Documentation 7.4

ZABBIX

02.04.2026

Contents

ユーザーマニュアル **6**

Zabbix ドキュメントへようこそ **6**
- Zabbix を使い始める . . . 6
- Zabbix Cloud . . . 6
- 開発者センター . . . 6
- コミュニティとその他のリソース . . . 6
- 1. はじめに . . . 6
  - 1 マニュアルの構成 . . . 7
  - 2 Zabbix とは . . . 7
  - 3 Zabbix の機能 . . . 8
  - 4 Zabbix の概要 . . . 9
  - 5 Zabbix 7.4.0 の新機能 . . . 10
  - 6 Zabbix 7.4.x の新機能 . . . 19
- 2. 定義 . . . 21
- 3. Zabbix のプロセス . . . 23
  - 1 Zabbix サーバー . . . 23
  - 2 Zabbix エージェント . . . 31
  - 3 Zabbix エージェント 2 . . . 35
  - 4 Zabbix プロキシ . . . 39
  - 5 Zabbix Java ゲートウェイ . . . 43
  - 6 zabbix_sender コマンド . . . 47
  - 7 zabbix_get コマンド . . . 48
  - 8 zabbix_js コマンド . . . 49
  - 9 Zabbix Web サービス . . . 50
- 4. インストール . . . 50
  - 1 Zabbix の入手 . . . 50
  - 2 要件 . . . 51
  - 3 ソースコードからのインストール . . . 60
  - 4 パッケージインストール . . . 76
  - 5 コンテナからのインストール . . . 89
  - 6 Web インターフェースのインストール . . . 98
  - 7 アップグレード手順 . . . 105
  - 8 既知の問題 . . . 118
  - 9 テンプレートの変更 . . . 129
  - 10 7.4.0 へアップグレード時の注意点 . . . 130
  - 11 7.4.x のアップグレードに関する注意事項 . . . 132
- 5. クイックスタート . . . 133
  - 1 ログインとユーザー設定 . . . 133
  - 2 新規ホスト . . . 138
  - 3 新規アイテム . . . 140
  - 4 新規トリガー . . . 141
  - 5 障害発生の通知 . . . 143
  - 6 新規テンプレート . . . 147
- 6. Zabbix アプライアンス . . . 149
- 7. 設定 . . . 152
  - 1 ホストとホストグループ . . . 162
  - 2 アイテム . . . 180
  - 3 トリガー . . . 392
  - 4 イベント . . . 411
  - 5 イベント相関 . . . 414

6 タグ付け . . . 420
7 視覚化 . . . 423
8 テンプレートとテンプレートグループ . . . 449
9 すぐに使えるテンプレート . . . 449
10 イベント時の通知 . . . 462
11 マクロ . . . 512
12 ユーザーとユーザーグループ . . . 527
13 secrets の保存 . . . 536
14 スケジュールレポート . . . 543
15 データのエクスポート . . . 547
8. サービス監視 . . . 553
1 サービスツリー . . . 554
2 SLA . . . 558
3 セットアップ例 . . . 559
9. Web 監視 . . . 563
1 Web 監視アイテム . . . 572
2 実際のシナリオ . . . 573
10. 仮想マシン監視 . . . 580
1 VMware 監視アイテムキー . . . 582
2 仮想マシンディスカバリキーフィールド . . . 600
3 VMware アイテムの JSON の例 . . . 604
4 VMware 監視設定例 . . . 608
11. メンテナンス . . . 613
12. 正規表現 . . . 618
13. 障害イベントの確認 . . . 624
1 障害の抑制 . . . 627
14. 設定のエクスポート/インポート . . . 628
1 テンプレートグループ . . . 629
2 ホストグループ . . . 630
3 テンプレート . . . 630
4 ホスト . . . 647
5 ネットワークマップ . . . 663
6 メディアタイプ . . . 671
15. ディスカバリ . . . 678
1 ネットワークディスカバリ . . . 678
2 アクティブエージェントの自動登録 . . . 687
3 ローレベルディスカバリ . . . 691
16. 分散監視 . . . 745
1 プロキシ . . . 746
17. 暗号化 . . . 755
1 証明書の利用 . . . 762
2 事前共有鍵の使用 . . . 768
3 トラブルシューティング . . . 771
18. Web インターフェース . . . 773
1 メニュー . . . 774
2 Web インターフェースセクション . . . 783
3 ユーザー設定 . . . 1012
4 グローバル検索 . . . 1017
5 フロントエンドメンテナンスモード . . . 1019
6 ページパラメーター . . . 1020
7 定義 . . . 1023
8 独自テーマ作成 . . . 1023
9 デバッグモード . . . 1024
10 Zabbix 用 Cookie . . . 1025
11 タイムゾーン . . . 1026
12 パスワードのリセット . . . 1027
13 期間セレクター . . . 1027
19. ベストプラクティス . . . 1028
1 セキュリティのベストプラクティス . . . 1029
2 設定のベストプラクティス . . . 1037
20. API . . . 1038
メソッドリファレンス . . . 1043
付録 1. 参考文献の解説 . . . 1860

付録 2. 7.2 から 7.4 での変更点 . . . 1866
付録 3. 7.4 での変更点 . . . 1868
21. 拡張機能 . . . 1868
1 ローダブルモジュール . . . 1870
2 プラグイン . . . 1879
3 フロントエンドモジュール . . . 1887
22. 付録 . . . 1888
1 インストールとセットアップ . . . 1888
2 プロセスの設定 . . . 1946
3 プロトコル . . . 2026
4 アイテム . . . 2051
5 サポートされている関数 . . . 2086
6 マクロ . . . 2117
7 単位記号 . . . 2155
8 日時フォーマット . . . 2157
9 コマンドの実行 . . . 2157
10 バージョン間の互換性 . . . 2158
12 Windows 向け Zabbix sender ダイナミックリンクライブラリ . . . 2159
13 サービス監視のアップグレード . . . 2160
14 その他の問題 . . . 2161
15 エージェント vs エージェント 2 の比較 . . . 2161
16 エスケープの例 . . . 2162
23. クイックリファレンスガイド . . . 2164
概要 . . . 2164
1 Zabbix エージェントで Linux を監視する . . . 2164
2 Zabbix エージェントで Windows を監視する . . . 2168
3 HTTP 経由で Apache を監視する . . . 2172
4 Zabbix エージェント 2 で MySQL を監視する . . . 2176
5 Zabbix で VMware を監視する . . . 2184
6 Zabbix でネットワークトラフィックを監視する . . . 2188
7 アクティブチェックを使用してネットワークトラフィックを監視する . . . 2191
8 ブラウザアイテムでウェブサイトの監視 . . . 2194
9 Zabbix エージェント 2(パッシブ) で Web サイトの証明書を監視する . . . 2199
10 Zabbix でネットワークスイッチやルーターを監視する . . . 2203
11 アクティブチェックを使用して Windows イベントログを監視する . . . 2210

Zabbix Cloud **2214**

Zabbix Cloud **2214**
Zabbix Cloud のセットアップと管理 . . . 2214
Zabbix Cloud について学ぶ . . . 2214
クラウドに Zabbix をデプロイする . . . 2214
ノード設定 . . . 2218
概要 . . . 2218
概要タブ . . . 2219
アクセスフィルタータブ . . . 2220
暗号化タブ . . . 2221
バックアップタブ . . . 2222
履歴タブ . . . 2222
メンテナンスタブ . . . 2223
アップグレードタブ . . . 2224
ユーザーの追加 . . . 2226
はじめに . . . 2226
組織へのユーザーの追加 . . . 2226
ノードへのユーザーの追加 . . . 2227
Zabbix Cloud の主な違い . . . 2228

開発者センター **2229**

開発者センター **2229**
Zabbix 上で構築する . . . 2230
モジュール . . . 2230
モジュールファイルの構造 . . . 2230
ウィジェット . . . 2238

チュートリアル . . . 2246
例 . . . 2272
Zabbix 用 Python ライブラリ . . . 2272
インストール . . . 2273
クイックスタートガイド . . . 2273
Zabbix API を使用する . . . 2277
Zabbix エージェントからデータを収集する . . . 2280
Zabbix サーバーまたはプロキシにデータを送信する . . . 2281
デバッグログ . . . 2285
プラグイン . . . 2286
例 . . . 2288
プラグインの作成 (チュートリアル) . . . 2288
プラグインインターフェース . . . 2293
拡張機能開発に関する変更 . . . 2293

Zabbix man ページ **2293**
zabbix_agent2 . . . 2293
名前 . . . 2294
概要 . . . 2294
説明 . . . 2294
オプション . . . 2294
ファイル . . . 2295
参照 . . . 2295
目次 . . . 2295
zabbix_agentd . . . 2295
名前 . . . 2295
概要 . . . 2296
説明 . . . 2296
オプション . . . 2296
ファイル . . . 2297
参照 . . . 2297
目次 . . . 2297
zabbix_get . . . 2297
名前 . . . 2297
概要 . . . 2298
説明 . . . 2298
オプション . . . 2298
例 . . . 2299
参照 . . . 2299
目次 . . . 2299
zabbix_js . . . 2299
名前 . . . 2300
概要 . . . 2300
説明 . . . 2300
オプション . . . 2300
例 . . . 2300
参照 . . . 2300
目次 . . . 2300
zabbix_proxy . . . 2301
名前 . . . 2301
概要 . . . 2301
説明 . . . 2301
オプション . . . 2301
ファイル . . . 2302
参照 . . . 2302
目次 . . . 2302
zabbix_sender . . . 2303
名前 . . . 2303
概要 . . . 2303
説明 . . . 2303
オプション . . . 2303
終了ステータス . . . 2305
例 . . . 2305

参照 . . . 2306
目次 . . . 2306
zabbix_server . . . 2306
名前 . . . 2307
概要 . . . 2307
説明 . . . 2307
オプション . . . 2307
ファイル . . . 2308
参照 . . . 2308
目次 . . . 2309
zabbix_web_service . . . 2309
名前 . . . 2309
概要 . . . 2309
説明 . . . 2309
オプション . . . 2309
ファイル . . . 2309
参照 . . . 2310
目次 . . . 2310

著作権に関する通知 **2310**

ユーザーマニュアル

Zabbix ドキュメントへようこそ

基本的なセットアップから高度な設定まで、Zabbix 監視を利用するための頼れるリソースです。このマニュアルでは、Zabbix のインストール、設定、運用に必要なすべてを網羅しています。

Zabbix 7.4 の新機能

Zabbix を使い始める

インストール　お使いのプラットフォームに Zabbix をインストールするための手順を、さまざまなオペレーティングシステムおよびデータベース構成に対応して段階的に説明します。

要件　Zabbix を正常に導入するための環境準備に役立つ、サポート対象プラットフォームおよびソフトウェア前提条件の一覧です。

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

Zabbix Cloud

Zabbix Cloud の利用を開始する　初期設定や監視対象デバイスの接続を含め、Zabbix Cloud インスタンスをすばやく稼働させるためのガイド付き手順です。

Zabbix Cloud とオンプレミスの比較　2 つの導入形態における管理、スケーラビリティ、メンテナンスの違いを比較し、どちらがあなたのワークフローを最もシームレスにサポートできるかを確認してください。

Zabbix Cloud を確認する　Zabbix Cloud の概要、主な機能、利点、サブスクリプションに含まれる内容を確認できます。セットアップや使用方法に関する一般的な質問については、FAQ を参照してください。

開発者センター

Web インターフェースモジュール　特定のユースケースに合わせて Zabbix Web インターフェースを拡張または変更するカスタムモジュールを構築するためのガイドとリファレンスです。

ウィジェット　ウィジェットの構造とロジックの内訳、およびニーズに合わせてカスタムダッシュボード要素を作成するための手順を説明します。

プラグイン　機能を拡張したり外部システムと統合したりするための、Zabbix プラグインの開発および管理方法の概要です。

コミュニティとその他のリソース

Zabbix フォーラム　あらゆるレベルの Zabbix の知識を持つユーザーが、アイデアを交換し、解決策を見つけ、経験を共有できる場です。

Zabbix ブログ　Zabbix チームとコミュニティの貢献者が厳選したニュース、チュートリアル、導入事例。

ハウツー動画　Zabbix を最大限に活用できるよう、デモンストレーション、ディスカッション、ヒントを紹介する動画ライブラリです。

1. はじめに

サイドバーから、参照したい項目を選択してください。

1 マニュアルの構成

構成

このマニュアルは、知りたい情報にすぐにたどり着けるようにするため、セクションとサブセクションに分かれています。

それぞれのセクションに移動するときには、サブセクションと個々のページに何が書かれているか、全体の内容を表示するために、セクションフォルダーを開いてください。

ユーザーが関係する情報を見落とさないように、関連するコンテンツ間のクロスリンクができるだけ多くはられています。

セクション

Introduction では、現在の Zabbix ソフトウェアに関する一般的な情報を提供しています。このセクションを読むことで、Zabbix を選ぶべき十分な理由を理解できるはずです。

Zabbix definitions では Zabbix で使用される用語を説明し、Zabbix processes では Zabbix コンポーネントの詳細を提供しています。

Installation およびQuickstart セクションは、Zabbix の利用を開始する際に役立ちます。Zabbix appliance は、Zabbix の使用感をすばやく試すための別の方法です。

Configuration は、このマニュアルの中でも最大級かつ最も重要なセクションの 1 つです。ここには、ホストの設定から重要なデータの収集と表示、さらに問題発生時に実行する通知やリモートコマンドの設定まで、環境を監視するために Zabbix をどのように設定するかについての重要なアドバイスが数多く含まれています。

Service monitoring では、監視環境を高いレベルで概観するために Zabbix を使用する方法を詳しく説明しています。

Web monitoring では、Web サイトの可用性を監視する方法を学ぶのに役立ちます。

Virtual machine monitoring では、VMware 環境の監視を設定するための手順を紹介しています。

Maintenance、Regular expressions、Problem acknowledgment、およびConfiguration export/import は、Zabbix ソフトウェアのこれらのさまざまな機能の使い方を説明する追加のセクションです。

Discovery には、ネットワークデバイス、アクティブエージェント、ファイルシステム、ネットワークインターフェースなどの自動検出を設定するための手順が含まれています。

Distributed monitoring では、より大規模で複雑な環境において Zabbix を利用する可能性を扱っています。

Encryption では、Zabbix コンポーネント間の通信を暗号化する方法について説明しています。

Web interface には、Zabbix の Web インターフェースを使用するための固有の情報が含まれています。

Best practices には、最適かつ安全なユーザー体験のための有用なヒントが含まれています。

API セクションでは、Zabbix API の利用に関する詳細を紹介しています。

Extensions は、Zabbix を拡張するための方法をまとめたものです。

技術情報の詳細な一覧はAppendixes に含まれています。

Quick reference guides には、Zabbix でタスクを実行するためのステップバイステップガイドが含まれています。

2 Zabbix とは

概要

Zabbix は Alexei Vladishev によって作成され、現在は Zabbix SIA が積極的に開発およびサポートを行っています。

Zabbix はエンタープライズクラスの分散監視ソリューションであり、オープンソースで配布されています。

Zabbix は、ネットワークの多数のパラメーターと、サーバーや仮想マシン、アプリケーション、サービス、データベース、Web サイト、クラウドなどの稼働状態と整合性を監視するためのソフトウェアです。Zabbix は柔軟な通知メカニズムを備え、すべてのイベントに対して電子メールによるアラートを設定できます。これによって、サーバーの問題に迅速に対応できます。Zabbix は、保存されたデータをもとに優れたレポート機能とデータ視覚化機能を提供し、キャパシティプランニングに最適なソフトウェアとなっています。

Zabbix はポーリングとトラッピングの両方をサポートしています。すべての Zabbix のレポート、統計情報、および設定パラメーターには、Web インターフェースからアクセスできます。Web インターフェースによりどこからでも、あなたのネットワークとサーバーの状態を確認することができます。Zabbix を適切に構成することで、IT インフラストラクチャの監視において重要な役割を果たすことができます。少数のサーバーを有する小規模な組織から、大量のサーバーを有する大企業まで、同様です。

Zabbix は無料です。Zabbix は AGPL-3.0 ライセンスの下で作成および配布されています。ソースコードは無料で配布され、誰でも自由に利用できます。

商用サポートは、Zabbix 社および世界中のパートナーによって提供されます。

詳細は、Zabbix の機能をご覧ください。

Zabbix のユーザー

世界中のあらゆる規模の多くの企業や組織が、メインの監視プラットフォームとして Zabbix を採用しています。

3 Zabbix の機能

概要

Zabbix は、高度に統合されたネットワーク監視ソリューションであり、1 つのパッケージで多様な機能を提供します。

データの収集

- 可用性とパフォーマンスのチェック
- ポーリングとトラッピング双方の SNMP、IPMI、JMX、VMware 監視のサポート
- チェックのカスタマイズ
- カスタマイズした間隔での詳細データの収集
- サーバーまたはプロキシによる実行と、エージェントによる実行

柔軟な閾値定義

- バックエンドデータベースからの値を参照して、トリガーと呼ばれる非常に柔軟な障害の閾値を定義可能

高度に設定可能なアラート機能

- エスカレーションスケジュール、受信者、メディアタイプなどをカスタマイズして通知可能
- マクロ変数を使用することにより、通知内容を分かりやすくすることが可能
- リモートコマンドを含む自動実行

リアルタイムグラフ

- 監視アイテムは組み込みのグラフ機能を使用して瞬時にグラフ化

Web 監視機能

- Web サイト上でのマウスクリックのパスをシミュレーションし、その機能と応答時間をチェック可能

幅広い可視化オプション

- 複数のアイテムを 1 つのビューに表示できるカスタムグラフ
- ネットワークマップ
- ダッシュボードスタイルで概要を表示するスライドショー
- レポート
- 監視リソースの高度な (ビジネス向け) 表示機能

履歴データのストレージ

- データは 1 つのデータベースに格納
- ヒストリの設定が可能
- 組み込みの Housekeeper(掃除) 処理

わかりやすい設定

- 監視対象デバイスをホストとして追加
- データベースに登録されるとホストは監視対象とされる
- 監視対象デバイスへのテンプレート適用

テンプレートの使用

- テンプレートで監視のグループ化
- テンプレートは他のテンプレートを継承可能

ネットワークディスカバリ

- ネットワークデバイスの自動検出
- エージェントの自動登録
- ファイルシステム、ネットワークインターフェース、SNMP OID の検出

高速な **Web** インターフェース

- PHP で記述された Web ベースのフロントエンド
- どこからでもアクセス可能
- クリックだけですべて実行可能
- 監査ログ

Zabbix API

- 大規模な操作やサードパーティのソフトウェア開発、その他の目的のために Zabbix をプログラムから操作できるインターフェースを提供

権限管理システム

- セキュリティで保護されたユーザー認証
- 特定のユーザー用の限定されたビュー

高機能かつ容易に拡張可能なエージェント

- 監視対象上に配置
- Linux と Windows の両方に配置可能

バイナリデーモン

- C 言語で記述されているため、高いパフォーマンスと少ないメモリ占有
- 移植が容易

複雑な環境への対応

- Zabbix プロキシを使用することでリモート監視を容易化

4 Zabbix の概要

アーキテクチャー

Zabbix は、いくつかの主なソフトウェアコンポーネントから構成されています。各コンポーネントの役割を以下に示します。

サーバー

Zabbix サーバー は、エージェントが可用性および整合性の情報と統計を報告する先となる中核コンポーネントです。サーバーは、すべての設定データ、統計データ、および運用データが保存される中央リポジトリです。

データベースストレージ

Zabbix によって収集されたデータと同様に、すべての設定情報もデータベースに保存されます。

Web インターフェース

どこからでもどんなプラットフォームからでも、Zabbix に簡単にアクセスできるように、Web インターフェースが提供されています。インターフェースは Zabbix サーバーの一部であり、通常はサーバーと同じ物理サーバー上で動作しますが、別サーバーで動作させることもできます。

プロキシ

Zabbix プロキシは、Zabbix サーバーの代わりにパフォーマンスと可用性データを収集します。プロキシは Zabbix のオプション部分ですが、1 台の Zabbix サーバーの負荷を分散するのにとても有益です。

エージェント

Zabbix エージェントは、監視対象上にインストールされ、ローカルリソースやアプリケーションを能動的に監視し、収集したデータを Zabbix サーバーに送信します。Zabbix 4.4 からは 2 種類のエージェントがあります: Zabbix エージェント(C 言語で実装され、低負荷で多くのプラットフォームをサポート)、Zabbix エージェント 2(Go 言語で実装され、柔軟でプラグインで拡張しやすい)。

データフロー

加えて、Zabbix 全体のデータフローを俯瞰的に見ることが非常に重要です。データを収集するアイテムを作成するためには、まず最初にホストを作成する必要があります。Zabbix で処理させるためには、最初にトリガーを作成するためのアイテムが必要です。アクションを作成するためには、トリガーが必要です。このように、例えば サーバー *X* で CPU 負荷が高すぎる時に通知を受け取りたい場合は、まず サーバー *X* のホストを作成し、続いて CPU を監視するためのアイテムを作成し、それから、CPU 負荷が高い場合に作動するトリガーを作成し、あなたにメールを送信するアクションを作成します。沢山のステップがあるように感じるかもしれませんが、テンプレートを使用すれば実はそんなことはありません。この設計であるため、非常に柔軟な構成が可能になっています。

