQL コンソールに root（推奨）または FILE 権限を持つ任意のユーザーでログインします。
- 稼働中の Zabbix サーバーで移行を行う場合は、*local_infile* 変数を有効にして MySQL を起動します。
- `history_upgrade_prepare.sql` を実行して、古いテーブルの名前を変更し、新しいテーブルを作成します。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

サーバー停止時の移行

移行中のタイムアウトを避けるため、データ移行の前に max_execution_time（MySQL）または max_statement_time（MariaDB）を無効にする必要があります。

MySQL の場合:

```
SET @@max_execution_time=0;
```

MariaDB の場合:

```
SET @@max_statement_time=0;
```

```
INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

古いテーブルを削除するには、移行後の手順に従ってください。

サーバー稼働中の移行

インポート/エクスポートが有効なパスを確認します:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

secure_file_priv の値がディレクトリへのパスである場合、エクスポート/インポートはそのディレクトリ内のファイルに対して実行されます。この場合、クエリ内のファイルパスをそれに合わせて編集するか、アップグレード中は *secure_file_priv* の値を空文字列に設定してください。

secure_file_priv の値が空の場合、エクスポート/インポートは任意の場所から実行できます。

secure_file_priv の値が NULL の場合は、エクスポートされたテーブルデータを含むパス（上記の例では /var/lib/mysql-files/）に設定してください。

詳細は、MySQL documentation または MariaDB documentation を参照してください。

データのエクスポート中のタイムアウトを避けるため、エクスポートの前に max_execution_time（MySQL）または max_statement_time（MariaDB）を無効にする必要があります。

MySQL の場合:

```
SET @@max_execution_time=0;
```

MariaDB の場合:

```
SET @@max_statement_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TER
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESC

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED
```

古いテーブルを削除するには、移行後の手順に従ってください。

PostgreSQL

エクスポートとインポートは、セッションが切断されないように tmux/screen で実行する必要があります。TimescaleDB を使用しているインストールでは、このセクションはスキップし、PostgreSQL + TimescaleDB に進んでください。

関連情報: 重要な注意事項

テーブルのアップグレード

- history_upgrade_prepare.sql を使用してテーブル名を変更します:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

サーバー停止中の移行

- 現在の履歴をエクスポートして一時テーブルにインポートし、その後、重複を無視しながら新しいテーブルにデータを挿入します:

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

INSERT のパフォーマンスを改善するためのヒントも参照してください: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL。

- 古いテーブルを削除するには、移行後の手順 に従ってください。

稼働中のサーバーでの移行

- 現在の履歴をエクスポートし、一時テーブルにインポートした後、重複を無視しながら新しいテーブルにデータを挿入します:

```
\copy history_old TO '/tmp/history.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history (
    itemid                   bigint                                   NOT NULL,
    clock                    integer         DEFAULT '0'              NOT NULL,
    value                    DOUBLE PRECISION DEFAULT '0.0000'         NOT NULL,
    ns                       integer         DEFAULT '0'              NOT NULL
);
\copy temp_history FROM '/tmp/history.csv' DELIMITER ',' CSV
INSERT INTO history SELECT * FROM temp_history ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_uint_old TO '/tmp/history_uint.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_uint (
    itemid                   bigint                                   NOT NULL,
    clock                    integer         DEFAULT '0'              NOT NULL,
    value                    numeric(20)     DEFAULT '0'              NOT NULL,
    ns                       integer         DEFAULT '0'              NOT NULL
);
\copy temp_history_uint FROM '/tmp/history_uint.csv' DELIMITER ',' CSV
INSERT INTO history_uint SELECT * FROM temp_history_uint ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_str_old TO '/tmp/history_str.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_str (
    itemid                   bigint                                   NOT NULL,
    clock                    integer         DEFAULT '0'              NOT NULL,
    value                    varchar(255)    DEFAULT ''               NOT NULL,
    ns                       integer         DEFAULT '0'              NOT NULL
);
\copy temp_history_str FROM '/tmp/history_str.csv' DELIMITER ',' CSV
INSERT INTO history_str (itemid,clock,value,ns) SELECT * FROM temp_history_str ON CONFLICT (itemid,clock,n

\copy history_log_old TO '/tmp/history_log.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_log (
    itemid                   bigint                                   NOT NULL,
    clock                    integer         DEFAULT '0'              NOT NULL,
    timestamp                integer         DEFAULT '0'              NOT NULL,
    source                   varchar(64)     DEFAULT ''               NOT NULL,
```

```
    severity                 integer         DEFAULT '0'               NOT NULL,
    value                    text            DEFAULT ''                NOT NULL,
    logeventid               integer         DEFAULT '0'               NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_log FROM '/tmp/history_log.csv' DELIMITER ',' CSV
INSERT INTO history_log SELECT * FROM temp_history_log ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_text_old TO '/tmp/history_text.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_text (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    text            DEFAULT ''                NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_text FROM '/tmp/history_text.csv' DELIMITER ',' CSV
INSERT INTO history_text SELECT * FROM temp_history_text ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

- 旧いテーブルを削除するには、移行後の手順に従ってください。

PostgreSQL + TimescaleDB

エクスポートとインポートは、セッションが切断されないように tmux/screen 内で実行する必要があります。
アップグレード中は Zabbix サーバーを停止しておく必要があります。

See also: Important notes

- history_upgrade_prepare.sql を使用してテーブル名を変更します。
  - 圧縮が有効な場合(デフォルトインストール)、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patc からスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - 圧縮が無効な場合、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compres からスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- 圧縮設定に基づいて TimescaleDB の hypertable 移行スクリプトを実行します:
  - 圧縮が有効な場合(デフォルトインストール)、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patc からスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
    ```

  - 圧縮が無効な場合、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compres からスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/trend
    ```

See also: INSERT パフォーマンスを改善するための Tips。

- 古いテーブルを削除するには、post-migration instructions に従ってください。

移行後

すべてのデータベースで、移行が完了したら、次のことを行います。

- すべてが期待通りに動作することを確認します。
- 古いテーブルを削除します。

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- TimescaleDB の場合は、次の古いテーブルも削除します。

```
DROP TABLE trends_old;
```

こちらもご参照ください

- パーティショニングのための auditlog テーブルの準備

auditlog テーブルをパーティション分割用に準備する

概要

一部のデータベース（例えば、MySQL）では、パーティション分割カラムがテーブルの一意制約の一部である必要があります。そのため、auditlog テーブルを時間でパーティション分割するには、主キーを auditid から複合キー auditid + clock に変更する必要があります。

このセクションでは、auditlog テーブルの主キーを変更する手順について説明します。

> **Attention:**
> このページで提供されている手順は、上級ユーザー向けに設計されています。これらの手順は、お使いの構成に合わせて調整が必要な場合がありますのでご注意ください。主キーの変更は、将来のアップグレードパッチと互換性がない場合もあるため、将来のアップグレードを手動で処理する必要があるかもしれません。

 主キーの変更は、auditlog テーブルのサイズによっては多くのリソースを消費し、時間がかかる操作となる場合があります。変更作業中は Zabbix サーバーを停止し、Zabbix フロントエンドをメンテナンスモードに切り替えることを推奨します。ただし、どうしても必要な場合は、ダウンタイムなしで主キーを変更する方法もあります（下記参照）。

auditlog テーブルのパーティショニングは、大規模なセットアップでのハウスキーピングなどを改善できます。Zabbix のハウスキーピングは現在、(TimescaleDB を除き）パーティショニングされたテーブルを利用できませんが、Zabbix のハウスキーピングを無効にし、スクリプトを使用してパーティションを削除することができます。

Zabbix 7.0 以降、TimescaleDB 用の auditlog テーブルはハイパーテーブルに変換され、ハウスキーパーがチャンク単位でデータを削除できるようになりました。既存の auditlog テーブルをハイパーテーブルにアップグレードするには、TimescaleDB スキーマのアップグレードを参照してください。

MySQL

インデックス再構築に関する重要な注意事項

MySQL は、ALTER TABLE 操作中に主キーのインデックスを自動的に再構築します。
ただし、データベースの最適なパフォーマンスを確保するために、OPTIMIZE TABLE ステートメントを使用してインデックスを手動でも再構築することを強く推奨します。

インデックスの再構築では、一時的にテーブル自体と同程度の追加ディスク容量が必要になる場合があります。
データとインデックスの現在のサイズを確認するには、次のステートメントを実行します。

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

利用可能なディスク容量が懸念される場合は、ダウンタイムなしで主キーを変更する の手順に従ってください。
他の選択肢もあります。

- sort_buffer_size MySQL パラメータを増やすと、インデックスを手動で再構築する際のディスク使用量を削減できる場合があります。
  ただし、この変数を変更すると、データベース全体のメモリ使用量に影響する可能性があります。
- 不要と思われるデータを削除して、空き容量を確保することを検討してください。
- housekeeper を実行する前に、*Data storage period* housekeeper パラメータを短くすることを検討してください。

ダウンタイムを伴う主キーの変更

1. 現在の auditlog テーブルの主キーを削除し、新しい主キーを追加します。

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. インデックスを再構築します（任意ですが、強く推奨します。インデックス再構築に関する重要な注意事項 を参照してください）。

```
OPTIMIZE TABLE auditlog;
```

ダウンタイムなしで主キーを変更する

主キーを手動で変更する方法については、ここで説明します。
または、Percona の pt-online-schema-change ツールキットを使用することもできます。
このツールキットは、auditlog テーブルの変更に使用する領域を最小限に抑えながら、以下の処理を自動的に実行します。

1. 新しい主キーを持つ新規テーブルを作成し、インデックスを作成します。

```
CREATE TABLE `auditlog_new` (
  `auditid`             varchar(25)                             NOT NULL,
  `userid`              bigint unsigned                         NULL,
  `username`            varchar(100)    DEFAULT ''              NOT NULL,
  `clock`               integer         DEFAULT '0'             NOT NULL,
  `ip`                  varchar(39)     DEFAULT ''              NOT NULL,
  `action`              integer         DEFAULT '0'             NOT NULL,
  `resourcetype`        integer         DEFAULT '0'             NOT NULL,
  `resourceid`          bigint unsigned                         NULL,
  `resource_cuid`       varchar(25)                             NULL,
  `resourcename`        varchar(255)    DEFAULT ''              NOT NULL,
  `recordsetid`         varchar(25)                             NOT NULL,
  `details`             longtext                                NOT NULL,
  PRIMARY KEY (auditid,clock)
) ENGINE=InnoDB;
CREATE INDEX `auditlog_1` ON `auditlog_new` (`userid`,`clock`);
CREATE INDEX `auditlog_2` ON `auditlog_new` (`clock`);
CREATE INDEX `auditlog_3` ON `auditlog_new` (`resourcetype`,`resourceid`);
```

2. テーブルを入れ替えます。

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. 古いテーブルから新しいテーブルへデータをコピーします。

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

必要に応じて WHERE clock 句を使った複数の INSERT INTO 文に分割し、チャンク単位で実行することで、リソースの過剰な使用を避けることができます。

4. 古いテーブルを削除します。

```
DROP TABLE auditlog_old;
```

PostgreSQL

インデックス再構築に関する重要な注意事項

PostgreSQL は、ALTER TABLE 操作中に主キーのインデックスを自動的に再構築します。
ただし、データベースの最適なパフォーマンスを確保するため、REINDEX TABLE CONCURRENTLY ステートメントを使用してインデックスを手動でも再構築することを強く推奨します。

インデックスの再構築では、一時的に現在インデックスで使用されているディスク容量の最大 3 倍が必要になる場合があります。
現在のインデックスサイズを確認するには、次のクエリを実行します。

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

利用可能なディスク容量が懸念される場合は、ダウンタイムなしで主キーを変更する の手順に従ってください。
他の選択肢もあります。

- PostgreSQL パラメーター maintenance_work_mem を増やすと、インデックスを手動で再構築する際のディスク使用量を削減できる場合があります。
  ただし、この変数を変更すると、データベース全体のメモリ使用量に影響する可能性があります。
- より多くの空き容量がある別のディスクまたはテーブルスペースがある場合は、インデックス再構築時の一時ストレージの場所を変更することを検討してください。
  PostgreSQL パラメーター temp_tablespaces を設定して、一時オブジェクト用の別のテーブルスペースを指定できます。
- 不要な可能性のあるデータを削除して、空き容量を確保することを検討してください。
- housekeeper を実行する前に、データ保存期間housekeeper パラメーターを短くすることを検討してください。

ダウンタイムを伴う主キーの変更

1. 現在の auditlog テーブルの主キーを削除し、新しい主キーを追加します。

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. インデックスを再構築します（任意ですが、強く推奨します。インデックス再構築に関する重要な注意事項を参照してください）。

```
REINDEX TABLE CONCURRENTLY auditlog;
```

ダウンタイムなしで主キーを変更する

ここでは、主キーを手動で変更する方法について説明します。
代替案として、新しいテーブルの作成、データのコピー、テーブルの入れ替えを行うために pg_repack 拡張機能を検討できます。

1. 新しい主キーを持つ新しいテーブルを作成し、インデックスを作成します。

```
CREATE TABLE auditlog_new (
  auditid               varchar(25)                             NOT NULL,
  userid                bigint                                  NULL,
  username              varchar(100)    DEFAULT ''              NOT NULL,
  clock                 integer         DEFAULT '0'             NOT NULL,
  ip                    varchar(39)     DEFAULT ''              NOT NULL,
  action                integer         DEFAULT '0'             NOT NULL,
  resourcetype          integer         DEFAULT '0'             NOT NULL,
  resourceid            bigint                                  NULL,
  resource_cuid         varchar(25)                             NULL,
  resourcename          varchar(255)    DEFAULT ''              NOT NULL,
  recordsetid           varchar(25)                             NOT NULL,
  details               text            DEFAULT ''              NOT NULL,
  PRIMARY KEY (auditid,clock)
);
CREATE INDEX auditlog_new_1 ON auditlog_new (userid,clock);
CREATE INDEX auditlog_new_2 ON auditlog_new (clock);
CREATE INDEX auditlog_new_3 ON auditlog_new (resourcetype,resourceid);
```

2. テーブルを入れ替えます。

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. 古いテーブルから新しいテーブルへデータをコピーします。

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

これは、過剰なリソース使用を避けるために、必要に応じて WHERE clock 句を使った複数の INSERT INTO 文に分割して実行できます。

4. 古いテーブルを削除します。

```
DROP TABLE auditlog_old;
```

参考

- データベースの主キーへのアップグレード

TimescaleDB のセットアップ

概要

Zabbix は、スケール時のパフォーマンス向上をサポートするために、データを自動的に時間ベースのチャンクにパーティション分割する PostgreSQL ベースのデータベースソリューションである TimescaleDB をサポートしています。

> **Warning:**
> 現在、Zabbix プロキシでは TimescaleDB はサポートされていません。

このページの手順は、以下のシナリオで使用できます。

- TimescaleDB データベースの作成、または既存の PostgreSQL テーブルから TimescaleDB への移行（設定を参照）。
- Zabbix のアップグレード時に既存の TimescaleDB データベーススキーマをアップグレードする（TimescaleDB スキーマのアップグレードを参照）。

設定

前提条件: データベースサーバーに、サポート対象バージョンの TimescaleDB 拡張機能がインストールされていること。インストール手順については、TimescaleDB documentation を参照してください。

> **Warning:**
> TimescaleDB をインストールする前に、official PostgreSQL リポジトリからサポート対象の PostgreSQL リリースをインストールしてください。

次を実行して、特定のデータベースに対して TimescaleDB 拡張機能を有効にします。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

このコマンドの実行には、データベース管理者権限が必要です。

> **Note:**
> 'public' 以外のデータベーススキーマを使用している場合は、上記のコマンドに SCHEMA 句を追加する必要があります。例:
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

次に、`postgresql/timescaledb/schema.sql` スクリプトを実行します。新規インストールの場合、このスクリプトは通常の PostgreSQL データベースが初期スキーマ/データで作成された後に実行する必要があります（database creation を参照）。

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> TimescaleDB バージョン 2.9.0 以降で `schema.sql` スクリプトを実行する際に、ベストプラクティスに従っていないことを示す警告メッセージが表示されても無視してください。この警告にかかわらず、設定は正常に完了します。

既存の履歴、トレンド、および監査ログデータの移行には、かなりの時間がかかる場合があります。移行期間中は、Zabbix サーバーおよび Web インターフェースを停止しておく必要があります。

`schema.sql` スクリプトは、次のハウスキーピングパラメータを設定します。

- アイテムの履歴保存期間を上書き
- アイテムのトレンド保存期間を上書き

履歴およびトレンドに対してパーティション化されたハウスキーピングを使用するには、これら両方のオプションを有効にする必要があります。また、履歴のみ、またはトレンドのみのいずれかについて、個別に上書きを有効にすることも可能です。

PostgreSQL および TimescaleDB では、`postgresql/timescaledb/schema.sql` スクリプトにより、さらに次の 2 つのパラメータが設定されます。

- 圧縮を有効化
- 7 日より古いレコードを圧縮

ハウスキーパーによって圧縮データを正常に削除するには、アイテムの履歴保存期間を上書きとアイテムのトレンド保存期間を上書きの両方のオプションを有効にする必要があります。上書きが無効で、テーブルに圧縮チャンクがある場合、ハウスキーパーはこれらのテーブルからデータを削除せず、ハウスキーピングおよびシステム情報セクションに不正な設定に関する警告が表示されます。

これらのパラメータはすべて、インストール後に 管理 > ハウスキーピング で変更できます。

> **Note:**
> `postgresql.conf` の PostgreSQL 構成パラメータを最適化するために、TimescaleDB が提供する timescaledb-tune ツールを実行することをお勧めします。

TimescaleDB スキーマのアップグレード

新しい TimescaleDB ハイパーテーブルを含むバージョンに Zabbix をアップグレードする場合、Zabbix サーバーはそれらのハイパーテーブルを自動的に設定しません（例えば、Zabbix 6.4 から 8.0 にアップグレードする場合、バージョン 7.0.0 および 7.0.2 で新しいハイパーテーブルが導入されています）。

新しい TimescaleDB ハイパーテーブルを設定するには、以下の手順に従ってください。

1. Zabbix サーバーを起動します。これにより既存のデータベースがアップグレードされます。
2. サーバーログファイルでデータベースのアップグレードが完了したことを確認し、完了したら Zabbix サーバーを停止します。サーバーがハイパーテーブルでないテーブルに対して圧縮を有効にしようとした場合、サーバーログに警告が記録されることに注意してください。

3. postgresql/timescaledb/schema.sql スクリプトを実行します。これにより新しい TimescaleDB ハイパーテーブルが設定されます。Zabbix 7.0.0 以降、スクリプトの場所と名前が postgresql/timescaledb.sql から postgresql/timescaledb/schema.sql に変更されていることに注意してください。

Attention:
TimescaleDB バージョン 2.9.0 以降で schema.sql スクリプトを実行する際に、ベストプラクティスに従っていないことを通知する警告メッセージは無視してください。この警告に関係なく、設定は正常に完了します。

TimescaleDB 圧縮

TimescaleDB のネイティブ圧縮は、TimescaleDB ハイパーテーブルであるすべての Zabbix テーブルでサポートされています。
TimescaleDB へのアップグレードまたは移行中は、大きなテーブルの初期圧縮にかなりの時間がかかる場合があります。

圧縮は「timescale」Timescale Community ライセンスではサポートされていますが、「apache」Apache 2.0 ライセンスではサポートされていないことに注意してください。
Zabbix が圧縮をサポートしていないことを検出すると、警告メッセージが Zabbix サーバーログに書き込まれ、ユーザーは Web インターフェースで圧縮を有効にできません。

Note:
圧縮を使用する前に、TimescaleDB documentationで圧縮について理解しておくことを推奨します。

圧縮にはいくつかの制限があることに注意してください。具体的には、次のとおりです。

- 圧縮済みチャンクの変更（挿入、削除、更新）は許可されません
- 圧縮済みテーブルのスキーマ変更は許可されません。

圧縮設定は、Zabbix Web インターフェースの *Administration* > *Housekeeping* セクションにある *History, trends and audit log compression* ブロックで変更できます。

Parameter	Default	Comments
Enable compression	Enabled	チェックボックスをオンまたはオフにしても、圧縮はすぐには有効化/無効化されません。圧縮は Housekeeper によって処理されるため、変更は最大で HousekeepingFrequency 時間の 2 倍後に反映されます (zabbix_server.conf で設定)。 圧縮を無効にした後は、圧縮期間に該当する新しいチャンクは圧縮されません。ただし、これまでに圧縮されたデータは圧縮されたまま残ります。以前に圧縮されたチャンクを非圧縮に戻すには、TimescaleDB documentationの手順に従ってください。
Compress records older than	7d	TimescaleDB 対応の古いバージョンの Zabbix からアップグレードする場合、圧縮はデフォルトでは有効になりません。 このパラメーターは 7 日未満に設定できません。 圧縮済みチャンクは不変であるため、この値より古い遅延データ（例: プロキシによって遅延したデータ）は破棄されます。

トレンドの更新性能を向上させるため、使用しているアイテム数に応じて、trends および trends_uint テーブルの"chunk_time_interval" を 30 日から 7 日以下に下げるとよい場合があります。
この設定の目的は、TimescaleDB のベストプラクティスに従い、チャンクサイズがシステムの利用可能なリソース内に収まるようにすることです。

Elasticsearch のセットアップ

Zabbix は、リレーショナルデータベースの代替として、Elasticsearch に履歴データを保存できます。

Attention:
Elasticsearch のサポートは現在実験的なものです。

このガイドでは、Elasticsearch のサポート対象バージョンのセットアップについて説明します。別のバージョンを使用している場合、一部の機能が想定どおりに動作しないことがあります。

Elasticsearch は、次の値のタイプを保存できます。

アイテムの値のタイプ	データベーステーブル	Elasticsearch type
Numeric (unsigned)	history_uint	uint
Numeric (float)	history	dbl
Character	history_str	str
Log	history_log	log
Text	history_text	text
Binary	history_bin	not supported by Zabbix
JSON	history_json	json

Note:
Elasticsearch は JSON 配列を受け付けません。JSON 値は、単一のオブジェクト、またはオブジェクトの集合のいずれかである必要があります。

重要な注意事項

- Elasticsearch には libcurl が必要です。詳細は要件 を参照してください。
- housekeeper は Elasticsearch からデータを削除しません。データの保持期間を制御するには、Index Lifecycle Management を参照してください。
- Zabbix は Elasticsearch 内でトレンドを計算または保存しません。古いデータを保持するには、history storage period を延長することを検討してください。
- Elasticsearch を使用する場合、データベースから値を取得する範囲クエリは、データ保存期間のタイムスタンプによって制限されます。
- Elasticsearch は Zabbix プロキシではサポートされていません。

Elasticsearch がまだインストールされていない場合は、続行する前に 公式インストールガイド を参照してください。

Elasticsearch の設定

履歴データを Elasticsearch に保存するには、以下の手順が必要です。

- 保存したい値の型ごとにインデックスを作成します。これは Elasticsearch がデータを保存する場所であり、リレーショナルデータベースのテーブルに相当します。
- 各インデックスにマッピングを定義します。これはデータの構造を定義し、テーブルスキーマに相当します。
- 保存前に値を処理するためのインジェストパイプラインを設定します（JSON 値や日付ベースのインデックスには必須です）。

Elasticsearch は、値の型ごとに単一のインデックスにデータを保存することも、複数の日付ベースのインデックスに分散して保存することもできます。両方のアプローチについて以下で説明します。

1 つのインデックスに履歴を保存する

この方法では、指定した値のタイプのすべての履歴データが 1 つのインデックス（例: `uint` や `text`）に書き込まれます。

数値（整数）の値のタイプのインデックスを作成するには、次のリクエスト（URL の/`uint` 付き）を Elasticsearch インスタンスに送信します。

```
curl -X PUT \
 http://localhost:9200/uint \
 -H 'content-type:application/json' \
 -d '{
     "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": { "type": "long" }
      }
   }
}'
```

Elasticsearch はインデックスが作成されたことを確認する応答を返します。

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Elasticsearch に保存したい追加の値のタイプごとに、同様のリクエストを送信する必要があります。

> **Note:**
> すべての値のタイプのマッピングはZabbix ソースリポジトリで利用できます。

たとえば、テキスト値のタイプのインデックスを作成するには:

```
curl -X PUT \
 http://localhost:9200/text \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

JSON 値タイプ

他の値タイプとは異なり、JSON 値は保存前に追加の処理が必要です。

以下のインデックスは、パース済み値と生値のために個別のフィールドを使用するため、各値を JSON としてパースし、正しいフィールドに保存するためのインジェストパイプラインが必要です。

JSON 値タイプのインデックスを作成するには、Elasticsearch インスタンスに次のリクエスト（URL に/json を指定）を送信します。

```
curl -X PUT \
 http://localhost:9200/json \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
    }
  }
}'
```

次に、インジェストパイプラインを作成します:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json \
 -H 'content-type:application/json' \
 -d '{
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "set": {
            "if": "ctx.value_parsed == null",
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ],
   "on_failure": [
      {
         "set": {
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ]
}'
```

Elasticsearch は、インジェストパイプラインが作成されたことを確認する応答を返します:

```
{"acknowledged": true}
```

日付ベースのインデックスに履歴を保存する

すべての履歴データを単一のインデックス（例: `uint`）に書き込む代わりに、Elasticsearch はこのデータを複数の日付ベースのインデックス（例: `uint-2026-01-01`、`uint-2026-01-02`）に分散できます。
これにより、時間の経過に伴うデータ量と保持期間をより簡単に管理できます。

これを有効にするには、次の操作が必要です。

- 保存する各値タイプごとに index template を作成します。これにより、Elasticsearch が新しい日付ベースのインデックスを自動作成する際に適用するマッピングが指定されます。
- 各値タイプごとに ingest pipeline を作成します。これは、受信した各値を処理し、正しい日付ベースのインデックスへ振り分けます。

インデックステンプレート

`text` インデックスのテンプレートを作成するには、以下の詳細でリクエストを送信します。

- Elasticsearch インスタンスの URL に `_template/text_template` を使用します。
- インデックス名に一致させるため、`"index_patterns"` フィールドに`"text*"` を使用します。
- `text` 値タイプのマッピングを使用します（Zabbix ソースリポジトリのマッピングを参照）。

```
curl -X PUT \
 http://localhost:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "text*" ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
```

```
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "format": "epoch_second", "type": "date" },
         "value": {
            "fields": {
               "analyzed": { "index": true, "type": "text", "analyzer": "standard" }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

json インデックスのテンプレート:

```
curl -X PUT \
 http://localhost:9200/_template/json_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [ "json*" ],
   "settings": {
      "number_of_shards": 5,
      "number_of_replicas": 1
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "itemid": { "type": "long" },
         "clock": { "type": "date", "format": "epoch_second" },
         "ns": { "type": "long" },
         "value_parsed": { "type": "flattened" },
         "value_raw": { "type": "keyword", "ignore_above": 1000000 }
      }
   }
}'
```

インジェストパイプライン

text インデックスのインジェストパイプラインを作成するには:

- Elasticsearch インスタンスの URL で _ingest/pipeline/text-pipeline を使用します。
- 各値をそのタイムスタンプに基づいて正しい日付ベースのインデックスにルーティングするために、date_index_nameプロセッサを含めます。

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/text-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily text index naming",
   "processors": [
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": ["UNIX"],
            "index_name_prefix": "text-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

json インデックスの場合、パイプラインは値を正しいインデックスにルーティングする前に JSON 値を解析する必要もあります:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json-pipeline \
```

```
 -H 'content-type:application/json' \
 -d '{
   "description": "daily json index naming"
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
            "ignore_failure": true
         }
      },
      {
         "script": {
            "source": "if (ctx.value_parsed == null || !(ctx.value_parsed instanceof Map)) { ctx.value_raw
         }
      },
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": [ "UNIX" ],
            "index_name_prefix": "json-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

Zabbix サーバーの設定

Zabbix サーバーの設定ファイル (zabbix_server.conf) で、HistoryProvider パラメーターを設定します。

たとえば、*Character*、*Log*、*Text*、*JSON* 型の値を Elasticsearch に保存し、*Numeric* 値はデータベースに保持するには、次のようにします。

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

日付ベースのインデックスを使用している場合は、パラメーターに date_index=1 を追加します。

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

変更後、Zabbix サーバーを再起動します。

```
systemctl restart zabbix-server
```

Zabbix Web インターフェースの設定

Zabbix のWeb インターフェース設定ファイル（zabbix.conf.php）で、$HISTORY_PROVIDERS 変数をサーバー設定に合わせて設定します。

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

トラブルシューティング

以下の手順で、Elasticsearch の設定に関する問題のトラブルシューティングに役立つ場合があります。

1. Elasticsearch または Zabbix サーバーのログでエラーを確認します。
2. 遅いクエリを特定するには、Zabbix サーバーの設定パラメータHistoryProvider で log_slow_queries オプションを使用します。
3. Elasticsearch が Zabbix サーバーおよび Zabbix Web インターフェースからのアクセスを許可していることを確認します。
4. auto_create_index が有効になっていることを確認します。

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"
```

```
#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

有効にするには、次のリクエストを送信します。

```
curl -X PUT \
 http://localhost:9200/_cluster/settings \
 -H 'content-type:application/json' \
 -d '{
   "persistent": {
      "action.auto_create_index": "true"
   }
}'

#### {"acknowledged": true, "persistent": {"action": {"auto_create_index": "true"} }, "transient": {} }
```

5. GET リクエストをそれぞれの URL に送信して、マッピング、テンプレート、および ingest pipeline が正しいことを確認します。

```
curl -X GET http://localhost:9200/json
curl -X GET http://localhost:9200/_template/json*
curl -X GET http://localhost:9200/_ingest/pipeline/json*
```

受信したレスポンスは、Elasticsearch API documentation の期待されるレスポンスと比較できます。

6. 失敗状態になっている shards がないか確認します。Elasticsearch を再起動すると解決する場合があります。
7. Elasticsearch にクエリを実行して、Zabbix によって収集されたデータが保存されているか確認します。たとえば、次のようにします。

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Elasticsearch の設定をリセットして最初からやり直す必要がある場合は、すべてのインデックス、テンプレート、および ingest pipeline を削除できます。

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

ClickHouse のセットアップ

Zabbix では、リレーショナルデータベースの代わりに、ClickHouseにヒストリデータを保存できます。

> **Note:**
> ClickHouse は、アプリケーションパフォーマンス監視（APM）のテレメトリデータソースとしても使用できます。詳細については、管理 > データソースを参照してください。

このガイドでは、ClickHouse のサポート対象バージョンのセットアップについて説明します。異なるバージョンを使用している場合、一部の機能が意図したとおりに動作しない可能性があります。

ClickHouse には、次の値の種類を保存できます。

アイテムの値の種類	データベーステーブル	ClickHouse の種類
数値（符号なし）	history_uint	uint
数値（浮動小数点）	history	dbl
文字	history_str	str
ログ	history_log	log
テキスト	history_text	text
バイナリ	history_bin	Zabbix ではサポートされていません

アイテムの値の種類	データベーステーブル	ClickHouse の種類
JSON	history_json	json

Note:
ClickHouse は JSON 配列を受け付けません。JSON 値は、単一のオブジェクトまたはオブジェクトの集合である必要があります。また、ClickHouse はNULL を含む JSON キーを、存在しないキーと同様に扱います。

重要な注意事項

- ハウスキーパーは、ClickHouse からデータを削除しません。データの保持期間を制御するには、ClickHouse でClickHouse データの保存期間を設定します。
- Zabbix は、ClickHouse でトレンドを計算または保存しません。古いデータを保持するには、履歴の保存期間を延長することを検討してください。
- 既存の Zabbix データベース（MySQL または PostgreSQL）から ClickHouse に履歴データを移行する場合は、ClickHouse スキーマおよび履歴の移行スクリプトを参照してください。
- ClickHouse は、Zabbix プロキシの履歴ストレージとしてはサポートされていません。ただし、APM を有効にした Zabbix プロキシのテレメトリデータソースとして使用できます。
- ClickHouse クラスターを使用している場合は、ClickHouse クラスターの設定を参照してください。

ClickHouse の設定

Zabbix のデータベースとユーザーを作成して設定し、データベーススキーマをインポートする必要があります。

このガイドでは、ClickHouse のDockerまたはパッケージによるインストール手順を説明します。

Docker

1. ClickHouse コンテナを実行する際に、Zabbix データベースとユーザーを作成および設定します:

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. ClickHouse が動作しており、接続できることを確認します:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"

#### 26.4.4.38
```

3. Zabbix ディレクトリにある history_all.sh スクリプトを使用して、データベーススキーマをインポートします:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
ClickHouse のデータ保存期間 (Time-To-Live, TTL) は、デフォルトで 31 日です。変更するには、データベーススキーマのインポート時に --ttl オプションを使用するか (例: 7 日間の場合は --ttl 604800)、後から設定 します。

4. テーブルが作成されたことを確認します:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

パッケージ

1. ClickHouse を起動します:

```
sudo service clickhouse-server start
```

2. Zabbix データベースとユーザーを作成して設定します:

```
clickhouse-client --user default

localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Zabbix ディレクトリにあるhistory_all.shスクリプトを使用して、データベーススキーマをインポートします:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
ClickHouse のデータ保存期間（Time-To-Live、TTL）は、デフォルトで 31 日です。変更するには、データベーススキーマのインポート時に--ttl オプションを使用するか（例: 7 日間の場合は--ttl 604800）、後から設定します。

4. テーブルが作成されたことを確認します:

```
clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Zabbix サーバーの設定

Zabbix サーバーの設定ファイル (zabbix_server.conf) で、HistoryProvider パラメーターを設定します。

たとえば、ClickHouse にサポートされているすべての値のタイプを保存するには、次のようにします。

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

変更を行った後、Zabbix サーバーを再起動します。

```
systemctl restart zabbix-server
```

Zabbix Web インターフェースの設定

Zabbix のWeb インターフェース設定ファイル（zabbix.conf.php）で、$HISTORY_PROVIDERS 変数をサーバーの設定に合わせて設定します。

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

追加設定

以下の手順は任意です。基本的なセットアップでは必要ありません。

ClickHouse のデータ保存期間の設定

ClickHouse のデータ保存期間（Time-To-Live、TTL）は、デフォルトで 31 日です。
変更するには、以下のコマンドを実行します。

以下の例では Docker を使用しています。
ClickHouse をパッケージでインストールした場合は、ClickHouse クライアントでクエリを直接実行してください。

1. テーブルを変更します（history_json と 3600 を必要な値に置き換えてください）：

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. 変更をすぐに適用します:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. データ保存期間が変更されたことを確認します:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"

#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. *Administration* > *Housekeeping* でデータ保存期間を更新するため、Zabbix サーバーを再起動します:

```
systemctl restart zabbix-server
```

ClickHouse クラスターの設定

このセクションでは、レプリケーションとシャーディングを備えた ClickHouse クラスターの設定手順を説明します。

以下の手順では、Docker と ClickHouse ドキュメントにある ClickHouse cluster_2S_2R（ClickHouse Keeper を使用して ClickHouse 26.4.4.38 を実行）を使用します。

開始する前に、Zabbix 用の ClickHouse クラスターの前面にロードバランサーを配置することを推奨します。
Zabbix サーバーおよび Web インターフェースの設定は単一の ClickHouse URL のみをサポートするため、1 つのノードにしか接続できません。
そのノードが停止した場合でも、ロードバランサーが同じ URL を通じて、正常な別のノードへ Zabbix を振り分けます。

1. Zabbix データベースを作成します:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

Attention:
ON CLUSTER 句は、コマンド実行時にクラスターの一部であるノードにのみ適用されます。
後でクラスターにノードを追加した場合は、このコマンドを再度実行してください。

2. Zabbix ユーザーを作成して設定します:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Zabbix ディレクトリにある history_all.sh スクリプトを使用して、データベーススキーマをインポートします:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
ClickHouse のデータ保存期間（Time-To-Live、TTL）は、デフォルトで 31 日です。
変更するには、データベーススキーマのインポート時に --ttl オプションを使用するか（例: 7 日間なら --ttl 604800）、後から設定してください。

Note:
ClickHouse クラスターで明示的なレプリカパスが必要な場合（たとえば、ClickHouse サーバー設定でレプリケートテーブルのデフォルトパスが設定されていない場合）は、スキーマスクリプト（history_schema.sh、history_uint_schema.sh など）を個別に実行し、各テーブルに異なるパスを設定してください:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

ClickHouse のレプリケーションの詳細については、ClickHouse ドキュメントの Replicating data および Replicated* table engines を参照してください。

4. 各テーブル種別に対してシャーディングを設定します。
テーブル名を変更し、元の名前を持つ Distributed テーブルを作成して、変更後のテーブルを参照させます。
この Distributed テーブルが、Zabbix が実際に読み書きする対象です。テーブルはデータを自動的に各シャードへ分散します。
たとえば、history テーブルの場合:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
ClickHouse のシャーディングの詳細については、ClickHouse ドキュメントの Distributed table engine を参照してください。

5. Zabbix サーバーを設定します。

6. Zabbix Web インターフェースを設定します。

7. Zabbix データが各シャードに分散されていることを確認します。
zabbix.history_local の行数は、Zabbix サーバーが継続的に新しいデータを挿入するため、繰り返し確認すると変化する場合があります。
各ノードで次のコマンドを実行します:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. レプリケーションが同期していることを確認します（active_replicas は total_replicas と一致する必要があります）:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

トラブルシューティング

以下の手順は、ClickHouse のセットアップで問題をトラブルシュートするのに役立つ場合があります。

1. ClickHouse または Zabbix サーバーのログでエラーを確認します。
2. 遅いクエリを特定するには、Zabbix サーバーの設定パラメーターHistoryProvider で log_slow_queries オプションを使用します。
3. ClickHouse が Zabbix サーバーおよび Zabbix Web インターフェースからのアクセスを許可していることを確認します。
4. ClickHouse にクエリを実行して、Zabbix によって収集されたデータが保存されているか確認します。例:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Zabbix 用 **NGINX** のセットアップに関するディストリビューション固有の注意事項

RHEL

RHEL 8、9、10 では、NGINX は AppStream に含まれているため、追加のリポジトリは必要ありません。次のコマンドで有効化してインストールします。

```
dnf module list nginx
dnf -y install nginx
```

RHEL 7 では、CentOS を使用している場合、次のコマンドで CentOS 7 Extras リポジトリから NGINX をインストールできます。

```
yum -y install nginx
```

最新のアップストリームビルドを使用する場合は、公式 NGINX リポジトリを `/etc/yum.repos.d/nginx.repo` に設定します。

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

SUSE Linux Enterprise Server 15 では、`php-fpm` を設定する必要があります（設定ファイルのパスはサービスパックによって若干異なる場合があります）：

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

Microsoft Entra ID を使用した **SAML** の設定

概要

このセクションでは、SAML 2.0 認証を使用して Microsoft Entra ID（旧 Microsoft Azure Active Directory）から Zabbix へのシングルサインオンおよびユーザーのプロビジョニングを設定するためのガイドラインを提供します。

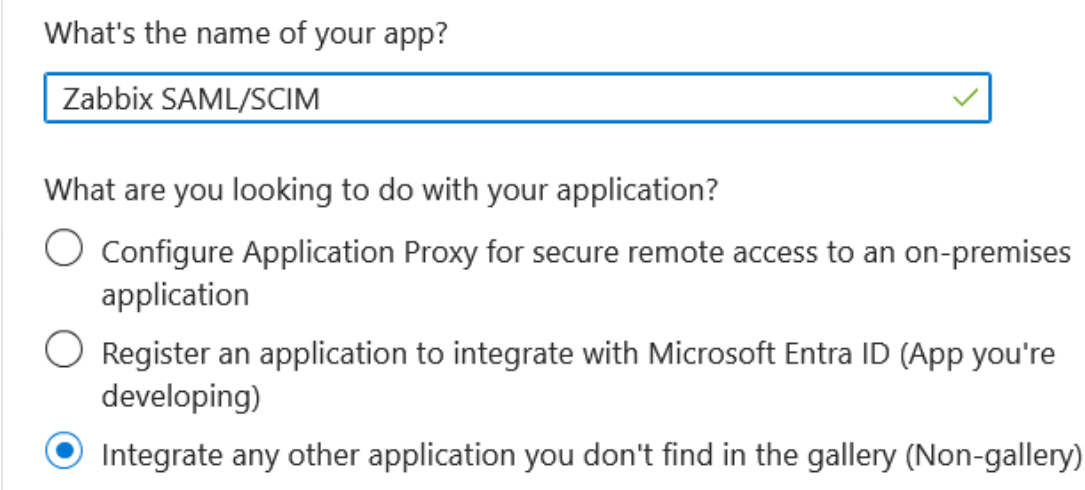
Microsoft Entra ID の設定

アプリケーションの作成

1. Microsoft Entra ID で Microsoft Entra admin center にログインします。テスト目的で、Microsoft Entra ID に無料試用アカウントを作成できます。

2. Microsoft Entra admin center で、*Applications* > *Enterprise applications* > *New application* > *Create your own application* を選択します。

3. アプリの名前を追加し、*Integrate any other application...* オプションを選択します。その後、*Create* をクリックします。

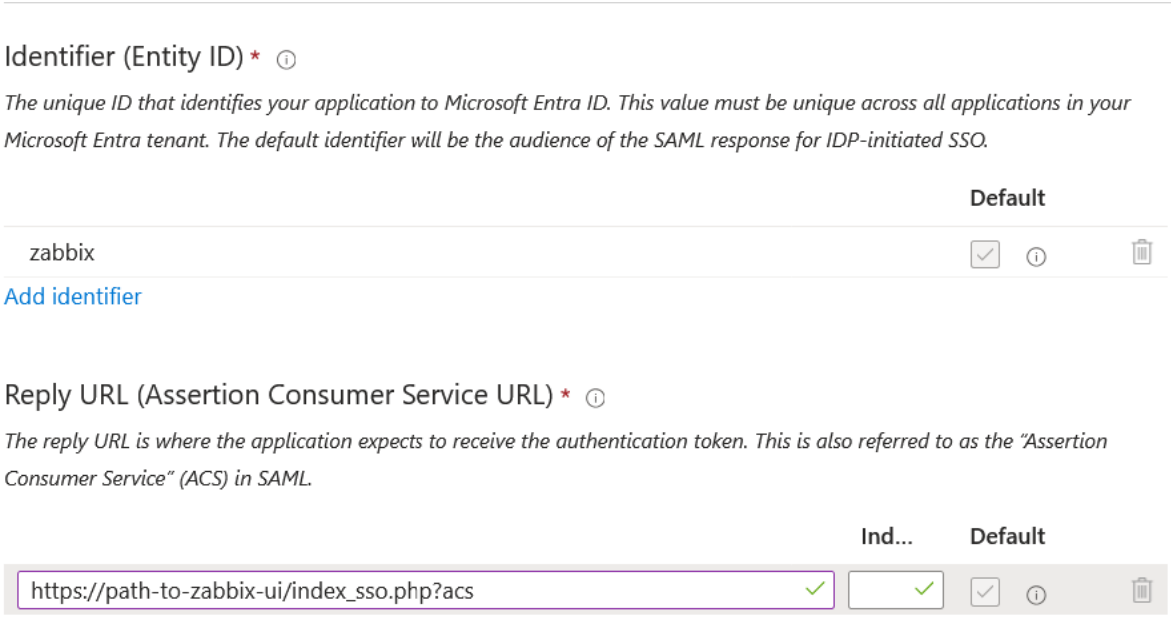


シングルサインオンの設定

1. アプリケーションページで *Set up single sign on* に移動し、*Get started* をクリックします。次に、*SAML* を選択します。

2. *Basic SAML Configuration* を編集します。

- *Identifier (Entity ID)* に、Microsoft Entra ID でアプリを識別するための一意の名前を設定します。例: `zabbix`
- *Reply URL (Assertion Consumer Service URL)* に、Zabbix のシングルサインオンエンドポイントを設定します: `https://<path-to-zabbix-ui>/index_sso.php?acs`



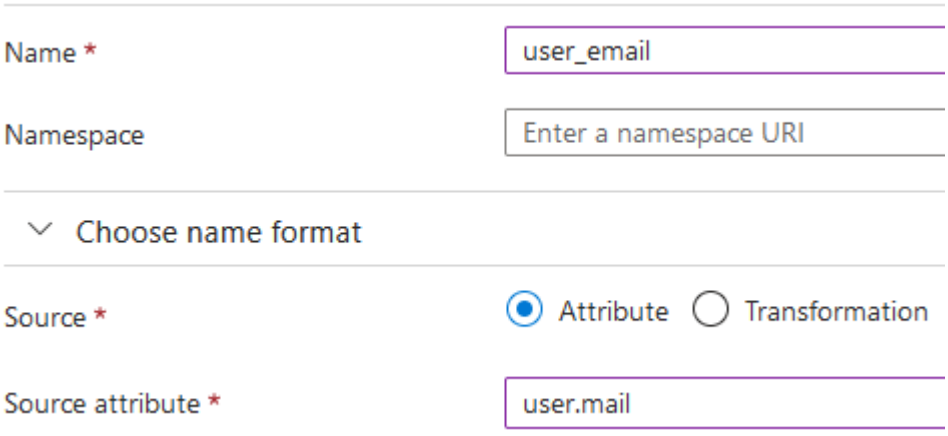
`https` が必須であることに注意してください。Zabbix でこれを使用するには、Web インターフェース設定ファイルに次の行を追加します。

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

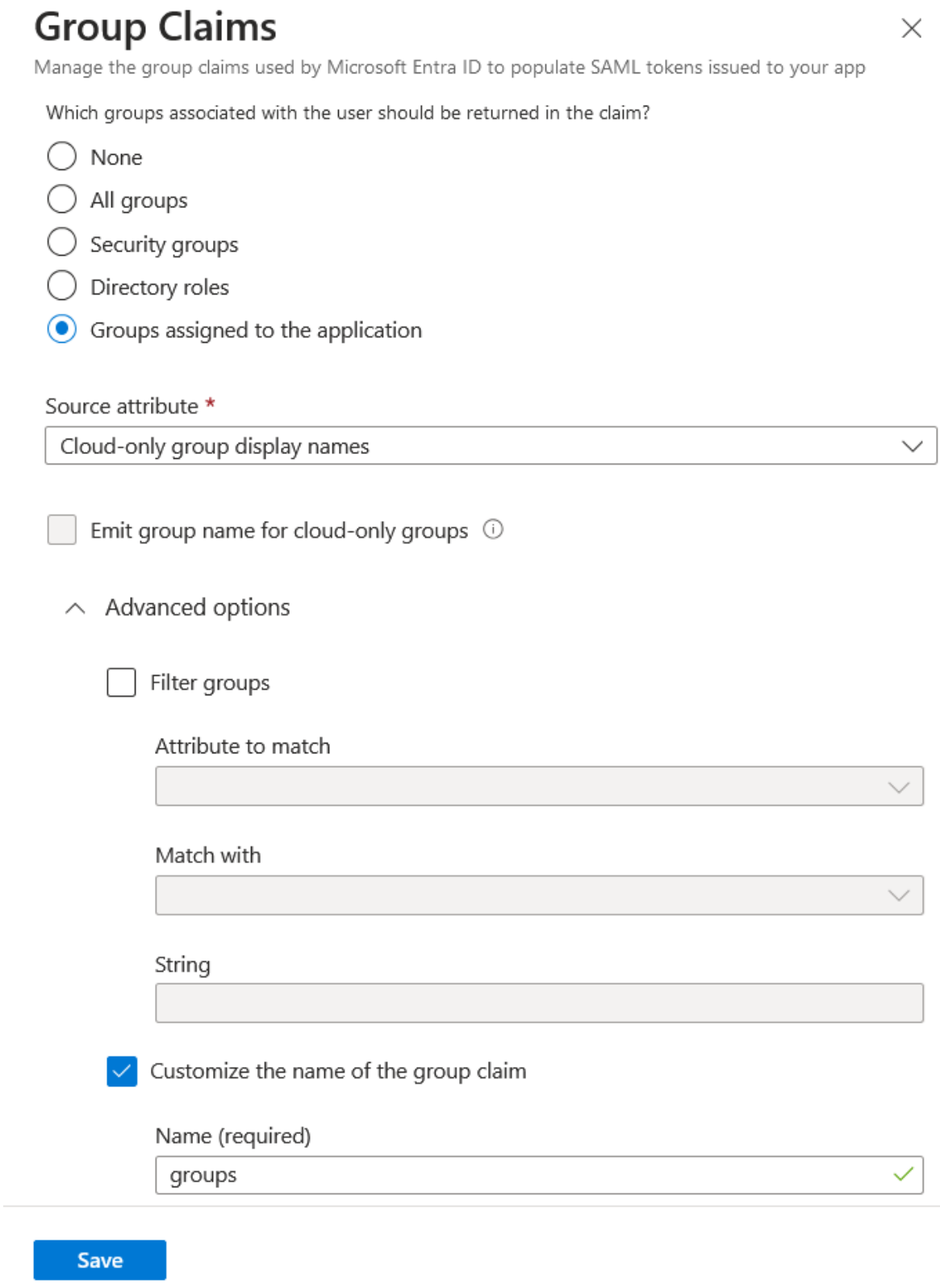
3. *Attributes & Claims* を編集します。Zabbix に渡すすべての属性（user_name、user_lastname、user_email、user_mobile、groups）を追加する必要があります。

属性名は任意です。異なる属性名を使用できますが、Zabbix の SAML 設定における対応するフィールド値と一致している必要があります。

- *Add new claim* をクリックして属性を追加します。



- *Add a group claim* をクリックして、Zabbix にグループを渡すための属性を追加します。



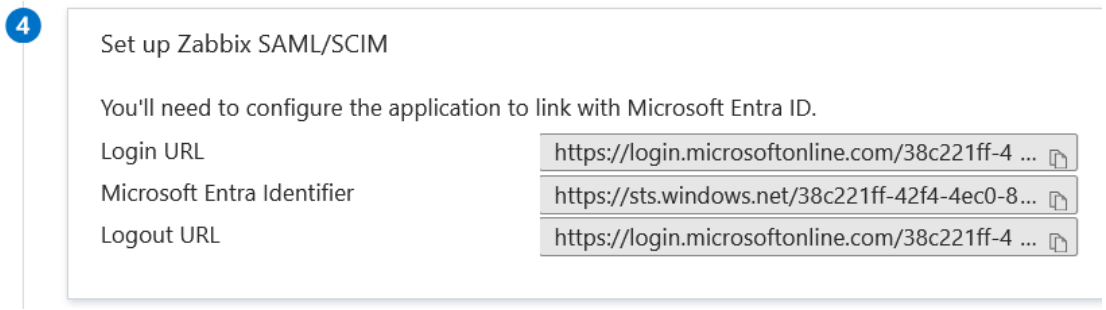
このクレームでは、選択した *Source attribute* によって、グループ ID ではなくグループ名が Zabbix に渡されることが重要です。そうしないと、JIT ユーザープロビジョニングが正しく機能しません。

4. *SAML Certificates* で、Entra ID が提供する Base64 証明書をダウンロードし、Zabbix の Web インターフェースのインストール先にある `conf/certs` に配置します（場所は、Zabbix のインストール元や OS によって異なる場合があります）。

次のコマンドを実行して、権限を 644 に設定します。

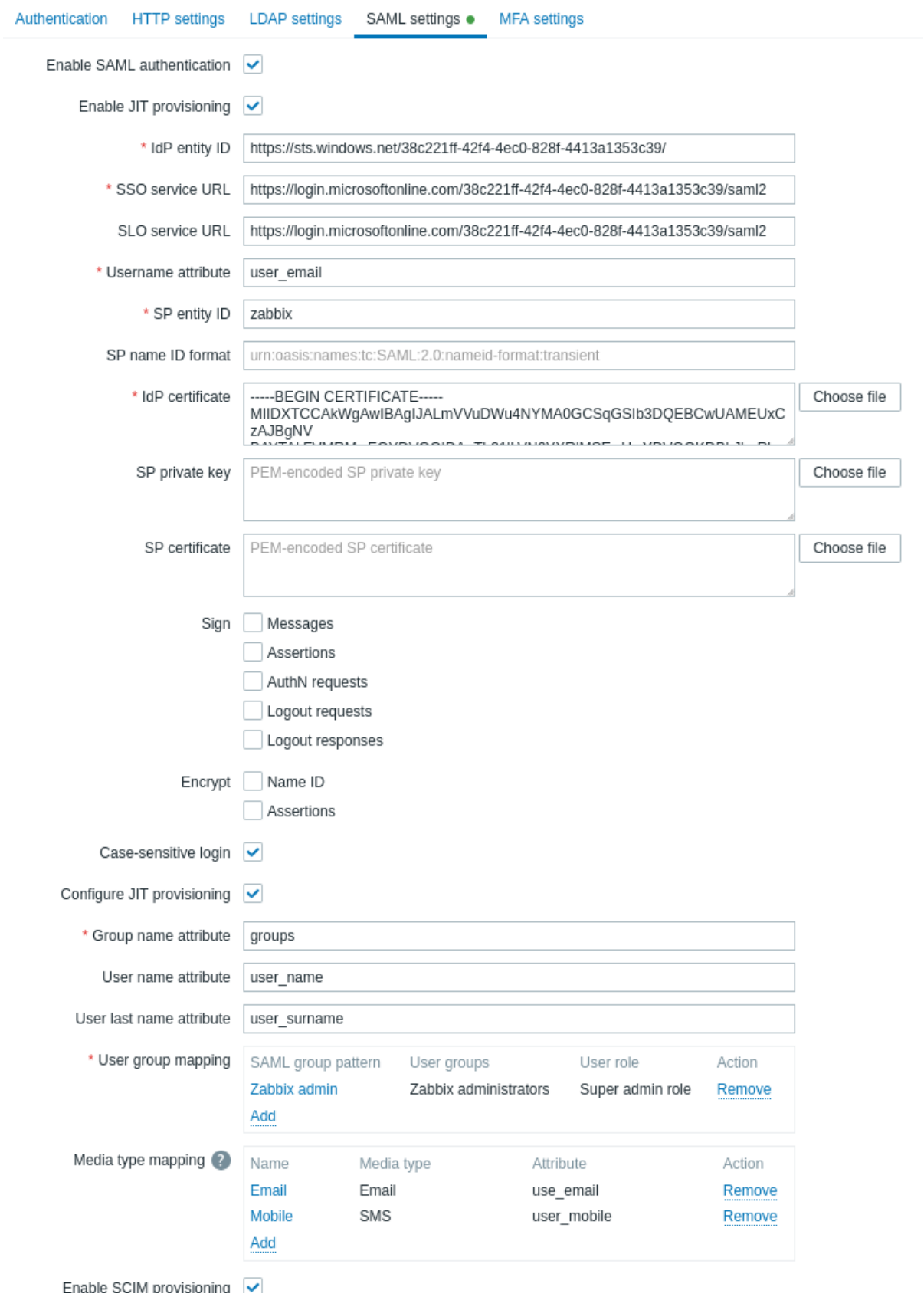
```
chmod 644 entra.cer
```

5. Entra ID の *Set up <your app name>* に表示される値を使用して、Zabbix の SAML 認証を設定します（次のセクションを参照）。



Zabbix の設定

1. Zabbix で、SAML 設定に移動し、Entra ID の設定に基づいて設定オプションを入力します。



Zabbix のフィールド	Entra ID の設定フィールド	サンプル値
IdP エンティティ *ID*	Microsoft Entra 識別子	
SSO サービス *URL*	ログイン URL	
SLO サービス *URL*	ログアウト URL	
SP エンティティ *ID*	識別子（エンティティ ID）	
ユーザー名属性	カスタム属性（クレーム）	`user_email`
グループ名属性	カスタム属性（クレーム）	`groups`
ユーザー名属性	カスタム属性（クレーム）	`user_name`

Zabbix のフィールド	Entra ID の設定フィールド	サンプル値
ユーザー姓属性	カスタム属性（クレーム）	`user_lastname`

2. Entra ID から提供された Base64 証明書を追加します。

Web インターフェース設定ファイル（`zabbix.conf.php`）で `$SSO['CERT_STORAGE'] = 'database'` が設定されている場合、SAML 設定中に Web インターフェースで証明書のテキストを貼り付けるか、証明書ファイルをアップロードできます。ファイルシステム上にファイルを用意する必要はありません。

設定ファイルで `$SSO['CERT_STORAGE'] = 'file'` が設定されている場合、証明書をファイルシステム上（デフォルトでは `ui/conf/certs`、または設定ファイルで設定されたパス）で利用できるようにする必要があります。また、Web インターフェースは証明書をデータベースに保存しません。
`$SSO['CERT_STORAGE']` が設定されていないかコメントアウトされている場合は、ファイルストレージが使用され、証明書は `ui/conf/certs` から読み込まれることに注意してください。

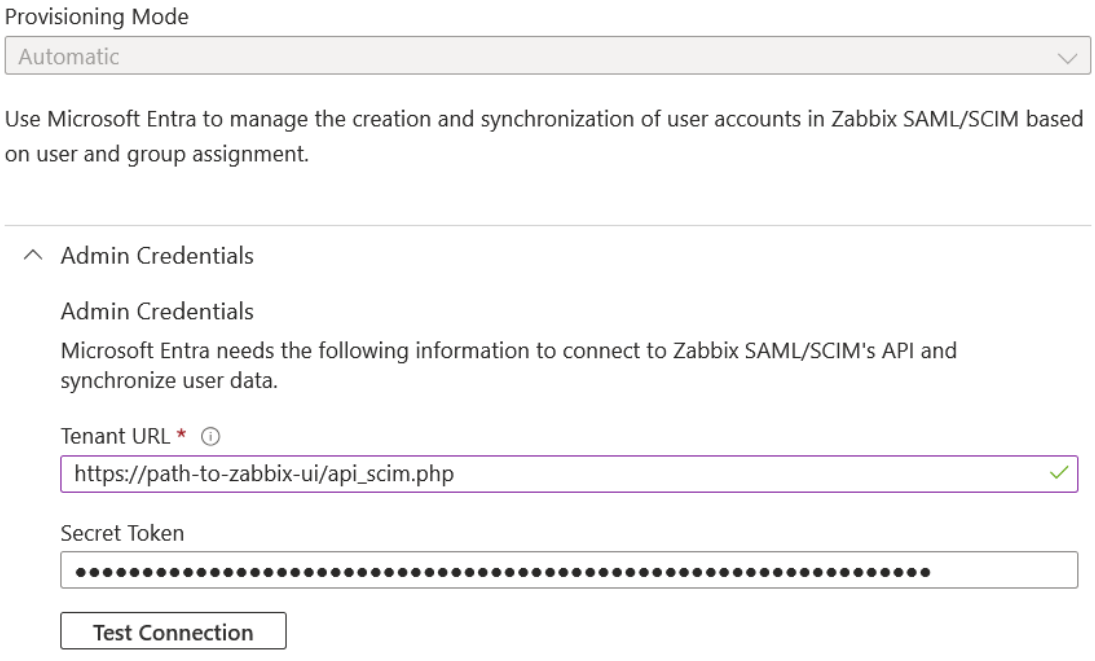
ユーザーグループのマッピングも設定する必要があります。
メディアのマッピングは任意です。

3. 更新ボタンをクリックして、これらの設定を保存します。

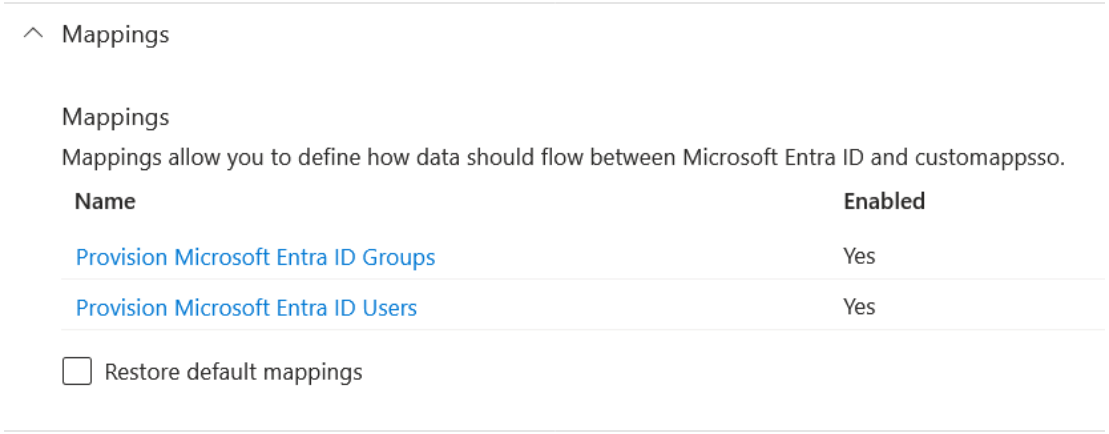
SCIM ユーザープロビジョニング

1. Entra ID アプリケーションページで、メインメニューから Provisioning ページを開きます。
Get started をクリックし、Automatic provisioning mode を選択します。

- *Tenant URL* に次の値を設定します: `https://<path-to-zabbix-ui>/api_scim.php`
- *Secret token* に、Super admin 権限を持つ Zabbix API トークンを入力します。
- 接続が確立されているか確認するには、*Test connection* をクリックします。

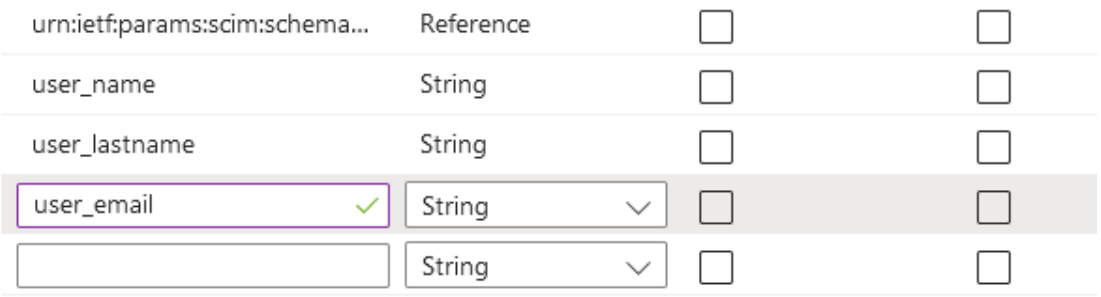


2. 次に、SCIM で Zabbix に渡されるすべての属性を追加できます。
そのためには、*Mappings* をクリックし、続いて *Provision Microsoft Entra ID Users* をクリックします。



Attribute Mapping リストの下部で *Show advanced options* を有効にし、*Edit attribute list for customappsso* をクリックします。

属性リストの下部に、型が'String' の独自の属性を追加します。



リストを保存します。

3. 次に、追加した属性のマッピングを追加できます。
Attribute Mapping リストの下部で *Add New Mapping* をクリックし、以下のようにマッピングを作成します。



すべてのマッピングを追加したら、マッピングのリストを保存します。



4. Zabbix へのユーザープロビジョニングの前提条件として、Entra ID でユーザーとグループを設定しておく必要があります。

そのためには、*Microsoft Entra admin center* に移動し、各 Users および Groups ページでユーザー/グループを追加します。

5. Entra ID でユーザーとグループを作成したら、アプリケーションの *Users and groups* メニューに移動し、それらをアプリに追加できます。

6. アプリの *Provisioning* メニューに移動し、*Start provisioning* をクリックして、ユーザーを Zabbix にプロビジョニングします。

Entra ID の Users PATCH リクエストは、メディアの変更をサポートしていないことに注意してください。

認証リクエストの署名

Entra ID で、署名付き認証リクエストの署名を検証するように設定できます。

これを機能させるには、公開鍵と秘密鍵を作成します。

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

権限を設定します。

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

次の設定を追加して、Zabbix のWeb インターフェース設定を更新します。

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

トラブルシューティング

SAML を使用して Zabbix にログインしようとするユーザーが、すでに Microsoft Edge のプロファイルにログインしている場合、Microsoft Edge ブラウザーで認証の問題が発生することがあります。このような問題の兆候として、ユーザーが Microsoft Edge のプライベートモードを使用すると Zabbix にログインできる場合があります。

この場合、認証の問題を回避するには、Zabbix のWeb インターフェース設定ファイルで `requestedAuthnContext` を `false` に設定する必要があります。

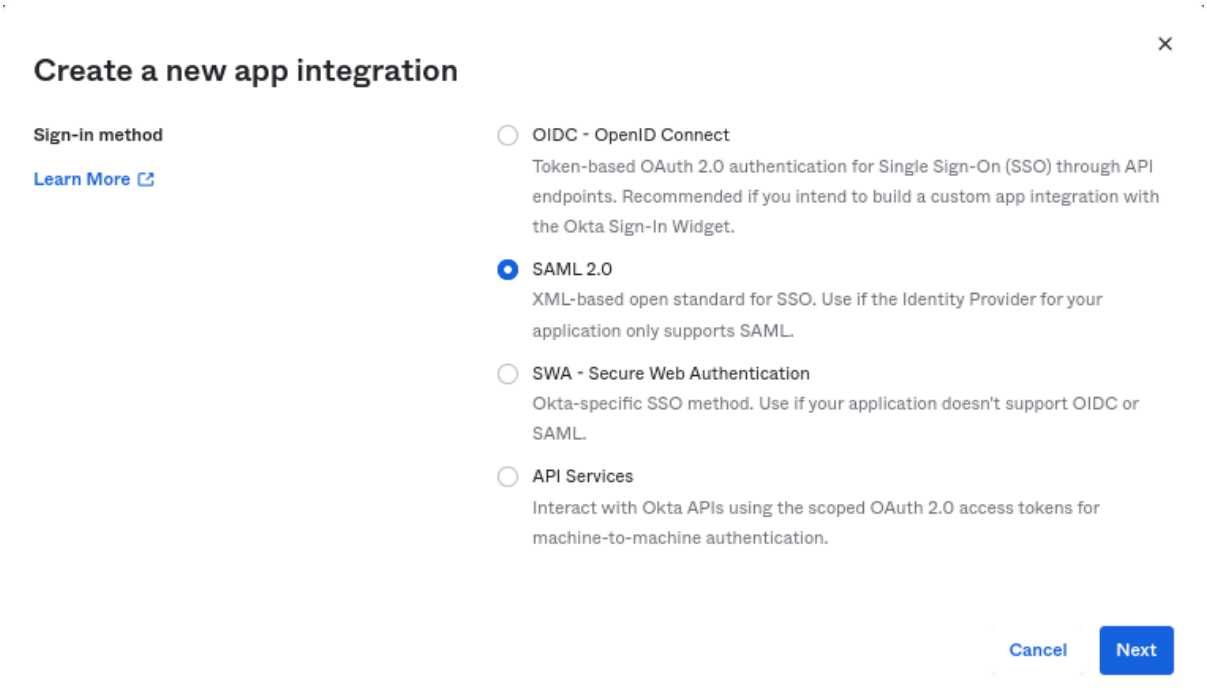
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

Okta を使用した **SAML** の設定

このセクションでは、Zabbix で SAML 2.0 認証とユーザーのプロビジョニングを有効にするために、Okta を設定するためのガイドラインを説明します。

Okta の設定

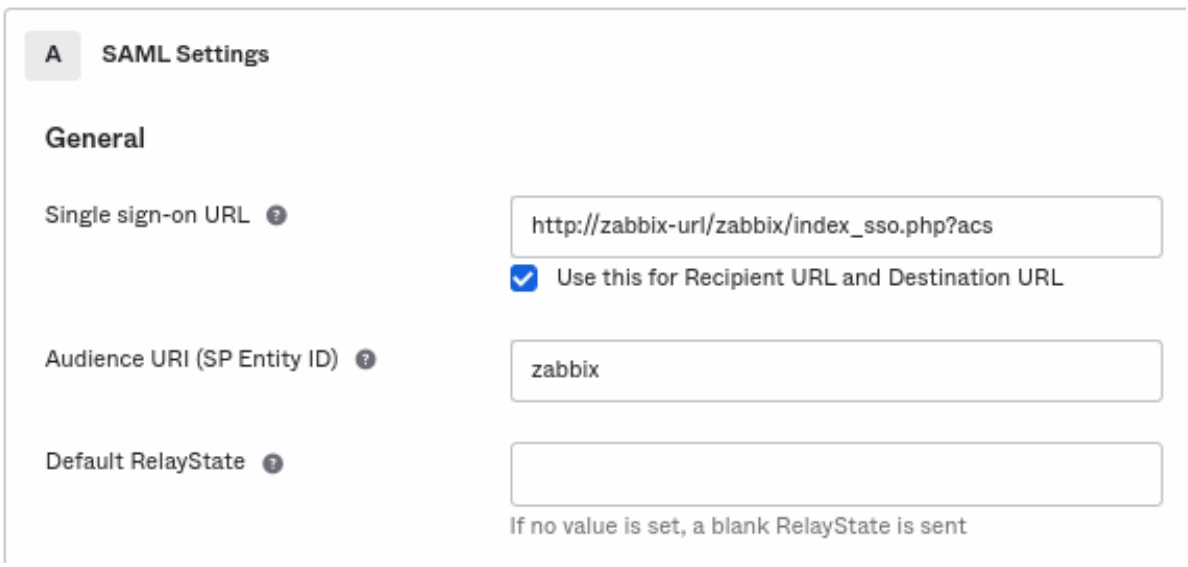
1. https://developer.okta.com/signup/ にアクセスし、アカウントを登録またはサインインします。
2. Okta の Web インターフェースで *Applications* > *Applications* に移動します。
3. *Create App Integration* をクリックします。



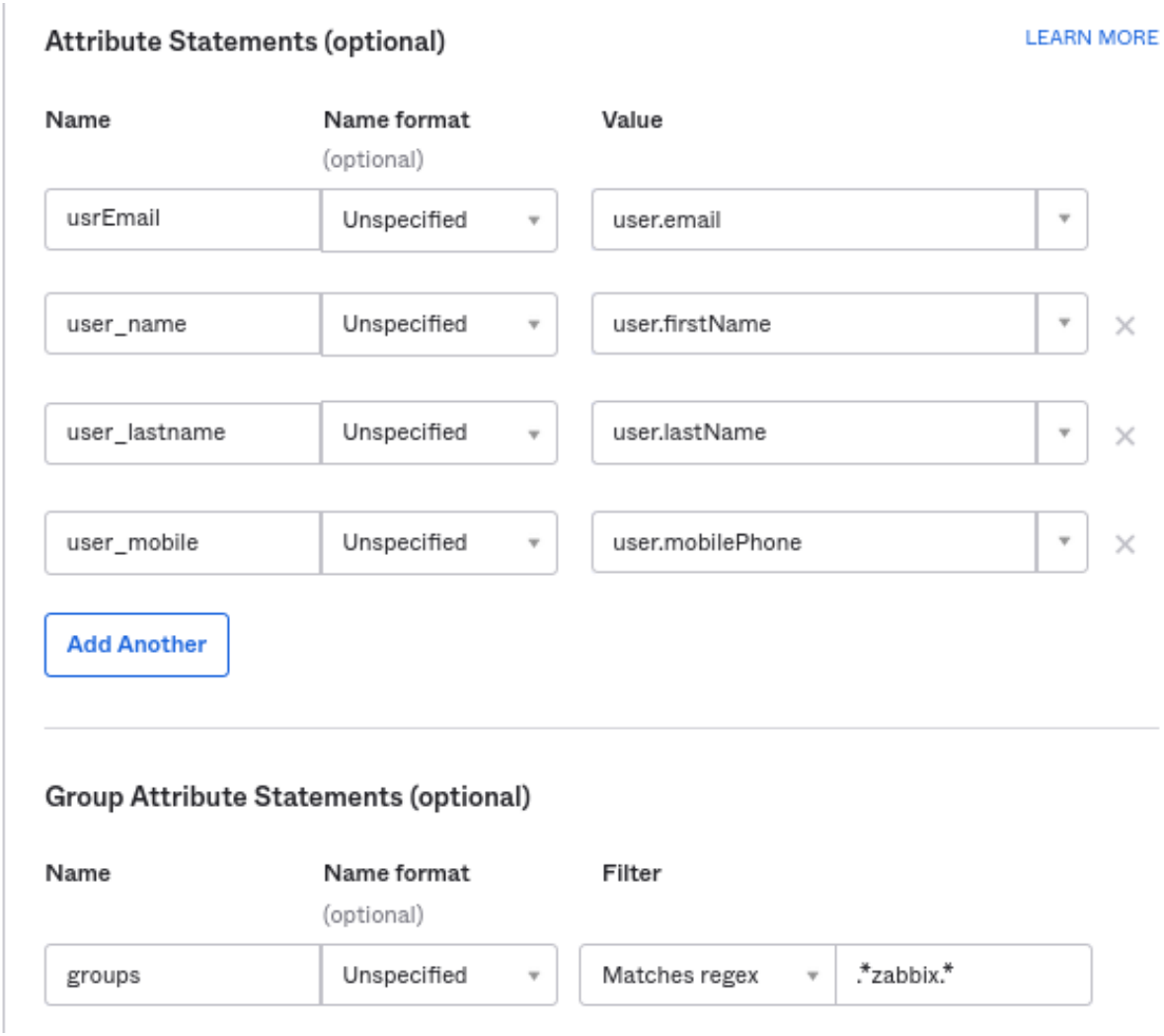
サインイン方法として”SAML 2.0” を選択し、*Next* をクリックします。

4. 一般設定でアプリ名を入力し、*Next* をクリックします。

5. SAML 設定で、以下の値を入力してから *Next* をクリックします。



- **General** で次を追加します:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` `https` ではなく `http` を使用してください。これにより、リクエスト内で `acs` パラメータが切り取られません。*Use this for Recipient URL and Destination URL* のチェックボックスもオンにしてください。
  - *Audience URI (SP Entity ID)*: `zabbix<br>` この値は、SAML アサーション内で一意のサービスプロバイダー識別子として使用されます。一致しない場合、操作は拒否されます。このフィールドには URL または任意の文字列を指定できます。
  - *Default RelayState*:
 このフィールドは空欄のままにしてください。カスタムリダイレクトが必要な場合は、Zabbix の *Users* > *Users* 設定で追加できます。
  - その他のフィールドは必要に応じて入力してください。
- **Attribute Statements/Group Attribute Statements** で次を追加します:



これらの属性ステートメントは、Zabbix と共有される SAML アサーションに挿入されます。

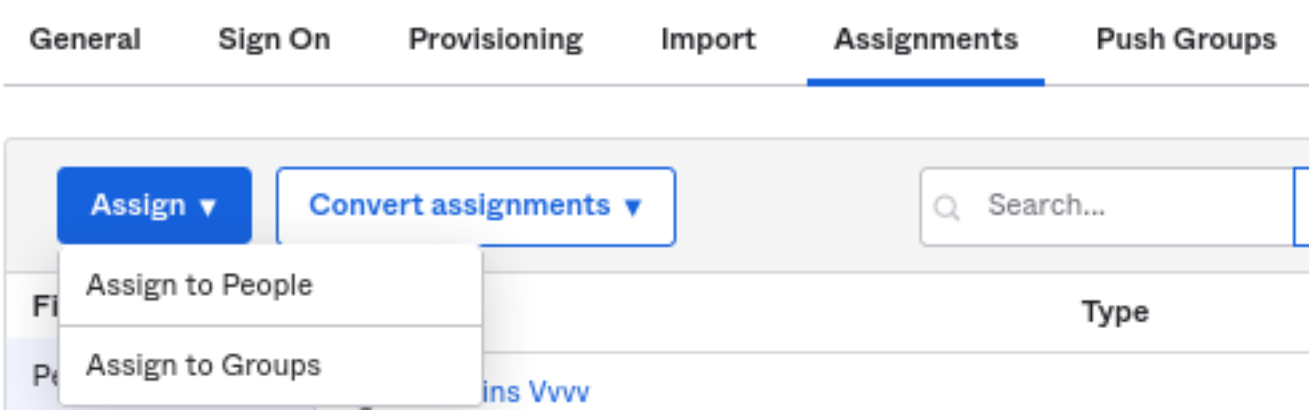
ここで使用している属性名は任意の例です。別の属性名を使用してもかまいませんが、Zabbix の SAML 設定にある各フィールド値と一致している必要があります。

JIT ユーザープロビジョニングを使わずに Zabbix への SAML サインインを設定する場合は、email 属性のみが必要です。

Note:
暗号化接続を使用する予定がある場合は、秘密鍵と公開鍵の暗号化証明書を生成し、公開証明書を Okta にアップロードしてください。証明書のアップロードフォームは、*Assertion Encryption* を”Encrypted” に設定すると表示されます（このパラメータを見つけるには *Show Advanced Settings* をクリックします）。

6. 次のタブで”I’m a software vendor. I’d like to integrate my app with Okta” を選択し、*Finish* をクリックします。

7. 新しく作成したアプリケーションの”Assignments” タブに移動し、*Assign* ボタンをクリックしてから、ドロップダウンから *Assign to People* を選択します。



8. 表示されるポップアップで、SAML 2.0 を使用して Zabbix に認証するユーザーにアプリを割り当て、*Save and go back* をクリックします。

9. IdP 証明書をダウンロードします。次のコマンドを実行して、権限を 644 に設定します:

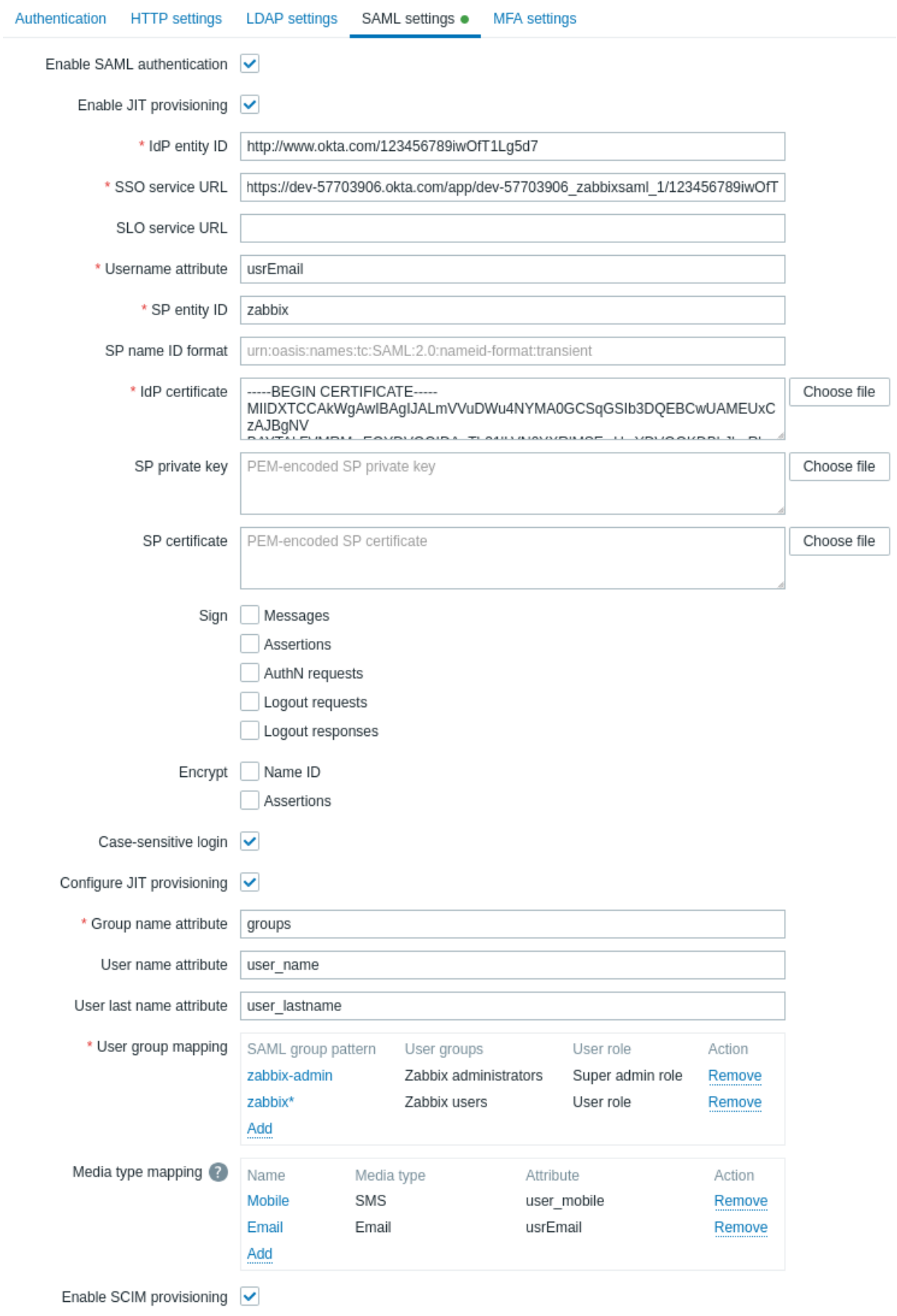
```
chmod 644 idp.crt
```

10. ”Sign On” タブに移動し、*View Setup Instructions* ボタンをクリックします。

セットアップの手順が新しいタブで開きます。Zabbix の設定中は、このタブを開いたままにしてください。

Zabbix の設定

1. Zabbix で、SAML 設定に移動し、Okta のセットアップ手順に基づいて設定オプションを入力します。



Zabbix のフィールド	Okta のセットアップフィールド	サンプル値
IdP エンティティ *ID*	Identity Provider Issuer	
SSO サービス *URL*	Identity Provider Single Sign-On URL	
ユーザー名属性	Attribute name	`usrEmail`
SP エンティティ *ID*	Audience URI	`zabbix`
グループ名属性	Attribute name	`groups`

Zabbix のフィールド	Okta のセットアップフィールド	サンプル値
ユーザー名属性	Attribute name	`user_name`
ユーザー姓属性	Attribute name	`user_lastname`

ユーザーグループとメディアのマッピングも設定する必要があります。

2. Okta の SAML セットアップ手順で提供された Base64 証明書を追加します。

Web インターフェース設定ファイル（`zabbix.conf.php`）で `$SSO['CERT_STORAGE'] = 'database'`が設定されている場合は、SAML 設定中に証明書のテキストを貼り付けるか、証明書ファイルを Web インターフェースからアップロードできます。ファイルシステム上にファイルを用意する必要はありません。

設定ファイルで `$SSO['CERT_STORAGE'] = 'file'`が設定されている場合、証明書をファイルシステム上（デフォルトでは `ui/conf/certs`、または設定ファイルで設定したパス）で利用できるようにする必要があります。この場合、Web インターフェースは証明書をデータベースに保存しません。`$SSO['CERT_STORAGE']` が設定されていないかコメントアウトされている場合は、ファイルストレージが使用され、証明書は `ui/conf/certs` から読み込まれることに注意してください。

3. Okta で *Assertion Encryption* が「Encrypted」に設定されている場合は、Zabbix でも *Encrypt* パラメータの「Assertions」チェックボックスを選択する必要があります。

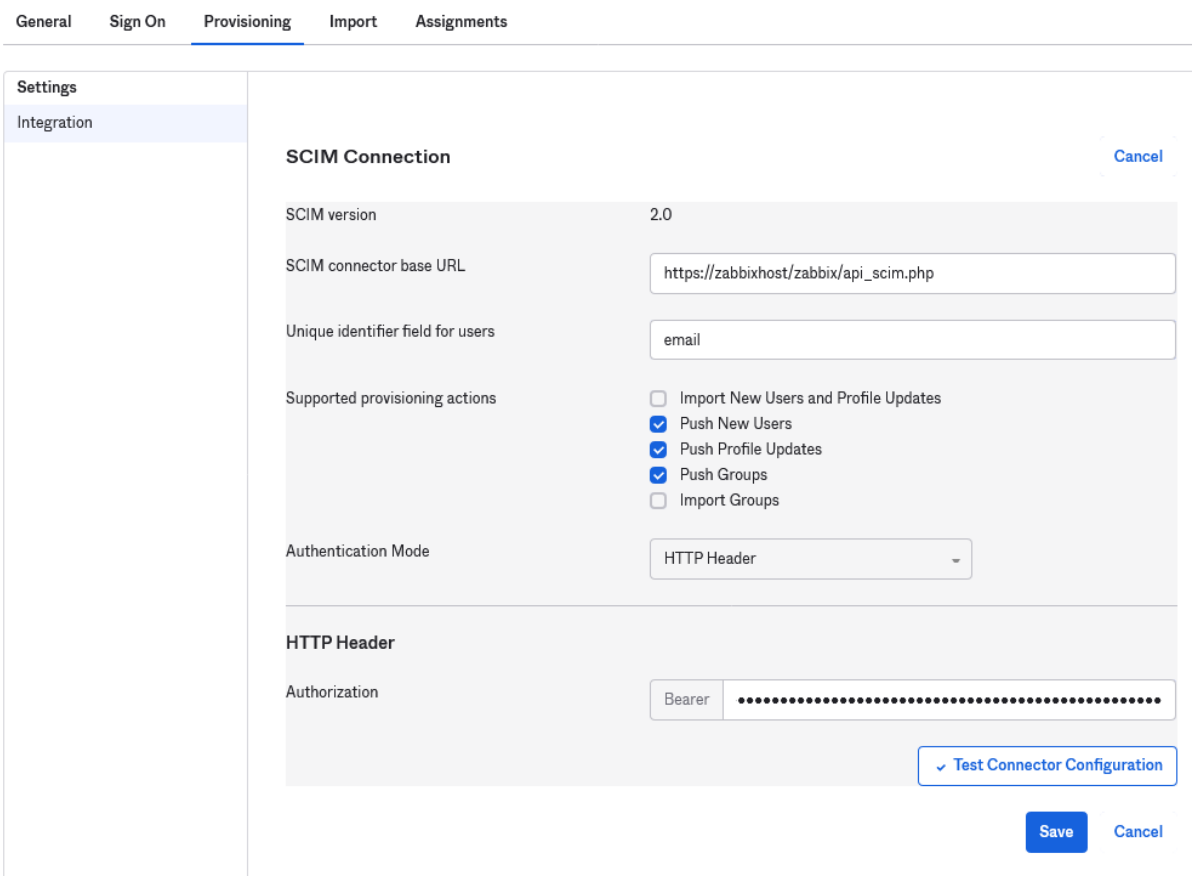
4. *Update* ボタンをクリックして、これらの設定を保存します。

SCIM プロビジョニング

1. SCIM プロビジョニングを有効にするには、Okta でアプリケーションの *General* > *App Settings* に移動します。

Enable SCIM provisioning チェックボックスを選択します。その結果、新しい *Provisioning* タブが表示されます。

2. SCIM 接続を設定するには、「Provisioning」タブに移動します。- *SCIM connector base URL* に、Zabbix の Web インターフェースへのパスを指定し、末尾に `api_scim.php` を追加します。例:
 `https://<your-zabbix-url>/zabbix/api_scim.php` - *Unique identifier field for users*: `email` - *Authentication mode*: `HTTP header` - *Authorization* に、特権管理者権限を持つ有効な API トークンを入力します。



Attention:
認証に問題がある場合は、Authorization header forwardingを参照してください。

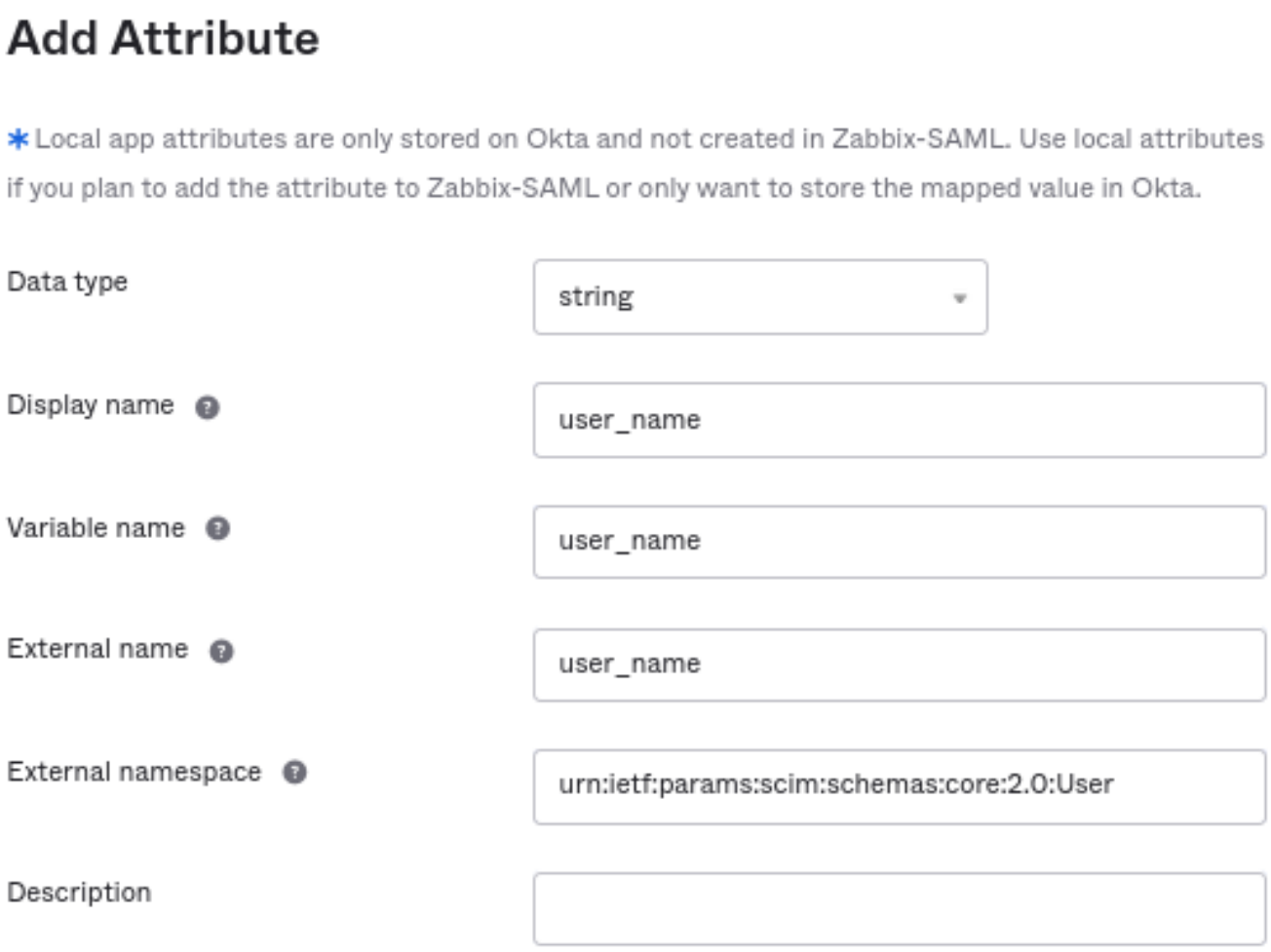
3. *Test Connector Configuration* をクリックして接続をテストします。すべてが正しく設定されている場合は、成功メッセージが表示されます。

4. *Provisioning* > *To App* で、次のチェックボックスが選択されていることを確認します。- Create Users - Update User Attributes - Deactivate Users

これにより、これらのリクエストタイプが Zabbix に送信されます。

5. SAML で定義されているすべての属性が SCIM でも定義されていることを確認します。アプリケーションのプロファイルエディターには、*Provisioning* > *To App* で *Go to Profile Editor* をクリックしてアクセスできます。

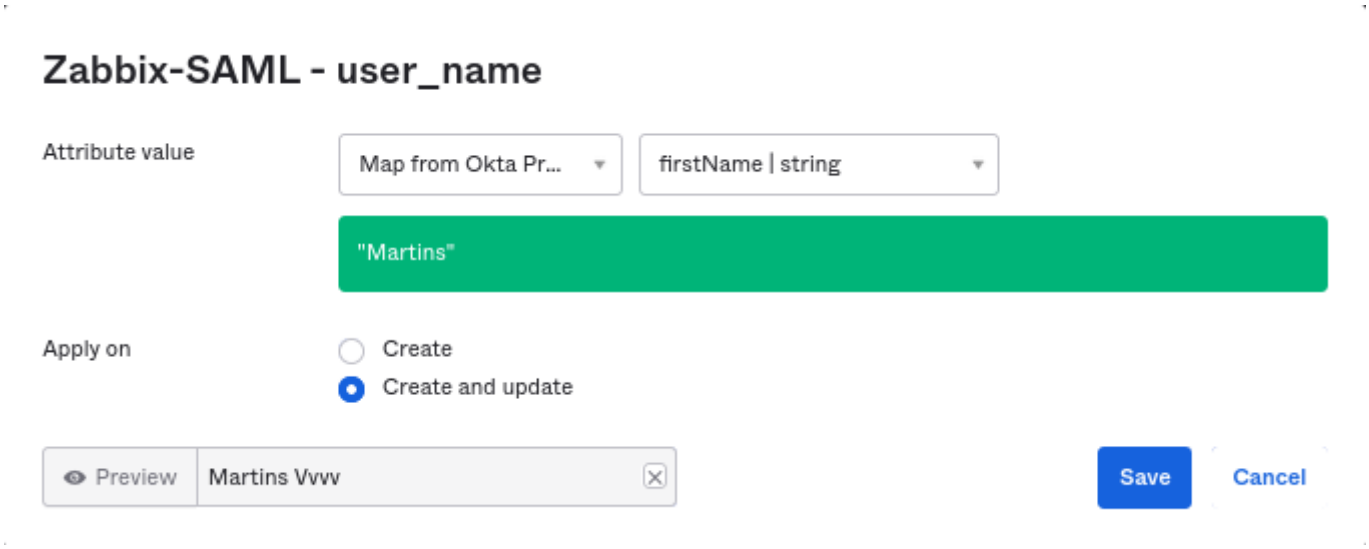
Add Attribute をクリックします。*Display name*、*Variable name*、*External name* に、SAML 属性名を入力します。例: `user_name`



External namespace はユーザースキーマと同じにする必要があります: `urn:ietf:params:scim:schemas:core:2.0:User`

6. アプリケーションの *Provisioning* > *To App* > *Attribute Mappings* に移動します。下部にある *Show Unmapped Attributes* をクリックします。新しく追加した属性が表示されます。

7. 追加した各属性をマッピングします。



8. 「Assignments」タブでユーザーを追加します。事前に、*Directory* > *People* でユーザーを追加しておく必要があります。これらの割り当てはすべて、リクエストとして Zabbix に送信されます。

9. 「Push Groups」タブでグループを追加します。Zabbix の SAML 設定におけるユーザーグループのマッピングパターンは、ここで指定するグループと一致している必要があります。一致しない場合、ユーザーを Zabbix で作成できません。

グループメンバーに関する情報は、変更が行われるたびに送信されます。

OneLogin を使用した **SAML** の設定

概要

このセクションでは、SAML 2.0 認証を使用してOneLoginから Zabbix へのシングルサインオンおよびユーザーのプロビジョニングを設定するためのガイドラインを提供します。

OneLogin の設定

アプリケーションの作成

1. OneLogin のアカウントにログインします。テスト目的で、OneLogin で無料の開発者アカウントを作成できます。

2. OneLogin の Web インターフェースで *Applications* > *Applications* に移動します。

3. "Add App" をクリックし、適切なアプリを検索します。このページのガイドラインは、*SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)* アプリの例に基づいています。

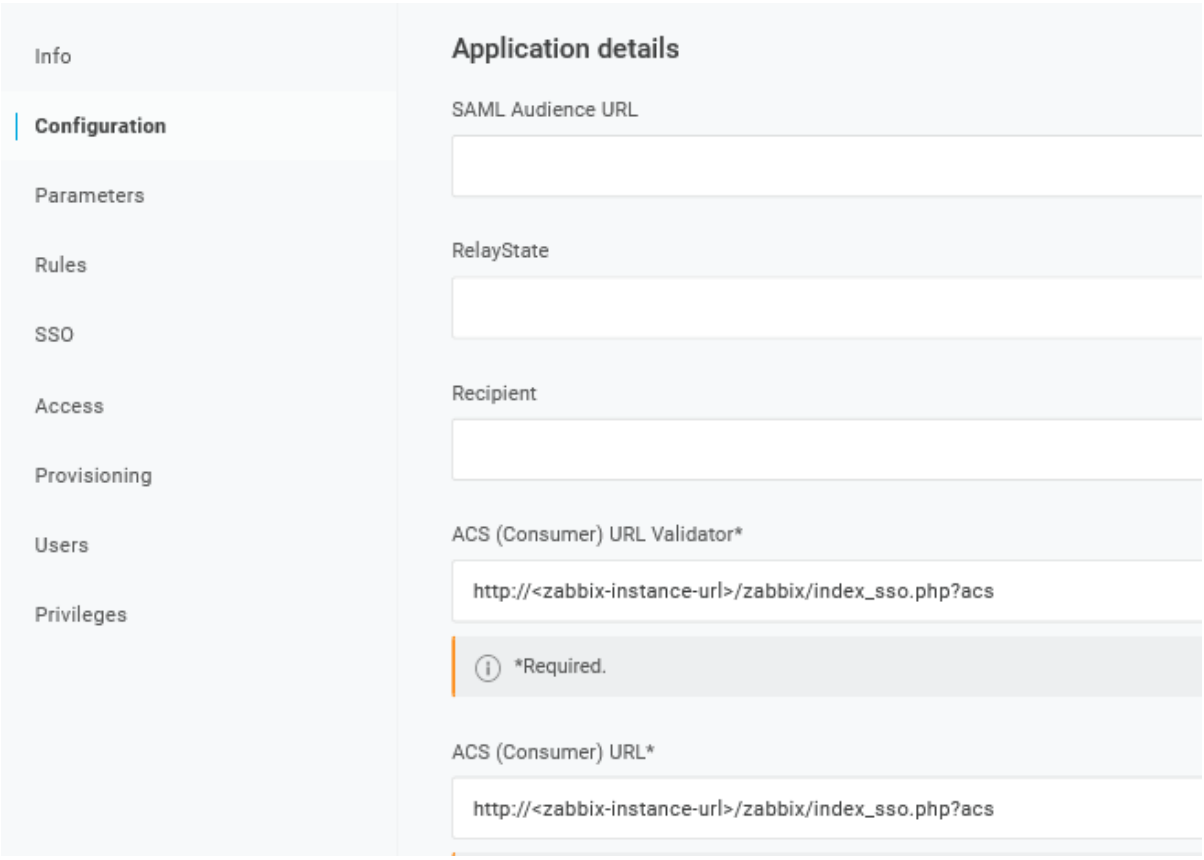
4. まず、アプリの表示名をカスタマイズするとよいでしょう。アイコンやアプリの詳細を追加することもできます。その後、*Save* をクリックします。

SSO/SCIM プロビジョニングの設定

1. *Configuration* > *Application details* で、Zabbix のシングルサインオンエンドポイント `http://<zabbix-instance-url>/zabbix/index_` を次のフィールドの値として設定します。

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

リクエストで acs パラメータが切り取られないように、「https」ではなく「http」を使用していることに注意してください。



「https」を使用することもできます。これを Zabbix で使用するには、Web インターフェース設定ファイル（`zabbix.conf.php`）に次の行を追加します。

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

その他のオプションはデフォルト値のままにします。

2. *Configuration* > *API connection* で、次の値を設定します。

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: `user_name`、`user_lastname`、`user_email`、`user_mobile` など、SCIM 経由で Zabbix に渡すすべてのカスタム属性を含める必要があります。

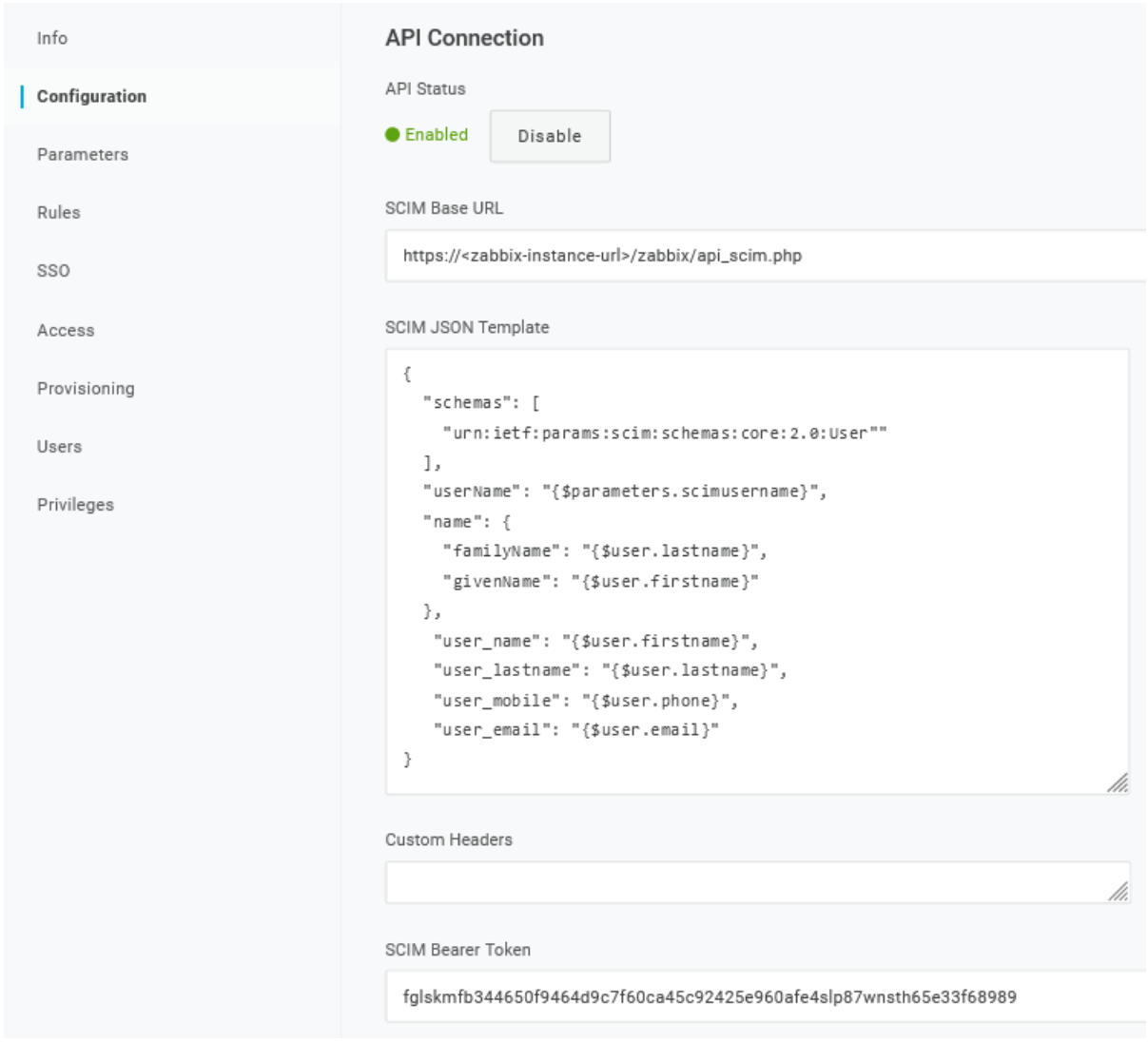
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

属性名は任意です。異なる属性名を使用できますが、Zabbix の SAML 設定における対応するフィールドの値と一致している必要があります。

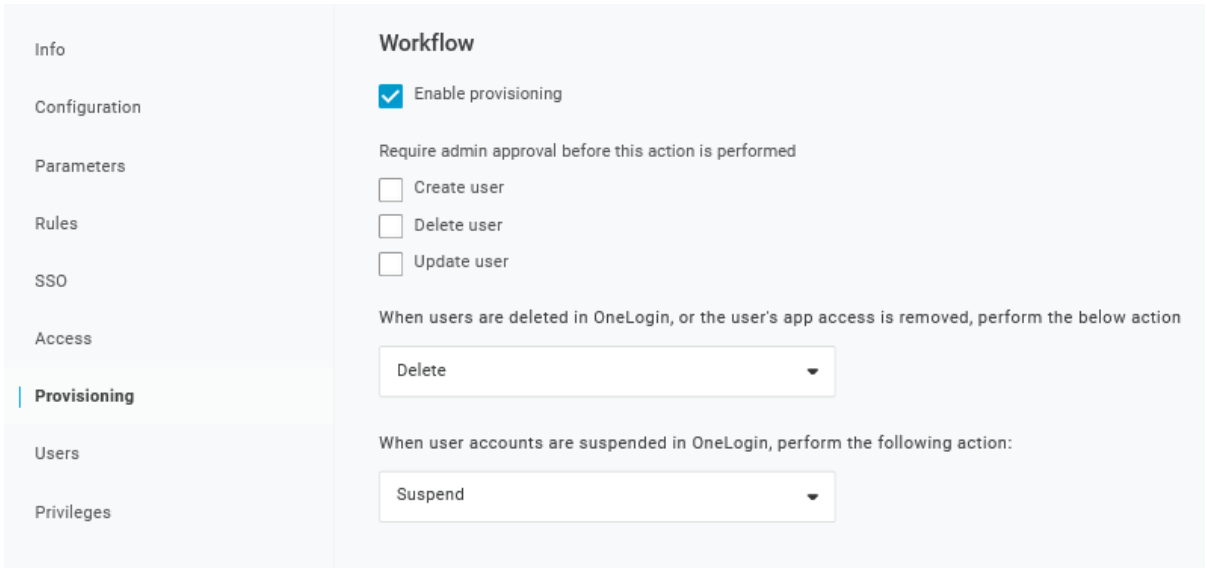
ユーザープロビジョニングを機能させるには、サービスプロバイダーで必須とされていない場合でも、OneLogin が応答で `givenName` と `familyName` を含む `name` 属性を受け取る必要があることに注意してください。そのため、アプリケーション設定のスキーマでこれを指定する必要があります。

- *SCIM Bearer Token*: Super admin 権限を持つ Zabbix API トークンを入力します。

Enable をクリックして接続を有効にします。

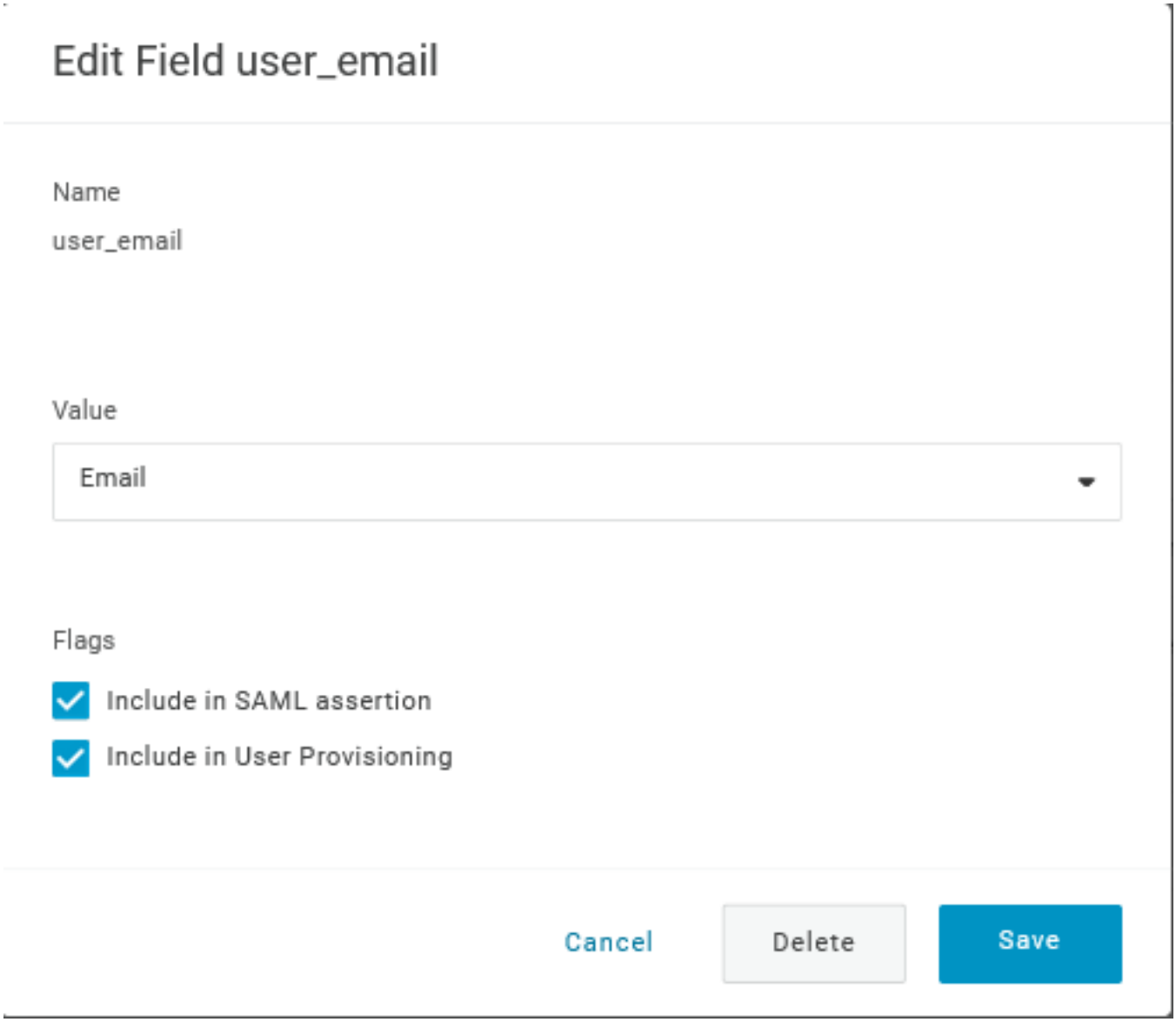


3. *Provisioning* ページで、Provisioning オプションを有効にします。



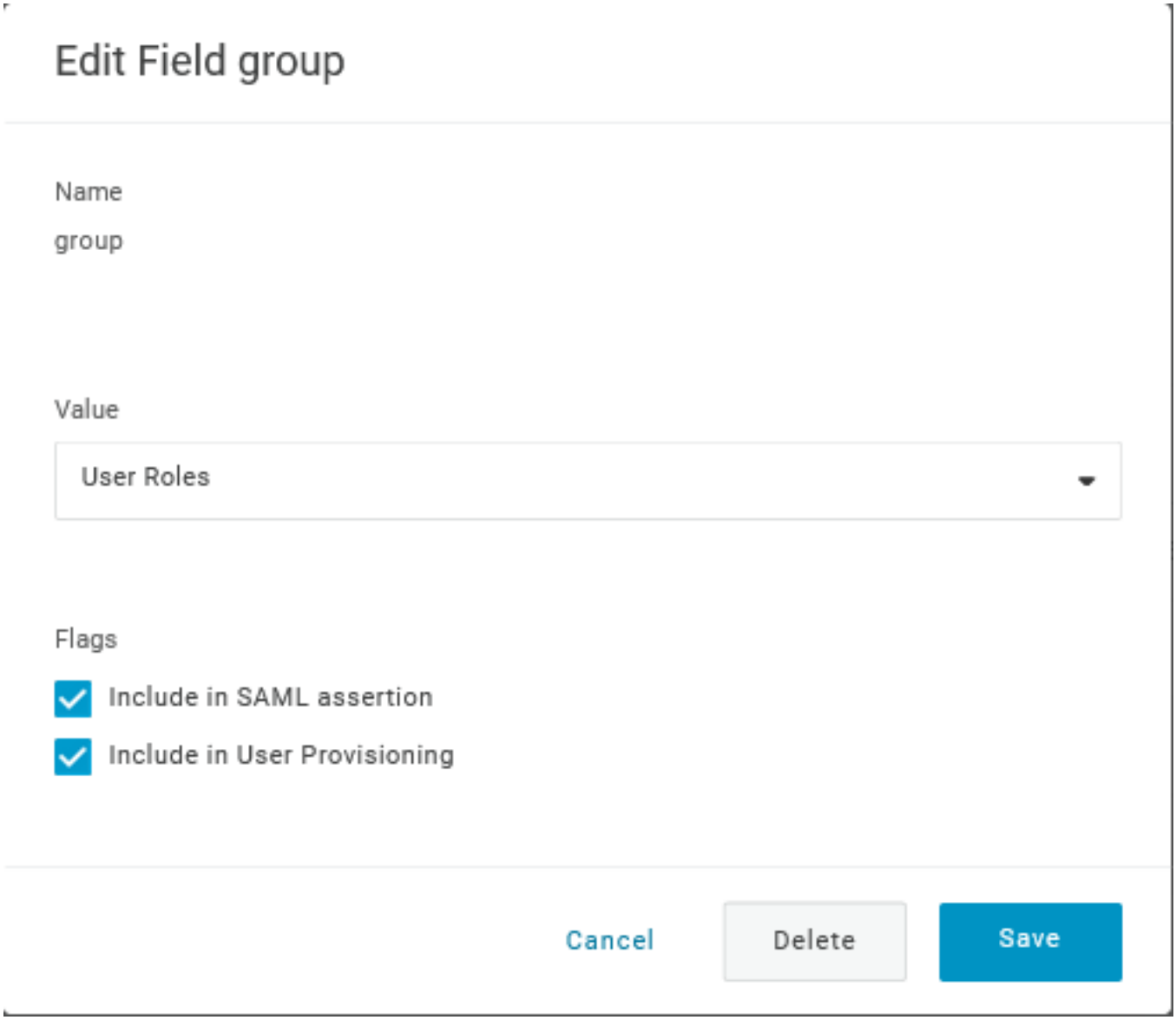
4. *Parameters* ページには、デフォルトパラメータの一覧が表示されます。

- `scimusername` が OneLogin のユーザーログイン値（例: メールアドレス）と一致していることを確認します。
- `Groups` パラメータの *Include in User Provisioning* オプションを選択します。
- 「+」をクリックして、SAML アサーションとユーザープロビジョニングに必要な `user_name`、`user_lastname`、`user_email`、`user_mobile` などのカスタムパラメータを作成します。

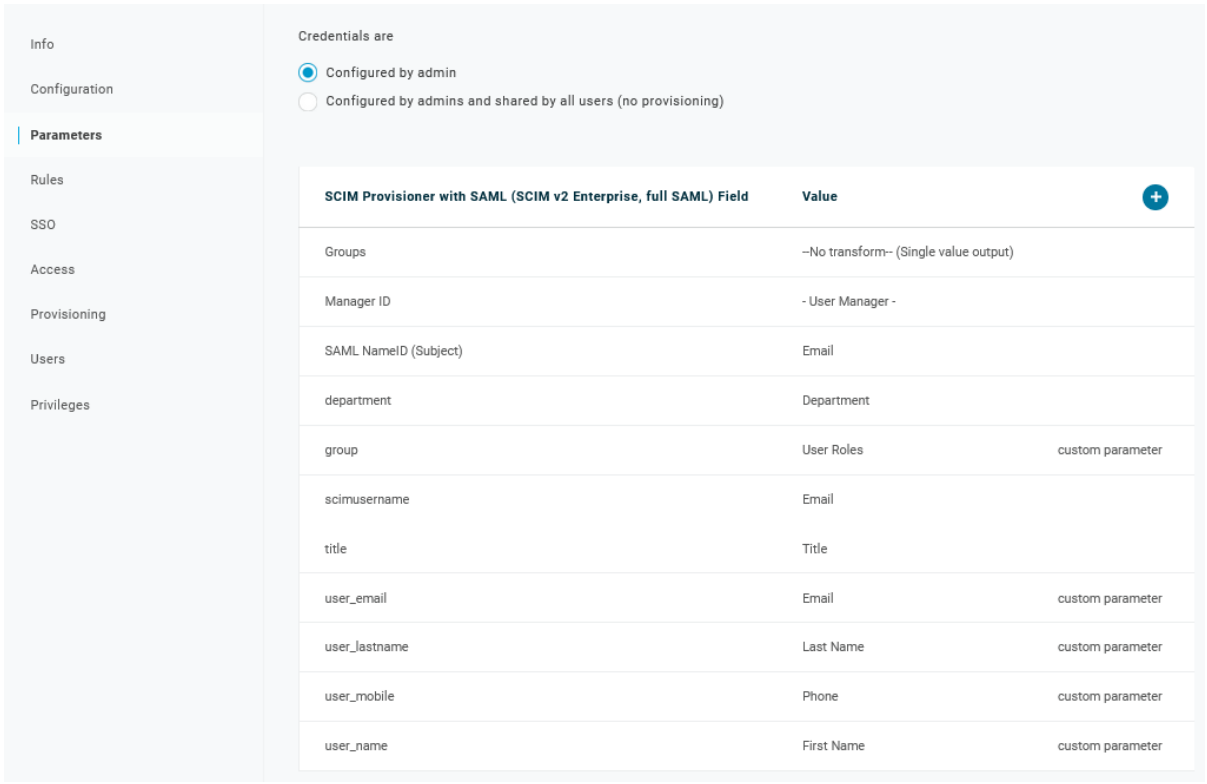


パラメータを追加する際は、*Include in SAML assertion* と *Include in User Provisioning* の両方のオプションを選択してください。

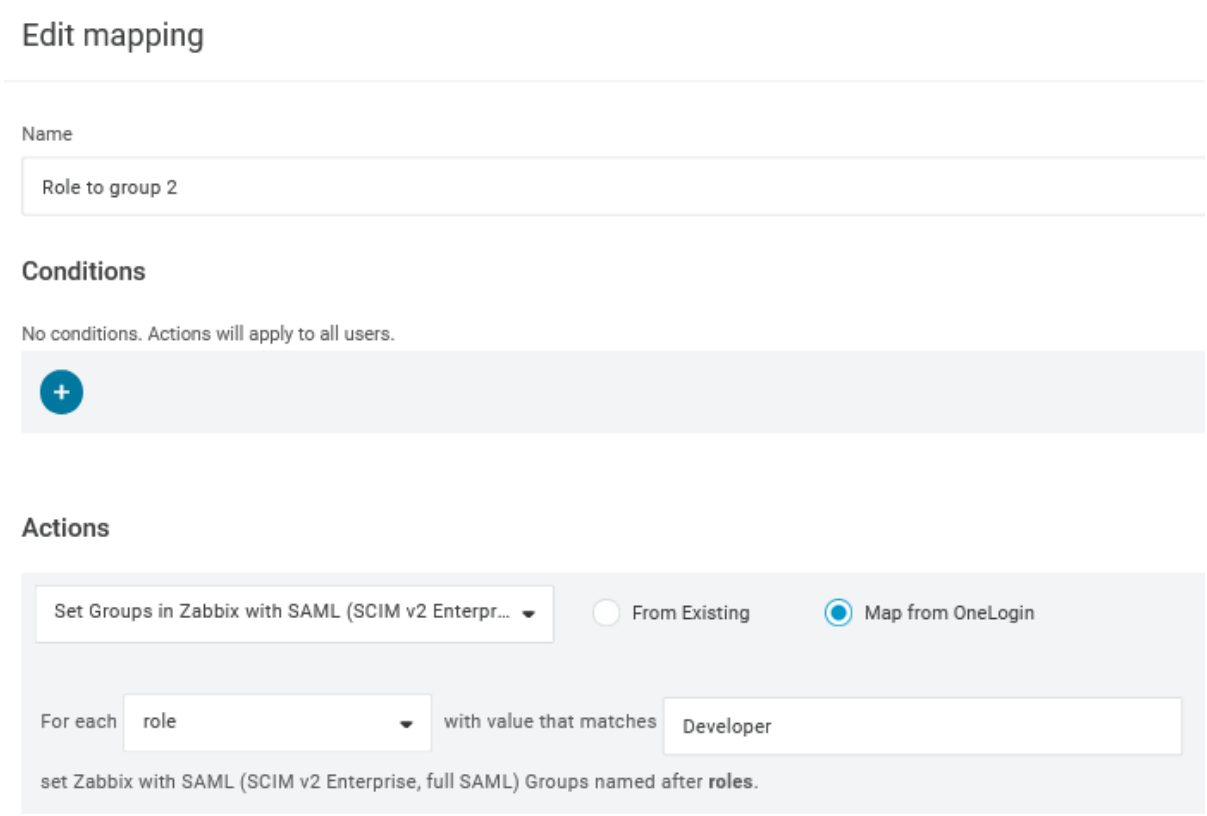
- OneLogin のユーザーロールに一致する `group` パラメータを追加します。ユーザーロールは、セミコロン `;` で区切られた文字列として渡されます。OneLogin のユーザーロールは、Zabbix でユーザーグループを作成するために使用されます。



パラメータの一覧を確認します。



5. *Rules* ページで、デフォルトの Groups パラメータに対するユーザーロールのマッピングを作成します。



正規表現を使用して、特定のロールをグループとして渡すこともできます。OneLogin は複数のロールを持つ属性を送信する際に ; を区切り文字として使用するため、ロール名に ; を含めないでください。

6. IdP 証明書をダウンロードします。次のコマンドを実行して、証明書に 644 の権限を設定します。

```
chmod 644 idp.crt
```

Zabbix の設定

1. Zabbix で、SAML 設定に移動し、OneLogin の設定に基づいて設定オプションを入力します。



Zabbix のフィールド	OneLogin の設定フィールド	サンプル値
IdP エンティティ *ID*	Issuer URL （OneLogin のアプリケーションの SSO タブを参照）	
SSO サービス *URL*	SAML 2.0 Endpoint (HTTP) （OneLogin のアプリケーションの SSO タブを参照）	

Zabbix のフィールド	OneLogin の設定フィールド	サンプル値
SLO サービス URL	SLO Endpoint (HTTP) （OneLogin のアプリケーションの SSO タブを参照）	
ユーザー名属性	Custom parameter	`user_email`
グループ名属性	Custom parameter	`group`
ユーザー名属性	Custom parameter	`user_name`
ユーザー姓属性	Custom parameter	`user_lastname`

ユーザーグループのマッピングも設定する必要があります。
メディアのマッピングは任意です。
これらの設定を保存するには、更新をクリックします。

2. OneLogin から提供された Base64 証明書を追加します。

Web インターフェース設定ファイル（`zabbix.conf.php`）で `$SSO['CERT_STORAGE'] = 'database'` が設定されている場合、SAML 設定中に Web インターフェースで証明書のテキストを貼り付けるか、証明書ファイルをアップロードできます。この場合、ファイルシステム上にファイルを用意する必要はありません。

設定ファイルで `$SSO['CERT_STORAGE'] = 'file'` が設定されている場合、証明書をファイルシステム上（デフォルトでは `ui/conf/certs`、または設定ファイルで指定されたパス）で利用できるようにする必要があります。また、Web インターフェースは証明書をデータベースに保存しません。
`$SSO['CERT_STORAGE']` が設定されていないかコメントアウトされている場合は、ファイルストレージが使用され、証明書は `ui/conf/certs` から読み込まれることに注意してください。

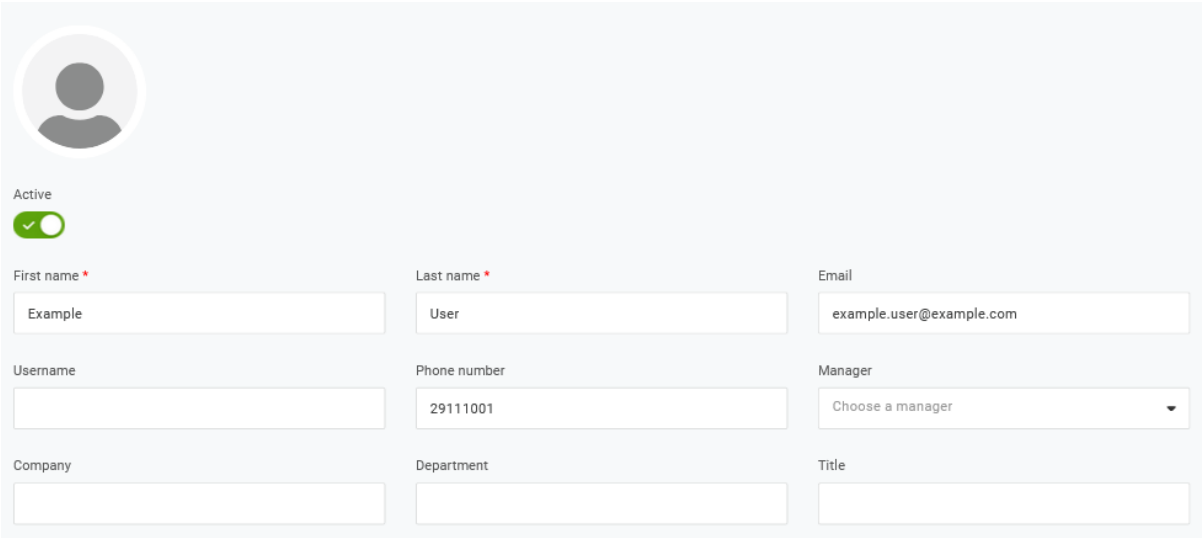
OneLogin で証明書をダウンロードするには、*Applications* > *SSO* に移動し、現在の証明書の下にある *View details* をクリックします。

3. これらの設定を保存するには、更新ボタンをクリックします。

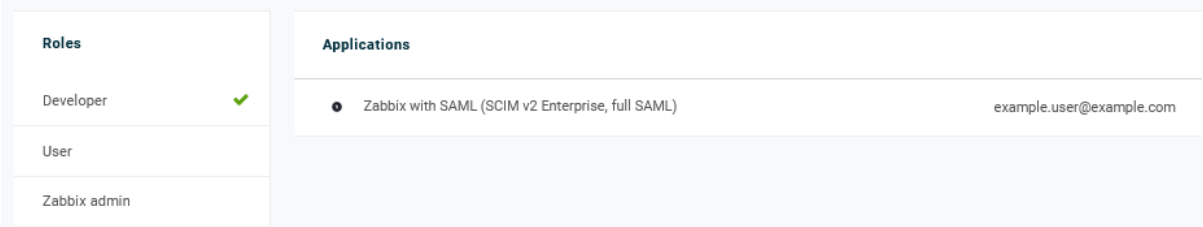
SCIM ユーザープロビジョニング

ユーザープロビジョニングを有効にすると、OneLogin でユーザーとそのロールを追加/更新し、それらを直ちに Zabbix へプロビジョニングできるようになります。

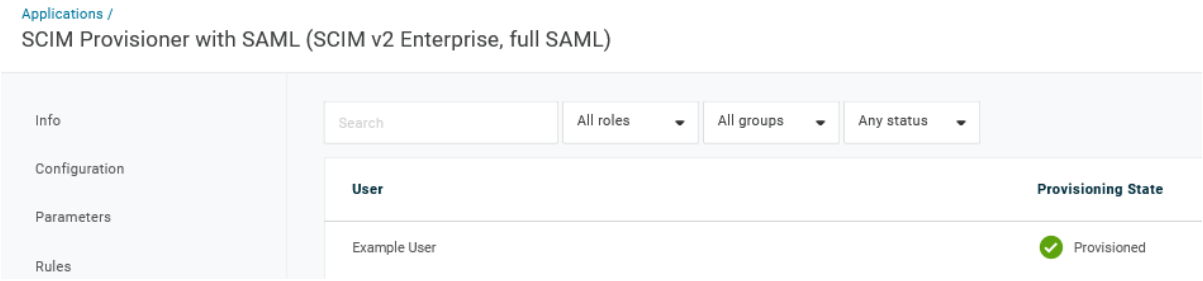
たとえば、新しいユーザーを作成できます。



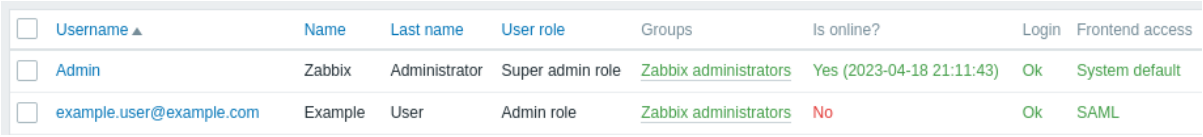
そのユーザーをユーザーロールと、ユーザーをプロビジョニングするアプリケーションに追加します。



ユーザーを保存すると、Zabbix へプロビジョニングされます。
Application > *Users* で、現在のアプリケーションユーザーのプロビジョニング状態を確認できます。



正常にプロビジョニングされると、そのユーザーは Zabbix のユーザー一覧に表示されます。



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2023-04-18 21:11:43)	Ok	System default
example.user@example.com	Example	User	Admin role	Zabbix administrators	No	Ok	SAML

スケジュールレポートの設定

概要

このセクションでは、Zabbix Web サービスのインストール方法と、スケジュールレポートの生成を有効にするための Zabbix の設定方法について説明します。

インストール

スケジュールレポートの生成を有効にするには、新しいZabbix Web サービスプロセスとGoogle Chrome ブラウザをインストールする必要があります。Web サービスは、Zabbix サーバーがインストールされているマシンと同じマシン、または別のマシンにインストールできます。Google Chrome ブラウザは、Web サービスがインストールされているのと同じマシンにインストールする必要があります。

公式の zabbix-web-service パッケージは、Zabbix リポジトリで入手できます。Google Chrome ブラウザはこれらのパッケージには含まれていないため、別途インストールする必要があります。

Zabbix web service をソースからコンパイルするには、Zabbix web service のインストール を参照してください。

インストール後、web service がインストールされているマシンで `zabbix_web_service` を実行します。

```
zabbix_web_service
```

設定

すべての関連要素間で適切に通信できるように、サーバー設定およびWeb インターフェース設定のパラメータが正しく設定されていることを確認してください。

Zabbix サーバー

Zabbix サーバーの設定ファイルで、次のパラメータを更新する必要があります: `WebServiceURL` および`StartReportWriters`。

WebServiceURL

このパラメータは、Web サービスとの通信を有効にするために必要です。URL は `http[s]://host:port/report` の形式で指定します。

- デフォルトでは、Web サービスはポート 10053 で待ち受けます。別のポートは、Web サービス設定ファイルで指定できます。
- `/report` パスの指定は必須です（このパスはハードコードされており、変更できません）。

例:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
Zabbix サーバーと Zabbix Web サービス間の暗号化を、証明書を使用して設定することを強く推奨します。デフォルトでは、Zabbix サーバーと Zabbix Web サービス間で送信されるデータは暗号化されないため、不正アクセスにつながる可能性があります。

StartReportWriters

このパラメータは、起動するレポートライタープロセスの数を決定します。設定されていない場合、または 0 に設定されている場合、レポート生成は無効になります。必要なレポートの数と頻度に応じて、1～100 個のレポートライタープロセスを有効にできます。

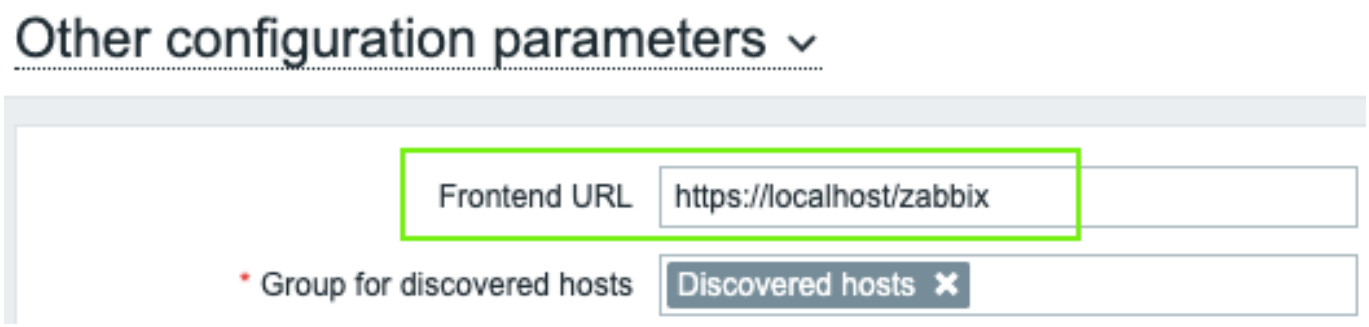
例:

```
StartReportWriters=3
```

Zabbix フロントエンド

Zabbix フロントエンドと Zabbix Web サービス間の通信を有効にするには、フロントエンド *URL* パラメータを設定する必要があります。

- フロントエンドの管理 > 一般 > その他メニューセクションに進みます
- フロントエンド *URL* パラメータに Zabbix Web インターフェースの完全な URL を指定します



Note:
セットアップ手順が完了したら、テストレポートを設定して送信し、すべてが正しく動作することを確認することをお勧めします。

アイテム

このセクションには、アイテムに関する参照情報と技術的な注意事項が含まれています。これには、特定のアイテムキーのパラメーター詳細、プラットフォーム固有の詳細、および特殊なアイテム設定に関する注意事項が含まれます。

サイドバーを使用して、用途に関連する情報を見つけてください。

vm.memory.size パラメータ

概要

このセクションでは、vm.memory.size[<mode>] エージェントアイテムのパラメータの詳細について説明します。

パラメータ

このアイテムでは、以下のパラメータを使用できます。

- **active** - 現在使用中、またはごく最近使用されたメモリであり、RAM 上に存在します
- **anon** - ファイルに関連付けられていないメモリ（ファイルから再読み込みできない）
- **available** - 利用可能なメモリ。プラットフォームに応じて計算方法が異なります（下表を参照）
- **buffers** - ファイルシステムのメタデータなどのためのキャッシュ
- **cached** - さまざまな用途のキャッシュ
- **exec** - 実行可能コード。通常は（プログラム）ファイルに由来します
- **file** - 最近アクセスされたファイルの内容のキャッシュ
- **free** - メモリを要求する任意のエンティティがすぐに利用できるメモリ
- **inactive** - 未使用としてマークされているメモリ
- **pavailable** - 'total' に対する'available' メモリの割合（`available/total*100` として計算）
- **pinned** - 'wired' と同じ
- **pused** - 'total' に対する'used' メモリの割合（`used/total*100` として計算）
- **shared** - 複数のプロセスから同時にアクセスされる可能性があるメモリ
- **slab** - カーネルが自身の使用のためにデータ構造をキャッシュするのに使用するメモリの総量
- **total** - 利用可能な物理メモリの総量
- **used** - 使用中のメモリ。プラットフォームに応じて計算方法が異なります（下表を参照）
- **wired** - 常に RAM 上に保持されるようマークされているメモリ。ディスクへ移動されることはありません。

Warning:
これらのパラメータの一部はプラットフォーム固有であり、お使いのプラットフォームでは利用できない場合があります。詳細はZabbix agent items を参照してください。

available および **used** のプラットフォーム固有の計算:

Platform	**”available”**	**”used”**
AIX	free + cached	実際に使用中のメモリ
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (RHEL 7 の 3.10 にもバックポート済み)	/proc/meminfo。詳細は Linux カーネルのdocumentationの”MemAvailable” を参照してください。 なお、すべてのページキャッシュを解放できるわけではなく、計算に low watermark が使用されるため、free + buffers + cached はもはや’available’ と等しくありません。	total - free
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:
vm.memory.size[used] と *vm.memory.size[available]* の合計は、必ずしも total と等しくなるとは限りません。たとえばFreeBSD では:
 * active、inactive、wired、cached メモリは、何らかの有用な情報を保持しているため、used と見なされます。
 * 同時に、inactive、cached、free メモリは、より多くのメモリを要求するプロセスに即座に割り当てられるため、available と見なされます。
したがって、inactive メモリは同時に used でもあり available でもあります。このため、*vm.memory.size[used]* アイテムは情報提供のみを目的として設計されており、一方 *vm.memory.size[available]* はトリガーで使用することを目的として設計されています。

以下も参照

1. 異なる OS でのメモリ計算に関する追加の詳細

Windows エージェントアイテムの最小権限レベル

概要

エージェントを使用してシステムを監視する場合は、エージェントがインストールされているホストからメトリックを取得することをお勧めします。最小特権の原則を使用するには、エージェントから取得するメトリックを判別する必要があります。

このドキュメントの表では、Zabbix エージェントの正しい動作を保証するための最小権限を選択できます。

エージェントの実行ユーザーを「LocalSystem」ではなく別のユーザーにした場合、エージェントを Windows サービスとして実行するために対象ユーザーは「ローカルポリシー」→「ユーザー権限の割り当て」から「サービスとしてログオン」の権限と、Zabbix エージェントのログファイルを作成、書き込み、削除する権限を持っている必要があります。対象ユーザーを ActiveDirectory の *PerformanceMonitorUsers* グループに追加する必要があります。

Note:
「技術的に許容できる最小限の」グループに基づくエージェントの権利を使用する場合は、監視対象のオブジェクトに対する権利を事前に提供する必要があります。

Windows でサポートされる共通エージェントアイテム

アイテムキー	ユーザーグループ	
	推奨	技術的に許容可能な最小値（機能は制限されます）
agent.hostname	Guests	Guests
agent.ping	Guests	Guests

アイテムキー	ユーザーグループ	
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Windows 固有のアイテムキー

アイテムキー	ユーザーグループ	
	推奨	技術的に最低限許容される値（機能は制限されます）
eventlog	Event Log Readers	Guests
net.if.list	Guests	Guests
perf_counter	Performance Monitor Users	Performance Monitor Users
proc_info	Administrators	Guests
service.discovery	Guests	Guests

アイテムキー	ユーザーグループ	
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Administrators	Guests
vm.vmemory.size	Guests	Guests

返される値のエンコーディング

Zabbix サーバーは、返されるすべてのテキスト値が UTF8 エンコーディングであることを想定しています。これは、Zabbix エージェント、SSH、Telnet など、あらゆる種類のチェックに関連します。

監視対象のさまざまなシステム/デバイスやチェックでは、値に非 ASCII 文字が返されることがあります。そのような場合、ほとんどすべての **zabbix** キーには、追加のアイテムキーパラメーター - **<encoding>** が含まれています。このキーパラメーターは任意ですが、返される値が UTF8 エンコーディングではなく、非 ASCII 文字を含む場合は指定する必要があります。指定しない場合、結果が予期しないものになり、動作が不安定になる可能性があります。

このような場合の、異なるデータベースバックエンドでの動作について以下に説明します。

MySQL

値に UTF8 以外のエンコーディングで非 ASCII 文字が含まれている場合、この文字とそれ以降の文字は、データベースがこの値を保存する際に破棄されます。警告メッセージは `zabbix_server.log` に書き込まれません。
 少なくとも MySQL バージョン 5.1.61 に該当します。

PostgreSQL

値に UTF8 以外のエンコーディングで ASCII 以外の文字が含まれている場合、SQL クエリは失敗します (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding)。その結果、データは保存されません。適切な警告メッセージが `zabbix_server.log` に書き込まれます。
 少なくとも PostgreSQL バージョン 9.1.3 に該当します。

大容量ファイルのサポート

大容量ファイルのサポートは、一般に LFS と略され、32 ビットオペレーティングシステム上で 2 GB を超えるファイルを扱える機能を指します。大容量ファイルのサポートは、少なくともログファイルの監視 とすべてのvfs.file.* アイテム に影響します。大容量ファイルのサポートは、Zabbix のコンパイル時におけるシステムの機能に依存しますが、procfs および swapctl との互換性がないため、32 ビット Solaris では完全に無効化されています。

センサー

概要

各センサー・チップは、sysfs `/sys/devices` ツリー内に独自のディレクトリを持ちます。すべてのセンサー・チップを見つけるには、`/sys/class/hwmon/hwmon*` からデバイスのシンボリックリンクをたどる方が簡単です。ここで * は実際の数字 (0,1,2,...) です。

センサーの読み取り値は、仮想デバイスの場合は `/sys/class/hwmon/hwmon*/` ディレクトリに、非仮想デバイスの場合は `/sys/class/hwmon/hwmon*/device` ディレクトリにあります。`hwmon*` または `hwmon*/device` ディレクトリ内にある `name` という名前のファイルには、チップの名前が含まれており、これはセンサー・チップで使用されているカーネル・ドライバの名前に対応します。

1 つのファイルにつき、1 つのセンサー読み取り値のみが記録されます。上記のいずれかのディレクトリ内でセンサー読み取り値を含むファイルの一般的な命名規則は、`<type><number>_<item>` です。ここで

- **type** - センサー・チップの場合は"in" (電圧)、"temp" (温度)、"fan" (ファン) など
- **item** - "input" (測定値)、"max" (上限値)、"min" (下限値) など
- **number** - 複数存在する可能性がある要素に常に使用されます (通常は 1 から始まりますが、電圧の場合は 0 から始まります)。ファイルが特定の要素を参照しない場合は、番号なしの単純な名前になります。

ホストで利用可能なセンサーに関する情報は、**sensors-detect** および **sensors** ツール (lm-sensors パッケージ) を使用して取得できます。**sensors-detect** は、利用可能なセンサーに必要なモジュールを特定するのに役立ちます。モジュールがロードされると、**sensors** プログラムを使用してすべてのセンサー・チップの読み取り値を表示できます。このプログラムで使用されるセンサー読み取り値のラベル付けは、一般的な命名規則 (`<type><number>_<item>`) とは異なる場合があります。

- `<type><number>_label` というファイルがある場合、このファイル内のラベルが `<type><number><item>` 名の代わりに使用されます。

- <type><number>_label ファイルがない場合、プログラムは /etc/sensors.conf (または /etc/sensors3.conf など) 内で名前の置換を検索します。

このラベル付けにより、ユーザーはどのようなハードウェアが使用されているかを特定できます。<type><number>_label ファイルも設定ファイル内のラベルもない場合は、name 属性 (hwmon*/device/name) でハードウェアの種類を特定できます。zabbix_agent が受け付けるセンサーの実際の名前は、**sensors** プログラムを -u パラメータ (sensors -u) で実行することで取得できます。

sensors プログラムでは、利用可能なセンサーはバスタイプ (ISA アダプタ、PCI アダプタ、SPI アダプタ、仮想デバイス、ACPI インターフェース、HID アダプタ) ごとに区切られています。

センサー識別子の取得

sensors コマンドで表示されるセンサーラベルは、チップベンダーによってラベル付けが異なるため、必ずしも直接使用できるとは限りません。たとえば、sensors の出力には次のような行が含まれる場合があります。

```
sensors
in0:          +2.24 V  (min =  +0.00 V, max =  +3.32 V)
Vcore:        +1.15 V  (min =  +0.00 V, max =  +2.99 V)
+3.3V:        +3.30 V  (min =  +2.97 V, max =  +3.63 V)
+12V:        +13.00 V  (min =  +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C  (low  = -127.0°C, high = +127.0°C)
```

このうち、内部識別子に直接対応する表示名は 1 つだけです。

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

他の表示ラベル（Vcore や +12V など）を使用しようとしても動作しません。

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Zabbix が受け付ける内部センサー識別子を見つけるには、sensors -u を実行します。出力には次のようなものが表示される場合があります。

```
sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

したがって、Vcore ラベルは識別子 in1 に、+12V は in4 に対応します。カーネル hwmon sysfs の仕様によると、これらはチップピンの電圧であり、一般的にはスケーリングが必要な場合があります。

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix では、電圧（in）だけでなく、電流（curr）、温度（temp）、ファン速度（fan）の値も取得できます。

互換性

レガシーディストリビューション

センサーの読み取り値は /proc/sys/dev/sensors ディレクトリから取得されます:

- **device** - デバイス名（<mode> が使用されている場合は正規表現）
- **sensor** - センサー名（<mode> が使用されている場合は正規表現）
- **mode** - 可能な値: avg, max, min（このパラメータが省略された場合、device と sensor はそのまま扱われます）

キーの例: sensor[w83781d-i2c-0-2d,temp1]。

現在のディストリビューション

センサーの読み取り値は /sys/class/hwmon ディレクトリから取得されます:

- **device** - デバイス名（正規表現ではありません）。デバイス名は、実際のデバイス名（例: 0000:00:18.3）または sensors プログラムで取得した名前（例: k8temp-pci-00c3）のいずれかです。どちらの名前を使用するかはユーザーが選択できます。
- **sensor** - センサー名（正規表現ではありません）。
- **mode** - 可能な値: avg, max, min（このパラメータを省略した場合、device と sensor はそのまま扱われます）。

キーの例:

sensor[k8temp-pci-00c3,temp,max] または sensor[0000:00:18.3,temp1]

sensor[smsc47b397-isa-0880,in,avg] または sensor[smsc47b397.2176,in1]

proc.mem アイテムの **memtype** パラメーターに関する注意事項

概要

memtype パラメータは、Linux、AIX、FreeBSD、Solaris の各プラットフォームでサポートされています。

これらすべてのプラットフォームでは、pmem、rss、vsize の 3 つの一般的な memtype 値がサポートされています。さらに、一部のプラットフォームでは、プラットフォーム固有の memtype 値もサポートされています。

AIX

AIX で memtype パラメータに指定できる値については、以下の表を参照してください。

サポートされている値	説明	procentry64 構造体のソース	互換性を試みる対象
vsize [1]	仮想メモリサイズ	pi_size	
pmem	実メモリの割合	pi_prm	ps -o pmem
rss	常駐セットサイズ	pi_trss + pi_drss	ps -o rssize
size	プロセスサイズ（コード + データ）	pi_dvm	ps gvw の SIZE 列
dsize	データサイズ	pi_dsize	
tsize	テキスト（コード）サイズ	pi_tsize	ps gvw の TSIZ 列
sdsize	共有ライブラリのデータサイズ	pi_sdsize	
drss	データ常駐セットサイズ	pi_drss	
trss	テキスト常駐セットサイズ	pi_trss	

AIX に関する注意事項:

1. AIX で proc.mem[] アイテムキーのパラメータを選択する場合は、プロセスの選択条件をできるだけ絞り込んで指定してください。そうしないと、proc.mem[] の結果に意図しないプロセスがカウントされる可能性があります。

例:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

この例は、コマンドライン（照合する正規表現）パラメータだけを指定すると、Zabbix エージェント自身がカウントされることを示しています。これはおそらく意図した結果ではありません。

2. プロセスの確認に ps -ef を使用しないでください。カーネル以外のプロセスのみが表示されます。Zabbix エージェントから認識できるすべてのプロセスを確認するには、ps -Af を使用します。

3. topasrec を例に、Zabbix エージェントの proc.mem[] がプロセスを選択する方法を見ていきましょう。

```
$ ps -Af | grep topasrec
root 10747984        1   0   Mar 16      -  0:00 /usr/bin/topasrec  -L -s 300 -R 1 -r 6 -o /var/perf daily
```

proc.mem[] には次の引数があります。

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

条件は次のとおりです。- 1 つ目の条件は、プロセス名（引数 <name>）です。この例では、Zabbix エージェントはこれを topasrec と認識します。一致させるには、topasrec を指定するか、空のままにします。- 2 つ目の条件は、ユーザー名（引数 <user>）です。一致させるには、root を指定するか、空のままにします。- プロセス選択で使用される 3 つ目の条件は、引数 <cmdline> です。Zabbix エージェントは、その値を /usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -ystart_time=04:08:54,Mar16,2023 と認識します。一致させるには、この文字列に一致する正規表現を指定するか、空のままにします。

引数 <mode> と <memtype> は、上記の 3 つの条件を使用した後に適用されます。

FreeBSD

FreeBSD で memtype パラメータに指定できる値については、以下の表を参照してください。

サポートされている値	説明	kinfo_proc 構造体内のソース	互換性を考
vsize	仮想メモリサイズ	kp_eproc.e_vm.vm_map.size または ki_size	ps -o vs
pmem	実メモリの割合	rss から計算	ps -o pm
rss	常駐セットサイズ	kp_eproc.e_vm.vm_rssize または ki_rssize	ps -o rss
size [1]	プロセスのサイズ（コード + データ + スタック）	tsize + dsize + ssize	
tsize	テキスト（コード）サイズ	kp_eproc.e_vm.vm_tsize または ki_tsize	ps -o ts
dsize	データサイズ	kp_eproc.e_vm.vm_dsize または ki_dsize	ps -o ds
ssize	スタックサイズ	kp_eproc.e_vm.vm_ssize または ki_ssize	ps -o ss

Linux

Linux で memtype パラメータに指定できる値については、以下の表を参照してください。

サポートされている値	説明	/proc/<pid>/status ファイルのソース
vsize [1]	仮想メモリサイズ	VmSize
pmem	実メモリの割合	(VmRSS/total_memory) * 100
rss	常駐セットサイズ	VmRSS
data	データセグメントのサイズ	VmData
exe	コードセグメントのサイズ	VmExe
hwm	常駐セットサイズのピーク値	VmHWM
lck	ロックされたメモリのサイズ	VmLck
lib	共有ライブラリのサイズ	VmLib
peak	仮想メモリサイズのピーク値	VmPeak
pin	固定されたページのサイズ	VmPin
pte	ページテーブルエントリのサイズ	VmPTE
size	プロセスのコード + データ + スタックセグメントのサイズ	VmExe + VmData + VmStk
stk	スタックセグメントのサイズ	VmStk
swap	使用されているスワップ領域のサイズ	VmSwap

Linux に関する注意事項:

1. 古い Linux カーネルでは、すべての memtype 値がサポートされているわけではありません。たとえば、Linux 2.4 カーネルでは、hwm、pin、peak、pte、swap の値はサポートされていません。
2. proc.mem[...,...,...,...,data] を使用して Zabbix エージェントのアクティブチェックプロセスを自己監視すると、エージェントの/proc/<pid>/status ファイルの VmData 行で報告される値より 4 kB 大きい値が表示されることを確認しています。自己測定の時点では、エージェントのデータセグメントが 4 kB 増加し、その後、以前のサイズに戻ります。

Solaris

Solaris で memtype パラメータに指定できる値については、以下の表を参照してください。

サポートされている値	説明	psinfo 構造体内のソース	互換性を試みる対象
vsize [1]	プロセスイメージのサイズ	pr_size	ps -o vsz
pmem	実メモリの割合	pr_pctmem	ps -o pmem
rss	常駐セットサイズ 過小評価される場合があります（man ps の rss の説明を参照してください）。	pr_rssize	ps -o rss

脚注

[1] デフォルト値

proc.mem および **proc.num** アイテムでのプロセス選択に関する注意事項

コマンドラインを変更するプロセス

一部のプログラムは、現在のアクティビティを表示する方法としてコマンドラインの変更を使用します。ユーザーは ps や top コマンドを実行することでアクティビティを確認できます。このようなプログラムの例としては、*PostgreSQL*、*Sendmail*、*Zabbix* などがあります。

Linux の例を見てみましょう。Zabbix エージェントプロセスの数を監視したいと仮定します。

ps コマンドは、対象のプロセスを次のように表示します。

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/zabbix_agentd.conf
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

名前とユーザーでプロセスを選択すると、目的を達成できます。

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

ここで、zabbix_agentd 実行ファイルの名前を zabbix_agentd_30 に変更し、再起動します。

ps の表示は次のようになります。

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/zabbix_agentd.conf
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

この状態で名前とユーザーでプロセスを選択すると、誤った結果が得られます。

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

なぜ実行ファイルの名前を単純に長くしただけで、まったく異なる結果になったのでしょうか？

Zabbix エージェントは、まずプロセス名をチェックします。/proc/<pid>/status ファイルを開き、Name 行を確認します。この場合、Name 行は次のようになります。

```
$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status
/proc/6715/status:Name:   zabbix_agentd_3
/proc/6716/status:Name:   zabbix_agentd_3
/proc/6717/status:Name:   zabbix_agentd_3
/proc/6718/status:Name:   zabbix_agentd_3
/proc/6719/status:Name:   zabbix_agentd_3
/proc/6720/status:Name:   zabbix_agentd_3
```

status ファイル内のプロセス名は 15 文字に切り詰められます。

同様の結果は ps コマンドでも確認できます。

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

明らかに、これは proc.num[] の name パラメータ値 zabbix_agentd_30 とは一致しません。status ファイルのプロセス名と一致しなかったため、Zabbix エージェントは /proc/<pid>/cmdline ファイルを参照します。

エージェントが"cmdline" ファイルをどのように認識するかは、次のコマンドで確認できます。

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/zabbix_agentd.conf<NUL>
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

この場合、/proc/<pid>/cmdline ファイルには、*C* 言語で文字列を終了するために使用される不可視の非表示のヌルバイトが含まれています。この例では、ヌルバイトは"<NUL>" として表示されています。

Zabbix エージェントは、メインプロセスの"cmdline" をチェックし、zabbix_agentd_30 を取得します。これは name パラメータ値 zabbix_agentd_30 と一致します。したがって、メインプロセスはアイテム proc.num[zabbix_agentd_30,zabbix] でカウントされます。

次のプロセスをチェックする際、エージェントは cmdline ファイルから zabbix_agentd_30: collector [idle 1 sec] を取得しますが、これは name パラメータ zabbix_agentd_30 とは一致しません。したがって、コマンドラインを変更しないメインプロセスのみがカウントされ、他のエージェントプロセスはコマンドラインを変更するため無視されます。

この例から、name パラメータはこの場合、proc.mem[] や proc.num[] でプロセスを選択するために使用できないことが分かります。

> **Note:**
> proc.get[] アイテムの場合、Zabbix エージェントがプロセス名のために"cmdline" をチェックする際、最後のスラッシュから最初のスペースまたはコロン記号までの名前の一部のみを使用します。cmdline ファイルから取得したプロセス名は、その先頭部分が status ファイルの短縮されたプロセス名と完全に一致する場合にのみ使用されます。このアルゴリズムは、フィルター内のプロセス名と JSON 出力内のプロセス名の両方で同じです。

cmdline パラメータに適切な正規表現を使用すると、正しい結果が得られます。

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

コマンドラインを変更するプログラムの監視には、proc.get[]、proc.mem[]、proc.num[] アイテムの使用に注意してください。

proc.get[]、proc.mem[]、proc.num[] アイテムに name および cmdline パラメータを設定する前に、proc.num[] アイテムや ps コマンドを使ってパラメータをテストすることをお勧めします。

Linux カーネルスレッド

proc.get[]、proc.mem[]、proc.num[] アイテムの cmdline パラメータでスレッドを選択できません

カーネルスレッドの 1 つを例にとってみましょう:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

これはプロセスの name パラメータで選択できます:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

しかし、プロセスの cmdline パラメータによる選択は機能しません:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

その理由は、Zabbix エージェントが cmdline パラメータで指定された正規表現をプロセスの/proc/<pid>/cmdline の内容に適用するためです。カーネルスレッドの場合、/proc/<pid>/cmdline ファイルは空です。そのため、cmdline パラメータは一致しません。

proc.mem[] および proc.num[] アイテムのスレッドカウント

Linux カーネルスレッドは proc.num[] アイテムによってカウントされますが、proc.mem[] アイテムのメモリは報告されません。

例：

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

しかし、カーネルスレッドと同じ名前のユーザープロセスがある場合は次のようになります。

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]
zabbix    9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

proc.num [] はカーネルスレッドとユーザープロセスの両方をカウントしました。proc.mem [] はユーザープロセスのメモリのみを報告し、カーネルスレッドメモリを 0 であるかのようにカウントします。これは ZBX_NOTSUPPORTED が報告された上記の場合とは異なります。

プログラム名がスレッドの 1 つと一致する場合は、proc.mem[] および proc.num[] アイテムを使用するときに注意してください。

パラメータを proc.mem[] および proc.num[] アイテムに入れる前に、proc.num[] アイテムと ps コマンドを使用してパラメータをテストすることをお勧めします。

net.tcp.service および **net.udp.service** チェックの実装詳細

net.tcp.service および net.udp.service チェックの実装については、service パラメータで指定される各種サービスごとに、このページで詳しく説明しています。

net.tcp.service アイテムのパラメータ

ftp

TCP 接続を作成し、応答の最初の 4 文字が"220 " であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 21 が使用されます。

http

特に何もせず TCP 接続を作成します。ポートが指定されていない場合はデフォルトのポート 80 が使用されます。

https

libcurl を使用し（そしてそれとともにのみ機能し）、証明書の信頼性を検証せず、SSL 証明書のホスト名を検証せず、応答ヘッダー (HEAD 要求）のみを読み出します。ポートが指定されていない場合はデフォルトポートの 443 が使用されます。

imap

TCP 接続を作成し、応答の最初の 4 文字が"* OK" であることを検証して"a1 LOGOUT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 143 が使用されます。

ldap

LDAP サーバーへの接続を開き、フィルターを (objectClass=*) に設定して LDAP 検索操作を実行して、最初のエントリの最初の属性が正常に取得できることを検証します。ポートが指定されていない場合はデフォルトのポート 389 が使用されます。

nntp

TCP 接続を作成し、応答の最初の 3 文字が"200" または"201" であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 119 が使用されます。

pop

TCP 接続を作成し、応答の最初の 3 文字が"+OK" であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 110 が使用されます。

smtp

TCP 接続を作成し、応答の最初の 3 文字が”220” であり、その後にスペース、行末、またはダッシュが続くことを検証します。ダッシュを含む行は複数行の応答で、ダッシュのない行が受信されるまで応答が再読み取りされます。次に”QUIT\r\n” を送信します。ポートが指定されていない場合はデフォルトのポート 25 が使用されます。

ssh

TCP 接続を作成します。接続が確立されている場合、両側で識別文字列（SSH-major.minor-XXXX）が交換されます。ここでのメジャーとマイナーはプロトコルバージョンであり、XXXX は文字列です。Zabbix は仕様に一致する文字列が見つかったかどうかを確認し、不一致の場合は文字列”SSH-major.minor-zabbix_agent\r\n” または”0\n” を送り返します。ポートが指定されていない場合は、デフォルトのポート 22 が使用されます。

tcp

特に何もせず TCP 接続を作成します。他のチェックとは異なり、ポート指定必須です。

telnet

TCP 接続を作成し、ログインプロンプト (の最後に':') が返ってくることを検証します。ポートが指定されていない場合はデフォルトのポート 23 が使用されます。

net.udp.service アイテムのパラメータ

ntp

UDP を介して SNTP パケットを送信し、RFC 4330、セクション 5に従って応答を検証します。ポートが指定されていない場合は、デフォルトのポート 123 が使用されます。

proc.get パラメーター

概要

proc.get[<name>,<user>,<cmdline>,<mode>] アイテムは、Linux、Windows、FreeBSD、OpenBSD、NetBSD でサポートされています。

アイテムによって返されるプロセスパラメータのリストは、オペレーティングシステムと’mode’ 引数の値によって異なります。

Linux

Linux では、各モードについて次のプロセスパラメータが返されます。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親 PID	ppid: 親 PID	processes: プロセス数
name: プロセス名	name: プロセス名	vsize: 仮想メモリサイズ
cmdline: 引数付きコマンド	user: プロセスを実行しているユーザー (実ユーザー)	pmem: 実メモリの割合
user: プロセスを実行しているユーザー (実ユーザー)	group: プロセスを実行しているグループ（実グループ）	rss: 常駐セットサイズ
group: プロセスを実行しているグループ（実グループ）	uid: ユーザー ID	data: データセグメントのサイズ
uid: ユーザー ID	gid: プロセスを実行しているグループの ID	exe: コードセグメントのサイズ
gid: プロセスを実行しているグループの ID	tid: スレッド ID	lib: 共有ライブラリのサイズ
vsize: 仮想メモリサイズ	tname: スレッド名	lck: ロックされたメモリのサイズ
pmem: 実メモリの割合	cputime_user: ユーザーモードで消費した CPU 時間の合計（値はクロックティックで表されます。秒単位に変換するには、`getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割ります）	pin: 固定されたページのサイズ
rss: 常駐セットサイズ	cputime_system: システムモードで消費した CPU 時間の合計（値はクロックティックで表されます。秒単位に変換するには、`getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割ります）	pte: ページテーブルエントリのサイズ

mode=process	mode=thread	mode=summary
data: データセグメントのサイズ	state: スレッドの状態	size: プロセスのコード、データ、スタックセグメントの合計サイズ
exe: コードセグメントのサイズ	ctx_switches: コンテキストスイッチの回数	stk: スタックセグメントのサイズ
hwm: 常駐セットサイズのピーク値	page_faults: ページフォールトの回数	swap: 使用中のスワップ領域のサイズ
lck: ロックされたメモリのサイズ		cputime_user: ユーザーモードで消費した CPU 時間の合計（値はクロックティックで表されます。秒単位に変換するには、`getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割ります）
lib: 共有ライブラリのサイズ		cputime_system: システムモードで消費した CPU 時間の合計（値はクロックティックで表されます。秒単位に変換するには、`getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割ります）
peak: 仮想メモリサイズのピーク値		ctx_switches: コンテキストスイッチの回数
pin: 固定されたページのサイズ		threads: スレッド数
pte: ページテーブルエントリのサイズ		page_faults: ページフォールトの回数
size: プロセスのコード、データ、スタックセグメントの合計サイズ		pss: 比例セットサイズメモリ
stk: スタックセグメントのサイズ		
swap: 使用中のスワップ領域のサイズ		
cputime_user: ユーザーモードで消費した CPU 時間の合計（値はクロックティックで表されます。秒単位に変換するには、`getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割ります）		
cputime_system: システムモードで消費した CPU 時間の合計（値はクロックティックで表されます。秒単位に変換するには、`getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で割ります）		
state: プロセスの状態（procfs から透過的に取得された長形式）		
ctx_switches: コンテキストスイッチの回数		
threads: スレッド数		
page_faults: ページフォールトの回数		
pss: 比例セットサイズメモリ		

BSD 系 OS

各モードで FreeBSD、OpenBSD、NetBSD で返されるプロセスパラメータは以下の通りです。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親プロセスの PID	ppid: 親プロセスの PID	processes: プロセス数
jid: ジェイルの ID(FreeBSD のみ)	jid: ジェイルの ID(FreeBSD のみ)	vsize: 仮想メモリサイズ
jname: ジェイル名 (FreeBSD のみ)	jname: ジェイル名 (FreeBSD のみ)	pmem: 実メモリの割合 (FreeBSD のみ)
name: プロセス名	name: プロセス名	rss: 常駐セットサイズ
cmdline: 引数付きコマンド	user: プロセスが実行されているユーザー (実ユーザー)	size: プロセスサイズ (コード + データ + スタック)
user: プロセスが実行されているユーザー (実ユーザー)	group: プロセスが実行されているグループ (実グループ)	tsize: テキスト (コード) サイズ

mode=process	mode=thread	mode=summary
group: プロセスが実行されているグループ (実グループ)	uid: ユーザー ID	dsize: データサイズ
uid: ユーザー ID	gid: プロセスが実行されているグループの ID	ssize: スタックサイズ
gid: プロセスが実行されているグループの ID	tid: スレッド ID	cputime_user: ユーザーモードで消費した CPU 時間 (秒単位)
vsize: 仮想メモリサイズ	tname: スレッド名	cputime_system: システムモードで消費した CPU 時間 (秒単位)
pmem: 実メモリの割合 (FreeBSD のみ)	cputime_user: ユーザーモードで消費した CPU 時間 (秒単位)	ctx_switches: コンテキストスイッチ数
rss: 常駐セットサイズ	cputime_system: システムモードで消費した CPU 時間 (秒単位)	threads: スレッド数 (NetBSD は非対応)
size: プロセスサイズ (コード + データ + スタック)	state: スレッド状態	stk: スタックセグメントサイズ
tsize: テキスト (コード) サイズ	ctx_switches: コンテキストスイッチ数	page_faults: ページフォールト数
dsize: データサイズ	io_read_op: システムが入力を実行しなければならなかった回数	fds: ファイルディスクリプタ数 (OpenBSD のみ)
ssize: スタックサイズ	io_write_op: システムが出力を実行しなければならなかった回数	swap: 使用スワップサイズ
cputime_user: ユーザーモードで消費した CPU 時間 (秒単位)		io_read_op: システムが入力を実行しなければならなかった回数
cputime_system: システムモードで消費した CPU 時間 (秒単位)		io_write_op: システムが出力を実行しなければならなかった回数
state: プロセス状態 (ディスクスリープ/実行中/スリープ/トレース停止/ゾンビ/その他)		
ctx_switches: コンテキストスイッチ数		
threads: スレッド数 (NetBSD は非対応)		
page_faults: ページフォールト数		
fds: ファイルディスクリプタ数 (OpenBSD のみ)		
swap: 使用スワップサイズ		
io_read_op: システムが入力を実行しなければならなかった回数		
io_write_op: システムが出力を実行しなければならなかった回数		

Windows

各モードで Windows で返されるプロセスパラメータは以下の通りです。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親プロセスの PID	ppid: 親プロセスの PID	processes: プロセス数
name: プロセス名	name: プロセス名	vmsize: 仮想メモリサ
user: プロセスを実行しているユーザー	user: プロセスを実行しているユーザー	wkset: プロセスのワー
sid: ユーザー SID	sid: ユーザー SID	cputime_user: ユーザー
vmsize: 仮想メモリサイズ	tid: スレッド ID	cputime_system: シス
wkset: プロセスのワーキングセットサイズ		threads: スレッド数
cputime_user: ユーザーモードで消費した CPU 時間の合計 (ミリ秒単位)		page_faults: ページフ
cputime_system: システムモードで消費した CPU 時間の合計 (ミリ秒単位)		handles: ハンドル数
threads: スレッド数		io_read_b: IO で読み込
page_faults: ページフォールト数		io_write_b: IO で書き込
handles: ハンドル数		io_read_op: IO の読み込
io_read_b: IO で読み込んだバイト数		io_write_op: IO の書き込
io_write_b: IO で書き込んだバイト数		io_other_b: 読み書き以
io_read_op: IO の読み込み操作数		io_other_op: 読み書き以
io_write_op: IO の書き込み操作数		
io_other_b: 読み書き以外の IO で転送されたバイト数		

mode=process	mode=thread	mode=summary
io_other_op: 読み書き以外の IO 操作数		

到達不能/利用不可なホストインターフェース設定

概要

いくつかの設定パラメーター により、エージェントチェック（Zabbix、SNMP、IPMI、JMX）が失敗し、ホストインターフェースに到達できなくなった場合に、Zabbix サーバーがどのように動作するかが定義されます。

到達不能なインターフェース

ホストインターフェースは、Zabbix、SNMP、IPMI、または JMX エージェントによるチェックが失敗した場合（ネットワークエラー、タイムアウト）に到達不能と見なされます。Zabbix 6.2.0 以降では、アクティブな Zabbix エージェントのチェックもインターフェースの可用性に影響します。アクティブチェックが利用できなくなると、エージェントインターフェース全体の可用性ステータスに反映されます。

インターフェースが到達不能になった時点から、**UnreachableDelay** により、1 つのアイテム（LLD ルールを含む）を使用して再チェックを行う頻度が定義されます。
これらの再チェックは、到達不能ポーラー（IPMI チェックの場合は IPMI ポーラー）によって実行されます。
デフォルトでは、連続する到達性チェックの間隔は 15 秒です。

> **Attention:**
> 非同期ポーラーによって実行されるチェックは、到達不能ポーラーに移動されません。

Zabbix サーバーのログでは、到達不能状態は次のようなメッセージで示されます。

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

ログメッセージには、失敗した正確なアイテムとその種類（Zabbix エージェント）が記載されます。

> **Note:**
> *Timeout* パラメータは、到達不能時にインターフェースをどの程度早く再チェックするかにも影響します。Timeout を 20 秒、UnreachableDelay を 30 秒に設定した場合、最初の試行から 50 秒後に次のチェックが行われます。

UnreachablePeriod パラメータは、到達不能期間の合計時間を定義します。デフォルトの UnreachablePeriod は 45 秒です。
この値は、インターフェースが利用不可と判定される前に複数回再チェックされるよう、UnreachableDelay よりも数倍大きく設定する必要があります。

内部アイテムzabbix[host,active_agent,available] を使用すると、到達不能なシナリオにおけるアクティブチェックの可用性を監視できます。

インターフェイスを到達可能に戻す

到達不能期間が終了すると、インターフェイスが再度ポーリングされ、インターフェイスを到達不能状態にしたアイテムの優先度が低下します。到達不能なインターフェイスが再び到達可能になると、監視は自動的に通常に戻ります。