このバージョンには重大な変更があります。

ネストされたローレベルディスカバリ

ローレベルディスカバリルール内にディスカバリプロトタイプを導入することで、オブジェクトの複数レベルのディスカバリを作成できるようになりました。例えば、データベースサーバー上のすべてのデータベースインスタンスを検出し、次に各インスタンスのテーブルスペースを検出し、最後に各テーブルスペースのテーブルを検出するといったことが可能です。

ディスカバリプロトタイプは、” 親” ディスカバリルール内にネストされたディスカバリルールです。ディスカバリプロトタイプには、独自のアイテム、トリガー、グラフ、ホスト、およびディスカバリプロトタイプがあります。

ネストされたディスカバリプロトタイプは、親ルールと同じ JSON 値を使用できますが、その JSON 値とは異なるデータの” スライス” を使用する場合があります。

ディスカバリプロトタイプのネストレベルは無制限です。

検出されたホスト上のホストプロトタイプ

検出されたホストでホストプロトタイプがサポートされるようになりました。これにより、Zabbix は検出された他のエンティティ (例: ハイパーバイザー、その仮想マシン、それらの仮想マシン内のコンテナ) 内のエンティティを自動的に検出し、監視できるようになります。

検出されたホスト上にホストプロトタイプを作成するには、ホストプロトタイプを含むローレベルディスカバリルールを作成するか、テンプレートをホストプロトタイプにリンクします。また、検出に使用したホストプロトタイプにテンプレートをリンクすると、検出されたホストはテンプレートからホストプロトタイプを継承します。

現在の構成に、他のホストプロトタイプを含むテンプレートを使用するホストプロトタイプが含まれている場合は、アップグレードに関する注意事項を参照してください。

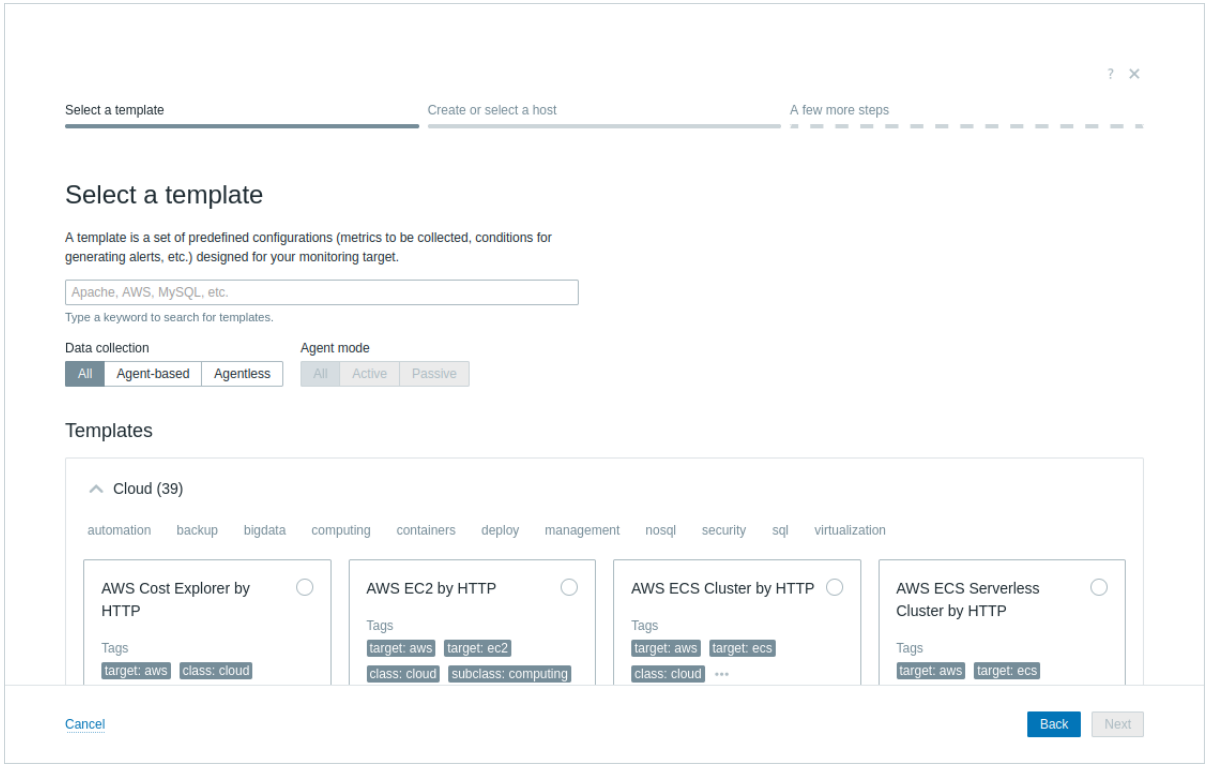
OAuth 2.0 認証

SMTP プロトコルで OAuth 2.0 認証がサポートされるようになりました。OAuth 認証を設定するには、メールのメディアタイプを設定する際に、認証パラメーターで”OAuth” を選択し、OAuthトークン取得のパラメーターを指定します。

Gmail、**Gmail** リレー、**Office365** のメールプロバイダでは、OAuth トークンの取得は自動化機能によって行われます。必要なのは、リダイレクトエンドポイント、クライアント *ID*、クライアントシークレットのパラメーター値のみです。Zabbix はその他の必要な値を自動的に入力します (参照: プロバイダー別の OAuth URL のデフォルト)。

ホストウィザード

新しいホストウィザードでは、Zabbix で監視対象（デバイス、アプリケーション、サービスなど）を設定するための、ガイド付きのステップバイステップのインターフェースが導入されています。
これにより、テンプレートの選択、Zabbix エージェントのインストール、ホストインターフェースの追加などの主要な手順をユーザーに案内し、新規または既存のホストの設定を簡素化します。

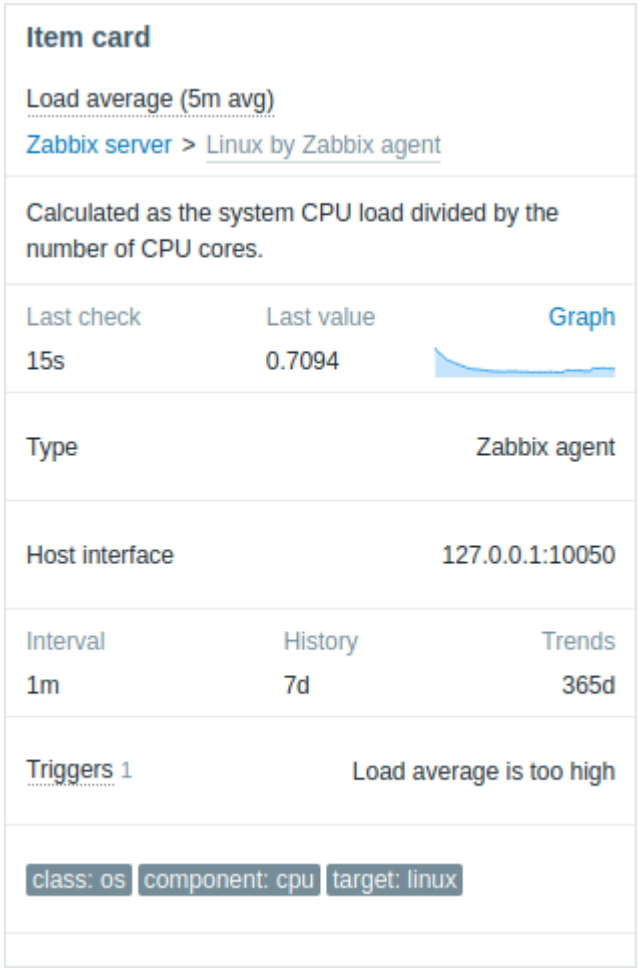


ホストウィザードには、データ収集 > ホストまたは各種 Web インターフェースセクションのホストメニューからアクセスできます。

以前の Zabbix バージョンからアップグレードする場合、ホストウィザードで動作させるにはテンプレートもアップグレードする必要があります。
手順については、テンプレートのアップグレードを参照してください。

ウィジェット　アイテムカード

アイテムカードウィジェットがダッシュボードウィジェットに追加され、単一のアイテムに関する包括的な情報をひと目で確認できる便利な方法が提供されます。



アイテム履歴

アイテム履歴ウィジェットでは、新しい値が下部に配置されるよう設定されている場合、スクロール位置も下部になります。これは、ログの最新の値を確認する際に便利です。

リアルタイムのウィジェット編集

ウィジェットの編集中に、ウィジェット設定の変更をリアルタイムでプレビューできるようになりました。
さらに、ウィジェット設定フォームをドラッグできるようになり、必要に応じて位置を変更できます。

グラフウィジェットフォーム内のグラフプレビューは削除されたことに注意してください。

アイテム　リトライオプション付き ICMP ping アイテム

リトライ回数を変更できる、ICMP ping によるホストの到達性監視用の新しい `icmppingretry[<target>,<retries>,<backoff>,<size>,` シンプルチェック が追加されました。

関数　タイムスタンプ追跡

タイムスタンプ追跡のために、新しい履歴関数が追加されました。

- `firstclock` - 定義された評価期間内における最も古い値のタイムスタンプ
- `lastclock` - 定義された評価期間内における N 番目に新しい値のタイムスタンプ
- `logtimestamp` - N 番目に新しいログアイテム値のログメッセージタイムスタンプ。

マクロ　アイテム値の時刻追跡

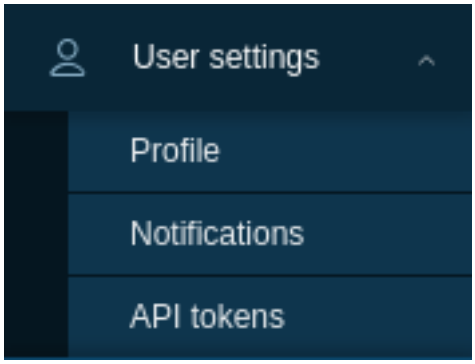
アイテム値の時刻追跡用の新しいマクロが追加されました。

Macro	Description
{ITEM.LASTVALUE.AGE}	最新のアイテム値が収集されてからマクロが評価されるまでに経過した時間。
{ITEM.LASTVALUE.DATE}	最新のアイテム値が収集された日付。
{ITEM.LASTVALUE.TIME}	最新のアイテム値が収集された時刻。
{ITEM.LASTVALUE.TIMESTAMP}	最新のアイテム値が収集された時点のタイムスタンプ。
{ITEM.VALUE.AGE}	アイテム値が収集されてからマクロが評価されるまでに経過した時間。

Macro	Description
{ITEM.VALUE.DATE}	アイテム値が収集された日付。
{ITEM.VALUE.TIME}	アイテム値が収集された時刻。
{ITEM.VALUE.TIMESTAMP}	アイテム値が収集された時点のタイムスタンプ。

通知　ユーザー通知用の独立したメニューセクション

ユーザーメディアの視認性を高めるため、ユーザー通知は ユーザー設定配下の独自のメニューセクションを持つようになりました。



新しい「通知」セクションには、以前はユーザープロファイルセクションの一部であった メディアと *Web* インターフェース通知の 2 つのタブが含まれています。

自身のユーザーメディアの管理

現在、すべてのユーザーはデフォルトで自身のメディアを管理できます。

ただし、自身のメディアの詳細を変更する権限は、ユーザーの役割に基づいて付与または取り消すことができます（*Create and edit own media* オプションを参照）。

さらに、他のユーザーのメディアの詳細を変更する Super admin ユーザーの権限も、ユーザーの役割に基づいて付与または取り消すことができます（*Create and edit user media* オプションを参照）。

マップ　ラベルの自動非表示

マップ要素/リンクのラベルを、ホバー時または選択時にのみ表示するよう設定できるようになりました。
この設定により、多くの要素が互いに近接しているマップで、視覚的な煩雑さを軽減できます。

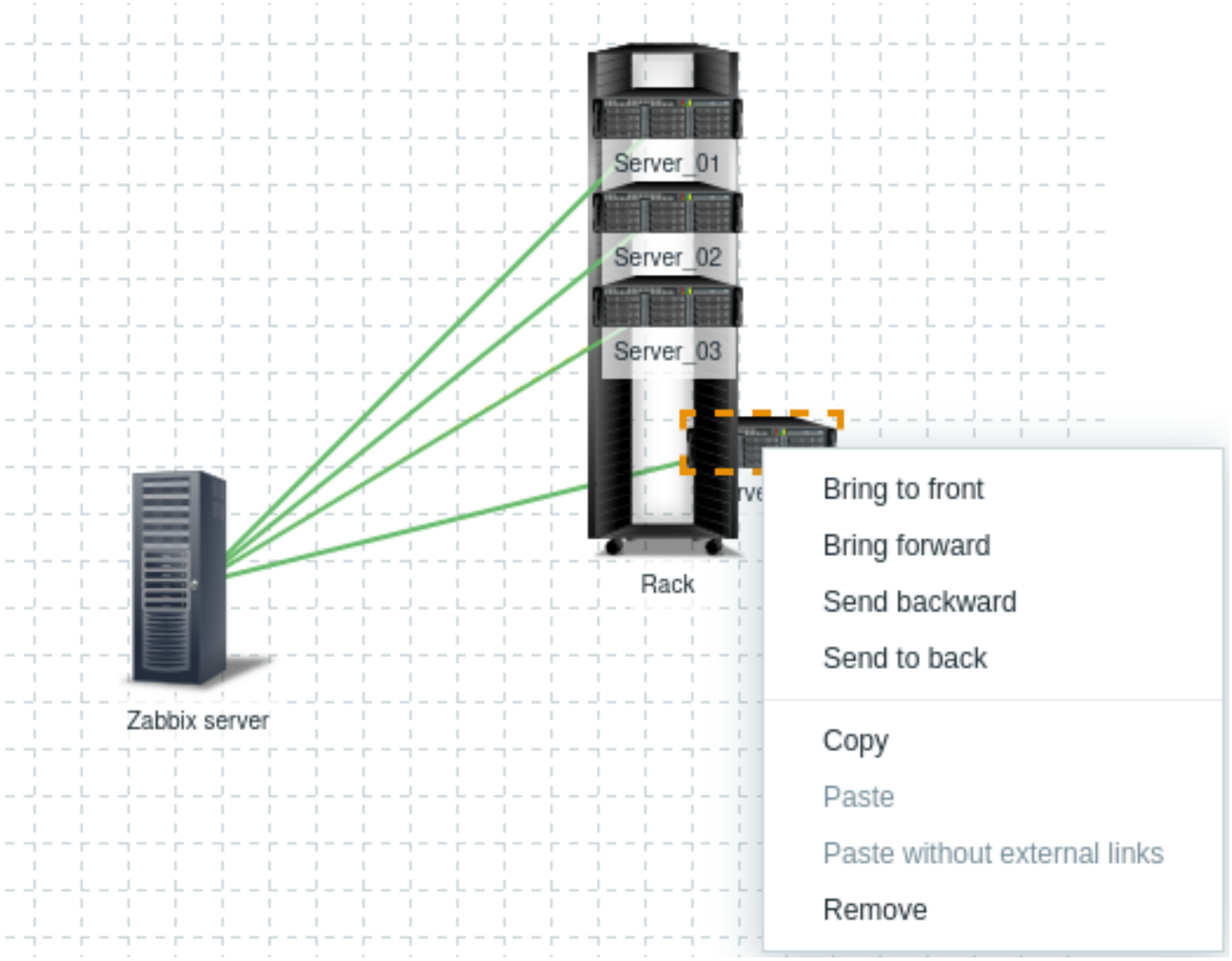
ラベルの自動非表示は、すべてのマップ要素/リンクに対してグローバルに設定することも、個々のマップ要素/リンクごとに設定することもできます。

背景画像の比例スケーリング

背景画像をマップサイズに合わせて、縦横比を保ったまま拡大・縮小できるようになりました。

要素の順序

要素を右クリックし、*Bring forward*/*Bring to front* または *Send backward*/*Send to back* オプションを選択することで、ある要素を別の要素の前面（またはその逆）に移動できるようになりました。



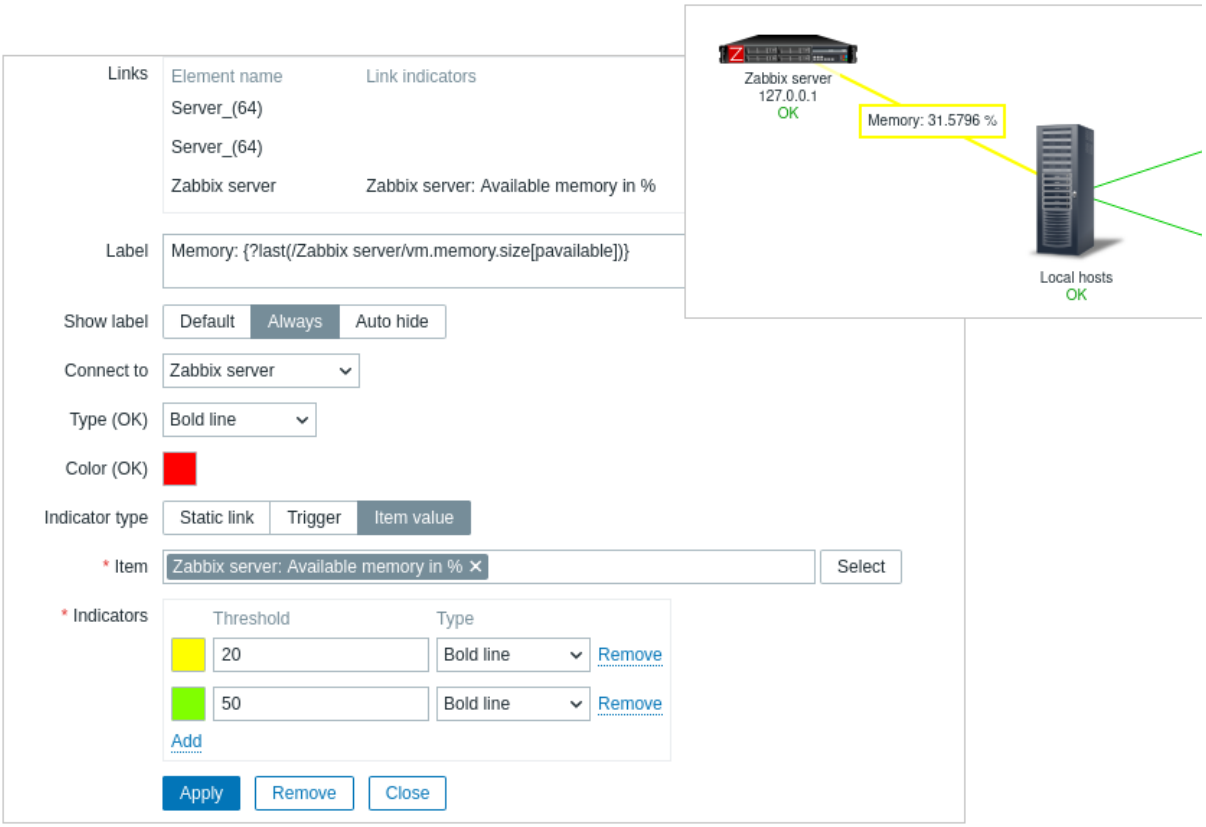
さらに、マップ要素に要素間のリンクとハイライトの両方が含まれている場合、リンクはハイライトの下に表示されるようになりました。

アイテム値に基づくリンクインジケーター

以前のバージョンでは、何らかのトリガーが障害状態になると、リンクのスタイルと色を調整できました。現在は、アイテム値に基づいて同様の機能を利用できます。

アイテム値が次の場合、リンクのスタイルと色を調整できます。

- 定義されたしきい値に達する（数値アイテムの場合）。
- 正規表現に一致する（テキストアイテムの場合）。



プロセス　履歴キャッシュの管理

一部のデータ収集シナリオでは、特定のアイテムが一時的にサーバー/プロキシの履歴キャッシュをブロックすることがあります。これにより、履歴データのデータベースへの書き込みが遅延し、システム全体の動作が遅くなる可能性があります。履歴キャッシュの管理を支援するため、以下の改善が導入されました。

- 手動でのキャッシュクリア**:** 新しい history_cache_clear=target ランタイムコマンドにより、Zabbix server/proxy で、特定のアイテムの ID を指定してその履歴キャッシュを手動でクリアできるようになりました。
- 自動キャッシュクリア**:** アイテムを無効化すると、そのアイテムは履歴キャッシュから即座に削除されます（ただし、ログ用に最後の値は保持されます）。同様に、ホストを無効化すると、そのホストのすべてのアイテムが履歴キャッシュから削除されます（ただし、それぞれの最後の値は保持されます）。
- キャッシュ診断ログ**:** 履歴キャッシュがいっぱいになると、Zabbix サーバー/プロキシはDebugLevel=3 から履歴キャッシュの診断情報をログに記録するようになりました（以前は DebugLevel=4）。このログには、履歴キャッシュ内で最も多くの値を保持しているアイテムが含まれます。

長期的なシステム安定性を確保するため、データ収集が利用可能なリソース（データベース性能、キャッシュサイズ、収集間隔、ログアイテムのパラメータなど）と適切にバランスしていることを確認してください。Zabbix の履歴キャッシュは、内部アイテムzabbix[wcache] を使用して監視できます。また、Zabbix server/proxy の履歴キャッシュサイズを増やすことも検討してください。