```
resuming Zabbix agent checks on host "New host": connection restored
```

> **Note:**
> インターフェイスが使用可能になっても、ホストは次の 2 つの理由ですべてのアイテムをすぐにポーリングしません。
> - ホストが過負荷になる可能性があります。
> - インターフェイスの復元時間は、計画されたアイテムのポーリングスケジュール時間と常に一致するとは限りません。
>
> そのため、インターフェイスが使用可能になった後、アイテムはすぐにはポーリングされず、次のポーリングラウンドに再スケジュールされます。

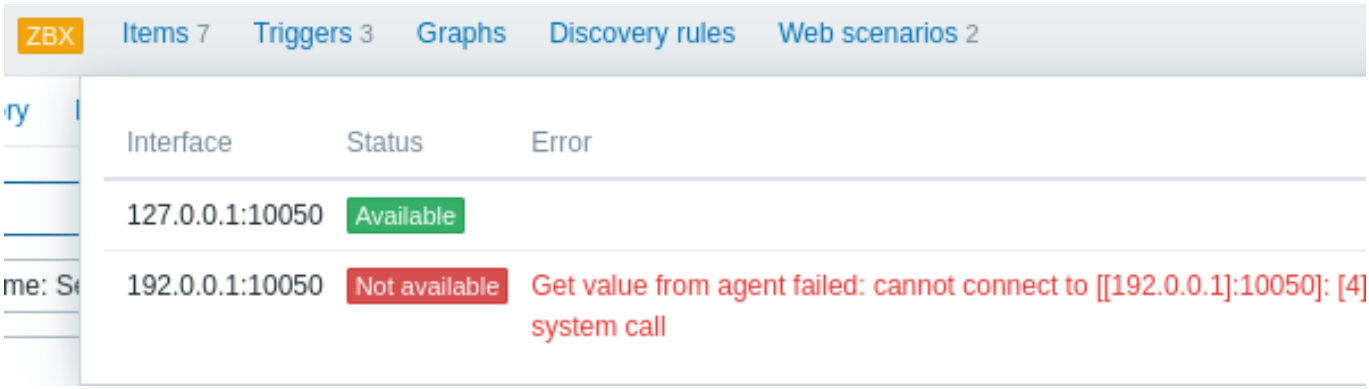
利用できないインターフェース

UnreachablePeriod が終了し、インターフェースが再び利用可能にならなかった場合、そのインターフェースは利用できないものとして扱われます。

サーバーログには、次のようなメッセージが表示されます。

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

また、フロントエンド では、ホストの可用性アイコンが緑/グレーから黄/赤に変わります（利用できないインターフェースの詳細は、ホストの可用性アイコンにマウスを合わせたときに表示されるヒントボックスで確認できます）。



UnavailableDelay パラメータは、インターフェースが利用できない間にどのくらいの頻度でインターフェースをチェックするかを定義します。

デフォルトでは 60 秒です（この場合、上記のログメッセージの「temporarily disabling」は、1 分間チェックを無効にすることを意味します）。

インターフェースへの接続が復旧すると、監視も自動的に通常状態に戻ります。

```
enabling Zabbix agent checks on host "New host": interface became available
```

Zabbix 統計のリモート監視

概要

Zabbix サーバーおよびプロキシの内部メトリクスの一部を、別の Zabbix インスタンスまたはサードパーティ製ツールからリモートでアクセス可能にすることができます。
これは、サポート担当者やサービスプロバイダーがクライアントの Zabbix サーバー/プロキシをリモートで監視する場合や、組織内で Zabbix が主要な監視ツールではない場合に、Zabbix の内部メトリクスをサードパーティ製システムで包括的な監視構成の一部として監視する場合に役立ちます。

Zabbix の内部統計は、新しい'StatsAllowedIP' server/proxy パラメータで指定された、設定可能なアドレスのセットに公開されます。要求は、これらのアドレスからのみ受け付けられます。

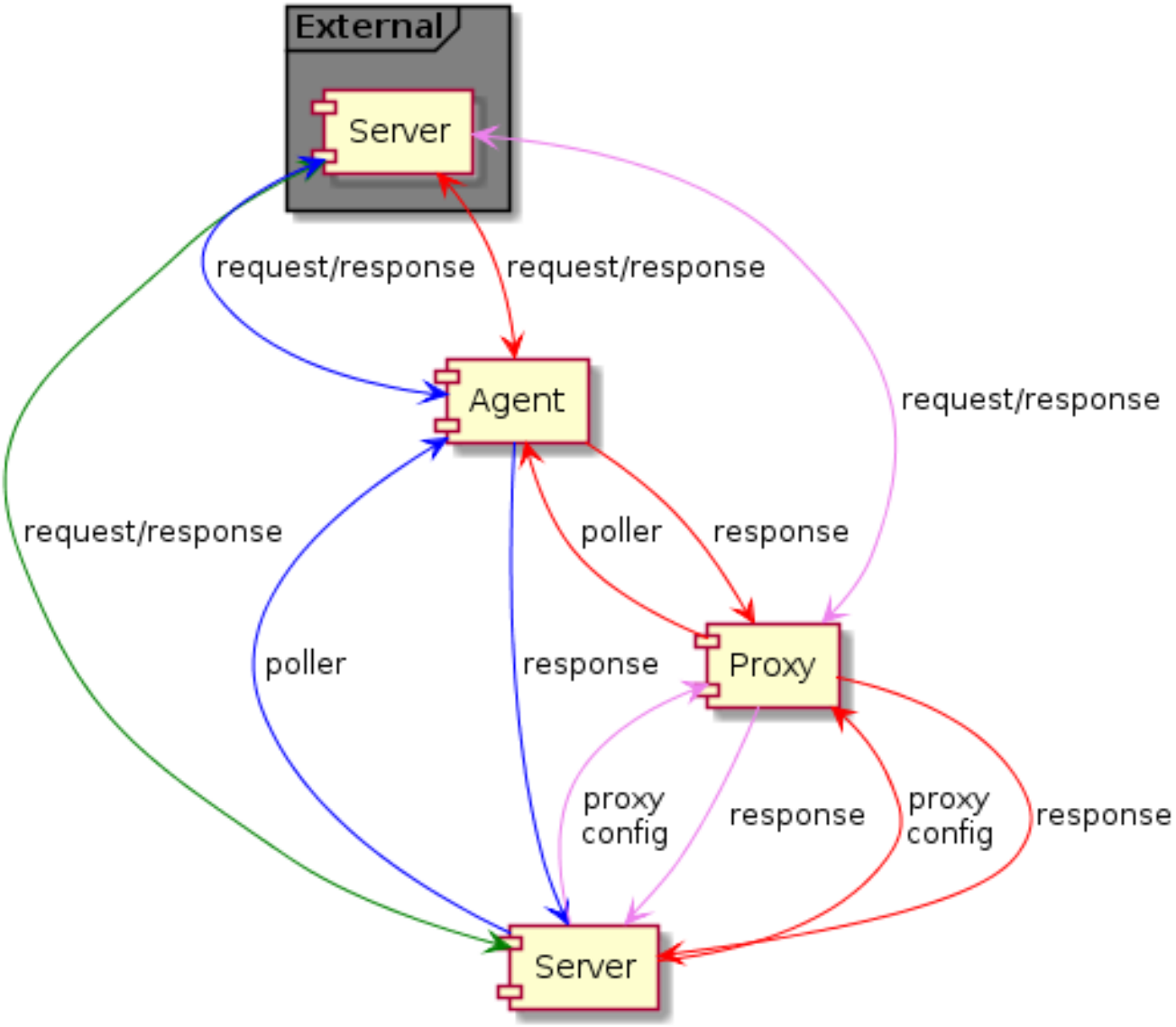
アイテム

別の Zabbix インスタンスで内部統計のクエリを設定するには、次の 2 つの項目を使用します。

- `zabbix[stats,<ip>,<port>]` internal item - Zabbix サーバー/プロキシの直接リモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用します
- `zabbix.stats[<ip>,<port>]` agent item - Zabbix サーバー/プロキシのエージェントベースのリモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用されます。

参考情報: 内部項目, Zabbix エージェントアイテム

次の図はコンテキストに応じた、いずれかのアイテムの使用法を示しています。



- - Server → external Zabbix instance (zabbix[stats,<ip>,<port>])
- - Server → proxy → external Zabbix instance (zabbix[stats,<ip>,<port>])
- - Server → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])
- - Server → proxy → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])

ターゲットインスタンスが外部インスタンスによるクエリを許可するには、ターゲットインスタンスの'StatsAllowedIP' パラメータに外部インスタンスのアドレスをリストします。

収集されるメトリクス

stats アイテムは統計情報をまとめて収集し、JSON 形式で返します。これは依存アイテムがデータを取得するための基礎となります。以下の内部メトリクスが、いずれかのアイテムによって返されます。

- zabbix[boottime]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[preprocessing] (サーバーのみ)
- zabbix[preprocessing_queue] (サーバーのみ)
- zabbix[process,<type>,<mode>,<state>] (プロセスタイプベースの統計のみ)
- zabbix[rcache,<cache>,<mode>]
- zabbix[requiredperformance]
- zabbix[triggers] (サーバーのみ)
- zabbix[uptime]
- zabbix[vcache,buffer,<mode>] (サーバーのみ)

- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] ('trends' キャッシュタイプはサーバーのみ)

テンプレート

テンプレートは、外部インスタンスから Zabbix サーバーまたはプロキシの内部メトリクスをリモート監視するために利用できます:

- リモート Zabbix サーバーヘルス
- リモート Zabbix プロキシヘルス

複数の外部インスタンスのリモート監視用テンプレートを使用するには、各外部インスタンス監視ごとに個別のホストが必要であることに注意してください。

トラッパープロセス

外部の Zabbix インスタンスからの内部メトリックリクエストの受信は、トラッパープロセスによって処理されます。このプロセスはリクエストを検証し、メトリックを収集し、JSON データバッファを作成し、準備した JSON を返します。例えば、サーバーからは以下のようになります。

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing": {
        "queued": {
          "count": N,
          "size": N
        },
        "direct": {
          "count": N,
          "size": N
        },
        "queue": N
    },
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "requiredperformance": N,
```

```
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,
        "misses": N,
        "mode": N
      }
    },
    "vmware": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "version": "N",
    "wcache": {
      "values": {
        "all": N,
        "float": N,
        "uint": N,
        "str": N,
        "log": N,
        "text": N,
        "not supported": N
      },
      "history": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "index": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "trend": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      }
    }
  }
}
```

内部キューアイテム

別の Zabbix インスタンス上の内部キュー統計をリモートからクエリするための、次の 2 つのアイテムもあります。

- zabbix[stats,<ip>,<port>,queue,<from>,<to>] 内部アイテム - リモートの Zabbix サーバー/プロキシに対して、内部キューを直接クエリするために使用します
- zabbix.stats[<ip>,<port>,queue,<from>,<to>] エージェントアイテム - リモートの Zabbix サーバー/プロキシに対して、エージェントベースで内部キューをクエリするために使用します

関連項目: 内部アイテム、Zabbix エージェントアイテム

Zabbix で **Kerberos** を設定する

概要

Kerberos 認証は、Zabbix の Web 監視や HTTP アイテムで使用できます。

このページでは、Debian/Ubuntu で Zabbix プロセス用の Kerberos プリンシパルを使用して、Zabbix サーバーが `www.example.com` の Web 監視を実行するための Kerberos の設定例について説明します。

設定

1. KDC とクライアントユーティリティーをインストールします。

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

パッケージのセットアップ中に、次のようなプロンプトに回答します。

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost（または FQDN）
Administrative server for your Kerberos realm: localhost（または FQDN）
```

2. わかりやすいホスト名をマッピングします（オプション、ローカルテスト用）。

DNS がない場合は、/etc/hosts を編集し、DC と Web サーバーのエントリを追加します。

```
sudo vi /etc/hosts
```

追加するエントリの例:

```
192.168.1.100  dc01.example.com dc01
```

3. Kerberos クライアントと KDC レルムを設定します。

```
sudo vi /etc/krb5.conf
```

設定例:

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

`.localdomain` やその他のパブリックではない名前を使用する場合は、ホスト名からレルムへのマッピングが機能するように、明示的なドメイン → レルムのマッピングを追加します。ここで不一致があると、`Server not found in Kerberos database` エラーが発生します。

4. Kerberos データベースを初期化します（KDC ホストで一度だけ実行）。プロンプトが表示されたら、安全なマスターパスワードを設定します。

```
sudo krb5_newrealm
```

5. クライアントが使用する正確なホスト名を使用して、HTTP/host.fqdn@REALM プリンシパルを作成します。小文字を使用することを推奨します（例: HTTP/web.example.com@EXAMPLE.COM）。大文字と小文字または名前の不一致があると、Server not found in Kerberos database エラーが発生します。

```
sudo kadmin.local
```

kadmin.local 内で次を実行します。

```
addprinc kerb-admin@EXAMPLE.COM      # 管理用プリンシパル
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

keytab を Web ホストに移動します（同じマシンの場合はローカルに保持できます）。Apache が使用できるように権限を設定します。

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### 確認
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Apache GSSAPI モジュールをインストールして有効にします。

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> すべての mod_auth_gssapi バージョンが、すべての Gssapi* ディレクティブをサポートしているわけではありません。Apache が Invalid command 'GssapiCredStore' で失敗する場合は、サポートされていないディレクティブを削除するか、モジュールをアップグレードしてください。

7. VirtualHost を設定します（DocumentRoot と Zabbix UI のパスを適宜変更してください）。

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

zabbix.conf 内に次を記述します。

```
<VirtualHost *:80>
    ServerName dc01.example.com
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Apache を再起動します。

```
sudo systemctl restart apache2
```

8. KDC サービスを有効化して起動し、待ち受けポートを確認します（KDC ホスト）。

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # または: sudo netstat -tnlp | grep :80
```

9. テスト用の TGT を取得します（チケットを使用するユーザーとして実行）。

チケット一覧に krbtgt/EXAMPLE.COM@EXAMPLE.COM が表示されることを確認します。チケットを必要とする OS ユーザー

と同じユーザーとして kinit を実行します（例: Web チェックの場合は zabbix、対話型ブラウザー SSO テストの場合は www-data/Apache）。別の OS ユーザーに発行されたチケットは、KRB5CCNAME と権限を調整しない限り表示されません。

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. 有効な TGT を持つクライアントから、curl で SPNEGO 交換をテストします。200 OK（またはアプリケーションへのリダイレクト）が返されれば、SPNEGO は成功しています。

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. 必要に応じて、Zabbix UI で HTTP 認証ログインを受け入れるように設定します。

Web インターフェース設定ファイル（zabbix.conf.php）で、次を設定します。

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

次に、Web インターフェースで HTTP 認証を設定します。

12. ブラウザーの設定（ここでは Firefox を例にします）: Negotiate を実行するホスト（dc01.example.com）を network.negotiate-auth.trusted-uris に設定し、ブラウザーが Kerberos トークンを自動的に送信できるようにします。

about:config 内で次を設定します。

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

これで http://dc01.example.com にアクセスすると、フォームを表示せずに Zabbix に直接ログインできるはずです。

13. キーとチケットを最新の状態に保ちます。Kerberos のデフォルトのチケット有効期間は約 10 時間です。有効期限切れを防ぐため、cron または systemd タイマーを追加します。

```
#### Web サービス用
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
#### 監視ユーザー用
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. 保守確認:

- klist -k /etc/apache2/http.keytab - サービスプリンシパルが keytab に存在することを確認します。
- sudo tail -f /var/log/apache2/error.log - GSSAPI エラーを監視します（gss_acquire_cred[_from]() failed to get server creds は、keytab または権限の問題、あるいはプリンシパルが存在しないことを示します）。
- curl --negotiate が 401/403 を返す場合、多くの場合、プリンシパルの誤り、チケットがない、ホストヘッダーの不一致、またはファイルシステム権限の問題が原因です。ログと /etc/krb5.conf のドメインマッピングを確認してください。

セキュリティとファイル権限に関する注意事項

Keytab ファイルは、必要とするアカウントだけが読み取れるようにしてください。
権限の例: zabbix ユーザーの keytab の場合は 0400 で所有者を zabbix:zabbix に設定し、Apache の keytab の場合は 0440 で所有者を root:www-data に設定します。

ホスト上に長期間有効な平文パスワードを保存しないでください。
可能な場合は、keytab またはドメインに参加しているマシンプリンシパルを使用してください。

KRB5CCNAME を設定するテストやスクリプトを実行したり、keytab をコピーしたりする場合は、操作後に所有者と権限を再確認してください。Web サーバーが認証情報を拒否する場合、ファイル権限の問題であることがよくあります。

modbus.get パラメータ

概要

以下の表は、アイテムmodbus.get のパラメーターの詳細を示しています。

パラメーター

パラメーター	説明	デフォルト	例
endpoint	プロトコルとエンドポイントのアドレス。 `protocol://connection_string` で定義 プロトコルの値: *rtu*, *ascii* (Agent 2 のみ), *tcp* コネクションストリングのフォーマット: *tcp* の場合 - `address:port` シリアルラインの場合: *rtu*, *ascii* - `port_name:speed:params` ここで 'speed' - 1200, 9600 など 'params' - データビット (5,6,7 または 8)、パリティ (n,e,o のいずれか、none/even/odd)、ストップビット (1 または 2)	プロトコル: なし *rtu/ascii* プロトコル: port_name: なし speed: 115200 params: 8n1 *tcp* プロトコル: address: なし port: 502	tcp://192.168.6.1:511 tcp://192.168.6.2 tcp://[::1]:511 tcp://::1 tcp://localhost:511 tcp://localhost rtu://COM1:9600:8n ascii://COM2:1200:7o2 rtu://ttyS0:9600 ascii://ttyS1
slave id	対象デバイスの Modbus アドレス (1～247)。MODBUS Messaging Implementation Guide (23 ページ) を参照 tcp デバイス (GW 以外) はこのフィールドを無視	serial: 1 tcp: 255 (0xFF)	2
function	空、またはサポートされている関数の値: 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	空	3
address	最初のレジスタ、コイル、または入力のアドレス。 'function' が空の場合、'address' は以下の範囲で指定: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 'function' が空でない場合、'address' フィールドは 0～65535 の範囲で、修正なしで使用 (PDU)	function が空: 00001 function が空でない: 0	9999
count	デバイスから読み取る連続した'type' の数。 Coil または Discrete input の場合、'type' = 1 ビット その他の場合: (count*sizeof(type))/2 = 実際に読み取るレジスタ数 'offset' が 0 でない場合、その値が' 実際の数' に加算される 許容範囲は 1:65535	1	2
type	データ型: Read Coil, Read Discrete Input の場合 - *bit* Read Holding Registers, Read Input Registers の場合: *int8* - 8 ビット *uint8* - 8 ビット (符号なし) *int16* - 16 ビット *uint16* - 16 ビット (符号なし) *int32* - 32 ビット *uint32* - 32 ビット (符号なし) *float* - 32 ビット *uint64* - 64 ビット (符号なし) *double* - 64 ビット	bit uint16	uint64

パラメーター	説明	デフォルト	例
endianness	エンディアンタイプ: *be* - ビッグエンディアン *le* - リトルエンディアン *mbe* - ミッドビッグエンディアン *mle* - ミッドリトルエンディアン 制限: 1 ビットの場合 - be 8 ビットの場合 - be,le 16 ビットの場合 - be,le	be	le
offset	'address' から始まるレジスタ数で、その結果は破棄される。 各レジスタのサイズは 16 ビット (ランダムリードアクセスをサポートしない機器のために必要)	0	4

VMware のカスタムパフォーマンスカウンター名の作成

概要

VMware のパフォーマンスカウンターのパスは `group/counter[rollup]` 形式で、次の要素で構成されます。

- `group` - パフォーマンスカウンターのグループ。たとえば *cpu*
- `counter` - パフォーマンスカウンター名。たとえば *usagemhz*
- `rollup` - パフォーマンスカウンターのロールアップタイプ。たとえば *average*

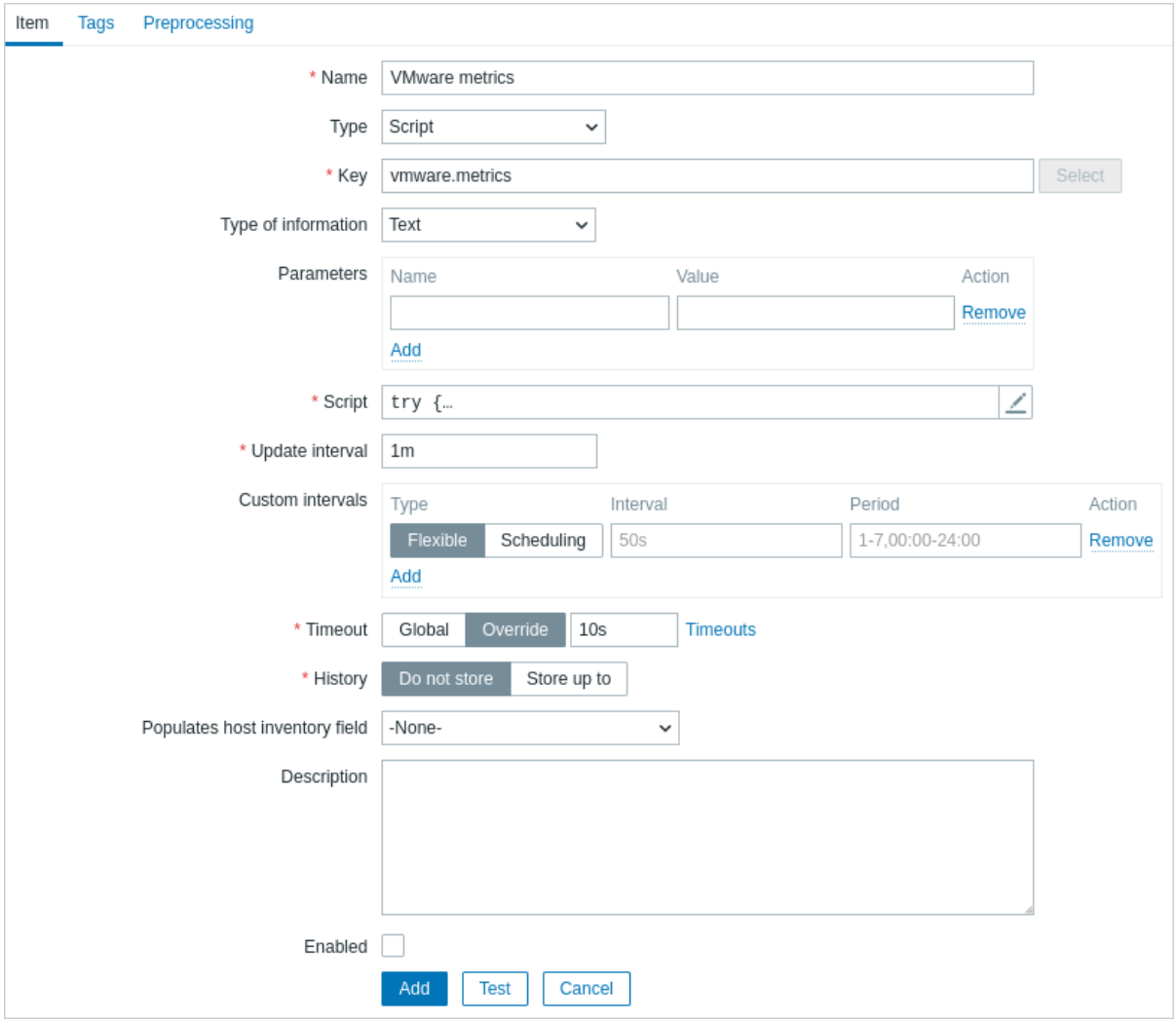
したがって、上記の例では次のカウンターパスになります。`cpu/usagemhz[average]`

パフォーマンスカウンターのグループの説明、カウンター名、およびロールアップタイプは、VMware documentation で確認できます。

Zabbix のスクリプトアイテムを使用することで、内部名を取得し、カスタムのパフォーマンスカウンター名を作成することも可能です。

設定

1. メインの VMware ホスト（**eventlog[]** アイテムが存在するホスト）に、以下のパラメータで無効化された Script アイテムを作成します。



- *Name*: VMware metrics
- *Type*: Script
- *Key*: vmware.metrics
- *Type of information*: Text
- *Script*: 下記のscript をコピーして貼り付けます
- *Timeout*: 10
- *History*: 保存しない
- *Enabled*: 無効

Script

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');

    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
```

```
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

アイテムを設定したら、*Test* ボタンをクリックし、次に *Get value* をクリックします。



受信した XML を任意の XML フォーマッターにコピーし、目的のメトリクスを見つけます。

1 つのメトリクスに対する XML の例:

```
<PerfCounterInfo xsi:type="PerfCounterInfo">
    <key>6</key>
```

```
    <nameInfo>
        <label>Usage in MHz</label>
        <summary>CPU usage in megahertz during the interval</summary>
        <key>usagemhz</key>
    </nameInfo>
    <groupInfo>
        <label>CPU</label>
        <summary>CPU</summary>
        <key>cpu</key>
    </groupInfo>
    <unitInfo>
        <label>MHz</label>
        <summary>Megahertz</summary>
        <key>megaHertz</key>
    </unitInfo>
    <rollupType>average</rollupType>
    <statsType>rate</statsType>
    <level>1</level>
    <perDeviceLevel>3</perDeviceLevel>
</PerfCounterInfo>
```

受信した XML からカウンターパスを抽出するには、XPath を使用します。上記の例では、XPath は次のようになります。

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

この場合のパフォーマンスカウンターパスは cpu/usagemhz[average] です

system.sw.packages.get の戻り値

概要

このセクションでは、system.sw.packages.get Zabbix エージェントアイテムの戻り値の詳細について説明します。

詳細

このアイテムの出力はオブジェクトの配列で、各オブジェクトには以下のプロパティが含まれます。

- **name** - パッケージ名
- **manager** - このデータを報告したパッケージマネージャー（rpm、dpkg、pacman、pkgtool、または portage）
- **version** - パッケージのバージョン
- **size** - 非圧縮時のパッケージサイズ（バイト単位）（利用できない場合は 0 に設定）
- **arch** - パッケージのアーキテクチャ
- **buildtime** - 2 つのエントリを持つオブジェクト:
  - **timestamp** - パッケージがビルドされた時刻の UNIX タイムスタンプ（利用できない場合は 0 に設定）
  - **value** - パッケージがビルドされた日時の人間が読みやすい形式の値（利用できない場合は空文字列に設定）
- **installtime** - 2 つのエントリを持つオブジェクト:
  - **timestamp** - パッケージがインストールされた時刻の UNIX タイムスタンプ（利用できない場合は 0 に設定）
  - **value** - パッケージがインストールされた日時の人間が読みやすい形式の値（利用できない場合は空文字列に設定）

例:

```
[
    {
        "name": "util-linux-core",
        "manager": "rpm",
        "version": "2.37.4-3.el9",
        "size": 1296335,
        "arch": "x86_64",
        "buildtime": {
            "timestamp" : 1653552239,
            "value" : "Sep 20 01:39:40 2021 UTC"
```

```
        },
        "installtime": {
            "timestamp" : 1660780885,
            "value" : "Aug 18 00:01:25 2022 UTC"
        }
    },
    {
        "name": "xfonts-base",
        "manager": "dpkg",
        "version": "1:1.0.5",
        "size": 7337984,
        "arch": "all",
        "buildtime": {
            "timestamp": 0,
            "value": ""
        },
        "installtime": {
            "timestamp": 0,
            "value": ""
        }
    }
]
```

net.dns.get の戻り値

概要

このセクションでは、net.dns.get Zabbix agent 2 アイテムの戻り値の詳細を説明します。

詳細

このアイテムの出力は、アイテムキーで指定されたパラメーターに基づく DNS レコード情報を含むオブジェクトです。

たとえば、net.dns.get[,example.com] アイテムは、拒否されたクエリの次の JSON を返す場合があります。

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

DNS サーバーの IP アドレスを指定すると、net.dns.get[192.0.2.1,example.com] アイテムは次の JSON を返す場合があります。

```
{
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "expire": 1209600,
                "mbox": "noc.dns.example.org.",
                "minttl": 3600,
                "ns": "ns.example.org.",
                "refresh": 7200,
```

```
                "retry": 3600,
                "serial": 2022091378
            },
            "rdlength": 44,
            "ttl": 1205,
            "type": "SOA"
        }
    ],
    "flags": [
        "RA"
    ],
    "query_time": "0.029556",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


接続に問題がある場合、net.dns.get[192.0.2.1,example.com] アイテムはエラーを返す場合があります。


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```


次の種類のエラーコードが可能です。

Scenario	"zbx_error_code"	"zbx_error_msg"
エラーはなく、DNS 応答が受信され解析された。	0	
DNS が停止している。	-1	"Communication error"
JSON の解析中にエラーが発生した	-2	"Received unexpected response"

追加パラメーターを指定すると、net.dns.get[192.0.2.1,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,nsid,edns0,aaf
アイテムは次の JSON を返す場合があります。


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
            ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
```

```
                "a": "192.0.2.1"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


関連情報

DNS レコードの詳細については、以下を参照してください。

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

net.if.get の戻り値

概要

このセクションでは、net.if.get アイテムの戻り値の詳細について説明します。このアイテムは、Linux 上で Zabbix エージェントおよび Zabbix エージェント 2 によってサポートされています。

詳細

このアイテムは、名前が指定された正規表現に一致するネットワークインターフェースに関する詳細情報を返します。出力は、config 配列と values 配列を含む JSON オブジェクトとして構成されます。

出力には、以下のプロパティが含まれます。

- **name** - インターフェース名。/proc/net/dev から読み取られます。
- **ifalias** - インターフェースのエイリアス。/sys/class/net/<name>/ifalias から読み取られます。

- **mac** - インターフェースの MAC アドレス。/sys/class/net/<name>/address から読み取られます。
- **type** - インターフェースのタイプ（physical、virtual、loopback）。
- **speed** - リンク速度（Mbits/sec）。Ethernet 固有で、/sys/class/net/<name>/speed から読み取られます。
- **duplex** - インターフェースの最新または現在の duplex 値(half、full)。Ethernet 固有で、/sys/class/net/<name>/duplex から読み取られます。
- **administrative_state** - 設定された状態（down、up）。/sys/class/net/<name>/flags から読み取られます。
- **operational_state** - インターフェースの RFC2863 運用状態 (unknown、notpresent、down、lowerlayerdown、testing、dormant、up)。sys/class/net/<name>/operstate から読み取られます。
- **carrier** - 物理リンクの有無（0 - 物理リンクがダウン、1 - 物理リンクがアップ）。/sys/class/net/<name>/carrier から読み取られます。
- **carrier_changes** - リンク状態がアップからダウン、またはその逆に変化した総回数。/sys/class/net/<name>/carrier_changes から読み取られます。
- **carrier_up_count** - リンクがアップになった回数。/sys/class/net/<name>/carrier_up_count から読み取られます。
- **carrier_down_count** - リンクがダウンになった回数。/sys/class/net/<name>/carrier_down_count から読み取られます。
- **in** - 受信トラフィック。/proc/net/dev から読み取られます。インターフェースカウンターを含みます。
  - **bytes** - 正常なバイト数。
  - **packets** - 正常なパケット数。
  - **errors** - 不正なパケットの総数。
  - **dropped** - 受信されたが処理されなかったパケット数 + ホストが取りこぼしたパケット数。
  - **overruns** - オーバーフローイベントの回数。
  - **frame** - フレーミングエラーのあるパケット数。
  - **compressed** - 正常に受信された圧縮パケット数。
  - **multicast** - 受信したマルチキャストパケット数。
- **out** - 送信トラフィック。/proc/net/dev から読み取られます。インターフェースカウンターを含みます。
  - **bytes** - 正常なバイト数。
  - **packets** - 正常なパケット数。
  - **errors** - 不正なパケットの総数。
  - **dropped** - 受信されたが処理されなかったパケット数 + ホストが取りこぼしたパケット数。
  - **overruns** - オーバーフローイベントの回数。
  - **collisions** - パケット送信中に発生したコリジョン数。
  - **carrier** - 送信中のキャリア喪失によるフレーム送信エラー数。
  - **compressed** - 送信された圧縮パケット数。

利用できないメトリクスは JSON 出力から除外されます。以下の例では、”lo” インターフェースにエイリアスが設定されていないため、ifalias プロパティは除外されています。

例:


```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
            "carrier": 0
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "type": "physical",
```

```
            "administrative_state": "up",
            "operational_state": "up",
            "carrier": 1
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
```

```
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


関連項目

ネットワークデバイスの sysfs ABI の詳細については、以下を参照してください。

- Linux カーネルのドキュメント

vfs.dev.get の戻り値

概要

このセクションでは、vfs.dev.get アイテムの戻り値の詳細を説明します。このアイテムは、Linux では Zabbix エージェントおよび Zabbix エージェント 2 によって、AIX では Zabbix エージェントによってサポートされています。

詳細

このアイテムは、名前が指定した正規表現に一致するブロックデバイスの詳細情報を返します。

出力は、config と、<mode> パラメーターに応じて values 配列を含む JSON オブジェクトとして構成されます。

config 配列には、選択したモードで利用可能なプロパティが含まれます。values 配列は、ブロックデバイス統計を提供するモードでのみ返されます。

利用できないメトリックは JSON 出力から除外されます。

以下のセクションでは、各モードで返されるプロパティについて説明します。

disks モード

vfs.dev.get[disks,.*] はブロックデバイスの設定データを返します。

プロパティ:

- **name** - ブロックデバイス名;
- **devid** - ブロックデバイス識別子。たとえば WWN ベースまたはカーネル生成のデバイス ID;
- **type** - ブロックデバイスタイプ (disk, partition, rom);
- **path** - ブロックデバイスノードへのパス;

- **model** - デバイスモデル;
- **serial** - デバイスのシリアル番号;
- **wwn** - デバイスの World Wide Name (WWN);
- **size_bytes** - デバイスサイズ（バイト）;
- **logical_block_size** - 論理ブロックサイズ（バイト）;
- **physical_block_size** - 物理ブロックサイズ（バイト）。

Linux の例:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
    }
  ]
}
```


AIX の例:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
```

```
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```

devices mode

`vfs.dev.get[devices,.*]` は、パーティションを含むブロックデバイスの設定データを返します。

プロパティ:

- **name** - ブロックデバイス名。
- **devid** - ブロックデバイス識別子。
- **type** - ブロックデバイスの種類 (`disk`, `partition`, `rom`)。
- **partitions** - デバイス上のパーティション。パーティション名とそのサイズをバイト単位で含みます。

Linux の例:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
    }
  ]
}
```

disk_stats モード

`vfs.dev.get[disk_stats,.*]` は、ブロックデバイスの設定データとディスク統計を返します。

プロパティ:

- **name** - ブロックデバイス名。

- **devid** - ブロックデバイス識別子。
- **type** - ブロックデバイスタイプ (`disk`, `partition`, `rom`)。
- **size_bytes** - デバイスサイズ（バイト）。
- **logical_block_size** - 論理ブロックサイズ（バイト）。
- **stats** - ブロックデバイスの I/O 統計。以下を含みます:
  - **reads_completed** - 完了した読み取り操作数;
  - **writes_completed** - 完了した書き込み操作数;
  - **bytes_read** - 読み取られたバイト数;
  - **bytes_written** - 書き込まれたバイト数;
  - **io_time_ms** - I/O に費やした合計時間（ミリ秒）。

Linux の例


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
```

```
    }
  ]
}
```

AIX の例:

```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    }
```

```
  ]
}
```

device_stats モード

`vfs.dev.get[device_stats,.*]` はブロックデバイスの設定データとデバイス統計を返します。

プロパティ:

- **name** - ブロックデバイス名。
- **devid** - ブロックデバイス識別子。
- **type** - ブロックデバイスタイプ (`disk`, `partition`, `rom`)。
- **size_bytes** - デバイスサイズ（バイト）。
- **stats** - ブロックデバイスの I/O 統計。以下を含みます:
  - **reads_completed** - 完了した読み取り操作数;
  - **writes_completed** - 完了した書き込み操作数;
  - **bytes_read** - 読み取られたバイト数;
  - **bytes_written** - 書き込まれたバイト数;
  - **io_time_ms** - I/O に費やした合計時間（ミリ秒）。

Linux の例:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
      "size_bytes": 1234567890
```

```
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sda1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sda2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
```

```
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "sdb1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    },
    {
      "name": "nvme0n1p1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
```

```
      "name": "nvme0n1p2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```

Windows の **system.cpu.util** アイテムに関する注意事項

system.cpu.util アイテムは、CPU 使用率のパーセンテージを提供します。

Windows 用 Zabbix エージェントでコレクタープロセスが開始されると、N 個の論理プロセッサ (スレッド) ごとに N 個のアイテム用のバッファが初期化されます。値は Windows パフォーマンスカウンターを使用してバッファ内で定期的に更新されます。これらの値は、0 <= n < N の範囲で `system.cpu.util[n]` が要求されたときに報告されます。

バッファ内のもう 1 つのアイテムは、`system.cpu.util[total]` に使用されます。

Zabbix は、NUMA システムとプロセッサグループをサポートしない古い Windows バージョンの両方をサポートするために、異なるパフォーマンスカウンターを自動的に選択します。

論理プロセッサ (スレッド) 数が 64 以下で、プロセッサグループ数が 1 の場合は、次のようになります。

```
N は論理プロセッサ（スレッド）数です。

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

それ以外の場合:

```
G はプロセッサグループ数、N はグループ内の論理プロセッサ（スレッド）数です。

\Processor Information(_Total)\% Processor Time
\Processor Information(0,0)\% Processor Time
\Processor Information(0,1)\% Processor Time
\Processor Information(0,2)\% Processor Time
...
```

```
\Processor Information(0,N-1)\% Processor Time
...
\Processor Information(G-1,0)\% Processor Time
\Processor Information(G-1,1)\% Processor Time
\Processor Information(G-1,2)\% Processor Time
...
\Processor Information(G-1,N-1)\% Processor Time
```

大きな **JSON** 値のサポート

JSON データ型 で大きな値（1MiB 以上）を確実に処理できるようにするには、必要に応じて以下の設定変更を確認し、適用してください。

1. MySQL または MariaDB を使用している場合は、`/etc/mysql/my.cnf` で許可される最大パケットサイズを増やします（一部のシステムでは既定値が 1MB の場合があります）。

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. 必要に応じて、追加の MySQL/MariaDB サーバーシステム変数 および InnoDB システム変数 を調整します。たとえば、次のようにします。

```
innodb_io_capacity=1900
connect_timeout=600
wait_timeout=57600
interactive_timeout=57600
net_read_timeout=7200
net_write_timeout=7200
net-buffer-length=32704
```

3. Zabbix サーバーの設定ファイルで`HistoryCacheSize` を 2G に増やします。

4. Zabbix プロキシを使用している場合は、プロキシの設定ファイルで`HistoryCacheSize` を 2G に、`ProxyMemoryBufferSize` を 2G に増やします。

5. Zabbix エージェントを使用している場合は、Zabbix サーバー/プロキシとエージェント間の通信タイムアウトを増やします（たとえば、アイテムのタイムアウト や、Zabbix のサーバー/プロキシ およびエージェント の設定ファイルにある `Timeout` パラメーター値）。そうしないと、それらの間のメッセージが完了せず、*"message length does not match expected length"* のようなエラーが表示される場合があります。また、必要に応じてスクリプトのタイムアウト やアイテムテスト のタイムアウトなど、他のタイムアウトも確認してください。

サポートされる関数

トリガー式と関数を参照してください。

マクロ

標準で利用可能なサポートされるマクロ および場所ごとにサポートされるユーザーマクロを使用できます。

ロケーションでサポートされるマクロ

概要

場所ごとにサポートされるマクロの表には、特定の場所でサポートされるすべてのマクロ（組み込み、式、およびユーザー定義）の参照が含まれています。

関連するグループのすべてのマクロがサポートされている場合は、グループ名のみが表示されます。各マクロ名またはグループ名をクリックすると、マクロの詳細を確認できます。

マクロの詳細の一覧には、アプリケーション領域ごとにグループ化された、すべての組み込みマクロの詳細な一覧が含まれています。マクロ値をカスタマイズする場合（たとえば、文字列を短縮したり、特定の部分文字列を抽出したりする場合）は、マクロ関数を使用できます。

ロケーションごとにサポートされるマクロ

ロケーション		サポートされるマクロ	コメント
アクション	期間条件、デフォルトのオペレーションステップの継続時間、ステップ継続時間	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
通知とコマンド	トリガーベース	アクションマクロ 日時マクロ {ESC.HISTORY} イベントマクロ、原因/症状イベントマクロ イベント更新マクロ: {EVENT.UPDATE.HISTORY}、 {EVENT.UPDATE.STATUS} 式マクロ: {?EXPRESSION} {FUNCTION.VALUE} ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、 {HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、 {HOST.NAME}、{HOST.PORT} 対象ホストマクロ（コマンドのみ） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}、{ITEM.STATE.ERROR}を除く） プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}、 {TRIGGER.EVENTS.*}、{TRIGGER.EXPRESSION}、 {TRIGGER.EXPRESSION.EXPLAIN}、 {TRIGGER.EXPRESSION.RECOVERY}、{TRIGGER.ID}、 {TRIGGER.HOSTGROUP.NAME}、{TRIGGER.NAME}、 {TRIGGER.NAME.ORIG}、{TRIGGER.NSEVERITY}、 {TRIGGER.SEVERITY}、{TRIGGER.STATUS}、 {TRIGGER.TEMPLATE.NAME}、{TRIGGER.URL}、 {TRIGGER.URL.NAME}、{TRIGGER.VALUE} ユーザー定義マクロ	メッセージでのマクロの使用も参照してください: メッセージ内でのマクロの使用

ロケーション		サポートされるマクロ	コメント
	障害更新	アクションマクロ 日時マクロ {ESC.HISTORY} イベントマクロ、原因/症状イベントマクロ、復旧イベントマクロ イベント更新マクロ: すべて（{EVENT.UPDATE.NSEVERITY}、{EVENT.UPDATE.SEVERITY}を除く） 式マクロ: {?EXPRESSION} 関数マクロ ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、{HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} 対象ホストマクロ（コマンドのみ） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}、{ITEM.STATE.ERROR}を除く） プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}、{TRIGGER.EVENTS.*}、{TRIGGER.EXPRESSION}、{TRIGGER.EXPRESSION.EXPLAIN}、{TRIGGER.EXPRESSION.RECOVERY}、{TRIGGER.ID}、{TRIGGER.HOSTGROUP.NAME}、{TRIGGER.NAME}、{TRIGGER.NAME.ORIG}、{TRIGGER.NSEVERITY}、{TRIGGER.SEVERITY}、{TRIGGER.STATUS}、{TRIGGER.TEMPLATE.NAME}、{TRIGGER.URL}、{TRIGGER.URL.NAME}、{TRIGGER.VALUE} {USER.FULLNAME} ユーザー定義マクロ	
	障害復旧	復旧イベントマクロ イベントマクロ {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
	ディスカバリ	アクションマクロ 日時マクロ ディスカバリマクロ イベントマクロ: {EVENT.AGE}、{EVENT.DATE}、{EVENT.ID}、{EVENT.OBJECT}、{EVENT.SOURCE}、{EVENT.TIME}、{EVENT.TIMESTAMP} 対象ホストマクロ（コマンドのみ） プロキシマクロ	
	自動登録	アクションマクロ 日時マクロ イベントマクロ: {EVENT.AGE}、{EVENT.DATE}、{EVENT.ID}、{EVENT.OBJECT}、{EVENT.SOURCE}、{EVENT.TIME}、{EVENT.TIMESTAMP} ホストマクロ: {HOST.HOST}、{HOST.IP}、{HOST.PORT}、{HOST.METADATA} 対象ホストマクロ（コマンドのみ） プロキシマクロ	
	サービスベース	アクションマクロ 日時マクロ {ESC.HISTORY} イベントマクロ: {EVENT.ACK.STATUS}、{EVENT.OPDATA}を除くすべて サービスマクロ ユーザー定義マクロ	
	サービス復旧	イベントマクロ（{EVENT.ACK.STATUS}、{EVENT.OPDATA}を除く）、復旧イベントマクロ	

ロケーション		サポートされるマクロ	コメント
	サービス更新	アクションマクロ 日時マクロ {ESC.HISTORY} イベントマクロ（{EVENT.ACK.STATUS}、{EVENT.OPDATA}を除く） イベント更新マクロ: {EVENT.UPDATE.DATE}、{EVENT.UPDATE.NSEVERITY}、{EVENT.UPDATE.SEVERITY}、{EVENT.UPDATE.STATUS}、{EVENT.UPDATE.TIME}、{EVENT.UPDATE.TIMESTAMP} サービスマクロ ユーザー定義マクロ	
内部通知		アクションマクロ 日時マクロ {ESC.HISTORY} イベントマクロ（{EVENT.NSEVERITY}、{EVENT.SEVERITY}を除く）、復旧イベントマクロ ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.DESCRIPTION}、{ITEM.DESCRIPTION.ORIG}、{ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG}、{ITEM.NAME}、{ITEM.NAME.ORIG}、{ITEM.VALUETYPE} プロキシマクロ	
	アイテムベース	アイテムマクロ: {ITEM.STATE}、{ITEM.STATE.ERROR}	
	LLD ルールベース	低レベルディスカバリ通知マクロ	
	トリガーベース	トリガーマクロ: {TRIGGER.DESCRIPTION}、{TRIGGER.EXPRESSION}、{TRIGGER.EXPRESSION.RECOVERY}、{TRIGGER.HOSTGROUP.NAME}、{TRIGGER.ID}、{TRIGGER.NAME}、{TRIGGER.NAME.ORIG}、{TRIGGER.NSEVERITY}、{TRIGGER.SEVERITY}、{TRIGGER.STATE}、{TRIGGER.STATE.ERROR}、{TRIGGER.TEMPLATE.NAME}、{TRIGGER.URL}、{TRIGGER.URL.NAME} ユーザー定義マクロ	
アラートスクリプトのパラメーターコネクタ		アラートマクロ ユーザー定義マクロ（グローバルのみ）	

ロケーション		サポートされるマクロ	コメント
	URL、ユーザー名、パスワード、*Bearer* トークン、*HTTP* プロキシ、*SSL* 証明書、*SSL* キーファイル、*SSL* キーパスワードフィールド	ユーザー定義マクロ（グローバルのみ）	
	タイムアウトフィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
グラフ名		式マクロ: {?EXPRESSION}	このロケーションでは、このマクロ内で秒をパラメーターとする **avg**、**last**、**max**、**min** 関数のいずれか 1 つのみがサポートされます（時間の接尾辞を使用できます）。 {HOST.HOST<1-9>} マクロを、このマクロ内のホストとして使用できます（インデックス付きマクロも参照してください）。 使用例: {?avg(/{HOST.HOST}/item.key,1h)} サポートされない使用例: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
ホスト、ホストプロトタイプ	インターフェース *IP/DNS*	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME} ユーザー定義マクロ [1]	[1] IP アドレスの場合、1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません
	インターフェースポート	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。

ロケーション		サポートされるマクロ	コメント
アイテム、アイテムプロトタイプ、**LLD** ルール	*SNMP* コミュニティ *SNMPv3* コンテキスト名、セキュリティ名、認証パスフレーズ、プライバシーパスフレーズ *IPMI* ユーザー名、パスワード	ユーザー定義マクロ	
	名前	ユーザー定義マクロ	LLD ルール名ではサポートされません。
	説明	ユーザー定義マクロ	
	キーのパラメーター	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ユーザー定義マクロ	サポートされる {HOST.*} マクロは、アイテムに対して選択されたインターフェースに解決されます。インターフェースのないアイテムで使用した場合は、優先順位の順に Zabbix エージェント、SNMP、JMX、または IPMI インターフェースに解決されます。ホストにいずれのインターフェースもない場合は'UNKNOWN' に解決されます。 Zabbix trapper アイテムのアイテムキーではマクロはサポートされません。
	更新間隔、カスタム間隔、タイムアウト（サポートされるアイテムタイプで利用可能）、保存期間（履歴/トレンド用）フィールド	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。

ロケーション		サポートされるマクロ	コメント
	失われたリソースを削除、失われたリソースを無効化フィールド	ユーザー定義マクロ	これらのフィールドはLLD ルールでのみサポートされます。 1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	前処理ステップのパラメーター（カスタムスクリプトを含む）、カスタムエラー処理パラメーター（値を設定および エラーを設定フィールド）	ユーザー定義マクロ	
	フィルターの正規表現	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ユーザー定義マクロ	これらのフィールドはLLD ルールでのみサポートされます。
	オーバーライドのフィルター正規表現	ユーザー定義マクロ	これらのフィールドはLLD ルールでのみサポートされます。
	オーバーライド操作: 更新間隔、履歴保存期間、トレンド保存期間（アイテムプロトタイプ用）	ユーザー定義マクロ	これらのフィールドはLLD ルールでのみサポートされます。 1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
アイテムタイプ	ブラウザーアイテムパラメーター名と値	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG} ユーザー定義マクロ	
	スクリプトアイテムパラメーター名と値		
	計算/集計アイテムの式(式定数、関数パラメーター、アイテムキーのパラメーター、(集計アイテムのみ）フィルター条件（ホストグループ名とタグ名）)	ユーザー定義マクロ	

ロケーション		サポートされるマクロ	コメント
	データベース監視アイテム ユーザー名、パスワード、*SQL* クエリフィールド	ユーザー定義マクロ	
	HTTP エージェントアイテム *URL*、クエリフィールド、リクエスト本文、ヘッダー（名前と値）、*SSL* 証明書ファイル、*SSL* キーファイルフィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} アイテムマクロ: {ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG} ユーザー定義マクロ	シークレットマクロを含む URL は、マクロが”******” として解決されるため動作しません。
	HTTP エージェントアイテム *HTTP* プロキシフィールド	アイテムマクロ: {ITEM.ID}、{ITEM.KEY}、{ITEM.KEY.ORIG} ユーザー定義マクロ	
	HTTP エージェントアイテム 必要なステータスコード、*HTTP* 認証のユーザー名/パスワード、*SSL* キーパスワードフィールド	ユーザー定義マクロ	
	HTTP エージェントアイテム 許可されたホストフィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME} ユーザー定義マクロ	
	JMX エージェントアイテム エンドポイントフィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.PORT} ユーザー定義マクロ	
	トラッパーアイテム 許可されたホストフィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP} ユーザー定義マクロ	
	SNMP エージェントアイテム *SNMP OID* フィールド	ユーザー定義マクロ	
	SSH および **Telnet** アイテム スクリプト	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ユーザー定義マクロ	
	SSH エージェントアイテム ユーザー名、公開鍵ファイル、秘密鍵ファイル、パスワードフィールド	ユーザー定義マクロ	
	Telnet エージェントアイテム ユーザー名、パスワードフィールド	ユーザー定義マクロ	
マップ			

ロケーション		サポートされるマクロ	コメント
ネットワークディスカバリ	*URL*、*URL* 名	ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、{HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} {HOSTGROUP.ID} ホストインベントリマクロ マップマクロ {TRIGGER.ID}	
	要素ラベル	式マクロ: {?EXPRESSION} [1] ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、{HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} {HOSTGROUP.ID} ホストインベントリマクロ マップマクロ トリガーマクロ: {TRIGGER.EVENTS.*}、{TRIGGER.ID}、{TRIGGER.PROBLEM.EVENTS.*}、{TRIGGERS.ACK}、{TRIGGERS.PROBLEM.ACK}、{TRIGGERS.PROBLEM.UNACK}、{TRIGGERS.UNACK}	[1] このロケーションでは、このマクロ内で秒をパラメーターとする **avg**、**last**、**max**、**min** 関数のいずれか 1 つのみがサポートされます (時間の接尾辞を使用できます)。 {HOST.HOST<1-9>} マクロを、このマクロ内のホストとして使用できます。 使用例: {?avg(/{HOST.HOST}/item.key,1h)} サポートされない使用例: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
	リンクラベル	式マクロ: {?EXPRESSION} [1]	
	図形ラベル	式マクロ: {?EXPRESSION} [1]	
	マップ図形内のテキストフィールド	{MAP.NAME}	
	更新間隔フィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	SNMP コミュニティ、*SNMP OID* フィールド *SNMPv3* コンテキスト名、セキュリティ名、認証パスフレーズ、プライバシーパスフレーズフィールド	ユーザー定義マクロ（グローバルのみ）	

ロケーション		サポートされるマクロ	コメント
プロキシ			
	アクティブエージェント用アドレス > ポート (プロキシがグループに属する場合) パッシブプロキシモードの場合: インターフェースの アドレスおよび ポートフィールド 上書きモードの アイテムタイプ用タイムアウトフィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
プロキシグループ			
	フェイルオーバー期間、プロキシの最小数フィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
スクリプト			
	コマンド、確認テキスト (タイプ: script、手動のホストアクション)	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.ID} [1]、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ホストインベントリマクロ [2] {MANUALINPUT} ユーザー名マクロ ユーザー定義マクロ	[1] 確認テキストのみ [2] Zabbix サーバーと Zabbix プロキシでサポートされます

ロケーション		サポートされるマクロ	コメント
	コマンド (タイプ: script、手動のイベントアクション)	日時マクロ イベントマクロ、原因/症状イベントマクロ、復旧イベントマクロ (復旧が発生した場合) イベント更新マクロ: {EVENT.UPDATE.HISTORY}、 {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、 {HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、 {HOST.NAME}、{HOST.PORT} ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}、{ITEM.STATE.ERROR}を除く） {MANUALINPUT} プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}、 {TRIGGER.EVENTS.*}、{TRIGGER.EXPRESSION}、 {TRIGGER.EXPRESSION.EXPLAIN}、 {TRIGGER.EXPRESSION.RECOVERY}、 {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}、 {TRIGGER.HOSTGROUP.NAME}、{TRIGGER.ID}、 {TRIGGER.NAME}、{TRIGGER.NAME.ORIG}、 {TRIGGER.NSEVERITY}、{TRIGGER.SEVERITY}、 {TRIGGER.STATUS}、{TRIGGER.TEMPLATE.NAME}、 {TRIGGER.URL}、{TRIGGER.URL.NAME}、{TRIGGER.VALUE} ユーザー名マクロ ユーザー定義マクロ	
	確認テキスト (タイプ: script、手動のイベントアクション)	イベントマクロ: {EVENT.ID}、{EVENT.NAME}、 {EVENT.NSEVERITY}、{EVENT.SEVERITY}、{EVENT.STATUS}、 {EVENT.VALUE} ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、 {HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ホストインベントリマクロ {MANUALINPUT} ユーザー名マクロ ユーザー定義マクロ	
	URL、確認テキスト (タイプ: URL、手動のホスト/イベントアクション)		
タグ名と値		ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、 {HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.LASTVALUE*}、{ITEM.LOG.*}、 {ITEM.VALUE*} トリガーマクロ: {TRIGGER.ID}（トリガータグ値のみ） ユーザー定義マクロ	タグ名と値のマクロは、イベント生成プロセス中にのみ解決されます。
トリガー			
	名前	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、 {HOST.IP}、{HOST.NAME}、{HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}、{ITEM.LOG.*}、 {ITEM.VALUE*} 位置マクロ/参照: $1...$9 ユーザー定義マクロ	

ロケーション		サポートされるマクロ	コメント
	イベント名	日時マクロ: {TIME}、{TIMESTAMP} 式マクロ: {?EXPRESSION} 関数マクロ ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}、{ITEM.LOG.*}、{ITEM.VALUE*} トリガーマクロ: {TRIGGER.EXPRESSION.EXPLAIN}	
	運用データ	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}、{ITEM.LOG.*}、{ITEM.VALUE*} ユーザー定義マクロ	
	式	{TRIGGER.VALUE} ユーザー定義マクロ [1]	[1] 定数と関数パラメーター内のみ。シークレットマクロはサポートされません
	タグの一致、メニュー項目名、メニュー項目 *URL* フィールド	ユーザー定義マクロ	シークレットマクロを含む URL は、マクロが”******” として解決されるため動作しません。
	説明	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}、{ITEM.LOG.*}、{ITEM.VALUE*} ユーザー定義マクロ	
	URL	{EVENT.ID} ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}、{ITEM.LOG.*}、{ITEM.VALUE*} {TRIGGER.ID}	
ユーザーパラメーターコマンド		位置マクロ/参照: $1...$9	
Web シナリオ	名前、エージェント、*HTTP* プロキシフィールド	ユーザー定義マクロ	

ロケーション		サポートされるマクロ	コメント
	更新間隔フィールド	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	変数、ヘッダー、*SSL* 証明書、*SSL* キーファイルフィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ユーザー定義マクロ	
	ユーザー、パスワード、*SSL* キーパスワードフィールド	ユーザー定義マクロ	
	Web シナリオのステップ 名前、変数（値のみ）フィールド	ユーザー定義マクロ	
	Web シナリオのステップ タイムアウト、必要なステータスコードフィールド	ユーザー定義マクロ	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	Web シナリオのステップ *URL*、ヘッダー（名前と値）、必要な文字列フィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ユーザー定義マクロ	
	Web シナリオのステップ *Post* フィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.IP}、{HOST.NAME}、{HOST.PORT}	
webhook			
メディアタイプ			
	パラメーター値	アラートマクロ トリガーベースの障害通知でサポートされるすべてのマクロ（上記参照）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	JavaScript スクリプト、*JavaScript* スクリプトパラメーター名	ユーザー定義マクロ（グローバルのみ）	
	メニュー項目名、メニュー項目 *URL*	{EVENT.TAGS.<tag name>}	
ウィジェット			

ロケーション		サポートされるマクロ	コメント
カスタムテキスト列	ゲージウィジェット 説明フィールド	ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、 {HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、 {HOST.NAME}、{HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.MACROS}（{ITEM.STATE.ERROR}を除く） ユーザー定義マクロ（グローバルのみ）	
	Honeycomb ウィジェット 主要/副ラベルフィールド アイテム値ウィジェット 説明フィールド		
	上位ホストウィジェット テキストデータ列	ホストマクロ: {HOST.CONN}、{HOST.DESCRIPTION}、 {HOST.DNS}、{HOST.HOST}、{HOST.ID}、{HOST.IP}、 {HOST.NAME}、{HOST.PORT} ホストインベントリマクロ ユーザー定義マクロ（グローバルのみ）	
	URL（動的）ウィジェット *URL* フィールド	ホストマクロ: {HOST.CONN}、{HOST.DNS}、{HOST.HOST}、 {HOST.ID}、{HOST.IP}、{HOST.NAME}、{HOST.PORT} ユーザー定義マクロ	
	監視 > 障害	ホストマクロ（{HOST.METADATA}を除く） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE.ERROR}を除く） トリガーマクロ: {TRIGGER.ID} イベントマクロ: {EVENT.ID} ユーザー定義マクロ	
	監視 > ホスト	ホストマクロ（{HOST.METADATA}を除く） ホストインベントリマクロ ユーザー定義マクロ	
	監視 > 最新データ	ホストマクロ（{HOST.METADATA}を除く） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE.ERROR}を除く） ユーザー定義マクロ	
	データ収集 > テンプレート	ホストマクロ: {HOST.DESCRIPTION}、{HOST.HOST}、 {HOST.ID}、{HOST.NAME}	
	データ収集 > ホスト	ホストマクロ（{HOST.METADATA}を除く） ホストインベントリマクロ ユーザー定義マクロ	

ロケーション	サポートされるマクロ	コメント
その他のロケーション		
稼働時間フィールド	ユーザー定義マクロ（グローバルのみ）	1 つのマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
アイテムタイプ用タイムアウトフィールド (グローバル設定)		
ユーザーメディア有効時フィールド	ユーザー定義マクロ（グローバルのみ）	
メールメディアタイプユーザー名、パスワードフィールド	ユーザー定義マクロ（グローバルのみ）	シークレットマクロの使用を推奨します。
スケジュールレポート件名、メッセージフィールド	日時マクロ: {TIME}、{TIMESTAMP}	

マクロの詳細

このリストには、アプリケーション領域ごとにグループ化された各組み込みマクロの詳細が含まれています:

- アクションマクロ
- アラートマクロ
- 日付/時刻マクロ
- ディスカバリーマクロ
- イベントマクロ
  - イベント更新マクロ
  - 原因/症状イベントマクロ
  - リカバリエベントマクロ
- エスカレーションマクロ
- 関数マクロ
- ホストマクロ
  - ターゲットホストマクロ
- ホストグループマクロ
- ホストインベントリマクロ
- アイテムマクロ
- ローレベルディスカバリ通知マクロ
- マップマクロ
- プロキシマクロ
- スクリプトマクロ
- サービスマクロ
- トリガーマクロ
- ユーザー名マクロ

アクションマクロ

{ACTION.ID}

トリガーされたアクションの数値 ID。

{ACTION.NAME}

トリガーされたアクションの名前。

アラートマクロ

{ALERT.MESSAGE}

アクション設定のデフォルトメッセージの値。

{ALERT.SENDTO}

ユーザーメディア設定の送信先の値。

{ALERT.SUBJECT}

アクション設定のデフォルトの件名の値。

日付と時刻のマクロ

{DATE}

現在の日付（yyyy.mm.dd 形式）。

{TIME}

現在の時刻（hh:mm:ss 形式）

{TIMESTAMP}

現在の時刻を UNIX タイムスタンプ形式で表示します。

ディスカバリーマクロ

{DISCOVERY.DEVICE.IPADDRESS}

検出されたデバイスの IP アドレス。
 常に利用可能であり、ホストが追加されるかどうかには依存しません。

{DISCOVERY.DEVICE.DNS}

検出されたデバイスの DNS 名。
 常に利用可能であり、ホストが追加されるかどうかには依存しません。

{DISCOVERY.DEVICE.STATUS}

検出されたデバイスのステータス (UP/DOWN)。

{DISCOVERY.DEVICE.UPTIME}

特定のデバイスのディスカバリーステータスが最後に変更されてからの時間。
 例: 1h 29m 01s。
 DOWN ステータスのデバイスの場合は、ダウンタイムの期間です。

{DISCOVERY.RULE.NAME}

デバイス/サービスの存在または不在を検出したディスカバリールールの名前。

{DISCOVERY.SERVICE.NAME}

検出されたサービスの名前。
 例: HTTP。

{DISCOVERY.SERVICE.PORT}

検出されたサービスのポート。
 例: 80。

{DISCOVERY.SERVICE.STATUS}

検出されたサービスのステータス (UP/DOWN)。

{DISCOVERY.SERVICE.UPTIME}

特定のサービスの検出ステータスが最後に変更されてからの時間。
 例: 1h 29m 01s。
 DOWN ステータスのサービスの場合は、ダウンタイムの期間です。

エスカレーションマクロ

{ESC.HISTORY}

以前に送信された通知のログ、そのエスカレーションステップとステータス (送信済み、進行中、失敗)。

イベントマクロ

{EVENT.ACK.STATUS}

イベントの確認状態（はい/いいえ）。

{EVENT.AGE}

アクションをトリガーしたイベントの経過時間 (秒単位の精度)。
 エスカレーションメッセージで便利です。例: 9m 13s。

{EVENT.DATE}

アクションをトリガーしたイベントの日付です。形式は `yyyy.mm.dd` です。
 例: 2025.04.14.

{EVENT.DURATION}

イベントの継続時間（問題イベントと復旧イベントの時間差）を秒単位で表示します。
 問題の復旧メッセージで便利です。

{EVENT.ID}

アクションをトリガーしたイベントの数値 ID。

{EVENT.NAME}

アクションをトリガーしたイベントの名前。

{EVENT.NSEVERITY}

イベントの深刻度の数値。
 可能な値：*0* - 未分類、*1* - 情報、*2* - 警告、*3* - 軽度障害、*4* - 障害、*5* - 重度障害。

{EVENT.OBJECT}

イベントオブジェクトの数値。
 可能な値: *0* - トリガー, *1* - 検出されたホスト, *2* - 検出されたサービス, *3* - 自動登録, *4* - アイテム, *5* - ローレベルディスカバリルール。

{EVENT.OPDATA}

問題の根本となるトリガーの運用データ。

{EVENT.SEVERITY}

イベントの深刻度の名前。

{EVENT.SOURCE}

イベントソースの数値。
 可能な値: *0* - トリガー, *1* - ディスカバリ, *2* - 自動登録, *3* - 内部, *4* - サービス。

{EVENT.STATUS}

アクションをトリガーしたイベントの言語値。

{EVENT.TAGS}

イベントタグのカンマ区切りリスト。
 タグが存在しない場合は空文字列に展開されます。

{EVENT.TAGSJSON}

イベントタグオブジェクトを含む JSON 配列です。
 タグが存在しない場合は、空の配列に展開されます。

{EVENT.TAGS.<tag name>}

タグ名で参照されるイベントタグの値。
 英数字以外の文字（英語以外のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。

{EVENT.TIME}

アクションをトリガーしたイベントの時刻（`hh:mm:ss` 形式）。
 例: 12:57:53。

{EVENT.TIMESTAMP}

アクションをトリガーしたイベントの UNIX タイムスタンプ。

{EVENT.VALUE}

アクションをトリガーしたイベントの数値。
 可能な値: *1* - 障害、*0* - 復旧。

原因/症状イベントマクロ

{EVENT.CAUSE.ACK.STATUS}

原因イベントの確認ステータス (Yes/No)。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.AGE}

原因イベントの経過時間（秒単位の精度）。
 エスカレーションメッセージで便利です。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.DATE}

原因イベントの日付（yyyy.mm.dd 形式）。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.DURATION}

原因イベントの継続時間（問題イベントと復旧イベントの時間差）を秒単位で表示します。
 問題復旧メッセージで便利です。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.ID}

原因イベントの数値 ID。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.NAME}

原因イベントの名前。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.NSEVERITY}

原因イベントの深刻度の数値。
 可能な値：*0* - 未分類、*1* - 情報、*2* - 警告、*3* - 軽度障害、*4* - 障害、*5* - 重度障害。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.OBJECT}

原因イベントオブジェクトの数値。
 可能な値: *0* - トリガー, *1* - 検出されたホスト, *2* - 検出されたサービス, *3* - 自動登録, *4* - アイテム, *5* - ローレベルディスカバリルール。
 症状イベントのコンテキストで使用されます。

原因となる問題の基礎となるトリガーの運用データ。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.SEVERITY}

原因イベントの深刻度の名前。
 可能な値：*Not classified*、*Information*、*Warning*、*Average*、*High*、*Disaster*。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.SOURCE}

原因イベントソースの数値。
 可能な値：*0* - トリガー、*1* - ディスカバリ、*2* - 自動登録、*3* - 内部。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.STATUS}

原因イベントの文字列値。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGS}

原因イベントタグのカンマ区切りリスト。
 タグが存在しない場合は空文字列に展開されます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGSJSON}

原因イベントのタグオブジェクトを含む JSON 配列。
 タグが存在しない場合は空の配列に展開されます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGS.<tag name>}

タグ名で参照される原因イベントのタグ値。
 英数字以外の文字（英語以外のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TIME}

原因イベントの時刻（hh:mm:ss 形式）。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TIMESTAMP}

原因イベントの UNIX タイムスタンプ。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.UPDATE.HISTORY}

原因となる問題の更新履歴（確認応答など）。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.VALUE}

原因イベントの数値
 可能な値：*1* - 障害、*0* - 復旧
 症状イベントのコンテキストで使用されます。

{EVENT.SYMPTOMS}

症状イベントの一覧です。以下の詳細が含まれます: ホスト名、イベント名、深刻度、経過時間、サービスタグ、および値。
 このマクロは原因イベントのコンテキストで使用され、症状イベントに関する情報を返します。

リカバリーイベントのマクロ

{EVENT.RECOVERY.DATE}

リカバリイベントの日付。`yyyy.mm.dd` 形式。

{EVENT.RECOVERY.ID}

リカバリーイベントの数値 ID。

{EVENT.RECOVERY.NAME}

リカバリーイベントの名前。

{EVENT.RECOVERY.STATUS}

リカバリーイベントの文字列値。

{EVENT.RECOVERY.TAGS}

回復イベントタグのカンマ区切りの一覧です。タグが存在しない場合は、空文字列に展開されます。

{EVENT.RECOVERY.TAGSJSON}

回復イベントのタグオブジェクトを含む JSON 配列です。タグが存在しない場合は、空の配列に展開されます。

{EVENT.RECOVERY.TIME}

復旧イベントの時刻。`hh:mm:ss` 形式。

{EVENT.RECOVERY.TIMESTAMP}

リカバリーイベントの UNIX タイムスタンプ。

{EVENT.RECOVERY.VALUE}

リカバリエベントの数値。

イベント更新マクロ

{EVENT.UPDATE.ACTION}

障害の更新中に実行されたアクションの人間が読める名称です。
 次の値に解決されます: *acknowledged*、*unacknowledged*、*commented*、*changed severity from (original severity) to (updated severity)*、および *closed*（1 回の更新で実行されるアクションの数に応じて異なります）。

{EVENT.UPDATE.ACTIONJSON}

障害の更新中に実行されたアクションの詳細を含む JSON 配列。
 指定可能な JSON プロパティ値:
- true (`acknowledge`、`unacknowledge`、`close`、`unsuppress`、`cause`、`symptom` プロパティの場合);
- < メッセージ文字列 > (`message` プロパティの場合);
- タイムスタンプ (`suppress_until` および `timestamp` プロパティの場合)、または 0 (`suppress_until` プロパティで無期限に抑制されている場合);
- 0, 1, 2, 3, 4, 5 (`old` および `new` の深刻度プロパティの場合)。

 例: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851`

{EVENT.UPDATE.DATE}

イベントの更新（障害確認応答など）の日付。形式は `yyyy.mm.dd` です。

{EVENT.UPDATE.HISTORY}

問題の更新履歴 (アクノレッジなど)。

{EVENT.UPDATE.MESSAGE}

問題の更新メッセージ。

{EVENT.UPDATE.NSEVERITY}

問題の更新操作中に設定された新しいイベントの深刻度の数値。

{EVENT.UPDATE.SEVERITY}

問題の更新操作中に設定された新しいイベントの深刻度の名前。

{EVENT.UPDATE.STATUS}

問題の更新ステータスの数値。
 可能な値：*0* - 問題/復旧イベントによって Webhook が呼び出された、*1* - 更新操作。

{EVENT.UPDATE.TIME}

イベントの更新（障害確認応答など）の時刻。形式は `hh:mm:ss` です。

{EVENT.UPDATE.TIMESTAMP}

イベント更新（確認など）の UNIX タイムスタンプ

関数マクロ

{FUNCTION.VALUE}

イベント発生時点におけるトリガー式内の N 番目のアイテムベースの関数の値。
 最初のパラメーターが `/host/key` の関数のみがカウントされます。

 このマクロは、数値インデックスとともに使用できます。たとえば、{FUNCTION.VALUE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などの関数を参照できます。indexed macros を参照してください。

{FUNCTION.RECOVERY.VALUE}

イベント発生時点における、復旧式内の N 番目のアイテムベース関数の値。*
 最初のパラメーターが `/host/key` の関数のみがカウントされます。

 このマクロは、数値インデックスとともに使用できます。たとえば {FUNCTION.RECOVERY.VALUE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などの関数を参照できます。indexed macros を参照してください。

ホストマクロ

{HOST.CONN}

ホストの設定に応じた、ホストの IP アドレスまたは DNS 名です。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースが 1 つもない場合、このマクロは'UNKNOWN' に解決されます。

 このマクロは、{HOST.CONN<1-9>} のように数値インデックスとともに使用でき、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{HOST.DESCRIPTION}

ホストの説明です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.DESCRIPTION<1-9>} のように数値インデックス付きで使用できます。indexed macros を参照してください。

{HOST.DNS}

ホストの DNS 名。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースが 1 つもない場合、このマクロは'UNKNOWN' に解決されます。

 このマクロは、{HOST.DNS<1-9>} のように数値インデックスとともに使用でき、トリガー式内の 1 番目、2 番目、3 番目などのホストを指します。indexed macros を参照してください。

{HOST.HOST}

技術的なホスト名。
 このマクロが履歴関数 の第 1 引数でプレースホルダーとして使用される場合、マクロ関数 はサポートされません。たとえば、`last(/{HOST.HOST}/{ITEM.KEY})` のようになります。

 このマクロは、{HOST.HOST<1-9>} のように数値インデックスとともに使用して、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定できます。インデックス付きマクロ を参照してください。

{HOST.ID}

ホスト ID。

 このマクロは、{HOST.ID<1-9>} のように数値インデックスとともに使用でき、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定できます。indexed macros を参照してください。

{HOST.IP}

ホストの IP アドレス。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに展開されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにいずれのインターフェースもない場合、このマクロは'UNKNOWN' に展開されます。

 このマクロは、{HOST.IP<1-9>} のように数値インデックス付きで使用でき、トリガー式内の 1 番目、2 番目、3 番目などのホストを指します。indexed macros を参照してください。

{HOST.METADATA}

ホストのメタデータ。

 アクティブエージェントの自動登録でのみ使用されます。

{HOST.NAME}

表示されるホスト名。

 このマクロは、{HOST.NAME<1-9>} のように数値インデックスとともに使用でき、トリガー式内で 1 番目、2 番目、3 番目などのホストを指します。indexed macros を参照してください。

{HOST.PORT}

ホストのエージェントポート。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにいずれのインターフェースもない場合、このマクロは'UNKNOWN' に解決されます。

 このマクロは、{HOST.PORT<1-9>} のように数値インデックスとともに使用して、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロ を参照してください。

ターゲットホストのマクロ

{HOST.TARGET.CONN}

ターゲットホストの IP アドレスまたは DNS 名。

{HOST.TARGET.DNS}

ターゲットホストの DNS 名。

{HOST.TARGET.HOST}

ターゲットホストの技術名。

{HOST.TARGET.IP}

ターゲットホストの IP アドレス。

{HOST.TARGET.NAME}

ターゲットホストの表示名。

ホストグループマクロ

{HOSTGROUP.ID}

ホストグループ ID。

ホストインベントリマクロ

{INVENTORY.ALIAS}

ホストインベントリの *Alias* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.ALIAS<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.ASSET.TAG}

ホストインベントリの *Asset tag* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.ASSET.TAG<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.CHASSIS}

ホストインベントリの *Chassis* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.CHASSIS<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.CONTACT}

ホストインベントリの *Contact* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.CONTACT<1-9>} を使用すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロを参照してください。

{INVENTORY.CONTRACT.NUMBER}

ホストインベントリの 契約番号フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.CONTRACT.NUMBER<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロを参照してください。

{INVENTORY.DEPLOYMENT.STATUS}

ホストインベントリの *Deployment status* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.DEPLOYMENT.STATUS<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.HARDWARE}

ホストインベントリの *Hardware* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.HARDWARE<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロ を参照してください。

{INVENTORY.HARDWARE.FULL}

ホストインベントリの *Hardware (Full details)* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.HARDWARE.FULL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.HOST.NETMASK}

ホストインベントリの ホストのサブネットマスクフィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.HOST.NETMASK<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.HOST.NETWORKS}

ホストインベントリの *Host networks* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.HOST.NETWORKS<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.HOST.ROUTER}

ホストインベントリの *Host router* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.HOST.ROUTER<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.HW.ARCH}

ホストインベントリの *Hardware architecture* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.HW.ARCH<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.HW.DATE.DECOMM}

ホストインベントリの ハードウェア廃止日フィールド。

 このマクロは、{INVENTORY.HW.DATE.DECOMM<1-9>} のように数値インデックスと組み合わせて使用でき、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロ を参照してください。

{INVENTORY.HW.DATE.EXPIRY}

ホストインベントリの *Date hardware maintenance expires* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.HW.DATE.EXPIRY<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.HW.DATE.INSTALL}

ホストインベントリの *Date hardware installed* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.HW.DATE.INSTALL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.HW.DATE.PURCHASE}

ホストインベントリの *Date hardware purchased* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.HW.DATE.PURCHASE<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.INSTALLER.NAME}

ホストインベントリの *Installer name* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.INSTALLER.NAME<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.LOCATION}

ホストインベントリの *Location* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.LOCATION<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.LOCATION.LAT}

ホストインベントリの *Location latitude* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.LOCATION.LAT<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.LOCATION.LON}

ホストインベントリの *Location longitude* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.LOCATION.LON<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.MACADDRESS.A}

ホストインベントリの *MAC address A* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.MACADDRESS.A<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.MACADDRESS.B}

ホストインベントリの *MAC* アドレス *B* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.MACADDRESS.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.MODEL}

ホストインベントリの *Model* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.MODEL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.NAME}

ホストインベントリの *Name* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.NAME<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロを参照してください。

{INVENTORY.NOTES}

ホストインベントリの *Notes* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.NOTES<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.OOB.IP}

ホストインベントリの *OOB IP address* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.OOB.IP<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.OOB.NETMASK}

ホストインベントリの *OOB* サブネットマスクフィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.OOB.NETMASK<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.OOB.ROUTER}

ホストインベントリの *OOB router* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.OOB.ROUTER<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.OS}

ホストインベントリの *OS* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.OS<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロを参照してください。

{INVENTORY.OS.FULL}

ホストインベントリの *OS (詳細)* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.OS.FULL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.OS.SHORT}

ホストインベントリの *OS (Short)* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.OS.SHORT<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.CELL}

ホストインベントリの *Primary POC cell* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.POC.PRIMARY.CELL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.EMAIL}

ホストインベントリの *Primary POC email* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.PRIMARY.EMAIL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.NAME}

ホストインベントリの *Primary POC name* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.PRIMARY.NAME<1-9>} を使用すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.NOTES}

ホストインベントリ内の *Primary POC notes* フィールドです。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.PRIMARY.NOTES<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.PHONE.A}

ホストインベントリの *Primary POC phone A* フィールド。

 このマクロは、{INVENTORY.POC.PRIMARY.PHONE.A<1-9>} のように数値インデックスと組み合わせて使用でき、トリガー式で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.PHONE.B}

ホストインベントリの *Primary POC phone B* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.PRIMARY.PHONE.B<1-9>} のように指定すると、トリガー式内で最初、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.POC.PRIMARY.SCREEN}

ホストインベントリ内の *Primary POC screen name* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.PRIMARY.SCREEN<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.CELL}

ホストインベントリの *Secondary POC cell* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.SECONDARY.CELL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.EMAIL}

ホストインベントリの *Secondary POC email* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.SECONDARY.EMAIL<1-9>} を使用すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.NAME}

ホストインベントリの *Secondary POC name* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.SECONDARY.NAME<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.NOTES}

ホストインベントリの *Secondary POC notes* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.POC.SECONDARY.NOTES<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.PHONE.A}

ホストインベントリの *Secondary POC phone A* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.SECONDARY.PHONE.A<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.PHONE.B}

ホストインベントリの *Secondary POC phone B* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.SECONDARY.PHONE.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.POC.SECONDARY.SCREEN}

ホストインベントリ内の *Secondary POC screen name* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.POC.SECONDARY.SCREEN<1-9>} のように指定すると、トリガー式内で最初、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SERIALNO.A}

ホストインベントリの シリアル番号 *A* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SERIALNO.A<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロを参照してください。

{INVENTORY.SERIALNO.B}

ホストインベントリの シリアル番号 *B* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SERIALNO.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロ を参照してください。

{INVENTORY.SITE.ADDRESS.A}

ホストインベントリの *Site address A* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SITE.ADDRESS.A<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SITE.ADDRESS.B}

ホストインベントリの *Site address B* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SITE.ADDRESS.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SITE.ADDRESS.C}

ホストインベントリの *Site address C* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SITE.ADDRESS.C<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SITE.CITY}

ホストインベントリの *Site city* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SITE.CITY<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SITE.COUNTRY}

ホストインベントリの *Site country* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SITE.COUNTRY<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SITE.NOTES}

ホストインベントリの *Site notes* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SITE.NOTES<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロを参照してください。

{INVENTORY.SITE.RACK}

ホストインベントリの *Site rack location* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SITE.RACK<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SITE.STATE}

ホストインベントリの *Site state/province* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SITE.STATE<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SITE.ZIP}

ホストインベントリの *Site ZIP/postal* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SITE.ZIP<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SOFTWARE}

ホストインベントリの *Software* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SOFTWARE<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SOFTWARE.APP.A}

ホストインベントリの *Software application A* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SOFTWARE.APP.A<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SOFTWARE.APP.B}

ホストインベントリの *Software application B* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SOFTWARE.APP.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.SOFTWARE.APP.C}

ホストインベントリの *Software application C* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SOFTWARE.APP.C<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SOFTWARE.APP.D}

ホストインベントリ内の *Software application D* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SOFTWARE.APP.D<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SOFTWARE.APP.E}

ホストインベントリの *Software application E* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.SOFTWARE.APP.E<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.SOFTWARE.FULL}

ホストインベントリの ソフトウェア（詳細）フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.SOFTWARE.FULL<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

{INVENTORY.TAG}

ホストインベントリの *Tag* フィールド。

 このマクロは、数値インデックス付きで使用できます。たとえば、{INVENTORY.TAG<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。indexed macros を参照してください。

{INVENTORY.TYPE}

ホストインベントリの *Type* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.TYPE<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.TYPE.FULL}

ホストインベントリの *Type (Full details)* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.TYPE.FULL<1-9>} のようにすると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.URL.A}

ホストインベントリの *URL A* フィールド。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{INVENTORY.URL.A<1-9>} のようにすると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.URL.B}

ホストインベントリの *URL B* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.URL.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{INVENTORY.URL.C}

ホストインベントリの *URL C* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.URL.C<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロ を参照してください。

{INVENTORY.VENDOR}

ホストインベントリの *Vendor* フィールド。

 このマクロは、数値インデックスとともに使用できます。たとえば、{INVENTORY.VENDOR<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを参照できます。インデックス付きマクロ を参照してください。

アイテムマクロ

{ITEM.DESCRIPTION}

通知の原因となったトリガー式内の N 番目のアイテムの説明です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{ITEM.DESCRIPTION<1-9>} のように数値インデックスとともに使用できます。indexed macros を参照してください。

{ITEM.DESCRIPTION.ORIG}

通知の原因となったトリガー式内の N 番目のアイテムの元の説明（マクロが未展開の状態）。

 このマクロは、{ITEM.DESCRIPTION.ORIG<1-9>} のように数値インデックスとともに使用して、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロ を参照してください。

{ITEM.ID}

通知の原因となったトリガー式内の N 番目のアイテムの数値 ID。

 このマクロは、{ITEM.ID<1-9>} のように数値インデックスとともに使用して、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定できます。indexed macros を参照してください。

{ITEM.KEY}

通知の原因となったトリガー式における N 番目のアイテムのキー。
 このマクロがhistory function の第 1 引数のプレースホルダーとして使用される場合、Macro functions はサポートされません。たとえば、`last(/{HOST.HOST}/{ITEM.KEY})` です。

 このマクロは、{ITEM.KEY<1-9>} のように数値インデックス付きで使用して、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macros を参照してください。

{ITEM.KEY.ORIG}

通知の原因となったトリガー式内の N 番目のアイテムの元のキー（マクロが展開されていない状態）。

 このマクロは、{ITEM.KEY.ORIG<1-9>} のように数値インデックスとともに使用して、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロ を参照してください。

{ITEM.LASTVALUE}

通知を引き起こしたトリガー式内の N 番目のアイテムの最新値。
 Web インターフェースでは、最新の履歴値が収集されてから *Max history display period* の時間を超えている場合、*UNKNOWN* に解決されます（*Administration > General* メニューセクションで設定）。
 障害名で使用した場合、このマクロは障害イベントを表示しているときに最新のアイテム値へは解決されず、代わりに障害が発生した時点のアイテム値を保持します。
 通知で使用した場合、状況によっては、トリガーが発火した時点の最新アイテム値に解決されないことがあります。たとえば、あるアイテムが”A” と”B” の 2 つの値を短時間で受信し、”A” に対してトリガーが発火した場合、わずかな処理遅延により、通知では最新値として”B” が表示されることがあります。これは、トリガーが発火した時点から通知が作成されるまでの間に最新のアイテム値が変化したためです。これを避けるには、{ITEM.VALUE} マクロを使用できます。これはトリガーが発火した瞬間の値に解決されるため、通知で正しい値が使用されます。
 これは `last(/{HOST.HOST}/{ITEM.KEY})` のエイリアスです。
 テキスト/ログアイテムの解決後の値は、以下の場所で Web インターフェースにより 20 文字に切り詰められます:
- 運用データ;
- トリガーの説明;
- トリガー URL;
- トリガー URL ラベル;
- アイテム値ウィジェットの説明。
 完全な値に解決するには、macro functions を使用できます。サーバー側では値は切り詰められないためです。たとえば: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス付きで使用できます。たとえば {ITEM.LASTVALUE<1-9>} です。indexed macros を参照してください。

{ITEM.LASTVALUE.AGE}

最新のアイテム値の収集からマクロ評価までに経過した時間。
 人間が読みやすい形式で表示されます (例: 1m 45s)。

 このマクロは、数値インデックスとともに使用できます。たとえば {ITEM.LASTVALUE.AGE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.DATE}

最新のアイテム値が収集された日付。
 YYYYMMDD 形式で表示されます。
 サーバーのタイムゾーンは、トリガー名 (*Monitoring* > 障害一覧)、イベント名、およびタグ名と値に使用されます。それ以外の場合は、ユーザーのタイムゾーンが使用されます。

 このマクロは数値インデックスとともに使用できます。たとえば、{ITEM.LASTVALUE.DATE<1-9>} を使用すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すことができます。indexed macros を参照してください。

{ITEM.LASTVALUE.TIME}

最新のアイテム値が収集された時刻。
 HHMMSS 形式で表示されます。
 トリガー名（*Monitoring* > 障害一覧）、イベント名、タグ名およびタグ値ではサーバーのタイムゾーンが使用されます。それ以外のすべての場合では、ユーザーのタイムゾーンが使用されます。

 このマクロは数値インデックスとともに使用できます。たとえば、{ITEM.LASTVALUE.TIME<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LASTVALUE.TIMESTAMP}

最新のアイテム値が収集された時点の UNIX タイムスタンプ。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.LASTVALUE.TIMESTAMP<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.AGE}

ログイベントの経過時間。
1 秒単位までの精度で指定できます。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.LOG.AGE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを示します。indexed macros を参照してください。

{ITEM.LOG.DATE}

ログエントリがログに書き込まれた日付。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{ITEM.LOG.DATE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.EVENTID}

イベントログ内のイベントの ID。
 Windows のイベントログ監視でのみ使用されます。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{ITEM.LOG.EVENTID<1-9>} を指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.NSEVERITY}

イベントログ内のイベントの数値重大度。
 Windows のイベントログ監視でのみ使用されます。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.LOG.NSEVERITY<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.SEVERITY}

イベントログ内のイベントの言語による深刻度。
 Windows のイベントログ監視でのみ使用されます。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.LOG.SEVERITY<1-9>} を指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.SOURCE}

イベントログ内のイベントのソース。
 Windows のイベントログ監視でのみ使用されます。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{ITEM.LOG.SOURCE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.TIME}

ログエントリがログに書き込まれた時刻。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.LOG.TIME<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを参照できます。indexed macros を参照してください。

{ITEM.LOG.TIMESTAMP}

ログエントリがログに書き込まれた時点の UNIX タイムスタンプ。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{ITEM.LOG.TIMESTAMP<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを参照できます。インデックス付きマクロ を参照してください。

{ITEM.NAME}

すべてのマクロが展開されたアイテムの名前。

 このマクロは、数値インデックス付きで使用できます。たとえば、{ITEM.NAME<1-9>} のようにすると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを参照できます。インデックス付きマクロを参照してください。

{ITEM.NAME.ORIG}

アイテムの元の名前（マクロは展開されていない状態）。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.NAME.ORIG<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.STATE}

通知の原因となったトリガー式に含まれる N 番目のアイテムの最新の状態。
 取りうる値: *Not supported*, *Normal*。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.STATE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを参照できます。indexed macros を参照してください。

{ITEM.STATE.ERROR}

アイテムがサポートされなくなった理由の詳細を含むエラーメッセージ。
 アイテムがサポートされない状態になり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。

{ITEM.VALUE}

次のいずれかに解決されます。

- トリガー式内の N 番目のアイテムの、履歴上の（イベント発生時点の）値。トリガー状態の変化の文脈で使用される場合、たとえばイベントの表示や通知の送信時。
- トリガー状態の変化の文脈なしで使用される場合の、トリガー式内の N 番目のアイテムの最新値。たとえば、ポップアップ選択ウィンドウでトリガー一覧を表示する場合。

この場合、{ITEM.LASTVALUE} と同じように動作します。

1 つ目の場合、履歴値がすでに削除されているか、または一度も保存されていない場合は *UNKNOWN* に解決されます。
2 つ目の場合は、Web インターフェースのみで、最新の履歴値が収集されてから *Max history display period* の時間を超えている場合 (*Administration > General* メニューセクションで設定)、*UNKNOWN* に解決されます。
 テキスト/ログアイテムの解決後の値は、次の場所では Web インターフェースによって 20 文字に切り詰められます。
- Operational data;
- Trigger description;
- Trigger URLs;
- Trigger URL labels;
- Description of the item value widget.
 完全な値に解決するには、macro functions を使用できます。サーバー側では値は切り詰められないためです。たとえば: `{{ITEM.VALUE}.regsub("(.*)", \1)}`。

 このマクロは、数値インデックス付きで使用することもできます。たとえば {ITEM.VALUE<1-9>} のようにして、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定できます。indexed macros を参照してください。

{ITEM.VALUE.AGE}

アイテム値の収集からマクロ評価までに経過した時間。
 人間が読みやすい形式で表示されます（例: 1m 45s）。

 このマクロは、数値インデックスとともに使用できます。たとえば {ITEM.VALUE.AGE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すことができます。indexed macros を参照してください。

{ITEM.VALUE.DATE}

アイテムの値が収集された日付。
 YYYYMMDD 形式で表示されます。
 トリガー名（*Monitoring* > 障害一覧）、イベント名、およびタグ名と値ではサーバーのタイムゾーンが使用されます。それ以外のすべての場合ではユーザーのタイムゾーンが使用されます。

 このマクロは数値インデックスとともに使用できます。たとえば、{ITEM.VALUE.DATE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.VALUE.TIME}

アイテムの値が収集された時刻。
 HHMMSS 形式で表示されます。
 サーバーのタイムゾーンは、トリガー名（*Monitoring* > 障害一覧）、イベント名、およびタグ名と値に使用されます。それ以外の場合は、ユーザーのタイムゾーンが使用されます。

 このマクロは数値インデックスとともに使用できます。たとえば、{ITEM.VALUE.TIME<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すことができます。indexed macros を参照してください。

{ITEM.VALUE.TIMESTAMP}

アイテムの値が収集された時点の UNIX タイムスタンプ。

 このマクロは、数値インデックスとともに使用できます。たとえば、{ITEM.VALUE.TIMESTAMP<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.VALUETYPE}

通知を発生させたトリガー式内の N 番目のアイテムの値の型。
 指定可能な値: *0* - 数値浮動小数点、*1* - 文字、*2* - ログ、*3* - 数値符号なし、*4* - テキスト。

 このマクロは、{ITEM.VALUETYPE<1-9>} のような数値インデックスとともに使用して、トリガー式内の最初、2 番目、3 番目などのアイテムを指すことができます。indexed macros を参照してください。

ローレベルディスカバリ通知マクロ

{LLDRULE.DESCRIPTION}

通知を発生させたローレベルディスカバリルールの説明。

{LLDRULE.DESCRIPTION.ORIG}

通知を発生させたローレベルディスカバリルールの元の説明 (マクロが展開されていないもの) です。

{LLDRULE.ID}

通知を発生させたローレベルディスカバリルールの数値 ID。

{LLDRULE.KEY}

通知を発生させたローレベルディスカバリルールのキー。

{LLDRULE.KEYORIG}

通知を発生させたローレベルディスカバリルールの元のキー (マクロ未解決)。

{LLDRULE.NAME}

通知を発生させたローレベルディスカバリルールの名前。

{LLDRULE.NAMEORIG}

通知を発生させたローレベルディスカバリルールの元の名前 (マクロ未解決)。

{LLDRULE.STATE}

低レベルディスカバリルールの最新の状態。
 取り得る値: *Not supported*, *Normal*。

 このマクロは、数値インデックスと組み合わせて使用できます。たとえば、{LLDRULE.STATE<1-9>} のように指定すると、トリガー式内の最初、2 番目、3 番目などのアイテムを指すことができます。indexed macros を参照してください。

低レベルディスカバリルールがサポートされなくなった理由の詳細を含むエラーメッセージ。
 低レベルディスカバリルールがサポートされなくなり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。

マップマクロ

{MAP.ID}

ネットワークマップの ID。

{MAP.NAME}

ネットワークマップ名。

プロキシマクロ

{PROXY.DESCRIPTION}

プロキシの説明です。
 次のいずれかに解決されます。

- トリガー式内の N 番目のアイテムのプロキシ（トリガーに基づく通知の場合）。ここではindexed マクロを使用できます。
- ディスカバリを実行したプロキシ（ディスカバリ通知の場合）。ここでは、インデックスを付けずに {PROXY.DESCRIPTION} を使用します。
- アクティブエージェントが登録したプロキシ(自動登録通知の場合)。ここでは、インデックスを付けずに {PROXY.DESCRIPTION} を使用します。

このマクロは、たとえば {PROXY.DESCRIPTION<1-9>} のように数値インデックスとともに使用でき、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{PROXY.NAME}

プロキシ名。
 次のいずれかに解決されます。

- トリガー式内の N 番目のアイテムのプロキシ（トリガーに基づく通知の場合）。ここではindexed マクロを使用できます。
- ディスカバリを実行したプロキシ（ディスカバリ通知の場合）。ここでは、インデックスを付けずに {PROXY.NAME} を使用します。
- アクティブエージェントが登録したプロキシ（自動登録通知の場合）。ここでは、インデックスを付けずに {PROXY.NAME} を使用します。

このマクロは、{PROXY.NAME<1-9>} のように数値インデックスとともに使用でき、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

スクリプトマクロ

{MANUALINPUT}

スクリプト実行時にユーザーが指定した手動入力値です。

サービスマクロ

{SERVICE.DESCRIPTION}

マクロが解決されたサービスの説明。

{SERVICE.ID}

アクションをトリガーしたサービスの数値 ID。

{SERVICE.NAME}

マクロが解決されたサービス名。

{SERVICE.ROOTCAUSE}

サービスの障害の原因となったトリガーの障害イベントのリスト。重大度とホスト名でソートされます。
 ホスト名、イベント名、重大度、経過時間、サービスタグと値の詳細が含まれます。

{SERVICE.TAGS}

サービスイベントタグのカンマ区切りリスト。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。タグが存在しない場合は、空文字列に展開されます。

{SERVICE.TAGSJSON}

サービスイベントタグオブジェクトを含む JSON 配列。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。タグが存在しない場合は、空の配列に展開されます。

{SERVICE.TAGS.<tag name>}

タグ名で参照されるサービスイベントタグの値。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。
 英数字以外の文字（英語以外のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。

トリガーマクロ

{TRIGGER.DESCRIPTION}

トリガーの説明。
 通知テキストで `{TRIGGER.DESCRIPTION}` が使用されている場合、トリガーの説明でサポートされているすべてのマクロが展開されます。

マップ内のマップ要素、または通知内の現在のイベントを生成したトリガーの認識済みイベントの数。

{TRIGGER.EVENTS.PROBLEM.ACK}

状態に関係なく、すべてのトリガーの認識された障害イベントの数。

{TRIGGER.EVENTS.PROBLEM.UNACK}

状態に関係なく、すべてのトリガーの未確認の障害イベントの数。

マップ内のマップ要素、または通知内の現在のイベントを生成したトリガーに対する未確認イベントの数。

{TRIGGER.EXPRESSION}

トリガーの式。

{TRIGGER.EXPRESSION.EXPLAIN}

部分的に評価されたトリガー式。
 アイテムベースの関数はイベント生成時点で評価され、その結果に置き換えられます。一方、その他の関数は式に記述されたとおりに表示されます。トリガー式のデバッグに使用できます。

{TRIGGER.EXPRESSION.RECOVERY}

トリガーの設定で *OK* イベントの生成が「リカバリー式」に設定されている場合は、トリガーのリカバリー式。それ以外の場合は空文字列を返します。

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

部分的に評価されたトリガー復旧式。
 アイテムベースの関数はイベント生成時点の結果に評価されて置き換えられますが、それ以外のすべての関数は式に記述されたとおりに表示されます。トリガー復旧式のデバッグに使用できます。

{TRIGGER.HOSTGROUP.NAME}

トリガーが定義されているホストグループのリストを、SQL クエリでソートし、カンマとスペースで区切ったものです。

{TRIGGER.ID}

このアクションをトリガーしたトリガーの数値 ID。

{TRIGGER.NAME}

マクロが解決されたトリガー名。
 アクションでは、`{EVENT.NAME}` を使用して、マクロが解決されたトリガーイベント/問題名を表示できることに注意してください。

{TRIGGER.NAME.ORIG}

元のトリガー名（マクロ未解決）。

{TRIGGER.NSEVERITY}

数値で表したトリガーの深刻度。
 指定可能な値: *0* - 未分類, *1* - 情報, *2* - 警告, *3* - 軽度障害, *4* - 障害, *5* - 重度障害。

問題状態のトリガーに対して認識された問題イベントの数。

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

障害状態のトリガーに対する未確認の障害イベントの数。

{TRIGGER.SEVERITY}

トリガーの深刻度名。
 管理 > 一般設定 > トリガーの表示オプションで定義できます。

{TRIGGER.STATE}

トリガー式の最新の状態。
 使用可能な値: *Unknown*, *Normal*。

トリガーがサポートされなくなった理由の詳細を含むエラーメッセージ。
 トリガーがサポートされなくなり、すぐに再びサポートされるようになった場合、エラーフィールドは空になることがあります。

{TRIGGER.STATUS}

オペレーションステップ実行時のトリガーの値。
 指定可能な値: *Problem*、*OK*。

{TRIGGER.TEMPLATE.NAME}

トリガーが定義されているテンプレートのリストを SQL クエリでソートし、カンマ区切りで表示します。トリガーがホストに定義されている場合は *UNKNOWN* となります。

{TRIGGER.URL}

トリガーの URL。

{TRIGGER.URL.NAME}

トリガー URL のラベル。

{TRIGGER.VALUE}

現在の数値のトリガー値。
 可能な値: *0* - トリガーは正常、*1* - トリガーは障害。

{TRIGGERS.ACK}

トリガーの状態に関係なく、マップエレメントに対して認識されたトリガーの数。
 トリガーのすべての障害イベントが認識されている場合、そのトリガーは認識されたと見なされます。

{TRIGGERS.PROBLEM.ACK}

マップ要素の認識済み障害トリガーの数。
 トリガーの障害イベントがすべて認識済みの場合、そのトリガーは認識済みとみなされます。

{TRIGGERS.PROBLEM.UNACK}

マップ要素の未確認の障害トリガーの数。
 トリガーの障害イベントのうち少なくとも 1 つが未確認の場合、そのトリガーは未確認とみなされます。

{TRIGGERS.UNACK}

トリガーの状態に関係なく、マップ要素の未確認トリガーの数。
 トリガーの障害イベントのうち、少なくとも 1 つが未確認の場合、そのトリガーは未確認とみなされます。

ユーザー名マクロ

{USER.FULLNAME}

イベントの確認を追加した、またはスクリプトを開始したユーザーの名前、姓、ユーザー名。

{USER.NAME}

スクリプトを開始したユーザーの名前。

{USER.SURNAME}

スクリプトを開始したユーザーの姓。

{USER.USERNAME}

スクリプトを開始したユーザーのユーザー名。

インデックス付きマクロ

インデックス付きマクロの構文 **{MACRO<1-9>}** は、トリガーの *Expression* フィールド内の N 番目のアイテム、関数、またはホストのみを参照できます。

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** は、トリガー式に含まれる 1 番目、2 番目、3 番目のホストの IP アドレスに解決されます（存在する場合）。
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** は、イベント発生時点のトリガー式に含まれる 1 番目、2 番目、3 番目のアイテムの値に解決されます（存在する場合）。
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** は、イベント発生時点の 1 番目、2 番目、3 番目のアイテムベースの関数の値に解決されます（存在する場合）。

トリガーの文脈では、インデックス付きマクロは常にトリガー設定の *Expression* フィールドを参照し、*Recovery expression* は参照しません。たとえば、復旧イベントでは、**{ITEM.VALUE2}** は復旧時点の障害式に含まれる 2 番目のアイテムの値に解決されます。

{HOST.HOST<1-9>} マクロは、グラフ名内の `{?func(/host/key,param)}` 式マクロでもサポートされています。たとえば、グラフ名内の `{?func(/{HOST.HOST2}/key,param)}` は、グラフ内の 2 番目のアイテムのホストに解決されます。

Warning:
インデックス付きマクロは、ここで挙げたケース以外の文脈では解決されません。それ以外の文脈では、インデックスなしのマクロ（**{HOST.HOST}, {HOST.IP}** など）を使用してください。

ロケーションでサポートされるユーザーマクロ

概要

このセクションには、ユーザー定義可能なマクロがサポートされている場所のリストが含まれています。

Note:
アクション、ネットワークディスカバリ、プロキシ、およびこのページのその他の場所セクションにリストされているすべての場所では、グローバルレベルのユーザーマクロのみがサポートされます。これらの場所では、ホストレベルおよびテンプレートレベルのマクロは解決されません。

Note:
マクロ値をカスタマイズするには（たとえば、短縮したり特定の部分文字列を抽出したりする場合）、マクロ関数を使用できます。

アクション

アクションでは、ユーザーマクロを次のフィールドで使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
トリガーベースの通知とコマンド		yes
トリガーベースの内部通知		yes
障害の更新通知		yes
サービスベースの通知とコマンド		yes
サービス更新通知		yes
期間		no
オペレーション		
	デフォルトの操作ステップ期間	no
	ステップ期間	no

ホスト/ホストプロトタイプ

ホストおよびホストプロトタイプの設定では、ユーザーマクロを以下のフィールドで使用できます。

場所		複数のマクロ/テキストとの混在[1]
インターフェース IP/DNS		DNS のみ
インターフェースポート		不可
SNMP v1, v2		
	SNMP コミュニティ	可
SNMP v3		
	コンテキスト名	可

場所		複数のマクロ/テキストとの混在[1]
	セキュリティ名	可
	認証パスフレーズ	可
	プライバシーパスフレーズ	可
IPMI		
	ユーザー名	可
	パスワード	可
タグ[2]		
	タグ名	可
	タグ値	可

アイテム / アイテムのプロトタイプ

アイテムまたはアイテムのプロトタイプの設定では、ユーザーマクロを次のフィールドで使用できます。

場所	
アイテム名	
アイテムキーのパラメータ	
更新間隔	
カスタム間隔	
タイムアウト（サポートされているアイテムタイプで使用可能）	
保存期間（ヒストリとトレンド）	
説明	
計算 / 集計アイテム	
	数式（式の定数と関数パラメータ、アイテムキーのパラメータ、（集計アイテ
データベースモニター	
	ユーザー名
	パスワード
	SQL クエリ
HTTP エージェント	
	URL[3]
	クエリフィールド
	リクエストボディ
	ヘッダー（名前と値）
	必須ステータスコード
	HTTP プロキシ
	HTTP 認証ユーザー名
	HTTP 認証パスワード
	SSL 証明書ファイル
	SSL キーファイル
	SSL キーパスワード
	許可されたホスト
JMX エージェント	
	JMX エンドポイント
スクリプトアイテム	
	パラメータ名と値
ブラウザアイテム	
	パラメータ名と値
Telemetry クエリアイテム	
	タイムシフト
	ルックバック制限
	粒度
SNMP エージェント	
	SNMP OID
SSH エージェント	
	ユーザー名
	公開鍵ファイル
	秘密鍵ファイル
	パスワード
	スクリプト
TELNET エージェント	

場所	
	ユーザー名
	パスワード
	スクリプト
Zabbix トラッパー	
	許可されたホスト
タグ[2]	
	タグ名
	タグの値
前処理ステップ	
	パラメータ（カスタムスクリプトを含む）
	カスタムエラー処理パラメータ（値を設定およびエラー時に設定フィールド）

ローレベルディスカバリ

ローレベルディスカバリルールでは、ユーザーマクロを以下のフィールドで使用できます。

場所		複数のマクロ/テキス
キーのパラメータ		はい
更新間隔		いいえ
カスタム間隔		いいえ
タイムアウト（サポートされているアイテムタイプで利用可能）		いいえ
失われたリソースの削除		いいえ
失われたリソースの無効化		いいえ
説明		はい
SNMP エージェント		
	SNMP OID	はい
SSH エージェント		
	ユーザー名	はい
	公開鍵ファイル	はい
	秘密鍵ファイル	はい
	パスワード	はい
	スクリプト	はい
TELNET エージェント		
	ユーザー名	はい
	パスワード	はい
	スクリプト	はい
Zabbix トラッパー		
	許可されたホスト	はい
データベースモニタ		
	ユーザー名	はい
	パスワード	はい
	SQL クエリ	はい
JMX エージェント		
	JMX エンドポイント	はい
HTTP エージェント		
	URL[3]	はい
	クエリフィールド	はい
	リクエストボディ	はい
	ヘッダー（名前と値）	はい
	必要なステータスコード	はい
	HTTP 認証のユーザー名	はい
	HTTP 認証のパスワード	はい
フィルター		
	正規表現	はい
オーバーライド		
	フィルター: 正規表現	はい
	操作: 更新間隔（アイテムのプロトタイプ用）	いいえ
	操作: 履歴保存期間（アイテムのプロトタイプ用）	いいえ
	操作: トレンド保存期間（アイテムのプロトタイプ用）	いいえ

ネットワークディスカバリ

ネットワークディスカバリルール では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
監視間隔		no
SNMP v1, v2		
	SNMP コミュニティ	yes
	SNMP OID	yes
SNMP v3		
	コンテキスト名	yes
	セキュリティ名	yes
	認証パスフレーズ	yes
	プライバシーパスフレーズ	yes
	SNMP OID	yes

プロキシ

プロキシの設定では、ユーザーマクロを次のフィールドで使用できます。

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (プロキシがグループに属している場合)	no
Interface address and port (passive proxy の場合)	no
アイテムタイプのタイムアウト	no

プロキシグループ

プロキシグループの設定では、ユーザーマクロを次のフィールドで使用できます。

Location	Multiple macros/mix with text[1]
フェイルオーバー期間	no
プロキシの最小数	no

テンプレート

テンプレートの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
タグ[2]		
	タグ名	yes
	タグ値	yes

トリガー

トリガーの設定では、以下のフィールドでユーザーマクロを使用できます。

場所		複数マクロ/テキストとの混在[1]
名前		可
オペレーショナルデータ		可
式（定数および関数パラメータのみ。シークレットマクロは非対応）		可
一致用タグ		可
メニューエントリ名		可
メニューエントリ URL[3]		可
説明		可
タグ[2]		
	タグ名	可
	タグ値	可

Web シナリオ

Web シナリオの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
名前		yes
監視間隔		no
エージェント		yes
HTTP プロキシ		yes
変数 (値のみ)		yes
ヘッダー (名前と値)		yes
ステップ		
	名前	yes
	URL[3]	yes
	変数 (値のみ)	yes
	ヘッダー (名前と変数)	yes
	タイムアウト	no
	要求文字列	yes
	要求ステータスコード	no
認証		
	ユーザー	yes
	パスワード	yes
	SSL 証明書ファイル	yes
	SSL 秘密鍵ファイル	yes
	SSL 秘密鍵パスワード	yes
タグ[2]		
	タグ名	yes
	タグ値	yes

その他の場所

ここに一覧されている場所に加えて、ユーザーマクロは次のフィールドでも使用できます。

場所	複数のマクロ/テキストとの混在[1]
グローバルスクリプト(URL、スクリプト、SSH、Telnet、IPMI)、確認テキストを含むwebhook	yes
JavaScript スクリプト	no
JavaScript スクリプトパラメーター名	no
JavaScript スクリプトパラメーター値	yes
ダッシュボード	
Top hosts ダッシュボードウィジェットのデータ型 *Text* の列	yes
Item value および *Gauge* ダッシュボードウィジェットの *Description* パラメーター	yes
Honeycomb ダッシュボードウィジェットのプライマリ/セカンダリラベル *Text* パラメーター	yes
URL ダッシュボードウィジェットの *URL*[3] パラメーター	yes
監視 > 障害、ホスト、最新データ	
カスタムテキスト列	yes

場所		複数のマクロ/テキストとの混在[1]
データ収集>テンプレート、ホスト>アラート>メディアタイプ>メッセージテンプレート	カスタムテキスト列	yes
	件名	yes
	メッセージ	yes
アラート>メディアタイプ>スクリプト	スクリプトパラメーター	yes

場所		複数のマクロ/テキストとの混在[1]
アラート > メディアタイプ > メディアタイプ	*Email* メディアタイプの *Username* および *Password* フィールド（*Authentication* が”Username and password” に設定されている場合; secret macros の使用を推奨）	yes
ユーザー > ユーザー > メディア	有効時	yes
管理 > 一般 > *GUI*	稼働時間	no
管理 > 一般 > タイムアウト	アイテムタイプのタイムアウト	no

場所		複数のマクロ/テキストとの混在[1]
管理>一般>コネクター		
	URL	yes
	Username	yes
	Password	yes
	Bearer token	yes
	Timeout	no
	HTTP proxy	yes
	SSL certificate file	yes
	SSL key file	yes
	SSL key password	yes

Zabbix でサポートされるすべてのマクロの完全な一覧については、supported macros を参照してください。

脚注

[1] あるフィールドで複数のマクロ、またはマクロとテキストの混在がその場所でサポートされていない場合は、1 つのマクロでフィールド全体を埋める必要があります。

[2] タグ名と値で使用されるマクロは、イベント生成プロセス中にのみ解決されます。

[3] secret macro を含む URL は機能しません。これは、その中のマクロが”******” として解決されるためです。

[4] Zabbix trapper アイテムのアイテムキーではマクロはサポートされていません。

バージョン互換性

サポートされるエージェント

Zabbix 8.0 と互換性を持たせるには、Zabbix エージェントはバージョン 1.6 以降、8.0 以前である必要があります。

一部のパラメータが変更されているため、古いエージェントの設定を確認する必要がある場合があります。たとえば、3.0 より前のバージョンでは、ログ に関連するパラメータが変更されています。

最新の機能、メトリクス、パフォーマンスの向上、メモリ使用量の削減を最大限に活用するには、サポートされている最新のエージェントを使用してください。

Windows XP に関する注意事項

- 32 ビット版 Windows XP では、6.0.x より新しい Zabbix エージェントを使用しないでください。
- Windows XP/Server 2003 では、Zabbix 4.0.x より新しいエージェントテンプレートを使用しないでください。新しいテンプレートでは英語のパフォーマンスカウンターが使用されていますが、これは Windows Vista/Server 2008 以降でのみサポートされています。

サポートされているエージェント 2

バージョン 4.4 以降の古い Zabbix agent 2 は Zabbix 8.0 と互換性があります。Zabbix agent 2 は 8.0 より新しいバージョンを使用しないでください。

Zabbix agent 2 のバージョン 4.4 および 5.0 を使用する場合、サポートされていないアイテムの更新にはデフォルトの 10 分間隔が使用されることに注意してください。

最新の機能、メトリクス、パフォーマンスの向上、メモリ使用量の削減を最大限に活用するには、最新のサポートされている agent 2 を使用してください。

サポートされている Zabbix プロキシ

Zabbix 8.0 との完全な互換性を保つには、プロキシのメジャーバージョンがサーバーと一致している必要があります。Zabbix 8.0.x プロキシのみが、Zabbix 8.0.x サーバーと完全に互換性があります。

古いプロキシは部分的にサポートされています。データの収集やスクリプトの実行は可能ですが、新しいアイテムなどの設定更新を受け取ることはできません。

Zabbix サーバーとの関係において、プロキシは以下のように分類されます。

- 最新（プロキシとサーバーが同じメジャーバージョンの場合）
- 旧式（プロキシのバージョンがサーバーより古いが、一部サポートされている場合）
- サポート外（プロキシのバージョンがサーバーの前 LTS リリースバージョンより古いまたはプロキシのバージョンがサーバーのメジャーバージョンより新しい場合）

例：

サーバーバージョン	最新プロキシバージョン	旧式プロキシバージョン	サポート外プロキシバージョン
6.4	6.4	6.0, 6.2	6.0 より古い; 6.4 より新しい
7.0	7.0	6.0, 6.2, 6.4	6.0 より古い; 7.0 より新しい
7.2	7.2	7.0	7.0 より古い; 7.2 より新しい
7.4	7.4	7.0	7.0 より古い; 7.4 より新しい
8.0	8.0	7.0, 7.2, 7.4	7.0 より古い; 8.0 より新しい

プロキシでサポートされる機能:

プロキシバージョン	データ更新	設定更新	タスク
Current	はい	はい	はい
Outdated	はい	いいえ	リモートコマンド（例: シェルスクリプト）； アイテム値の即時チェック（つまり、今すぐ実行）； 注: 前処理の実際の値を使ったテストはサポートされていません。
Unsupported	いいえ	いいえ	いいえ

互換性のないデーモン起動された Zabbix プロセスのバージョンがログに警告メッセージとして出力されます。

サポートされている XML ファイル

バージョン 1.8 以降の XML ファイルが Zabbix 8.0 でのインポートにサポートされています。

Attention:
XML エクスポート形式では、トリガーの依存関係は名前のみで保存されます。同じ名前のトリガーが複数存在し（たとえば、異なる深刻度や式を持つ）、それらの間に依存関係が定義されている場合、それらをインポートすることはできません。そのような依存関係は、XML ファイルから手動で削除し、インポート後に再追加する必要があります。

時間期間の構文

概要

日時を設定するには、次の形式を使用する必要があります。

`d-d,hh:mm-hh:mm`

記号は次のことを表します。

記号	説明
d	曜日: 1 - 月、2 - 火、...、7 - 日
hh	時: 00-24
mm	分: 00-59

セミコロン (;) をセパレーターとして複数の期間を指定できます。

d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...

空白ではデフォルト値である 1-7,00:00-24:00 になります。

Attention:
指定時間の上限は含まれていません。したがって、09:00-18:00 を指定すると、最後の 1 秒は 17:59:59 になります。

例

営業時間月～金 9:00～18:00

1-5,09:00-18:00

平日営業時間と週末営業時間月～金 9:00～18:00 土～日 10:00～16:00

1-5,09:00-18:00; 6-7,10:00-16:00

単位記号

概要

86400、104857600、1000000 のような大きな値を扱うのは難しく、エラーの原因となることがあります。設定を簡素化し、可読性を向上させるために、Zabbix は値の乗数として機能する単位記号（サフィックス）をサポートしています。

サフィックスなしのトリガー式:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

サフィックス付きのトリガー式:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

サフィックスは、アイテムやウィジェットなど他のエンティティの設定を簡素化したり、アイテム値を人間が読みやすい形式で表示したりするのにも役立ちます。

Note:
エンティティの設定フィールドがサフィックスをサポートしているかどうかを確認するには、常に設定するエンティティの関連ページを参照してください。

時間のサフィックス

Zabbix Web インターフェースでは、エンティティの設定で次の時間サフィックスをサポートしています:
 s - 秒 *(*使用した場合、未加工の値と同じように動作します*)*;
 m - 分;
 h - 時間;
 d - 日;
 w - 週;
 M - 月 *(*トレンド関数のみ*)*;
 y - 年 *(*トレンド関数のみ*)*。

時間サフィックスは整数値に対してのみサポートされます。たとえば、1h はサポートされますが、1,5h や 1.5h はサポートされません。代わりに 90m を使用してください。

メモリサイズの接尾辞

Zabbix は次のメモリサイズの接尾辞をサポートしています:
 K - キロバイト
 M - メガバイト
 G - ギガバイト
 T - テラバイト

アイテム値のサフィックス

サフィックスは、数値のアイテム値を人間が読みやすい形式で表示するためにも使用できます。

これを有効にするには、アイテムを設定する際に、*Units* フィールドで次のいずれかのサフィックスを使用します:
 B - バイト;
 Bps - 1 秒あたりのバイト数;
 s - 秒。最大 3 つの最も大きい非ゼロの時間単位で表示;
 uptime - 経過時間を hh:mm:ss または N days, hh:mm:ss 形式で表示;
 unixtime - Unix タイムスタンプ。yyyy.mm.dd hh:mm:ss 形式で表示。

これらのサフィックスの解釈と表示には、次の追加ルールが適用されます:

- B と Bps では、Zabbix は JEDEC 標準に従い、2 進ベースの変換を使用します (1K = 1024B)。
- その他の単位 (Hz、W など) では、Zabbix は 10 進ベースの変換を使用します (1K = 1000)。
- s (秒) では:

- 形式には yyy mmm ddd hhh mmm sss ms が含まれます。表示されるのは、最大 3 つの最も大きい非ゼロの時間単位のみです (例: 1M 10d 4h)。
- 単位が 0 で、かつ 2 つの非ゼロ単位の間にある場合は省略されます (例: 10d 0h 56m ではなく 10d 56m)。

Units を使用すると、次の倍率サフィックスがアイテム値に自動的に適用されます:
 K, **M**, **G**, **T** - kilo, mega, giga, tera;
 P, **E**, **Z**, **Y** - peta, exa, zetta, yotta *(これらは Web インターフェースでのみ適用されます)*。

単位変換を防ぐには、! プレフィックスを使用します (例: !B または!s)。

次の例は、受信したアイテム値が指定された単位に基づいてどのように変換されるかを示しています:

```
1 B → 1 B
1024 B → 1 KB
1536 B → 1.5 KB
881764 B → 881.76 KB
881764 !B → 881764 B

0.0000155 s → 0.016ms
3470400 s → 1M 10d 4h
2606400 s → 1M 4h
2592000 s → 1M
2592001 s → 1M
2592001 !s → 2592001 s

17764 uptime → 04:56:04
86400 uptime → 1 day, 00:00:00
881764 uptime → 10 days, 04:56:04
32417764 uptime → 375 days, 04:56:04
32417764 !uptime → 32417764 uptime