履歴同期プロセスのトランザクション統計

履歴同期プロセスのプロセスタイトルに、Zabbix サーバー およびプロキシ の履歴同期トランザクションに関する詳細な統計情報が表示されるようになりました。

セキュリティ　Web インターフェースとサーバー間の TLS 暗号化

Zabbix Web インターフェースと Zabbix サーバー間の通信を暗号化できるようになりました。この機能は、サーバー設定の新しいパラメータ（TLSListen、TLSFrontendAccept、TLSFrontendCertIssuer、TLSFrontendCertSubject、FrontendAllowedIP）によって制御されます。

サーバー/プロキシごとに独立してシークレットボールトマクロを解決する

Resolve secret vault macros by が「Zabbix server and proxy」に設定されている場合、vault シークレットマクロの値を Zabbix サーバーと Zabbix プロキシがそれぞれ独立して取得するよう設定できるようになりました。

プラグイン　ロード可能なプラグインのカスタムクエリパス設定

MySQL、Oracle、PostgreSQL 用の Zabbix エージェント 2 プラグイン設定ファイル内の `Plugins.*.CustomQueriesPath` パラメータ、および MSSQL 用の `Plugins.MSSQL.CustomQueriesDir` に、デフォルト値が設定されました。この機能はデフォルトでは無効になっており、新たに導入された `Plugins.*.CustomQueriesEnabled` パラメータを使用して有効にできます。

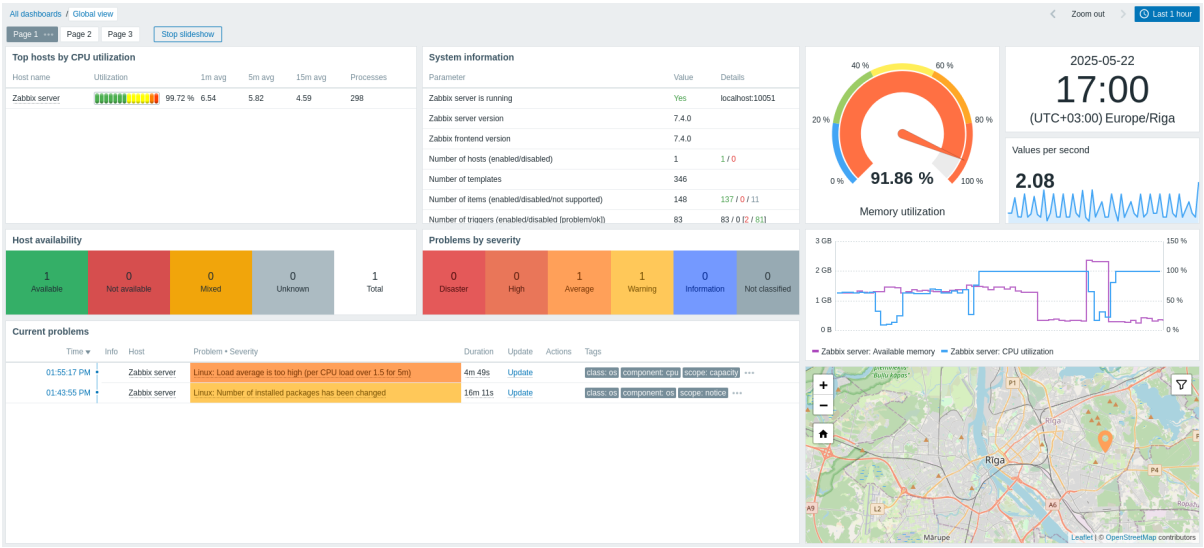
Web インターフェース　フォームでのインラインバリデーション

以下の Web インターフェースのフォームでインラインバリデーションがサポートされるようになり、フィールドへの入力後に入力エラーが即座に表示されます。

- テンプレート設定
- ホスト設定
- アイテムおよびアイテムのプロトタイプ設定
- トリガーおよびトリガーのプロトタイプ設定
- モジュール管理

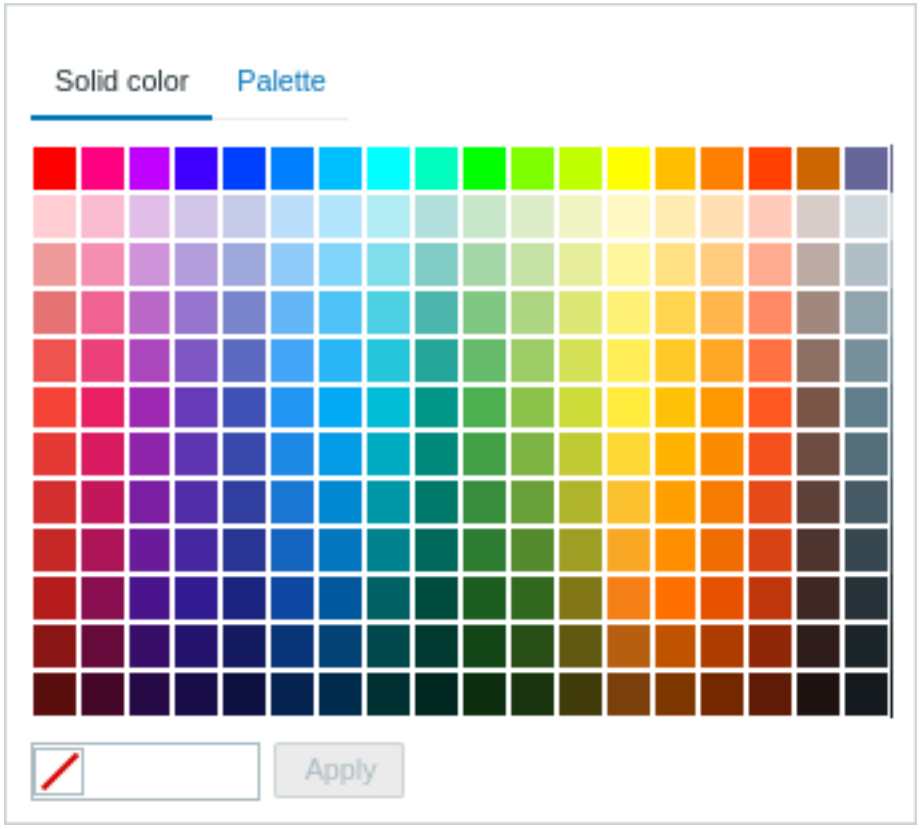
デフォルトダッシュボードを更新

新しい Zabbix インストール環境における *Global view* のデフォルトdashboardsは、最新のダッシュボードウィジェットを含むように更新されました。

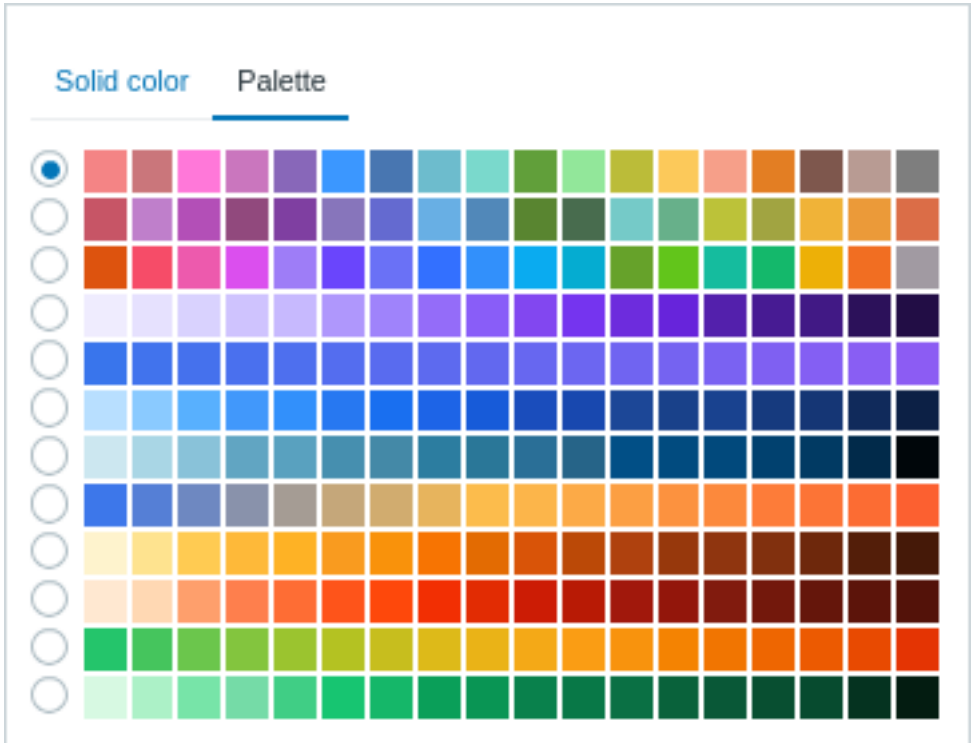


パレット対応で強化されたカラーピッカー

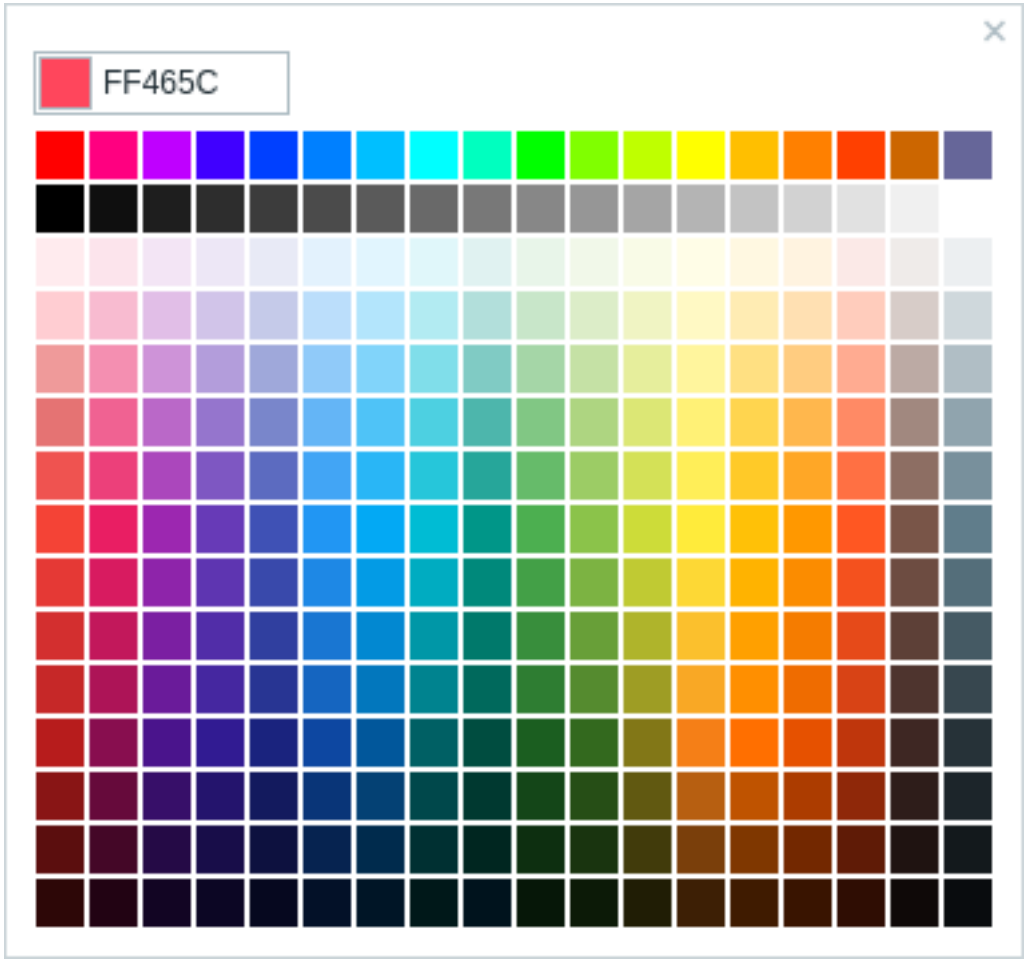
graph およびpie chart ウィジェットのカラーピッカーは、使いやすさを向上させるために再設計されました。単色の選択や 16 進数入力に加えて、*Palette* タブに切り替えて定義済みの色の行から選択できるようになりました。各系列にはそれぞれ異なる色相が割り当てられるため、より明確に区別できます。また、このピッカーはキーボードによる完全な操作とライブバリデーションにも対応しています。



Zabbix 7.4 の場合（*Solid color* タブ）



Zabbix 7.4 の場合（*Color palette* タブ）



Zabbix 7.4 より前

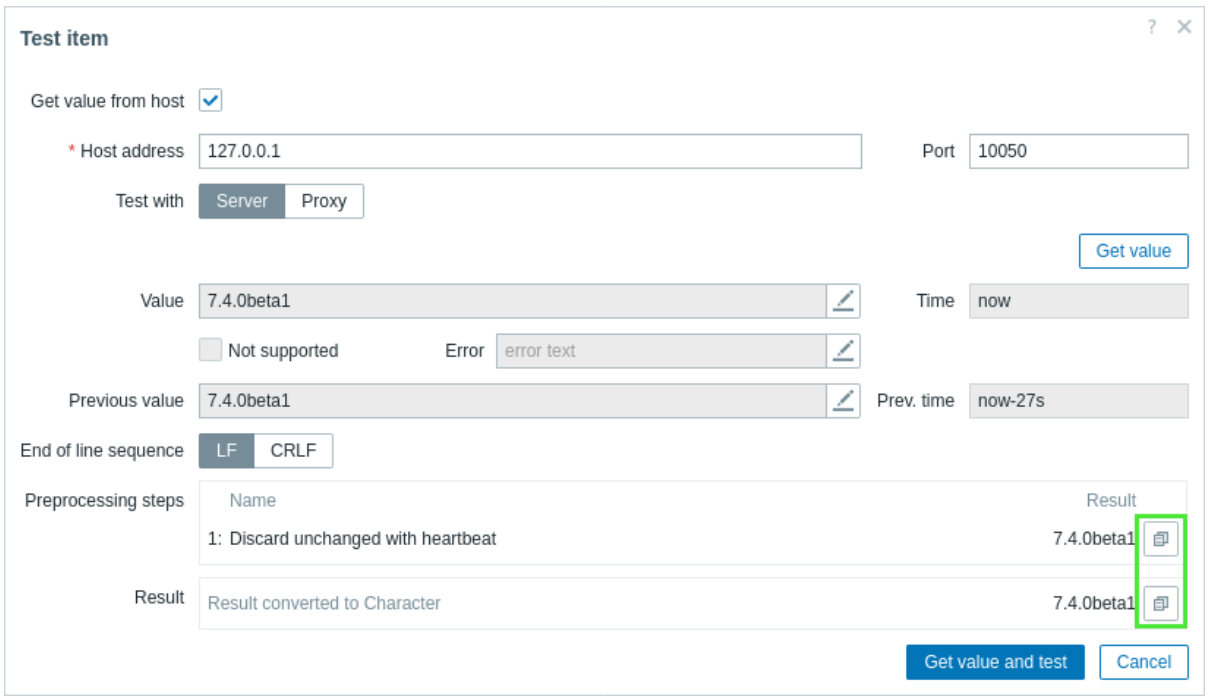
モーダルフォーム

いくつかの Web インターフェースフォームが、モーダル（ポップアップ）ウィンドウで開くようになりました。

- グラフの設定
- グラフプロトタイプの設定

前処理のtest resultsのコピーがより簡単に

追加された *Copy to clipboard* ボタンにより、前処理のtest resultsで値を簡単にコピーできるようになりました。



同様のボタンが、作成されたAPI tokensの *Copy* リンクの代わりに使用されるようになった点にも注意してください。

ドキュメント　マイナーリリース向けの統合ドキュメントページ

Zabbix のメジャーリリースにおけるマイナーバージョンのリリースドキュメントは、今後、それぞれ新機能 およびアップグレードに関する注意事項 の単一のドキュメントページにまとめられます。

6 Zabbix 7.4.x の新機能

このページでは、Zabbix のメジャーバージョンのマイナーリリースに含まれる新機能に関する総合的な情報を提供します。

メジャーバージョンの新機能も参照してください。

Zabbix 7.4.9 の新機能　webhook 連携

- 新しい連携により、Zabbix のwebhookメディアタイプを使用して、Zabbix をIBM Maximo Service Requestにリンクできるようになりました。
- GLPI webhook メディアタイプが更新され、OAuth2 認可に対応した RESTful API v2 のサポート（デフォルトで有効）、設定フラグによる従来の REST API v1 のオプションサポート、障害イベントのチケット作成、およびユーザー設定可能な緊急度マッピングに対応しました。

MySQL 9.6 のサポート

MySQL の最大サポート対象バージョンは、9.6.X になりました。

TimescaleDB 2.26 のサポート

TimescaleDB の最大サポート対象バージョンは、2.26.X になりました。

MariaDB 12.2 のサポート

MariaDB の最大サポート対象バージョンは、12.2.X になりました。

Zabbix 7.4.8 の新機能　TimescaleDB 2.25 のサポート

TimescaleDB の最大サポート対象バージョンは、2.25.X になりました。

Zabbix 7.4.7 の新機能　TimescaleDB 2.24 のサポート

TimescaleDB の最大サポート対象バージョンは、2.24.X になりました。

Zabbix 7.4.6 の新機能　ログファイルの作成時における制限付き権限に関連する互換性のない変更を参照してください。

TimescaleDB 2.23 のサポート

TimescaleDB の最大サポート対象バージョンは、2.23.X になりました。

MySQL 9.5 のサポート

MySQL の最大サポート対象バージョンは、9.5.X になりました。

エージェント MSI インストーラーパッケージの拡張コマンドラインパラメータ

Zabbix エージェント MSI インストーラーパッケージ（Zabbix エージェントおよび Zabbix エージェント 2 の両方）は、コマンドラインからのインストールを実行する際に、以下のパラメータをサポートするようになりました。

- `HOSTNAMEITEM` - ホスト名の取得に使用する Zabbix エージェントのアイテムを指定します
- `UNSAFEUSERPARAMETERS` - ユーザー定義パラメータの引数に、すべての文字（例: ' " ` * ?）を渡せるようにします
- `STARTAGENTS` - パッシブチェックを処理する、事前に fork された `zabbix_agentd` インスタンスの数を指定します
- `DONOTSTART` - MSI インストーラーが Zabbix エージェントサービスを起動しないようにします
- `SOURCEIP` - Zabbix サーバー/プロキシへの外向き接続、または特定のアイテム（例: `web.page.get`、`net.tcp.port`）の実行時に行われる接続に使用する送信元 IP アドレスを指定します

ICMP ping アイテムの最大タイムアウト

icmpping* アイテムには、ハードコードされた最大 600 秒のタイムアウトが適用されるようになりました。

Zabbix 7.4.5 の新機能　このマイナーバージョンには、機能上の変更はありません。

Zabbix 7.4.4 の新機能　注意**!** グラフと Zabbix エージェント 2 の MySQL プラグインに関連するエラーのため、このバージョンへのアップグレードはお勧めできません。

PostgreSQL 18 のサポート

PostgreSQL の最大サポート対象バージョンは 18.X になりました。

TimescaleDB 2.22 のサポート

TimescaleDB の最大サポート対象バージョンは、2.22.X になりました。

Redis プラグイン — TLS サポートと起動時検証

TLS サポートが Zabbix エージェント 2 のRedis プラグインに追加されました。

プラグイン TLS 構成の起動時検証が実装され、検証/エラーメッセージが改善されました。無効な構成ロジック (例: `TLSCAFile` を指定せずに接続タイプ `verify_full` を使用) により、Zabbix エージェント 2 の起動が妨げられる可能性があります。

Zabbix 7.4.3 の新機能　Oracle プラグインの TNS 名サポート

Zabbix エージェント 2 の Oracle プラグインに、Oracle ネイティブの TNS 名のサポートが追加されました。TNS キーまたは TNS 値は、アイテムキーパラメーターまたはプラグイン設定(名前付きセッションまたはデフォルトパラメーターの場合) のレベルで指定できます。

TNS 名のサポートにより、単一の Zabbix ホストを持つクラスター化された DB 環境での接続記述が可能になります。詳細については、Oracle プラグインのREADMEをご覧ください。

zabbix_get および zabbix_js の最大タイムアウトの延長

zabbix_getおよびzabbix_jsコマンドラインユーティリティの `timeout` パラメーターの最大値が 600 秒に延長されました。

MariaDB 12.0 のサポート

MariaDB の最大サポート対象バージョンは、12.0.X になりました。

障害ビューと障害ウィジェットで行全体の強調表示

行全体をハイライトオプションが、非コンパクトな監視 > 障害ビューと障害ダッシュボードウィジェットで使用できるようになりました。

Zabbix 7.4.2 の新機能　Oracle プラグインの追加ロールのサポート

Zabbix エージェント 2 の Oracle プラグインアイテムでは、ログインオプションで SysBACKUP、SysDG、SysKM、SysRAC などの追加ロールがサポートされるようになりました。