881764 unixtime → 1970-01-11 04:56:04 AM

17764 Hz → 17.76 KHz
86400 Hz → 86.4 KHz
881764 Hz → 881.76 KHz
32417764 Hz → 32.42 MHz

0 ! → 0
0 !! → 0 !
```

Note:
Zabbix 4.0 より前は、ms、rpm、RPM、% からなるハードコードされた単位の停止リストがありました。この停止リストは非推奨となっているため、そのような単位の変換を防ぐ正しい方法は !ms、!rpm、!RPM、!% です。

コマンド実行

Zabbix は、外部チェック、ユーザーパラメータ、system.run アイテム、カスタムアラートスクリプト、リモートコマンド、およびグローバルスクリプトに共通の機能を使用します。

実行手順

Note:
デフォルトでは、Zabbix のすべてのスクリプトは *sh* シェルを使用して実行され、デフォルトのシェルを変更することはできません。別のシェルを利用するには、回避策としてスクリプトファイルを作成し、コマンド実行時にそのスクリプトを呼び出すことができます。

コマンド/スクリプトは、Unix および Windows プラットフォームの両方で同様に実行されます:

1. Zabbix（親プロセス）が通信のためのパイプを作成する
2. Zabbix が、作成される子プロセスの出力としてパイプを設定する
3. Zabbix が子プロセスを作成する（コマンド/スクリプトを実行する）
4. 子プロセスのために新しいプロセスグループ（Unix の場合）またはジョブ（Windows の場合）が作成される

5. Zabbix は、タイムアウトが発生するか、他端で誰も書き込んでいない（すべてのハンドル/ファイルディスクリプタが閉じられた）場合までパイプから読み込む。子プロセスがさらにプロセスを作成し、それらが終了またはハンドル/ファイルディスクリプタを閉じる前に終了することがある点に注意してください。
6. タイムアウトに達していない場合、Zabbix は最初の子プロセスが終了するかタイムアウトが発生するまで待機する
7. 最初の子プロセスが終了し、タイムアウトに達していない場合、Zabbix は最初の子プロセスの終了コードをチェックし、0 と比較する（0 以外の値は実行失敗と見なされる。これは、Zabbix サーバーおよび Zabbix プロキシで実行されるカスタムアラートスクリプト、リモートコマンド、ユーザースクリプトにのみ該当）
8. この時点で、すべてが完了したと見なされ、プロセスツリー全体（つまり、プロセスグループまたはジョブ）が終了する

Attention:
Zabbix は、最初の子プロセスが終了し、かつ他のプロセスが出力ハンドル/ファイルディスクリプタを開いたままにしていない場合に、コマンド/スクリプトの処理が完了したと見なします。処理が完了すると、作成されたすべてのプロセスが終了します。

コマンド内のすべてのダブルクォートとバックスラッシュはバックスラッシュでエスケープされ、コマンドはダブルクォートで囲まれます。

終了コードの確認

終了コードは、以下の条件で確認されます。

- Zabbix サーバーおよび Zabbix プロキシで実行されるカスタムアラートスクリプト、リモートコマンド、ユーザースクリプトにのみ適用されます。
- 0 以外の終了コードは、すべて実行失敗と見なされます。
- 失敗した実行の標準エラー出力および標準出力の内容は収集され、Web インターフェースで利用可能になります（実行結果が表示される場所）。
- Zabbix エージェント/プロキシ上で実行されるリモートコマンドについては、agent/proxy の LogRemoteCommands パラメータを有効にすることで、追加のログエントリを作成できます。

失敗したコマンド/スクリプトに対する Web インターフェースのメッセージおよびログエントリの例:

- 失敗した実行の標準エラー出力および標準出力の内容（存在する場合）。
- "Process exited with code: N."（出力が空で、終了コードが 0 以外の場合）。
- "Process killed by signal: N."（シグナルによってプロセスが終了した場合、Linux のみ）。
- "Process terminated unexpectedly."（不明な理由でプロセスが終了した場合）。

関連項目

- 外部チェック
- ユーザーパラメータ
- system.run アイテム
- カスタムアラートスクリプト
- リモートコマンド
- グローバルスクリプト

エスケープの例

概要

このページでは、さまざまなコンテキストで正規表現を使用する際に正しいエスケープを使用する例を示します。

Note:
トリガー式コンストラクターを使用すると、正規表現に正しいエスケープが自動的に追加されます。

例

コンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}`

注意:

- バックスラッシュはエスケープされません;
- 引用符はエスケープされます。

アイテムキーのパラメータ内のマクロ関数

正規表現: `.+:(\d+)$<br>` アイテムキー: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

注意:

- regsub マクロ関数内の正規表現はダブルクォートで囲まれています（閉じ括弧を含むため）;
- 正規表現の周囲の引用符はエスケープされます（アイテムの 3 番目のパラメータ全体がダブルクォートで囲まれているため）;
- アイテムキーの 3 番目のパラメータはカンマを含むためダブルクォートで囲まれています。

LLD マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

注意:

- バックスラッシュはエスケープされません;
- 引用符はエスケープされます。

ユーザーマクロコンテキスト内の **LLD** マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` コンテキスト付きユーザーマクロ: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

注意:

- LLD のバックスラッシュのエスケープは変わりません;
- LLD マクロをユーザーマクロコンテキストに挿入する際、文字列にする必要があります:

1. マクロ式の周囲に引用符が追加されます;
2. 引用符がエスケープされます。合計で 3 つの新しいバックスラッシュが追加されます。

関数の文字列パラメータ（任意）

例として `concat` を使用します。

文字列内容: `\.+\"[a-z]+<br>` 式: `concat("abc", "\\.\\\"[a-z]+")`

注意:

- 文字列パラメータはバックスラッシュと引用符の両方のエスケープが必要です。

関数の文字列パラメータ内の **LLD** マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` 式: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

注意:

- 文字列パラメータはバックスラッシュと引用符の両方のエスケープが必要です;
- 文字列がアンコートされた後にマクロが解決されるため、もう 1 段階のエスケープが追加されます;

関数の文字列パラメータ内のコンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` 式: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

注意:

- 前の例と同様に、追加のエスケープレイヤーが必要です;
- バックスラッシュと引用符は、最上位のエスケープ（文字列パラメータであることによる）でのみエスケープされます。

関数内のユーザーマクロコンテキスト内の **LLD** マクロ関数

正規表現: `\.+\"([a-z])+<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` コンテキスト付きユーザーマクロ: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` 式: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

3 つのエスケープレイヤーに注意してください:

1. LLD マクロ関数用（バックスラッシュのエスケープなし）;
2. ユーザーマクロコンテキスト用（バックスラッシュのエスケープなし）;
3. 関数の文字列パラメータ用（バックスラッシュのエスケープあり）。

文字列内のコンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` 何らかの式の文字列内、例えば: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

注意:

- 文字列もバックスラッシュのエスケープが必要です;
- 文字列も引用符のエスケープが必要です;
- 再び 2 段階のエスケープのケース:

1. バックスラッシュのエスケープなしでユーザーマクロコンテキスト用のエスケープ;
2. バックスラッシュのエスケープありで文字列用のエスケープ。

Zabbix とは

Zabbix は Alexei Vladishev によって作成され、現在は Zabbix SIA によって積極的に開発およびサポートされています。

Zabbix は、エンタープライズクラスのオープンソース分散監視ソリューションです。

Zabbix は、ネットワークの多数のパラメータに加え、サーバー、仮想マシン、アプリケーション、サービス、データベース、Web サイト、クラウドなどの状態と整合性を監視するソフトウェアです。Zabbix は柔軟な通知メカニズムを使用しており、ほぼすべてのイベントに対してメールベースのアラートを設定できます。これにより、サーバーの障害に迅速に対応できます。Zabbix は、保存されたデータに基づく優れたレポート機能とデータ可視化機能を提供します。このため、Zabbix はキャパシティプランニングに最適です。

Zabbix はポーリングとトラッピングの両方をサポートしています。Zabbix のすべてのレポートと統計情報、および設定パラメータには、Web ベースの Web インターフェースからアクセスできます。Web ベースの Web インターフェースにより、ネットワークの状態とサーバーの健全性をどこからでも確認できます。適切に設定された Zabbix は、IT インフラストラクチャの監視において重要な役割を果たします。これは、数台のサーバーを持つ小規模な組織にも、多数のサーバーを持つ大企業にも当てはまります。

Zabbix は無償で利用できます。Zabbix は AGPL-3.0 ライセンスの下で作成および配布されています。これは、ソースコードが自由に配布され、一般に公開されていることを意味します。

商用サポートは、Zabbix 社および世界各地のパートナーによって提供されています。

Zabbix の機能について詳しく説明します。

Zabbix のユーザー　世界中のさまざまな規模の多くの組織が、主要な監視プラットフォームとして Zabbix を利用しています。

アーキテクチャ　Zabbix は、いくつかの主要なソフトウェアコンポーネントで構成されています。それぞれの役割を以下に示します。

サーバー

Zabbix server は、エージェントが可用性、整合性情報、および統計を報告する中央コンポーネントです。サーバーは、すべての設定データ、統計データ、および運用データが保存される中央リポジトリです。

データベースストレージ

Zabbix によって収集されたデータと同様に、すべての設定情報はデータベースに保存されます。

Web インターフェース

どこからでも、どのプラットフォームからでも Zabbix に簡単にアクセスできるように、Web ベースのインターフェースが提供されています。このインターフェースは Zabbix サーバーの一部であり、通常は（必ずしもそうではありませんが）サーバーを実行しているのと同じ物理マシン上で動作します。

プロキシ

Zabbix proxy は、Zabbix サーバーの代わりにパフォーマンスおよび可用性データを収集できます。プロキシは Zabbix の導入における任意の構成要素ですが、単一の Zabbix サーバーの負荷を分散するうえで非常に有効です。

エージェント

Zabbix エージェントは、監視対象に導入され、ローカルのリソースやアプリケーションを能動的に監視し、収集したデータを Zabbix サーバーに報告します。Zabbix 4.4 以降では、利用可能なエージェントには 2 種類あります。Zabbix agent（軽量で、多くのプラットフォームでサポートされ、C で記述）と Zabbix agent 2（柔軟性が高く、プラグインによる拡張が容易で、Go で記述）です。

データフロー　さらに一歩引いて、Zabbix 内の全体的なデータフローを確認することも重要です。データを収集するアイテムを作成するには、まずホストを作成する必要があります。Zabbix のもう一方の側面では、トリガーを作成するにはまずアイテムが必要です。アクションを作成するにはトリガーが必要です。したがって、*Server X* の CPU 負荷が高すぎるというアラートを受け取りたい場合は、まず *Server X* のホストエントリを作成し、次にその CPU を監視するアイテムを作成し、CPU が高すぎる場合に発動するトリガーを設定し、その後でメールを送信するアクションを作成する必要があります。手順が多いように見えるかもしれませんが、テンプレートを使用すれば実際にはそれほど複雑ではありません。ただし、この設計により、非常に柔軟な構成を作成することが可能になります。

機能　Zabbix は、1 つのパッケージで多彩な機能を提供する、高度に統合されたネットワーク監視ソリューションです。

データ収集

- 可用性とパフォーマンスのチェック
- SNMP（トラップとポーリングの両方）、IPMI、JMX、VMware 監視のサポート
- カスタムチェック
- 任意の間隔で必要なデータを収集
- サーバー/プロキシおよびエージェントによって実行

柔軟なしきい値定義

- バックエンドデータベースの値を参照する、トリガーと呼ばれる非常に柔軟な障害しきい値を定義できます

高度に設定可能な通知

- 通知の送信は、エスカレーションスケジュール、受信者、メディアタイプに応じてカスタマイズできます
- マクロ変数を使用して、通知を意味のある有用なものにできます
- 自動アクションにはリモートコマンドが含まれます

リアルタイムグラフ作成

- 監視対象のアイテムは、組み込みのグラフ作成機能を使用して即座にグラフ化されます

Web 監視機能

- Zabbix は、Web サイト上でのマウスクリックのシミュレーションパスをたどり、機能と応答時間を確認できます

豊富な可視化オプション

- 複数のアイテムを 1 つのビューにまとめられるカスタムグラフを作成可能
- ネットワークマップ
- ダッシュボード形式の概要でのスライドショー
- レポート
- 監視リソースの高レベル（ビジネス）ビュー

履歴データの保存

- データはデータベースに保存されます
- 履歴の保持期間を設定可能
- 組み込みのハウスキーピング処理

簡単な設定

- 監視対象デバイスをホストとして追加
- ホストはデータベースに登録されると監視対象として取得されます
- 監視対象デバイスにテンプレートを適用

テンプレートの使用

- テンプレート内でチェックをグループ化
- テンプレートは他のテンプレートを継承できます

ネットワークディスカバリ

- ネットワークデバイスの自動検出
- エージェントの自動登録
- ファイルシステム、ネットワークインターフェース、SNMP OID の検出

高速な **Web** インターフェース

- PHP で作成された Web ベースの Web インターフェース
- どこからでもアクセス可能
- クリック操作だけで利用可能
- 監査ログ

Zabbix API

- Zabbix API は、一括操作、サードパーティ製ソフトウェアとの統合、その他の目的のために、Zabbix へのプログラム可能なインターフェースを提供します。

権限システム

- 安全なユーザー認証
- 特定のユーザーを特定のビューに制限可能

多機能で拡張しやすいエージェント

- 監視対象に導入
- Linux と Windows の両方に導入可能

バイナリデーモン

- パフォーマンスと小さなメモリフットプリントのために C で記述
- 移植が容易

複雑な環境にも対応

- Zabbix プロキシを使用してリモート監視を容易に実現

用語集　このセクションでは、Zabbix で一般的に使用される用語の意味を説明します。

ホスト

- 物理デバイスまたは仮想デバイス、アプリケーション、サービス、その他の監視対象パラメータの論理的に関連する集合。

ホストグループ

- ホストの論理的なグループ。
ホストグループは、異なるユーザーグループにホストへのアクセス権を割り当てる際に使用されます。

アイテム

- ホストから取得したい特定のデータ、つまりデータのメトリクス。

値の前処理

- データベースに保存する前に行う、受信したメトリクス値の変換。

トリガー

- 障害のしきい値を定義し、アイテムで受信したデータを「評価」するために使用される論理式。

受信したデータがしきい値を上回ると、トリガーは正常から障害の状態になります。
受信したデータがしきい値を下回ると、トリガーは正常の状態を維持するか、正常の状態に戻ります。

テンプレート

- *1* つまたは複数のホストに適用できる、エンティティ（アイテム、トリガー、グラフ、ローレベルディスカバリルール、*Web* シナリオ）のセット。

テンプレートは、ホスト上での監視タスクの展開を迅速化し、監視タスクに対する一括変更を容易にするために使用されます。
テンプレートは個々のホストに直接リンクされます。

テンプレートグループ

- テンプレートの論理的なグループ。
テンプレートグループは、異なるユーザーグループにテンプレートへのアクセス権を割り当てる際に使用されます。

イベント

- トリガーの状態変更やディスカバリ、エージェントの自動登録など、注意を要する事象が *1* 回発生したもの。

イベントタグ

- イベントに付ける事前定義済みのマーカー。
イベント相関や権限の細分化などに使用できます。

イベント相関

- 障害とその解決を柔軟かつ正確に関連付ける方法。

たとえば、あるトリガーによって報告された障害を、別のトリガーによって解決できるように定義できます。別のトリガーでは、異なるデータ収集方法を使用することもできます。

障害

- 「障害」状態になっているトリガー。

障害の更新

- コメントの追加、確認、深刻度の変更、手動でのクローズなど、*Zabbix* が提供する障害管理オプション。

アクション

- イベントに対応するための事前定義済みの手段。

アクションは、オペレーション（通知の送信など）と、オペレーションを実行する条件で構成されます。

エスカレーション

- アクション内でオペレーションを実行するためのカスタムシナリオ。通知の送信やリモートコマンドの実行を順番に行います。

マクロ

- コンテキストに応じて特定の値に解決される変数のプレースホルダー。

メディア

- 通知を配信する手段、つまり配信チャネル。

通知

- 選択したメディアチャネルを介してユーザーに送信される、イベントに関するメッセージ。

リモートコマンド

- 特定の条件が成立したときに、監視対象のホスト上で自動的に実行される事前定義済みのコマンド。

***Web* シナリオ**

- *Web* サイトの可用性を確認するための、*1* つまたは複数の *HTTP* リクエスト。

***Web* インターフェース**

- *Zabbix* に付属する *Web* インターフェース。

ダッシュボード

- 重要な情報の概要や視覚化を、ウィジェットと呼ばれる視覚単位で表示する、カスタマイズ可能な *Web* インターフェースのセクション。

ウィジェット

- 特定の種類とソースの情報（概要、マップ、グラフ、時計など）を表示する視覚単位で、ダッシュボードで使用されます。

Zabbix API

- *Zabbix API* を使用すると、*JSON RPC* プロトコルによって *Zabbix* オブジェクト（ホスト、アイテム、グラフなど）の作成、更新、取得や、その他のカスタムタスクを実行できます。

***Zabbix* サーバー**

- 監視を実行し、*Zabbix* プロキシおよびエージェントと連携し、トリガーを計算し、通知を送信する *Zabbix* ソフトウェアの中央プロセス。データの中央リポジトリでもあります。

***Zabbix* プロキシ**

- *Zabbix* サーバーに代わってデータを収集し、サーバーの処理負荷を軽減できるプロセス。

***Zabbix* エージェント**

- 監視対象に配置され、ローカルリソースとアプリケーションをアクティブに監視するプロセス。

Zabbix* エージェント *2

- ローカルリソースとアプリケーションをアクティブに監視するための新世代の *Zabbix* エージェント。監視用のカスタムプラグインを使用できます。

> **Attention:**
> Zabbix エージェント 2 は Zabbix エージェントと多くの機能を共有しているため、機能の動作が同じ場合、ドキュメント内の「Zabbix エージェント」という用語は Zabbix エージェントと Zabbix エージェント 2 の両方を指します。
> 機能が異なる場合にのみ、Zabbix エージェント 2 と明記します。

暗号化

- Transport Layer Security（TLS）プロトコルを使用した、Zabbix コンポーネント（サーバー、プロキシ、エージェント、zabbix_sender および zabbix_get ユーティリティ）間の暗号化通信のサポート。

エージェントの自動登録

- *Zabbix* エージェント自体がホストとして登録され、監視を開始する自動化されたプロセス。

ネットワークディスカバリ

- ネットワークデバイスを自動的に検出する機能。

ローレベルディスカバリ

- 特定のデバイス上の低レベルエンティティ（ファイルシステム、ネットワークインターフェースなど）を自動的に検出する機能。

ローレベルディスカバリルール

- デバイス上の低レベルエンティティを自動的に検出するための定義のセット。

アイテムプロトタイプ

- 特定のパラメータを変数として持ち、ローレベルディスカバリに使用できるメトリクス。
ローレベルディスカバリの後、変数は検出された実際のパラメータに自動的に置き換えられ、メトリクスは自動的にデータ収集を開始します。

トリガープロトタイプ

- 特定のパラメータを変数として持ち、ローレベルディスカバリに使用できるトリガー。
ローレベルディスカバリの後、変数は検出された実際のパラメータに自動的に置き換えられ、トリガーは自動的にデータの評価を開始します。

その他の *Zabbix* エンティティのプロトタイプもローレベルディスカバリで使用されます。グラフプロトタイプ、ホストプロトタイプ、ホストグループプロトタイプなどがあります。

Zabbix Cloud

Zabbix Cloud

Zabbix Cloud で利用可能な設定オプションと、組織およびノードのアクセス制御についてご確認ください。

現在、Zabbix Cloud ノードでは **Zabbix 7.0** が稼働しています。
 Zabbix 7.0 の詳細については、公式ドキュメントを参照してください。

Zabbix Cloud のセットアップと管理

Zabbix Cloud の利用を開始する　初期設定や監視対象デバイスの接続など、Zabbix Cloud インスタンスを迅速に稼働させるためのガイド付き手順。

ノードの設定　アクセス、データ処理、リソース設定に焦点を当てて、Zabbix Cloud ノードのセットアップ方法を学びます。

ユーザーの追加　Zabbix Cloud 組織およびそのノード内でユーザーを追加し、ロールを割り当てる方法について説明します。

Zabbix Cloud について学ぶ

Zabbix Cloud とオンプレミス　2 つの導入方法の管理、拡張性、保守性の違いを比較し、どちらがワークフローをよりシームレスにサポートできるかを学びましょう。

Zabbix Cloud を探る　Zabbix Cloud の概要、主な機能、特典、サブスクリプションに含まれる内容について説明します。セットアップや使用方法に関するよくある質問については、FAQ をご覧ください。

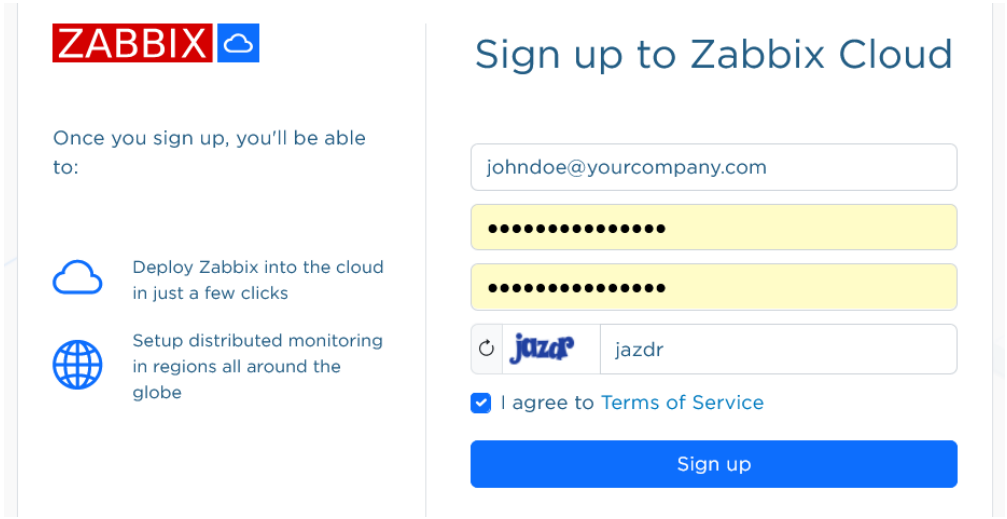
クラウドで **Zabbix** をデプロイする

Zabbix をクラウドで開始するには、次の手順に従います。

1. cloud.zabbix.comにログインするか、サインアップします。
2. クラウドノードを作成します。
3. クラウドで Zabbix の使用を開始します。

ログインまたはサインアップ　既存の Google、GitHub、または Microsoft アカウントを使用して、cloud.zabbix.comにログインできます。

または、サインアップページに移動して登録します。



サインアップすると、新しいアカウントの確認メールが届きます。

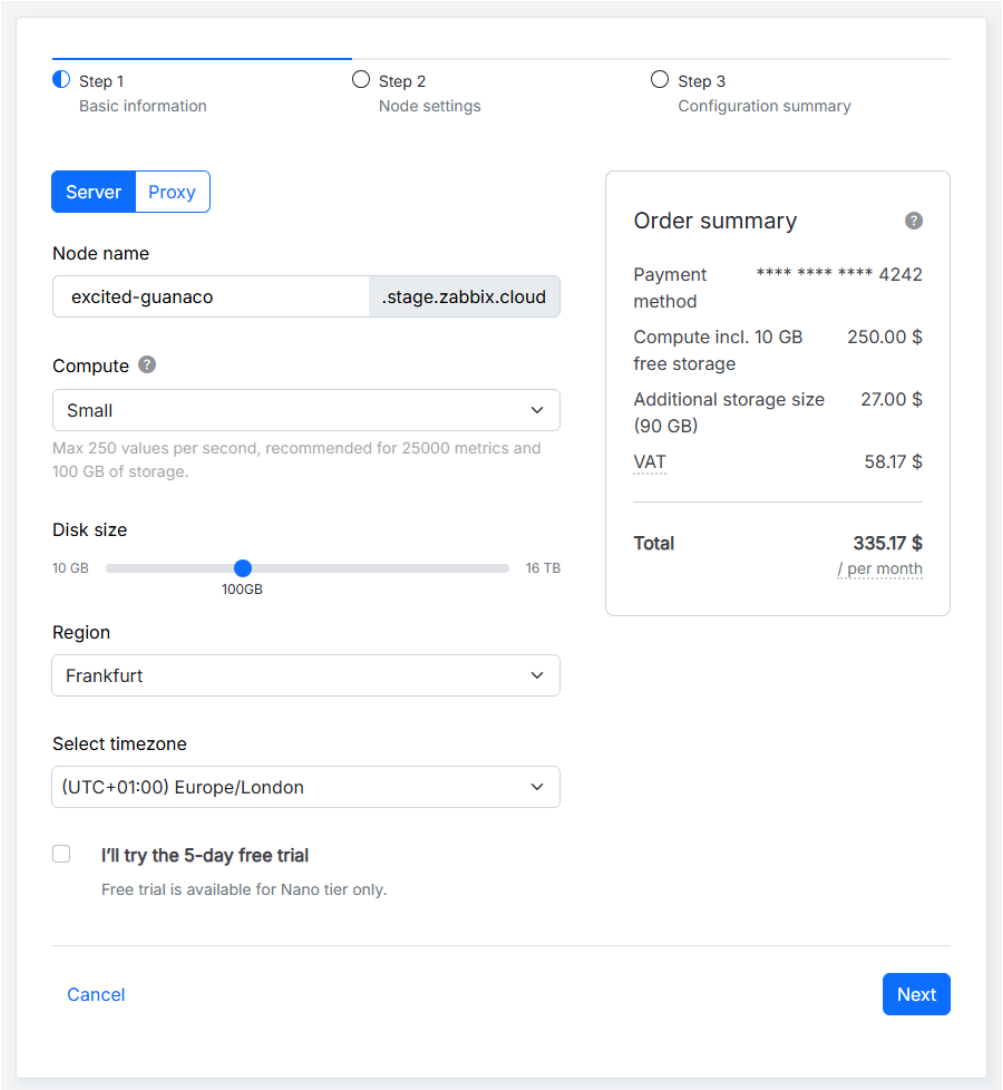
cloud.zabbix.comに移動してログインします。ログイン時にワンタイムパスワードの入力を求められます。このパスワードはメールで送信されます。

クラウドノードの作成　クラウドノードは、Zabbix Cloud にデプロイされた Zabbix インスタンスです。サーバーノードまたはプロキシノードとしてデプロイできます。

ログインすると、デフォルトでアカウントに *Private* 組織がすでに存在していることを確認できます。デフォルトの組織名は後から変更できます。

Create New Node をクリックします。3 段階のノード設定フォームが開きます。

最初のステップでは、基本情報を選択します。



Parameter	Description
Server/Proxy	Zabbix サーバーインスタンスと Zabbix プロキシインスタンスのどちらをデプロイするかを選択します。
Node name	ノード名は `<your-name>.zabbix.cloud` 形式の一意のサブドメインとなり、ノードのサービスへのアクセスに使用されます。
Compute	ノードのコンピューティング能力を選択します。 必要なリソースレベルを判断するには、料金の詳細を参照してください。より高い能力が必要になった場合は、いつでもアップグレードできます。トライアルではないノードを作成する前に、請求方法の設定が必要です。*Billing* メニューから請求情報を追加してください。 *Nano* ティアのノードを試すために、5 日間の無料トライアルを利用できます。トライアル期間中は課金されませんが、期間終了後に有料ノードへ変換しない場合、ノードは削除されます。
Disk size	スライダーには、ティアに基づく推奨最小値がデフォルトで設定されています。 大量の履歴またはトレンドを保存する場合は、値を増やしてください。ディスク容量が大きいほど月額料金が高くなることに注意してください。また、自分でディスク容量を縮小することはできません。縮小するにはサポートに問い合わせる必要があります。
Region	ノードをデプロイするリージョンを選択します。
Select timezone	Zabbix サーバーのタイムゾーンを選択します。このタイムゾーンは Zabbix サーバーで使用され、メンテナンスおよびユーザープロファイルのタイムゾーンとは独立しています。

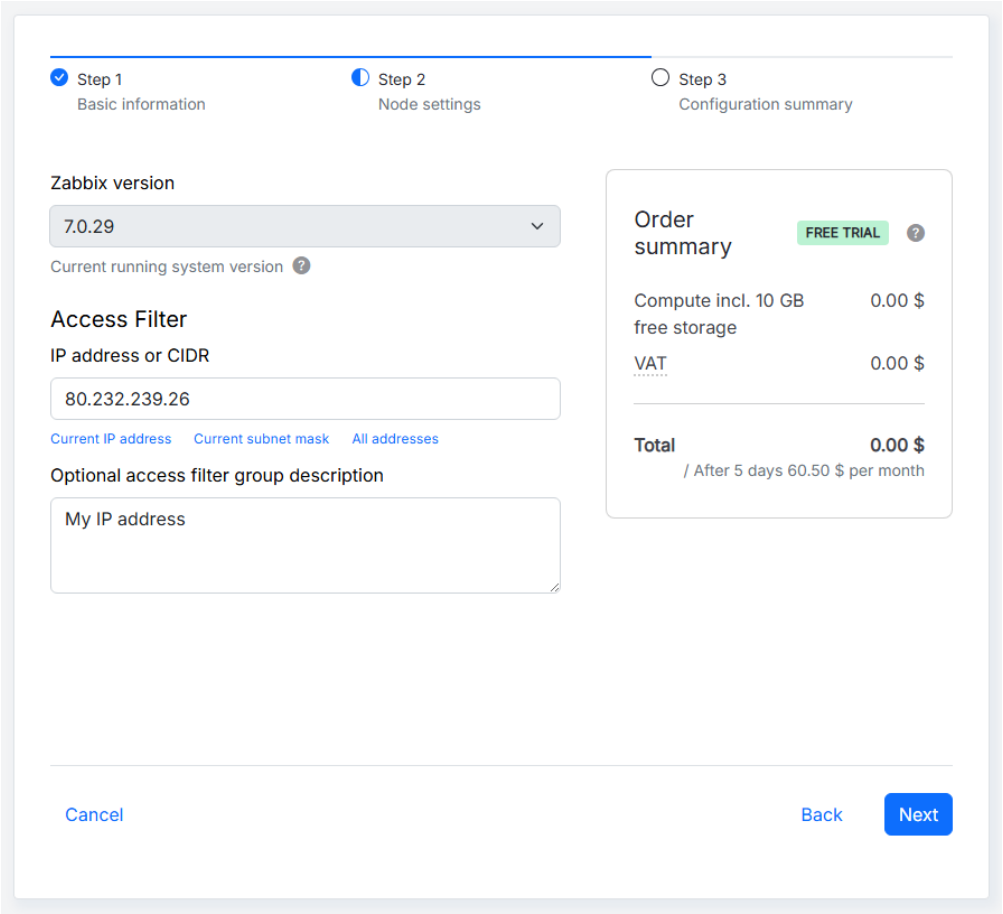
設定が完了したら、*Next* をクリックします。

2 番目のステップでは、ノード設定を選択します。

2 番目のステップで使用できる設定オプションは、サーバーノードとプロキシノードのどちらを作成するかによって異なります。

サーバーノードの設定

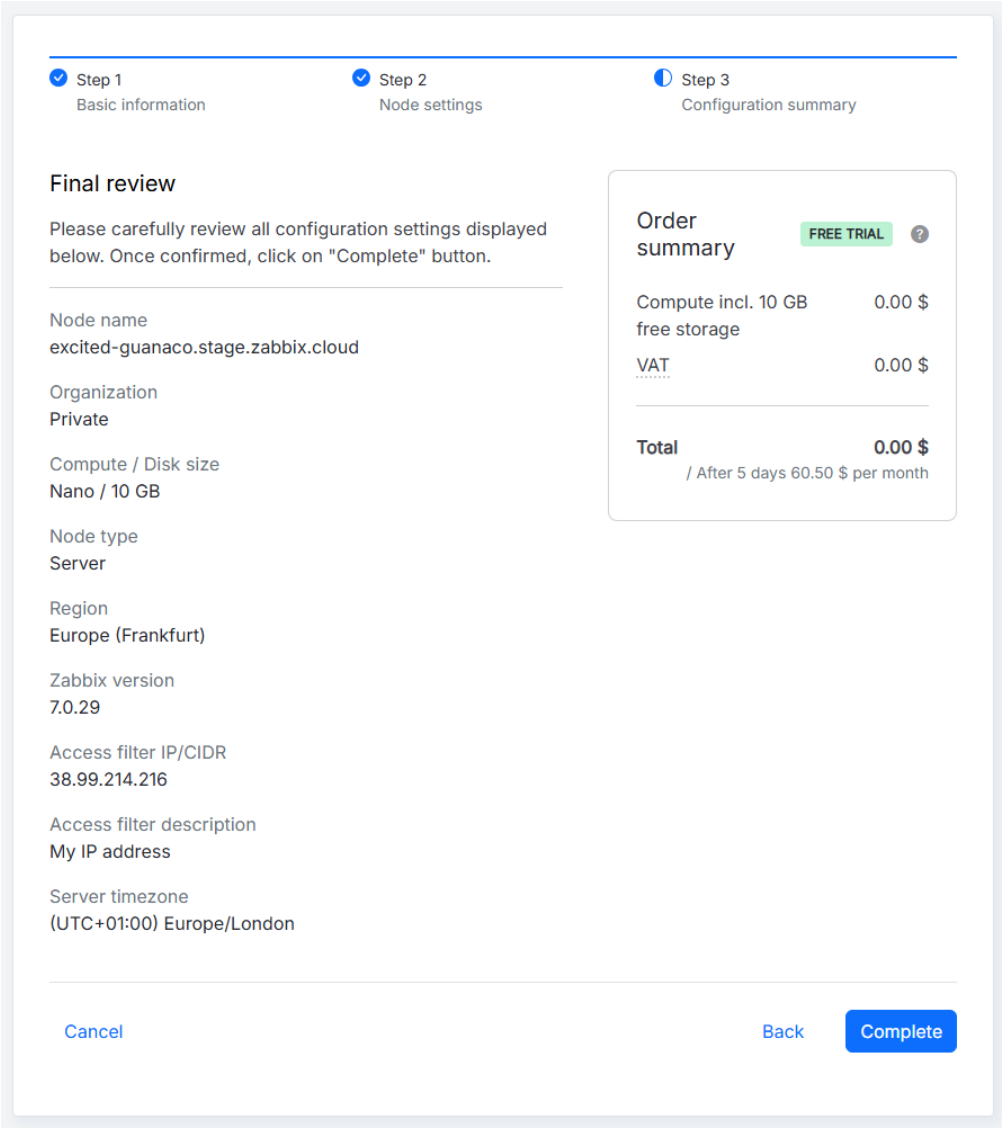
サーバーノードの作成時には、次のパラメータを使用できます。



Parameter	Description
Zabbix version	Zabbix のバージョンを選択します。Zabbix Cloud は最新の LTS バージョンをサポートし、マイナーバージョンの更新を自動的に適用します。
IP address or CIDR	ノードの初期アクセスフィルターを指定します。IP アドレスまたは CIDR を手動で入力するか、該当するオプションを選択してフィールドに自動入力します。サーバーノードの場合、このフィルターはサーバー、API、および Web インターフェースへの接続に適用されます。フィルターは後からAccess filters タブで変更できます。
Optional access filter group description	複数の IP アドレスまたは CIDR を含めることができるアクセスフィルターグループに、任意の説明を追加します。

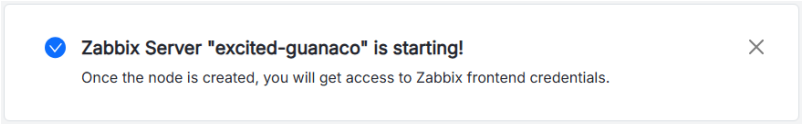
Warning:
ノードの作成中、*IP address or CIDR* フィールドには、Zabbix Cloud へのアクセスに使用した IP アドレスがあらかじめ入力されています。この値を変更して、別の初期アクセスフィルターを指定できます。ノードの作成後は、*Access filters* タブからアクセスフィルターを追加、編集、または削除できます。

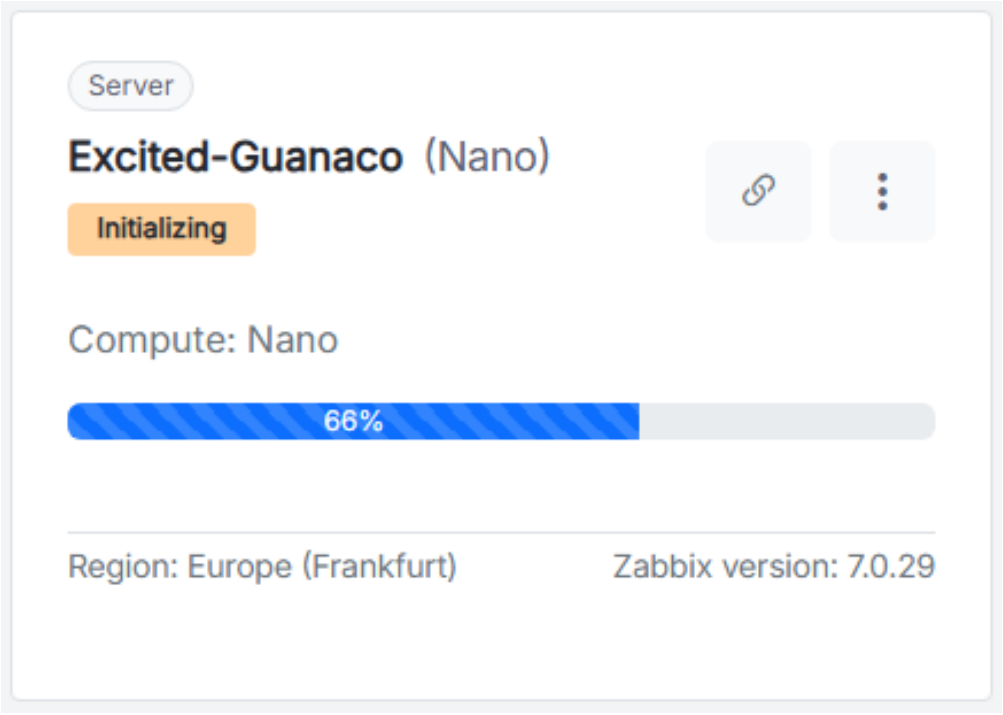
3 番目のステップには、ノードの設定概要が表示されます。



設定を確認し、すべて正しいことを確認したら、トライアルノードの場合は *Complete* を、有料ノードの場合は *Proceed to payment* をクリックします。それ以外の場合は、*Back* ボタンをクリックして前のステップに戻るか、*Cancel* をクリックして最初からやり直します。

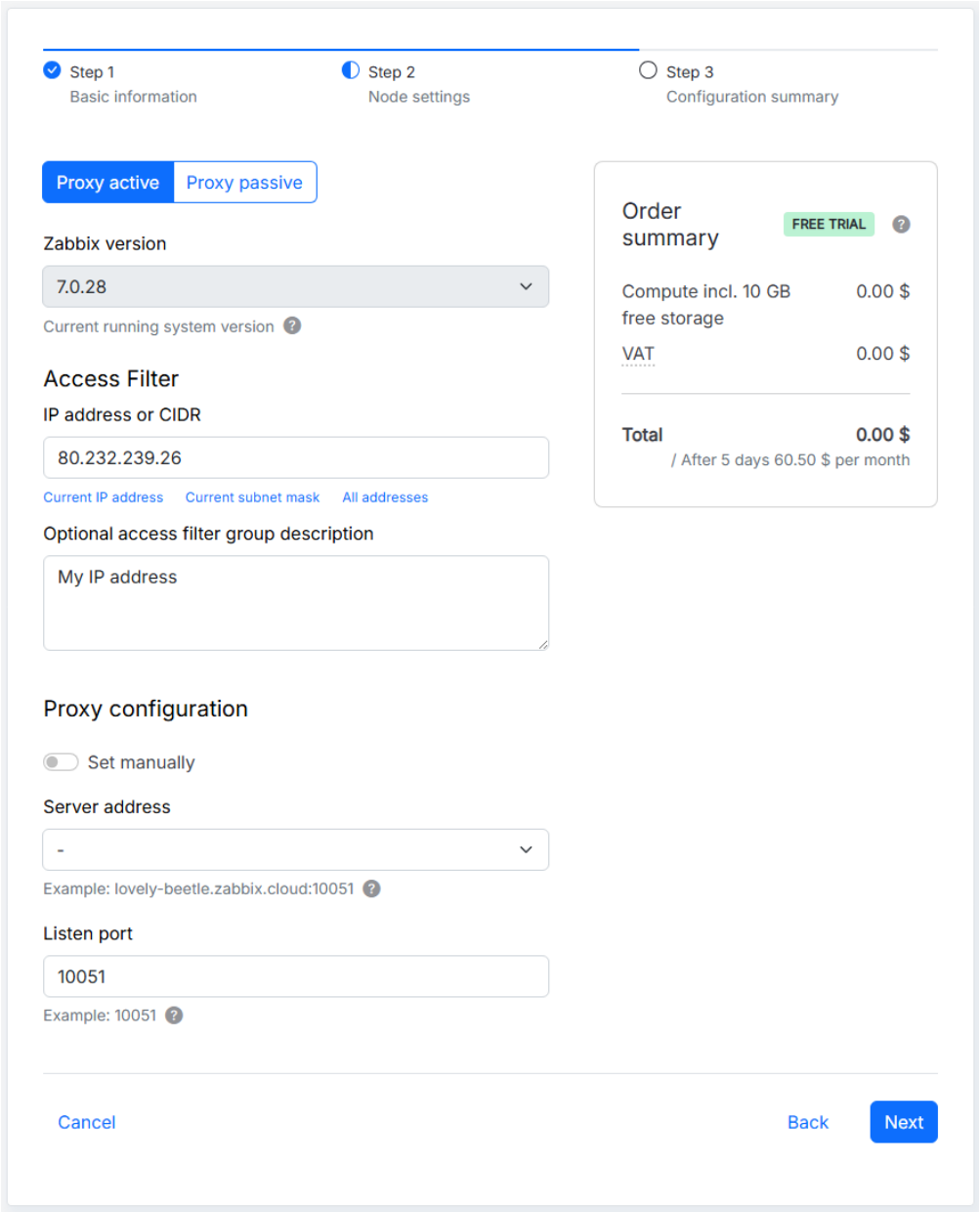
ノードの初期化中は、対応するメッセージが表示され、ノードはすでに *Organizations* の下に表示されます。





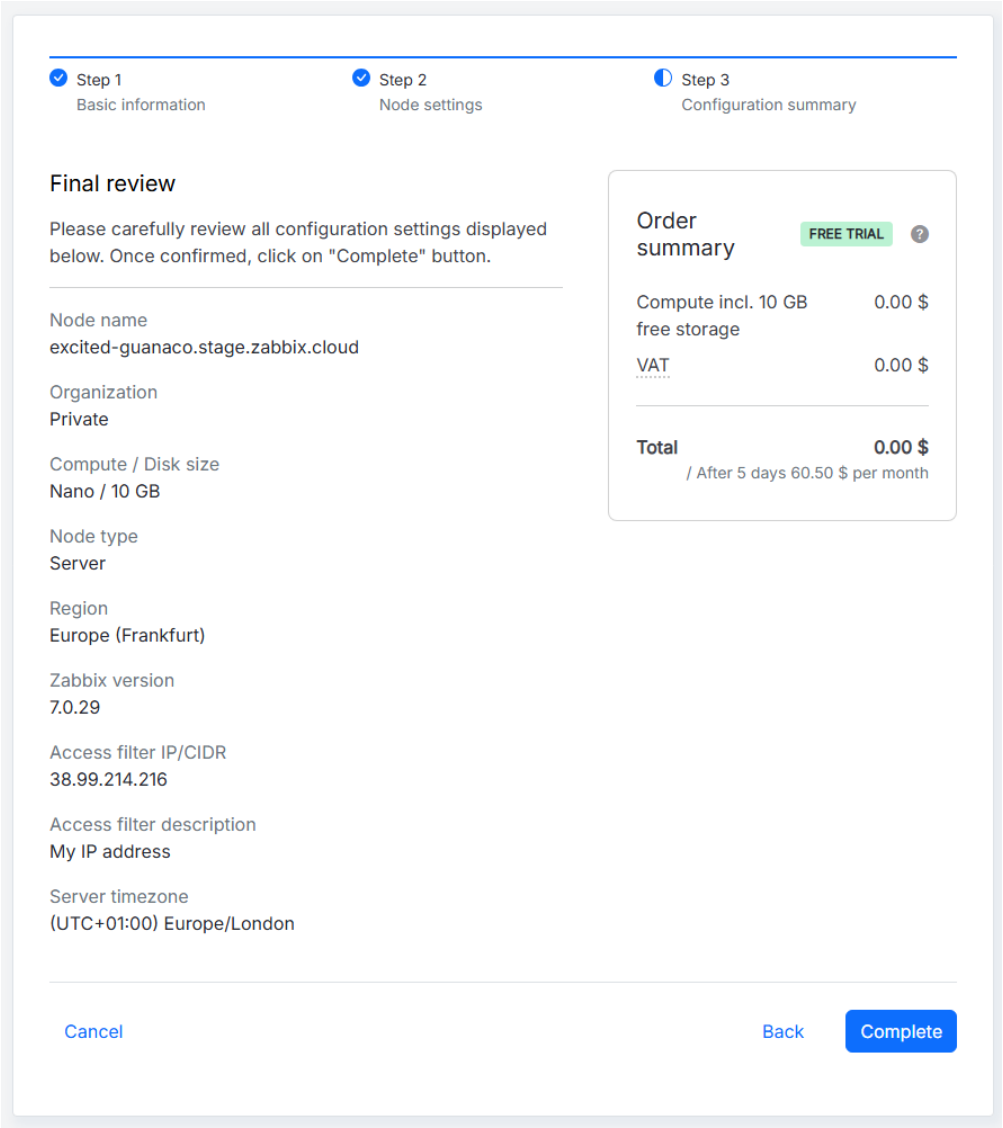
プロキシノードの設定

プロキシノードの作成時には、次のパラメータを使用できます。



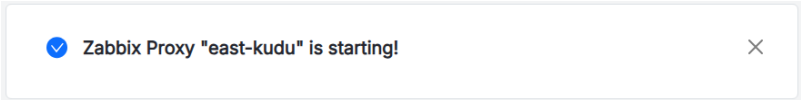
Parameter	Description
Proxy active/Proxy passive	プロキシモードを選択します。アクティブモードでは、プロキシが Zabbix サーバーへの接続を開始します。パッシブモードでは、Zabbix サーバーがプロキシへの接続を開始します。
Zabbix version	Zabbix のバージョンを選択します。Zabbix Cloud は最新の LTS バージョンをサポートし、マイナーバージョンの更新を自動的に適用します。
IP address or CIDR	プロキシの初期インバウンド IP アクセスフィルターを指定します。IP アドレスまたは CIDR を手動で入力するか、該当するオプションを選択してフィールドに自動入力します。このフィルターは、プロキシへの到達を許可するネットワーク要求を定義し、後からAccess filters タブで変更できます。
Optional access filter group description	複数の IP アドレスまたは CIDR を含めることができるアクセスフィルターグループに、任意の説明を追加します。
Set manually	Zabbix サーバーの指定方法を選択します。無効にすると、利用可能なサーバーノードの一覧からサーバーノードを選択します。有効にすると、サーバーアドレスを手動で指定します。
Server address	Zabbix サーバーのアドレスを指定します。*Set manually* が無効の場合は、一覧からサーバーノードを選択します。有効の場合は、最大 10 個のサーバーアドレスを指定できます。アクティブプロキシの場合は、Zabbix サーバーの IP アドレスまたは DNS 名を指定します。パッシブプロキシの場合は、Zabbix サーバーの IP アドレス、DNS 名、または CIDR を指定します。アクセスフィルターは個別に設定され、サーバーアドレスを指定しても自動的には更新されません。
Listen port	プロキシが受信接続を待ち受けるポートを指定します。デフォルトのポートは 10051 です。リッスンポートはノードの作成後に変更できません。

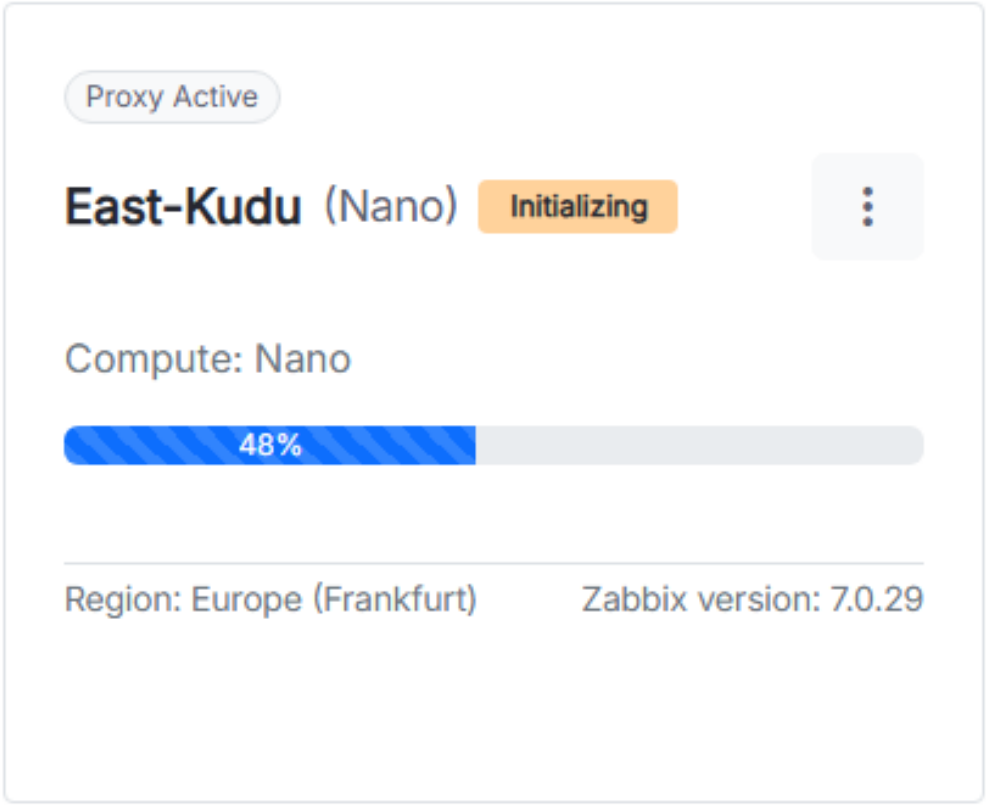
3 番目のステップでは、ノードの設定概要が表示されます。



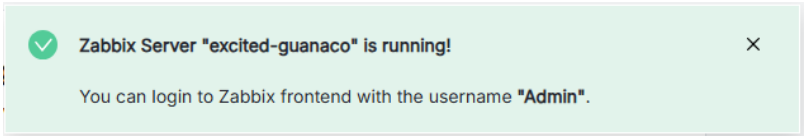
設定を確認し、すべて正しいことを確認したら、トライアルノードの場合は *Complete* を、有料ノードの場合は *Proceed to payment* をクリックします。それ以外の場合は、*Back* ボタンをクリックして前のステップに戻るか、*Cancel* をクリックして最初からやり直します。

ノードの初期化中は、対応するメッセージが表示され、ノードはすでに *Organizations* の下に表示されます。





Zabbix Cloud プロキシを設定するには、Zabbix Cloud プロキシのセットアップを参照してください。

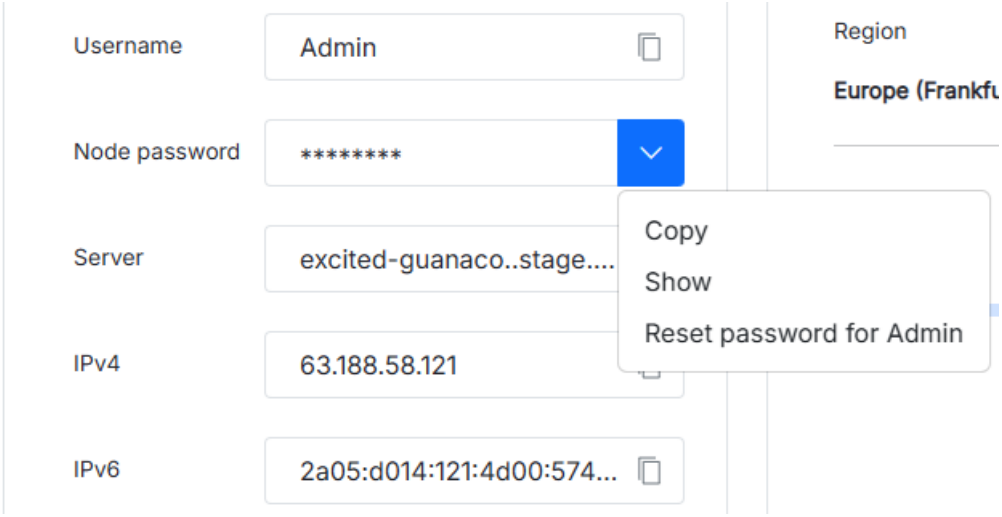


Zabbix の使用を開始する

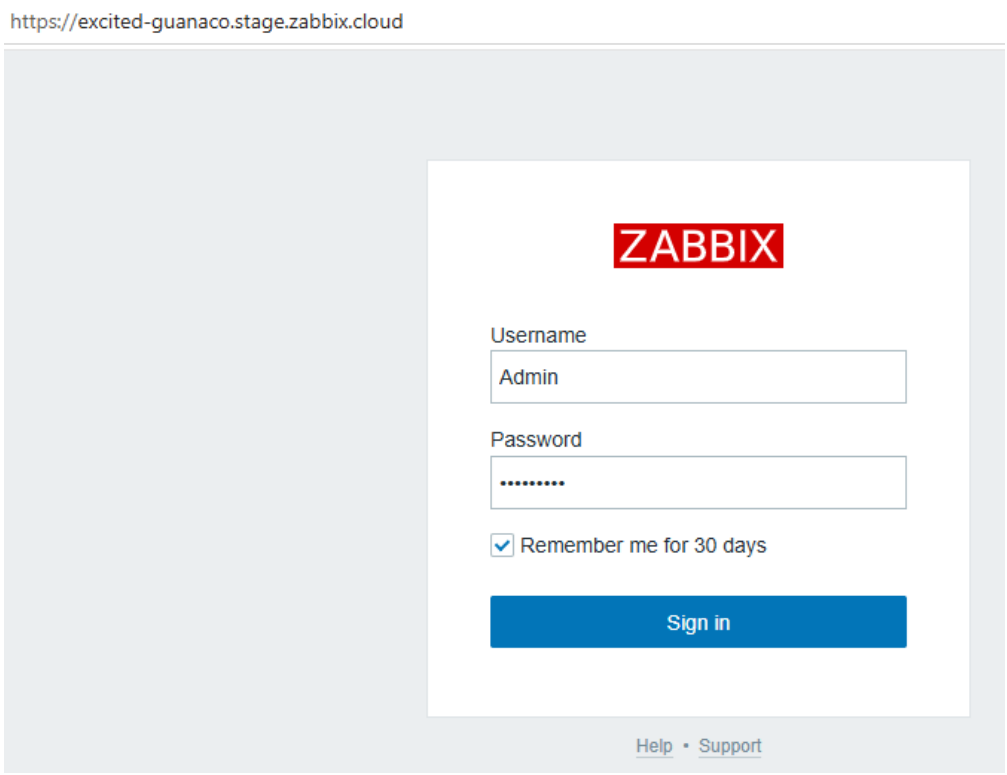
サーバーノードのステータスが *Initializing* から *Running* に変わると、Zabbix の Web インターフェースにログインできます。

初期 Web インターフェースパスワードにアクセスするには、作成したノードをクリックします。

ノードの設定（概要、アクセスフィルター、暗号化、バックアップなど）が表示されます。**Connection info** で *Node password* フィールドを見つけ、ドロップダウンから *Reset password for Admin* を選択します。パスワードがリセットされたら、同じドロップダウンをクリックして *Copy* を選択します。



Web インターフェースの URL をクリックして、Zabbix の Web インターフェース（`excited-guanaco.zabbix.cloud`）を開きます。



次の認証情報を使用してログインします。

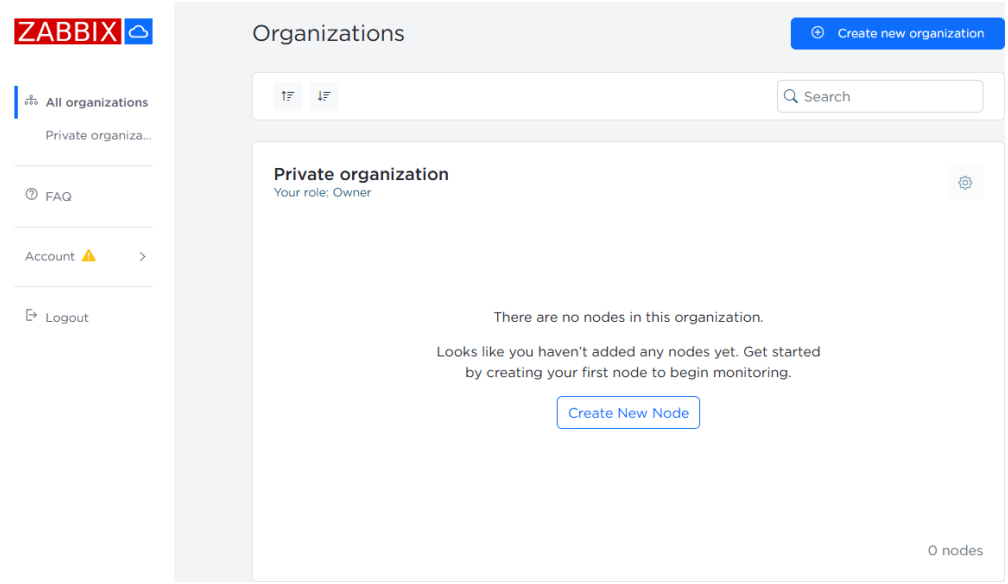
- *Username* - `Admin`
- *Password* - コピーしたパスワードを入力

監視を開始するには、Zabbix Cloud サーバーへの接続を許可する IP アドレスをアクセスフィルターで定義する必要があります。少なくとも 1 つ定義する必要があります。

組織管理　組織は、1 つ以上のノードを格納するコンテナです。組織を使用すると、次のようなリソースの構成や責任の委任も行えます。

- ノード管理
- 請求

Zabbix Cloud にログインすると、すべての組織ページが表示されます。デフォルトでは、アカウント用のプライベート組織が作成されます。

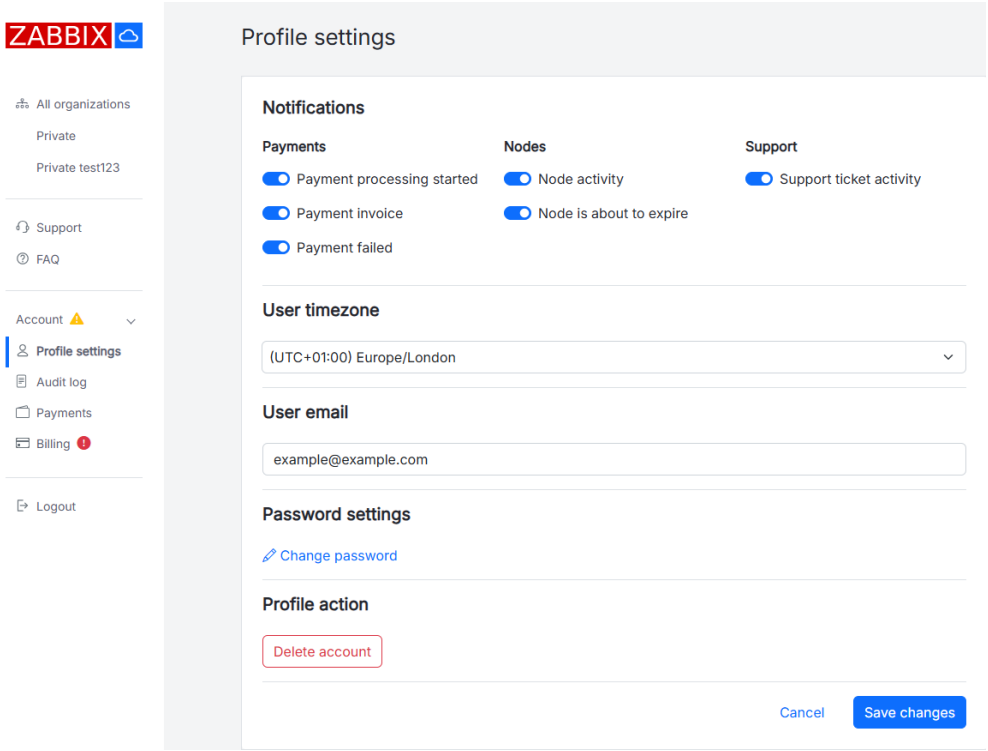


この組織の所有者として、ユーザー、管理者、ノード、組織名を管理できます。 設定ボタンをクリックすると、管理オプションにアクセスできます。

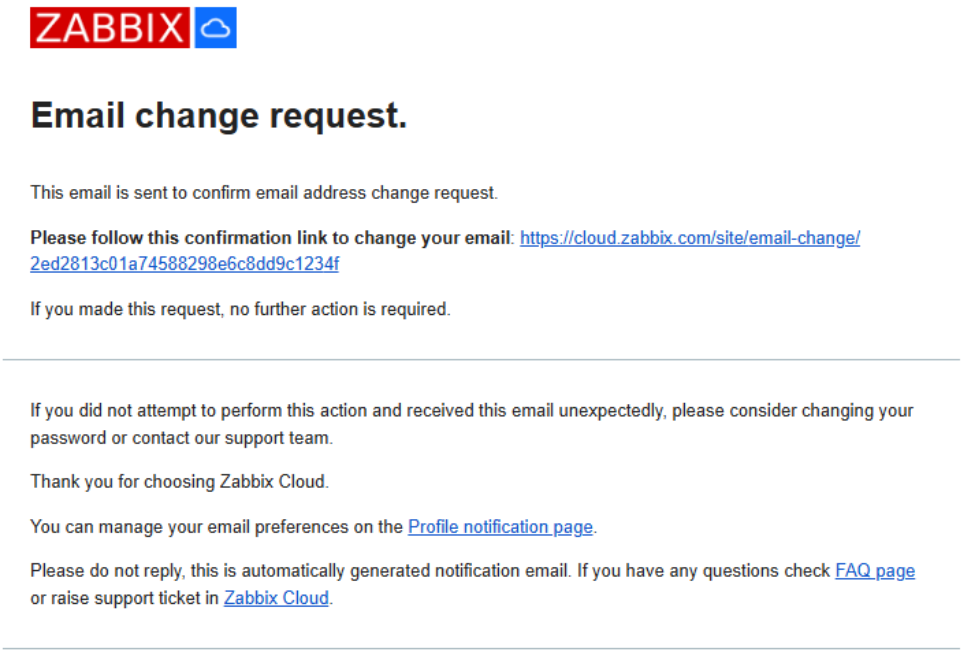
アカウント管理　アカウントのメールアドレスの変更

アカウントのメールアドレスを変更するには、*Account* > *Profile settings* に移動します。

User email フィールドに新しいメールアドレスを入力し、*Save changes* をクリックします。



新しいメールアドレスに確認メッセージが送信されます。メッセージ内のリンクに従って、メールアドレスの変更を確認します。



確認リンクを開くと、確認メッセージと更新されたメールアドレスが User email フィールドに表示された *Profile settings* ページが開きます。

変更が確認されると、変更を確認するメールが新旧両方のメールアドレスに送信されます。以前のメールアドレスに送信されるメッセージには、以前のメールアドレスと新しいメールアドレスが記載されます。

ノード設定

概要

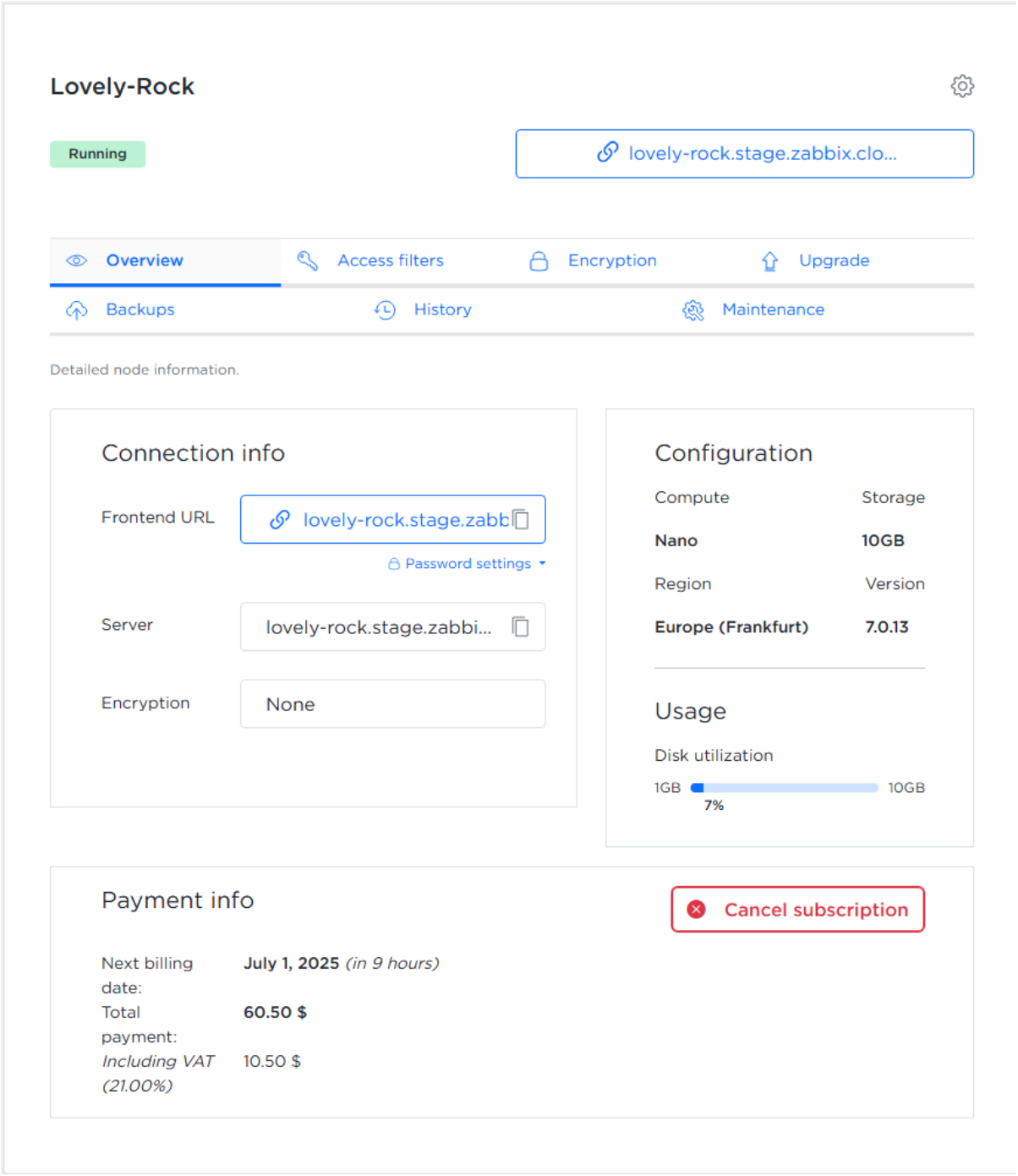
クラウドノードは、クラウドにデプロイされた Zabbix コンポーネントです。サーバーノードには Zabbix サーバー、データベース、Web インターフェースが含まれ、プロキシノードには Zabbix プロキシとデータベースが含まれます。クラウドノードには、専用のコンピューティングリソース（CPU、メモリ）とディスクストレージがあります。

ユーザーは、1 つ以上のクラウドノードを作成できます。クラウドノードは組織（構造、地域など）にグループ化でき、分散監視環境を構築できます。

クラウドノードと組織は、Zabbix Cloudのユーザーインターフェースを使用して作成します。ノードの設定ページにアクセスするには、すべての組織、組織メニュー、またはノードの起動後に表示されるノード設定ページリンクからノードをクリックします。

ノード設定ビューでは、各タブでノードの特定の項目を管理できます。

- 概要 - ノードの詳細を表示し、Web インターフェースを開き、ノードユーザーを管理します。
- アクセスフィルター - IP アドレスまたはサブネットによるアクセスを制御します。
- 暗号化 - 証明書、キー、オプションの CRL をアップロードして TLS を設定します。
- バックアップ - ノードのスナップショットを作成、リストア、削除し、次回のスケジュール済みバックアップを表示します。
- 履歴 - 監査ログ、履歴、トレンドの保存期間を、あらかじめ設定された単位と 1 日 3 回までの更新制限で設定します。
- メンテナンス - タイムゾーン、曜日、時間を指定して、毎週のメンテナンス時間帯を定義します。
- アップグレード - コンピューティングリソースやストレージリソースを拡張し、トライアルまたは支払いのステータスを管理します。

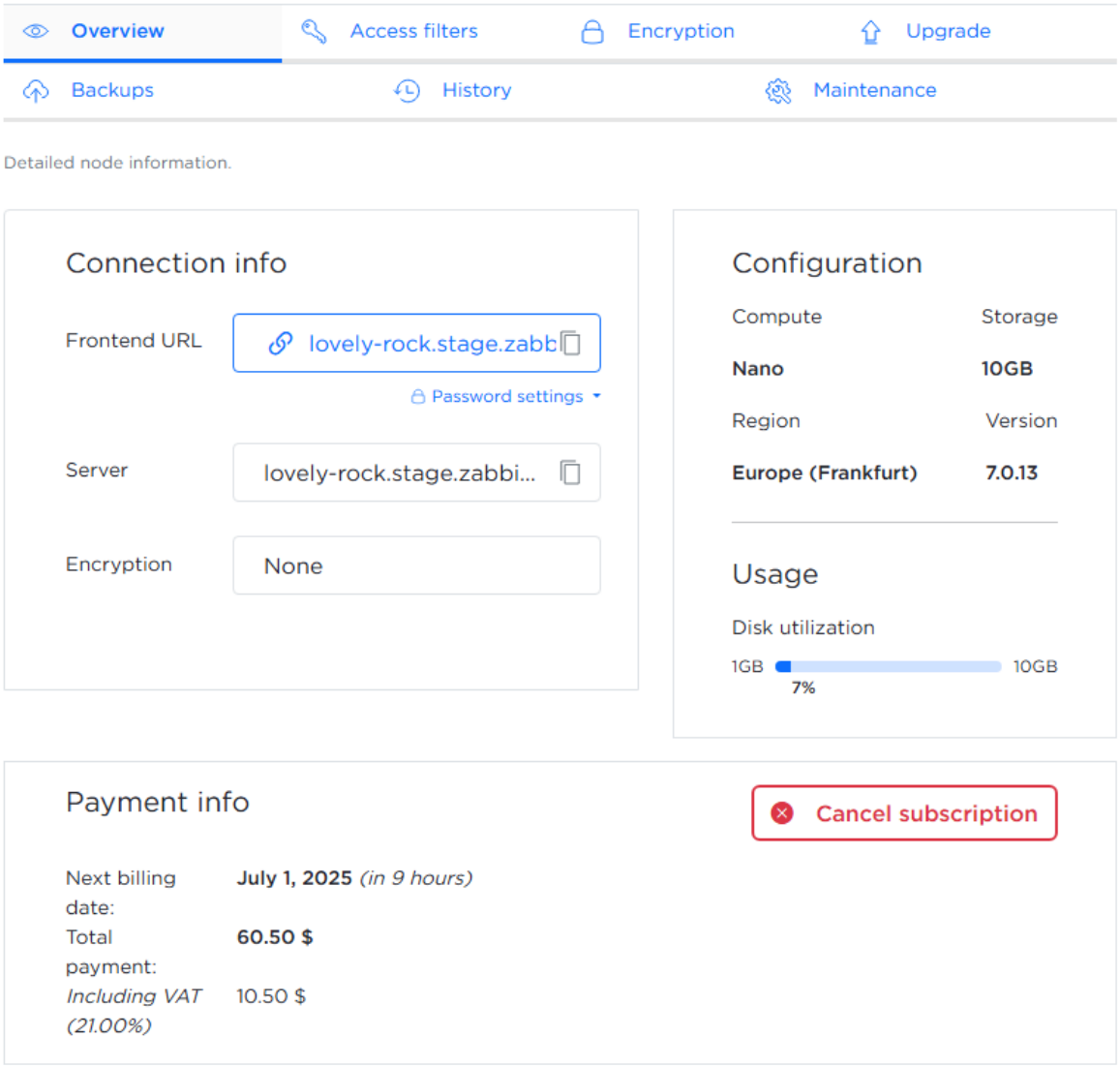


Note:
ノードの種類（サーバーまたはプロキシ）、ページを表示するユーザーの種類、およびノードのステータス（トライアルまたは非トライアル）によっては、一部のタブを利用できない場合があります。

概要タブ

概要タブには、ノードに関する詳細情報が表示されます。表示される情報は、ノードがサーバーかプロキシかによって異なります。

サーバーノード　サーバーノードの概要タブには、以下のセクションがあります。



接続情報:

- クリップボードにコピーするオプションがあるクリック可能な Web インターフェース URL
- ユーザー名
- ノードのパスワードとその設定
- クリップボードにコピーするオプションがあるサーバーのホスト名
- クリップボードにコピーするオプションがある IPv4 アドレス
- クリップボードにコピーするオプションがある IPv6 アドレス
- 現在の暗号化方式

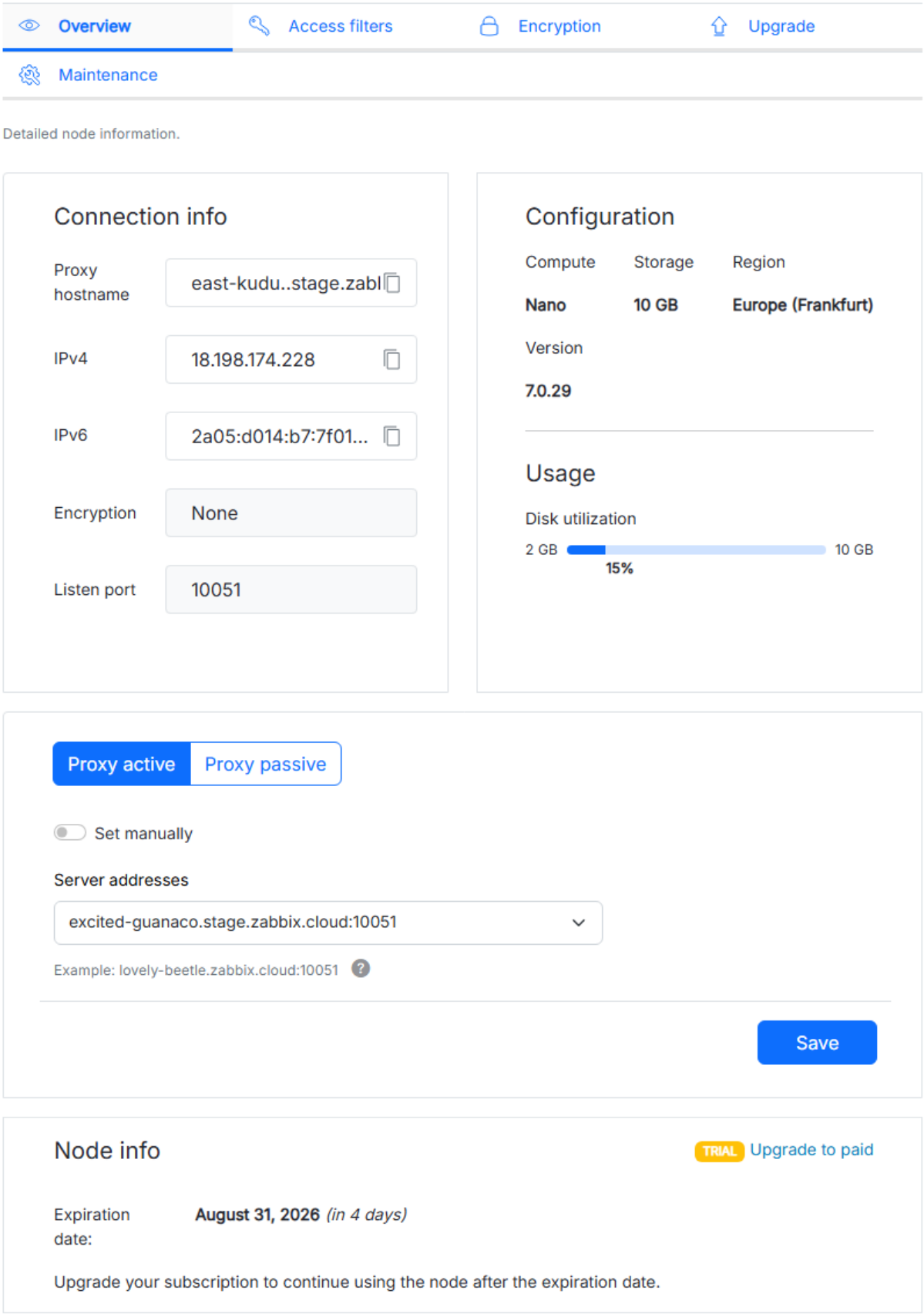
設定と使用状況:

- コンピューティア、最大ストレージ容量、リージョン
- Zabbix コンポーネントのバージョン
- ディスク使用率

ノード情報/支払い情報:

- 有効期限（トライアルノードおよび前払いノードの場合）
- ノードの初期化が完了すると、ノードの種類に応じたコンテキスト固有のボタンが表示されます。- 有料プランにアップグレード - 無料トライアル中のノードで使用できます - サブスクリプションをキャンセル - 有効なサブスクリプションがある有料ノードで使用できます - 有料プランに再登録 - サブスクリプションがキャンセルされた場合に表示されます - 有効期間が長く、キャンセルできない前払いノードには、アクションボタンは表示されません

プロキシノード　プロキシノードの概要タブには、次のセクションがあります。



接続情報:

- クリップボードにコピーするオプション付きのプロキシホスト名
- クリップボードにコピーするオプション付きの IPv4 アドレス
- クリップボードにコピーするオプション付きの IPv6 アドレス
- 現在の暗号化方式
- 現在のリッスンポート

設定と使用状況:

- コンピューティア、最大ストレージ、リージョン
- Zabbix コンポーネントのバージョン
- ディスク使用率

プロキシモード:

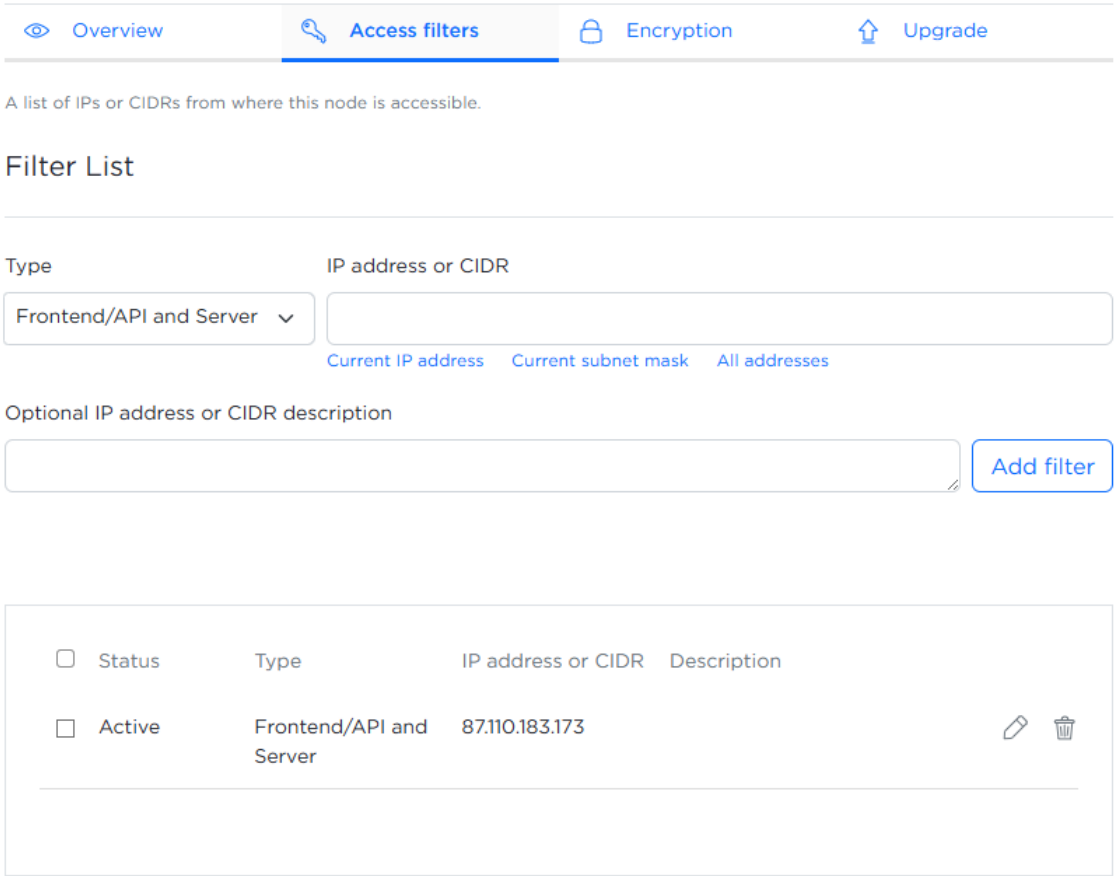
- アクティブプロキシモードとパッシブプロキシモードを切り替えます
- サーバーアドレスドロップダウンからサーバーアドレスを選択するか、手動で設定を有効にしてサーバーアドレスを手動で入力します

ノード情報/支払い情報:

- 有効期限（トライアルノードおよび前払いノードの場合）
- ノードの初期化が完了すると、ノードの種類に応じてコンテキスト固有のボタンが表示されます。- 有料版にアップグレード - 無料トライアル中のノードで使用できます - サブスクリプションをキャンセル - 有効なサブスクリプションがある有料ノードで使用できます - 有料版に再登録 - サブスクリプションがキャンセルされている場合に表示されます - 有効期間が長く、キャンセルできない前払いノードには、アクションボタンは表示されません

アクセスフィルタータブ

アクセスフィルタータブでは、IP アドレスまたはサブネットによって Zabbix ノードへのアクセスを制限できます。



アクセスフィルターを設定するには、次のフィールドに入力します。

- タイプ - サーバーノードの場合、ドロップダウンから制限する Zabbix コンポーネントを選択します。Web インターフェース/API のみ、サーバーのみ、または両方を選択できます。プロキシノードでは、受信 *IP* のみ使用できます。プロキシへの到達を許可する受信ネットワーク要求を定義します。
- *IP* アドレスまたは *CIDR* - IP アドレスまたは CIDR を手動で入力するか、該当するオプションを選択してフィールドに自動入力します。
- 任意の *IP* アドレスまたは *CIDR* の説明 - 各 IP またはサブネットに任意の説明を追加します。

フィールドへの入力が完了したら、フィルターを追加をクリックします。追加したフィルターは下の表に表示され、ステータス、タイプ、IP アドレスまたはサブネット、説明が示されます。

Warning:
ノードの初期化中に、使用した IP アドレスに対するアクセスフィルターが自動的に作成されます。後で別の IP アドレスまたはネットワークから接続する場合は、そのアドレスをここに追加する必要があります。追加しないとアクセスがブロックされます。

次の操作を実行できます。

-  - フィルターを編集します。
-  - チェックボックスを使用して、1 つまたは複数のフィルターを削除します。

暗号化タブ

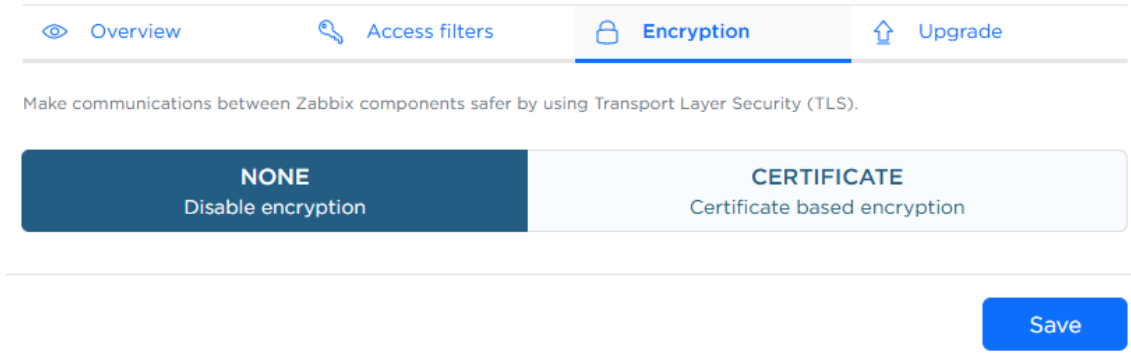
暗号化タブでは、必要な証明書ファイルをアップロードして、Zabbix コンポーネント間の通信に対する TLS 暗号化を設定できます。

証明書を使用した安全な通信を有効にするには、*CERTIFICATE* を選択し、コンピューターから次のファイルをアップロードします。

- ルート *CA* 証明書 - ピア証明書の検証に使用する、信頼された CA 証明書。ファイルを選択をクリックし、信頼されたルート CA 証明書を含むファイル（通常は `.crt` または `.pem` ファイル、PEM エンコード）を選択します。
- 証明書チェーン - 独自の証明書に対する信頼チェーンを表します。ファイルを選択をクリックし、証明書および中間 CA 証明書を含む PEM エンコードされたファイル（例：`.crt` または `.pem`）を選択します。

- 秘密鍵 - 証明書に対応する秘密鍵。ファイルを選択をクリックし、PEM エンコードされた `.key` ファイルを選択します。
- 失効した証明書（オプション）- 証明書失効リスト（CRL）を管理している場合は、ファイルを選択をクリックし、信頼しない証明書を指定する CRL ファイルを選択します。

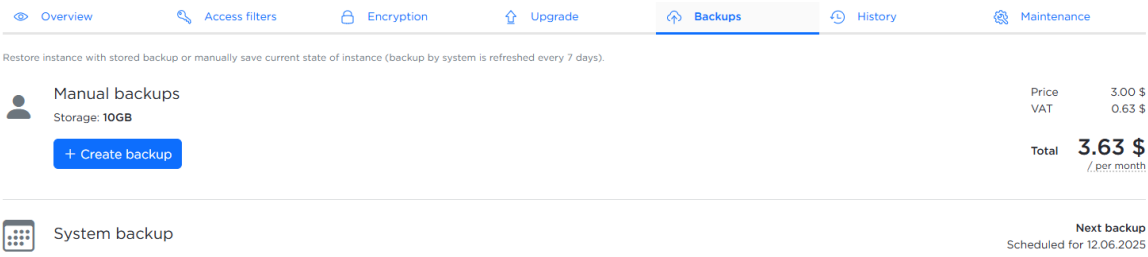
ファイルを選択したら、保存をクリックします。ファイルがアップロード、検証され、Zabbix コンポーネント間の TLS 暗号化を設定するために適用されます。



バックアップタブ

バックアップタブでは、Zabbix ノードのバックアップを表示、作成、リストア、削除できます。

保存されている任意のバックアップからインスタンスをリストアしたり、いつでも新しい手動バックアップを作成したりできます。このタブには、次回のスケジュール済みバックアップの日付も表示されます。



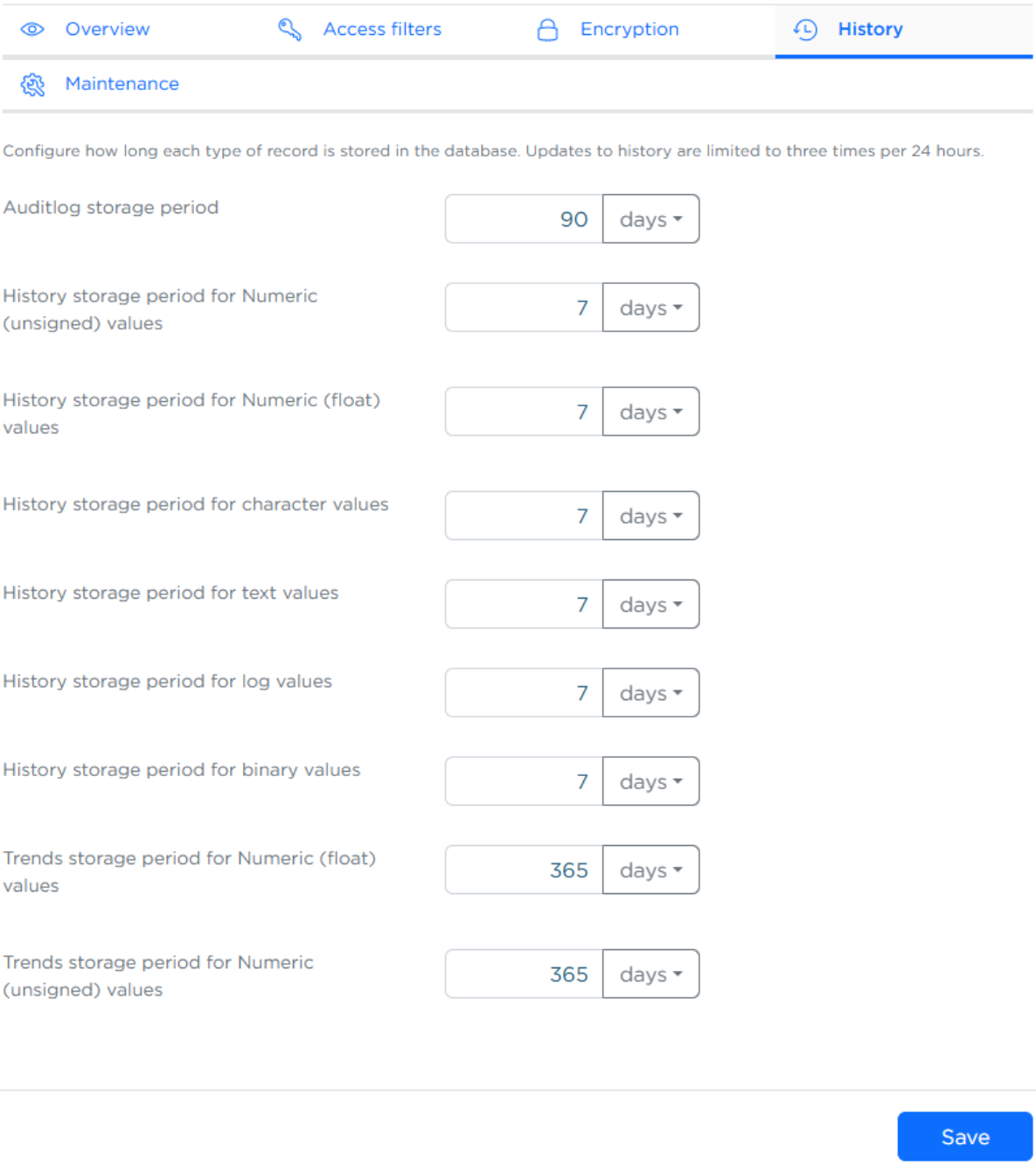
バックアップを作成するには、次の手順に従います。

1. バックアップの作成ボタンをクリックします。
2. ポップアップウィンドウで、必要に応じて説明を入力します。
3. 作成をクリックして確定します。

Warning:
追加のバックアップには、ストレージ使用量に基づいて月額料金が発生します。

履歴タブ

履歴タブでは、さまざまな種類の履歴データを保存する期間を設定できます。



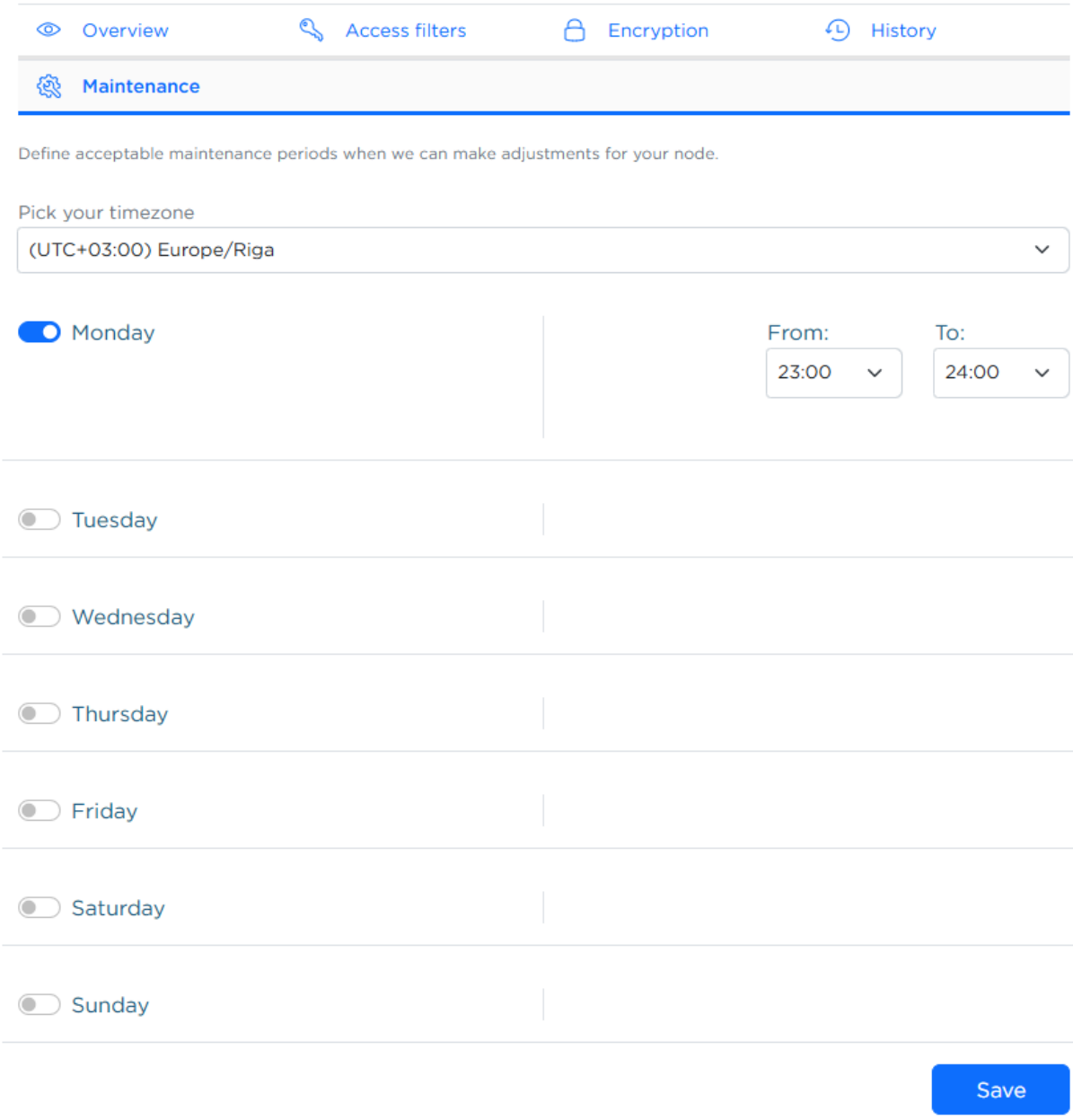
各データタイプ（数値、テキスト、ログなど）の保存期間を設定し、単位（日、月、年）を選択できます。変更は 24 時間ごとに最大 3 回まで行えます。

値を更新したら、保存をクリックして変更を適用します。

Warning:
保存期間を長くすると、ディスク使用量が増加し、パフォーマンスに影響する可能性があります。実際のデータ量と分析のニーズに基づいて保存期間を設定することを推奨します。
 Zabbix Cloud では、従来の Zabbix ハウスキーピングの代わりに、データベースパーティショニングを使用してデータストレージを管理します。これにより、より高速なパフォーマンスと、より予測しやすい保存期間の処理が可能になります。

メンテナンスタブ

メンテナンスタブでは、ノードの更新や調整を実行できるメンテナンス時間帯を定義できます。

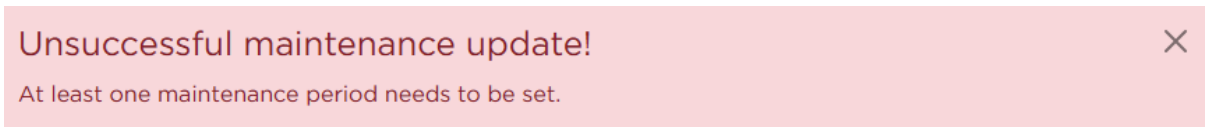


以下を実行する必要があります。

- ドロップダウンからタイムゾーンを選択します。
- メンテナンスの曜日を少なくとも 1 つ、時間帯を少なくとも 1 つ選択します。

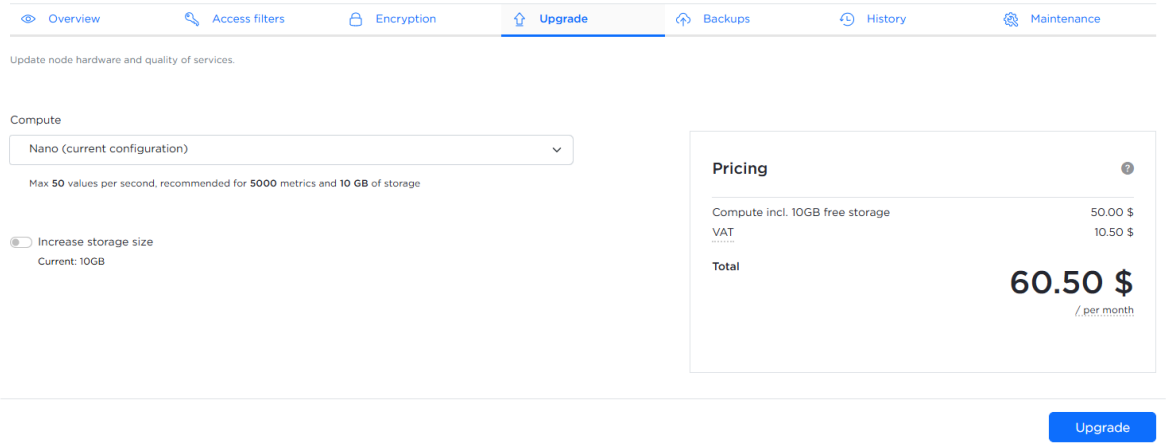
保存をクリックして変更を適用します。

曜日または時間帯が選択されていない場合は、エラーが表示されます。



アップグレードタブ

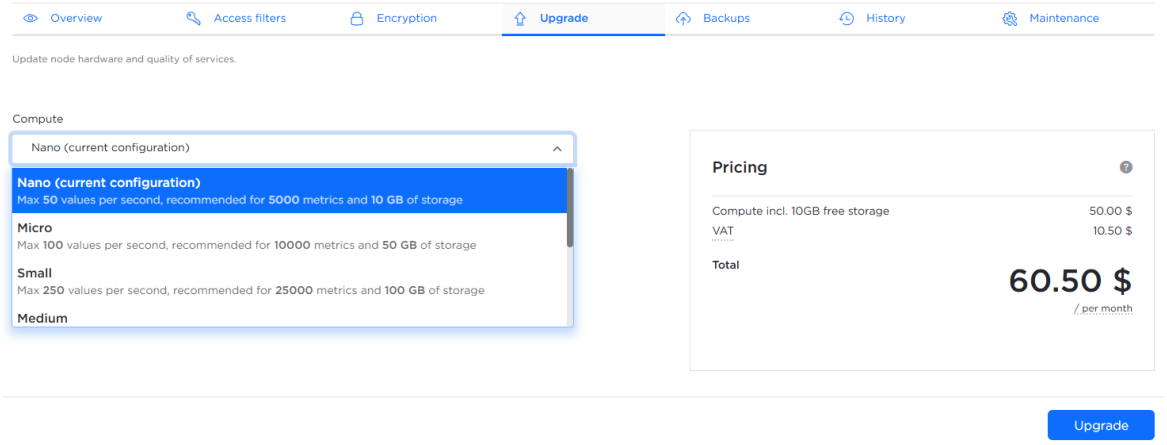
アップグレードタブでは、ノードのリソースを拡張したり、サブスクリプションを管理したりできます。



コンピューティアまたはディスクサイズを増やすことができます。ディスクサイズを増やせるのは 24 時間に 1 回のみで、UI から縮小することはできません（縮小を希望する場合は、Zabbix サポートにお問い合わせください）。無料トライアルのノードは、有料サブスクリプションに変更できます。

コンピューティアを増やすには:

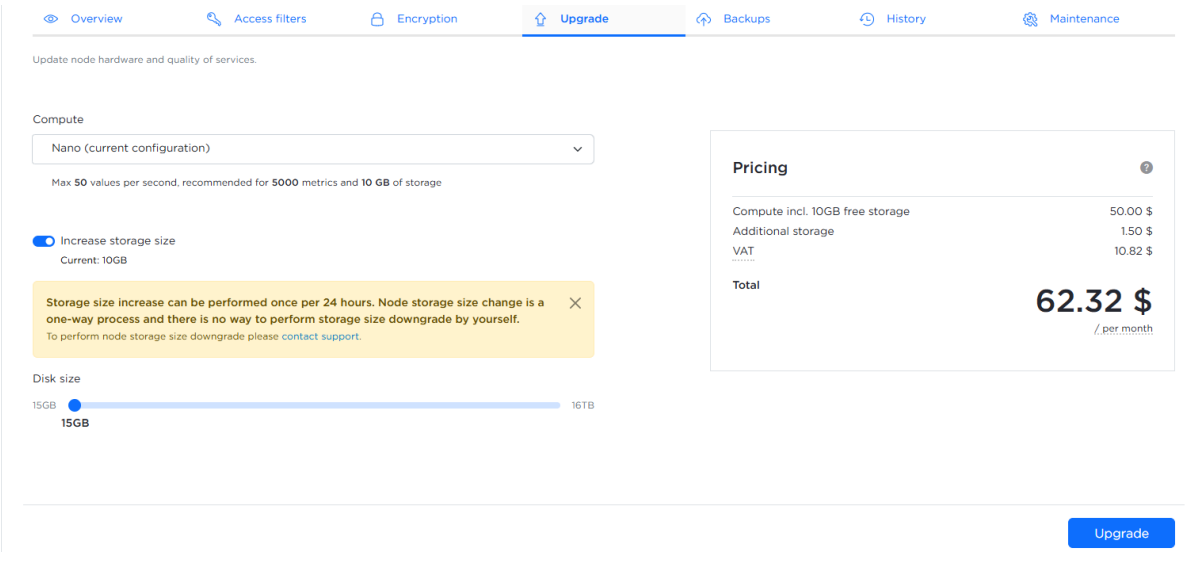
1. コンピュートドロップダウンをクリックします。



2. 希望する構成ティアを選択します。
3. アップグレードボタンをクリックします。

ディスクサイズを増やすには:

1. ストレージサイズを増やすラジオボタンをクリックします。
2. ディスクサイズスライダーを希望する値に調整します。



3. アップグレードをクリックします。

Note:
変更を確定する前に、更新後の合計月額料金が料金テーブルに表示されます。

ユーザーの追加

はじめに

以下の例では、組織やノードにユーザーを追加する方法について説明します。

Note:
Web インターフェースのパスワードのコピー、クリア、リセットは、所有者のみが実行できます。

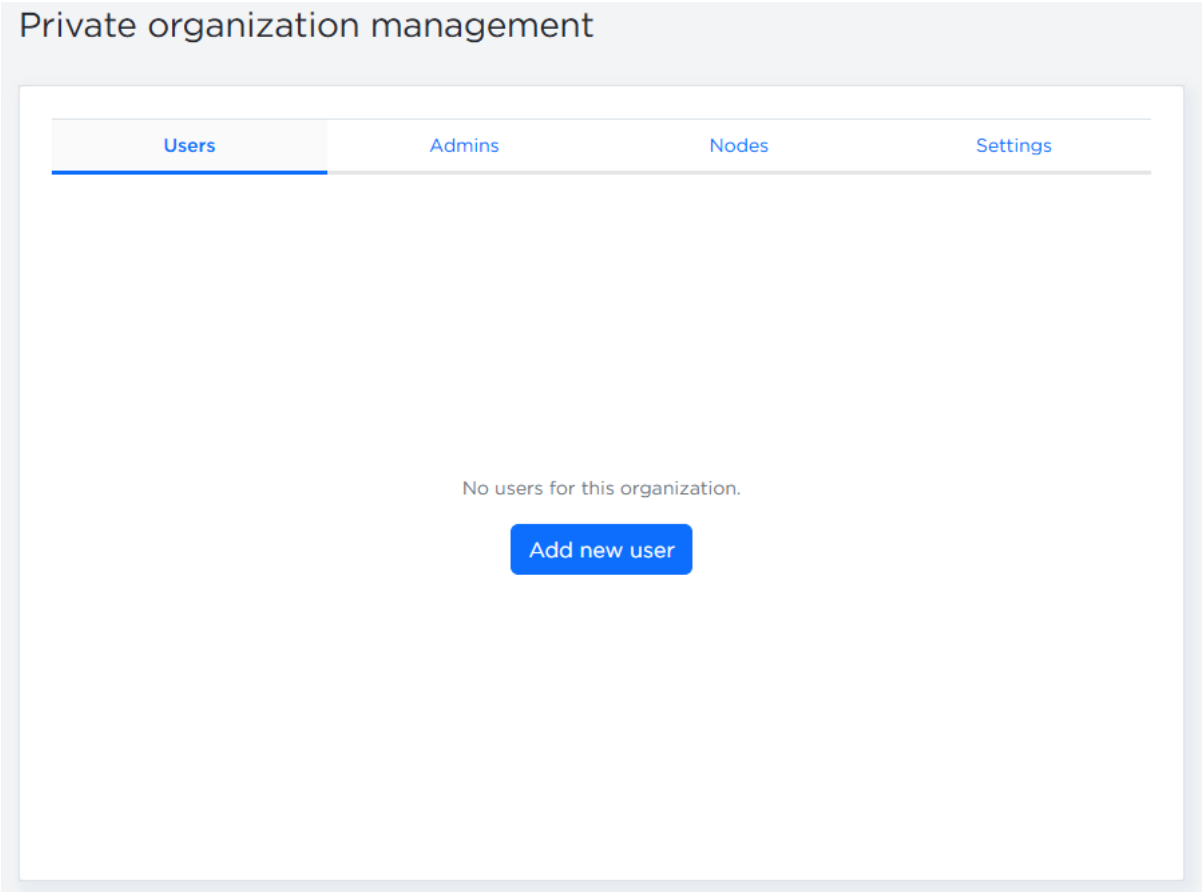
組織へのユーザー追加

Zabbix のユーザーと同様に、Zabbix Cloud でもユーザーを作成し、組織やそのノード内で何ができるかを制御するためのロールを割り当てることができます。

組織にユーザーを追加するには:

1. すべての組織ページを開きます。

2. 特定の組織の  設定ボタンをクリックして、管理メニューに入ります。



3. ユーザーまたは管理者タブで、新しいユーザーを追加をクリックします。

4. ユーザーのメールアドレスを入力し、管理者またはユーザーのロールを選択します。

ユーザーロールを選択すると、そのユーザーは組織に追加されますが、いずれのノードへのアクセス権も付与されませんのでご注意ください。

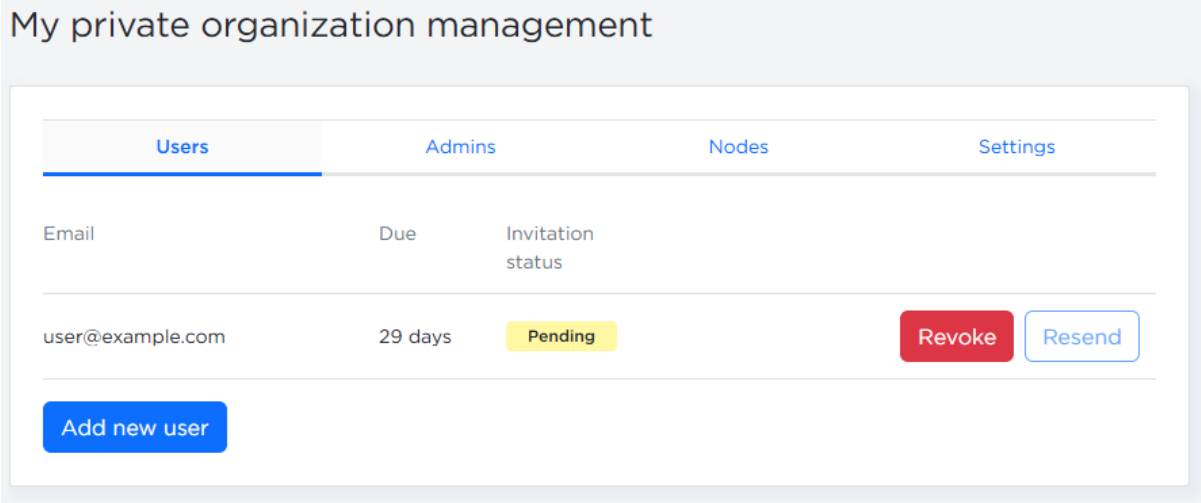
各ロールは異なるレベルのアクセス権を付与します:

- ユーザー - 割り当てられたノードにアクセスでき、課金が発生しない設定作業を実行できます。例:

    - アクセスフィルターの管理
    - 暗号化の設定
    - 履歴/トレンドの保持期間の設定
    - メンテナンスのスケジューリング
- 管理者 - ユーザーのすべての権限に加え、課金が発生する可能性のある操作も実行できます。例:
    - バックアップの作成と復元
    - ノードリソース（コンピュート階層やディスクサイズ）のアップグレードまたはダウングレード
    - サブスクリプションのキャンセル
- オーナー - 組織を完全に管理できます。管理者のすべての権限に加え、オーナーは以下も可能です:
    - ノードの作成と削除
    - ユーザーや管理者の追加・削除
    - 組織名の変更や削除
    - 請求情報の管理

> **Attention:**
> オーナーの支払い方法で、すべてのサービスの料金が請求されます。銀行の確認が必要な場合は、管理者が操作を開始した場合でも、オーナーが取引を承認する必要があります。

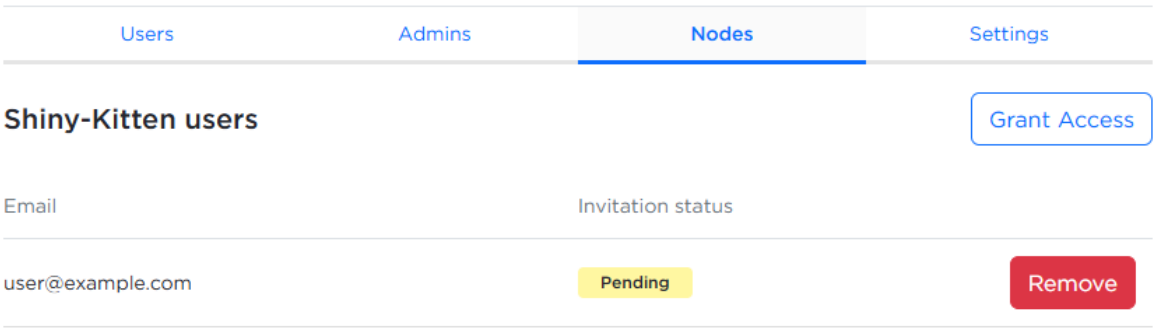
招待されたユーザーは保留中のステータスで追加されます。
招待を取り消したり、再送信したりできます（再送信は 10 分に 1 回のみ可能で、この制限に達するとボタンが無効になります）。



ノードへのユーザー追加

ノードにユーザーを追加するには:

1. ノードタブに移動します。
2. アクセス権を付与をクリックします。
3. ドロップダウンから組織に追加したユーザーを選択します。
4. アクセス権を付与をクリックします。



または、ノード設定ウィンドウからアクセス権を付与することもできます:

1.  設定ボタンをクリックします。

2. ユーザーを追加を選択します。
3. ユーザーを選択し、アクセス権を付与をクリックします。

招待されたユーザーは、招待が送信されたのと同じメールアドレスを使用して Zabbix Cloud アカウントを作成し、招待を受け入れる必要があります。すでにそのメールアドレスで Zabbix Cloud アカウントを持っている場合は、サインインして招待を確認する必要があります。招待は特定のメールアドレスに紐付けられており、別のアカウントから受け入れることはできません。

Zabbix Cloud の主な違い

はじめに

Zabbix Cloud は、ユーザーがクラウド上で独立した Zabbix インスタンス（クラウドノード）を実行できるクラウドベースのプラットフォームです。

通常のオンプレミス環境での Zabbix 導入と比較すると、次のような違いがあります。

- インストールやメンテナンスは不要です。
- クラウドノードは Zabbix によって完全に管理されます。
- クラウドノードは数分で起動できます。

このサービスは従量課金制で提供されています（無料トライアルも利用できます）。

比較表　オンプレミスの Zabbix インストールとクラウドノードでは、多くの機能が共通していますが、相違点もあります。以下に概要比較表を示します。以降のセクションでは、選択した項目について詳しく説明します。

機能 / 項目	Zabbix Cloud	Zabbix オンプレミス
リリースサイクル	LTS リリースに合わせて更新	Zabbix のバージョン（安定版、ベータ版、カスタムビルド）を選択
使いやすさ	すぐに使用できるプラットフォーム。Zabbix が完全に管理	インストール、設定、メンテナンスが必要
脆弱性へのパッチ適用	OS/Zabbix へのパッチ適用を自動化	ユーザーが手動でパッチを適用
HTTPS 証明書	すぐに使用できる有効な証明書（Let's Encrypt は使用しない）	デフォルトでは自己署名（Let's Encrypt またはその他の CA を設定可能）
DB パフォーマンス (INSERT/UPDATE/SELECT)	最大速度を実現する自動パーティショニング。長時間実行される DELETE なし	手動でのパーティショニングとクリーンアップ (長時間の DELETE トランザクションが発生する可能性あり）
パフォーマンスチューニング	Zabbix Cloud チームによる事前チューニング (変更履歴なし）	チューニングパラメータを完全に可視化し、制御可能
ファイアウォール制御	Zabbix Cloud チームが管理。GUI/トラッパーポートはパブリックに公開されない	ファイアウォールルールとポート公開を完全に制御可能
ポート *25* の送信トラフィック	スパムや不正利用を防ぐため、デフォルトでブロック。代わりに SMTP ポート 587 または 465 を使用	Zabbix による制限なし。ネットワークおよびホスティングプロバイダーに依存
DB の分離と保護	高度に分離され、DB/SSH への直接アクセス不可	ネットワークおよびホストのセキュリティに依存
高可用性（*HA*）	高可用性のクラウドプラットフォーム上で稼働 データ収集の耐障害性を高めるために Zabbix プロキシを使用可能	ユーザーが HA を管理
SNMP トラップ	専用プロキシ経由のみ（HA/負荷分散されたトラップレシーバーなし）	直接またはプロキシ経由。HA/負荷分散が可能
SNMP ポーリング	プロキシなしでは困難（各デバイスに NAT/カスタムポートが必要）	ネイティブのSNMP ポーリング。プロキシは任意
保持期間の設定	履歴/トレンド/監査は UI 経由のみ（API または設定ファイルは使用不可）	Zabbix のサーバー設定ファイルまたはAPIで設定可能
カスタムスクリプト	`AlertScriptsPath`、`ExternalScripts`、Web インターフェースおよびコミュニティモジュールはサポート対象外	完全にサポート（スクリプトパス、モジュール、インテグレーション）
ODBC 監視	PostgreSQL のみ（ドライバー `{postgresql}`）。MariaDB プラグインはプレースホルダーとして存在	PostgreSQL、MySQL、Oracle などの ODBC を完全に設定可能

機能 / 項目	Zabbix Cloud	Zabbix オンプレミス
ODBC 呼び出しの制限	不可（StartODBCPollers=1のみ）。負荷の高い同期クエリにより、他の処理がブロックされる	ポーラー数とスケジューリングを完全に調整可能
SAML 証明書のアップロード	サポート対象外	UI または API 経由でサポート
スケジュールレポート	独自の Email メディアを作成する必要あり（スクリプトメディアはサポート対象外）	スクリプトメディアと Email メディアの両方をすぐに使用可能
アクティブチェック用のホストインターフェース	プラットフォームが発行する IP アドレス（自動的に作成）	インターフェースはユーザーが管理。IP はユーザーが制御

機能の違い　SNMP トラップ

SNMP トラップは、専用の Zabbix プロキシを介してのみサポートされます。SNMP トラップ監視が必要な場合、SNMP トラップは単一の IP アドレスに送信する必要があるため、プロキシの自動負荷分散や高可用性は使用できません。

SNMP ポーリング

プロキシを使用しない場合、SNMP ポーリングには各デバイスを NAT およびカスタムポート経由で公開する必要があります。

保持設定

履歴、トレンド、監査ログの保持期間は、クラウドの Web インターフェースからのみ設定できます。これらの設定は、`zabbix_server.conf` や API を通じて設定することはできません。

アイテムごとの履歴の手動上書きはサポートされていません（パーティショニングはグローバルに制御されます）。クラウドノードの URL は、オンプレミスインストールと同じクエリパラメータを受け付けません。

カスタムスクリプト

Zabbix Cloud では、以下のカスタムスクリプトタイプはサポートされていません。

- アラートスクリプト (`AlertScriptsPath`)
- 外部スクリプト (`ExternalScripts`)
- Zabbix サーバー上の Web インターフェーススクリプト

コミュニティで開発された Web インターフェースモジュールはインストールできません。

ODBC 監視

Zabbix Cloud は、ODBC 監視を PostgreSQL のみサポートしています。公式の Zabbix ODBC テンプレートを使用し、テンプレート内で接続文字列を以下のように定義してください。

```
Driver={postgresql}
```

MariaDB プラグインもインストールされていますが、現在は動作しません。使用する場合は、以下のように定義してください。

```
Driver={mysql}
```

Attention:
Zabbix Cloud で Oracle MySQL 8.0 が動作する既知の設定はありません。単純なクエリ（`SELECT 1` など）は成功する場合がありますが、より複雑なクエリは `SQL_ERROR` を返します。

 ODBC コールの制限（`StartODBCPollers=1` のみ）はできません。重い同期レポートはパフォーマンスに影響を与える可能性があり、同時に実行できる `SELECT` は 1 つだけです。

インフラストラクチャへのアクセス

Zabbix Cloud は、基盤となるノードへの SSH アクセスを提供せず、またデータベースへの直接接続（例：ポート 3306）も許可しません。すべての設定、監視、トラブルシューティングの作業は、クラウド UI または API を介して実行する必要があり、OS およびデータベース層が分離され安全に保たれるようになっています。

ポート 25 の送信トラフィック

スパムや不正利用を防止するため、Zabbix Cloud ノードから TCP ポート 25 への送信接続は、デフォルトでブロックされています。これは次の項目に影響します。

- ポート 25 の SMTP サーバーを経由して送信するEmailメディアタイプ。新しい Email メディアタイプでは、*SMTP server port* がデフォルトで 25 に設定されているため、SMTP サーバーがサポートしている設定に応じて、587（*Connection security* を STARTTLS に設定）または 465（SSL/TLS）に変更してください。*Office365 relay* メールプロバイダーは、Microsoft のリレエンドポイントにポート 25 で接続するため、使用できません。

- Email メディアタイプが必要なScheduled reports。
- `net.tcp.service[smtp]` simple checksや、network discoveryルールの SMTP チェックなど、クラウドノード自体がポート 25 に対して実行するチェック。これらのチェックでは、SMTP サービスが稼働している場合でも停止していると報告されます。このようなチェックは、代わりに自ネットワーク内の Zabbix プロキシまたは Zabbix エージェントを介して実行してください。

デフォルトの *Cloud Email* メディアタイプは影響を受けません。

アクティブチェックインターフェース

クラウドでのアクティブチェックでは、オンプレミスネットワークとは関係のない IP を持つホストインターフェースが自動的に作成されます。デフォルトでは、この IP は外部 IP である場合があります。一貫性を保つため、作成後にホストインターフェースを手動で削除または調整できます。

SAML 証明書のアップロード

SAML 認証は、暗号化されていない、または署名されていない SAML メッセージを受け入れるプロバイダーがほとんどいないため、サポートされていません。

スケジュールレポート

デフォルトの *Cloud Email* メディアタイプはスクリプトトランスポートを使用するため、スケジュールレポートには使用できません。レポートをメールで送信するには、次の手順を実行します。

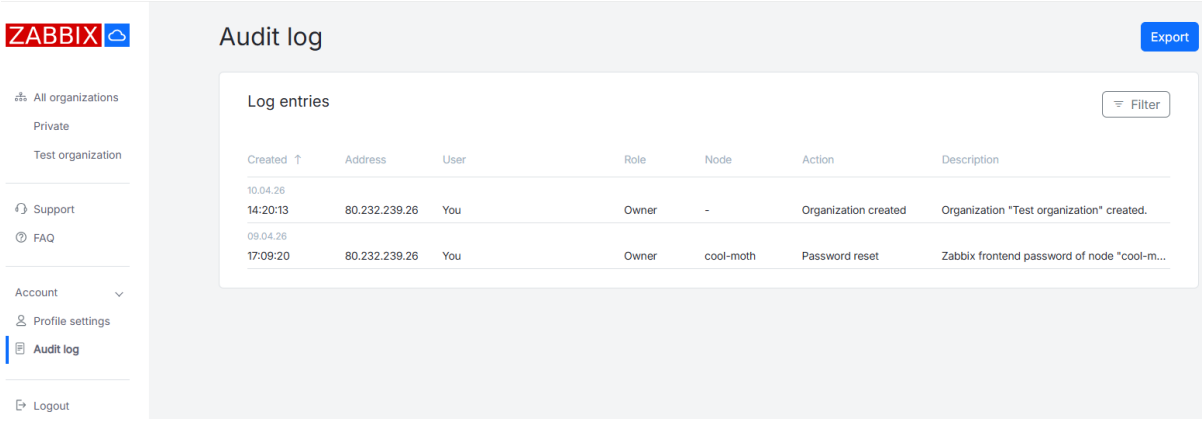
1. 新しい Email メディアタイプを作成します。
2. ユーザー設定 > メディアで、ユーザーにメールメディアを割り当てます。
3. スケジュールレポートの設定時に、そのメディアタイプを選択します。

Attention:
このメディアタイプの *SMTP* サーバーポートにはポート 25 を使用しないでください。ポート 25 の送信トラフィックを参照してください。

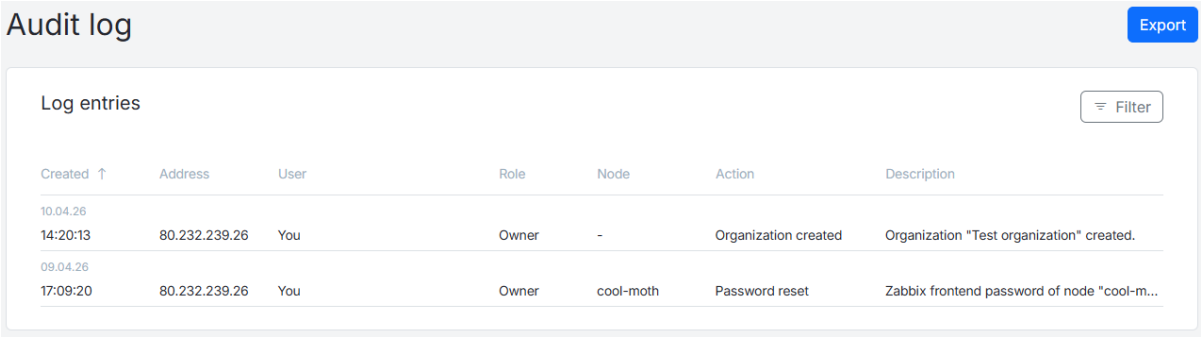
監査ログ

概要　Zabbix Cloud では、監査ログにユーザーおよびシステムのアクティビティが記録されます。

ログインして Zabbix Cloud アカウントにアクセスしたら、アカウント > 監査ログに移動して監査ログの記録を表示します。ユーザーが利用できる監査ログエントリは、そのユーザーの組織ロールによって異なります。



監査ログレコードの表示　監査ログレコードは表形式で表示されます。



この表には、次の情報が含まれます。

Column	Description
Created	ログエントリが作成された日時。表示される時刻は、*Account* > *Profile settings* の *User timezone* ドロップダウンで選択したユーザーのタイムゾーンに対応します。
User	操作を実行したorganization user の名前。
Role	操作を実行したユーザーのrole。
Node	操作に関連するノードのName。
Action	実行された操作の名前。
Description	実行された操作の説明。

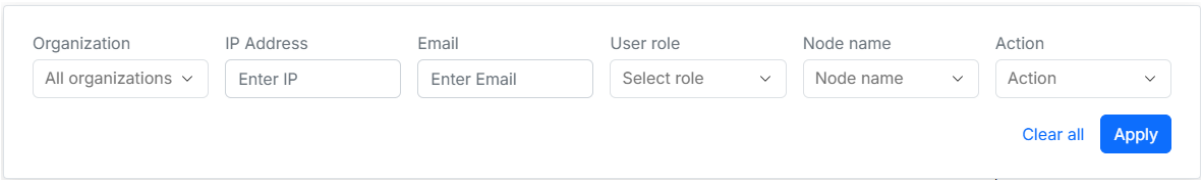
列名をクリックすると、レコードを昇順または降順で並べ替えることができます。

Filtered 監査ログレコードは、*Export* ボタンを使用して `.csv` 形式でエクスポートできます。

フィルターの使用

Filter ボタンを使用してフィルターを開きます。

1 つ以上のフィルタリングパラメータを選択した後、*Apply* をクリックすると、フィルターされた結果が表示されます。フィルターをクリアするには、*Clear all* をクリックします。



1 つ以上のパラメータでレコードをフィルタリングできます。次のフィルタリングパラメータを使用できます。

パラメータ	説明
Organization	すべての組織または特定の組織を選択します。
IP Address	IP アドレスを指定します。
Email	ユーザーに関連付けられたメールアドレスを指定します。
User role	すべてのロール、または特定のロール（Owner、Admin、User）を選択します。
Node name	すべてのノードまたは特定のノードを選択します。
Action	特定のアクションタイプを選択します。

監査ログには、次のアクションが記録されます。

- ノードを作成
- ノードを削除
- ノードをアップグレード
- ノードバージョンを更新
- パスワードをリセット
- メンテナンスを更新
- 履歴設定を更新
- ノードの有効期限を延長
- プロキシ設定を更新
- バックアップを作成
- バックアップを削除
- バックアップを復元

- バックアップを更新
- アクセスフィルターを追加
- アクセスフィルターを更新
- アクセスフィルターを削除
- 暗号化を有効化
- 暗号化を無効化
- サブスクリプションをキャンセル
- サブスクリプションを更新
- サブスクリプションの支払いに失敗
- 支払いに確認が必要
- 組織を作成
- 組織を削除
- 管理者を招待
- ユーザーを招待
- 管理者が参加
- ユーザーが参加
- 管理者を組織から削除
- ユーザーを組織から削除
- ノードを組織に追加
- ノードを組織から削除
- ノードへのアクセスを許可
- ノードへのアクセスを取り消し
- ユーザー設定を更新
- 支払いを作成
- 支払いを完了
- 支払いに失敗
- *VAT* 請求を適用
- *VAT* の検証に失敗
- 請求情報を更新
- 支払い方法を更新
- 支払い方法を削除
- 前払いノードを作成
- 前払いノードを更新
- インシデントを開始
- インシデントを終了

Zabbix Cloud プロキシをセットアップする

概要　このページでは、Zabbix Cloud にデプロイされた Zabbix プロキシを設定し、リモート環境のホストから監視データを収集する方法について説明します。

この手順は、主に Zabbix Cloud 上に作成された Zabbix サーバーとプロキシを対象としています。

また、Zabbix サーバーが Zabbix Cloud でホストされていない場合に必要な設定についても説明します。

前提条件　Zabbix サーバーがインストール、設定され、実行されていること。または、Zabbix Cloud 上で Zabbix サーバーノードが実行されていること。

プロキシノードを作成する　Zabbix Cloud プロキシノードを作成します。

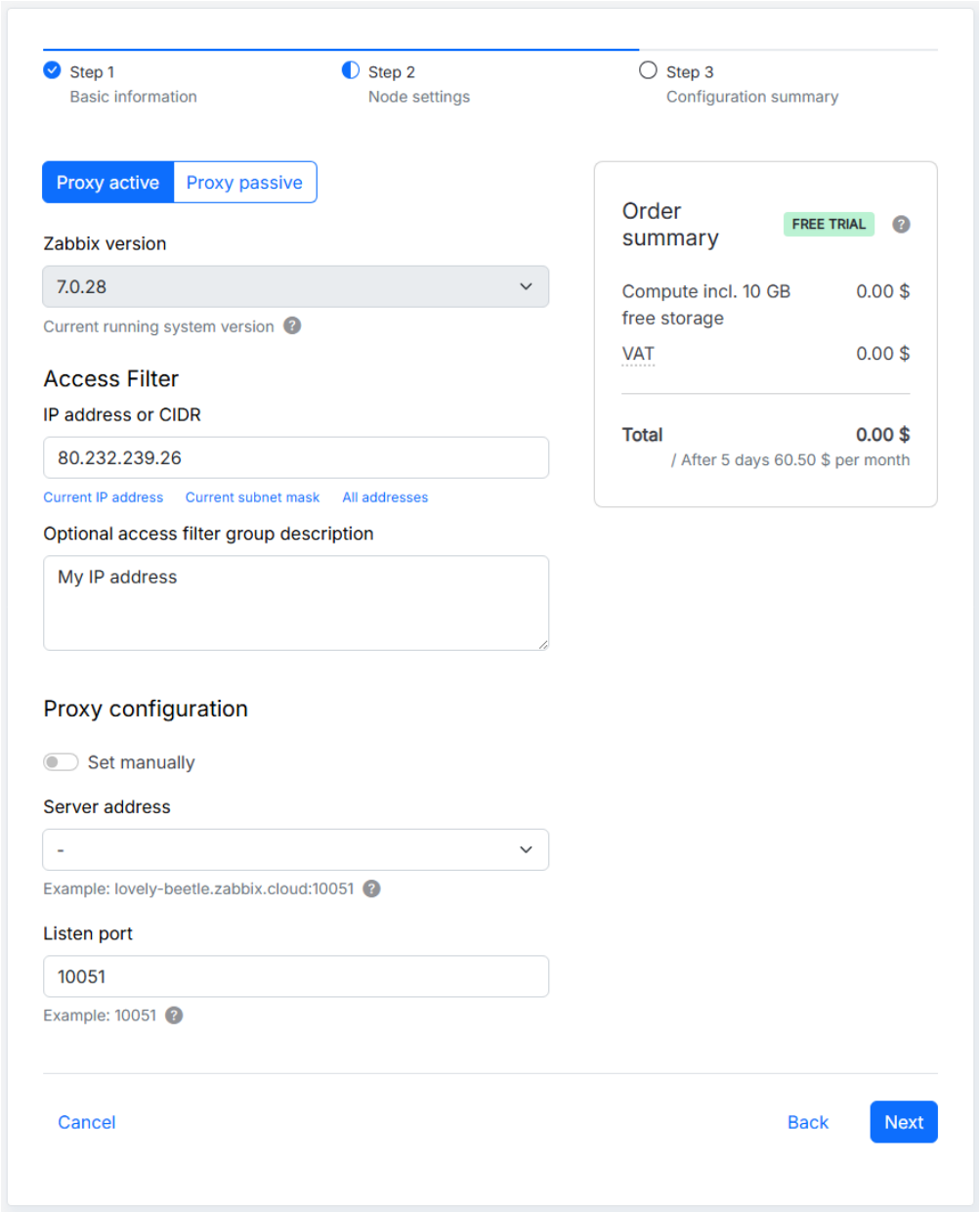
ノードの作成時に、次の手順を実行します。

1. *Proxy* を選択します。

2. プロキシが Zabbix サーバーと通信する方法に応じて、*Proxy active* または *Proxy passive* を選択します。

3. ノードの IP アドレスまたは CIDR アクセスフィルターを設定します（ノードの作成後に設定することもできます）。

4. プロキシが接続する Zabbix サーバーを指定します。



5. ノードの作成を完了します。

プロキシノードの作成について詳しくは、クラウドに Zabbix をデプロイするを参照してください。

アクセスフィルターの設定　Zabbix Cloud では、アクセスフィルターを使用して、クラウドノードに接続できる IP アドレスまたはサブネットを制御します。

ノードの初期化中に、ノードへのアクセスに使用された IP アドレスに対するアクセスフィルターが自動的に作成されます。これには、監視対象のホストやその他の Zabbix コンポーネントのアドレスが必ずしも含まれるとは限りません。必要なアドレスは手動で追加してください。

詳細については、ノードの設定 — アクセスフィルターを参照してください。

プロキシノード

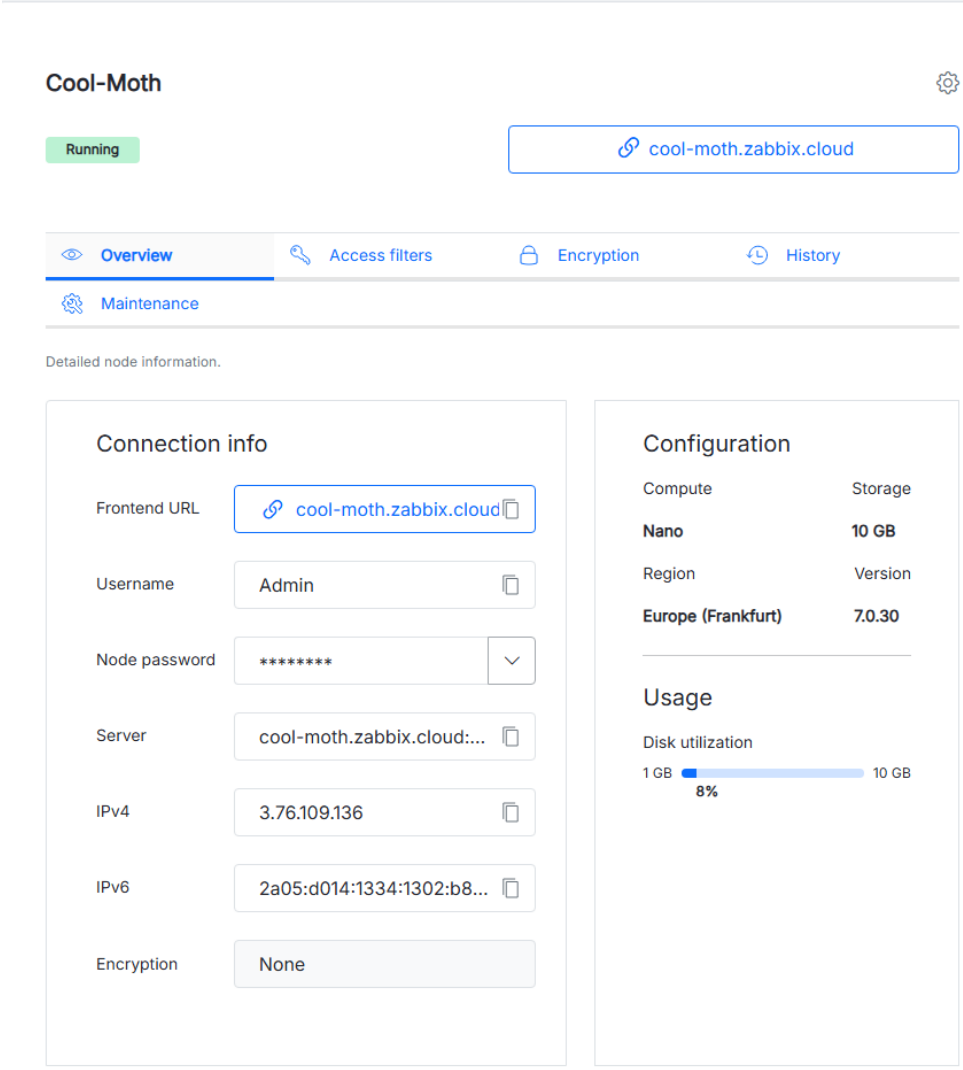
プロキシへの接続元として想定される IP アドレスまたはサブネットを追加します。

監視対象のホストでアクティブエージェントチェックを使用している場合、エージェントはプロキシへの接続を開始します。そのため、監視対象ホストの IP アドレスまたはサブネットを、プロキシノードのアクセスフィルターで許可する必要があります。

Proxy Passive をプロキシモードとして設定した場合、Zabbix サーバーがプロキシに接続します。そのため、Zabbix サーバーの IP アドレスまたはサブネットを、プロキシノードのアクセスフィルターで許可する必要があります。

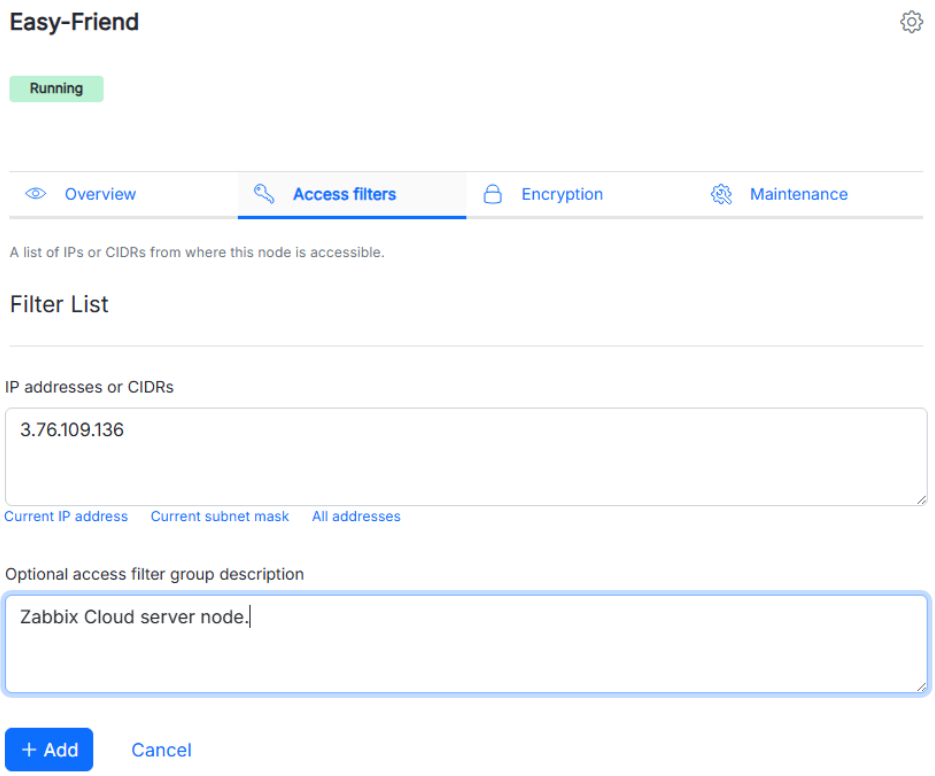
Proxy passive を選択し、サーバーノードを使用する場合:

1. サーバーノードの *Overview* タブに移動し、ノードの IPv4 または IPv6 アドレスをコピーします。

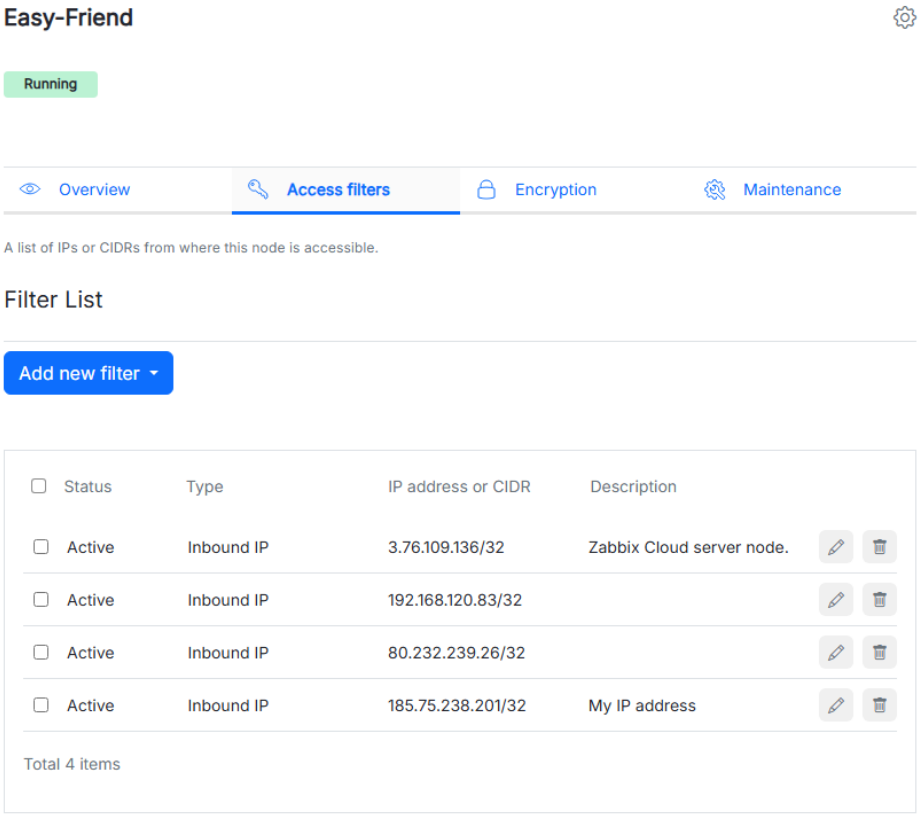


2. プロキシノードの *Access filters* タブに移動し、*Add new filter* > *Add* をクリックします。

3. コピーしたアドレスを *IP addresses or CIDRs* フィールドに貼り付け、*Optional access filter group description* フィールドに説明を追加します。



4. *Add* をクリックします。



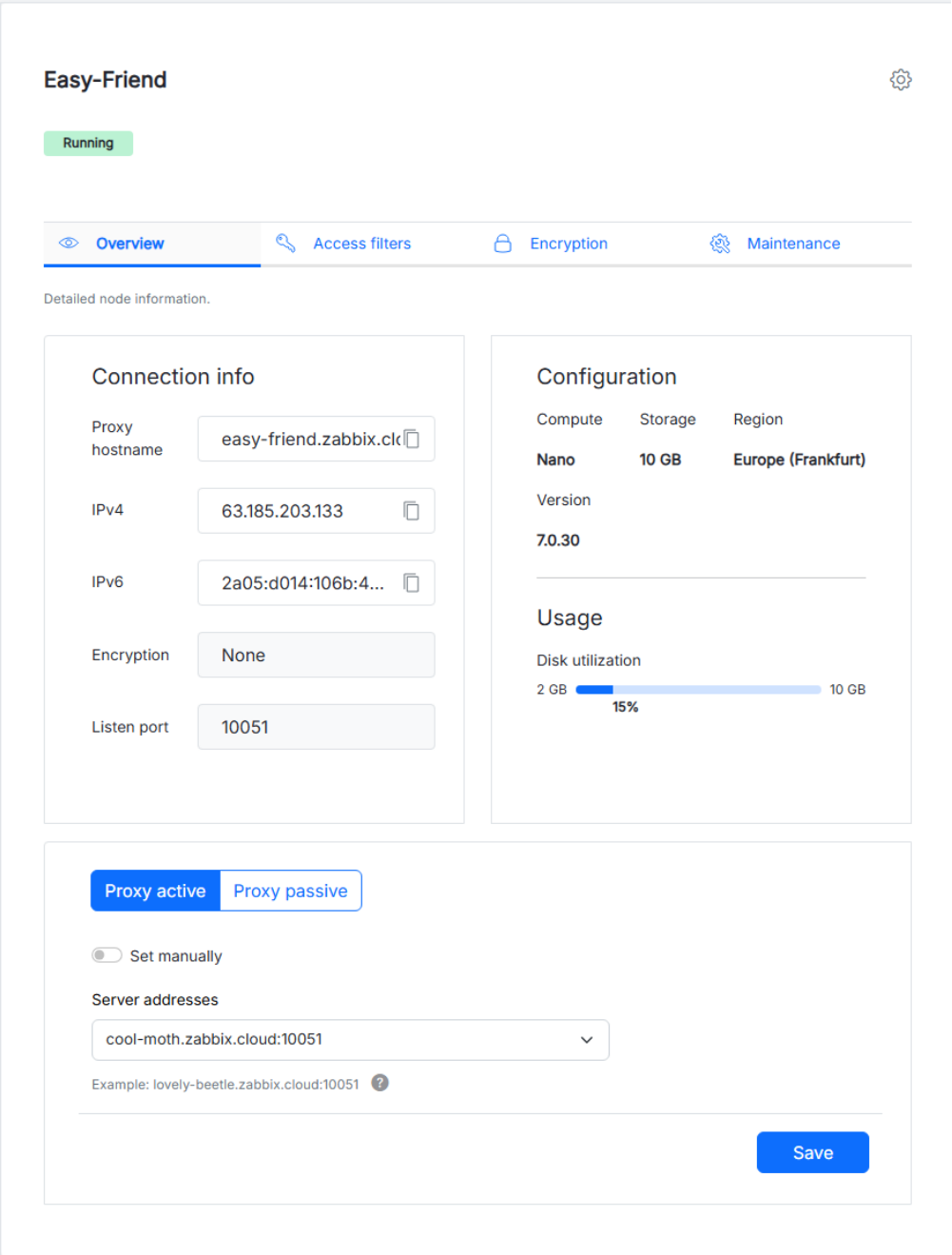
アクティブチェックを使用する Zabbix エージェントの IP アドレスについて、手順 2～4 を実行します。

サーバーノード

Zabbix サーバーが Zabbix Cloud 上でホストされている場合は、サーバーノードでもアクセスフィルターを設定します。

アクティブモードのプロキシの場合、プロキシがサーバーへの接続を開始するため、プロキシノードの IP アドレスまたは CIDR をサーバーノードのアクセスフィルターに追加します。

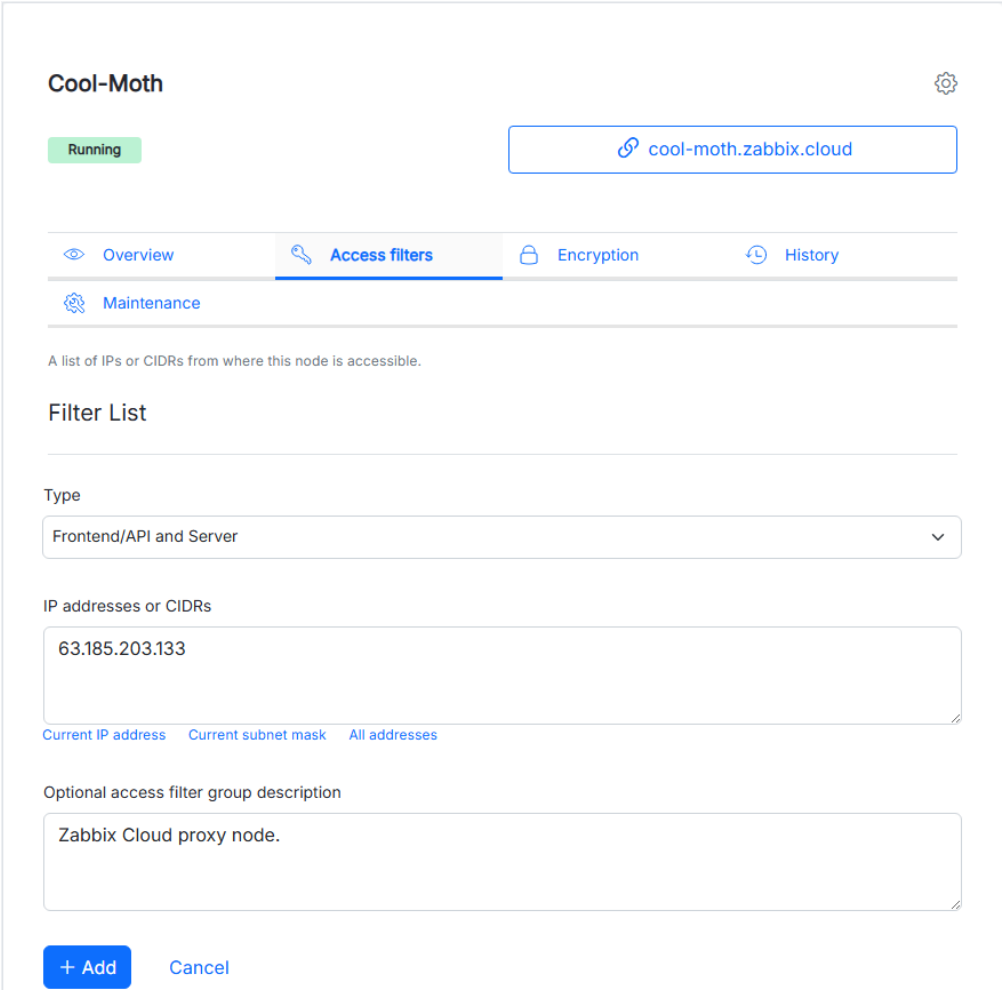
1. プロキシノードの概要タブに移動し、ノードの IPv4 または IPv6 アドレスをコピーします。



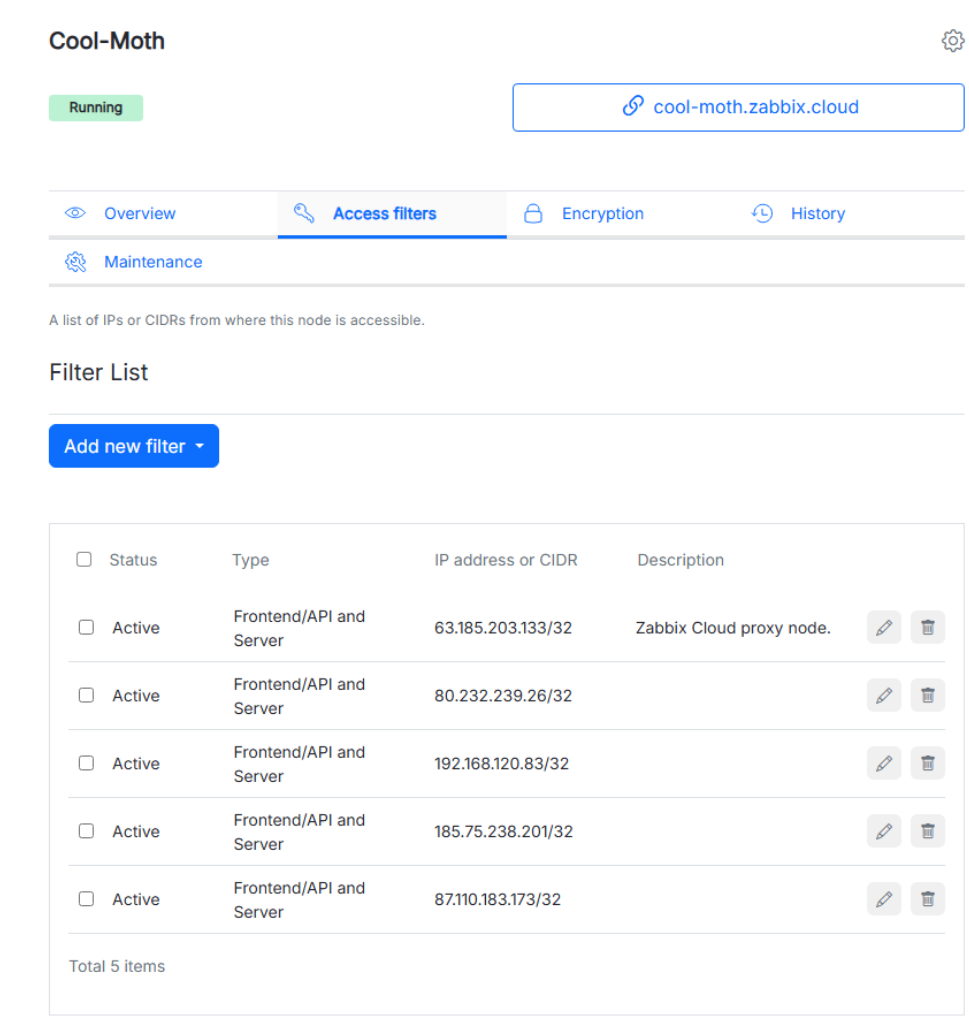
2. サーバーノードのアクセスフィルタータブに移動し、新しいフィルターを追加 > 追加をクリックします。

3. コピーしたアドレスを *IP* アドレスまたは *CIDR* フィールドに貼り付け、アクセスフィルターグループの説明（オプション）フィールドに説明を追加します。

4. 適切なフィルタータイプを選択します。



5. 追加をクリックします。



プロキシの設定　Zabbix サーバーと通信するには、Zabbix の Web インターフェースでプロキシを設定する必要があります。

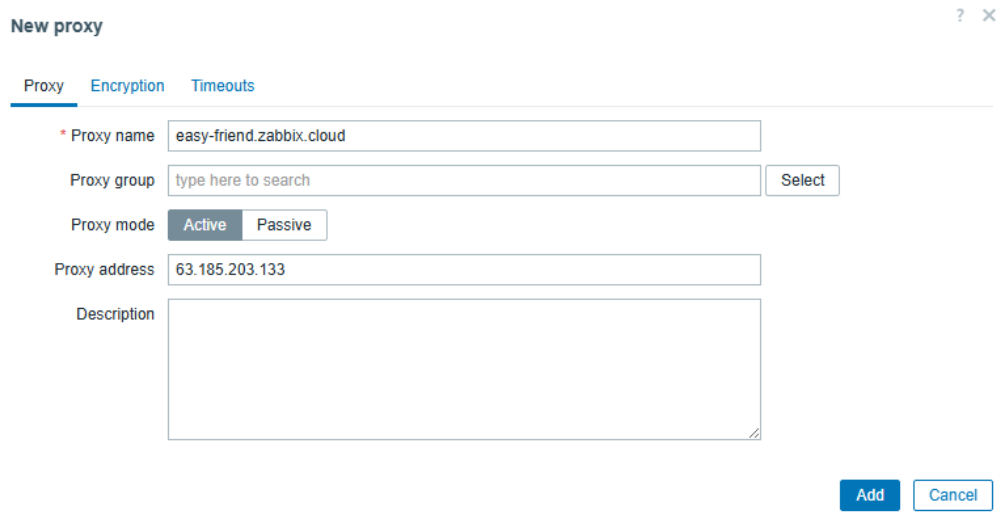
Zabbix の Web インターフェースで 管理 > プロキシを開き、プロキシの作成をクリックして、必要に応じてプロキシを設定します。

1. プロキシ名フィールドにプロキシ名を追加します。
プロキシに設定されているプロキシのホスト名と一致している必要があります。
入力するか、プロキシの 概要タブからコピーします。

2. プロキシの作成時に選択したものに対応するプロキシモードを選択します。

3. アクティブプロキシの場合は、プロキシの IP アドレスまたは DNS 名を入力します。

4. 追加をクリックします。



詳細については、プロキシを使用した分散監視を参照してください。

監視対象ホストの設定　プロキシを介して監視するホストを追加します。

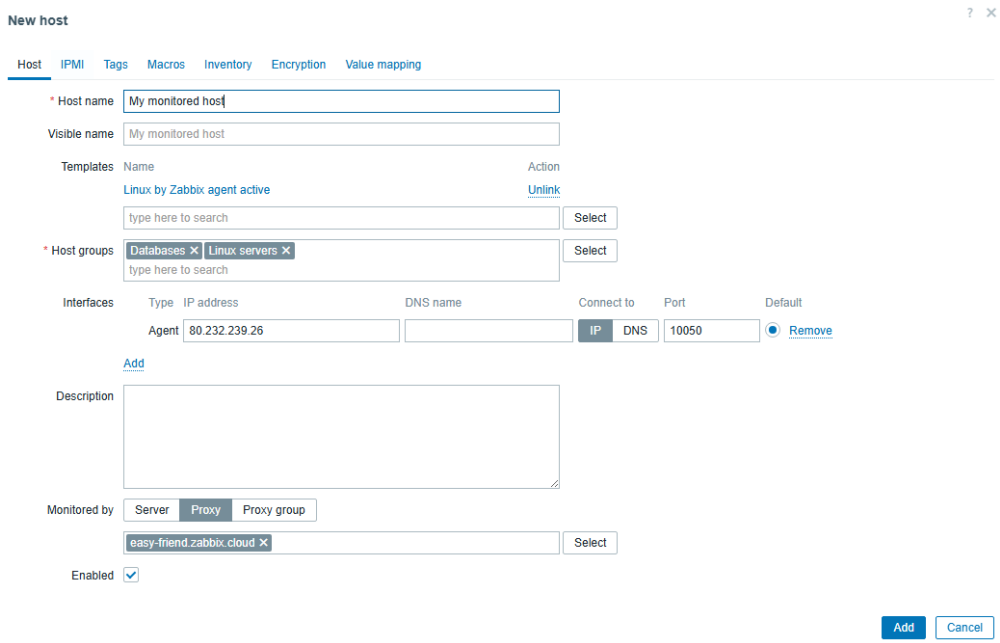
1. データ収集 > ホストに移動します。

2. ホストを作成するか、既存のホストを開きます。

3. 監視方法で プロキシを選択し、プロキシ名を入力して検索・選択するか、選択ボタンを押してプロキシの一覧から選択します。

4. ホストインターフェースとその他のパラメータを設定します。

5. 追加または 更新をクリックします。



プロキシによる監視対象として設定されたホストは、アイテムのタイプとエージェントのモードに応じて、そのプロキシを介して監視データを送受信します。

Zabbix エージェントの設定　監視対象のホストで Zabbix エージェントまたは Zabbix エージェント 2 を使用する場合、適切な Zabbix コンポーネントと通信するようにエージェントを設定します。

設定は、エージェントがパッシブチェックとアクティブチェックのどちらを使用するかによって異なります。

パッシブエージェントチェック

パッシブチェックでは、Zabbix プロキシがエージェントへの接続を開始し、監視データを要求します。

Zabbix エージェントの設定ファイルを開き、Zabbix Cloud プロキシの IP アドレスからの接続を許可するように Server パラメータを更新します。

例:

```
Server=<proxy_IP>
```

監視対象のホストで、デフォルトでは 10050 のエージェントポートへのプロキシからの受信 TCP 接続が許可されていることを確認してください。

プロキシは、ネットワーク経由で監視対象のホストに到達できる必要があります。

アクティブエージェントチェック

アクティブチェックでは、エージェントが Zabbix プロキシへの接続を開始し、収集した監視データを送信します。

Zabbix エージェントの設定ファイルを開き、ServerActive パラメータを Zabbix Cloud プロキシのアドレスで更新します。

例:

ServerActive=<proxy_IP>:<proxy_listen_port>

<proxy_IP> を Zabbix Cloud プロキシの IP アドレスに、<proxy_listen_port> をプロキシノードの作成時に設定したリッスンポートに置き換えます。デフォルトのリッスンポートは 10051 です。

ServerActive にプロキシを指定すると、エージェントは Zabbix サーバーから直接ではなく、プロキシからアクティブチェックの設定を取得します。

監視対象のホストから、設定したリッスンポートでプロキシへの送信 TCP 接続を確立できることを確認してください。

パッシブエージェントチェックとアクティブエージェントチェックの詳細については、パッシブエージェントチェックとアクティブエージェントチェックを参照してください。

外部 **Zabbix** サーバー　Zabbix Cloud プロキシは、Zabbix Cloud の外部にデプロイされた Zabbix サーバーとも通信できます。

この場合、外部 Zabbix サーバーは、選択したプロキシモードに従ってクラウドプロキシと通信できる必要があります。

アクティブモードのプロキシでは、プロキシが外部 Zabbix サーバーへの接続を開始します。プロキシからの接続を受け入れるように外部 Zabbix サーバーを設定し、プロキシがネットワーク経由でサーバーに到達できることを確認してください。

パッシブプロキシでは、Zabbix サーバーがプロキシへの接続を開始します。外部 Zabbix サーバーのIP アドレスまたはサブネットをプロキシノードのアクセスフィルターに追加し、サーバーがネットワーク経由でプロキシに到達できることを確認してください。

外部 Zabbix サーバーでは、Zabbix Cloud のアクセスフィルターは使用されません。代わりに、外部環境で必要なファイアウォールおよびネットワークルールを設定してください。

監視対象のホストは、外部 Zabbix サーバーと同じ環境、別のネットワーク、またはプロキシからアクセス可能なネットワークに配置できます。

エージェントがアクティブチェックを使用する場合は、ServerActive パラメータが Zabbix Cloud プロキシを指すように設定してください。エージェントがパッシブチェックを使用する場合は、プロキシからの接続を許可するように Server パラメータを設定してください。

設定を確認する　プロキシ、エージェント、アクセスフィルター、ネットワーク接続を設定したら、プロキシが Zabbix サーバーと通信していることを確認します。

1. 管理 > プロキシに移動します。

2. プロキシの 状態を確認します。

3. 監視 > 最新データに移動し、最近のアイテムの値が利用可能であることを確認します。

プロキシがサーバーと通信していない場合は、以下を確認します。

- プロキシモード
- サーバーのアドレスとポート
- プロキシのアクセスフィルター
- サーバーが Zabbix Cloud でホストされている場合は、サーバーのアクセスフィルター
- エージェントの設定
- ファイアウォールおよびルーティングルール
- Server または ServerActive に設定されているアドレス

モバイルアプリ

モバイルアプリ

Zabbix モバイルアプリを使用すると、モバイルデバイスから Zabbix 環境を監視できます。障害の表示、詳細の確認、障害の更新、対応が必要な場合のプッシュ通知の受信が可能です。

アプリを入手（近日公開）

アプリケーション用に **Zabbix** をセットアップする

Zabbix ブリッジアダプターをセットアップする　モバイルデバイスから Zabbix に接続できるように、Zabbix ブリッジアダプターをインストールします。

デバイスの追加　Zabbix Web インターフェースでモバイルデバイスを追加し、Zabbix サーバーに接続します。

プッシュ通知を設定する　障害が発生したときに、モバイルデバイスでアラートを受け取ります。

アプリについて学ぶ

アプリの概要　Zabbix モバイルアプリの基本的な使用方法を学びます。

安全なモバイルアクセス　アプリ、モバイルデバイス、Zabbix ブリッジアダプターを保護し、紛失または侵害されたデバイスからのアクセスを取り消します。

アプリ用に **Zabbix** を設定する

Zabbix モバイルアプリ（近日公開予定）を使用するには、まず次の操作を行う必要があります。

1. Zabbix ブリッジアダプターを設定します。
2. プッシュ通知を設定します。
3. Zabbix の Web インターフェースでモバイルデバイスを追加します。

以下のセクションでは、それぞれの手順について説明します。

設定後は、アプリと Zabbix ブリッジアダプターのセキュリティを維持するために、モバイルアクセスの保護を参照してください。

Zabbix ブリッジアダプターをセットアップする　Zabbix サーバーがモバイルアプリと通信できるようにするには、Zabbix ブリッジアダプターをセットアップする必要があります。

ブリッジアダプターは、パッケージを使用して、Zabbix サーバーと同じシステム、または別のシステムにインストールできます。

> **Attention:**
> 以下の手順は Ubuntu 24.04 を対象としています。その他のオペレーティングシステムの場合は、お使いの OS 向けのZabbix の入手ページからリポジトリおよびパッケージのインストールコマンド（wget、dpkg、apt など）を取得し、それらを使用して zabbix-bridge-adapter パッケージをインストールしてください。

1. root 権限で新しいシェルセッションを開始します。

```
sudo -s
```

2. Zabbix リポジトリをインストールします。

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Zabbix ブリッジアダプターをインストールします。

```
apt install zabbix-bridge-adapter
```

4. Zabbix サーバー設定ファイルを編集します。

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Zabbix サーバーを再起動します。例（その他のオプションを参照）：

```
systemctl restart zabbix-server
```

6. レポート > システム情報から Zabbix サーバー ID をコピーします。次の手順で使用します。

7. Zabbix ブリッジアダプター設定ファイルを編集します。

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

> **Attention:**
> DataDir で指定したディレクトリには 0700 の権限が必要です。また、そのディレクトリに保存されるすべてのファイルには 0600 の権限が必要です。

8. Zabbix ブリッジアダプターのエンドユーザーライセンス契約（EULA）を確認して同意します。

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Zabbix ブリッジアダプターを起動します。

```
systemctl start zabbix-bridge-adapter
```

Attention:
起動すると、Zabbix ブリッジアダプターはサーバー ID を使用して、Zabbix サーバーを中央ホスト型の Zabbix Bridge に登録します。登録状態は、DataDir内の状態ファイルに保存されます。ブリッジアダプターを再インストールして状態ファイルが失われると、同じ Zabbix サーバーの登録に失敗する可能性があります。これを避けるには、ブリッジアダプターを再インストールする前に状態ファイルをバックアップし、再インストール後に復元してください。

プッシュ通知を設定する　監視環境で対応が必要な事象が発生したときにモバイルデバイスでアラートを受け取るには、次の手順を実行します。

1. プッシュ通知メディアタイプを作成します（またはデフォルトのメディアタイプを有効にします）。
2. アクションを作成します（またはデフォルトの *Zabbix* 管理者に障害を報告トリガーアクションを有効にします）。
3. デバイスを追加して、メディアタイプをユーザーに自動的に追加します（またはユーザーメディアを手動で追加します）。

デバイスの追加　デバイスで Zabbix モバイルアプリを使用するには、Zabbix の Web インターフェースでそのデバイスを追加する必要があります。手順については、以下を参照してください。

- ユーザー > デバイス（スーパーユーザーとして、自分または他のユーザーのモバイルデバイスを追加）
- ユーザー設定 > デバイス（任意のユーザーとして、自分のモバイルデバイスを追加）

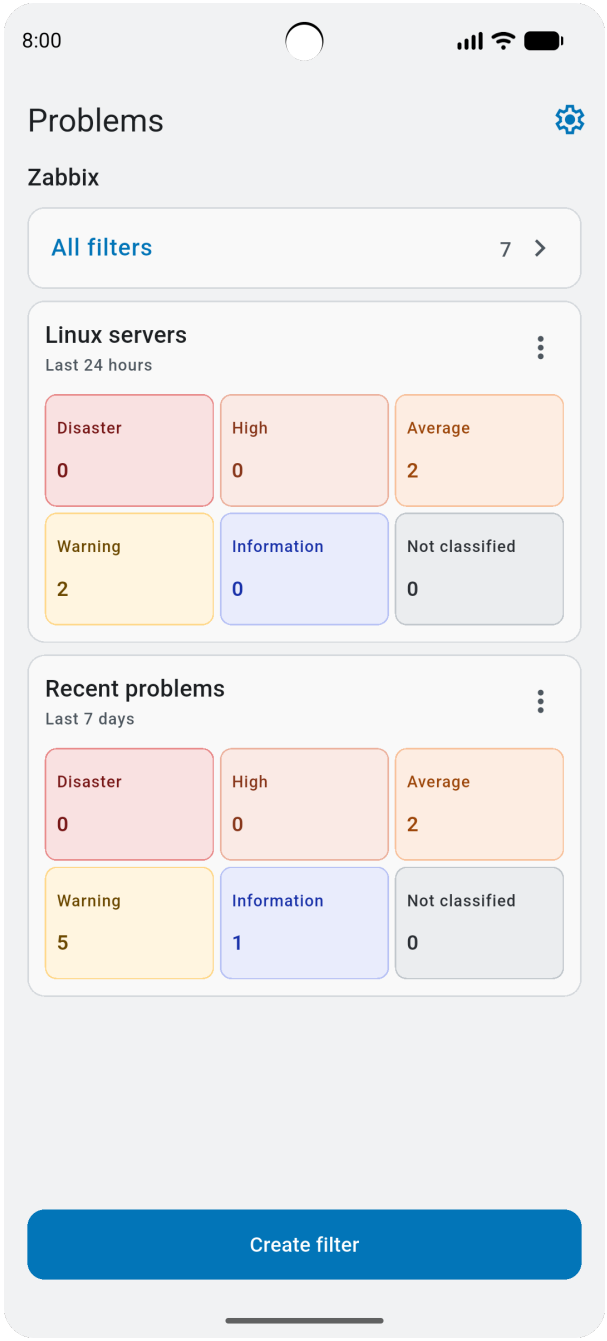
アプリの概要

Zabbix モバイルアプリ（近日公開予定）を使用すると、モバイルデバイスから Zabbix 環境を監視できます。障害の表示、詳細の確認、障害の更新を行えるほか、対応が必要な事象が発生したときにプッシュ通知を受け取ることができます。

このページでは、モバイルアプリの概要と使用方法について説明します。

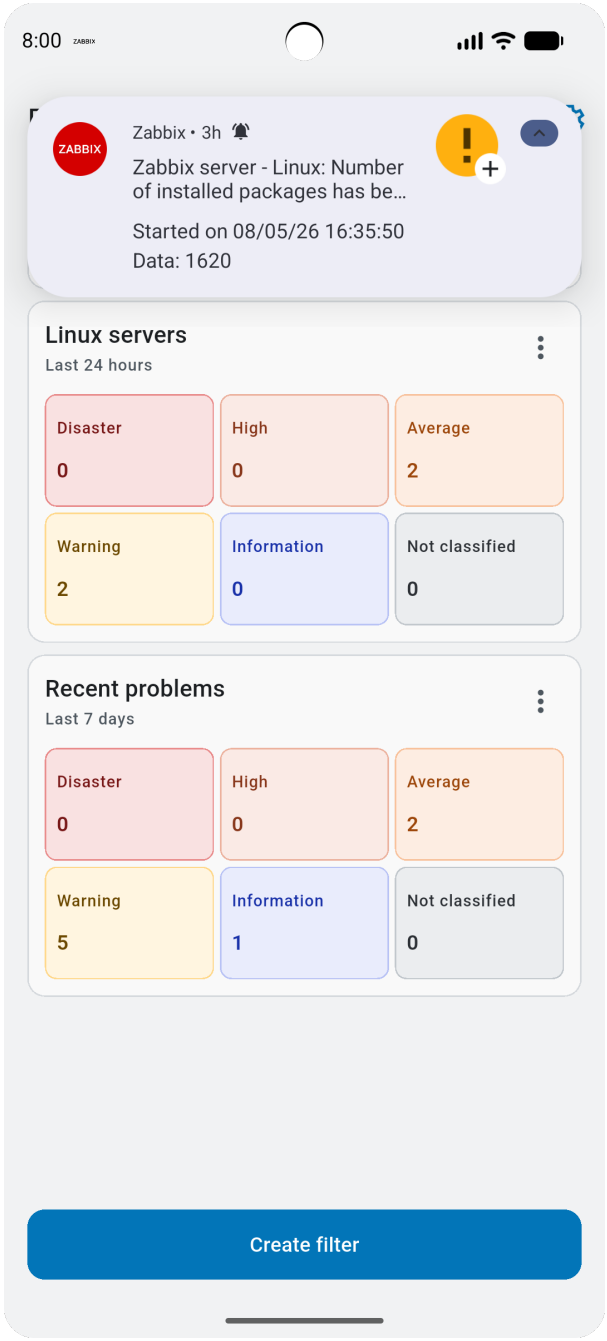
障害　障害画面には、ピン留めしたフィルターに基づく障害の概要が表示されます。以下が含まれます。

- 設定アイコン - タップするとアプリの設定が表示されます。
- すべてのフィルター - タップするとフィルターの表示と管理ができます。
- Zabbix サーバー名（Zabbix Web インターフェースのインストール時に設定されていない場合は *Zabbix*）。
- ピン留めしたフィルター - フィルター名または深刻度をタップすると、障害一覧画面が開きます。
  3 点メニューをタップすると、フィルターの編集、複製、ピン留め解除、削除ができます。
  ピン留めしたフィルターはアルファベット順に並び、最近の障害フィルターは編集および削除できないことに注意してください。
- フィルターの作成 - タップすると新しいフィルターを作成できます。



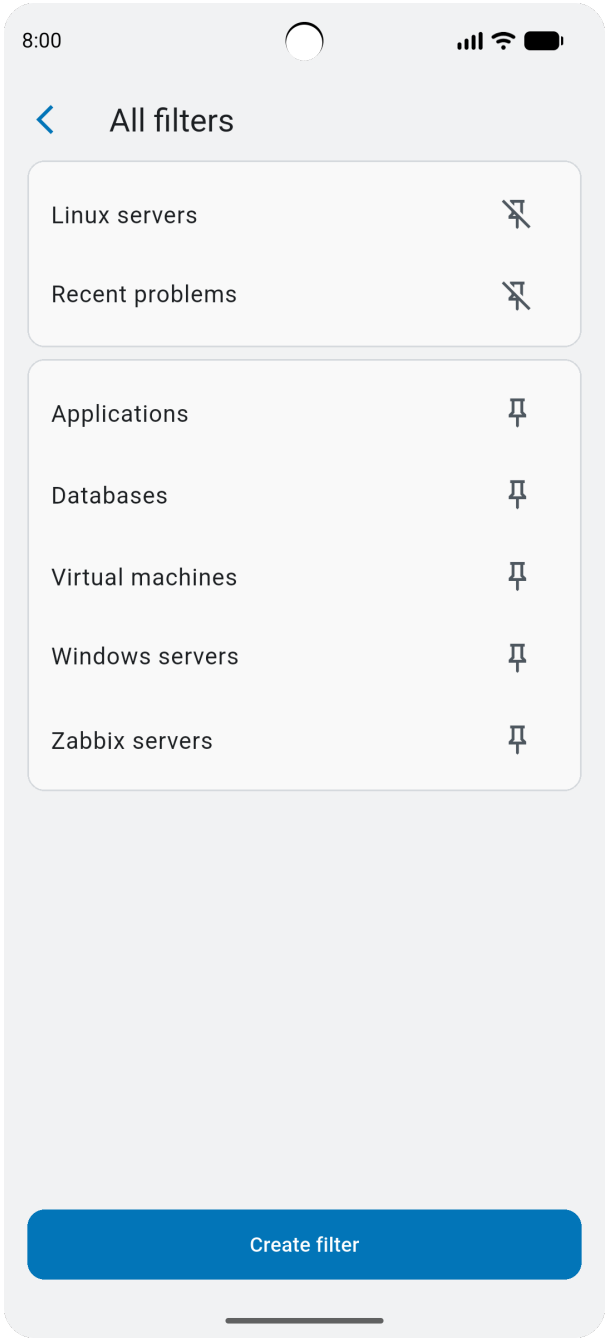
通知　Zabbix サーバーが障害を検知すると、モバイルデバイスに通知を送信できます。通知には、障害が発生したホスト、名前、およびpush 通知メディアタイプ用に設定されたメッセージテンプレートから生成されたメッセージが含まれます。

通知をタップすると、障害の詳細画面が開きます。



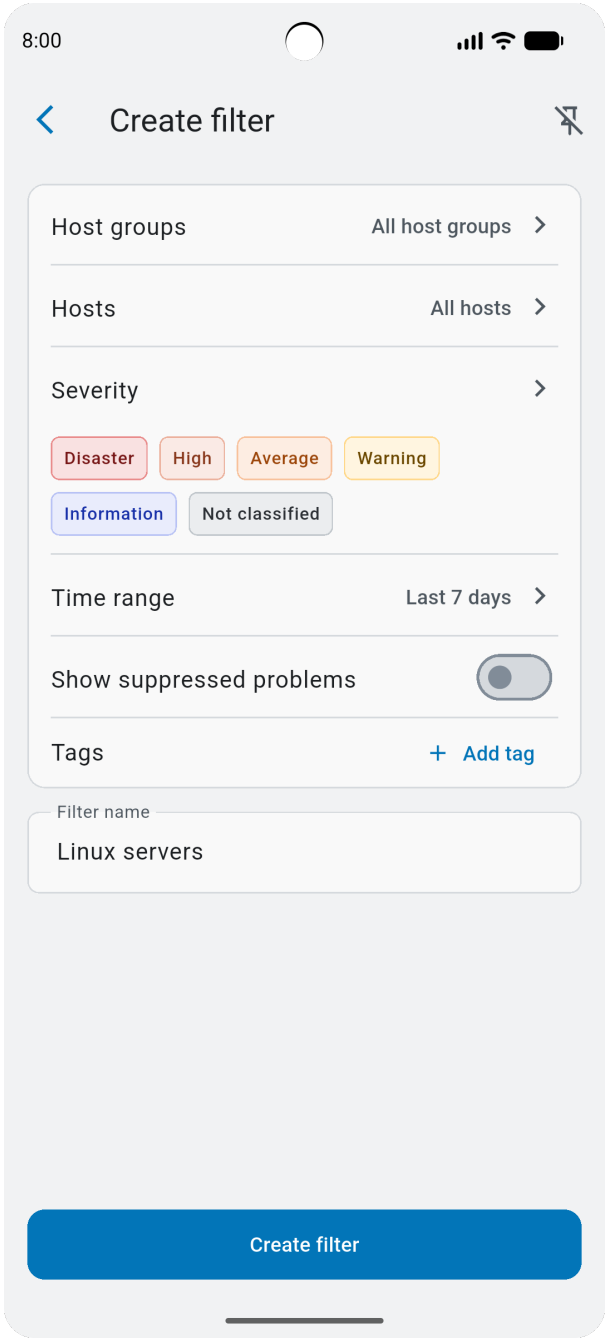
すべてのフィルター　すべてのフィルター画面では、フィルターを管理できます。以下が含まれます。

- フィルター名 - タップしてフィルターを編集します。
- ピンアイコン - タップして、障害画面にフィルターをピン留め、またはピン留めを解除します。
- フィルターの作成 - タップして新しいフィルターを作成します。



フィルターの作成　フィルターの作成画面では、監視 > 障害のオプションと同様のオプションを使用して、フィルターを作成できます（詳細を参照）。未解決の障害を表示する期間として、時間範囲を設定することもできます。

フィルターの作成をタップして、フィルターを保存します。新しいフィルターは障害 画面にピン留めされます。これを変更するには、ピンアイコンをタップします。



障害一覧　障害一覧画面には、フィルターに一致する障害（例: 最近の障害）が表示されます。

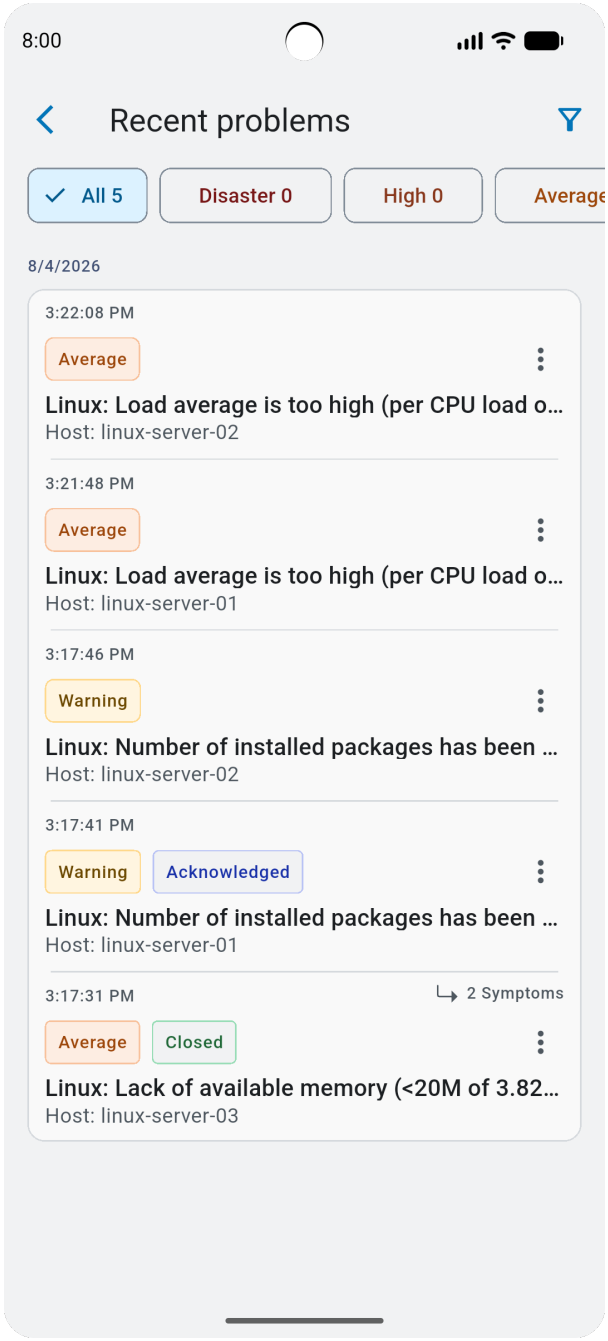
フィルターアイコンまたは障害の深刻度をタップすると、一覧をさらに絞り込めます。

各障害について、次の情報を確認できます。

- 開始時刻と日付。
- 深刻度、名前、ホスト。
- 障害が認識済みか、最近クローズされたか（自動的に解決されたか、手動でクローズされたか）。
- 症状があるかどうか。タップすると症状の障害一覧が表示されます。

障害をタップすると、障害の詳細画面が開きます。

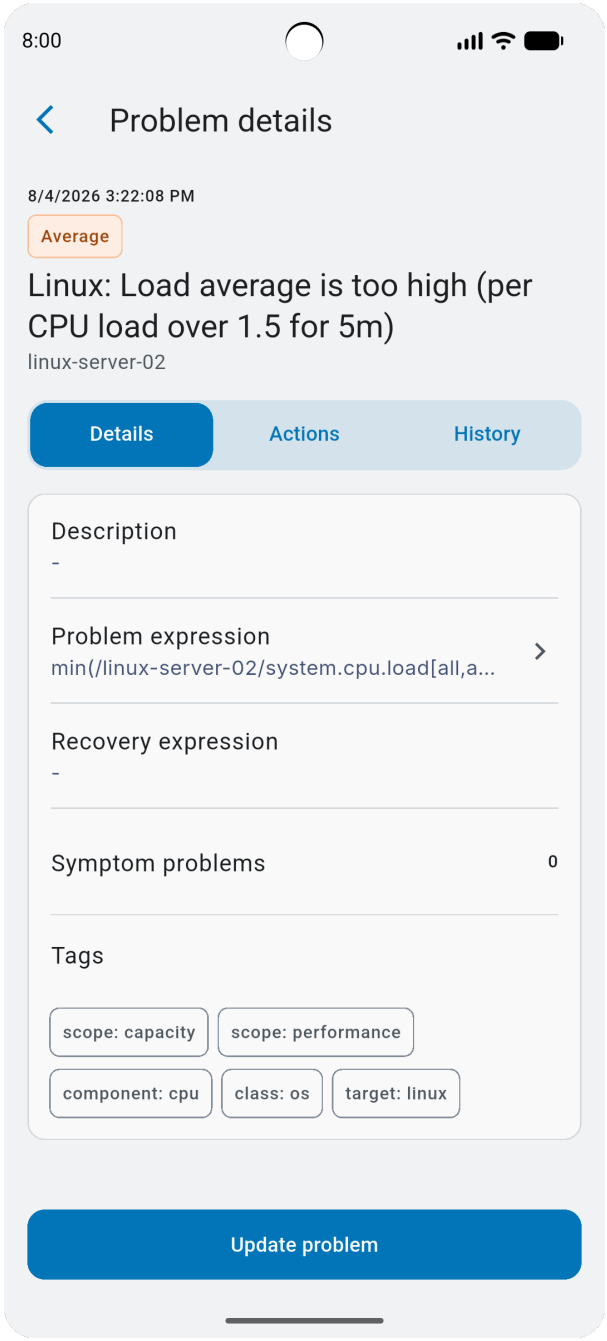
3 点メニューをタップすると、障害をすばやく認識したり、指定した時間だけ抑制したりできます。



障害の詳細　障害の詳細画面には、障害の詳細、アクション、履歴が表示されます。

- 詳細タブ - 障害の説明、エクスプレッション、症状の障害（タップすると障害の一覧が表示されます）、タグが表示されます。
- アクションタブ - 障害のタイムライン（障害の発生時刻または認識時刻、通知が送信されたかどうか、その他のアクション）が表示されます。
- 履歴タブ - 障害の経過時間、継続時間、復旧時間が表示されます（障害が継続中の場合は–）。

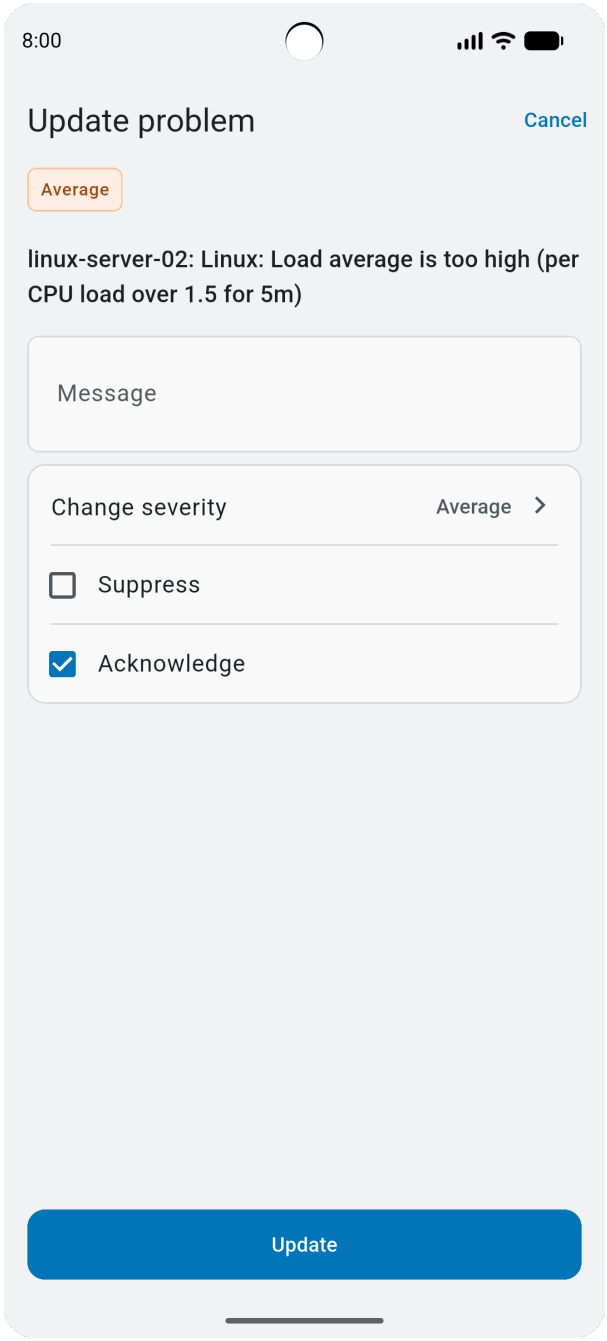
障害を更新するには、障害の更新をタップします。



障害の更新　障害の更新画面では、障害を更新できます。

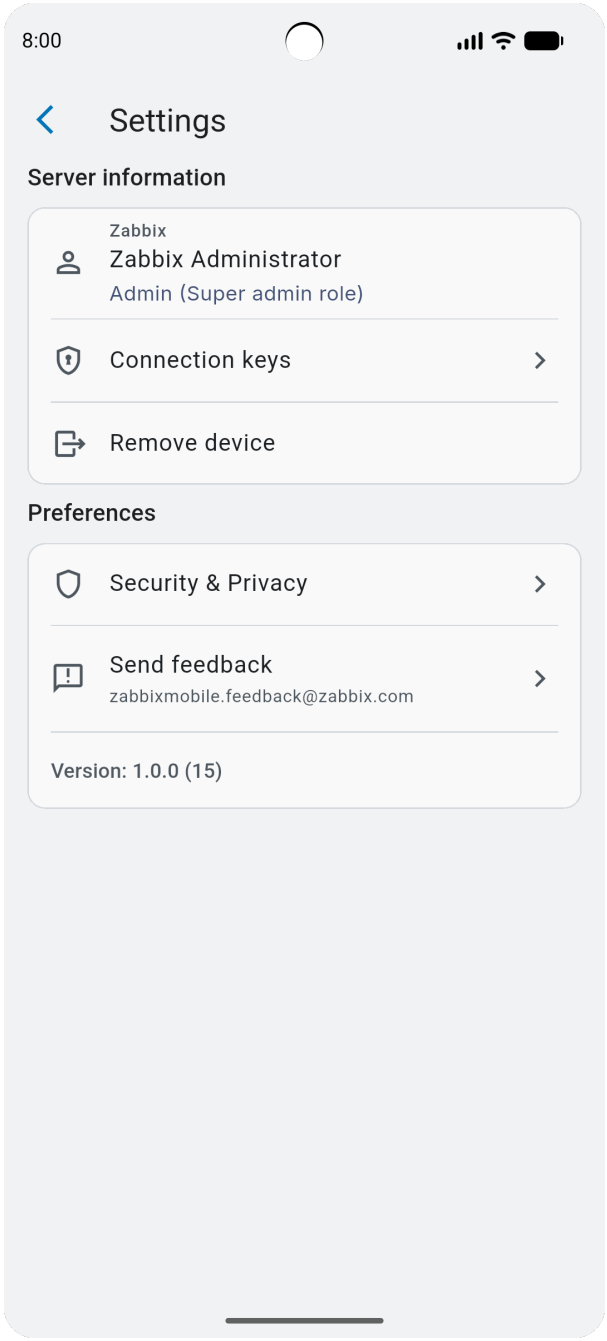
- メッセージを追加します。
- 障害の深刻度を変更します。
- 任意の時間、障害を抑制します。
- 障害を確認または確認解除します。
- 障害をクローズします。

更新をタップして、変更を保存します。



設定　設定画面では、アプリの設定にアクセスできます。以下の項目が含まれます。

- Zabbix サーバー名（Zabbix Web インターフェースのインストール時に設定されていない場合は、*Zabbix*）。
- モバイルデバイスが追加されているユーザーの名前、タイプ、およびロール。
- 接続キー - 開くと、サーバー ID、デバイス ID、その他の詳細を確認してコピーできます。
- デバイスを削除 - タップすると、Zabbix からモバイルデバイスを削除できます。
- セキュリティとプライバシー - 開くと、アプリロックを有効にしたり、診断データの共有を管理したりできます。
- フィードバックを送信 - 開くと、Zabbix にフィードバックを送信できます。
- アプリのバージョン。



セキュアなモバイルアクセス

このページでは、Zabbix モバイルアプリと Zabbix ブリッジアダプターを安全に保つためのベストプラクティスについて説明します。

モバイルデバイスの保護　モバイルデバイス上のアプリを他の人が開けないようにするには、アプリの設定 でアプリロックを有効にします。

デバイスを紛失した場合や侵害された場合は、Zabbix の Web インターフェースから削除してください。これにより、そのデバイスのアクセス認証情報が無効化および削除され、デバイスから Zabbix にアクセスできなくなります。ユーザーのロールの権限 に応じて、次のセクションでデバイスを削除できます。

- ユーザー設定 > デバイス - 任意のユーザーが自分のデバイスを削除できます（自分のデバイスの管理権限が必要です）。
- ユーザー > デバイス - スーパーユーザーは任意のユーザーのデバイスを削除できます（ユーザーデバイスの管理権限が必要です）。

モバイルデバイスからアクセスできる Zabbix の領域を制限することもできます。そのためには、アプリに必要な権限だけを付与するユーザーロールをユーザーに割り当てます。最小限必要な権限は次のとおりです。

セクション	設定
UI 要素へのアクセス	すべてのチェックボックスの選択を解除します。
サービスへのアクセス	サービスへの読み書きアクセスと サービスへの読み取り専用アクセスを None に設定します。
モジュールへのアクセス	すべてのチェックボックスの選択を解除します。
API へのアクセス	有効チェックボックスの選択を解除します。
アクションへのアクセス	すべてのチェックボックスの選択を解除します。 ユーザーがアプリで障害を更新できるようにするには、必要なアクションも選択します。 - 障害コメントの追加 - 深刻度の変更 - 障害の承認 - 障害の抑制 - 障害のクローズ - 自分のメディアの作成と編集 ユーザーがアプリで表示できる障害は、ホストへのアクセスにも依存することに注意してください。
デバイス	有効チェックボックスを選択します。 ユーザーが自分のモバイルデバイスを追加および削除できるようにするには、自分のデバイスの管理チェックボックスも選択します。それ以外の場合は、スーパーユーザーがユーザー > デバイス でデバイスを管理する必要があります。

Zabbix ブリッジアダプターの保護　Zabbix ブリッジアダプターは、データ（状態ファイルと暗号化キー）を、Zabbix ブリッジアダプターの設定ファイルの`DataDir`パラメータで指定されたディレクトリに保存します。ディレクトリのパーミッションが 0700 であり、その中に保存されているすべてのファイルのパーミッションが 0600 であることを確認してください。

Zabbix ブリッジアダプターは、モバイルデバイスからのリクエストを Zabbix API に渡します。Zabbix ブリッジアダプター経由で呼び出せる API メソッドを制限するには、`AllowedAPIMethods`パラメータを使用します。たとえば、モバイルアプリに必要な最小限のメソッドのみを許可します。モバイルアプリの一部の機能を使用させたくない場合は、さらに少ないメソッドだけを許可することもできます（たとえば、モバイルアプリのユーザーが障害を更新できないようにする場合は、`event.acknowledge` を許可しないでください）。

開発者センター

開発者センター

プラグイン、モジュール、ウィジェットに関するガイダンスを活用して、Zabbix 向けのカスタム拡張機能の開発を始めましょう。

Zabbix で構築する

プラグイン　機能を拡張したり外部システムと統合したりするための、Zabbix プラグインの開発および管理方法の概要です。

Zabbix 用 **Python** ライブラリ　公式の `zabbix_utils` ライブラリを使用して Zabbix API と連携し、統合を構築するためのリソースです。

Web インターフェースモジュール　特定のユースケースに合わせて Zabbix の Web インターフェースを拡張または変更するカスタムモジュールを構築するためのガイドとリファレンス。

ウィジェット　ウィジェットの構造とロジックの内訳、およびニーズに合わせてカスタムダッシュボード要素を作成するための手順です。

プラグインの作成

このページは、Zabbix エージェント 2 用の独自プラグイン（独立したデータコレクター）の開発に関心がある方を対象としています。

カスタムプラグインを使用すると、提供されている組み込みプラグインおよびロード可能なプラグインの機能を超えて、Zabbix エージェント 2 の機能を拡張できます。

各プラグインは、構造を定義し、1 つ以上のインターフェース（*Exporter*、*Configurator*、*Runner*）を実装する *Go* パッケージです。
詳細については、プラグインインターフェースおよび接続図を参照してください。

このガイドでは、カスタムのロード可能なプラグインを作成する方法について説明します。

追加のガイダンスについては、以下のリポジトリも参照してください。

- サンプルプラグイン
- 既存のロード可能なプラグイン

作成するもの　これは、**MyIP** という名前のシンプルなロード可能プラグインを作成するためのステップバイステップのチュートリアルです。
このプラグインは、Zabbix エージェントホストの外部 IP アドレスを返す単一のアイテムキー（`myip`）を実装します。

ステップ **1:** セットアップ

1. プラグインは標準的な Go モジュールです。
   プラグインの依存関係を追跡するため、まずプラグインディレクトリで `go.mod` ファイルを初期化します:

```
cd path/to/plugins/myip # プラグインディレクトリに移動
go mod init myip
```

2. 必須の依存関係である Zabbix Go SDK (`golang.zabbix.com/sdk`) をインストールします:

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

`$LATEST_COMMIT_HASH` は、適切なリリースブランチにある `golang.zabbix.com/sdk` のリポジトリの最新の HEAD コミットハッシュに置き換えてください。
例えば次のようになります:

```
go get golang.zabbix.com/sdk@af85407
```

なお、`golang.zabbix.com/sdk` のバージョニングは現在サポートされていませんが、将来的に変更される可能性があります。

追加の依存関係は、必要に応じて `go get` を使用してインストールできます。

3. プラグインのソースコード用に空の `main.go` ファイルを作成します:

```
touch main.go
```

これで初期セットアップは完了し、プラグインの開発を開始できます。

ステップ **2:** プラグインの構造　前のステップでインストールした `golang.zabbix.com/sdk` モジュールは、プラグイン開発に必要なフレームワークを提供し、すべてのプラグインが一貫した構造を持つようにします。

1. 基本的な実行フローを設定します。

まず、プラグインのメイン実行フローを定義します。`main.go` に次のコードを追加します。

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

これにより、プラグインの基本的な実行フローが確立されます。`run` 関数には、後でプラグインの中核となるロジックを追加します。

2. プラグインインターフェースを確認します。

Zabbix エージェント 2 のプラグインは、`golang.zabbix.com/sdk/plugin` パッケージのインターフェースを実装する構造体として表現する必要があります。

- *Accessor* - プラグイン名の設定やアイテムキーのタイムアウト処理など、すべてのプラグインで実装する必要がある基本的なメソッドを定義します。
- 次の機能プラグインインターフェースのうち、1 つ以上:

- *Exporter* - ポーリングを実行し、値（または複数の値）、何も返さない結果、またはエラーを返します。多くの場合、*Collector* インターフェースと組み合わせて使用します。
- *Collector* - データの定期的な収集を管理します。
- *Runner* - プラグインの起動およびシャットダウン手順を定義します。
- *Watcher* - エージェント内部のスケジューラーを介さずに、独立したメトリクスのポーリングを可能にします。トラップベースまたはイベント駆動型の監視に役立ちます。
- *Configurator* - プラグインが設定を読み取り、適用する方法を定義します。

これらのインターフェースは自分で実装することも、Zabbix Go SDK が提供するデフォルトのインターフェースを使用し、必要に応じて変更することもできます。このチュートリアルでは、デフォルトの実装を使用します。

3. プラグイン構造体を作成します。

次に、*plugin* パッケージをインポートし、`plugin.Base` 構造体を埋め込んだ `myIP` 構造体を作成します。

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

現在、`myIP` 構造体は Accessor インターフェースを満たしています。機能プラグインインターフェースの 1 つである Exporter を実装するメソッドは、このチュートリアルの後半で追加します。

ステップ **3:** アイテムキーを定義する　プラグインがデータを収集して Zabbix サーバーまたはプロキシに提供するには、アイテムキーが必要です。

1. エラーハンドリングのために *errs* パッケージをインポートします。

```
import "golang.zabbix.com/sdk/errs"
```

2. `run()` 関数内で `plugin.RegisterMetrics()` 関数を使用してアイテムキーを登録します。

```
func run() error {
    p := &myIP{}

    // `myip` アイテムキーを登録します。
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // プラグイン名
        "myip",                             // アイテムキー名
        "Returns the host's IP address.", // アイテムキーの説明
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

複数のアイテムキーを登録するには、各メトリクスごとに *metric name* と *description* のパラメータを繰り返します。
例:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

ステップ **4:** ハンドラーの設定　ハンドラーは、エージェントとプラグイン間の通信を仲介します。

1. *container* パッケージをインポートします。

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. `run()` 関数内に、ハンドラーを作成して設定するコードを追加します。

```
func run() error {
    p := &myIP{}

    // `myip` アイテムキーを登録します。
    err := plugin.RegisterMetrics(
        p,
```

```
        "MyIP",                                 // プラグイン名
        "myip",                                 // アイテムキー名
        " ホストの IP アドレスを返します。", // アイテムキーの説明
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // 新しいハンドラーを作成します。
    h, err := container.NewHandler("MyIP") // プラグイン名
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // プラグインからエージェントへログを転送するようにロギングを設定します。
    // p.Logger.Infof、p.Logger.Debugf などで利用できます。
    p.Logger = h

    // プラグインの実行を開始します。
    // エージェントから終了要求を受信するまでブロックします。
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

ステップ **5:** データ収集を実装する　データ収集は Exporter インターフェースを介して行われ、このインターフェースは Export メソッドを定義します。

```
func Export(
  key string,              // 収集するアイテムキー。
  params []string,         // アイテムキーに渡される引数 (`myip[arg1, arg2]`)。
  context ContextProvider // アイテムキーデータ収集に関するメタデータ。
) (any, error)
```

1. HTTP リクエストとレスポンス読み取りに必要なパッケージをインポートします。

```
import (
    "io"
    "net/http"
)
```

2. myIP 構造体の Export メソッドを実装します。

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // プラグインは、`key`パラメータに基づいて異なるデータ収集ロジックを使用できます。
    // この実装では、指定された`key`がサポートされていることのみを確認します。
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // このログは agent 2 のログに転送されます。
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // データを収集して返します。
```

```
    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

ステップ **6:** プラグインのビルドと設定

1. プラグインをビルドするには、次を実行します:

```
go mod tidy
go build
```

これにより、現在のディレクトリに実行可能ファイル `myip` が作成されます。

2. Zabbix エージェント 2 がプラグインを使用するように設定します:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

`$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` を、ステップ 5 で作成した `myip` のパスに置き換えてください。

設定パラメータ名のプラグイン名（このチュートリアルでは *MyIP*）は、*plugin.RegisterMetrics()* 関数で定義されたプラグイン名と一致している必要があります。

3. プラグインとその `myip` アイテムをテストするには、次を実行します:

```
zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf -t myip
```

出力にはホストの外部 IP アドレスが含まれ、次のような形式になります:

```
myip                                            [s|192.0.2.1]
```

これで、Zabbix エージェント 2 用のシンプルなロード可能プラグインが作成されました。おめでとうございます!