SNMPv3 認証情報の自動リロード

SNMP エージェントの walk[] および get[] アイテムチェックは、対応する SNMPv3 インターフェースが変更された場合、更新された SNMPv3 認証およびプライバシー設定を自動的にリロードするようになりました。手動キャッシュリロードコマンド-R snmp_cache_reloadは、トラブルシューティングや、SNMPv3 仕様に完全に準拠していないデバイスのために引き続き使用できます。

smart.disk.discovery に **type** パラメーターを追加

smart.disk.discoveryアイテム (Zabbix agent 2 S.M.A.R.T. plugin) に、ディスクをスキャンする値を指定するためのオプションの **type** パラメーターが追加されました。

Zabbix 7.4.1 の新機能　TimescaleDB 2.21 のサポート

TimescaleDB の最大サポート対象バージョンは、2.21.X になりました。

サーバー/プロキシの保存前処理診断情報

サーバー/プロキシのランタイム診断情報 (`diaginfo` ランタイムコマンドを実行することでアクセス可能) の保存前処理セクションに、受信値の数とサイズでソートされた上位アイテムセクションが追加されました。処理時間が最も長い上位アイテム ID が強調表示されます。

この情報には、diaginfo ランタイムオプション (例: `zabbix_server -R diaginfo=preprocessing`) を Zabbix サーバーまたはプロキシを実行することでアクセスできます。

保存前処理のないアイテムについては保存前処理マネージャを迂回

処理負荷を軽減するため、保存前処理も依存アイテムも存在しないアイテムのアイテム値は、ヒストリキャッシュに直接追加されるか、LLD マネージャに送信されるようになりました (参照: 保存前処理の詳細)。以前は、保存前処理の設定に関わらず、すべてのアイテム値が保存前処理マネージャを通過していました。

GLPI webhook 連携を更新

GLPI メディアタイプ は、アプリケーショントークンを使用した認証をサポートするようになりました。

2. 定義

概要　このセクションでは、Zabbix で一般的に使用される用語の意味について説明します。

定義　ホスト

- 監視対象の任意の物理または仮想デバイス、アプリケーション、サービス、あるいはその他の論理的に関連付けられた監視パラメータの集合。

ホストグループ

- ホストの論理的なグループ化。
ホストグループは、異なるユーザーグループに対してホストへのアクセス権を割り当てる際に使用されます。

アイテム

- ホストから受信したい特定のデータ、すなわちデータのメトリクス。

値の前処理

- 受信したメトリクス値をデータベースに保存する前に行う変換。

トリガー

- 問題のしきい値を定義し、アイテムで受信したデータを「評価」するために使用される論理式。

受信したデータがしきい値を上回ると、トリガーは「Ok」から「Problem」状態に移行します。受信したデータがしきい値を下回ると、トリガーは「Ok」状態のまま維持されるか、「Ok」状態に戻ります。

テンプレート

- *1* つまたは複数のホストに適用できるよう準備されたエンティティ（アイテム、トリガー、グラフ、ローレベルディスカバリルール、*Web* シナリオ）のセット。

テンプレートの役割は、ホスト上での監視タスクの展開を迅速化することです。また、監視タスクに対する一括変更を容易にすることにも役立ちます。テンプレートは個々のホストに直接リンクされます。

テンプレートグループ

- テンプレートの論理的なグループ化。
テンプレートグループは、異なるユーザーグループに対してテンプレートへのアクセス権を割り当てる際に使用されます。

イベント

- トリガーの状態変化やディスカバリ/エージェント自動登録の実行など、注意を要する事象の単一の発生。

イベントタグ

- イベント用の事前定義されたマーカー。イベント相関、権限の細分化などで使用できます。

イベント相関

- 問題とその解決を柔軟かつ正確に関連付ける方法。

たとえば、あるトリガーによって報告された問題が、別のトリガーによって解決されるよう定義できます。その別のトリガーでは、異なるデータ収集方法を使用することも可能です。

問題

- 「*Problem*」状態にあるトリガー。

問題の更新

- コメントの追加、確認、深刻度の変更、手動クローズなど、*Zabbix* が提供する問題管理オプション。

アクション

- イベントに対応するための事前定義された手段。

アクションは、操作（例: 通知の送信）と条件（その操作をいつ実行するか）で構成されます。

エスカレーション

- アクション内で操作を実行するためのカスタムシナリオ。通知の送信/リモートコマンドの実行の一連の流れ。

マクロ

- コンテキストに応じて特定の値に解決される変数プレースホルダー。

メディア

- 通知を配信するための手段。配信チャネル。

通知

- 選択したメディアチャネルを通じてユーザーに送信される、何らかのイベントに関するメッセージ。

リモートコマンド

- 何らかの条件に応じて監視対象ホスト上で自動的に実行される、事前定義されたコマンド。

Web シナリオ

- *Web* サイトの可用性を確認するための *1* つまたは複数の *HTTP* リクエスト。

Web インターフェース

- *Zabbix* に付属する *Web* インターフェース。

ダッシュボード

- 重要な情報の要約や可視化を、ウィジェットと呼ばれる視覚単位で表示する、カスタマイズ可能な *Web* インターフェースのセクション。

ウィジェット

- 特定の種類およびソースの情報を表示する視覚単位（要約、マップ、グラフ、時計など）で、ダッシュボードで使用されます。

Zabbix API

- *Zabbix API* を使用すると、*JSON RPC* プロトコルを利用して *Zabbix* オブジェクト（ホスト、アイテム、グラフなど）を作成、更新、取得したり、その他の任意のカスタムタスクを実行したりできます。

Zabbix サーバー

- 監視を実行し、*Zabbix* プロキシおよびエージェントとやり取りし、トリガーを計算し、通知を送信する *Zabbix* ソフトウェアの中核プロセス。データの中央リポジトリでもあります。

Zabbix プロキシ

- *Zabbix* サーバーに代わってデータを収集し、サーバーの処理負荷の一部を軽減できるプロセス。

Zabbix エージェント

- ローカルリソースおよびアプリケーションを能動的に監視するために、監視対象に配置されるプロセス。

Zabbix エージェント ***2***

- ローカルリソースおよびアプリケーションを能動的に監視するための新世代の *Zabbix* エージェントで、監視にカスタムプラグインを使用できます。

> **Attention:**
> Zabbix エージェント 2 は Zabbix エージェントと多くの機能を共有しているため、ドキュメント内で「Zabbix エージェント」という用語は、機能動作が同じであれば Zabbix エージェントと Zabbix エージェント 2 の両方を指します。Zabbix エージェント 2 は、その機能が異なる場合にのみ個別に明記されます。

暗号化

- *TLS*（*Transport Layer Security*）プロトコルを使用した、*Zabbix* コンポーネント（サーバー、プロキシ、エージェント、*zabbix_sender* および *zabbix_get* ユーティリティ）間の暗号化通信のサポート。

エージェント自動登録

- *Zabbix* エージェント自体がホストとして登録され、監視が開始される自動化プロセス。

ネットワークディスカバリ

- ネットワークデバイスの自動検出。

ローレベルディスカバリ

- 特定のデバイス上の低レベルエンティティの自動検出
(例: ファイルシステム、ネットワークインターフェースなど)。

ローレベルディスカバリルール

- デバイス上の低レベルエンティティを自動検出するための定義セット。

アイテムプロトタイプ

- 変数として特定のパラメータを持つメトリクスで、ローレベルディスカバリ用に準備されたもの。ローレベルディスカバリ後、変数は自動的に実際に検出されたパラメータに置き換えられ、そのメトリクスは自動的にデータ収集を開始します。

トリガープロトタイプ

- 変数として特定のパラメータを持つトリガーで、ローレベルディスカバリ用に準備されたもの。ローレベルディスカバリ後、変数は自動的に実際に検出されたパラメータに置き換えられ、そのトリガーは自動的にデータ評価を開始します。

他のいくつかの Zabbix エンティティのプロトタイプもローレベルディスカバリで使用されます。たとえば、グラフプロトタイプ、ホストプロトタイプ、ホストグループプロトタイプがあります。

3. Zabbix のプロセス

サイドバーから、参照したい項目を選択してください。

1 Zabbix サーバー

概要

Zabbix サーバーは、Zabbix のソフトウェアの中核となるプロセスです。

Zabbix サーバーは、データのポーリングとトラッピングを処理し、それをトリガーによって判断し、ユーザーに対して通知を行います。Zabbix エージェントや Zabbix プロキシからシステムの可用性と整合性に関するデータを受け取る中核となるコンポーネントです。Zabbix サーバー自体は、シンプルなサービスチェックの機能を使用して、ネットワークに接続されたサービス（Web サーバーやメールサーバーなど）をリモートからチェックすることができます。

Zabbix サーバーは、すべての設定、統計、運用データが保存される中核となるリポジトリで、監視対象のシステムで障害が発生したときに、管理者に対して能動的に警告を通知する役割を担います。

Zabbix サーバーの基本的な機能は、3 つの異なるコンポーネントに分けられます。その 3 つとは、Zabbix サーバー、Web インターフェース、データベースストレージです。

Zabbix のすべての設定情報はデータベース上に保存され、Zabbix サーバーと Web インターフェースの両方がそのデータベースと通信します。例えば、Web インターフェース（または API）を使用して新しいアイテムを作成した時には、データベース内の items テーブルに追加されます。次に、Zabbix サーバーはおよそ 1 分ごとに有効なアイテムの一覧を取得するため items テーブルを検索し、Zabbix サーバー内のキャッシュに保存します。このため、Web インターフェースで行った変更が最新データとして反映されるまでに最大 2 分かかります。

Zabbix サーバーの起動

パッケージを使用してインストールした場合

Zabbix サーバーはデーモンプロセスとして実行されます。サーバーを起動するには以下を実行します。

```
systemctl start zabbix-server
```

これはほとんどの GNU/Linux システムで動きます。他のシステムでは、以下のように実行する必要がある場合があります。

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

同様に、停止/再起動/ステータス表示するには以下を実行します。

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

手動での起動

上記の方法で起動できない場合は手動で起動する必要があります。zabbix_server バイナリへのパスを見つけ、以下のように実行します:

```
zabbix_server
```

Zabbix サーバーでは、以下のコマンドラインパラメーターを使用できます:

```
-c --config <file>              設定ファイルへのパスを指定する（デフォルトは/usr/local/etc/zabbix_server.conf）
-f --foreground                 Zabbix サーバーをフォアグラウンドで実行する
-R --runtime-control <option>   管理機能を実行する
-T --test-config                設定ファイルを検証して終了する
-h --help                       ヘルプを表示する
-V --version                    バージョン情報を表示する
```

コマンドラインパラメーターを使用して Zabbix サーバーを実行する例:

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

ランタイム制御

ランタイム制御オプション:

Option	Description	Target
config_cache_reload	設定キャッシュを再読み込みします。現在キャッシュの読み込み中である場合は無視されます。	
history_cache_clear[=<target>]	HA で指定したアイテムの履歴キャッシュをクリアします。 アイテムのすべての値に影響しますが、最初と最後の値は除きます。	**target** - アイテムの ID
diaginfo[=<**section**>]	サーバーのログファイルに診断情報を収集します。	**historycache** - 履歴キャッシュ統計 **valuecache** - 値キャッシュ統計 **preprocessing** - 前処理マネージャー統計 **alerting** - アラートマネージャー統計 **lld** - LLD マネージャー統計 **locks** - ミューテックスの一覧 (*BSD* システムでは空です) **connector** - キューが最も大きいコネクタの統計
ha_status	高可用性 (HA) クラスターのステータスをログに記録します。	
ha_remove_node=target	名前または ID で指定した高可用性 (HA) ノードを削除します。 アクティブ/スタンバイノードは削除できないことに注意してください。	**target** - ノードの名前または ID (ha_status を実行して取得できます)
ha_set_failover_delay=delay	高可用性 (HA) フェイルオーバー遅延を設定します。時間サフィックス をサポートします。例: 10s、1m。	
proxy_config_cache_reload[=<target>]	プロキシ設定キャッシュを再読み込みします。	**target** - プロキシ名のカンマ区切りリスト target を指定しない場合は、すべてのプロキシの設定を再読み込みします
secrets_reload	Vault からシークレットを再読み込みします。	
service_cache_reload	サービスマネージャーのキャッシュを再読み込みします。	
snmp_cache_reload	SNMP キャッシュを再読み込みします — すべてのホストについて SNMP エンジンのプロパティ (engine time、engine boots、engine id、credentials) をクリアします。SNMP の問題をトラブルシュートする際に、グローバルなキャッシュクリアを強制するために使用します。	
housekeeper_execute	ハウスキーピング 手順を開始します。 ハウスキーピング手順が現在進行中の場合は無視されます。	

Option	Description	Target
trigger_housekeeper_execute	トリガーのハウスキーピング手順を開始します。 トリガーのハウスキーピング手順が現在進行中の場合は無視されます。 トリガーのハウスキーピング手順が開始されるまで、すでに削除されたトリガーによって発生した障害が、引き続きサービス障害を生成し、それらをサービスに割り当てる可能性があります。頻繁に検出/未検出されるトリガーに基づくサービスのステータス計算ルールが多数ある構成では、ProblemHousekeepingFrequency サーバー設定パラメータを調整して、ハウスキーピング手順の頻度を増やすことを検討してください。	
log_level_increase[=**<target>**]	ログレベルを上げます。target が指定されていない場合は、すべてのプロセスに影響します。 *BSD* システムではサポートされていません。	**process type** - 指定したタイプのすべてのプロセス (例: poller) すべてのサーバープロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号 (例: poller,3) **pid** - プロセス識別子 (1 から 65535)。それより大きい値の場合は、target を'process type,N' として指定してください。
log_level_decrease[=**<target>**]	ログレベルを下げます。target が指定されていない場合は、すべてのプロセスに影響します。 *BSD* システムではサポートされていません。	
prof_enable[=**<target>**]	プロファイリングを有効にします。 target が指定されていない場合は、すべてのプロセスに影響します。 プロファイリングを有効にすると、すべての rwlocks/mutexes の詳細が関数名ごとに提供されます。	**process type** - 指定したタイプのすべてのプロセス (例: history syncer) プロファイリング対象としてサポートされるプロセスタイプ: alerter, alert manager, availability manager, configuration syncer, discovery manager, escalator, history poller, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, lld manager, lld worker, odbc poller, poller, preprocessing manager, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector **process type,N** - プロセスタイプと番号 (例: history syncer,1) **pid** - プロセス識別子 (1 から 65535)。それより大きい値の場合は、target を'process type,N' として指定してください。 **scope** - `rwlock`、`mutex`、`processing` は、プロセスタイプと番号 (例: history syncer,1,processing) またはそのタイプのすべてのプロセス (例: history syncer,rwlock) とともに使用できます
prof_disable[=**<target>**]	プロファイリングを無効にします。 target が指定されていない場合は、すべてのプロセスに影響します。	**process type** - 指定したタイプのすべてのプロセス (例: history syncer) プロファイリング対象としてサポートされるプロセスタイプ: `prof_enable` を参照 **process type,N** - プロセスタイプと番号 (例: history syncer,1) **pid** - プロセス識別子 (1 から 65535)。それより大きい値の場合は、target を'process type,N' として指定してください。

ランタイム制御を使用してサーバー設定キャッシュをリロードする例:

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

ランタイム制御を使用してプロキシ設定をリロードする例:

```
### すべてのプロキシ設定をリロードする:
zabbix_server -R proxy_config_cache_reload

### Proxy1 と Proxy2 の設定をリロードする:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

ランタイム制御を使用してアイテムのヒストリキャッシュをクリアする例:

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

ランタイム制御を使用して診断情報を収集する例:

```
### サーバーログファイルで利用可能なすべての診断情報を収集する:
zabbix_server -R diaginfo

### サーバーログファイルでヒストリキャッシュ統計を収集する:
zabbix_server -R diaginfo=historycache
```

ランタイム制御を使用して SNMP キャッシュをリロードする例:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Zabbix UI を介して SNMPv3 インターフェースが更新されると、Zabbix はほとんどの場合、そのインターフェースの新しい SNMPv3 資格情報を自動的にリロードします。資格情報が変更された後ポーリングがまだ失敗した場合 (たとえば、engineBoots/engineID の矛盾または非 RFC デバイスのために)、またはグローバル SNMP キャッシュをトラブルシューティングのために強制的にクリアする必要がある場合にのみ、`-R snmp_cache_reload` を使用します。

ランタイム制御を使用してハウスキーパーの実行をトリガーする例:

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

ランタイム制御を使用してログレベルを変更する例:

```
### すべてのプロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### 2 番目のポーラープロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### PID 1234 のプロセスのログレベルを上げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### すべての http ポーラープロセスのログレベルを下げる:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

HA フェイルオーバー遅延を最小 10 秒に設定する例:

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

プロセス起動ユーザー

Zabbix サーバーは、非 root ユーザーで動作するように設計されています。起動されたどんな非 root ユーザーでも起動すれば動作します。したがって、非 root ユーザーで問題なくサーバーを実行できます。

'root' として起動しようとするとハードコードされた'zabbix' ユーザーに切り替わります。このユーザーは、システム上に存在する必要があります。サーバーの設定ファイル内の'AllowRoot' パラメーターを変更した時のみ、'root' で実行させることができます。

Zabbix サーバーとZabbix エージェントが同じマシン上で動作する場合、サーバーの実行とエージェントの実行に、異なるユーザーを使用することをお勧めします。どちらも同じユーザーで実行されている場合、エージェントはサーバーの設定ファイルにアクセスでき、Zabbix 内のどの管理レベルのユーザーでも、極めて容易にデータベースのパスワード等を取得することができてしまいます。

設定ファイル

zabbix_server の設定の詳細は、設定ファイルを参照してください。

起動スクリプト

スクリプトは、システムの起動/シャットダウン中の Zabbix プロセスの自動起動/停止に使用されます。スクリプトは、misc/init.d ディレクトリの下にあります。

サーバープロセスタイプとスレッド

- agent poller - ワーカースレッドを備えた、パッシブチェック用の非同期ポーラープロセス
- alert manager - アラートキューマネージャー
- alert syncer - アラート DB ライター
- alerter - 通知送信用プロセス
- availability manager - ホストの可用性更新用プロセス
- browser poller - ブラウザアイテムチェック用ポーラー
- configuration syncer - 設定データのインメモリキャッシュを管理するプロセス
- configuration syncer worker - アイテム名内のユーザーマクロ値を解決して同期するプロセス
- connector manager - コネクター用マネージャープロセス
- connector worker - connector manager からのリクエストを処理するプロセス
- discovery manager - デバイスディスカバリー用マネージャープロセス
- discovery worker - discovery manager からのディスカバリータスクを処理するプロセス
- escalator - アクションのエスカレーション用プロセス
- ha manager - 高可用性を管理するプロセス
- history poller - データベース接続を必要とする計算チェックを処理するプロセス
- history syncer - 履歴 DB ライター
- housekeeper - 古いデータ（アイテムの履歴とトレンド、ユーザーセッション、イベントなど）および削除されたオブジェクトによって残されたデータを削除するプロセス
- http agent poller - ワーカースレッドを備えた、HTTP チェック用の非同期ポーラープロセス
- http poller - Web 監視ポーラー
- icmp pinger - icmpping チェック用ポーラー
- internal poller - 内部チェック用ポーラー
- ipmi manager - IPMI ポーラーマネージャー
- ipmi poller - IPMI チェック用ポーラー
- java poller - Java チェック用ポーラー
- lld manager - ローレベルディスカバリータスク用マネージャープロセス
- lld worker - ローレベルディスカバリータスク用ワーカープロセス
- odbc poller - ODBC チェック用ポーラー
- poller - パッシブチェック用の通常ポーラー
- preprocessing manager - 前処理ワーカースレッドを備えた前処理タスクのマネージャー
- preprocessing worker - データ前処理用スレッド
- proxy poller - パッシブプロキシ用ポーラー
- proxy group manager - プロキシの負荷分散と高可用性のマネージャー
- report manager- スケジュールレポート生成タスクのマネージャー
- report writer - スケジュールレポートを生成するプロセス
- self-monitoring - サーバー内部統計を収集するプロセス
- service manager - history syncer、task manager、alert manager から問題、問題タグ、および問題復旧に関する情報を受け取り、サービスを管理するプロセス
- snmp poller - ワーカースレッドを備えた、SNMP チェック用の非同期ポーラープロセス（walk[OID] および get[OID] アイテムのみ）
- snmp trapper - SNMP トラップ用トラッパー
- task manager - 他のコンポーネントから要求されたタスクをリモート実行するプロセス（例: 問題のクローズ、問題の確認、アイテム値の即時チェック、リモートコマンド機能）
- timer - メンテナンス処理用タイマー
- trapper - アクティブチェック、トラップ、プロキシ通信のためのトラッパー
- trigger housekeeper - その後削除されたトリガーによって生成された問題およびイベントを削除するプロセス
- unreachable poller - 到達不能なデバイス用ポーラー
- vmware collector - VMware サービスからのデータ収集を担当する VMware データコレクター

これらのプロセスタイプは、サーバーログファイルを使用して確認できます。

Zabbix 7.4.6 以降、サーバーログファイルはファイル所有者のみに読み書き権限を持つ状態で作成されます。さらに、このファイルは所有グループによる読み取りが可能です。その他の権限はすべて拒否されます。

さまざまな種類の Zabbix サーバープロセスは、内部アイテム **zabbix[process,<type>,<mode>,<state>]** を使用して監視できます。

history syncer トランザクション統計

history syncer プロセスタイトルには、history syncer トランザクションに関する詳細な統計が表示されます:

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

”A+B triggers” について:

- A - ヒストリ値に基づいて処理されたトリガー。
- B - タイマーに基づいて処理されたトリガー。

”processed...in N (<timings>) sec” で示される処理時間は以下のとおりです:

- アイテムの値をデータベースに書き込むのにかかった時間。
- アイテムデータ (状態、エラー、ホストインベントリなど) の更新にかかった時間。
- トレンドをデータベースにフラッシュするのにかかった時間。
- トリガーの計算にかかった時間。
- イベントとアクションの処理にかかった時間。

サポートするプラットフォーム

サーバーの処理のセキュリティ要件やミッションクリティカルな性質から、必要なパフォーマンス、フォールトトレランス、および復元力を一貫して提供できる OS は UNIX だけです。Zabbix は市場に出回っているバージョンで動作します。

Zabbix サーバーは以下のプラットフォーム上で動作確認済みです:

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

Note:
Zabbix は他の UNIX 系 OS でも動作する可能性があります。

ロケール

いくつかのテキスト形式のアイテムを正しく処理できるようにするため、サーバーには UTF-8 のロケールが必要であることに注意してください。最近のほとんどの UNIX ライクなシステムにはデフォルトで UTF-8 ロケールがありますが、システムによっては具体的に設定しなければ使用することができません。

1 高可用性 **(HA)**

概要

実質的にダウンタイムを発生させられない重要なインフラストラクチャーでは、高可用性 (HA) が求められます。したがって、停止してしまう可能性があるサービスについては、現在のサービスが停止してしまったときに処理を引き継げるようにするためのフェイルオーバーオプションが必要です。

Zabbix は、セットアップが簡単で、以前の HA の専門知識を必要としない組み込まれた高可用性のソリューションを提供します。組み込まれた Zabbix の HA は、Zabbix サーバーのソフトウェア/ハードウェアの障害に対する保護やメンテナンスに伴うダウンタイムを削減するのに役に立つでしょう。

Zabbix の高可用性モードでは、複数の Zabbix サーバーがクラスター内のノードとして実行されます。クラスター内の 1 台の Zabbix サーバーがアクティブである間、他のサーバーはスタンバイ状態となり、必要に応じて引き継ぐことができるよう準備されます。



Zabbix の HA への切り替えは戻すことが可能で、いつでもスタンドアロン環境に戻すことができます。

参照: 実装の詳細

高可用性の有効化

Zabbix サーバーをクラスターノードとして起動

Zabbix サーバーをクラスターノードとして起動するには、サーバー設定で 2 つのパラメーターを設定する必要があります。

- **HANodeName** パラメーターは、HA クラスターノードとなる Zabbix サーバーごとに指定する必要があります。

これは、エージェントおよびプロキシ設定でサーバーが参照される一意のノード識別子 (例: `zabbix-node-01`) です。HANodeName が指定されていない場合、サーバーはスタンドアロンモードで起動されます。

- **NodeAddress** パラメーターは、ノードごとに指定する必要があります。

NodeAddress パラメーター (アドレス: ポート番号) は、Zabbix フロントエンドがアクティブなサーバーノードに接続するために使用されます。NodeAddress は、それぞれの Zabbix サーバーの IP または FQDN 名と一致する必要があります。

設定ファイル変更後、すべての Zabbix サーバーを再起動します。これでクラスターノードとして機能します。サーバーの最新ステータスは、レポート → システム情報で確認できます。

```
zabbix_server -R ha_status
```

上記ランタイムコマンドは、現在の HA クラスターの状態を Zabbix サーバーログ (および stdout) に記録します。

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

Web インターフェースの準備

Web インターフェースの設定ファイル (Web インターフェースのファイルがあるディレクトリ内の `conf/zabbix.conf.php`) 内で、Zabbix サーバーのアドレス: ポート番号が 設定されていないことを確認してください。

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

Zabbix の Web インターフェースは、Zabbix データベースのノードテーブルから設定を読み取って、アクティブノードを自動的に検出します。アクティブノードのノードアドレスが Zabbix サーバーのアドレスとして使用されます。

Zabbix プロキシの設定

HA クラスターノード (サーバー) は、パッシブまたはアクティブモードの Zabbix プロキシの設定ファイルにリスト化されている必要があります。

パッシブプロキシの場合、ノード名をプロキシ設定ファイルの Serverパラメーターにカンマ区切りでリスト化する必要があります。

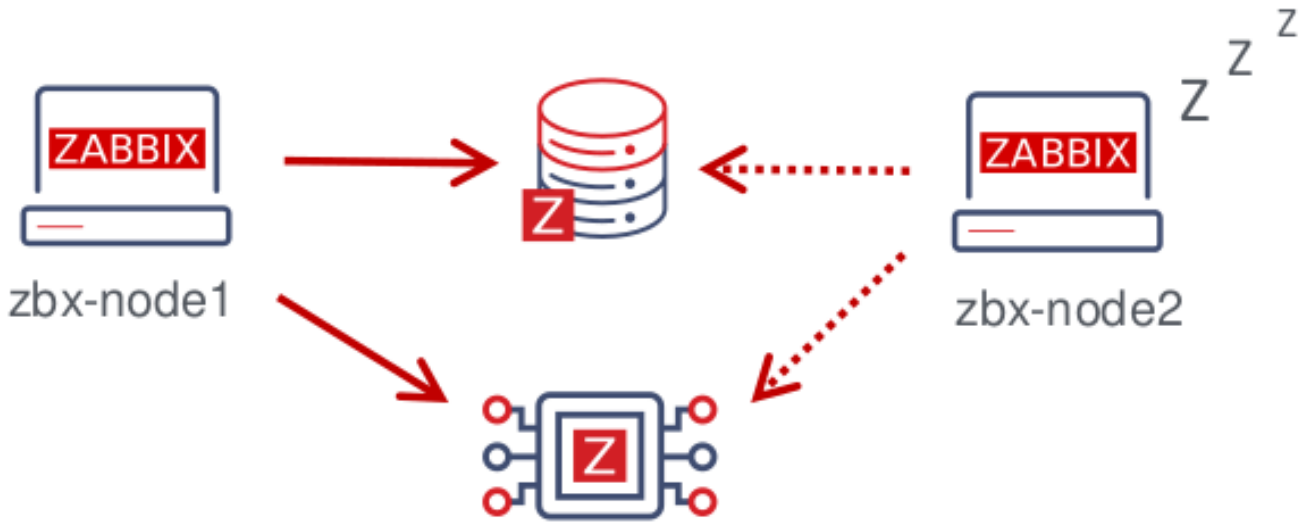
```
Server=zabbix-node-01,zabbix-node-02
```

アクティブプロキシの場合、ノード名をプロキシ設定ファイルの Serverパラメーターにセミコロン区切りでリスト化する必要があります。

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

Zabbix エージェントの設定

HA クラスターのノード (サーバー) は、Zabbix エージェントまたは Zabbix エージェント 2 の設定ファイルに並べて記述することが必要です。



パッシブチェックを行うのならば、ノード名を Serverパラメーターにカンマ区切りで設定する必要があります。

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

アクティブチェックを行うのならば、ノード名を ServerActiveパラメーターに設定する必要があります。ノードと他のサーバーとはカンマで、他のノードとはセミコロン区切ります。例:

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

スタンバイノードへのフェイルオーバー

アクティブなノードが停止すると、Zabbix は自動的に別のノードにフェイルオーバーします。フェイルオーバーが発生するには、少なくとも 1 つのノードがスタンバイ状態である必要があります。

フェイルオーバーはどのくらいの速さで実行されるか。すべてのノードは 5 秒ごとに最終アクセス時刻 (および変更されている場合はステータス) を更新します。したがって:

- アクティブノードがシャットダウンし、そのステータスが" 停止" と報告された場合、別のノードが **5** 秒以内に引き継ぎます。
- アクティブノードがシャットダウンするか、ステータスを更新できずに使用できなくなった場合、スタンバイノードはフェイルオーバー遅延 +5 秒待機して引き継ぎます。

フェイルオーバー遅延は設定可能で、サポートされる範囲は 10 秒から 15 分 (デフォルト 1 分) です。フェイルオーバー遅延の変更は次のコマンドを実行します:

```
zabbix_server -R ha_set_failover_delay=5m
```

HA クラスターの管理

HA クラスターの現在のステータスは、専用のランタイムコントロールオプションを使用して管理できます。

- `ha_status` - Zabbix サーバーログ (および標準出力) に HA クラスターのステータスを記録します。
- `ha_remove_node=target` - <target> で識別される HA ノードを削除します。<target> はノードの名前または ID であり、実行中の ha_status の出力から取得できます。例:

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

アクティブ/スタンバイノードは削除できないことに注意してください。

- `ha_set_failover_delay=delay` - HA フェイルオーバー遅延の設定 (10 秒から 15 分まで。タイムサフィックスがサポートされています。例: 10s, 1m)

ノードのステータスは次のように監視できます:

- レポート → システム情報
- システム情報のダッシュボードウィジェット
- サーバーの `ha_status` ランタイム制御オプション (上記を参照)

zabbix[cluster,discovery,nodes] の内部アイテムは、高可用性ノード情報を含む JSON を返すため、ノード検出に使用できます。

高可用性の無効化

高可用性クラスターを無効にするには:

- 設定ファイルのバックアップコピーを作成する
- スタンバイノードを停止する
- アクティブなプライマリサーバーから HANodeName パラメータを削除
- プライマリサーバーを再起動 (スタンドアロンモードで起動します)

HA クラスターのアップグレード

HA ノードのメジャーバージョンアップグレードを実行するには、次の手順を実行します:

- 全てのノードを停止する。
- データベースのフルバックアップを作成する。
- データベースがレプリケーションを使用している場合は、すべてのノードが同期していて問題がないことを確認してください。レプリケーションが壊れている場合はアップグレードしないでください。
- データベースのアップグレードを実行する単一ノードを選択し、HANodeName をコメントアウトしてその構成をスタンドアロンモードに変更して、アップグレードします。
- データベースのアップグレードが完了したことを確認します (システム情報には、Zabbix サーバーが実行中であると表示されます)。
- ノードを HA モードで再起動します。
- 残りのノードをアップグレードして起動します (この時点でデータベースはすでにアップグレードされているため、スタンドアロンモードに変更する必要はありません)。

マイナーバージョンのアップグレードでは、最初のノードをアップグレードし、それがアップグレードされて実行されていることを確認してから、次のノードでのアップグレードを開始するだけで十分です。

実装の詳細

高可用性 (HA) クラスターはオプトインソリューションであり、Zabbix サーバーでサポートされています。ネイティブ HA ソリューションは、簡単に使用できるように設計されており、複数のサイトで機能し、Zabbix が認識するデータベースに関する特定の要件はありません。ユーザーは、環境の高可用性要件に最適なものに応じて、ネイティブの Zabbix HA ソリューションまたはサードパーティの HA ソリューションを自由に使用できます。

このソリューションは複数の zabbix_server インスタンスまたはノードで構成されています。すべてのノードについて:

- 個別に構成
- 同じデータベースを使用
- いくつかのモードがあるかもしれません: アクティブ、スタンバイ、使用不可、停止

一度にアクティブ (稼働) できるノードは 1 つだけです。スタンバイノードは、HA マネージャーという 1 つのプロセスのみを実行します。スタンバイノードは、データの収集、処理、その他の通常のサーバーの仕事を行いません。ポートをリッスンしません。データベース接続は最小限です。

アクティブノードとスタンバイノードは両方とも、最終アクセス時刻を 5 秒ごとに更新します。各スタンバイノードは、アクティブノードの最終アクセス時刻を監視します。アクティブノードの最終アクセス時刻が'failover delay' 秒を超えている場合、スタンバイノードは自身をアクティブノードに切り替え、以前のアクティブノードに' 使用不可' ステータスを割り当てます。

アクティブノードは自身のデータベース接続を監視します。接続が failover delay-5 秒を超えて失われた場合は、すべての処理を停止してスタンバイモードに切り替ります。アクティブノードは、スタンバイノードのステータスも監視します。スタンバイノードの最終アクセス時刻が'failover delay' 秒を超えている場合、スタンバイノードには' 使用不可' ステータスが割り当てられます。

ノードは、Zabbix のマイナーバージョン間で互換性があるように設計されています。

2 Zabbix エージェント

概要

Zabbix エージェントは、ローカルリソースおよびアプリケーション（ハードドライブ、メモリ、プロセッサ統計情報など）をアクティブに監視するために、監視対象に配置されます。

エージェントは運用情報をローカルで収集し、さらに処理するためにデータを Zabbix サーバーへ報告します。障害が発生した場合 (ハードディスクの空き容量不足やサービスプロセスのクラッシュなど)、Zabbix サーバーは、その障害を報告した特定のマシンの管理者に対してアクティブにアラートを送信できます。

Zabbix エージェントは、統計情報の収集にネイティブなシステムコールを使用しているため、非常に高効率です。

パッシブチェックとアクティブチェック

Zabbix エージェントは、パッシブチェックとアクティブチェックを実行できます。

- パッシブチェック - Zabbix エージェントは、Zabbix サーバー（またはプロキシ）からのリクエストに応答します。
  例えば、サーバーがデータ（例: CPU 負荷）を要求し、エージェントが結果を返します。
- アクティブチェック - Zabbix エージェントは、Zabbix サーバー（またはプロキシ）からのリクエストを待たずにデータを収集して送信します。
  まず、監視アイテムの一覧をサーバーから取得し（CPU 負荷、利用可能メモリなど）、次に必要なデータを収集して、新しい値を定期的にサーバーへ送信します。

エージェントのチェックタイプは、対応する監視のアイテムタイプを選択して設定します。
Zabbix エージェントは、「Zabbix agent」または「Zabbix agent (active)」タイプのアイテムを処理します。

サポートするプラットフォーム

サポートするプラットフォームについては、要件ページを参照してください。

UNIX 系 OS 上の Zabbix エージェント

UNIX 系 OS 上の Zabbix エージェントは、監視対象のホストで実行されます。

インストール

Zabbix エージェントは、Linux ベースのシステムに以下のいずれかの方法でインストールできます。

- Zabbix パッケージ - Agent コンポーネントを選択し（Zabbix のバージョン、OS ディストリビューション、OS バージョンを選択した後）、表示される手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、`--enable-agent` オプションを指定してソースを設定することで、Zabbix エージェントをコンパイルします。

Attention:
一般的に、32 ビット版の Zabbix エージェントは 64 ビットシステムでも動作しますが、場合によっては失敗することがあります。

Note:
事前コンパイル済みの Zabbix エージェントバイナリは、macOS、IBM AIX、FreeBSD、OpenBSD、Solaris 向けにダウンロードできます。現在の Zabbix サーバー/プロキシのバージョンと互換性のあるレガシーバイナリは、NetBSDおよびHP-UX向けに利用できます。

パッケージとしてインストールした場合

Zabbix エージェントはデーモンプロセスとして実行されます。
エージェントは次を実行して起動できます:

```
systemctl start zabbix-agent
```

これは、ほとんどの GNU/Linux システムで動作します。
その他のシステムでは、次を実行する必要がある場合があります:

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

Zabbix エージェントの停止、再起動、またはステータスの確認には、次のコマンドを使用します:

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

手動で起動する

`zabbix_agentd` バイナリを見つけて直接実行することで、Zabbix エージェントを起動できます。例えば、次のようにします。

```
zabbix_agentd
```

Windows 上での Zabbix エージェント

Windows 上の Zabbix エージェントは、Windows サービスとして実行されます。

インストール

Zabbix エージェントは、以下のいずれかの方法で Windows にインストールできます。

- コンパイル済み Zabbix エージェントバイナリ - Zabbix エージェントの MSI インストーラーパッケージをダウンロードし、MSI からの Windows エージェントのインストール ページの手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、Windows での Zabbix エージェントのビルド ページの手順に従ってください。

ZIP アーカイブから Zabbix エージェントを Windows サービスとしてインストールする方法の詳細については、Microsoft Windows 上の Zabbix エージェント ページを参照してください。

オプション

1 台のホスト上で、複数のエージェントインスタンスを実行できます。単一のインスタンスでは、デフォルトの設定ファイル、またはコマンドラインで指定した設定ファイルを使用できます。複数インスタンスの場合、各エージェントインスタンスはそれぞれ専用の設定ファイルを持つ必要があります（いずれか 1 つのインスタンスはデフォルトの設定ファイルを使用できます）。

以下のコマンドラインパラメータを Zabbix エージェントで使用できます。

Parameter	Description
UNIX および **Windows** エージェント	
-c --config <config-file>	設定ファイルへのパス。 このオプションを使用して、デフォルト以外の設定ファイルを指定できます。 UNIX では、デフォルトは /usr/local/etc/zabbix_agentd.conf、またはcompile-time 変数 *--sysconfdir* または *--prefix* で設定された値です。 Windows では、デフォルトは C:\Program Files\Zabbix Agent\zabbix_agentd.conf です。
-f --foreground	Zabbix エージェントをフォアグラウンドで実行します（デフォルト: true）。
-p --print	既知のアイテムを表示して終了します。 注: user parameter の結果も返すには、設定ファイルを指定する必要があります（デフォルトの場所にない場合）。
-t --test <item key>	指定したアイテムをテストして終了します。 注: user parameter の結果も返すには、設定ファイルを指定する必要があります（デフォルトの場所にない場合）。
-T --test-config	設定ファイルを検証して終了します。
-h --help	ヘルプ情報を表示します。
-V --version	バージョン番号を表示します。
UNIX エージェントのみ	
-R --runtime-control <option>	管理機能を実行します。runtime control を参照してください。
Windows エージェントのみ	
-m --multiple-agents	複数のエージェントインスタンスを使用します（-i、-d、-s、-x オプションとともに使用）。 インスタンスのサービス名を区別するため、各サービス名には指定した設定ファイルの Hostname 値が含まれます。
-S --startup-type <value>	Zabbix Windows エージェントサービスの起動タイプを設定します。指定可能な値: `automatic` - *(デフォルト)* Windows 起動時にサービスを自動的に開始します。 `delayed` - 自動起動されるサービスの起動完了後に、サービスの開始を遅延させます（Windows Server 2008/Vista 以降で利用可能）。 `manual` - サービスを手動で開始します（ユーザーまたはアプリケーションによる）。 `disabled` - サービスを無効化し、ユーザーまたはアプリケーションが開始できないようにします。 このオプションは -i オプションと一緒に使用することも、すでにインストール済みのサービスの起動タイプを変更するために単独で使用することもできます。
-i --install	Zabbix Windows エージェントをサービスとしてインストールします。
-d --uninstall	Zabbix Windows エージェントサービスをアンインストールします。
-s --start	Zabbix Windows エージェントサービスを開始します。
-x --stop	Zabbix Windows エージェントサービスを停止します。

コマンドラインパラメータの具体的な例:

- 値付きですべての組み込みエージェントアイテムを表示する
- 指定した設定ファイルで定義された”mysql.ping” キーを持つ user parameter をテストする
- デフォルトの設定ファイルパス C:\Program Files\Zabbix Agent\zabbix_agentd.conf を使用して、Windows 用の”Zabbix Agent” サービスをインストールする
- エージェント実行ファイルと同じフォルダにある設定ファイル zabbix_agentd.conf を使用して、Windows 用の”Zabbix Agent [Hostname]” サービスをインストールし、設定ファイルの Hostname 値をサービス名に追加して一意にする
- エージェント実行ファイルと同じフォルダにある設定ファイル zabbix_agentd.conf を使用して、インストール済みの Windows 用”Zabbix Agent” サービスの起動タイプを変更する

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

実行時制御

実行時制御オプションを使用すると、エージェントプロセスのログレベルを変更できます。

Option	Description	Target
log_level_increase[=<target>]	ログレベルを上げます。 target が指定されていない場合、すべてのプロセスが対象になります。	target には以下を指定できます。 **process type** - 指定したタイプのすべてのプロセス（例: listener） すべてのエージェントのプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号（例: listener,3） **pid** - プロセス識別子（1～65535）。これより大きい値の場合は、target を'process-type,N' として指定してください。
log_level_decrease[=<target>]	ログレベルを下げます。 target が指定されていない場合、すべてのプロセスが対象になります。	
userparameter_reload	現在の設定ファイルから *UserParameter* および *Include* オプションの値を再読み込みします。	

例:

- すべてのプロセスのログレベルを上げる
- 3 番目の listener プロセスのログレベルを上げる
- PID 1234 のプロセスのログレベルを上げる
- すべてのアクティブチェックプロセスのログレベルを下げる

```
zabbix_agentd -R log_level_increase
zabbix_agentd -R log_level_increase=listener,3
zabbix_agentd -R log_level_increase=1234
zabbix_agentd -R log_level_decrease="active checks"
```

Note:
実行時制御は、OpenBSD、NetBSD、および Windows ではサポートされていません。

エージェントのプロセスタイプ

- `active checks` - アクティブチェックを実行するプロセス
- `collector` - データ収集を行うプロセス
- `listener` - パッシブチェックを待ち受けるプロセス

これらのプロセスタイプは、エージェントのログファイルを使用して確認できます。

Zabbix 7.4.6 以降、エージェントのログファイルはファイル所有者のみに読み書き権限を持つ状態で作成されます。さらに、このファイルは所有グループによる読み取りが可能です。その他の権限はすべて拒否されます。

実行ユーザー

UNIX 上の Zabbix エージェントは、root 以外のユーザーとして実行されるように設計されています。
起動時に指定された root 以外のユーザーとして実行されます。
そのため、エージェントは任意の root 以外のユーザーで問題なく実行できます。

'root' として実行しようとすると、ハードコードされた'zabbix' ユーザーに切り替わります。このユーザーはシステム上に存在している必要があります。
エージェントを'root' として実行できるのは、エージェント設定ファイル内の'AllowRoot' パラメータをそれに応じて変更した場合のみです。

設定ファイル

Zabbix エージェントの設定の詳細については、zabbix_agentd またはWindows agent の設定ファイルオプションを参照してください。

ロケール

一部のテキスト系エージェントアイテムが期待どおりの内容を返せるようにするため、エージェントでは UTF-8 ロケールが必要であることに注意してください。
最近のほとんどの Unix 系システムでは UTF-8 ロケールがデフォルトになっていますが、システムによっては明示的に設定する必要がある場合があります。

終了コード

Zabbix エージェントは、正常に終了した場合は 0 を返し、失敗した場合は 1 を返します。

3 Zabbix エージェント 2

概要

Zabbix エージェント 2 は、Go で記述された新世代のZabbix エージェント です（Zabbix エージェントの一部の C コードを再利用しています）。これは、以下を目的として設計されています。

- TCP 接続数を削減する。
- チェックの並行性を向上させる。
- プラグインによって容易に拡張できるようにする。プラグインは、最小限のコードでシンプルなチェックを提供し、長時間実行されるスクリプトや定期レポートを伴う独立したデータ収集で構成される複雑なチェックもサポートする。
- これまでのすべての機能をサポートし、Zabbix エージェントの置き換えとして機能する。

Zabbix エージェント 2 でサポートされるすべてのメトリクスは、プラグインによって収集されます。

パッシブチェックとアクティブチェック

Zabbix エージェント 2 は Zabbix エージェントと同様にパッシブおよびアクティブチェックをサポートしています。さらに、Zabbix エージェント 2 アクティブチェックは、柔軟な/スケジューリング間隔をサポートし、1 つのアクティブサーバー内の同時実行性をチェックします。

Note:
デフォルトでは、Zabbix エージェント 2 はリソース使用量の急増を防ぐために、アイテムの更新間隔内の条件付きランダム時間にアクティブチェックの最初のデータ収集をスケジュールします。エージェントの再起動直後に予定された更新確認間隔にないアクティブチェックを実行するには、設定ファイルで `ForceActiveChecksOnStart` パラメーター (グローバルレベル) または `Plugins.<プラグイン名>.System.ForceActiveChecksOnStart` (特定のプラグインチェックにのみ影響します) を設定します。プラグインレベルのパラメーターが設定されている場合、グローバルパラメーターを上書きします。

チェックの同時実行

異なるプラグインからのチェックは、同時に実行できます。1 つのプラグイン内で同時に実行されるチェック数は、プラグインの capacity 設定によって制限されます。各プラグインにはハードコードされた capacity 設定（デフォルトは 1000）がある場合があり、*Plugins* 設定パラメータの `Plugins.<PluginName>.System.Capacity=N` 設定を使用して、その値を下げることができます。

サポートするプラットフォーム

サポートするプラットフォームについては、要件ページを参照してください。

UNIX 系 OS 上の Zabbix エージェント 2

UNIX 系 OS 上の Zabbix エージェント 2 は、監視対象のホストで実行されます。

インストール

Zabbix エージェント 2 は、以下のいずれかの方法で Linux ベースのシステムにインストールできます。

- Zabbix パッケージ - Agent 2 コンポーネントを選択し（Zabbix のバージョン、OS ディストリビューション、OS バージョンを選択した後）、手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、`--enable-agent2` オプションを指定して設定を行い、エージェントをコンパイルします。

Note:
Zabbix エージェント 2 の監視機能は、別途提供されるロード可能なプラグインによって拡張できます。詳細については、ロード可能なプラグインを参照してください。

パッケージでインストールした場合

Zabbix エージェント 2 はフォアグラウンドプロセスとして実行され、バックグラウンド実行には外部サービスマネージャー (systemd など) に依存します。Zabbix エージェント 2 には Linux 上でのデーモン化サポートが組み込まれていません。

エージェントは、次のコマンドを実行して起動できます:

```
systemctl start zabbix-agent2
```

Zabbix エージェント 2 を停止、再起動、またはステータスを確認するには、次のコマンドを使用します:

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

手動で起動

zabbix_agent2 バイナリを見つけて直接実行することで、Zabbix エージェントを起動できます。例:

```
zabbix_agent2
```

Windows システム上のエージェント 2

Zabbix エージェント 2 はスタンドアロンプロセスとして実行されますが、Windows サービスとして実行することもできます。

インストール

Zabbix エージェント 2 は、Windows では次のいずれかの方法でインストールできます。

- コンパイル済み Zabbix エージェントバイナリ - エージェントの MSI インストーラーパッケージをダウンロードし、MSI からの Windows エージェントのインストール ページの手順に従ってください。
- Zabbix ソース - ソースファイルをダウンロードし、Windows での Zabbix エージェント 2 のビルド ページの手順に従ってください。

Note:
Zabbix エージェント 2 の監視機能は、別途提供されるロード可能なプラグインによって拡張できます。詳細については、ロード可能なプラグイン を参照してください。

Windows サービスとして Zabbix エージェント 2 を（ZIP アーカイブから）インストールする際の追加情報については、Microsoft Windows 上の Zabbix エージェント ページを参照してください。

オプション

Zabbix エージェント 2 では次のコマンドラインパラメーターを使用できます:

パラメーター	説明
UNIX と **Windows** エージェント	
-c --config <config-file>	設定ファイルへのパスを指定します。 このオプションを使用してデフォルトではない設定ファイルを指定できます。 UNIX の場合、デフォルトは/usr/local/etc/zabbix_agent2.conf であるか、コンパイル時の変数*--sysconfdir* または*--prefix* で設定されます。 Windows の場合、デフォルトは C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf です。
-f --foreground	Zabbix エージェントをフォアグラウンドで実行します (デフォルト: true)。
-p --print	既知のアイテムを出力して終了します。 注: ユーザーパラメーターの結果も返すには、(デフォルトの場所にない場合は) 設定ファイルを指定する必要があります。
-t --test <item key>	指定されたアイテムをテストして終了します。 注: ユーザーパラメーターの結果も返すには、(デフォルトの場所にない場合は) 設定ファイルを指定する必要があります。
-T --test-config	設定ファイルを検証して終了します。
-h --help	ヘルプを出力して終了します。
-v --verbose	デバッグ情報を出力します。このオプションは-p および-t オプションとともに使用します。
-V --version	エージェントのバージョン番号を出力して終了します。

パラメーター	説明
-R --runtime-control <option> **Windows** エージェントのみ	管理機能を実行します。ランタイム制御を参照してください。
-m --multiple-agents	複数のエージェントインスタンスを使用する (-i, -d, -s, -x オプションと一緒に)。 インスタンスのサービス名を区別するために、各サービス名には、指定された設定ファイルのホスト名の値が含まれます。
-S --startup-type <value>	Zabbix Windows エージェントサービスのスタートアップの種類を設定します。許可される値: `automatic` - *(デフォルト)* Windows 起動時にサービスを自動的に開始する。 `delayed` - 自動的に開始されるサービスが開始を完了するまで、サービスの開始を遅らせる。 `manual` - (ユーザーかアプリケーションが) サービスを手動で開始する。 `disabled` - サービスを無効にして、ユーザーまたはアプリケーションがサービスを開始できないようにします。 このオプションを-i オプションと一緒に使用することも、個別に使用して、すでにインストールされているサービスのスタートアップの種類を変更することもできます。
-i --install	Zabbix Windows エージェントをサービスとしてインストールします。
-d --uninstall	Zabbix Windows エージェントサービスをアンインストールします。
-s --start	Zabbix Windows エージェントサービスを開始します。
-x --stop	Zabbix Windows エージェントサービスを停止します。

コマンドラインパラメーターの使用の具体的な例:

- すべての組み込みエージェントアイテムの値を出力する
- 指定された設定ファイルで定義された”mysql.ping” キーを使用してユーザーパラメーターをテストする
- Windows の”Zabbix エージェント” サービスを設定ファイル C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf のデフォルトのパスを用いてインストールする
- エージェントの実行可能ファイルと同じフォルダーにある構成ファイル zabbix_agent2.conf を使用して、インストールされている Windows の”Zabbix Agent” サービスのスタートアップの種類を変更する

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

ランタイム制御

ランタイム制御は、リモート制御のためのいくつかのオプションを提供します。

オプション	説明
log_level_increase	ログレベルを上げます。
log_level_decrease	ログレベルを下げます。
metrics	利用可能なメトリックを一覧表示します。
version	エージェントのバージョンを表示します。
userparameter_reload	現在の設定ファイルから *UserParameter* および *Include* オプションの値を再ロードします。
help	ランタイムコントロールのヘルプを表示します。

例:

- エージェント 2 のログレベルを上げる
- ランタイム制御オプションを出力する

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

設定ファイル

Zabbix agent 2 の設定パラメータは、一部の例外を除き、ほとんどが Zabbix エージェントのものと一致しています。

パラメータ		説明
Zabbix agent 2 固有の設定パラメータ		
	ControlSocket	実行時制御ソケットのパスです。Zabbix agent 2 は、実行時コマンドに制御ソケットを使用します。
	EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	これらのパラメータは、アクティブアイテム向けに Zabbix agent 2 の永続ストレージを設定するために使用されます。
	ForceActiveChecksOnStart	再起動後にエージェントがアクティブチェックを直ちに実行するか、時間をかけて均等に分散して実行するかを決定します。
	Plugins	プラグインは、`Plugins.<Plugin name>.<Parameter>=<value>` の形式で独自のパラメータを持つ場合があります。共通のプラグインパラメータとして `System.Capacity` があり、同時に実行できるチェック数の上限を設定します。
	StatusPort	HTTP ステータス要求を受け付け、設定済みプラグインの一覧および一部の内部パラメータを表示するために、Zabbix agent 2 が待ち受けるポートです。
Zabbix エージェント固有の設定パラメータで、Zabbix agent 2 ではサポートされないもの		
	AllowRoot, *User*	デーモン化をサポートしていないため、Zabbix agent 2 ではサポートされません。
	EnableRemoteCommands	非推奨です。代わりに、両方のエージェントで `AllowKey` および `DenyKey` パラメータが使用されます。
	ListenBacklog	Zabbix エージェントでは、TCP キュー内の保留中接続の最大数を定義するために使用されます。Zabbix agent 2 ではサポートされません。
	LoadModule, *LoadModulePath*	ロード可能なモジュールは、Zabbix agent 2 ではサポートされません。
	MaxLinesPerSecond, *LogRemoteCommands*	Zabbix agent 2 では、代わりに `Plugins.Log.MaxLinesPerSecond` および `Plugins.SystemRun.LogRemoteCommands` が使用されます。

パラメータ	説明
StartAgents	このパラメータは、Zabbix エージェントでパッシブチェックの並行性を高める、または無効にするために使用されます。Zabbix agent 2 では、並行性はプラグインレベルで設定され、capacity 設定によって制限できます。Server パラメータが指定されていない場合、Zabbix Agent 2 ではパッシブチェックは無効になります。

詳細については、Zabbix agent2 の設定ファイルオプションを参照してください。

終了コード

Zabbix エージェント 2 は古い OpenSSL バージョン (1.0.1、1.0.2) でコンパイルすることもできます。

この場合、Zabbix は OpenSSL でロックするためのミューテックスを提供します。ミューテックスのロックまたはロック解除が失敗した場合、エラーメッセージが標準エラーストリーム (STDERR) に出力され、エージェント 2 はそれぞれリターンコード 2 または 3 で終了します。

4 Zabbix プロキシ

概要

Zabbix プロキシは、1 つ以上の監視対象から監視データを収集し、その情報を Zabbix サーバーに送信するプロセスであり、基本的にサーバーに代わって機能します。収集されたすべてのデータはローカルにバッファリングされてから、プロキシが属する Zabbix サーバーに転送されます。

プロキシのデプロイはオプションですが、単一の Zabbix サーバーの負荷を分散するのに非常に役立つ場合があります。プロキシがデータを収集する場合、サーバーの CPU とディスク I/O の負荷が少なくなります。

Zabbix プロキシはローカル管理者がいなくても、リモートロケーション、ブランチ、ネットワークを一元的に監視するための理想的なソリューションです。

Zabbix プロキシにはサーバーとは別のデータベースが必要です。

> **Attention:**
> Zabbix プロキシでサポートされているデータベースは SQLite、MySQL、PostgreSQL であることに注意してください。

参照: 分散環境でのプロキシの使用

プロキシの起動

パッケージを使用してインストールした場合

Zabbix プロキシはデーモンプロセスとして実行されます。プロキシを起動するには以下を実行します。

```
systemctl start zabbix-proxy
```

これはほとんどの GNU/Linux システムで動きます。他のシステムでは、以下のように実行する必要がある場合があります。

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

同様に、Zabbix プロキシの停止/再起動/ステータス表示するには以下を実行します。

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

手動での起動

上記の方法で起動できない場合は手動で起動する必要があります。zabbix_proxy バイナリへのパスを見つけ、以下のように実行します:

```
zabbix_proxy
```

Zabbix プロキシでは、以下のコマンドラインパラメーターを使用できます:

```
-c --config <file>              設定ファイルへのパスを指定する
-f --foreground                 Zabbix プロキシをフォアグラウンドで実行する
-R --runtime-control <option>   管理機能を実行する
-T --test-config                設定ファイルを検証して終了する
-h --help                       ヘルプを表示する
-V --version                    バージョン情報を表示する
```

コマンドラインパラメーターを使用して Zabbix プロキシを実行する例:

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

ランタイム制御

ランタイム制御オプション:

オプション	説明	ターゲット
config_cache_reload	設定キャッシュをリロードします。キャッシュが現在ロードされている場合は無視されます。 アクティブな Zabbix プロキシは Zabbix サーバーに接続し、設定データを要求します。 パッシブ Zabbix プロキシは、次回サーバーがプロキシに接続するときに、Zabbix サーバーから設定データを要求します。	
history_cache_clear[=<**target**>]	IDで指定されたアイテムのヒストリキャッシュをクリアします。 最初と最後の値を除き、アイテムのすべての値に影響します。	**target** - アイテムの ID
diaginfo[=<**section**>]	プロキシログファイルに診断情報を収集します	**historycache** - ヒストリキャッシュ統計 **preprocessing** - 保存前処理マネージャー統計 **locks** - ミューテックスリスト (*BSD* システムでは空)
snmp_cache_reload	SNMP キャッシュをリロードします。すべてのホストの SNMP プロパティ (engine time, engine boots, engine id, credentials) をクリアします。SNMP の問題のトラブルシューティングを行うときに、グローバルキャッシュを強制的にクリアするために使用します。	
housekeeper_execute	ハウスキーピングを開始します。ハウスキーピングが現在進行中の場合は無視されます。	
log_level_increase[=<**target**>]	ログレベルを上げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 *BSD* システムではサポートされていません。	**process type** - 指定されたタイプのすべてのプロセス (例: poller) すべてのプロキシプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号 (例: poller,3) **pid** - プロセス ID(1〜65535)。値が大きい場合は、ターゲットを'process type,N' として指定します。
log_level_decrease[=<**target**>]	ログレベルを下げます。ターゲットが指定されていない場合はすべてのプロセスに影響します。 *BSD* システムではサポートされていません。	

オプション	説明	ターゲット
prof_enable[=<**target**>]	プロファイリングを有効にします。 ターゲットが指定されていない場合はすべてのプロセスに影響します。 有効なプロファイリングでは、関数名ごとにすべての rwlocks/mutex の詳細が提供されます。	**process type** - 指定されたタイプのすべてのプロセス (例: history syncer) すべてのプロキシプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号 (例: history syncer,1) **pid** - プロセス ID(1〜65535)。値が大きい場合は、ターゲットを'process type,N' として指定します。 **scope** - `rwlock`、`mutex`、`processing` はプロセスタイプと番号とともに (例: history syncer,1,processing)、あるいは全てのプロセスタイプとともに (例: history syncer,rwlock) 使用できます。
prof_disable[=<**target**>]	プロファイリングを無効にします。 ターゲットが指定されていない場合はすべてのプロセスに影響します。	**process type** - 指定されたタイプのすべてのプロセス (例: history syncer) すべてのプロキシプロセスタイプを参照してください。 **process type,N** - プロセスタイプと番号 (例: history syncer,1) **pid** - プロセス ID(1〜65535)。値が大きい場合は、ターゲットを'process type,N' として指定します。

ランタイム制御を使用してプロキシ設定キャッシュをリロードする例:

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

ランタイム制御を使用してアイテムのヒストリキャッシュをクリアする例:

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

ランタイム制御を使用して診断情報を収集する例:

```
### プロキシログファイルに利用可能なすべての診断情報を収集する:
zabbix_proxy -R diaginfo

### プロキシログファイルにヒストリキャッシュ統計を収集する:
zabbix_proxy -R diaginfo=historycache
```

ランタイム制御を使用して SNMP キャッシュをリロードする例:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
SNMPv3 インターフェースが Zabbix UI 経由で更新されると、ほとんどの場合、Zabbix はそのインターフェースの新しい SNMPv3 資格情報を自動的に再ロードします。`-R snmp_cache_reload` は、資格情報の変更後もポーリングが失敗する場合 (たとえば、engineBoots/engineID の不一致や非 RFC デバイスが原因)、またはトラブルシューティングのためにグローバル SNMP キャッシュを強制的にクリアする必要がある場合にのみ使用してください。

ランタイム制御を使用してハウスキーパーの実行をトリガーする例:

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

ランタイム制御を使用してログレベルを変更する例:

```
### すべてのプロセスのログレベルを上げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### 2 番目のポーラープロセスのログレベルを上げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### PID 1234 のプロセスのログレベルを上げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234
```

```
### すべての http ポーラープロセスのログレベルを下げる:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

プロセスユーザー

Zabbix プロキシは root 以外のユーザーとして実行するように設計されていて、起動された root 以外のユーザーとして実行されます。したがって root 以外のユーザーで問題なくプロキシを起動できます。

'root' として実行しようとすると、ハードコードされた'zabbix' ユーザーに切り替わるため、'zabbix' ユーザーがシステム上に存在している必要があります。'root' として実行するためには、プロキシの設定ファイルの'AllowRoot' パラメーターを適切に変更する必要があります。

設定ファイル

zabbix_proxy の詳細設定については設定ファイルオプションを参照してください。

プロキシのプロセスタイプとスレッド

- agent poller - ワーカースレッドを持つ、パッシブチェック用の非同期ポーラープロセス
- availability manager - ホストの可用性更新用プロセス
- browser poller - ブラウザアイテムチェック用ポーラー
- configuration syncer - 設定データのインメモリキャッシュを管理するプロセス
- data sender - プロキシデータ送信プロセス
- discovery manager - デバイスディスカバリ用の管理プロセス
- discovery worker - discovery manager からのディスカバリタスクを処理するプロセス
- history syncer - 履歴 DB ライター
- housekeeper - 古いアイテム履歴を削除するプロセス
- http agent poller - ワーカースレッドを持つ、HTTP チェック用の非同期ポーラープロセス
- http poller - Web 監視ポーラー
- icmp pinger - icmpping チェック用ポーラー
- internal poller - 内部チェック用ポーラー
- ipmi manager - IPMI ポーラーマネージャー
- ipmi poller - IPMI チェック用ポーラー
- java poller - Java チェック用ポーラー
- odbc poller - ODBC チェック用ポーラー
- poller - パッシブチェック用の通常ポーラー
- preprocessing manager - 前処理ワーカースレッドを持つ前処理タスクの管理プロセス
- preprocessing worker - データ前処理用スレッド
- self-monitoring - サーバー内部統計を収集するプロセス
- snmp poller - ワーカースレッドを持つ、SNMP チェック用の非同期ポーラープロセス（walk[OID] および get[OID] アイテムのみ）
- snmp trapper - SNMP トラップ用トラッパー
- task manager - 他のコンポーネントから要求されたタスクをリモート実行するプロセス（例: 問題のクローズ、問題の確認、アイテム値の即時チェック、リモートコマンド機能）
- trapper - アクティブチェック、トラップ、プロキシ通信のためのトラッパー
- unreachable poller - 到達不能なデバイス用ポーラー
- vmware collector - VMware サービスからのデータ収集を担当する VMware データコレクター

これらのプロセスタイプは、プロキシのログファイルを使用して確認できます。

Zabbix 7.4.6 以降、プロキシのログファイルはファイル所有者のみに読み書き権限を持つ状態で作成されます。さらに、このファイルは所有グループによる読み取りが可能です。その他の権限はすべて拒否されます。

さまざまな種類の Zabbix プロキシプロセスは、内部item **zabbix[process,<type>,<mode>,<state>]** を使用して監視できます。

history syncer トランザクション統計

history syncer プロセスタイトルには、history syncer トランザクションに関する詳細な統計が表示されます:

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

"processed...in N (<timings>) sec" で示される処理時間は以下のとおりです:

- アイテムの値をデータベースに書き込むのにかかった時間。
- アイテムデータ (状態、エラー) の更新にかかった時間。

サポートするプラットフォーム

Zabbix プロキシは、Zabbix サーバーと同じサポートするプラットフォームのリストに記載されているプラットフォームで実行可能です。

メモリバッファ

メモリバッファを使用すると、データベースにアクセスせずに新しいデータ (アイテムの値、ネットワークディスカバリ、ホスト自動登録) をバッファに保存し、Zabbix サーバーにアップロードできます。Zabbix 7.0 から、プロキシにメモリバッファが導入されました。

Zabbix 7.0 より前のインストールでは、収集されたデータは Zabbix サーバーにアップロードされる前にデータベースに保存されました。これらのインストールでは、Zabbix 7.0 にアップグレードした後もこれがデフォルトの動作となります。

フォーマンスを最適化するには、プロキシ上でメモリバッファの使用を構成することをお勧めします。ProxyBufferModeの値を”disk”(既存のインストールのハードコードされたデフォルト) から”hybrid”(推奨) か”memory” に変更することで可能になります。メモリバッファサイズ (ProxyMemoryBufferSizeパラメーター) を設定することも必要です。

ハイブリッドモードでは、プロキシが停止している場合、バッファがいっぱいである場合、またはデータが古すぎる場合に、未送信のデータをデータベースにフラッシュすることでバッファをデータ損失から保護します。すべての値がデータベースにフラッシュされると、プロキシはメモリバッファの使用に戻ります。

メモリモードではメモリバッファが使用されますが、データ損失に対する保護はありません。プロキシが停止した場合、またはメモリが過剰になった場合、未送信のデータは破棄されます。

ハイブリッドモード (ProxyBufferMode=hybrid) は Zabbix 7.0 以降のすべての新規インストールに適用されます。

ProxyMemoryBufferSizeやProxyMemoryBufferAgeなどの追加パラメーターは、それぞれメモリバッファサイズとバッファ内のデータの最大存続期間を定義します。

注意: 構成が競合すると、プロキシはエラーを出力し起動に失敗します。たとえば次のような場合です:

- ProxyBufferMode が”hybrid” か”memory” であり、ProxyMemoryBufferSize が”0” の場合。
- ProxyBufferMode が”hybrid” か”memory” であり、ProxyLocalBuffer”0” でない場合。

ロケール

いくつかのテキスト形式のアイテムを正しく処理できるようにするため、プロキシには UTF-8 のロケールが必要であることに注意してください。最近のほとんどの UNIX ライクなシステムにはデフォルトで UTF-8 ロケールがありますが、システムによっては具体的に設定しなければ使用することができません。

メンテナンス中のキューの計算

> **Attention:**
> Zabbix プロキシはメンテナンス期間を認識しません。詳細については、メンテナンス中のキューの計算を参照してください。

5 Zabbix Java ゲートウェイ

概要

Zabbix Java ゲートウェイは、ソースコードまたはパッケージからインストールできます。

JMX アプリケーションを監視するためのネイティブサポートは、”Zabbix Java ゲートウェイ” と呼ばれる Zabbix デーモン形式で存在します。Zabbix Java ゲートウェイは Java で書かれたデーモンです。ホスト上の特定の JMX カウンター値を見つけるために、Zabbix サーバーは Zabbix Java ゲートウェイでクエリを実行します。Zabbix Java ゲートウェイは、JMX 管理 APIを使用して対象のアプリケーションへリモートでクエリを実行します。アプリケーションに追加のソフトウェアをインストールする必要はありません。コマンドラインで`-Dcom.sun.management.jmxremote` オプションを使用して起動するだけです。

Java ゲートウェイは、Zabbix サーバーまたはプロキシからの着信接続を受け入れ、” パッシブプロキシ” としてのみ使用できます。Zabbix プロキシとは違い、Zabbix プロキシからも使用できます (Zabbix プロキシは Zabbix プロキシにチェーンできません)。各 Java ゲートウェイへのアクセスは、Zabbix サーバーまたはプロキシ設定ファイルで直接設定されるため、Zabbix サーバーまたは Zabbix プロキシごとに設定できる Java ゲートウェイは 1 つだけです。ホストに **JMX** エージェントタイプのアイテムとその他のタイプのアイテムがある場合、**JMX** エージェントアイテムのみが取得のために Java ゲートウェイに渡されます。

Java ゲートウェイ経由でアイテムを更新する必要がある場合、Zabbix サーバーまたはプロキシは Java ゲートウェイに接続して値を要求し、Java ゲートウェイはその値を取得してサーバーまたはプロキシに返します。そのため、Java ゲートウェイは値をキャッシュしません。

Zabbix サーバーまたはプロキシには、オプション **StartJavaPollers** によって制御される、Java ゲートウェイに接続する特定のタイプのプロセスがあります。内部的には、Java ゲートウェイは **START_POLLERS**オプションによって制御される複数のスレッドを開始します。サーバー側では、接続に **Timeout** 秒以上かかると終了しますが、Java ゲートウェイは JMX カウンターから値を取得する

ためにまだビジーである可能性があります。これを解決するために、Java ゲートウェイには **TIMEOUT** オプションがあり、JMX ネットワーク操作のタイムアウトを設定できます。

Zabbix サーバーまたはプロキシはパフォーマンスを向上させるために、単一の JMX ターゲットへのリクエストを可能な限りプールし (アイテム間隔の影響を受けます)、それらを単一の接続で Java ゲートウェイに送信しようとします。

StartJavaPollers を **START_POLLERS** 以下にすることをお勧めします。そうしないと、受信リクエストを処理するために、Java ゲートウェイでスレッドが利用できない状況が発生する可能性があります。このような場合、Java ゲートウェイは ThreadPoolExecutor.CallerRunsPolicy を使用します。これは、メインスレッドが受信リクエストを処理し、一時的に新しいリクエストを受け入れないことを意味します。

Zabbix Java ゲートウェイを使用して Wildfly ベースの Java アプリケーションを監視しようとしている場合は、Wildfly のダウンロードページで入手可能な最新の jboss-client.jar をインストールしてください。

2 RHEL パッケージでのセットアップ

概要

次の情報は、パッケージでインストールした場合の Zabbix Java ゲートウェイセットアップに役立ちます。

Java ゲートウェイの設定と実行

Zabbix Java ゲートウェイの設定パラメーターは、次のファイルで調整できます。

`/etc/zabbix/zabbix_java_gateway.conf`

詳細については、Zabbix Java ゲートウェイ設定パラメーターを参照してください。

Zabbix Java ゲートウェイを起動するには:

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

サーバー起動時に Zabbix Java ゲートウェイを自動的に起動するには:

RHEL 7 以降:

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

RHEL 7 より前:

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

Java ゲートウェイを使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これはサーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメーターを指定します。

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

デフォルトでは、サーバーは JMX 監視に関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです。

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

ログの内容を増やしたい場合は、下記ファイルを編集します。

`/etc/zabbix/zabbix_java_gateway_logback.xml`

`level="info"` を”debug” または”trace” に変更します (詳細なトラブルシューティング用):

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

```
</configuration>
```

JMX 監視

詳しくはJMX 監視ページをご覧ください。

3 Debian/Ubuntu パッケージでのセットアップ

概要

次の情報は、パッケージでインストールした場合の Zabbix Java ゲートウェイセットアップに役立ちます。

Java ゲートウェイの設定と実行

Java ゲートウェイの設定は次のファイルで行えます。

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

詳細については、Zabbix Java ゲートウェイ設定パラメーターを参照してください。

Zabbix Java ゲートウェイを起動するには:

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

サーバー起動時に Zabbix Java ゲートウェイを自動的に起動するには:

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

Java ゲートウェイを使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これはサーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメーターを指定します。

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

デフォルトでは、サーバーは JMX 監視に関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

```
StartJavaPollers=5
```

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Zabbix Java ゲートウェイのログファイルは次のとおりです。

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

ログの内容を増やしたい場合は、下記ファイルを編集します。

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

`level="info"` を”debug” または”trace” に変更します (詳細なトラブルシューティング用):

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

</configuration>
```

JMX 監視

詳しくはJMX 監視ページをご覧ください。

ソースからのセットアップ

概要

ソースからインストールした場合、次の情報は Zabbix Java ゲートウェイのセットアップに役立ちます。

ファイルの概要

ソースから Java ゲートウェイを取得した場合、シェルスクリプト、JAR、および設定ファイルのコレクションが $PREFIX/sbin/zabbix_java の下にあるはずです。これらのファイルの役割を以下にまとめます。

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

Java ゲートウェイ JAR ファイル本体

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

Java ゲートウェイの依存関係: Logback、SLF4J、および Android JSONライブラリ。

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

Logback の構成ファイル。

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

Java ゲートウェイを開始および停止するための便利なスクリプト。

`settings.sh`

上記の起動スクリプトとシャットダウンスクリプトによって提供される設定ファイル。

Java ゲートウェイの設定と実行

デフォルトでは、Java ゲートウェイはポート 10052 でリッスンします。別のポートで Java ゲートウェイを実行したい場合は、settings.sh スクリプトで指定できます。このオプションやその他のオプションの指定方法については、Java ゲートウェイ設定ファイルの説明を参照してください。

> **Warning:**
> ポート 10052 はIANA 登録済みではありません。

設定完了後、起動スクリプトを実行して Java ゲートウェイを起動できます。

`./startup.sh`

同様に Java ゲートウェイが不要になったら、シャットダウンスクリプトを実行して停止できます。

`./shutdown.sh`

サーバーやプロキシとは異なり、Java ゲートウェイは軽量でありデータベースを必要としません。

Java ゲートウェイを使用するためのサーバー設定

Java ゲートウェイが稼働している状態で、Zabbix サーバーに Zabbix Java ゲートウェイの場所を伝える必要があります。これはサーバー設定ファイルで JavaGateway および JavaGatewayPort パラメーターを指定することによって行われます。JMX アプリケーションが実行されているホストが Zabbix プロキシによって監視されている場合は、代わりにプロキシ設定ファイルで接続パラメーターを指定します。

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

デフォルトでは、サーバーは JMX 監視に関連するプロセスを開始しません。使用する場合は、事前にフォークされた Java ポーラーのインスタンスの数を指定する必要があります。これは通常のポーラーとトラッパーを指定するのと同じ方法で行います。

`StartJavaPollers=5`

設定が完了したら、サーバーまたはプロキシを再起動してください。

Java ゲートウェイのデバッグ

Java ゲートウェイに問題がある場合、またはフロントエンドのアイテムについて表示されるエラーメッセージが十分でない場合は、Java ゲートウェイのログファイルを確認することをお勧めします。

デフォルトでは、Java ゲートウェイはそのアクティビティを/tmp/zabbix_java.log ファイルにログレベル”info” で記録します。その情報だけでは不十分な場合、ログレベル”debug” の情報を取得する必要があります。ログレベルを上げるには、lib/logback.xml ファイルを変更し、<root> タグの level 属性を”debug” に変更します。

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

Zabbix サーバーや Zabbix プロキシとは異なり、logback.xml ファイルを変更した後に Zabbix Java ゲートウェイを再起動する必要はありません。logback.xml の変更は自動的に取得されます。デバッグが完了したら、ログレベルを”info” に戻すことができます。

別のファイルまたはデータベースのようなまったく別のメディアにログを記録する場合は、必要に応じて logback.xml ファイルを調整します。詳細については、Logback Manualを参照してください。

デバッグ目的の場合、デーモンではなくコンソールアプリケーションとして Java ゲートウェイを起動すると便利な場合があります。これを行うには、settings.sh で PID_FILE 変数をコメントアウトします。PID_FILE を省略した場合、startup.sh スクリプトは Java ゲートウェイをコンソールアプリケーションとして起動し、代わりに Logback に lib/logback-console.xml ファイルを使用させます。これにより、コンソールにログが記録されるだけでなく、ログレベル”debug” も有効になります。

最後に、Java ゲートウェイはロギングに SLF4J を使用するため、適切な JAR ファイルを lib ディレクトリに配置することで、Logback を選択したフレームワークに置き換えることができます。詳細については、SLF4J Manualを参照してください。

JMX 監視

詳しくはJMX 監視ページをご覧ください。

6 zabbix_sender コマンド

概要

Zabbix sender は、処理のためにパフォーマンスデータを Zabbix サーバーへ送信するために使用できるコマンドラインユーティリティです。

このユーティリティは通常、可用性データおよびパフォーマンスデータを定期的に送信するための長時間実行されるユーザースクリプトで使用されます。

結果を Zabbix サーバーまたはプロキシへ直接送信するには、trapper アイテム タイプを設定する必要があります。

組み込みで Zabbix sender のように動作する機能を持つZabbix 用 Python ライブラリ も参照してください。

Zabbix sender の実行

Zabbix UNIX sender の実行例:

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

解説:

- z - Zabbix サーバーのホスト (IP アドレスも使用可能)
- s - 監視対象ホストの名称 (Zabbix フロントエンドに登録されているもの)
- k - アイテムキー
- o - 送信値

> **Attention:**
> 空白を含むオプションは、ダブルクォーテーションで囲む必要があります。

Zabbix sender を使用して入力ファイルから複数の値を送信できます。詳細については、Zabbix sender のマニュアルページを参照してください。

設定ファイルが指定されている場合、Zabbix sender はエージェントの ServerActive 設定パラメータで定義されているすべてのアドレスを使用してデータを送信します。バッチデータの送信が 1 つのアドレスへの送信で失敗した場合、他のアドレスへの送信を試み、次のバッチはこのアドレスに送信されません。

Zabbix sender はファイルの最初にバイト順マーク (BOM) がない UTF-8 エンコーディング (UNIX ライクなシステムと Windows の両方) の文字列を受け入れます。

Windows 上の Zabbix sender も同様に実行できます:

```
zabbix_sender.exe [options]
```

zabbix_sender のリアルタイム送信シナリオは、連続して渡される複数の値を収集し、単一の接続でサーバーに送信します。前の値から 0.2 秒以内の値を同じスタックに入れることができますが、最大ポーリング時間は依然として 1 秒です。

> **Note:**
> 指定された設定ファイルに無効な (*parameter=value* 表記に従っていない) パラメータエントリが存在する場合、Zabbix sender は終了します。

ローレベルディスカバリで Zabbix sender を実行する

ローレベルディスカバリの JSON 形式の値を送信する Zabbix sender を実行する例:

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

これが機能するには、ローレベルディスカバリルールに Zabbix トラッパーアイテムタイプ (この例では、`trapper.discovery.item` キー) が必要です。

7 zabbix_get コマンド

概要

Zabbix get はコマンドラインユーティリティであり、Zabbix エージェントと通信して、エージェントから必要な情報を取得するために使用できます。

このユーティリティは通常、Zabbix エージェントのトラブルシューティングに使用されます。

Zabbix get と同様の機能が組み込まれているZabbix 用 Python ライブラリ も参照してください。

Zabbix get の実行

UNIX で Zabbix get を実行して、エージェントからプロセッサ負荷値を取得する例:

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

Zabbix get を実行して、Web サイトから文字列をキャプチャする例:

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

このアイテムキーにはスペースが含まれているため、シェルにアイテムキーを引き渡すのにコーテーションが使用されていることに注意してください。コーテーションはアイテムキーの一部ではなく、シェルによってトリミングされ、Zabbix エージェントには渡されません。

アイテムキーがサポートされていない場合、Zabbix get は終了コード 1 を返します。

Zabbix get は、次のコマンドラインパラメーターを受け入れます。

```
-s --host <host name or IP>      ホスト名またはホストの IP アドレスを指定します
-p --port <port number>          ホスト上で動作しているエージェントのポート番号を指定します (デフォルト: 10050)
-I --source-address <IP address> ソースの IP アドレスを指定します
-t --timeout <seconds>           タイムアウトを指定します。有効な範囲: 1-600 秒 (デフォルト: 600 秒)
-k --key <item key>              値を取得したいアイテムのキーを指定します
-P --protocol <value>     エージェントとの通信に使用されるプロトコル。値:
    auto - JSON プロトコルを使用して接続し、平文プロトコルでフォールバックと再試行します (デフォルト)
    json - JSON プロトコルを使用して接続します
    plaintext - アイテムキーのみが送信される平文プロトコルを使用して接続します (6.4.x 以前のリリース)
-h --help                        ヘルプを表示します
-V --version                     バージョン番号を表示します

--tls-connect <value>    エージェントに接続する方法。値:
    unencrypted - 暗号化せずに接続する (デフォルト)
    psk - TLS と事前共有キーを使用して接続する
    cert - TLS と証明書を使用して接続する
--tls-ca-file <CA file>     ピア証明書検証用のトップレベル CA 証明書を含むファイルのフルパス名
--tls-crl-file <CRL file>    取り消された証明書を含むファイルのフルパス名
--tls-agent-cert-issuer <cert issuer>    許可されたエージェント証明書発行者
--tls-agent-cert-subject <cert subject>    許可されたエージェント証明書の件名
```

```
--tls-cert-file <cert file>     証明書または証明書チェーンを含むファイルのフルパス名
--tls-key-file <key file>     秘密キーを含むファイルのフルパス名
--tls-psk-identity <PSK-identity>     事前共有キーを識別するために使用される、大文字と小文字を区別する一意の文字列
--tls-psk-file <PSK-file>     事前共有キーを含むファイルのフルパス名
--tls-cipher13 <cipher-string>     OpenSSL 1.1.1 以降の TLS 1.3 の暗号文字列。デフォルトの暗号スイートの選択基準をオー
--tls-cipher <cipher-string>     GnuTLS 優先順位文字列（TLS 1.2 以降）または OpenSSL 暗号文字列（TLS 1.2 のみ）。デ
```

詳細については、Zabbix get のマニュアルページを参照してください。

Windows 上でも Zabbix get は同様に実行できます:

```
zabbix_get.exe [options]
```

8 zabbix_js コマンド

概要

zabbix_js は、埋め込みスクリプトのテストに使用できるコマンドラインユーティリティです。

このユーティリティは、文字列パラメータを使用してユーザースクリプトを実行し結果を出力します。スクリプトは、埋込みの zabbix スクリプトエンジンを使用して実行されます。

コンパイルまたは実行エラーの場合、zabbix_js はエラーを stderr に出力し、コード 1 で終了します。

使用方法

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

zabbix_js は、次のコマンドラインパラメータを受け付けます。

```
-s, --script script-file          実行するスクリプトのファイル名を指定します。ファイル名として "-" を指定した場合、スクリ
-i, --input input-file            入力情報のファイル名を指定します。内容は値としてスクリプトに渡されます。-p とは同時に指
-p, --param input-param           入力パラメータを指定します。値としてスクリプトに渡される変数です。-i とは同時に指定でき
-l, --loglevel log-level          ログレベルを指定します。範囲: 0-5。
-t, --timeout timeout             タイムアウトを秒単位で指定します。有効範囲: 1～600 秒（デフォルト: 10 秒）。
-h, --help                        ヘルプ情報を表示します。
-V, --version                     バージョン番号を表示します。
-w <webdriver url>                ブラウザ監視を有効にします。
```

例:

例 1: 値を含む変数を使用してスクリプトを実行する

スクリプト (script-file.js):

```
return value;
```

使用方法:

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

出力: example_value

例 2: 入力情報を含むファイルを使用してスクリプトを実行する

スクリプトのファイル (script-file.js):

```
return value;
```

入力情報を含むファイル (example.txt):

```
Example of input information from the file
```

使用方法:

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

出力: `Example of input information from the file`

例 3: stdin から読み込んだ入力を使用してスクリプトを実行する

スクリプトのファイル (`script-file.js`):

```
return value;
```

使用方法:

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

入力 (stdin):

```
Example of input from stdin
```

出力: `Example of input from stdin`

9 Zabbix Web サービス

概要

Zabbix Web サービスは、外部 Web サービスとの通信に使用されるプロセスです。現在、Zabbix Web サービスは、定期レポートの生成と送信に使用されており、将来的には機能が追加される予定です。

Zabbix サーバーは HTTP(S) 経由で Web サービスに接続します。Zabbix Web サービスはGoogle Chromeが同じホストにインストールされている必要があります。一部のディストリビューションでは Chromium でも機能する場合があります (既知の問題)。

インストール

公式の zabbix-web-service パッケージはZabbix リポジトリから入手できます。

Zabbix Web サービスをソースからコンパイルするには、`--enable-webservice` 設定オプションを指定します。

Zabbix Web サービスを設定するには、`zabbix_web_service.conf`設定ファイルのパラメーターを更新します。

> **Attention:**
> Zabbix サーバーと Zabbix Web サービスの間の暗号化を証明書を使用して設定することを強くお勧めします。デフォルトでは、Zabbix サーバーと Zabbix Web サービスの間で送信されるデータは暗号化されないため、不正アクセスにつながる可能性があります。

4. インストール

サイドバーから、参照したい項目を選択してください。

1 Zabbix の入手

概要

Zabbix を入手するには 4 つの方法があります。

- 配布パッケージからのインストール
- 最新のソースアーカイブをダウンロードして自分でコンパイル
- コンテナでのインストール
- 仮想アプライアンスをダウンロード

最新の配布パッケージ、コンパイル済みのソース、または仮想アプライアンスについては、Zabbix ダウンロードページにアクセスしてください。最新バージョンへのリンクが掲示されています。

Zabbix ソースコードの取得

Zabbix ソースコードを取得する方法はいくつかあります。

- 公式 Zabbix Web サイトから、リリース済みの安定版をダウンロードできます
- 公式 Zabbix Web サイトの開発者ページから、nightly build をダウンロードできます