```
package main

import (
    "io"
    "net/http"

    "golang.zabbix.com/sdk/errs"
    "golang.zabbix.com/sdk/plugin"
    "golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
    plugin.Base
}

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}
```

```
func run() error {
    p := &myIP{}

    // `myip` アイテムキーを登録します。
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                              // プラグイン名
        "myip",                              // アイテムキー名
        " ホストの IP アドレスを返します。", // アイテムキーの説明
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // 新しいハンドラーを作成します。
    h, err := container.NewHandler("MyIP") // プラグイン名
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // プラグインからエージェントへログを転送するようにロギングを設定します。
    // p.Logger.Infof、p.Logger.Debugf などで利用できます。
    p.Logger = h

    // プラグインの実行を開始します。
    // エージェントから終了要求を受信するまでブロックします。
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // プラグインは、`key` パラメータに基づいて異なるデータ収集ロジックを使用できます。
    // この実装では、指定された `key` がサポートされていることのみを確認します。
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // ログはエージェント 2 のログに転送されます。
    p.Infof(
        "%q キーを %d 個のパラメータで処理するリクエストを受信しました",
        key,
        len(params),
    )

    // データを収集して返します。

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
```

```
    }

    return string(body), nil
}
```

完全なソースコード

プラグインインターフェース

このセクションでは、利用可能なプラグインインターフェースについて説明します。

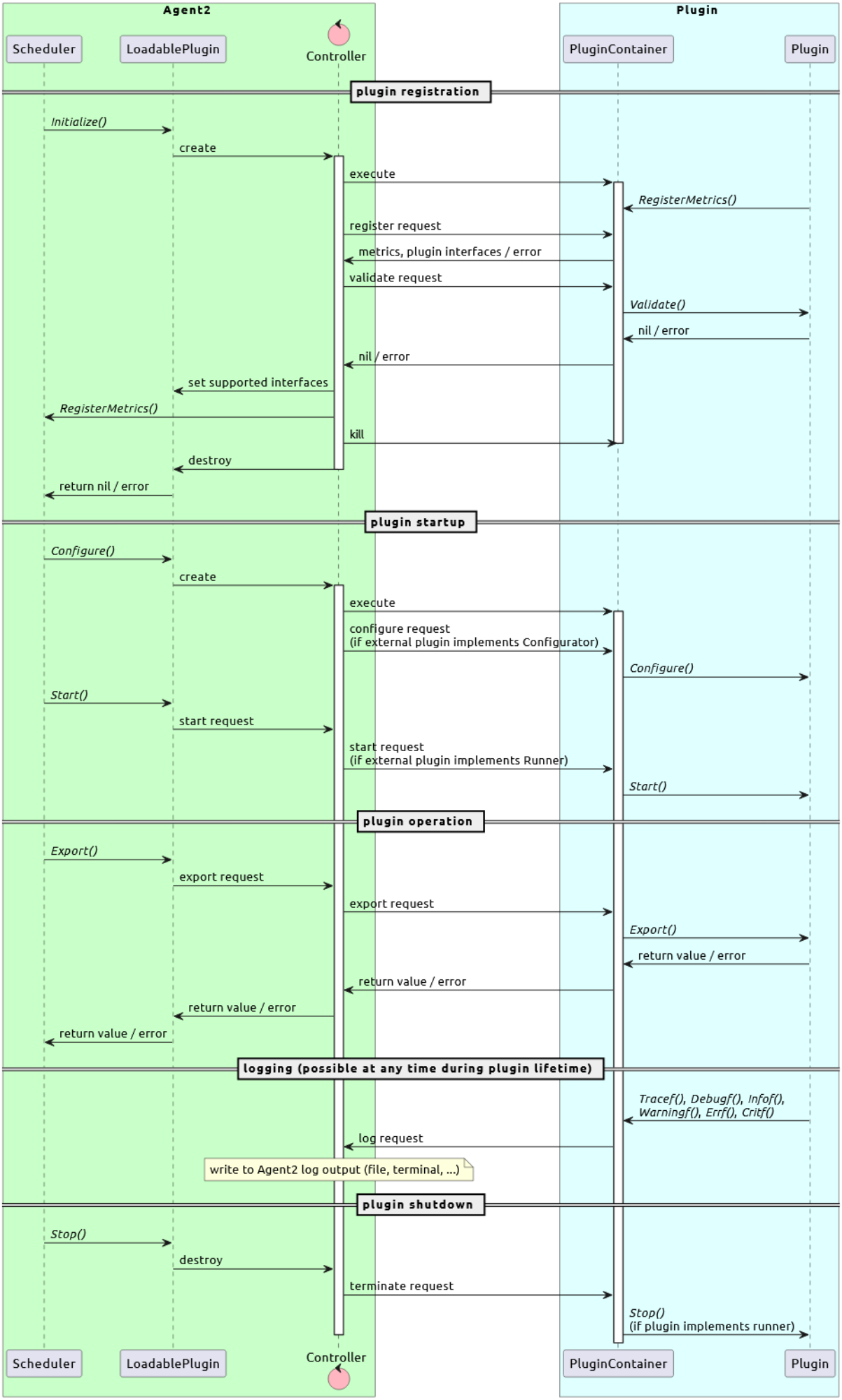
plugin.Exporter　*Exporter* は、ポーリングを実行して値 (複数可)、何も返さない、またはエラーを返す最もシンプルなインターフェースです。これは、事前に解析されたアイテムキー、パラメーター、およびコンテキストを受け入れます。他のすべてのプラグインインターフェースへのアクセスは排他的であり、プラグインがすでにタスクを実行している場合はメソッドを呼び出すことはできません。また、プラグインごとに最大 100 回の同時 *Export()* 呼び出しの制限がありますが、これは各プラグインの要件に応じて減らすことができます。

plugin.Configurator　*Configurator* インターフェースは、Zabbix エージェント 2 の設定ファイルからプラグイン設定パラメーターを提供します。

plugin.Runner　*Runner* インターフェースは、プラグインの開始時 (アクティブ化) に初期化を実行し、プラグインの停止時 (非アクティブ化) に非初期化を実行する手段を提供します。たとえば、プラグインは Runner インターフェースを実装することで、バックグラウンドの *goroutine* を開始/停止できます。

接続図　Zabbix エージェント 2 は、Linux では UNIX ソケット、Windows では Named Pipes を使用して、プラグインと双方向に接続します。

以下の接続図は、Zabbix エージェント 2 とロード可能なプラグイン間の通信プロセスおよびメトリクス収集プロセスを示しています。


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

Zabbix 用 Python ライブラリ

zabbix_utilsは、Zabbix の公式 Python ライブラリです。

zabbix_utils を使用すると、以下のことができます。

- Zabbix API を使用して Zabbix オブジェクトを管理する（ホストの作成、アイテムの更新、イベントの取得など）。
- Zabbix エージェントからデータを収集する（Zabbix get と同様）。
- Zabbix サーバーまたはプロキシにデータを送信する（Zabbix sender と同様）。

このライブラリは、同期および非同期のスクリプト実行の両方をサポートしており、シンプルなワークフローから、大規模な並列処理（大量データ収集、一括エクスポート、アラート処理、効率的なシステムスケーリングなど）まで対応できます。

このライブラリは、サポートされている Zabbix サーバーバージョンおよび最新の Python リリースと互換性があります。

まずライブラリをインストールし、クイックスタートガイドに従ってください。

インストール

zabbix_utilsは、以下のいずれかの方法でインストールできます。

要件:

- Zabbix 6.0 以降（6.0、7.0、7.2、7.4 でテスト済み）
- Python 3.8 以降（3.8～3.13 でテスト済み）
- aiohttp 非同期 HTTP フレームワーク（非同期モードのみ必須）

Python Package Index (PyPI)

これはほとんどの環境で最も一般的な方法です:

```
pip install zabbix_utils
pip install zabbix_utils[async] # 非同期サポートのみ
```

Zabbix リポジトリ

依存関係をシステムのパッケージマネージャーで管理したい場合は、この方法を使用します。

Linux ディストリビューション用のZabbix パッケージをダウンロードし、次のコマンドを実行します。

RHEL およびその派生ディストリビューションの場合:

```
dnf install python3-zabbix-utils
dnf install epel-release # 非同期サポートのみ
dnf install python3-aiohttp # 非同期サポートのみ
```

Debian/Ubuntu およびその派生ディストリビューションの場合:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # 非同期サポートのみ
```

ソース (GitHub)

最新の開発バージョンを利用したい場合は、この方法を使用してください:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # 非同期サポートのみ
python3 setup.py install
```

非同期モードに必要な依存関係をインストールするには、上記のいずれかの方法も使用できます。

クイックスタートガイド

zabbix_utils をインストールした後、スクリプトで使用できます。

このクイックスタートガイドでは、以下の方法を説明します。

- Zabbix エージェントからホスト名を取得する。
- Zabbix API を使用してホストとトラッパーアイテムを作成する。
- アイテムに値を送信する。

このガイドでは、スクリプトを段階的に紹介し、各部分を説明します。完全なスクリプトはページの最後に記載しています。

また、このガイドでは Zabbix サーバー、エージェント、API がローカルで動作していることを前提としています。

Zabbix エージェントからホスト名を取得する

まず、Zabbix エージェントが稼働しているシステムのホスト名を取得します。このホスト名は、Zabbix API を使用してホストを作成する際に必要となります。

1. zabbix_utils から `Getter` クラスをインポートします。このクラスはZabbix getのように動作し、Zabbix エージェントからデータをリクエストできます。

2. 127.0.0.1 のポート `10050` でローカルの Zabbix エージェントに接続する `Getter` インスタンスを作成します。

3. `Getter` インスタンスの `get()` メソッドを呼び出して、Zabbix エージェントからホスト名をリクエストします。

- `get()` メソッドはアイテムキーをパラメータとして受け取り、そのアイテムを Zabbix エージェントにリクエストします。
- `get()` メソッドはオブジェクトを返し、ホスト名はそのオブジェクトの `value` 属性に格納されます。
- `system.hostname`は、ホスト名を返す Zabbix エージェントの組み込みキーです。

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Zabbix API への接続とログイン

次に、Zabbix API に接続します。これにより、スクリプトでホストやアイテムなどの Zabbix オブジェクトを管理できるようになります。

1. zabbix_utils から `ZabbixAPI` クラスをインポートします。

2. ZabbixAPI インスタンスを作成し、Zabbix Web インターフェースの URL を指定します。

3. ZabbixAPI インスタンスの `login()` メソッドを呼び出し、ユーザー名とパスワードを指定します。アカウントには Zabbix API にアクセスするための権限が必要です。

4. API 操作後は、`logout()` メソッドを呼び出してセッションを閉じます。

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### ここに API 操作を記述

api.logout()
```

Zabbix でホストを作成する

Zabbix エージェントからホスト名を取得し、Zabbix API に接続できたら、新しいホストを Zabbix に作成できます。

1. ZabbixAPI インスタンスで`host.create()` API メソッドを呼び出し、ホストの詳細を指定します。

- `host` - Zabbix エージェントから取得したホスト名を含む `hostname` 変数を設定します。
- `interfaces` - ホスト上で実行されている Zabbix エージェントへの接続情報を含みます。
- `groups` - ホストが所属する少なくとも 1 つのホストグループを含みます。

2. Zabbix API は新しく作成されたホストの ID を返すため、この ID を後で使用するために `host_id` 変数に保存します。

3. ホストが作成されたことを確認するメッセージを出力します。

```
api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
```

```
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")
```

Zabbix で新しいアイテムを作成する

ホストを作成した後、そのホストにアイテムを追加できます。

1. 作成するアイテムの一意なキーを定義します。

2. ZabbixAPI インスタンスのitem.create()メソッドを呼び出し、アイテムの詳細を指定します。

- hostid - 直前に作成したホストの ID を含む host_id 変数を設定します。
- name - アイテムの名前。
- key_ - 直前に定義した item_key 変数を設定します。
- type - 2（トラッパーアイテム）を設定します。次の手順でスクリプトから送信される値を受信するために必要です。
- value_type - 3（数値（整数））を設定します。このアイテムが格納するデータの型です。

3. Zabbix API は新しく作成されたアイテムの ID を返すため、この ID を item_id 変数に保存して後で使用します。

4. アイテムが作成されたことを確認するメッセージを出力します。

```
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3,
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")
```

ホストに値を送信する

ホストとアイテムを作成したので、データを送信できます。

1. time クラスをインポートし、データを送信する前に数秒待ちます。これにより、Zabbix が新しいホストとアイテムを完全に処理したことを確認できます。

2. zabbix_utils から Sender クラスをインポートします。このクラスはZabbix senderのように動作し、スクリプトから Zabbix にデータを送信できます。

3. 127.0.0.1 のローカル Zabbix サーバーにポート 10051 で接続する Sender インスタンスを作成します。

4. Sender インスタンスの send_value() メソッドを呼び出して、ホストのアイテムに値を送信し、次の詳細を指定します:

- hostname - Zabbix エージェントから取得したホスト名を含む hostname 変数を設定します。
- item_key - スクリプトの前の部分で定義した item_key 変数を設定します。
- 1 - 送信する値。

4. 値が送信されたことを確認するメッセージを出力します。

```
import time
time.sleep(10)

from zabbix_utils import Sender

sender = Sender(server='127.0.0.1', port=10051)

response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {response}")
```

このコードが正常に実行されると、Zabbix が値を受信したことを示す成功メッセージが表示されます:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

また、Zabbix の Web インターフェース（監視 > 最新データ）で値を確認できます。

完全なスクリプト

以下は、ホスト名の取得、Zabbix API を使用したホストとアイテムの作成、Zabbix へのデータ送信のすべてのステップを組み合わせた完全なスクリプトです。

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Zabbix エージェントからホスト名を取得
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Zabbix API に接続してログイン
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Zabbix にホストを作成
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")

### Zabbix にアイテムを作成
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")

### API からログアウト
api.logout()

### Zabbix が新しいホストを処理するまで待機
time.sleep(10)

### ホストに値を送信
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)
```

```
print(f"Sender response: {response}")
```

Zabbix API の使用

zabbix_utilsを使用すると、Zabbix APIを利用して、ホストの作成、アイテムの更新、イベントの取得など、Zabbix オブジェクトを管理できます。

API リクエストは同期モードまたは非同期モードで実行できます。

- 同期モードでは、Python スクリプトがリクエストを送信し、応答を待ってから次に進みます。これは、単純で順次的かつ予測可能な操作に適しています。
- 非同期モードでは、スクリプトは各応答を待たずにリクエストを送信し、他の操作を並行して進めることができます。これは、遅いリクエストや大量のデータを扱う場合により効率的です。

このページの例は同期モードに焦点を当てていますが、非同期モードも同様のパターンに従います。追加の例はGitHub の zabbix_utilsリポジトリにあります。

インポート

Zabbix API を操作するために zabbix_utils を使用するには、スクリプト内で ZabbixAPI クラスをインポートします:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> すでにコミュニティの Python ライブラリを使用している場合は、そのインポートを zabbix_utils の ZabbixAPI に置き換えることができます。

ログイン

API リクエストを行う前に、ZabbixAPI インスタンスを作成し、Zabbix API にログインする必要があります。

資格情報の管理方法に最適な方法を選択してください。ユーザー名とパスワード、またはAPI トークンを使用できます。

初期化時

ZabbixAPI インスタンスを作成する際に、Zabbix Web インターフェースの URL と認証情報を一度に指定できます。

```
### ユーザー名とパスワード:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### API トークン:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

接続パラメータをまとめて管理したい場合は、Python の辞書として渡し、キーワード引数のアンパック構文を使うこともできます。

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

login() の使用

まず、Zabbix Web インターフェースの URL だけで ZabbixAPI インスタンスを作成し、後で認証情報を使って login() メソッドを使用できます。

```
### ユーザー名とパスワード:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API トークン:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

環境変数の使用

まず、認証情報を環境変数として保存できます。例えば、ターミナルを使用して以下のようにします。

```
### ユーザー名とパスワード:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### トークン:
export ZABBIX_TOKEN="your_api_token"
```

その後、同じターミナルセッションからスクリプトを起動すれば、スクリプト内の ZabbixAPI インスタンスには Zabbix Web インターフェースの URL だけが必要です。

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

URL も環境変数に保存することができます。

```
### ユーザー名とパスワード:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
export ZABBIX_URL="https://127.0.0.1/zabbix"

### トークン:
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

URL と認証情報がすべて環境変数として設定されている場合、スクリプト内の ZabbixAPI インスタンスには引数が全く必要ありません。

```
api = ZabbixAPI()
```

ログアウト

ユーザー名とパスワードでログインした場合、API 操作が完了したら logout() メソッドを呼び出してください。

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### ここに API アクションを記述

api.logout()
```

Note:
API トークンを使用している場合は、logout() メソッドの呼び出しは不要です。

API リクエスト

ログイン後、Zabbix API のメソッドリファレンスに記載されているメソッドを呼び出すことで、任意の API リクエストを実行できます。

API メソッドは、以下のフォーマットで呼び出します。

```
api_instance.zabbix_object.method(parameters)
```

Note:
一部の Zabbix API メソッドやオブジェクト名には、Python の予約語 (例：import) が使われています。Python でエラーを回避するため、メソッドやオブジェクト名の末尾にアンダースコアを付けてください (例：api.configuration.import_)。

例えば、host.getでホストの一覧を取得するには、以下のようにします。

```
### 1. zabbix_utils から ZabbixAPI をインポート
from zabbix_utils import ZabbixAPI

### 2. ZabbixAPI インスタンスを作成し、ログイン
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

```
api.login(user="Admin", password="zabbix")

### 3. フィルタに一致するホストを取得
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. 取得したホストをループし、各ホストの詳細を表示
for host in hosts:
    print(host)

### 5. API からログアウトし、セッションを閉じる
api.logout()
```

非同期モード

非同期モードでは、スクリプトが各 API リクエストの完了を待たずに API リクエストを送信できます。多くの API リクエストを実行する必要がある場合や、一部のリクエストに時間がかかる場合に、スクリプトの効率を向上させることができます。

非同期モードを使用する場合、同期モードと比較して重要な違いがあります。

- Python の `asyncio` モジュールをインポートします（事前に必要な依存関係をインストールする必要があります）。
- ZabbixAPI の代わりに `AsyncZabbixAPI` をインポートします。
- コードを `async` 関数内に記述します。
- `login()` や `logout()` を含む API メソッドの呼び出し時に `await` を使用します。

> **Attention:**
> AsyncZabbixAPI インスタンスを作成する際、初期化時にログインすることはできません。

例えば、非同期モードで`host.get`を使用してホストのリストを表示するには、次のようにします。

```
### 1. 非同期モード用に asyncio と、zabbix_utils から AsyncZabbixAPI をインポートします:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. すべての API 操作を実行するメインの async 関数を定義します:
async def main():

    # 3. AsyncZabbixAPI インスタンスを作成し、ログインします（await が必要）:
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. フィルターに一致するホストを取得します（await が必要）:
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. 返されたホストを繰り返し処理し、各ホストの詳細を表示します:
    for host in hosts:
        print(host)

    # 6. API からログアウトしてセッションを閉じます（await が必要）:
    await api.logout()
```

```
### 7. asyncio のイベントループを使用して async main() 関数を実行します:
asyncio.run(main())
```

例

以下の例は、zabbix_utils ライブラリを使用した一般的な Zabbix API タスクを示しています。

追加の例は、zabbix_utils GitHub リポジトリの examples ディレクトリにあります。

Zabbix エージェントからデータを収集する

概要

zabbix_utilsを使用すると、Zabbix エージェントからデータを収集できます（Zabbix getと同様）。

データは同期モードまたは非同期モードで収集できます。

- 同期モードでは、Python スクリプトがデータを要求し、続行する前にデータを待機します。これは、単純で順次的かつ予測可能な操作に適しています。
- 非同期モードでは、スクリプトは各応答を待たずにデータを要求し、他の操作を並行して実行できます。これは、遅いリクエストや大量のデータバッチに対してより効率的です。

このページの例は同期モードに焦点を当てていますが、非同期モードも同様のパターンに従います。追加の例は、zabbix_utils GitHub リポジトリで利用できます。

インポート

zabbix_utils を使用してアイテム値を収集するには、Python スクリプトで Getter クラスをインポートします。

```
from zabbix_utils import Getter
```

リクエストデータ

アイテムの値を要求するには、次の手順を実行します。

1. Zabbix エージェントの IP アドレスとポートを指定して、Getter インスタンスを作成します。
2. 取得したいアイテムのキーを指定して、Getter インスタンスの get() メソッドを呼び出します。

たとえば、system.uname アイテムのデータを要求するには、次のようにします。

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

デフォルト以外の IP の使用

スクリプトを実行しているサーバーに複数の IP アドレスがある場合、Zabbix エージェントへの接続時に Getter が使用する source_ip を指定できます。

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

タイムアウトの使用

Getter に応答 timeout を設定して、スクリプトが Zabbix エージェントからの応答をどれだけ待ってから諦めるかを制御できます。

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

暗号化の使用

Getter には組み込みの暗号化サポートはありませんが、サードパーティーのライブラリを使用してラッパーを作成することで提供できます。

```
def psk_wrapper(sock, tls):
    # ...
    # ソケット用の TLS PSK ラッパーの実装
    # ...

agent = Getter(
    host='192.0.2.1',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

レスポンス

Zabbix エージェントからのレスポンスはライブラリによって処理され、AgentResponse オブジェクトとして返されます。

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

非同期モード

非同期モードを使用すると、スクリプトは各値の到着を待たずに値を収集できます。そのため、多数の値を収集する必要がある場合や、一部の値の収集に時間がかかる場合に、スクリプトをより効率的にできます。

非同期モードを使用する場合、同期モードと比べて重要な違いがあります。

- Python の asyncio モジュールをインポートします（先に必要な依存関係をインストールする必要があります）。
- Getter ではなく AsyncGetter をインポートします。
- コードは async 関数内に記述します。
- get() メソッドを呼び出すときは await を使用します。

たとえば、非同期モードを使用して単一の値を収集するには、次のようにします。

```
### 1. 非同期モード用に asyncio を、zabbix_utils から AsyncGetter をインポートします:
import asyncio
from zabbix_utils import AsyncGetter

### 2. すべてのデータ要求を実行するメインの async 関数を定義します:
async def main():
    agent = AsyncGetter(host='192.0.2.1', port=10050)

    # 3. Zabbix エージェントから system.uname の値を取得します（await が必要です）:
    response = await agent.get('system.uname')

    # 4. Zabbix エージェントが返した値を表示します:
    print(response.value)

### 5. asyncio のイベントループを使用して async の main() 関数を実行します:
asyncio.run(main())
```

Zabbix サーバーまたはプロキシへのデータ送信

zabbix_utilsを使用すると、Zabbix senderと同様に、Zabbix サーバーまたはプロキシのトラッパーアイテムにアイテムの値を送信できます。

単一の値、複数の値、さらには複数の Zabbix クラスタを対象に送信することも可能です。

データは同期モードまたは非同期モードで送信できます:

- 同期モードでは、Python スクリプトは値を送信し、応答を待ってから処理を続行します。これは、シンプルで順次的かつ予測可能な操作に適しています。
- 非同期モードでは、スクリプトは各応答を待たずに値を送信し、他の操作を並行して進めることができます。これは、遅いリクエストや大量のデータバッチに対してより効率的です。

このページの例は同期モードに焦点を当てていますが、非同期モードも同様のパターンに従います。追加の例はGitHub リポジトリにもあります。

インポート

zabbix_utils を使用してアイテム値を送信するには、スクリプトで Sender クラスをインポートします:

```
from zabbix_utils import Sender
```

複数の値を送信する場合は、ItemValue クラスもインポートできます:

```
from zabbix_utils import Sender, ItemValue
```

単一の値を送信する

アイテムの値を送信するには:

1. Sender インスタンスを作成し、Zabbix サーバーまたはプロキシの IP アドレスとポートを指定します。
2. Sender インスタンスで send_value() メソッドを次の形式で呼び出します:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

例えば、Linux server ホストの service.status トラッパーアイテムに 1 を送信するには:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

デフォルト以外の IP の使用

スクリプトを実行しているサーバーに複数の IP アドレスがある場合、Zabbix サーバーまたはプロキシに値を送信する際に Sender が使用する source_ip を指定できます。

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

タイムアウトの使用

Sender の応答 timeout を設定することで、スクリプトが Zabbix サーバーまたはプロキシからの応答を待つ最大時間を制御できます。

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

エージェント設定ファイルの使用

zabbix_utils に、ローカルの Zabbix エージェント またはエージェント 2 の設定ファイルからServer またはServerActive パラメータを読み取らせることができます。この場合、Sender インスタンスの作成時に接続パラメータを指定する必要はありません。

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
ServerActive に複数のサーバーインスタンスを含む 1 つ以上の Zabbix クラスターが設定されている場合、Sender は各クラスターで最初に利用可能なサーバーにデータを送信します。ServerActive が設定されていない場合は、Server のアドレスがデフォルトポート（10051）とともに使用されます。

暗号化の使用

Sender には組み込みの暗号化サポートはありませんが、サードパーティのライブラリを使用してラッパーを作成することで提供できます。

```
def psk_wrapper(sock, tls):
    # ...
    # ソケットの TLS PSK ラッパーの実装
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

単一値のレスポンス

Zabbix サーバーまたはプロキシから返されたレスポンスはライブラリによって処理され、TrapperResponse オブジェクトとして返されます:

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

非同期モード

非同期モードでは、Python スクリプトが Zabbix サーバーやプロキシからの応答を待たずに値を送信できます。多くの値を送信する必要がある場合や、一部の値の送信に時間がかかる場合に、スクリプトをより効率的に実行できます。

非同期モードを使用する場合、同期モードと比較していくつか重要な違いがあります。

- Python の asyncio モジュールをインポートします（事前に必要な依存関係をインストールする必要があります）。
- Sender の代わりに AsyncSender をインポートします。
- コードは async 関数内に記述します。
- send_value() メソッドを呼び出す際は await を使用します。

例えば、非同期モードで単一の値を送信するには:

```
### 1. 非同期モード用に asyncio と、zabbix_utils から AsyncSender をインポート:
import asyncio
from zabbix_utils import AsyncSender

### 2. すべてのデータ送信操作（await が必要）を実行するメインの async 関数を定義:
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Zabbix サーバーまたはプロキシから返された応答を出力:
    print(response)

### 4. asyncio のイベントループを使って async main() 関数を実行:
asyncio.run(main())
```

複数の値を送信する

複数の値を送信するには:

1. send_value() メソッドと同じフォーマットで、ItemValue オブジェクトの配列を用意します。
2. Zabbix サーバーまたはプロキシの IP アドレスとポートを指定して、Sender インスタンスを作成します。
3. Sender インスタンスで send_value() の代わりに send() メソッドを呼び出し、送信する値のオブジェクト配列を指定します。

例えば、異なるホストに 5 つの値を送信するには:

```
items = [
    ItemValue('server-de', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-fr', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-uk', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-nl', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-pl', 'service.status', 'up', 1770887205, 100),
]

sender = Sender(server='127.0.0.1', port=10051)
response = sender.send(items)
```

カスタムチャンクサイズの使用

1 つのリクエストでトラッパーアイテムが受け入れられる値より多くの値を送信する必要がある場合は、それらをチャンクに分割できます。

デフォルトでは、チャンクサイズは 250 値です。Sender インスタンスを作成する際に chunk_size パラメータを設定することで、これを変更できます。

たとえば、5 つの値を 3 つのチャンク（2-2-1）で送信するには、chunk_size パラメータを 2 に設定します。

```
items = [
    ItemValue('server-de', 'service.status', 'up'),
    ItemValue('server-fr', 'service.status', 'up'),
    ItemValue('server-uk', 'service.status', 'up'),
    ItemValue('server-nl', 'service.status', 'up'),
    ItemValue('server-pl', 'service.status', 'up'),
]

sender = Sender(server='127.0.0.1', port=10051, chunk_size=2)
response = sender.send(items)
```

複数の Zabbix クラスタに値を送信する

複数の Zabbix クラスタに値を送信するには:

1. Zabbix クラスタの配列を用意します。クラスタに複数のノードがある場合、値は各クラスタの最初の利用可能なノードに送信されます。
2. Zabbix クラスタの配列を指定して Sender を作成します。
3. send_value()メソッドと同じ形式で、Sender インスタンスの send_value() メソッドを呼び出します。

例えば、各クラスタの最初の利用可能なノードに値を送信するには:

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]

sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

複数値のレスポンス

デフォルトでは、Sender はすべてのホストまたはクラスタに対して値を送信した集計結果を返します。

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

より詳細な情報が必要な場合は、response.details 属性を使用して、各クラスタおよび各チャンクの結果を確認できます。

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"Value sent successfully in {response.time}")
else:
    print(response.details)
    # {
```

```
    #     127.0.0.1:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000051",
    #             "chunk": 1
    #         }
    #     ],
    #     zabbix.example.local:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000057",
    #             "chunk": 1
    #         }
    #     ]
    # }

    for node, chunks in response.details.items():
        for resp in chunks:
            print(f"Processed {resp.processed} of {resp.total} at {node.address}:{node.port}")
            # Processed 1 of 1 at 127.0.0.1:10051
            # Processed 1 of 1 at zabbix.example.local:10051
```

デバッグログ

Zabbix API の利用時や、Zabbix エージェントからのデータ収集、Zabbix サーバーやプロキシへのデータ送信時に問題が発生した場合、デバッグログを有効にして詳細な情報を取得できます。

zabbix_utils ライブラリは、Python 標準の `logging` モジュールを使用しています。

ライブラリからの詳細なデバッグメッセージを表示するには:

1. スクリプトで Python の `logging` モジュールをインポートします。
2. ログ出力のフォーマットを設定し、ログレベルを DEBUG に設定します。

```
import logging
from zabbix_utils import Getter

logging.basicConfig(
    format=u'[%(asctime)s] %(levelname)s %(message)s',
    level=logging.DEBUG
)

agent = Getter(host='127.0.0.1', port=10050)
resp = agent.get('system.uname')

print(resp.value)
```

デバッグログを有効にすると、リクエストやレスポンスの詳細情報が出力されます。例:

```
[2023-10-01 12:00:01,587] DEBUG Content of the packet: b'ZBXD\x01\x0c\x00\x00\x00\x00\x00\x00\x00system.un
[2023-10-01 12:00:01,722] DEBUG Zabbix response header: b'ZBXD\x01C\x00\x00\x00C\x00\x00\x00'
[2023-10-01 12:00:01,723] DEBUG Zabbix response body: Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
[2023-10-01 12:00:01,724] DEBUG Response from [127.0.0.1:10050]: Linux test_server 5.15.0-3.60.5.1.el9uek.
Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
```

モジュール

PHP フロントエンドモジュールとは

- モジュールは、マニフェストファイルで定義された一意の ID、名前、説明、作成者、およびその他のフィールドを持つエンティティであり、PHP、Javascript、およびその他のファイルは、Zabbix フロントエンドインストールの *modules* ディレクトリ内の 1 つのディレクトリ (例: *zabbix/ui/modules*) に配置されます。
- モジュールは、正しい操作を保証するために、単純なルールに準拠する必要があります。
- モジュールは、管理者によってフロントエンドにインストール (解凍) され、有効にされる必要があります。

モジュールの用途

- カスタムフロントエンドセクションを介して新しい機能を追加する。
- カスタムダッシュボードウィジェットタイプを作成する (ウィジェットモジュールを参照)。
- 既存の機能をオーバーライドまたは拡張する。

モジュールが使用できない用途

- 新しい API メソッドの登録、または既存のメソッドの変更。

モジュールの動作

- アクションコードを実行する前に、各 HTTP リクエストで有効なモジュールが起動されます。
- モジュールは新しいアクションを登録するか、既存のアクションを再定義します。
- モジュールは新しいフロントエンドセクションを追加し、既存のセクションを削除または再定義します。
- モジュールは必要に応じて、onBeforeAction や onTerminate などのフロントエンドイベントにフックします。
- 要求されたアクションは、最終的にアクションコード (デフォルトのアクションまたはモジュール定義のアクションコード) を実行することによって実行されます。

次に進むべき道　実際にやってみるか、最初にガイドラインを読むかに関わらず、これらのページには独自のモジュールを作成するために必要な情報と手順が記載されています。

- 最初のモジュールを作成するためのステップバイステップのチュートリアル
- モジュールファイル構造
- ウィジェットモジュールの詳細
- 再利用するモジュールの例

モジュールファイルの構造

モジュールに関連するすべてのコードは、Zabbix フロントエンドインストールの **modules** ディレクトリ内の 1 つのディレクトリに保存されます (例: *zabbix/ui/modules*)。

```
example_module_directory/           (必須)
   manifest.json                    (必須)   メタデータとアクションの定義。
   Module.php                                 モジュールの初期化とイベント処理。
   actions/                                   アクションコントローラーファイル。
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
       data_export/
           ExportAsXml.php
           ExportAsExcel.php
   views/                                     ビューファイル。
       example.something.view.php
       example.something.delete.php
   assets/                                    ビューで使用する追加ファイル。manifest.json で指定する必要があります。
       js/                                    ビューで使用する JavaScript ファイル。
           example.something.view.js.php
       css/                                   ビューで使用する CSS ファイル。
           example.something.css
        image.png                             ビューで使用する画像。
        example.something.file                ビューで使用する任意のファイル。
```

モジュールファイルツリー

モジュールの作成　サンプルのモジュール作成手順は、次のステップで構成されます（利用可能な場合は、ファイル名またはフォルダ名をクリックすると、そのステップの詳細を表示できます）：

1. **zabbix/ui/modules/** 内に、モジュール用の新しいディレクトリを作成します。
2. モジュールのメタデータを含むmanifest.json ファイルを追加します。
3. *views* フォルダを作成し、モジュールのビューを定義します。
4. *actions* フォルダを作成し、モジュールのアクションを定義します。
5. Module.php（ダッシュボードウィジェットの場合は Widget.php）ファイルを作成し、初期化とイベント処理のルールを定義します。
6. JavaScript ファイル用の *assets* フォルダ（*assets/js* に配置）、CSS スタイル用のフォルダ（*assets/css* に配置）、またはその他の追加ファイルを作成します。
7. manifest.json に、必要なビュー、アクション、およびアセットファイルを指定していることを確認します。
8. Zabbix Web インターフェースでモジュールを登録 し、使用を開始します。

> **Note:**
> manifest.json ファイルを作成した時点で、モジュールを登録して有効化できます。モジュールが有効になると、Zabbix Web インターフェースを更新するだけで、モジュールファイルに加えた変更をすぐにプレビューできます。

manifest.json

任意のモジュールには manifest.json ファイルが必要です。このファイルはモジュールの主要ディレクトリに配置する必要があります（例: *zabbix/ui/modules/module_name/manifest.json*）。

最低限、manifest.json には次のフィールドを指定する必要があります。

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

manifest.json でサポートされるパラメータ（詳細はパラメータ名をクリックしてください）：

Parameter	Description	Required
manifest_version	モジュールのマニフェストバージョン。	Yes
id	一意のモジュール ID。	
name	管理セクションに表示されるモジュール名。	
namespace	モジュールクラス用の PHP 名前空間。	
version	モジュールのバージョン。	
type	モジュールのタイプ。ウィジェットの場合は *widget* に設定する必要があります。	ウィジェットの場合は Yes、それ以外は
widget	ウィジェット設定。ウィジェット専用です。	
actions	モジュールに登録するアクション。	
assets	含める CSS スタイルと JavaScript ファイル。	No
author	モジュールの作成者。	
config	カスタムモジュールオプションのデフォルト値。	
description	モジュールの説明。	
url	モジュールの説明へのリンク。	

manifest_version

モジュールのマニフェストバージョン。現在サポートされているバージョンは **2.0** です。

タイプ: Double

例:

```
"manifest_version": 2.0
```

id

モジュール ID。一意である必要があります。将来的に名前の競合が発生しないように、モジュールにはプレフィックス (作成者名、会社名など) を使用することをお勧めします。たとえば、レッスンのサンプルのモジュールで、モジュール名が”My module” の場合、ID は”example_my_module” になります。

タイプ: String

例:

```
"id": "example_my_module"
```

name

管理セクションに表示されるモジュール名。

タイプ: String

例:

```
"name": "My module"
```

namespace

モジュールクラスの PHP 名前空間。

タイプ: String

例:

```
"namespace": "ClockWidget"
```

version

モジュールのバージョン。バージョンは管理セクションに表示されます。

タイプ: String

例:

```
"version": "1.0"
```

type

モジュールのタイプ。ウィジェットの場合必須で、”widget” と等しくなければなりません。

タイプ: String

デフォルト: ”module”

例:

```
"type": "widget"
```

actions

モジュールに登録するアクション。各アクションの *class* オブジェクトキーの定義は必須ですが、他のアクションキーはオプションです。

タイプ: Object

*type*が *module* の場合にサポートされるオブジェクトキー:

- **write.your.action.name** (object) - アクション名は小文字 [a-z] で記述し、単語をドットで区切ります。サポートするキー:
  - **class** (string; 必須) - アクションクラス名。
  - **layout** (string) - アクションレイアウト。サポートする値: *layout.json*、*layout.htmlpage* (デフォルト)、*null*。
  - **view** (string) - アクションビュー。

例:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

*type*が *widget* の場合にサポートされるオブジェクトキー:

- **widget.{id}.view** (object) - ウィジェットビューのファイルとクラス名。**{id}** をウィジェットのid値に置き換えます (例: *widget.example_clock.view*)。サポートするキー:
  - **class** (string; 必須) - デフォルトの CControllerDashboardWidgetView クラスを拡張するウィジェットビューモードのアクションクラス名。クラスソースファイルは *actions* ディレクトリに配置する必要があります。

    - **view** (string) - ウィジェットビュー。*views* ディレクトリに配置する必要があります。ビューファイルがデフォルトで期待される *widget.view.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。
- **widget.{id}.edit** (object) - ウィジェット設定ビューのファイル名。**{id}** をウィジェットのid値に置き換えます (例: *widget.example_clock.edit*)。サポートするキー:
    - **class** (string; 必須) - ウィジェット設定ビューモードのアクションクラス名。クラスソースファイルは *actions* ディレクトリに配置する必要があります。
    - **view** (string) - ウィジェット設定ビュー。*views* ディレクトリに配置する必要があります。ビューファイルがデフォルトで期待される *widget.edit.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。

例:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

assets

含める CSS スタイルと JavaScript ファイル。

タイプ: Object

サポートするオブジェクトキー:

- **css** (array) - 含める CSS ファイル。ファイルは *assets/css* に配置する必要があります。
- **js** (array) - 含める JavaScript ファイル。ファイルは *assets/js* に配置する必要があります。

例:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

author

モジュールの作成者。作成者は管理セクションに表示されます。

タイプ: String

例:

```
"author": "John Smith"
```

config

モジュールオプションのデフォルト値。オブジェクトにはカスタムキーが含まれる場合があります。指定した場合、これらの値はモジュールの登録中にデータベースに書き込まれます。後で追加された新しい変数は、最初の呼び出し時に書き込まれます。その後、変数値はデータベース内でのみ直接変更できます。

タイプ: Object

例:

```
"config": {
    "username": "Admin",
    "password": "",
    "auth_url": "https://example.com/auth"
}
```

description

モジュールの説明。

タイプ: String

例:

```
"description": "This is a clock widget."
```

widget

ウィジェットの設定。*type*が *widget* に設定されている場合に使用されます。

タイプ: Object

サポートするオブジェクトキー*:*

- **name** (string) - ウィジェットリストおよびデフォルトのヘッダーとして使用されます。空の場合、モジュールの”name” パラメーターが使用されます。
- **size** (object) - デフォルトのウィジェットの寸法。サポートするキー:
  - *width* (integer) - デフォルトのウィジェットの幅。
  - *height* (integer) - デフォルトのウィジェットの高さ。
- **form_class** (string) - ウィジェットフィールドフォームを持つクラス。*includes* ディレクトリに配置する必要があります。クラスがデフォルトで期待する *WidgetForm.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。
- **js_class** (string) - デフォルトの CWidget クラスを拡張するウィジェットビューモードの JavaScript クラスの名前。クラスにはダッシュボードがロードされます。クラスのソースファイルは、assets/js ディレクトリに配置する必要があります。参照: assets。
- **use_time_selector** (boolean) - ウィジェットにダッシュボード時間セレクターが必要かどうかを決定します。サポートする値: *true*、*false* (デフォルト)。
- **refresh_rate** (integer) - ウィジェットのリフレッシュレート (秒単位) (デフォルト: 60)。

例:

```
"widget": {
    "name": "",
    "size": {
        "width": 12,
        "height": 5
    },
    "form_class": "WidgetForm",
    "js_class": "CWidget",
    "use_time_selector": false,
    "refresh_rate": 60
}
```

url

モジュールの説明へのリンクです。ウィジェットの場合、ウィジェットの追加またはウィジェットの編集ウィンドウでヘルプアイコン をクリックすると、このリンクが開きます。**url** が指定されていない場合、ヘルプアイコンをクリックすると一般的なダッシュボードウィジェットページが開きます。

タイプ: String

例:

```
"url": "http://example.com"
```

アクション

アクションは、モジュールの’ ビジネスロジック’ を担当します。アクションは通常、コントローラーとアクションビューで構成されます。

モジュールは次の操作を実行できます。

- Zabbix フロントエンドで既に定義されているアクションを呼び出す。
- デフォルトのアクションをカスタムアクションで上書きする。
- まったく新しいアクションを定義する。

デフォルトのアクション動作をカスタム動作で上書きするには、モジュール設定で同じ名前のアクションを定義します。アクションが呼び出されると、デフォルトの Zabbix アクションの代わりにモジュールアクションが実行されます。

アクションファイルは *actions* フォルダーに保存する必要があります。アクションはmanifest.jsonで指定する必要があります。

コントローラー

アクションコントローラーのワークフロー:

1. HTTP リクエストで渡されたすべてのパラメータが有効であることを確認します:

- コントローラーの *checkInput()* メソッドを呼び出します。
- CNewValidator.php で定義されたバリデーションルールを使用します。
- *validateInput()* メソッドを呼び出します。

2. ユーザー権限を確認します。
3. 渡されたパラメータに従ってデータを準備します: *checkInput()* が true を返した場合、Zabbix はコントローラーの *doAction()* メソッドを呼び出します。
4. ビュー用の **$data** 配列を準備します。レスポンスを **$data** 配列に格納するには、*CControllerResponseData* と *setResponse()* メソッドを使用します。

例:

```
/**
 * 入力パラメータを検証します。
 *
 * @return bool
 */
protected function checkInput(): bool {
    $ret = $this->validateInput([
        'status' => 'in '.implode(',', [HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED])
    ]);

    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }

    return $ret;
}

/**
 * ユーザー権限を確認します。
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * アクションを実行し、レスポンスオブジェクトを生成します。
 */
protected function do Action(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
利用可能なコントローラークラスの完全な一覧は、Zabbix のソースコードで確認できます。

ビュー

ビューファイルはコントローラーからデータを受け取り、その HTML 表示を準備します。

> **Note:**
> Web インターフェースモジュールのビューの定義は、モジュールがウィジェットでない限り任意です。
 ダッシュボードウィジェットには、少なくとも 2 つのビューが必要です。1 つは編集モード用、もう 1 つは表示モード用です（*views* ディレクトリに保存する必要があります）。

ビューでは、事前定義された Zabbix の HTML クラス（*/zabbix/ui/include/classes/html* にあるもの）を使用でき、新しい HTML クラスや CSS クラスを追加することもできます。新しいクラスは、モジュールの *assets* フォルダに保存する必要があります。

例:

```
...
(new CColHeader(_('Name')))
```

これにより、新しい列名 *Name* が追加され、上部のテーブル行が他の Zabbix ページと同様のスタイルになります。

アクションビュー

アクションビューを定義するためのリファレンスファイルです。

```
<?php declare(strict_types = 1);

/**
* @var CView $this
*/

$this->includeJsFile('example.something.view.js.php');

(new CWidget())
    ->setTitle(_('Something view'))
    ->addItem(new CDiv($data['name']))
    ->addItem(new CPartial('module.example.something.reusable', [
        'contacts' => $data['contacts']
    ])
    ->show();
```

アセット

assets フォルダーには、他のディレクトリに属さないファイルやサブフォルダーを含めることができます。次の目的で使用できます:

- JavaScript スタイル (*assets/js*内にある必要があります)
- CSS スタイル (*assets/css*内にある必要があります)
- 画像
- フォント
- その他、含める必要のあるもの

assets/js

assets/js ディレクトリは予約されており、JavaScript ファイルのみを含める必要があります。ウィジェットで使用するには、これらのファイルを*manifest.json*で指定します。

例:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css は予約されており、CSS スタイルファイルのみを含める必要があります。ウィジェットで使用するには、これらのファイルを*manifest.json*で指定します。

例:

```
"assets": {
    "css": ["mywidget.css"]
}
```

CSS スタイル

CSS ファイルでは、カスタム属性 `theme` を使用して、特定の Web インターフェーステーマ用に異なるスタイルを定義できます。

使用可能なテーマとその属性値:

- **Blue** - [theme='blue-theme']
- **Blue (classic)** - [theme='blue-classic-theme']
- **Dark** - [theme='dark-theme']
- **Dark blue** - [theme='dark-blue-theme']
- **Dark (classic)** - [theme='dark-classic-theme']
- **High-contrast light** - [theme='hc-light']
- **High-contrast dark** - [theme='hc-dark']

例:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

新しいモジュールの登録

このセクションでは、Zabbix フロントエンドに新しいモジュールを追加する方法について説明します。

前提条件　続行する前に、次のことを確認してください:

- モジュールは、Zabbix フロントエンドインストールの *modules* ディレクトリ内にあります (例: *zabbix/ui/modules*)。
- モジュールには、少なくともmanifest.jsonファイルの基本バージョンがあります。
- Zabbix の管理メニューセクションにアクセスできます (ユーザーの役割の super admin が必要です)。

Note:
フロントエンドは、互換性のないモジュールをインストールせず、認識もしません。

モジュールの追加　*Administration* > *General* > *Modules* ページを開き、*Scan directory* をクリックします。



一覧からモジュールを見つけて有効化します。

モジュールを有効化するには、*Disabled* のハイパーリンクをクリックします。モジュールの状態が *Enabled* に変わります。

モジュール名をクリックすると、作者、バージョン、または短い説明（manifest で定義されている場合）など、モジュールに関する追加情報を表示できます。

ウィジェットのプレビュー　ウィジェットモジュールは追加されると、すぐにダッシュボードのウィジェットリストに表示されます。

ダッシュボードを開き、編集モードに切り替えて、通常通りウィジェットをダッシュボードに追加できます。

ウィジェットに変更を加えた場合は、ダッシュボードをリフレッシュして、最新の更新内容でウィジェットがどのように表示されるかを確認してください。

ウィジェット

ウィジェットは、ダッシュボードに使用される Zabbix フロントエンドモジュールです。特に明記されていない限り、すべてのモジュールガイドラインはウィジェットにも適用されます。

ただし、ウィジェットはモジュールとは大きく異なります。ウィジェットを作成するには、次の操作を行います。

- manifest.json ファイルでタイプ”widget” を指定します (”type”: ”widget”)。
- 少なくとも 2 つのビューを含めます。1 つはウィジェットプレゼンテーションモード用、もう 1 つはウィジェット構成モード用です (example.widget.view.php と example.widget.edit.php)。
- ウィジェットプレゼンテーション (WidgetView.php) 用のコントローラー。
- デフォルトのウィジェットクラスを使用して拡張します。

設定

このページでは、カスタム設定フィールドを使用してウィジェット設定ビューを作成するために使用できるクラスについて説明します。ウィジェット設定ビューは、ユーザーがプレゼンテーションのウィジェットパラメーターを設定できるウィジェットの一部です。

Widget

プライマリウィジェットクラス、すべてのダッシュボードウィジェットの基本クラス *CWidget* を拡張します。デフォルトのウィジェットの動作をオーバーライドするために必要です。

Widget クラスは、ウィジェットのルートディレクトリ (*zabbix/ui/modules/my_custom_widget* など) に配置する必要があります。

Widget.php example

```
<?php

namespace Modules\MyCustomWidget;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const MY_CONSTANT = 0;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

WidgetForm

WidgetForm クラスは、デフォルトクラス *CWidgetForm* を拡張し、データベース内でウィジェット構成の保存構造を定義し、入力検証を処理するために必要な*CWidgetField*フィールドのセットを含みます。

WidgetForm クラスは *includes* ディレクトリに配置する必要があります。クラスに別の名前が付いている場合は、その名前を *manifest.json* ファイルの*widget/form_class*パラメーターで指定する必要があります。

includes/WidgetForm.php example

```
<?php

namespace Modules\MyCustomWidget\Includes;

use Modules\MyCustomWidget\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox,
    CWidgetFieldColor
};
```

```
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

CWidgetFormView クラスは、*WidgetForm* クラスで定義されたフィールドの表示ロジックを指定し、設定ビューでレンダリングされた際の外観と動作を決定するために必要です。

CWidgetFormView クラスは、次のメソッドをサポートしています。

- *addField()* - *CWidgetFieldView* クラスのインスタンスをパラメータとして受け取ります。各*CWidgetField* クラスには、ウィジェット設定ビューで使用するための対応する *CWidgetFieldView* クラスがあります。
- *addFieldset()* - フィールドを折りたたみ可能なコンテナにまとめる *CWidgetFieldsGroupView* クラスのインスタンスを受け取ります。
- *addFieldsGroup()* - フィールドをグループとして視覚的に（境界線付きで）まとめる *CWidgetFormFieldsetCollapsibleView* のインスタンスを受け取ります。
- *includeJsFile()* - ウィジェット設定ビューに JavaScript ファイルを追加できます。
- *addJavaScript()* - ウィジェット設定ビューの読み込み直後に実行されるインライン JavaScript を追加できます。

CWidgetFormView クラスは、*views* ディレクトリに配置する必要があります。

views/widget.edit.php の例

```
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
```

```
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

JavaScript クラスを使用して、ウィジェット設定ビューに動的な動作と対話性を追加できます。たとえば、*CWidgetFormView*クラスで定義されたカラーピッカーを初期化できます。

JavaScript クラスはフォームとともにロードする必要があるため、メソッド *includeJsFile()* および *addJavaScript()* を使用して*CWidgetFormView*クラスで参照する必要があります。

以下の例では、シングルトンクラスインスタンスがすぐに作成され、*window.my_custom_widget_form* という名前で保存されます。したがって、フォームを 2 回目に開くとインスタンスが再作成されます。

JavaScript クラスは *views* ディレクトリに配置する必要があります。

views/widget.edit.js.php example

```
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

    init({color_palette}) {
        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }
    }
};
```

CWidgetField

CWidgetField クラスは、すべてのフォームフィールドクラス（*CWidgetFieldCheckBox*、*CWidgetFieldTextArea*、*CWidgetFieldRadioButtonList* など）が継承する基底クラスです。
CWidgetField を拡張するクラスは、ウィジェット設定値の受け取り、保存、検証を担当します。

以下の *CWidgetField* クラスを使用できます。

CWidgetField class	Database field type	Description
CWidgetFieldCheckBox	int32	単一のチェックボックス。
CWidgetFieldCheckBoxList	array of int32	1 つの設定フィールドに含まれる複数のチェックボックス。
CWidgetFieldColor	string	色の選択フィールド。
CWidgetFieldDatePicker	string	日付の選択フィールド。
CWidgetFieldIntegerBox	int32	整数を入力するフィールド。最小値と最大値を設定できます。
CWidgetFieldLatLng	string	緯度、経度、地図のズームレベルをカンマ区切りで入力するテキストボックス。
CWidgetFieldMultiSelect	-	マルチセレクトフィールドの基底クラス。すべての CWidgetFieldMultiSelect* クラスが拡張します。
CWidgetFieldMultiSelectActionID		*Alerts > Actions* で定義されたアクションの一覧からアクションを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectGraphID		カスタムグラフを選択するマルチセレクトフィールド。

CWidgetField class	Database field type	Description
CWidgetFieldMultiSelectGraphPrototype	ID	カスタムグラフのプロトタイプを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectGroup	ID	ホストグループを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectHost	ID	ホストを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectHostInventory	array of int32	ホストインベントリフィールドを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectItem	ID	アイテムを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectItemPrototype	ID	アイテムのプロトタイプを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectMap	ID	マップを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectMediaType	ID	メディアタイプを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectOverrideHost	ID	ダッシュボードまたは別のウィジェットからホストのデータソースを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectService	ID	サービスを選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectSla	ID	SLA を選択するマルチセレクトフィールド。
CWidgetFieldMultiSelectUser	ID	ユーザーを選択するマルチセレクトフィールド。
CWidgetFieldNavTree	array of (multiple mixed)	*Map navigation tree* ウィジェットで使用するマップのナビゲーションツリー構造を保存します。
CWidgetFieldNumericBox	string	浮動小数点数を入力するフィールド。
CWidgetFieldPatternSelect	-	パターン選択フィールドの基底クラス。すべての `CWidgetPatternSelect*` クラスが拡張します。
CWidgetFieldPatternSelectHost	string	ホストを選択するマルチセレクトフィールド。ホスト名のパターンに対応します。
CWidgetFieldPatternSelectItem	string	アイテムを選択するマルチセレクトフィールド。アイテム名のパターンに対応します。
CWidgetFieldRadioButtonList	int32	1 つ以上のラジオボタンで構成されるラジオボタングループ。
CWidgetFieldRangeControl	int32	整数型の値を選択するスライダー。
CWidgetFieldReference	string	ダッシュボード上のこのウィジェットの一意の識別子を保存します。ウィジェット間の通信に使用されます。
CWidgetFieldSelect	int32	ドロップダウン選択ボックス。
CWidgetFieldSeverities	array of int32	トリガーの深刻度がプリセットされた *CWidgetFieldCheckBoxList*。
CWidgetFieldSparkline	array of (multiple mixed)	スパークラインチャートを設定できます。
CWidgetFieldTags	array of (string, int32, string)	1 つ以上のタグフィルター行を設定できます。
CWidgetFieldTextArea	string	複数行のテキストを入力するテキストエリア。
CWidgetFieldTextBox	string	1 行のテキストを入力するテキストボックス。
CWidgetFieldThresholds	array of (string, string)	色と数値の組み合わせを設定できます。
CWidgetFieldTimePeriod	array of string	期間を選択するフィールド。
CWidgetFieldTimeZone	string	タイムゾーンのドロップダウン。
CWidgetFieldUrl	string	URL を入力するテキストボックス。

Note:
`int32` フィールドの最大長は 10 文字、`ID` フィールドは 20 文字、`string` フィールドは 65535 文字です。
複合型（例: `array of (string, int32, string)`）の場合、制限は各要素に個別に適用されます。
フィールド固有の長さ制限は、`setMaxLength()` などのメソッドを使用して設定できます。

プレゼンテーション

このページでは、ウィジェットプレゼンテーションビューの作成に使用できるコンポーネントについて説明します。ウィジェットプレゼンテーションビューは、ウィジェットの設定に従ってデータを受信し、コンテナ内のダッシュボードに表示するウィジェットの一部です。

プレゼンテーションビューは、次の 3 つの部分で設定されます。

- ウィジェットアクション
- ウィジェットビュー
- JavaScript

Widget action

ウィジェットアクションクラス (*WidgetView*) には、プレゼンテーションビューモードでウィジェットを操作するためのメソッドが含まれています。ウィジェットアクションの大部分は、デフォルトのコントローラークラス *CControllerDashboardWidgetView* を使

用または拡張します。

ウィジェットのアクションクラスは *actions* ディレクトリに配置され、*manifest.json* ファイル内の*actions*パラメーター (*actions/widget.{id}.view/class*) で指定される必要があります。

actions/WidgetView.php example (implemented in the Zabbix-native *System information* widget)

```
class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getDefaultName()),
            'system_info' => CSystemInfoHelper::getData(),
            'info_type' => $this->fields_values['info_type'],
            'user_type' => CWebUser::getType(),
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

Widget view

ウィジェットビュークラス (*CWidgetView*) は、ウィジェットプレゼンテーションビューの構築を担当します。

ウィジェットビュークラスは *views* ディレクトリに配置する必要があります。ウィジェットビュークラスを含むファイルの名前がデフォルト (*widget.view.php*) とは異なる場合は、*manifest.json* ファイルの*actions*パラメーター (*actions/widget.{id}.view/view*) で指定する必要があります。

views/widget.view.php example

```
<?php

/**
 * My custom widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

JavaScript クラスは、ウィジェットデータの更新、ウィジェットのサイズの変更、ウィジェット要素の表示など、ウィジェットの動作を決定する責任があります。

すべての JavaScript 操作は、すべてのダッシュボードウィジェットのベース JavaScript クラスである *CWidget* を使用または拡張をします。*CWidget* クラスには、ウィジェット動作のデフォルトの実装を備えた一連のメソッドが含まれています。ウィジェットの複雑さに応じて、これらのメソッドはそのまま使用または拡張できます。

CWidget クラスには、次のメソッドが含まれています。

- ウィジェットのライフサイクルを定義するメソッド: *_init()*, *_registerEvents()*, *_doActivate()*, *_doDeactivate()*, *_doDestroy()*, *setEditMode()*.
- ウィジェットのデータ更新と表示を処理するメソッド: *_promiseUpdate()*, *_getUpdateRequestData()*, *_processUpdateResponse(response)*, *_processUpdateErrorResponse(error)*, *_setContents()*.
- ウィジェットの外観を変更するメソッド: *resize()*, *_hasPadding()*.

JavaScript クラスは *assets/js* ディレクトリに配置し、*manifest.json* ファイルの*assets* (*assets/js*) パラメーターで指定する必要があります。

ライフサイクルメソッド

ウィジェットのライフサイクルメソッドは、ダッシュボードによって呼び出され、ダッシュボード内に存在する間、ウィジェットのライフサイクルのさまざまな段階で呼び出されます。

onInitialize() メソッドは、HTML またはデータ操作を実行せずに、ウィジェットの初期状態や値を定義します。このメソッドは通常、ウィジェットをダッシュボードページに追加するか、ダッシュボードページをロードするか、ウィジェットが作成されたときに呼び出されます (ウィジェットオブジェクトがインスタンス化されています)。

例:

```
onInitialize() {
    this._time_offset = 0;
    this._interval_id = null;
    this._clock_type = CWidgetClock.TYPE_ANALOG;
    this._time_zone = null;
    this._show_seconds = true;
    this._time_format = 0;
    this._tzone_format = 0;
    this._show = [];
    this._has_contents = false;
    this._is_enabled = true;
}
```

onStart() メソッドは、データ操作を実行せずに、ウィジェットの HTML 構造を定義します。このメソッドは、ダッシュボードページの最初のアクティブ化、つまりダッシュボードとそのウィジェットがユーザーに完全に表示される前に呼び出されます。

例:

```
onStart() {
    this._events.resize = () => {
        const padding = 25;
        const header_height = this._view_mode === ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER
            ? 0
            : this._header.offsetHeight;

        this._target.style.setProperty(
            '--content-height',
            `${this._cell_height * this._pos.height - padding * 2 - header_height}px`
        );
    }
}
```

onActivate() メソッドは、カスタムイベントリスナー (ユーザーアクションに応答するためのもの) を有効にし、ウィジェットの更新サイクル (コンテンツを最新の状態に保つためのもの) を開始することにより、ウィジェットをアクティブにします。このメソッドは、ダッシュボードページがアクティブになったとき、つまりユーザーインターフェースに完全に表示されるときに呼び出されます。

onActivate() メソッドが呼び出される前は、ウィジェットは非アクティブ状態 (WIDGET_STATE_INACTIVE) であることに注意してください。呼び出しが成功した後、ウィジェットはアクティブ状態 (WIDGET_STATE_ACTIVE) に移行します。アクティブ状態では、ウィジェットはレスポンシブで、イベントに耳を傾け、コンテンツを定期的に更新し、他のウィジェットと対話できます。

例:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

onDeactivate() メソッドは、カスタムイベントリスナーを非アクティブ化し、ウィジェットの更新サイクルを停止することにより、ウィジェットのアクティビティとインタラクティブ性を停止します。このメソッドは、ダッシュボードページが無効になっている場合、つまり切り替えまたは削除されたとき、またはダッシュボードページからウィジェットが削除されたときに呼び出されます。

onDeactivate() メソッドが呼び出される前は、ウィジェットはアクティブ状態 (WIDGET_STATE_ACTIVE) であることに注意してください。呼び出しが成功した後、ウィジェットは非アクティブ状態 (WIDGET_STATE_INACTIVE に移行します。

例:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

onDestroy() メソッドは、ウィジェットがダッシュボードから削除される前にクリーンアップタスクを実行します。クリーンアップタスクとは、ウィジェットの初期化中に確立されたデータベース接続を閉じたり、リソースリークを回避するために一時的なデータをクリーンアップしてシステムメモリを解放したり、不要なイベントの取り扱いやメモリリークを防ぐためにサイズ変更イベントまたはボタンクリックに関連するイベントリスナーの登録を解除したりなどします。このメソッドは、ウィジェットまたはそれを含むダッシュボードページが削除されたときに呼び出されます。

Note that before the *onDestroy()* method is invoked, a widget in an active state (`WIDGET_STATE_ACTIVE`) is always deactivated with the invocation of the *onDeactivate()* method. *onDestroy()* メソッドが呼び出される前にアクティブな状態 (`WIDGET_STATE_ACTIVE`) のウィジェットは、*onDeactivate()* メソッドの呼び出しで常に非アクティブ化されることに注意してください。

例:

```
onDestroy() {
    if (this._filter_widget) {
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_MARK, this._events.mark);
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, this._events.select);
    }
}
```

onEdit() メソッドは、ダッシュボードが編集モードに移行したときのウィジェットの外観と動作を定義します。このメソッドは、ユーザーがウィジェットの編集ボタンやダッシュボードのダッシュボードの編集ボタンを操作して、ダッシュボードが編集モードに移行したときに呼び出されます。

例:

```
onEdit() {
    this._deactivateGraph();
}
```

更新プロセスメソッド

ウィジェット更新プロセスメソッドは、Zabbix サーバーまたはその他のデータソースから更新されたデータを取得し、ウィジェットに表示する役割を担います。

promiseUpdate() メソッドは、通常、Web リクエストまたは API 呼び出しを使用してデータを取得することにより、データ更新プロセスを開始します。このメソッドは、ダッシュボードページが表示されたときに呼び出され、その後、ダッシュボードページが別のダッシュボードページに切り替わるまで定期的に呼び出されます。

以下は、ほとんどの Zabbix ネイティブウィジェットが使用する *promiseUpdate()* メソッドのデフォルトの実装の例です。デフォルトの実装では、*promiseUpdate()* メソッドは、サーバーからデータを取得するための一般的なパターンに従います。適切な URL とリクエストパラメーターを使用して新しい `Curl` オブジェクトを作成し、*getUpdateRequestData()* メソッドで構築されたデータオブジェクトを使用して *fetch()* メソッドで POST リクエストを送信し、*processUpdateResponse(response)* または *processUpdateErrorResponse(error)* に応じてレスポンス (またはエラーレスポンス) を処理します。この実装は、通常 JSON 形式でデータを取得し、一貫した方法で処理するため、ほとんどのウィジェットに適しています。

```
promiseUpdate() {
    const curl = new Curl('zabbix.php');

    curl.setArgument('action', `widget.${this._type}.view`);

    return fetch(curl.getUrl(), {
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify(this.getUpdateRequestData()),
        signal: this._update_abort_controller.signal
    })
        .then((response) => response.json())
        .then((response) => {
            if ('error' in response) {
                this.processUpdateErrorResponse(response.error);

                return;
            }

            this.processUpdateResponse(response);
        });
    }
```

_getUpdateRequestData() メソッドは、ウィジェットの状態と設定からさまざまなプロパティとその対応する値 (ウィジェット識別子、フィルター設定、時間範囲など) を収集し、ウィジェットを更新するためのサーバーリクエストデータを準備し、更新リクエストでサーバーに送信される必要な情報を表すデータオブジェクトを構築します。このメソッドは、ウィジェット更新プロセス中にデフォルトの [*_promiseUpdate()*] メソッドの一部としてのみ呼び出されます。

デフォルトの実装:

```
_getUpdateRequestData() {
    return {
        templateid: this._dashboard.templateid ?? undefined,
        dashboardid: this._dashboard.dashboardid ?? undefined,
        widgetid: this._widgetid ?? undefined,
        name: this._name !== '' ? this._name : undefined,
        fields: Object.keys(this._fields).length > 0 ? this._fields : undefined,
        view_mode: this._view_mode,
        edit_mode: this._is_edit_mode ? 1 : 0,
        dynamic_hostid: this._dashboard.templateid !== null || this.supportsDynamicHosts()
            ? (this._dynamic_hostid ?? undefined)
            : undefined,
        ...this._content_size
    };
}
```

processUpdateResponse(response) メソッドは、更新リクエスト後にサーバーから受信したレスポンスを処理し、更新プロセスがエラーなしで成功している場合、ウィジェットデータをクリアし、*setContents()* メソッドで新しいコンテンツを表示します。このメソッドは、デフォルトの *promiseUpdate()* メソッドの一部として、つまりウィジェットの更新プロセス中のみに呼び出されます。

デフォルトの実装:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

processUpdateErrorResponse(error) メソッドは、レスポンスがエラーである場合、更新リクエスト後にサーバーから受信したレスポンスを処理し、エラーメッセージ/-s を表示します。このメソッドは、デフォルトの *promiseUpdate()* メソッドの一部として、つまりウィジェット更新プロセス中のみ呼び出されます。

デフォルトの実装:

```
_processUpdateErrorResponse(error) {
    this._setErrorContents({error});
}

_setErrorContents({error}) {
    const message_box = makeMessageBox('bad', error.messages, error.title)[0];

    this._content_body.innerHTML = '';
    this._content_body.appendChild(message_box);

    this._removeInfoButtons();
}
```

setContents(response) メソッドは、ウィジェットの更新プロセスがエラーなく成功した場合にウィジェットの内容を表示します。これには、DOM 要素の操作、UI コンポーネントの更新、スタイルの適用、フォーマットなどが含まれます。このメソッドは、デフォルトの *processUpdateResponse(response)* メソッドの一部として、つまり更新リクエスト後にサーバーから受信したレスポンスを処理するプロセス中のみ呼び出されます。

デフォルトの実装:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

プレゼンテーション変更メソッド

ウィジェットのプレゼンテーション変更メソッドは、ウィジェットの外観を変更します。

onResize() メソッドは、新しいウィジェットサイズに対応するためにウィジェットの視覚要素の調整を担当します。これには、要素の再配置、要素の寸法調整、テキストの切り捨て、サイズ変更中の応答性を向上させるための遅延読み込みの実装などが含まれます。このメソッドは、ウィジェットがサイズ変更されたときに呼び出されます。たとえば、ユーザーがウィジェットを手動でサイズ変更するときやブラウザウィンドウのサイズが変更されたときです。

例:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

hasPadding() メソッドは、ヘッダーを表示に構成されている場合、ウィジェットの下部に 8px の垂直パディングを適用します。このメソッドは、ダッシュボードページがアクティブになったとき、つまりユーザーインターフェースの表示されるページになったときに呼び出されます。

デフォルトの実装:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

一部のウィジェットの場合、使用可能なすべてのウィジェットスペースを使用して、たとえばカスタムの背景色を構成する必要があります。以下は、zabbix ネイティブのアイテムの値ウィジェットで使用される *hasPadding()* メソッドの実装の例です。

```
hasPadding() {
    return false;
}
```

チュートリアル

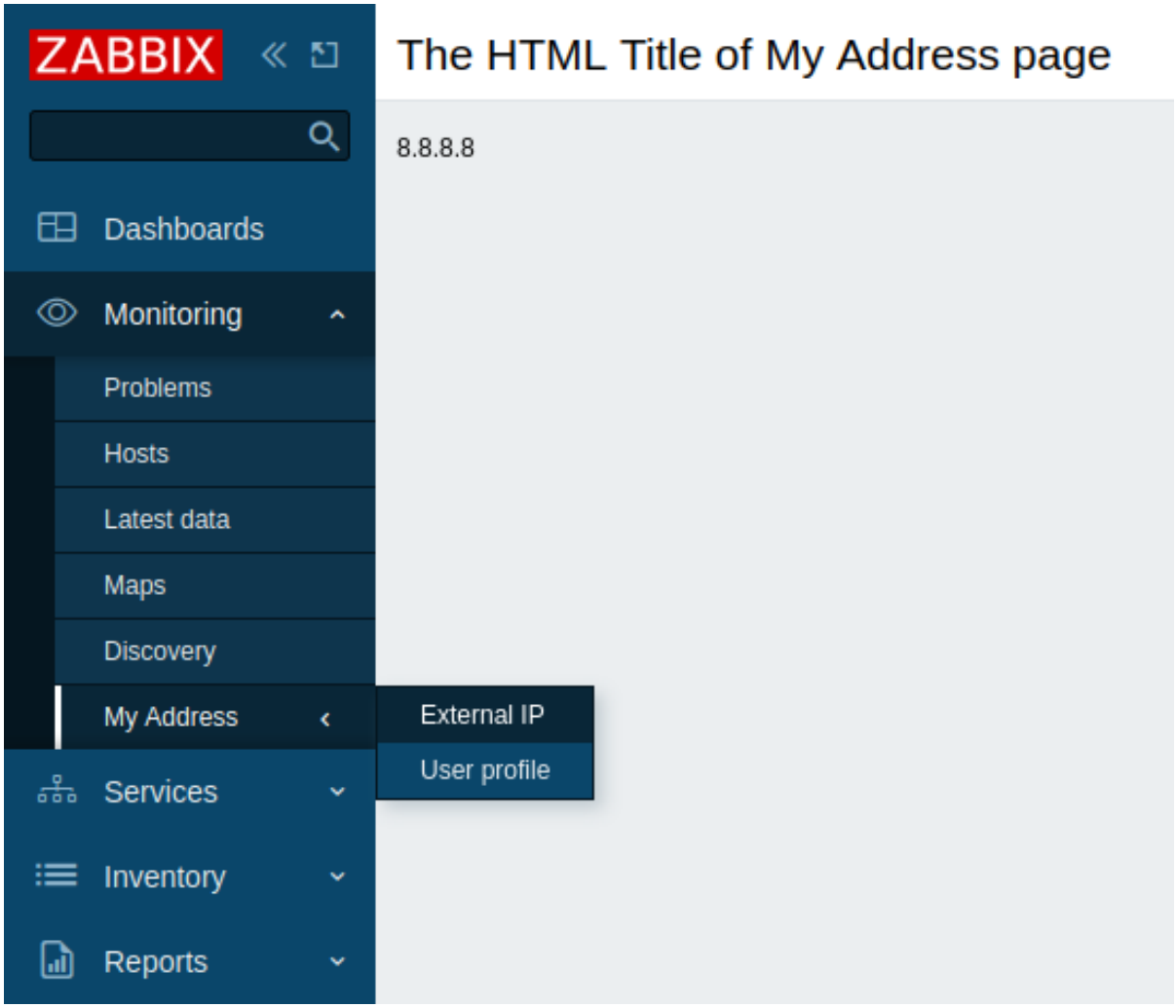
このセクションには、Zabbix でカスタムモジュールとウィジェットを構築する方法を説明する、実践的なステップバイステップのチュートリアルが含まれています。

モジュールの作成 **(**チュートリアル**)**

これは、シンプルな Zabbix フロントエンドモジュールを作成する方法を示すステップバイステップのチュートリアルです。このモジュールのすべてのファイルを ZIP アーカイブとしてダウンロードできます: MyAddress.zip。

作成するもの

このチュートリアルでは、まず新しい *My Address* メニューセクションを追加する Web インターフェースモジュールを作成し、次にそれを *https://api.seeip.org* に HTTP リクエストを送信して、その応答、つまりあなたのコンピュータの IP アドレスを、新しく作成した *My Address* メニューセクション内の新しいページに表示する、より高度なWeb インターフェースモジュールへと変換します。完成したモジュールは次のようになります。



パート I - 新しいメニューセクション

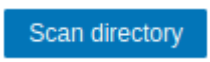
Zabbix Web インターフェースに空のモジュールを追加する

1. Zabbix Web インターフェースのインストール先の *modules* ディレクトリに、*MyAddress* ディレクトリを作成します（例: *zabbix/ui/modules*）。
2. 基本的なモジュールメタデータを含む *manifest.json* ファイルを作成します（サポートされているparameters の説明を参照してください）。

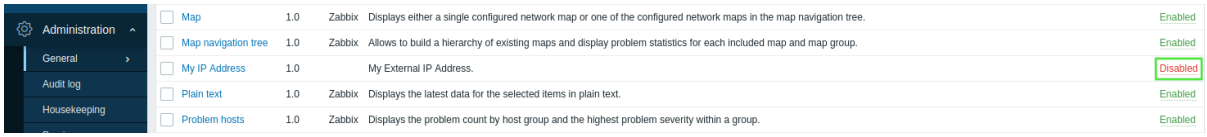
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Zabbix Web インターフェースで、*Administration* > *General* > *Modules* セクションに移動し、*Scan directory* ボタンをクリックします。



4. 一覧から新しいモジュール *My IP Address* を見つけ、”Disabled” ハイパーリンクをクリックして、モジュールの状態を”Disabled” から”Enabled” に変更します（モジュールが一覧に表示されない場合は、troubleshooting セクションを参照してください）。



これでモジュールは Web インターフェースに登録されました。ただし、まだモジュールの機能を定義していないため、表示はされません。モジュールディレクトリにコンテンツを追加すると、ページを更新した際に Zabbix Web インターフェースで変更がすぐに反映されます。

メニューセクションを作成する

1. *MyAddress* ディレクトリに *Module.php* ファイルを作成します。

このファイルは、デフォルトの *CModule* クラスを拡張する新しい *Module* クラスを実装します。*Module* クラスは、メインメニューに新しい *My Address* メニューセクションを追加します。

setAction() メソッドは、メニューセクションをクリックしたときに実行するアクションを指定します。まずは、あらかじめ定義されている *userprofile.edit* アクションを使用できます。これにより、*User profile* ページが開きます。このチュートリアルのpart III では、カスタムアクションの作成方法を学びます。

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

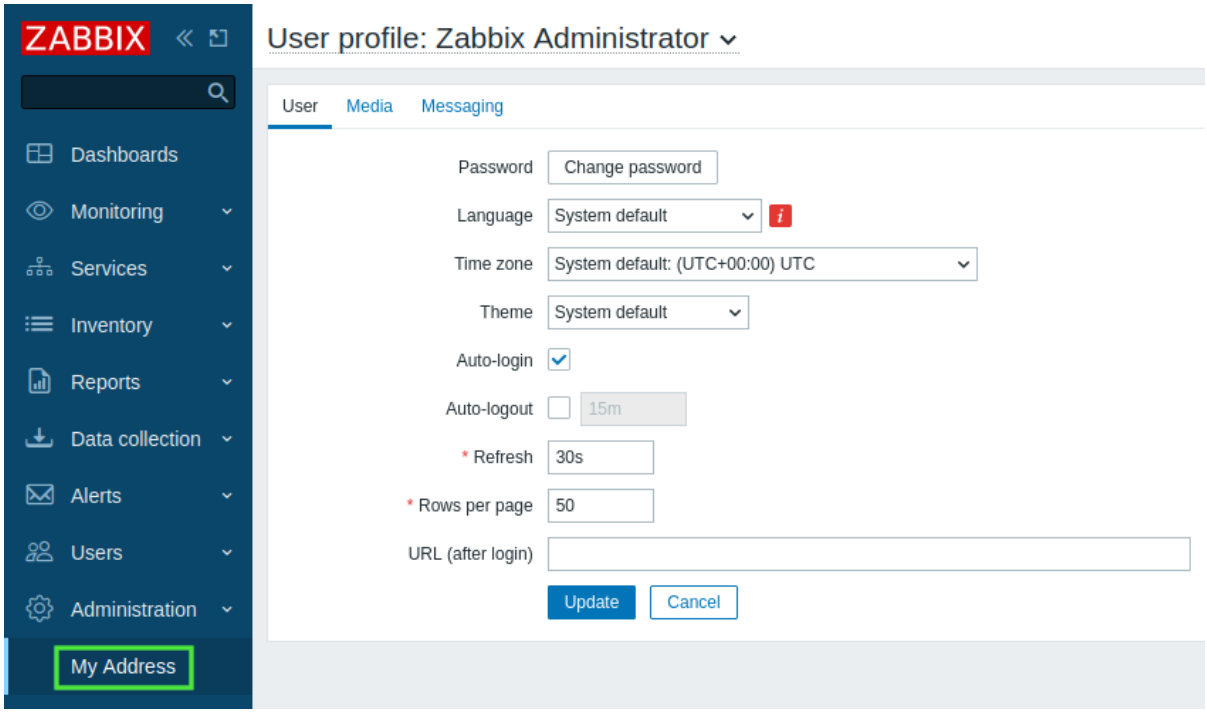
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
'userprofile.edit' は他のアクションに置き換えられます。たとえば、'charts.view'（カスタムグラフを開く）、'problems.view'（*Monitoring* > 障害を開く）、または 'report.status'（*System information* レポートを開く）などです。

3. Zabbix Web インターフェースを更新します。これで、Zabbix のメインメニューの下部に新しい *My Address* セクションが表示されます。*My Address* をクリックすると、*User profile* ページが開きます。



パート II - メニューセクションの配置変更

このパートでは、*My Address* メニューセクションを *Monitoring* セクションへ移動し、さらにその中にネストされたメニューを追加します。
その結果、ユーザーは *Monitoring* > *My Address* メニューセクションから 2 つのサブメニューページにアクセスできるようになります。

1. *Module.php* ファイルを開いて編集します。

ui/modules/MyAddress/Module.php

```
<?php

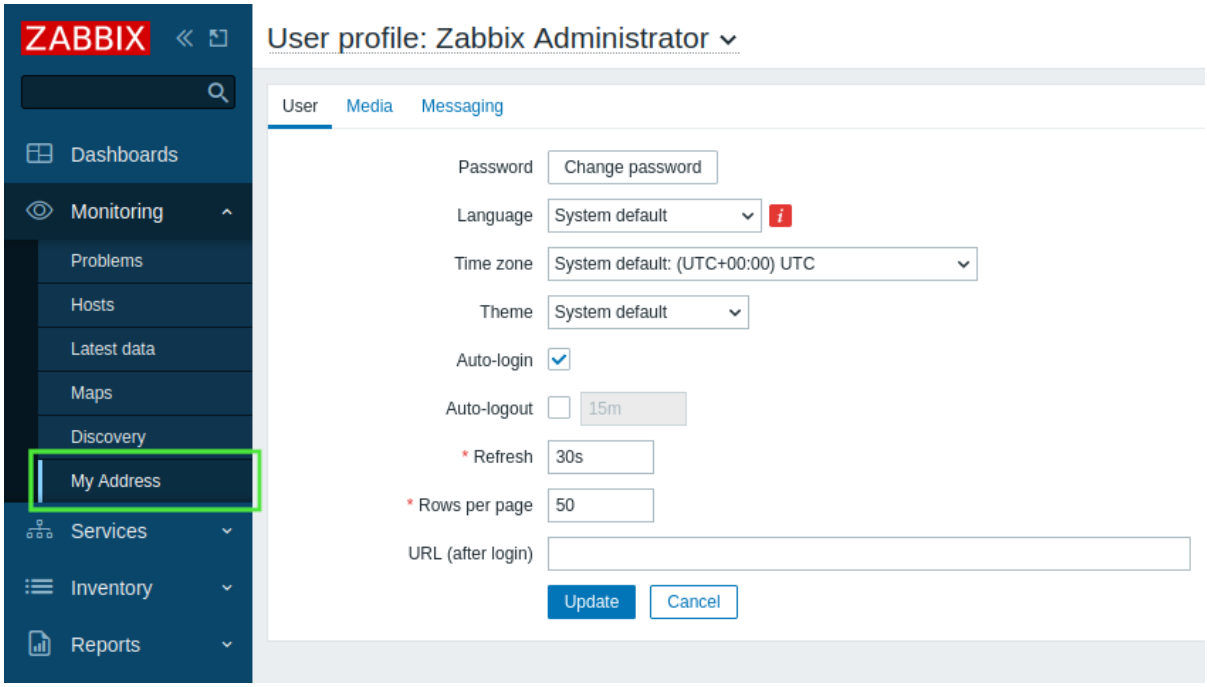
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Zabbix Web インターフェースを更新します。*Monitoring* メニューセクションを展開し、*My address* セクションが *Discovery* セクションの下に配置されていることを確認します。



3. *My Address* メニューセクションにネストされたページを追加するには、再度 *Module.php* ファイルを開いて編集します。

この手順では、2 つのサブセクションを作成します。

- 次の手順で定義する新しい'my.address' アクションを実行する *External IP*;
- *User profile* ページを開くために、あらかじめ定義されている'userprofile.edit' アクションを実行する *User profile*。

ネストされたメニューでは、前の手順で使用したクラスに加えて *CMenu* クラスを使用する必要があることに注意してください。

ui/modules/MyAddress/Module.php

```
<?php

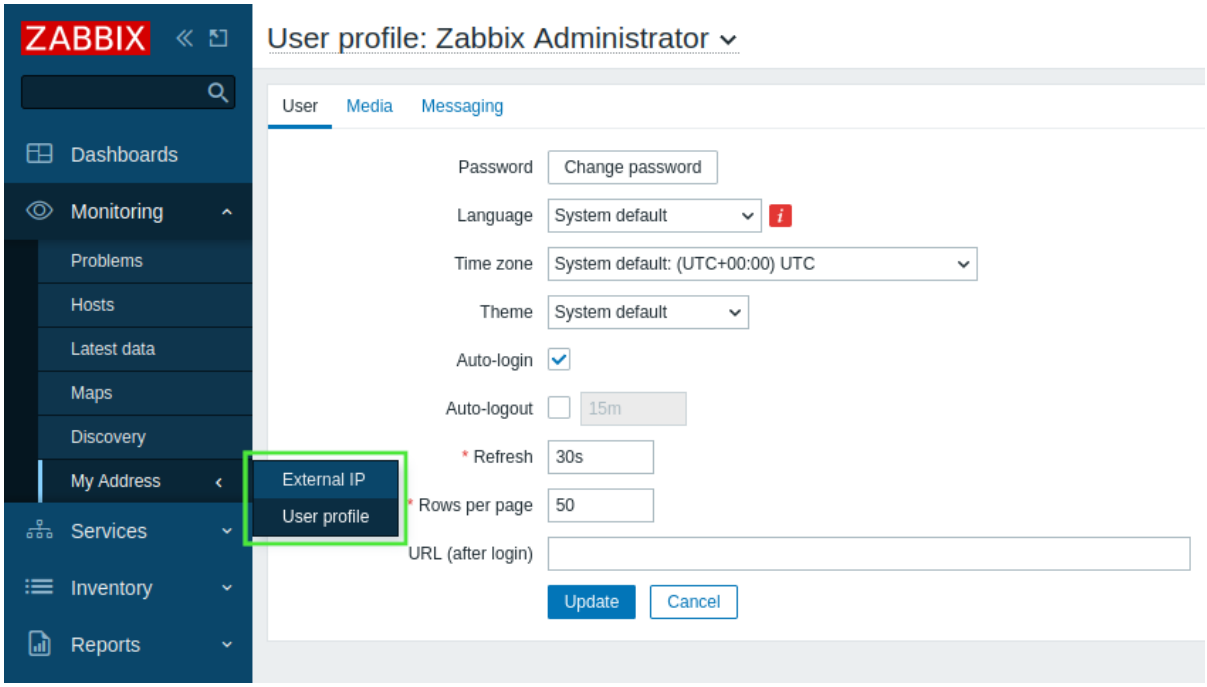
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenu,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Zabbix Web インターフェースを更新します。*My address* メニューセクションに、*External IP* と *User profile* の 2 つのページを持つ 3 階層目のメニューが追加されていることを確認します。



第 III 部 - モジュールアクション

アクションは 2 つのファイルで実装されます - *actions/MyAddress.php* と *views/my.address.php* です。
actions/MyAddress.php ファイルはビジネスロジックの実装を担当し、***views/my.address.php*** ファイルはビューを担当します。

1. *MyAddress* ディレクトリ内に *actions* ディレクトリを作成します。
2. *actions* ディレクトリ内に *MyAddress.php* ファイルを作成します。

アクションロジックは *MyAddress* クラスで定義されます。
このアクションクラスは、*init()*, *checkInput()*, *checkPermissions()*, *doAction()* の 4 つの関数を実装します。
Zabbix Web インターフェースは、アクションが要求されたときに *doAction()* 関数を呼び出します。
この関数は、モジュールのビジネスロジックを担当します。

Attention:
データは連想配列として整理する必要があります。
配列は多次元にすることができ、ビューが期待する任意のデータを含めることができます。

ui/modules/MyAddress/actions/MyAddress.php

```
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
```

```
        $response = new CControllerResponseData($data);
        $this->setResponse($response);
    }
}
```

3. *MyAddress* ディレクトリ内に新しい *views* ディレクトリを作成します。
4. *views* ディレクトリ内に *my.address.php* ファイルを作成し、モジュールビューを定義します。

変数 $data は、特に定義しなくてもビュー内で利用できることに注意してください。
フレームワークが連想配列を自動的にビューへ渡します。

ui/modules/MyAddress/views/my.address.php

```
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

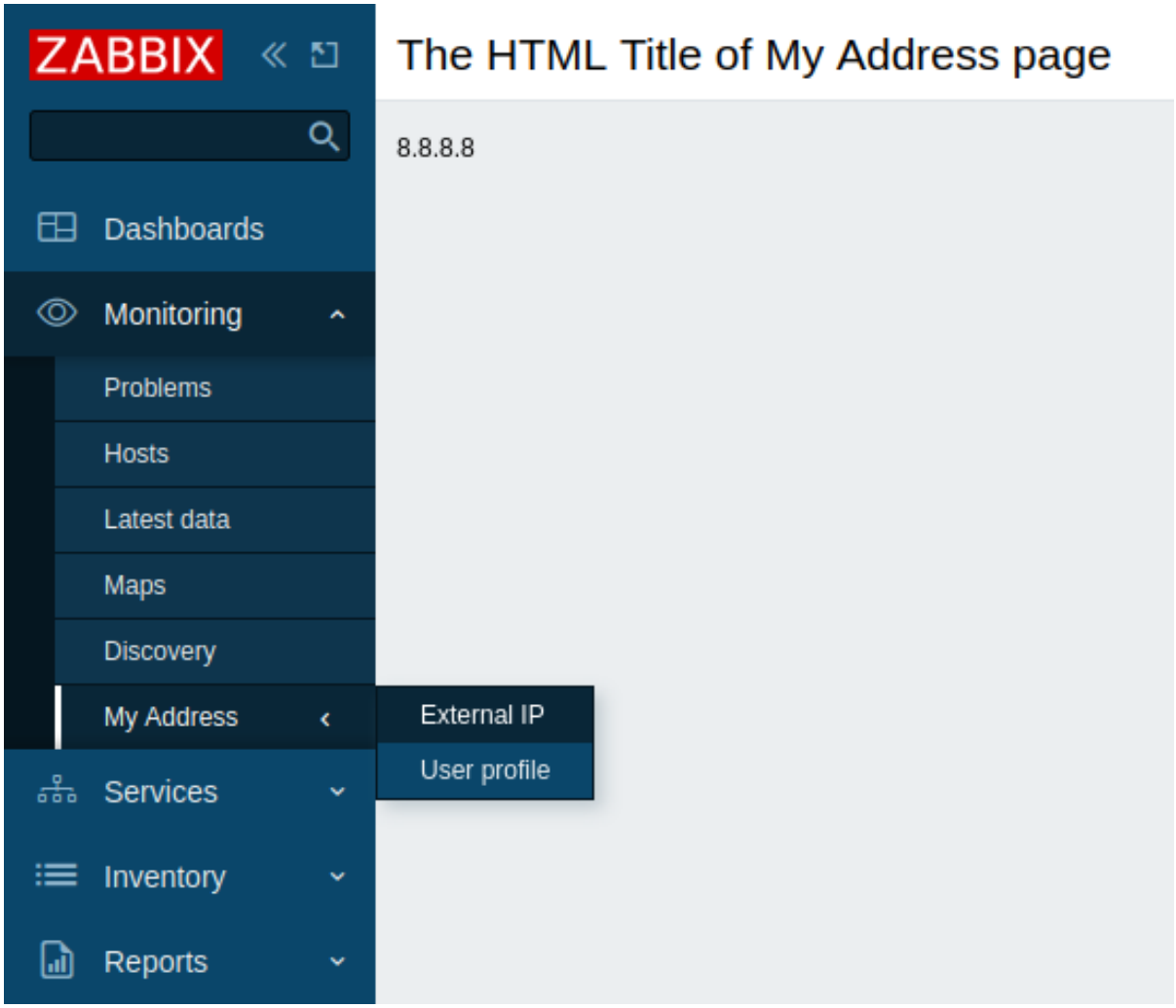
5. モジュールアクションは *manifest.json* ファイルに登録する必要があります。*manifest.json* を開き、次の内容を含む新しい `actions` オブジェクトを追加します。

- アクションキーは、アクション名を小文字 (a-z) で、単語をドットで区切って記述します（例: `my.address`）。
- `my.address` オブジェクトの `class` キーの値として、アクションクラス名 (`MyAddress`) を指定します。
- `my.address` オブジェクトの `view` キーの値として、アクションビュー名 (`my.address`) を指定します。

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

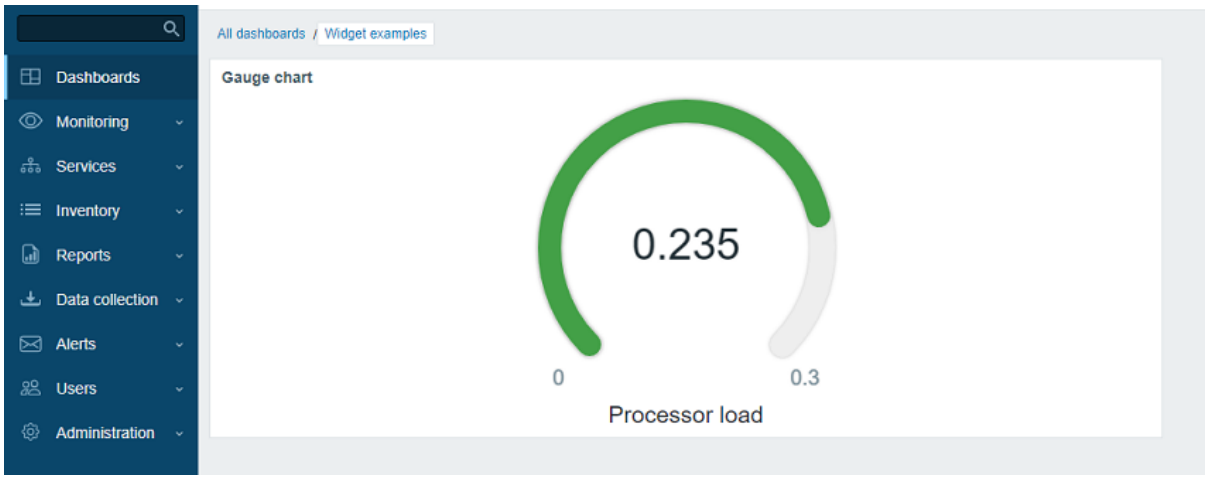
6. Zabbix Web インターフェースを更新します。*My address* > *External IP* をクリックすると、コンピューターの IP アドレスを確認できます。



ウィジェットの作成 **(**チュートリアル**)**

これは、シンプルなダッシュボードウィジェットを作成する方法を示すステップバイステップのチュートリアルです。このウィジェットのすべてのファイルを ZIP アーカイブとしてダウンロードできます: lesson_gauge_chart.zip。

作成するもの　このチュートリアルでは、まず基本的な「Hello, world!」ウィジェットを作成し、その後、アイテムの値をゲージチャートとして表示するより高度なウィジェットに変換します。完成したウィジェットは次のようになります。



パート **I - "Hello, world!"**　このセクションでは、最小限必要なウィジェット要素を作成し、新しいウィジェットを Zabbix フロントエンドに追加する方法を学びます。

Zabbix Web インターフェースに空のウィジェットを追加する

1. Zabbix Web インターフェースのインストール先にある `modules` ディレクトリ内に、`lesson_gauge_chart` ディレクトリを作成します（例: *zabbix/ui/modules*）。

> **Note:**
> すべてのカスタムウィジェットは外部モジュールとして扱われるため、Zabbix Web インターフェースのインストール先にある `modules` ディレクトリに追加する必要があります（例: *zabbix/ui/modules*）。`zabbix/ui/widgets` ディレクトリは Zabbix 標準ウィジェット用に予約されており、Zabbix UI とともに更新されます。

2. 基本的なウィジェットのメタデータを含む `manifest.json` ファイルを作成します（サポートされているparameters の説明を参照してください）。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

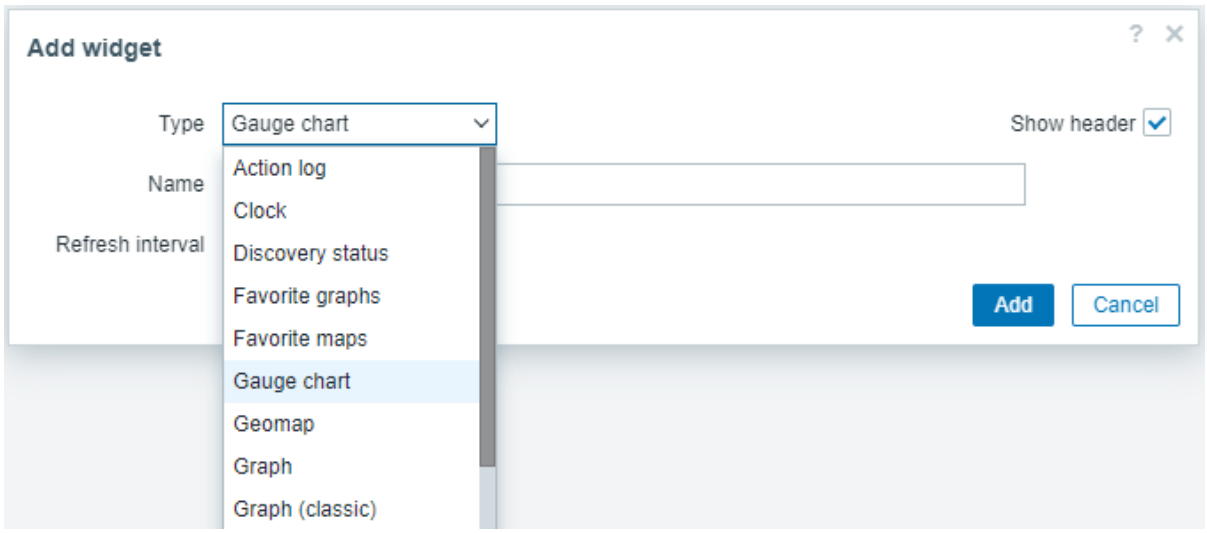
3. Zabbix Web インターフェースで、*Administration* > *General* > *Modules* セクションに移動し、*Scan directory* ボタンをクリックします。



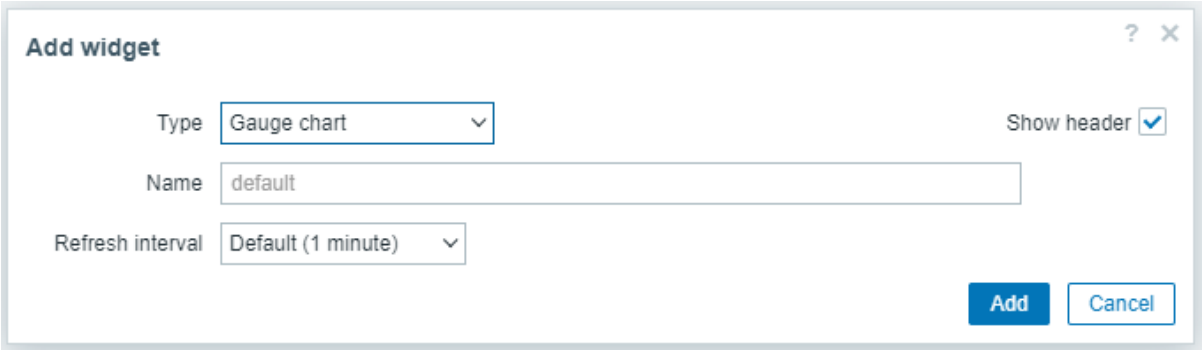
4. 一覧から新しいモジュール *Gauge chart* を見つけ、”Disabled” のハイパーリンクをクリックして、モジュールの状態を”Disabled” から”Enabled” に変更します（モジュールが一覧に表示されない場合は、troubleshooting セクションを参照してください）。



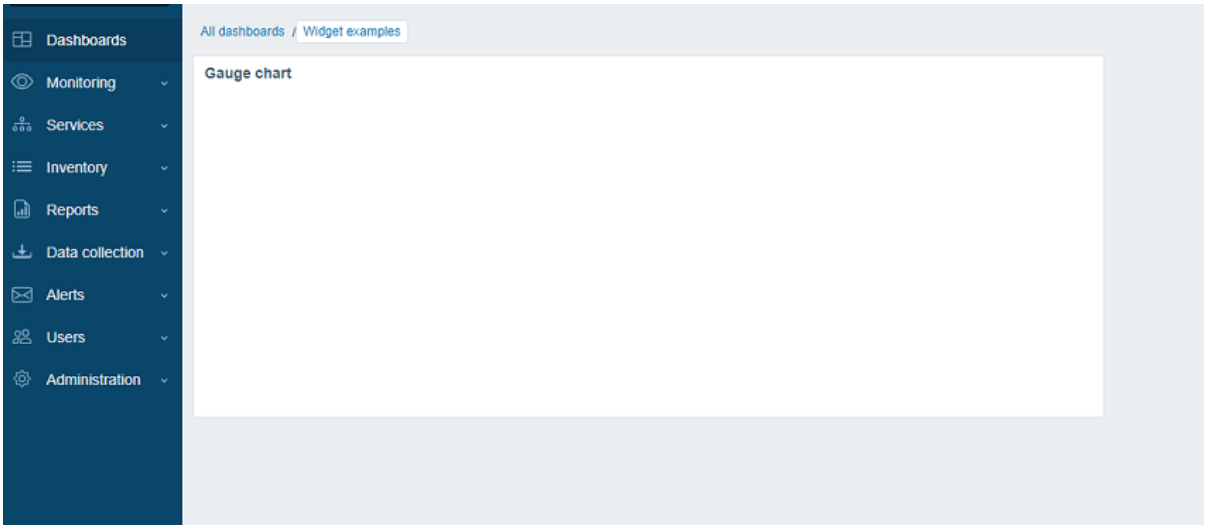
5. ダッシュボードを開き、編集モードに切り替えて新しいウィジェットを追加します。*Type* フィールドで”Gauge chart” を選択します。



6. この時点で、*Gauge chart* ウィジェットの設定には共通のウィジェット項目である *Name* と *Refresh interval* □□□□□□□ が含まれています。*Add* をクリックして、ウィジェットをダッシュボードに追加します。



7. ダッシュボードに空のウィジェットが表示されるはずです。右上隅の *Save changes* をクリックして、ダッシュボードを保存します。



ウィジェットビューを追加する

Note:
ウィジェットの **view** ファイルは views ディレクトリに配置する必要があります（このチュートリアルでは ui/modules/lesson_gauge_chart/views/）。ファイル名が既定の widget.view.php の場合は、manifest.json ファイルに登録する必要はありません。ファイル名が異なる場合は、manifest.json ファイルの actions/widget.lesson_gauge_chart.view セクションで指定してください。

1. lesson_gauge_chart ディレクトリに views ディレクトリを作成します。
2. views ディレクトリに widget.view.php ファイルを作成します。

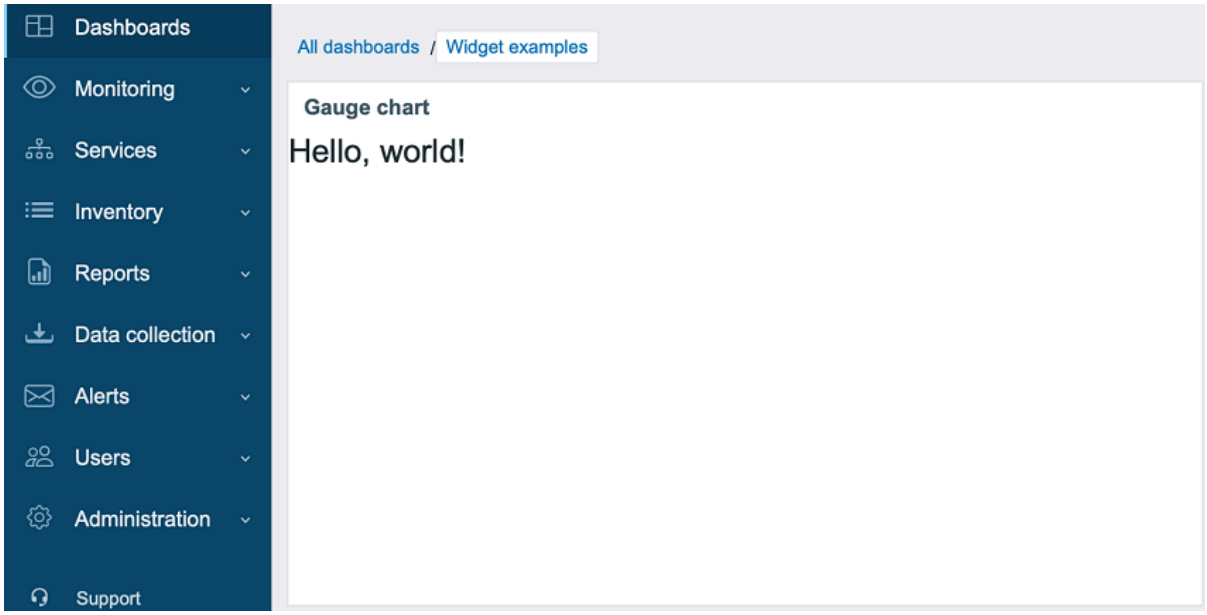
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. ダッシュボードを更新します。これで *Gauge chart* ウィジェットに”Hello, world!” が表示されます。



パート **II** - ゲージチャート　設定ビューに設定を追加し、widget view で使用する

このセクションでは、widget の設定フィールドを追加し、その入力値を widget view でテキストとして表示する方法を学びます。

widget の設定は、フォーム (Zabbix\Widgets\CWidgetForm) と widget フォームビュー (widget.edit.php) で構成されます。フィールド (Zabbix\Widgets\CWidgetField) を追加するには、Zabbix\Widgets\CWidgetForm を継承する WidgetForm クラスを作成する必要があります。

フォームには、ユーザーが入力した値を検証するために使用される、さまざまな種類のフィールド (Zabbix\Widgets\CWidgetField) が含まれます。各入力要素タイプに対応するフォームフィールド (Zabbix\Widgets\CWidgetField) は、値をデータベースに保存するための単一の形式に変換します。

Note:
widget の **form** ファイルは includes ディレクトリに配置する必要があります（このチュートリアルでは ui/modules/lesson_gauge_chart/includes/）。ファイル名がデフォルトの WidgetForm.php であれば、manifest.json ファイルに登録する必要はありません。ファイル名が異なる場合は、manifest.json ファイルの widget/form_class セクションで指定してください。

1. lesson_gauge_chart ディレクトリ内に新しい includes ディレクトリを作成します。
2. includes ディレクトリ内に WidgetForm.php ファイルを作成します。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. widget 設定フォームに *Description* フィールドを追加します。これは通常のテキストフィールドで、ユーザーは任意の文字セットを入力できます。これには CWidgetFieldTextBox クラスを使用できます。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

use Zabbix\Widgets\Fields\CWidgetFieldTextBox;

class WidgetForm extends CWidgetForm {
```

```
    public function addFields(): self {
        return $this
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

4. views ディレクトリに widget 設定ビュー用の widget.edit.php ファイルを作成し、新しい *Description* フィールドのビューを追加します。CWidgetFieldTextBox フィールドクラスのビューは CWidgetFieldTextBoxView です。

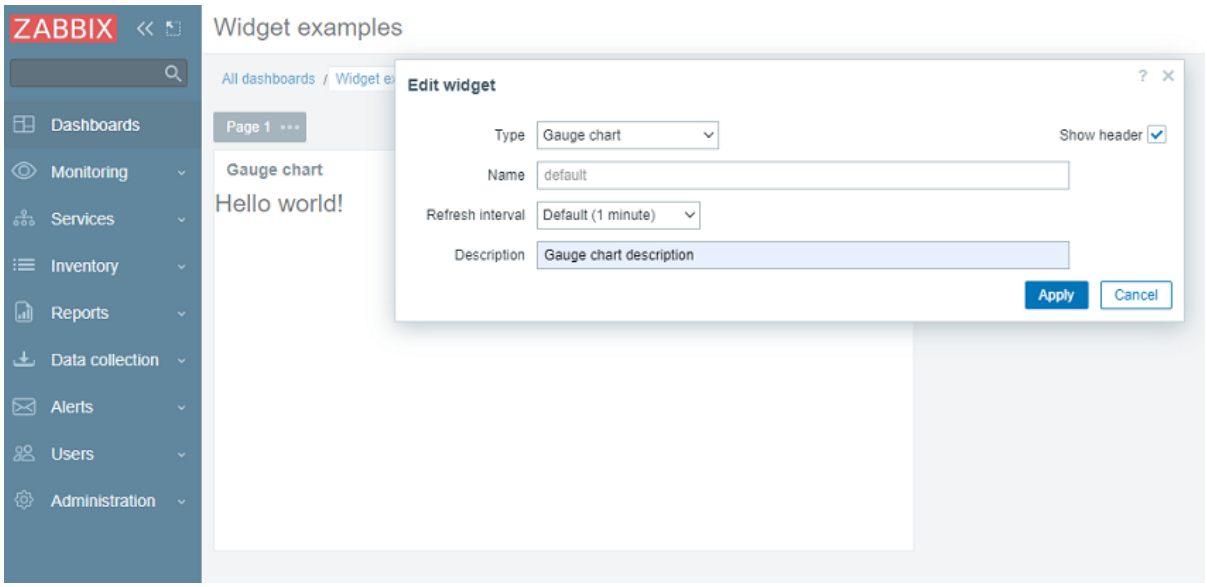
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. ダッシュボードに移動し、widget の歯車アイコンをクリックして widget 設定フォームを開きます。
6. widget 設定フォームに新しい *Description* テキストフィールドが追加されています。たとえば *Gauge chart description* など、任意の値を入力します。



7. widget 設定フォームで *Apply* をクリックします。その後、右上の *Save changes* をクリックしてダッシュボードを保存します。新しい説明はどこにも表示されず、widget には引き続き”Hello, world!” が表示されていることに注意してください。

新しい説明を widget に表示するには、*Description* フィールドの値をデータベースから取得し、widget view に渡す必要があります。そのために、action クラスを作成します。

8. lesson_gauge_chart ディレクトリ内に新しい actions ディレクトリを作成します。
9. actions ディレクトリに WidgetView.php ファイルを作成します。WidgetView action クラスは CControllerDashboardWidgetVi クラスを継承します。

widget 設定フィールドの値は、action クラスの $fields_values プロパティに保存されます。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

10. manifest.json を開き、actions/widget.lesson_gauge_chart.view セクションに WidgetView を action クラスとして登録します。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. これで、$data['description'] に含まれる description フィールドの値を widget view で使用できるようになります。views/widget.view.php を開き、静的テキスト”Hello, world!” を $data['description'] に置き換えます。

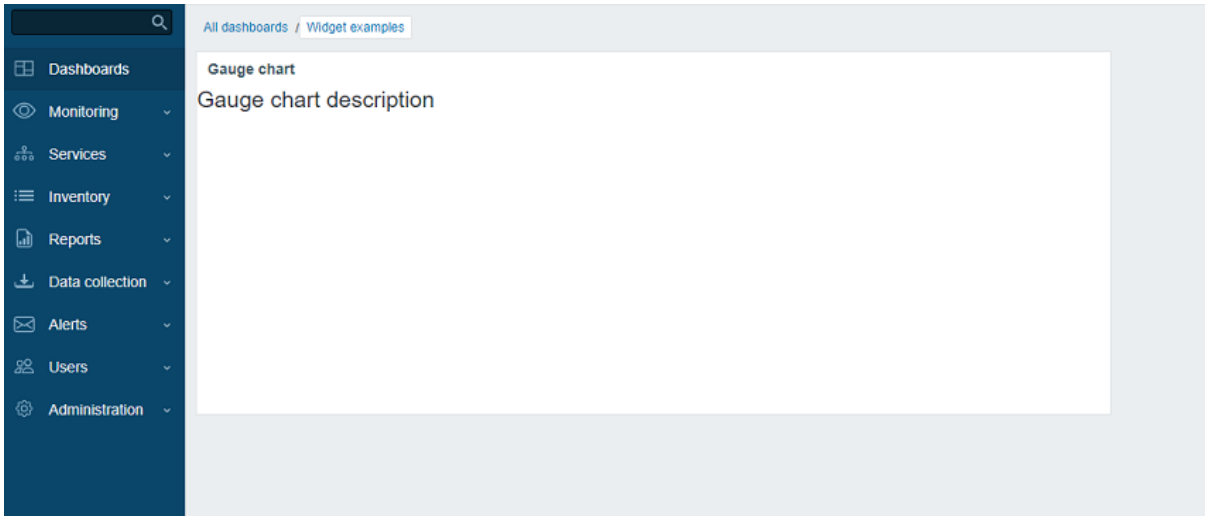
ui/modules/lesson_gauge_chart/views/widget.view.php

```php
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. ダッシュボードページを更新します。これで”Hello, world!” の代わりに widget の説明テキストが表示されるはずです。



API 経由でアイテムの値を取得する

このウィジェットは、ユーザーが選択したアイテムの最新値を表示する必要があります。
そのため、ウィジェット設定でアイテムを選択できるようにする必要があります。

このセクションでは、ウィジェットフォームにアイテム選択フィールドを追加する方法と、そのフィールドの表示要素を設定画面に追加する方法を学びます。
その後、ウィジェットコントローラーは API リクエストを通じてアイテムデータとその値を取得できるようになります。
取得した値は、ウィジェット表示画面に表示できます。

1. includes/WidgetForm.php を開き、CWidgetFieldMultiSelectItem フィールドを追加します。
   これにより、設定フォームでアイテムを選択できるようになります。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

2. views/widget.edit.php を開き、フィールドの表示コンポーネントを設定画面に追加します。

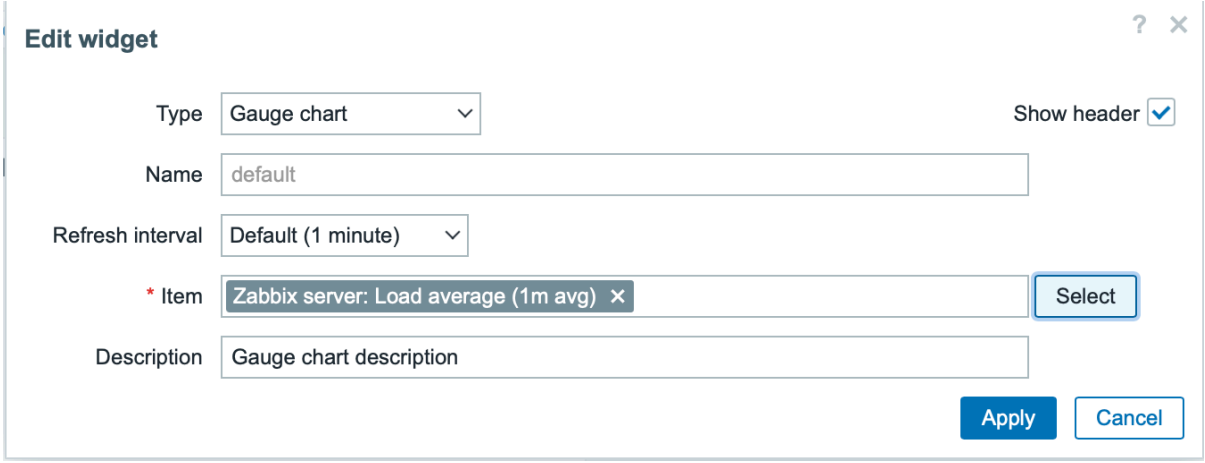
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. ダッシュボードに戻り、ウィジェット内の歯車アイコンをクリックして、ウィジェット設定フォームを開きます。
4. ウィジェット設定フォームに新しい入力フィールド *Item* が追加されています。
   ホスト"Zabbix server" とアイテム"Load average (1m avg)" を選択します。



5. ウィジェット設定フォームで *Apply* をクリックします。
   その後、右上隅の *Save changes* をクリックしてダッシュボードを保存します。
6. actions/WidgetView.php を開いて修正します。

これ以降、アイテム ID はウィジェットコントローラー内の $this->fields_values['itemid'] で利用できるようになります。doAction() コントローラーメソッドは、API メソッドitem.get を使用してアイテムデータ（名前、値の型、単位）を取得し、API メソッドhistory.get を使用してアイテムの最新値を取得します。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
```

```
            ]
        ]);

        $value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'value' => $value,
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

7. views/widget.view.php を開き、アイテムの値をウィジェット表示に追加します。

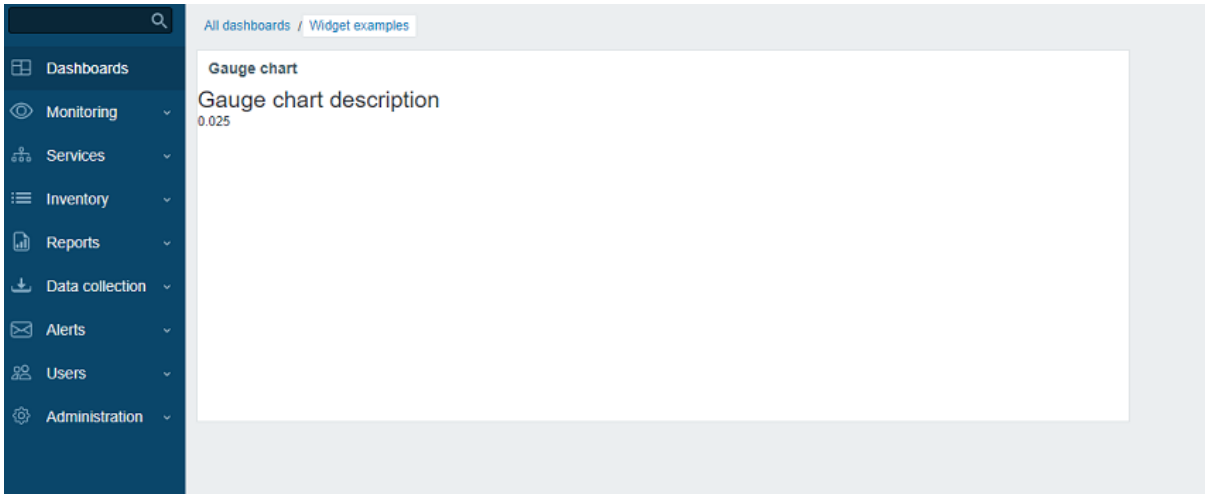
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. ダッシュボードページを更新します。
   ウィジェットにアイテムの最新値が表示されます。



設定ビューに詳細設定を追加する

このセクションでは、色、最小値と最大値、単位、そして前の手順で作成した *Description* フィールドなどの任意パラメータを含む、展開/折りたたみ可能な *Advanced configuration* セクションを追加する方法を学びます。

1. メインのウィジェットディレクトリ `lesson_gauge_chart` に `Widget.php` ファイルを作成し、新しい `Widget` クラスを定義します。

`Widget` クラスは `CWidget` 基底クラスを継承し、既定のウィジェット設定（この場合は翻訳）を追加/上書きします。以下に示すJavaScript は、データがない場合に文字列”No data” を表示します。
”No data” 文字列は Zabbix の UI 翻訳ファイルに含まれています。

ウィジェット定数がある場合は、それらも `Widget` クラスで定義することを推奨します。

`ui/modules/lesson_gauge_chart/Widget.php`

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. `includes/WidgetForm.php` を開き、新しいフィールド *Color*（カラーピッカー）、*Min*（数値フィールド）、*Max*（数値フィールド）、*Units*（選択）を追加し、次の手順で使用できるようにカラーピッカーの既定のカラーパレットを定義します。

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};
```

```
use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. views/widget.edit.php を開き、設定ビューにフィールドの視覚コンポーネントを追加します。

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
```

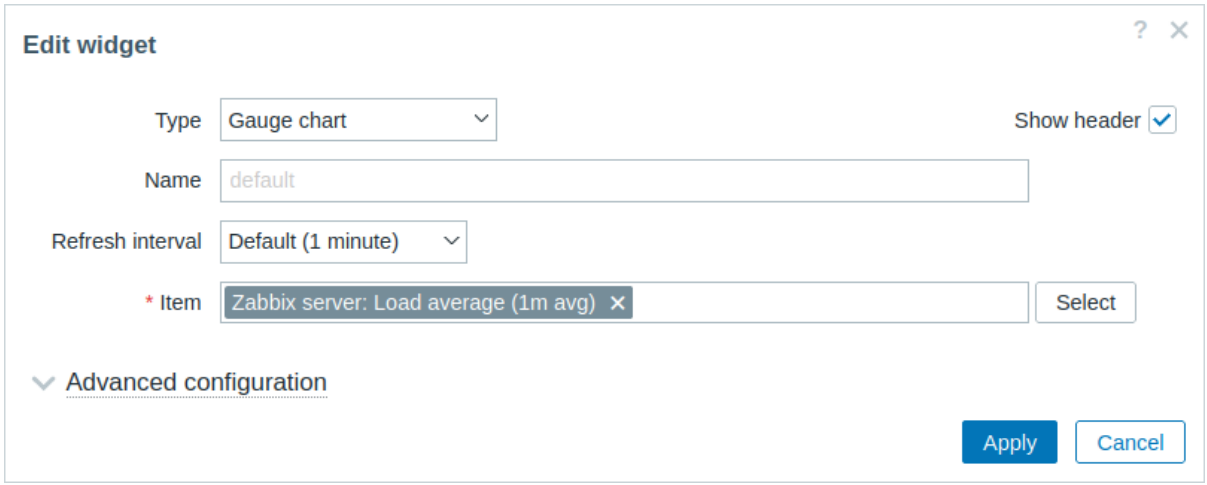
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

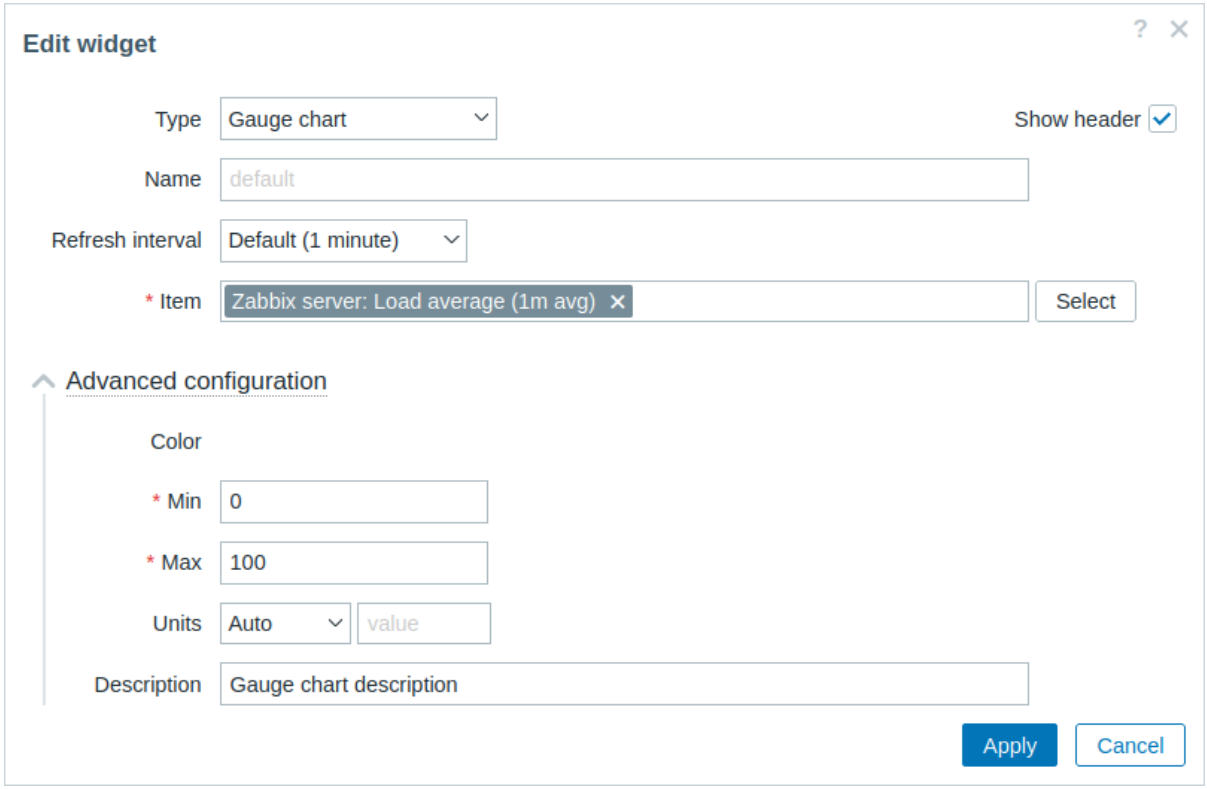
Note:
CWidgetFormView クラスの addField() メソッドは、2 番目のパラメータとして CSS クラス文字列を受け取ります。

4. ダッシュボードに戻り、編集モードに切り替えて、ウィジェット内の歯車アイコンをクリックし、ウィジェット設定フォームを開きます。
   ウィジェット設定フォームには、新しい展開/折りたたみ可能なセクション *Advanced configuration* が追加されています。



5. *Advanced configuration* セクションを展開すると、追加のウィジェット設定フィールドが表示されます。
   フィールド *Color* にはまだカラーピッカーがないことに注意してください。
   これは、カラーピッカーを JavaScript で初期化する必要があるためです。これは次のセクション - ウィジェットに *JavaScript*

を追加する - で追加します。



ウィジェットに JavaScript を追加する

このセクションでは、JavaScript で作成したゲージチャートを追加し、最新値が正常か、または高すぎる/低すぎるかを表示する方法を学びます。

1. views ディレクトリに widget.edit.js.php ファイルを作成します。

JavaScript は、設定画面でカラーピッカーを初期化する役割を担います。

ui/modules/lesson_gauge_chart/views/widget.edit.js.php

```
<?php

use Modules\LessonGaugeChart\Widget;

?>

window.widget_lesson_gauge_chart_form = new class {

    init({color_palette}) {
        this._unit_select = document.getElementById('value_units');
        this._unit_value = document.getElementById('value_static_units');

        this._unit_select.addEventListener('change', () => this.updateForm());

        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }

        this.updateForm();
    }
```

```
    updateForm() {
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

2. views/widget.edit.php を開き、widget.edit.js.php ファイルを設定画面に追加します。
   これには includeJsFile() メソッドを使用します。
   インライン JavaScript を追加するには、addJavaScript() メソッドを使用します。

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

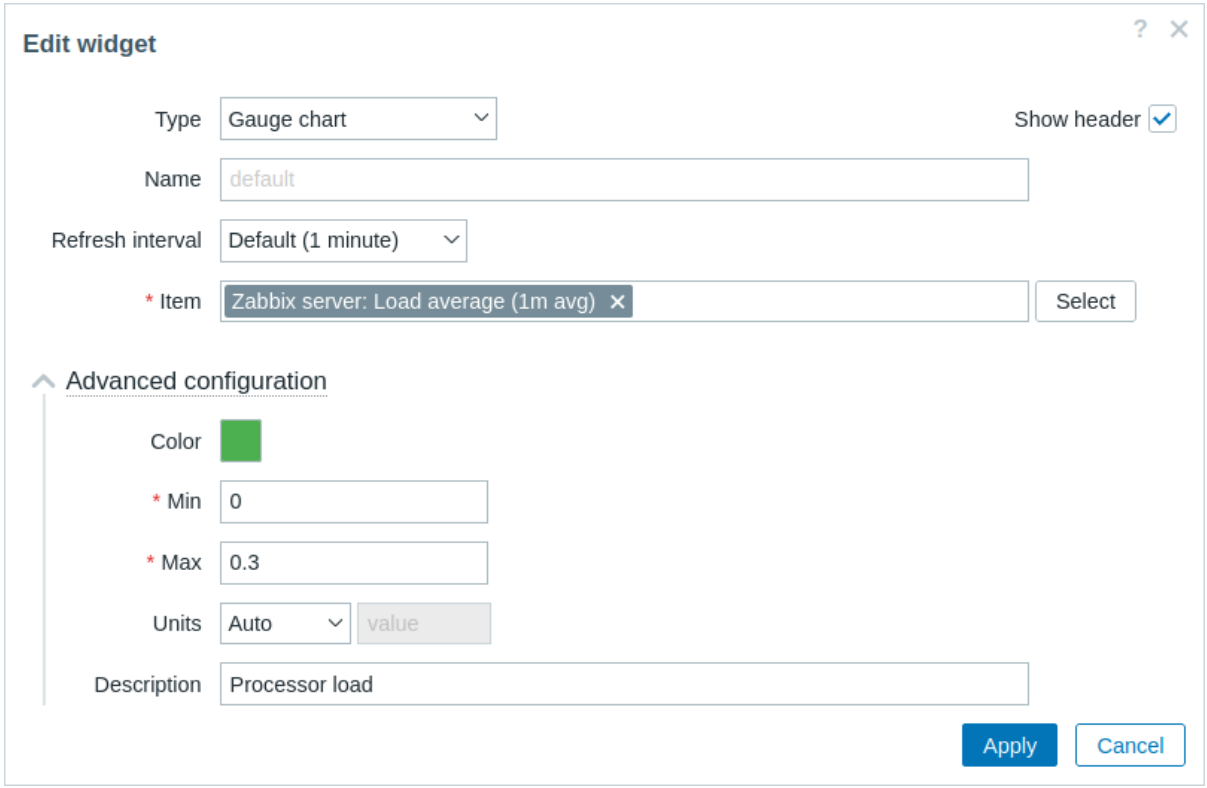
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. ダッシュボードに戻り、ウィジェット内の歯車アイコンをクリックして、ウィジェット設定フォームを開きます。
   次に、*Advanced configuration* セクションを展開して、初期化されたカラーピッカーを表示します。
   各フィールドに値を入力し、ゲージチャートの色を選択します。



4. ウィジェット設定フォームで *Apply* をクリックします。
   その後、右上の *Save changes* をクリックしてダッシュボードを保存します。

5. actions/WidgetView.php を開き、コントローラーを更新します。

$this->fields_values プロパティには、すべての *Advanced configuration* フィールドの値が含まれるようになっています。
設定と選択したアイテム値をウィジェットビューに渡せるよう、コントローラーを完成させます。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
```

```
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. views/widget.view.php を開いて修正します。

次の手順で描画するゲージチャート用のコンテナと、説明用のコンテナを作成する必要があります。

JavaScript に値を JSON オブジェクトとして渡すには、setVar() メソッドを使用します。

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. lesson_gauge_chart ディレクトリに新しい assets ディレクトリを作成します。
   このディレクトリは、JavaScript、CSS、さらにフォントや画像などの他のアセットを保存するために使用します。
8. ウィジェットビュー用の JavaScript のために、assets ディレクトリ内に js ディレクトリを作成します。
9. assets/js ディレクトリに class.widget.js ファイルを作成します。

この JavaScript ウィジェットクラスは、すべてのダッシュボードウィジェットのベース JavaScript クラスである CWidget を継承します。

ダッシュボードは、ウィジェットの正しい実装に依存しており、各 JavaScript メソッドを呼び出すことで、関連する情報をウィジェットに伝えます。
また、ダッシュボードは、何らかの操作が発生したときにウィジェットがイベントを生成することも期待しています。
そのため、CWidget クラスには、ウィジェットの動作のデフォルト実装を持つ一連のメソッドが含まれており、クラスを継承することでカスタマイズできます。

このケースではカスタマイズが必要なため、次のウィジェット動作について独自ロジックを実装します。

- ウィジェットの初期化。ウィジェットの初期状態を定義する役割があります（onInitialize() メソッドを参照）。
- ウィジェット更新処理が成功し、エラーがない場合のウィジェット内容の表示、つまりゲージチャートの描画（processUpdateResponse(response) メソッド、および関連する _resizeChart() と _updatedChart() メソッドを参照）。
- ウィジェットのサイズ変更（onResize() メソッド、および関連する _resizeChart() メソッドを参照）。

ゲージチャートウィジェットのその他の側面については、ウィジェット動作のデフォルト実装を使用します。
CWidget クラスの JavaScript メソッドの詳細については、JavaScript を参照してください。

この JavaScript はウィジェットビューに必要なため、ダッシュボードページと一緒に読み込まれる必要があります。
JavaScript の読み込みを有効にするには、手順 10 に示すように assets/js と js_class パラメータを manifest.json ファイルで更新する必要があります。

ui/modules/lesson_gauge_chart/assets/js/class.widget.js

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }

    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');

            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
```

```
        }

        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;

        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;

                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);
```

```
                ctx.beginPath();
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';

                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. manifest.json を開き、次を追加します。

- assets/js セクションの配列にファイル名 (class.widget.js) を追加する。
- widget セクションの js_class パラメータにクラス名 (WidgetLessonGaugeChart) を追加する。

これで WidgetLessonGaugeChart クラスはダッシュボードと一緒に自動的に読み込まれるようになります。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
```

```
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "js": ["class.widget.js"]
    }
}
```

ウィジェットに CSS スタイルを追加する

このセクションでは、カスタム CSS スタイルを追加してウィジェットの見た目をより魅力的にする方法を学びます。

1. ウィジェットのスタイル用に、assets ディレクトリ内に新しい css ディレクトリを作成します。
2. assets/css ディレクトリに widget.css ファイルを作成します。
   ウィジェット要素のスタイル設定には、セレクタ div.dashboard-widget-{widget id} を使用します。
   ウィジェット全体の CSS を設定するには、セレクタ form.dashboard-widget-{widget id} を使用します。

ui/modules/lesson_gauge_chart/assets/css/widget.css

```
div.dashboard-widget-lesson_gauge_chart {
    display: grid;
    grid-template-rows: 1fr;
    padding: 0;
}

div.dashboard-widget-lesson_gauge_chart .chart {
    display: grid;
    align-items: center;
    justify-items: center;
}

div.dashboard-widget-lesson_gauge_chart .chart canvas {
    background: white;
}

div.dashboard-widget-lesson_gauge_chart .description {
    padding-bottom: 8px;
    font-size: 1.750em;
    line-height: 1.2;
    text-align: center;
}

.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

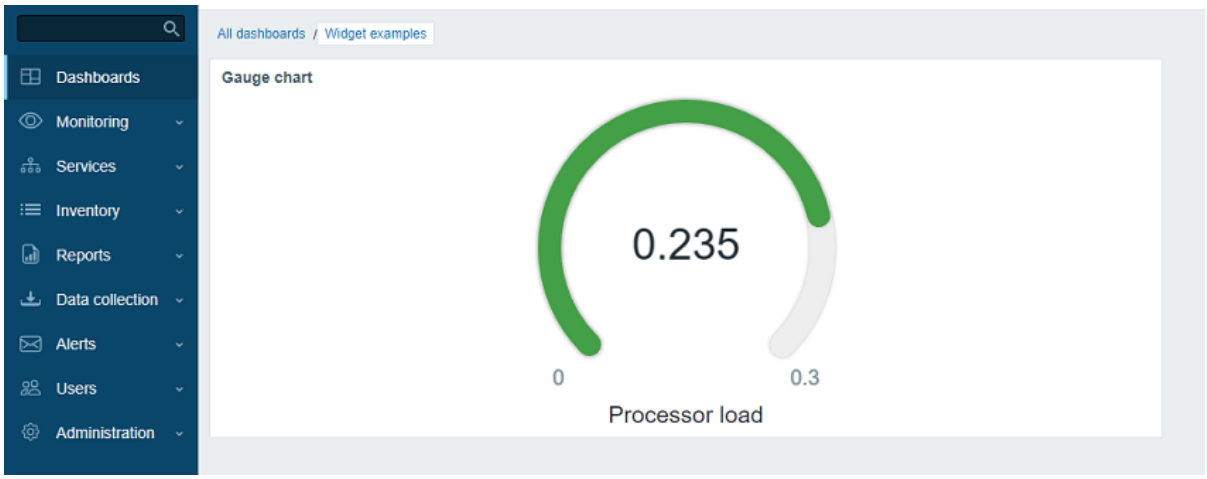
3. manifest.json を開き、assets/css セクションの配列に CSS ファイル名 (widget.css) を追加します。
   これにより、widget.css で定義された CSS スタイルがダッシュボードページとともに読み込まれるようになります。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
```

```
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. ダッシュボードページを更新して、ウィジェットの完成版を確認します。



例

このセクションでは、カスタムモジュールのベースとして使用できる、サンプルモジュールとウィジェットのファイルを提供します。

モジュールの使用方法：

1. ZIP アーカイブをダウンロードします。
2. Zabbix Web インターフェースをインストールの *modules* ディレクトリ内の別のディレクトリにコンテンツを解凍します (例: */zabbix/ui/modules*)。
3. Zabbix Web インターフェースにモジュールを登録します。

モジュールの例

- ホストグループ作成時に、設定されたユーザーグループに読み取り権限を付与します - hg_auto_perm.zip

ウィジェットの例

- 最小限のウィジェット - widget_min.zip
- CSS のみを使用した”Hello, world” ウィジェット - hello_world_css.zip
- JavaScript のみを使用した”Hello, world” ウィジェット - hello_world_js.zip
- PHP を使用した”Hello, world” ウィジェット - hello_world_php.zip

Note:
Zabbix ネイティブウィジェットも例として利用できます。

7.4 から **8.0** への変更　ZBXNEXT-10445 string ウィジェットフィールド の最大長が 2048 文字から 65535 文字に拡張されました。
Zabbix ネイティブウィジェット 内の特定の string フィールドには、独自の長さ制限も設定されるようになりました（例: setMaxLength() 関数で設定）。

Zabbix man ページ

Zabbix のコマンドラインツールおよびサービスのリファレンスドキュメントで、利用可能なコマンド、オプション、使用例を記載しています。

zabbix_agent2

```
ZABBIX_AGENT2(8)            システム管理者マニュアル            ZABBIX_AGENT2(8)

名前
       zabbix_agent2 - Zabbix エージェント 2

書式
       zabbix_agent2 [-c config-file]
       zabbix_agent2 [-c config-file] [-v] -p
       zabbix_agent2 [-c config-file] [-v] -t item-key
       zabbix_agent2 [-c config-file] -R runtime-option
       zabbix_agent2 [-c config-file] -T
       zabbix_agent2 -h
       zabbix_agent2 -V

説明
       zabbix_agent2 は、さまざまなサービスのパラメータを監視するための
       アプリケーションです。

オプション
       -c, --config config-file
              デフォルトの設定ファイルの代わりに、別の config-file を使用します。

       -R, --runtime-control runtime-option
              runtime-option に従って管理機能を実行します。

           実行時制御オプション
               userparameter_reload
                   設定ファイルからユーザーパラメータを再読み込みします

               log_level_increase
                   ログレベルを上げます

               log_level_decrease
                   ログレベルを下げます

               help
                   利用可能な実行時制御オプションを一覧表示します

               metrics
                   利用可能なメトリクスを一覧表示します
               version
                   バージョンを表示します
```

-p, --print
既知のアイテムを表示して終了します。各アイテムについて、一般的なデフォルト値が使用されるか、またはテスト用の特定のデフォルト値が提供されます。これらのデフォルト値は、アイテムキーのパラメータとして角括弧内に表示されます。返される値は角括弧で囲まれ、パイプ文字で区切られた返却値の型が先頭に付加されます。ユーザーパラメータの型は、エージェントが考えられるすべての返却値を判定できないため、常に t です。動作中として表示されるアイテムについても、権限や環境が異なる可能性があるため、実行中のエージェントデーモンに対して Zabbix サーバーまたは zabbix_get から問い合わせた場合に動作することは保証されません。返却値の型は次のとおりです。

d 小数部を持つ数値。

m サポートされていません。これは、ログ監視アイテムのようにアクティブモードでのみ動作するアイテムや、複数の収集値を必要とするアイテムを問い合わせたことが原因である可能性があります。権限の問題や不正なユーザーパラメータによっても、未サポート状態になる場合があります。

s テキスト。最大長の制限はありません。

t テキスト。s と同じです。

u 符号なし整数。

-t, --test item-key
単一のアイテムをテストして終了します。出力の説明については --print を参照してください。

-v, --verbose
アイテムのテスト時または既知のアイテムの表示時に、詳細出力を有効にします。

-T, --test-config
設定ファイルを検証して終了します。

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

ファイル
/usr/local/etc/zabbix_agent2.conf
Zabbix エージェント 2 設定ファイルのデフォルトの場所です（コンパイル時に変更されていない場合）。

関連項目
Documentation

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_AGENT2(8)

zabbix_agentd

ZABBIX_AGENTD(8) システム管理者マニュアル ZABBIX_AGENTD(8)

名前

zabbix_agentd - Zabbixエージェントデーモン

書式

```
zabbix_agentd [-c config-file]
zabbix_agentd [-c config-file] -p
zabbix_agentd [-c config-file] -t item-key
zabbix_agentd [-c config-file] -R runtime-option
zabbix_agentd [-c config-file] -T
zabbix_agentd -h
zabbix_agentd -V
```

説明

zabbix_agentd は、さまざまなサーバーパラメータを監視するためのデーモンです。

オプション

-c, --config config-file
デフォルトの設定ファイルの代わりに、別の config-file を使用します。

-f, --foreground
Zabbixエージェントをフォアグラウンドで実行します。

-R, --runtime-control runtime-option
runtime-option に従って管理機能を実行します。

ランタイム制御オプション
userparameter_reload
設定ファイルからユーザーパラメータを再読み込みします

log_level_increase[=target]
ログレベルを上げます。target が指定されていない場合は、すべてのプロセスに影響します

log_level_decrease[=target]
ログレベルを下げます。target が指定されていない場合は、すべてのプロセスに影響します

ログレベル制御の対象
process-type
指定した種類のすべてのプロセス（active checks、collector、listener）

process-type,N
プロセスの種類と番号（例: listener,3）
pid プロセス識別子。65535 まで指定できます。これより大きい値の場合は、target を "process-type,N" として指定してください

-p, --print
既知のアイテムを表示して終了します。各アイテムには、汎用のデフォルト値が使用されるか、テスト用の固有のデフォルト値が提供されます。これらのデフォルト値は、アイテムキーのパラメータとして角括弧内に表示されます。返される値は角括弧で囲まれ、返却値の型がパイプ文字で区切られて先頭に付加されます。ユーザーパラメータでは、エージェントは考えられるすべての返却値を判定できないため、型は常に t です。動作中として表示されるアイテムは、権限や環境が異なる可能性があるため、実行中のエージェントデーモンに対して Zabbixサーバー または zabbix_get から問い合わせた場合に動作することを保証するものではありません。返却値の型は次のとおりです。

d 小数部を持つ数値。

m サポートされていません。これは、ログ監視アイテムのようにアクティブモードでのみ動作するアイテムや、複数の収集値を必要

```
                とするアイテムを問い合わせたことが原因である可能性があります。権限の問題や不正なユーザーパラメータによっても、サポートされていない状態になることがあります。

            s   テキスト。最大長に制限はありません。

            t   テキスト。s と同じです。

            u   符号なし整数。

        -t, --test item-key
               単一のアイテムをテストして終了します。出力の説明については --print を参照してください。

        -T, --test-config
               設定ファイルを検証して終了します。

        -h, --help
               このヘルプを表示して終了します。

        -V, --version
               バージョン情報を出力して終了します。

ファイル
        /usr/local/etc/zabbix_agentd.conf
               Zabbixエージェント設定ファイルのデフォルトの場所（コンパイル時に変更されていない場合）。

関連項目
        Documentation

        zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                          2022-01-06                  ZABBIX_AGENTD(8)
```

zabbix_get

```
ZABBIX_GET(1)               一般コマンドマニュアル              ZABBIX_GET(1)

名前
        zabbix_get - Zabbix取得ユーティリティ

書式
        zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t timeout] -k item-key
        zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t timeout] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file] [--tls-agent-cert-issuer cert-issuer] [--tls-agent-cert-subject cert-subject] --tls-cert-file cert-file --tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k item-key
        zabbix_get -s host-name-or-IP [-p port-number] [-I IP-address] [-t timeout] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k item-key
        zabbix_get -h
        zabbix_get -V

説明
        zabbix_get は、Zabbixエージェントからデータを取得するためのコマンドラインユーティリティです。
```

オプション

-s, --host host-name-or-IP
ホストのホスト名またはIPアドレスを指定します。

-p, --port port-number
ホスト上で動作しているエージェントのポート番号を指定します。デフォルトは10050です。

-I, --source-address IP-address
送信元IPアドレスを指定します。

-t, --timeout seconds
タイムアウトを指定します。有効範囲: 1～600秒（デフォルト: 30）

-k, --key item-key
値を取得するアイテムのキーを指定します。

-P, --protocol value
エージェントとの通信に使用するプロトコル。値:

auto	JSONプロトコルを使用して接続し、失敗した場合はプレーンテキストプロトコルで再試行します（デフォルト）
json	JSONプロトコルを使用して接続します
plaintext	アイテムキーのみを送信するプレーンテキストプロトコルを使用して接続します（6.4.x以前のリリース）

--tls-connect value
エージェントへの接続方法。値:

unencrypted	暗号化なしで接続します（デフォルト）
psk	TLSおよび事前共有鍵を使用して接続します
cert	TLSおよび証明書を使用して接続します

--tls-ca-file CA-file
ピア証明書の検証用に、最上位CA証明書を含むファイルのフルパスを指定します。

--tls-crl-file CRL-file
失効した証明書を含むファイルのフルパスを指定します。

--tls-agent-cert-issuer cert-issuer
許可するエージェント証明書の発行者。

--tls-agent-cert-subject cert-subject
許可するエージェント証明書のサブジェクト。

--tls-cert-file cert-file
証明書または証明書チェーンを含むファイルのフルパスを指定します。

--tls-key-file key-file
秘密鍵を含むファイルのフルパスを指定します。

--tls-psk-identity PSK-identity
PSK-identity文字列。

--tls-psk-file PSK-file
事前共有鍵を含むファイルのフルパスを指定します。

--tls-cipher13 cipher-string
TLS 1.3向けの、OpenSSL 1.1.1以降用の暗号スイート文字列。デフォルトの暗号スイート選択基準を上書きします。OpenSSLのバージョンが1.1.1未満の場合、このオプションは使用できません。

--tls-cipher cipher-string
GnuTLS優先順位文字列（TLS 1.2以降用）またはOpenSSL暗号文字列（TLS 1.2のみ）。デフォルトの暗号スイート選択基準を上書きします。

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

終了ステータス
zabbix_get が Zabbixエージェントから値の取得に成功した場合、終了ステータスは0です。値の取得に失敗した場合、終了ステータスは1です。

例
```
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
--tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file
--tls-agent-cert-issuer "CN=Signing CA,OU=IT operations,O=Example
Corp,DC=example,DC=com" --tls-agent-cert-subject "CN=server1,OU=IT op-
erations,O=Example Corp,DC=example,DC=com" --tls-cert-file /home/zab-
bix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]"
--tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd"
--tls-psk-file /home/zabbix/zabbix_agentd.psk
```

関連項目
ドキュメント

zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_GET(1)

zabbix_js

ZABBIX_JS(1) 一般コマンドマニュアル ZABBIX_JS(1)

名前
zabbix_js - Zabbix JSユーティリティ

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

説明
zabbix_js は、組み込みスクリプトのテストに使用できるコマンドラインユーティリティです。

オプション
-s, --script script-file
実行するスクリプトのファイル名を指定します。ファイル名として '-' を指定した場合、スクリプトは stdin から読み込まれます。

-p, --param input-param
スクリプトに渡す入力パラメータを指定します。パラメータは文字列としてそのまま渡され、スクリプト内では 'value' 変数を介して利用できます。-i とは同時に指定できません。

-i, --input input-file
スクリプトに渡す入力パラメータを含むファイルを指定します。ファイルの内容は文字列としてそのまま渡され、スクリプト内では 'value' 変数を介して利用できます。ファイル名として '-' を指定した場合、入力は stdin から読み込まれます。-p とは同時に指定できません。

-w, --webdriver url
webdriver のURLを指定します。

-l, --loglevel log-level
ログレベルを指定します（範囲: 0-5）。

-t, --timeout timeout
タイムアウトを秒単位で指定します。有効な範囲: 1～600秒（デフォルト: 10）

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

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

関連項目
Documentation

zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_JS(1)

zabbix_proxy

ZABBIX_PROXY(8) System Manager's Manual ZABBIX_PROXY(8)

名前
zabbix_proxy - Zabbix プロキシデーモン

書式
zabbix_proxy [-c config-file]
zabbix_proxy [-c config-file] -R runtime-option
zabbix_proxy [-c config-file] -T
zabbix_proxy -h
zabbix_proxy -V

説明
zabbix_proxy は、監視データをデバイスから収集し、Zabbix サーバーに送信するデーモンです。

オプション
-c, --config config-file
デフォルトの設定ファイルの代わりに、別の config-file を使用します。

-f, --foreground
Zabbix プロキシをフォアグラウンドで実行します。

-R, --runtime-control runtime-option
runtime-option に従って管理機能を実行します。

ランタイム制御オプション
config_cache_reload
設定キャッシュを再読み込みします。現在キャッシュを読み込み中の場合は無視されます。アク

snmp_cache_reload
SNMP キャッシュを再読み込みします。

housekeeper_execute
housekeeper を実行します。現在 housekeeper が実行中の場合は無視されます。

diaginfo[=section]
指定した section の内部診断情報をログに出力します。section には historycache、preproces

prof_enable[=target]
プロファイリングを有効にします。target が指定されていない場合は、すべてのプロセスに影響

prof_disable[=target]
プロファイリングを無効にします。target が指定されていない場合は、すべてのプロセスに影響

log_level_increase[=target]
ログレベルを上げます。target が指定されていない場合は、すべてのプロセスに影響します。

log_level_decrease[=target]
ログレベルを下げます。target が指定されていない場合は、すべてのプロセスに影響します。

ログレベル制御の対象
process-type
指定した種類のすべてのプロセス（availability manager、browser poller、configuration syn

process-type,N
プロセスの種類と番号（例: poller,3）

pid プロセス識別子

プロファイリング制御の対象
process-type
指定した種類のすべてのプロセス（availability manager、browser poller、configuration syn

process-type,N
プロセスの種類と番号（例: history syncer,1）

pid プロセス識別子

scope
(rwlock, mutex, processing) を process-type と組み合わせて使用できます（例: history syn

-T, --test-config
設定ファイルを検証して終了します。

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

ファイル
/usr/local/etc/zabbix_proxy.conf

Zabbix プロキシ設定ファイルのデフォルトの場所（コンパイル時に変更されていない場合）。

関連項目

Documentation

zabbix_agentd(8), zabbix_get(1), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_PROXY(8)

zabbix_sender

ZABBIX_SENDER(1) 一般コマンドマニュアル ZABBIX_SENDER(1)

名前

zabbix_sender - Zabbix senderユーティリティ

書式

```
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host -k key -o value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file]
[--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject
cert-subject] --tls-cert-file cert-file --tls-key-file key-file
[--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k key -o
value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
[--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
[--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
--tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
host --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k
key -o value
zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
host] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string]
[-T] [-N] [-r] -i input-file
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
```

```
pher cipher-string] -k key -o value
zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
[-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
pher cipher-string] [-T] [-N] [-r] -i input-file
zabbix_sender -h
zabbix_sender -V
```

説明

zabbix_senderは、監視データをZabbixサーバーまたはプロキシに送信する
コマンドラインユーティリティです。Zabbixサーバー上で、対応するキーを
持つZabbix trapperタイプのアイテムを作成する必要があります。このアイテム
の「許可されたホスト」フィールドで指定されたホストからの値のみが受け入れ
られることに注意してください。

オプション

-c, --config config-file
: config-fileを使用します。Zabbix senderは、エージェントの設定ファイル
からサーバーの詳細を読み取ります。デフォルトでは、Zabbix senderは
設定ファイルを読み取りません。サポートされるパラメータは、Hostname、
ServerActive、SourceIP、TLSConnect、TLSCAFile、TLSCRLFile、
TLSServerCertIssuer、TLSServerCertSubject、TLSCertFile、TLSKeyFile、
TLSPSKIdentity、TLSPSKFileのみです。HostnameItemパラメータで定義された
Hostnameは取得されません。この場合、ホスト名をコマンドラインで指定する
必要があります（-sオプションを参照）。エージェントのServerActive設定
パラメータで定義されたすべてのアドレスがデータ送信に使用されます。
バッチデータの送信がいずれかのアドレスで失敗した場合、それ以降の
バッチはそのアドレスには送信されません。

-z, --zabbix-server server
: ZabbixサーバーのホストまたはIPアドレス。ホストがプロキシによって
監視されている場合は、プロキシのホスト名またはIPアドレスを使用する
必要があります。--configと併用した場合、エージェントの設定ファイル
で指定されたServerActiveパラメータのエントリを上書きします。

-p, --port port
: サーバー上で実行されているZabbixサーバーのtrapperのポート番号を
指定します。デフォルトは10051です。--configと併用した場合、エージェント
の設定ファイルで指定されたServerActiveパラメータのポートエントリを
上書きします。

-I, --source-address IP-address
: 送信元IPアドレスを指定します。--configと併用した場合、エージェントの
設定ファイルで指定されたSourceIPパラメータを上書きします。

-t, --timeout seconds
: タイムアウトを指定します。有効範囲: 1～300秒（デフォルト: 60）

-s, --host host
: アイテムが属するホスト名を指定します（ZabbixのWebインターフェースに
登録されている名前）。ホストのIPアドレスおよびDNS名は使用できません。
--configと併用した場合、エージェントの設定ファイルで指定された
Hostnameパラメータを上書きします。

-k, --key key
: 値の送信先となるアイテムキーを指定します。

-o, --value value
: アイテムの値を指定します。

-i, --input-file input-file
: 入力ファイルから値を読み込みます。標準入力から値を読み取るには、
<input-file>として-を指定します。ファイルの各行には、空白で区切られた

<hostname> <key> <value>が含まれます。各値はそれぞれ別の行に指定する必要があります。各行には、空白で区切られた3つのエントリ <hostname> <key> <value>を含める必要があります。ここで「hostname」はZabbixのWebインターフェースに登録されている監視対象ホストの名前、「key」は対象アイテムのキー、「value」は送信する値です。エージェント設定ファイルまたは--host引数のホスト名を使用するには、<hostname>として-を指定します。

入力ファイルの行の例:

```
"Linux DB3" db.connections 43
```

値のタイプは、ZabbixのWebインターフェースのアイテム設定で正しく設定する必要があります。Zabbix senderは1つの接続で最大250個の値を送信します。入力ファイルの内容はUTF-8エンコーディングでなければなりません。入力ファイルのすべての値は、上から下へ順番に送信されます。エントリは次のルールに従ってフォーマットする必要があります。

- 引用符付きエントリと引用符なしエントリを使用できます。
- 二重引用符が引用文字です。
- 空白を含むエントリは引用符で囲む必要があります。
- 引用符付きエントリ内の二重引用符およびバックスラッシュ文字は、バックスラッシュでエスケープする必要があります。
- 引用符なしエントリではエスケープはサポートされません。
- 引用符付き文字列では、改行エスケープシーケンス（\n）がサポートされます。
- エントリ末尾の改行エスケープシーケンスは削除されます。

-T, --with-timestamps
: このオプションは--input-fileオプションとの併用時のみ使用できます。

入力ファイルの各行には、空白で区切られた4つのエントリ <hostname> <key> <timestamp> <value>を含める必要があります。タイムスタンプはUnixタイムスタンプ形式で指定する必要があります。対象アイテムを参照するトリガーがある場合、すべてのタイムスタンプは昇順でなければなりません。そうでない場合、イベントの計算が正しく行われません。

入力ファイルの行の例:

```
"Linux DB3" db.connections 1429533600 43
```

詳細については、--input-fileオプションを参照してください。

「データなし」タイプのメンテナンス期間中のホストにタイムスタンプ付きの値を送信した場合、その値は破棄されます。ただし、期限切れのメンテナンス期間にタイムスタンプ付きの値を送信することは可能で、その値は受け入れられます。

-N, --with-ns
: このオプションは--with-timestampsオプションとの併用時のみ使用できます。

入力ファイルの各行には、空白で区切られた5つのエントリ <hostname> <key> <timestamp> <ns> <value>を含める必要があります。

入力ファイルの行の例:

```
"Linux DB3" db.connections 1429533600 7402561 43
```

詳細については、--input-fileオプションを参照してください。

-r, --real-time
値を受信するとすぐに1つずつ送信します。標準入力から読み取る場合に使用できます。

--tls-connect value
サーバーまたはプロキシへの接続方法。値:

unencrypted
暗号化せずに接続（デフォルト）

psk
TLSと事前共有鍵を使用して接続

cert
TLSと証明書を使用して接続

--tls-ca-file CA-file
ピア証明書の検証に使用する、最上位CA証明書を含むファイルの完全なパス名。

--tls-crl-file CRL-file
失効した証明書を含むファイルの完全なパス名。

--tls-server-cert-issuer cert-issuer
許可するサーバー証明書の発行者。

--tls-server-cert-subject cert-subject
許可するサーバー証明書のサブジェクト。

--tls-cert-file cert-file
証明書または証明書チェーンを含むファイルの完全なパス名。

--tls-key-file key-file
秘密鍵を含むファイルの完全なパス名。

--tls-psk-identity PSK-identity
PSK-identity文字列。

--tls-psk-file PSK-file
事前共有鍵を含むファイルの完全なパス名。

--tls-cipher13 cipher-string
TLS 1.3用のOpenSSL 1.1.1以降の暗号文字列。デフォルトの暗号スイート選択基準を上書きします。OpenSSLのバージョンが1.1.1未満の場合、このオプションは使用できません。

--tls-cipher cipher-string
GnuTLSの優先度文字列（TLS 1.2以降）またはOpenSSLの暗号文字列（TLS 1.2のみ）。デフォルトの暗号スイート選択基準を上書きします。

-v, --verbose
詳細モード。詳細を表示するには-vvを指定します。

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

終了ステータス
値が送信され、すべての値がサーバーによって正常に処理された場合、終了ステータスは0です。データが送信されたものの、少なくとも1つの値の処理に

失敗した場合、終了ステータスは2です。データの送信に失敗した場合、終了ステータスは1です。

例

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

監視対象ホストのmysql.queriesアイテムの値として342.45を送信します。エージェント設定ファイルで定義された監視対象ホストとZabbixサーバーを使用します。

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host" -k mysql.queries -o 342.45

エージェント設定ファイルで定義されたZabbixサーバーを使用して、Monitored Hostホストのmysql.queriesアイテムの値として342.45を送信します。

zabbix_sender -z 192.168.1.113 -i data_values.txt

data_values.txtファイルの値を、IPアドレス192.168.1.113のZabbixサーバーに送信します。ホスト名とキーはファイル内で定義されています。

echo "- hw.serial.number 1287872261 SQ4321ASDF" | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

コマンドラインからタイムスタンプ付きの値を、エージェント設定ファイルで指定されたZabbixサーバーに送信します。入力データのダッシュは、同じ設定ファイルのホスト名も使用することを示します。

echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113 -p 10000 -i -

コマンドラインから、IPアドレス192.168.1.113、ポート10000のZabbixサーバーにアイテムの空の値を送信します。空の値は空の二重引用符で示す必要があります。

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o 342.45 --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-cert-file /home/zabbix/zabbix_agentd.crt --tls-key-file /home/zabbix/zabbix_agentd.key

TLSと証明書を使用して、Monitored Hostホストのmysql.queriesアイテムの値として342.45をIPアドレス192.168.1.113のサーバーに送信します。

zabbix_sender -z 192.168.1.113 -s "Monitored Host" -k mysql.queries -o 342.45 --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk

TLSと事前共有鍵（PSK）を使用して、Monitored Hostホストのmysql.queriesアイテムの値として342.45をIPアドレス192.168.1.113のサーバーに送信します。

関連項目

ドキュメント

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_SENDER(1)

zabbix_server

ZABBIX_SERVER(8) System Manager's Manual ZABBIX_SERVER(8)

名前

zabbix_server - Zabbix サーバーデーモン

概要

zabbix_server [-c config-file]

```
zabbix_server [-c config-file] -R runtime-option
zabbix_server [-c config-file] -T
zabbix_server -h
zabbix_server -V
```

説明

zabbix_server は Zabbix ソフトウェアの中核となるデーモンです。

オプション

-c, --config config-file
デフォルトの設定ファイルの代わりに、別の config-file を使用します。

-f, --foreground
Zabbix サーバーをフォアグラウンドで実行します。

-R, --runtime-control runtime-option
runtime-option に従って管理機能を実行します。

ランタイム制御オプション

config_cache_reload
設定キャッシュを再読み込みします。現在キャッシュを読み込み中の場合は無視されます。PID

snmp_cache_reload
SNMP キャッシュを再読み込みします。

housekeeper_execute
housekeeper を実行します。現在 housekeeper が実行中の場合は無視されます。

trigger_housekeeper_execute
trigger housekeeper を実行します（削除されたトリガーに対する障害を削除します）。現在 t

secrets_reload
Vault からシークレットを再読み込みします。設定キャッシュまたはシークレットが現在読み込

diaginfo[=section]
指定した section の内部診断情報をログに出力します。section には historycache、preproces

prof_enable[=target]
プロファイリングを有効にします。target が指定されていない場合は、すべてのプロセスに影

prof_disable[=target]
プロファイリングを無効にします。target が指定されていない場合は、すべてのプロセスに影

ha_status
高可用性クラスタの状態を表示します。アクティブノードでのみ実行できます。

ha_remove_node
ha_remove_node[=name|ID] ha_status に一覧表示される名前または ID によって高可用性ノート

ha_set_failover_delay[=delay]
高可用性のフェイルオーバー遅延を設定します。delay には時間の接尾辞を指定できます。アク

log_level_increase[=target]
ログレベルを上げます。target が指定されていない場合は、すべてのプロセスに影響します。

log_level_decrease[=target]
ログレベルを下げます。target が指定されていない場合は、すべてのプロセスに影響します。

ログレベル制御の対象

process-type
指定した type のすべてのプロセス（alerter, alert manager,
alert syncer, availability manager, browser poller, config-
uration syncer, configuration syncer worker, connector man-

ager, connector worker, discovery manager, escalator, ha m-
anager, history poller, history syncer, housekeeper, h-
ttp poller, http agent poller, icmp pinger, internal poll-
er, ipmi manager, ipmi poller, java poller, lld manager,
lld worker, odbc poller, poller, エージェント poller, preprocesi-
ng manager, preprocessing worker, proxy poller, proxy group
manager, report manager, report writer, self-monitoring,
service manager, snmp poller, snmp trapper, supervisor,
task manager, timer, trapper, trigger housekeeper, unreac-
hable poller, vmware collector)

process-type,N
プロセスの type と番号 (例: poller,3)

pid プロセス識別子

プロファイリング制御の対象
process-type
指定した type のすべてのプロセス (alerter, alert manager,
alert syncer, availability manager, browser poller, config-
uration syncer, configuration syncer worker, connector man-
ager, connector worker, discovery manager, escalator, ha m-
anager, history poller, history syncer, housekeeper,
http poller, http agent poller, icmp pinger, internal po-
ller, ipmi manager, ipmi poller, java poller, lld manager,
lld worker, odbc poller, poller, エージェント poller, preproc-
essing manager, proxy poller, proxy group manager, repo-
rt manager, report writer, self-monitoring, service mana-
ger, snmp poller, snmp trapper, task manager, timer, trap-
per, trigger housekeeper, unreachable poller, vmware collec-
tor)

process-type,N
プロセスの type と番号 (例: history syncer,1)

pid プロセス識別子

scope
(rwlock, mutex, processing) は process-type とともに使用できます
(例: history syncer,1,processing)

-T, --test-config
設定ファイルを検証して終了します。

-h, --help
このヘルプを表示して終了します。

-V, --version
バージョン情報を出力して終了します。

ファイル
/usr/local/etc/zabbix_server.conf
Zabbix サーバー設定ファイルのデフォルトの場所です (コンパイル時に変更されていない場合)。

関連項目
Documentation

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1),
zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)       システム管理者マニュアル       ZABBIX_WEB_SERVICE(8)

名前
       zabbix_web_service - Zabbix Webサービス

書式
       zabbix_web_service [-c config-file]
       zabbix_web_service [-c config-file] -T
       zabbix_web_service -h
       zabbix_web_service -V

説明
       zabbix_web_service は、ZabbixコンポーネントにWebサービスを提供するための
       アプリケーションです。

オプション
       -c, --config config-file
              デフォルトの設定ファイルの代わりに、別の config-file を使用します。

       -T, --test-config
              設定ファイルを検証して終了します。

       -h, --help
              このヘルプを表示して終了します。

       -V, --version
              バージョン情報を出力して終了します。

ファイル
       /usr/local/etc/zabbix_web_service.conf
              Zabbix Webサービス設定ファイルのデフォルトの場所
              （コンパイル時に変更されていない場合）。

関連項目
       Documentation

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06             ZABBIX_WEB_SERVICE(8)
```

著作権表示

Zabbix ドキュメントは AGPL-3.0 ライセンスの下では配布されません。Zabbix ドキュメントの利用は以下の条件に従うものとします。

このドキュメントの印刷物を作成できるのは、個人的な利用に限ります。実際の内容がいかなる方法でも変更または編集されない限り、他の形式への変換は許可されます。このドキュメントをいかなる形態または媒体でも公開または配布してはなりません。ただし、Zabbix が配布する方法（すなわち、Zabbix の Web サイトでダウンロードできるように電子的に配布する方法）と同様の方法でドキュメントを配布する場合、または USB や同様の媒体で配布する場合は除きます。ただし、その場合は、ドキュメントがソフトウェアと同じ媒体で一緒に配布されることが条件となります。印刷物の配布や、このドキュメントの全部または一部を他の出版物で利用するなど、その他の利用には、Zabbix の正式な代表者から事前に書面による同意を得る必要があります。Zabbix は、上記で明示的に付与されていない本ドキュメントに関するすべての権利を留保します。