- Git ソースコードリポジトリシステムから、最新の開発版を取得できます。
  - 完全なリポジトリの主要な場所は次のとおりです。https://git.zabbix.com/scm/zbx/zabbix.git
  - master およびサポート対象のリリースは、Github にもミラーされています。https://github.com/zabbix/zabbix

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

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

Zabbix のすべてのソースコードを取得するには、コードを配置したいディレクトリに移動して、次を実行します。

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

2 要件

ハードウェア要件

メモリ

Zabbix には物理メモリとディスク容量の両方が必要です。必要なディスク容量は、監視対象のホストとパラメーターの数によって大きく異なります。監視対象パラメーターのヒストリを長期間保存することを計画している場合は、データベースにヒストリを保存するのに十分なスペースを確保するため、少なくとも数ギガバイトを検討する必要があります。各 Zabbix デーモンプロセスには、データベースサーバーへの複数の接続が必要です。接続に割り当てられるメモリの量はデータベースエンジンの構成によって異なります。

> **Note:**
> 物理メモリが多ければ多いほど、データベース (したがって Zabbix) の動作が速くなります。

CPU

Zabbix、特に Zabbix データベースは、監視するパラメーターの数や選択したデータベースエンジンに応じて、かなりの CPU リソースを必要とする場合があります。

その他のハードウェア

Zabbix で SMS 通知サポートを使用するには、シリアル通信ポートとシリアル GSM モデムが必要です。USB シリアルコンバータも使用できます。

ハードウェア構成例

この表は **Linux/BSD/Unix** プラットフォームを想定したハードウェア構成の例を示しています。

これらは、開始時のサイズとハードウェア構成の例です。Zabbix のインストールはそれぞれ異なります。Zabbix を本番環境に導入する前に、ステージング環境または開発環境で Zabbix システムのパフォーマンスをベンチマークし、要件を十分に理解してください。

インストールサイズ	監視メトリクス[1]	CPU/vCPU コア	メモリ (GiB)	データベース	Amazon EC2[2]
小	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
中	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
大	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
特大	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] 1 メトリクス = 1 アイテム + 1 トリガー + 1 グラフ
 [2] ARM64 または x86_64 アーキテクチャを使用する Amazon 汎用 EC2 インスタンスの例では、本番環境にインストールする前に、Zabbix のインストール評価およびテスト中にコンピュート/メモリ/ストレージの最適化などの適切なインスタンスタイプを選択する必要があります。

Note:
実際の構成は、アクティブなアイテムの数とリフレッシュレートに大きく依存します (詳細については、このページのデータベースのサイジングセクションを参照してください)。大規模なインストールの場合は、データベースを別のサーバーで実行することを強くお勧めします。

サポートするプラットフォーム

セキュリティ要件と監視サーバーのミッションクリティカルな性質から、UNIX は必要な性能と耐障害性、回復力を一貫して提供できる唯一の OS です。Zabbix は市場をリードするバージョンで動作します。

Zabbix のコンポーネントは以下のプラットフォームで動作確認されています:

プラットフォーム	サーバー	エージェント	エージェント 2	コメント
Linux	x	x	x	
Windows	-	x	x	Zabbix エージェントは、Windows XP (64bit)/Server 2003 以降、すべてのデスクトップおよびサーバーバージョンでサポートされています。 Zabbix エージェント 2 は、Windows 10 (32bit)/Server 2016 以降、すべてのデスクトップおよびサーバーバージョンでサポートされています。重要なセキュリティの脆弱性を防ぐために、サポートする Go バージョンでのみコンパイルされているためです。GO 1.21 から必須 Windows バージョンが上昇し、Zabbix エージェント 2 の最小バージョンが Windows 10/Server 2016 になります。
Mac OS X	x	x	-	
IBM AIX	x	x	-	Zabbix エージェントは、バージョン 6.1 TL07 / 7.1 TL01 以下の AIX プラットフォームでは動作しません。
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
Zabbix サーバー/エージェントは他の Unix 系 OS でも動作する可能性があります。

Attention:
Zabbix は、暗号化してコンパイルされている場合はコアダンプを無効にし、システムがコアダンプの無効化を許可していない場合は起動しません。

ソフトウェア要件

Zabbix は最新の Web サーバー、主要なデータベースエンジン、および PHP スクリプト言語を組み合わせて動作します。

サードパーティ製の外部周辺ソフトウェア

必須と記載されている場合、必要なソフトウェア/ライブラリは厳密に必要です。オプションのものは、特定の機能をサポートするために必要です。

Software	Mandatory status	Supported versions	Comments
MySQL/Percona	いずれか 1 つ	8.0.30-9.6.X	MySQL（または Percona）を Zabbix のバックエンドデータベースとして使用する場合に必要です。InnoDB エンジンが必要です。 MySQL バージョンのサポート追加: - 9.5.X は Zabbix 7.4.6 以降; - 9.6.X は Zabbix 7.4.9 以降。 サーバー/プロキシのビルドには、C API (libmysqlclient) ライブラリの使用を推奨します。
MariaDB		10.5.00-12.2.X	InnoDB エンジンが必要です。 推奨バージョンは 11.4 です。 サーバー/プロキシのビルドには、MariaDB Connector/C ライブラリの使用を推奨します。 MariaDB バージョンのサポート追加: - 12.0.X は Zabbix 7.4.3 以降; - 12.2.X は Zabbix 7.4.9 以降。 あわせて参照: MariaDB で発生する可能性のあるデッドロック および MariaDB 10.5.1-10.5.9 における UI 要素へのアクセス。
PostgreSQL		13.0-18.X	PostgreSQL を Zabbix のバックエンドデータベースとして使用する場合に必要です。 インストール規模によっては、特定の操作でデータベースが使用するメモリ量を十分に確保し、クエリ実行に時間がかかりすぎないようにするため、PostgreSQL の *work_mem* 設定プロパティ（デフォルト値は 4MB）を増やす必要がある場合があります。 PostgreSQL バージョンのサポート追加: - 18.X は Zabbix 7.4.4 以降。
TimescaleDB for PostgreSQL		2.13.0-2.26.X	TimescaleDB を PostgreSQL データベース拡張として使用する場合に必要です。圧縮をサポートする TimescaleDB Community Edition をインストールしてください。 PostgreSQL 15 は TimescaleDB 2.10 以降でサポートされる点に注意してください。 PostgreSQL と TimescaleDB のバージョン互換性の詳細については、TimescaleDB documentation も参照できます。 TimescaleDB バージョンのサポート追加: - 2.20.X、2.21.X は Zabbix 7.4.1 以降; - 2.22.X は Zabbix 7.4.4 以降; - 2.23.X は Zabbix 7.4.6 以降; - 2.24.X は Zabbix 7.4.7 以降; - 2.25.X は Zabbix 7.4.8 以降; - 2.26.X は Zabbix 7.4.9 以降。
SQLite	オプション	3.3.5-3.34.X	SQLite は Zabbix プロキシでのみサポートされます。SQLite を Zabbix プロキシデータベースとして使用する場合に必要です。
Elasticsearch		7.X	Elasticsearch は Zabbix サーバーでのみサポートされ、履歴データの保存専用です。Elasticsearch のサポートは現在実験的です。サーバー/プロキシ に必要なソフトウェアも参照してください。
smartmontools		7.1 以降	Zabbix エージェント 2 に必要です。
who			user count プラグインに必要です。
dpkg			system.sw.packages プラグインに必要です。
pkgtool			system.sw.packages プラグインに必要です。
rpm			system.sw.packages プラグインに必要です。
pacman			system.sw.packages プラグインに必要です。
q applets			q applets の一部である `qlist` と `qsize` は、Gentoo Linux 上の system.sw.packages プラグインに必要です。

Note:
Zabbix はオペレーティングシステムで利用可能なデータベースでも動作しますが、最良の利用体験のため、公式のデータベース開発元リポジトリからインストールしたデータベースの使用を推奨します。

Web インターフェース

Zabbix Web インターフェースでサポートされる最小画面幅は 1200px です。

必須と記載されている場合、必要なソフトウェア/ライブラリは厳密に必須です。オプションのものは、特定の機能をサポートするために必要です。

Software	Mandatory status	Supported versions	Comments
PHP	Yes	8.0.0 - 8.4.X	
Apache *Nginx*	One of	2.4 以降 1.20 以降	
MySQL *PostgreSQL*	One of	サードパーティ製外部周辺ソフトウェアを参照	
PHP extensions			
mysqli	Yes		MySQL が Zabbix バックエンドデータベースとして使用される場合に必要です。
pgsql			PostgreSQL が Zabbix バックエンドデータベースとして使用される場合に必要です。
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*)。ユーザースクリプトのサポートに必要です。
gd		2.0.28 以降	php-gd（ディストリビューターによって個別パッケージとして提供される場合）。PHP GD 拡張は、PNG 画像（*--with-png-dir*）、JPEG 画像（*--with-jpeg-dir*）、および FreeType 2（*--with-freetype-dir*）をサポートしている必要があります。一部の Web インターフェース言語で発生する可能性のあるグラフ内のテキスト重なりを回避するには、バージョン 2.3.0 以降が必要になる場合があります。
libxml		2.6.15 以降	php-xml（ディストリビューターによって個別パッケージとして提供される場合）
xmlwriter			php-xmlwriter（ディストリビューターによって個別パッケージとして提供される場合）
xmlreader			php-xmlreader（ディストリビューターによって個別パッケージとして提供される場合）
ctype			php-ctype (*--enable-ctype*)
session			php-session（ディストリビューターによって個別パッケージとして提供される場合）
ldap	No		php-ldap。LDAP 認証に必要です。
openssl			php-openssl。SAML 認証に必要です。
gettext			php-gettext (*--with-gettext*)。翻訳に必要です。
cURL		7.19.4 以降	php-curl。Duo Universal Prompt MFAおよびSMTP 認証に必要です。

Zabbix に同梱されるサードパーティ製 Web インターフェースライブラリ:

Library	Mandatory status	Supplied version	Comments
jQuery JavaScript Library	Yes	3.6.0	クロスブラウザ開発のプロセスを簡素化する JavaScript ライブラリです。
jQuery UI		1.12.1	jQuery 上に構築された、ユーザーインターフェースのインタラクション、エフェクト、ウィジェット、およびテーマのセットです。

Library	Mandatory status	Supplied version	Comments
SAML PHP Toolkit		4.3.1	Zabbix へのサインインを可能にするために、SAML 2.0 認証サポートを追加する PHP ツールキットです。 同梱バージョン履歴: - Zabbix 7.4.0 以降は 4.0.0; - Zabbix 7.4.8 以降は 4.3.1（推奨）。
Symfony Yaml Component		5.1.0	YAML 形式で Zabbix 設定要素をエクスポートおよびインポートするためのサポートを追加します。

Note:
Zabbix は、Apache、MySQL、PostgreSQL の以前のバージョンでも動作する場合があります。

Attention:
デフォルトの DejaVu 以外のフォントでは、PHP 関数imagerotateが必要になる場合があります。これがない場合、グラフ表示時にこれらのフォントが正しくレンダリングされない可能性があります。この関数は、PHP がバンドル版 GD でコンパイルされている場合にのみ利用可能ですが、Debian やその他のディストリビューションでは通常そうではありません。

Zabbix Web インターフェースコードの作成およびデバッグに使用されるサードパーティ製ライブラリ:

Library	Mandatory status	Minimum version	Description
Composer	No	2.4.1	PHP 用のアプリケーションレベルのパッケージマネージャーで、PHP ソフトウェアおよび必要なライブラリの依存関係を管理するための標準形式を提供します。
PHPUnit		8.5.29	Zabbix Web インターフェースをテストするための PHP ユニットテストフレームワークです。
SASS		3.4.22	Cascading Style Sheets（CSS）に解釈およびコンパイルされるプリプロセッサスクリプト言語です。

クライアント側の Web ブラウザ

Cookie と JavaScript を有効にする必要があります。

Google Chrome、Mozilla Firefox、Microsoft Edge、Apple Safari、Opera の最新安定バージョンに対応しています。

Warning:
IFrame の same-origin ポリシーが実装されているため、Zabbix を別ドメインのフレームに配置することはできません。

 ただし、Zabbix フレームに配置されたページと Zabbix の Web インターフェースが同じドメインにある場合、そのページは JavaScript 経由で Zabbix の Web インターフェースにアクセスできます。`http://secure-zabbix.com/cms/page.html` のようなページを `http://secure-zabbix.com/zabbix/`のダッシュボードに配置すると、JS から Zabbix に完全にアクセスすることができます。

サーバー/プロキシ

必須要件は常に必須です。特定の機能をサポートするには、オプションの要件が必要です。

要件	必須ステータス	説明
libpcre2	Yes	Perl 互換正規表現(PCRE) のサポートには PCRE2 ライブラリが必要です。 PCRE2 v10.x がサポートされています。
libevent		一括メトリクスサポートと IPMI 監視に必要です。バージョン 2.0.10 以降。
libevent-pthreads		プロセス間通信に必要です。
libpthread		ミューテックスおよび読み取り/書き込みロックのサポートに必要です (libc の一部である可能性があります)。
libresolv		DNS 解決に必要です (libc の一部である可能性があります)。
libiconv		テキストのエンコード/フォーマット変換に必要です (libc の一部である可能性があります)。Linux 上の Zabbix サーバーには必須です。
libz		圧縮のサポートに必要です。

要件	必須ステータス	説明
libm		Math ライブラリ。Zabbix サーバーのみ必要です。
libmysqlclient	どれか一つ	MySQL を使用する場合は必要です。
libmariadb		MariaDB を使用する場合は必要です。
libpq5		PostgreSQL を使用する場合は必要です。*libpq5* のバージョンは、PostgreSQL データベースのバージョン以上である必要があります。
libsqlite3		SQLite を使用する場合は必要です。Zabbix プロキシのみ必要です。
libOpenIPMI	No	IPMI のサポートに必要です。Zabbix サーバーのみ必要です。
libssh2 または *libssh*		SSH チェックに必要です。バージョン 1.8.0 以降 (libssh2)、0.9.0 以降 (libssh)。
libcurl		次の機能に必要です: - Web 監視、VMware 監視、HTTP エージェントアイテム (バージョン 7.19.1 以降) - Zabbix エージェント web.page.*アイテム (バージョン 7.19.1 以降。参照: エージェント要件) - SMTP 認証 (Basic: バージョン 7.20.0 以降。OAuth: バージョン 7.33 以降。参照: Web インターフェース要件) - Elasticsearch (バージョン 7.28.0 以降) バージョン 7.28.0 以降を推奨します。 `web.page.*` アイテムのアップグレードされた cURL 機能を利用するには、Zabbix サーバー/プロキシを再起動します。 SMTP 認証の場合、実行時に `libcurl-full` パッケージを使用します。
libxml2		VMware 監視と XML XPath の保存前処理に必要です。
net-snmp		SNMP のサポートに必要です。バージョン 5.3.0 以降。 強力な暗号化プロトコル (AES192/AES192C、AES256/AES256C) のサポートは、net-snmp ライブラリ 5.8 以降で利用可能です。RHEL 8+ ベースのシステムでは、net-snmp 5.8.15 以降を使用することをお勧めします。
libunixodbc		データベース監視に必要です。
libgnutls または *libopenssl*		暗号化を使用する場合に必要です。 最小バージョン: *libgnutls* - 3.1.18、*libopenssl* - 1.0.1
libldap		LDAP のサポートに必要です。
fping		ICMP ping アイテムに必要です。
c-ares		Zabbix が`--with-ares` オプションで構成されている場合、非同期 DNS 解決に必要です。それ以外の場合、*libevent* が使用されます。 最小バージョン: 1.16.0

エージェント

要件	必須ステータス	説明
libpcre2	Yes	Perl 互換正規表現(PCRE) のサポートには、PCRE/PCRE2 ライブラリが必要です。 PCRE2 v10.x がサポートされています。 ログの監視に必要です。Windows でも必要です。
libpthread		ミューテックスおよび読み書きロックのサポートに必要です (libc の一部である可能性があります)。Windows では不要です。
libresolv		DNS 解決に必要です (libc の一部である可能性があります)。Windows では不要です。
libiconv		ログアイテム、ファイルコンテンツ、ファイル regex および regmatch アイテム (libc の一部である可能性があります) での UTF-8 へのテキストエンコーディング/フォーマット変換に必要です。Windows では不要です。
libgnutls または *libopenssl*	No	暗号化を使用する場合に必要です。 最小バージョン: *libgnutls* - 3.1.18、*libopenssl* - 1.0.1 Microsoft Windows では OpenSSL 1.1.1 以降が必要です。
libldap		LDAP を使用するのに必要です。Windows ではサポートされていません。
libcurl		‘Zabbix エージェント web.page.*アイテムの拡張サポートに必要です。 libcurl がない場合でも、基本的な機能は利用可能です (例: web.page.get[http://example.com])。libcurl を使用すると、エージェントは認証情報付きの HTTP URL(例: http://user:password@example.com) や HTTPS URL などの追加機能をサポートします。 バージョン 7.19.1 以降が必要です (バージョン 7.28.0 以降を推奨します)。 アップグレードされた cURL 機能を使用するには、Zabbix エージェントを再起動してください。

要件	必須ステータス	説明
libmodbus		Modbus モニタリングを使用する場合にのみ必要です。 バージョン 3.0 以降。

エージェント 2

要件	必須ステータス	説明
Go	はい	ソースから Zabbix エージェント 2 およびそのプラグインをビルドするために必要です。 Go 1.24.10 以降がサポートされています。インストール手順については go.dev を参照してください。 Zabbix エージェント 2 およびそのプラグインで使用される Go ライブラリは、Zabbix Git リポジトリに一覧表示されています（リポジトリ内で `indirect` としてマークされているライブラリは、他の必須ライブラリの依存関係です）。 - Zabbix agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	はい	Perl Compatible Regular Expression (PCRE) をサポートするために、PCRE2 ライブラリが必要です。 PCRE2 v10.x がサポートされています。 ログ監視に必要です。また、Windows でも必要です。
libopenssl	いいえ	暗号化を使用する場合に必要です。 UNIX プラットフォームでは OpenSSL 1.0.1 以降が必要です。 OpenSSL ライブラリでは PSK サポートが有効になっている必要があります。 LibreSSL はサポートされていません。 Microsoft Windows システムでは OpenSSL 1.1.1 以降が必要です。

Web サービス

Zabbix Web サービスを使用して定期レポートを生成するのに、Google Chrome の最新の安定バージョンがサポートされています。

Web サービスの構築に必要な Go のバージョンは、Zabbix エージェント 2で使用されているバージョンと同じです。

Java ゲートウェイ

ソースリポジトリまたはアーカイブから Zabbix を入手した場合、必要な依存関係はすでにソースツリーに含まれています。

ディストリビューションのパッケージから Zabbix を入手した場合、必要な依存関係はパッケージングシステムによってすでに提供されています。

上記のいずれの場合も、ソフトウェアはそのまま使用可能であり、追加のダウンロードは必要ありません。

ただし、これらの依存関係について独自のバージョンを使用したい場合（たとえば、何らかの Linux ディストリビューション向けのパッケージを準備している場合）、以下に Java ゲートウェイでの動作が確認されているライブラリバージョンの一覧を示します。Zabbix は、これらのライブラリの他のバージョンでも動作する可能性があります。

次の表は、現在 Java ゲートウェイのオリジナルコードに同梱されている JAR ファイルを示しています。

Library	Mandatory status	Bundled version	Comments
android-json	はい	4.3r1	JSON（JavaScript Object Notation）は軽量なデータ交換形式です。これは、Android SDK から抽出された org.json 互換の Android 実装です。
logback-classic		1.5.16	
logback-core		1.5.16	
slf4j-api		2.0.16	

Java ゲートウェイは、Oracle Java またはオープンソースの OpenJDK（バージョン 1.6 以降）のいずれかを使用してビルドできます。Zabbix が提供するパッケージは OpenJDK を使用してコンパイルされています。次の表は、ディストリビューションごとに Zabbix パッケージのビルドに使用される OpenJDK パッケージを示しています。

Distribution	OpenJDK package
AlmaLinux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 9	`java-11-openjdk-devel` (amd64, arm64: 11.0.18.0.10-3)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4.0.1; arm64: 11.0.20.0.8-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 24.04	`default-jdk-headless` (amd64, arm64: 2:1.21-75+exp1)
Ubuntu 22.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72build2)
Ubuntu 20.04	`default-jdk-headless` (amd64, arm64: 2:1.11-72)

デフォルトのポート番号

次のリストは各コンポーネントのデフォルトポート番号です。

Zabbix コンポーネント	ポート番号	プロトコル	接続の種類
Zabbix エージェント	10050	TCP	オンデマンド
Zabbix エージェント 2	10050	TCP	オンデマンド
Zabbix サーバー	10051	TCP	オンデマンド
Zabbix プロキシ	10051	TCP	オンデマンド
Zabbix Java ゲートウェイ	10052	TCP	オンデマンド
Zabbix Web サービス	10053	TCP	オンデマンド
Zabbix Web インターフェース	80	HTTP	オンデマンド
	443	HTTPS	オンデマンド
Zabbix トラッパー	10051	TCP	オンデマンド

Note:
Zabbix 通信を有効にするには、ファイアウォールでポートを開く必要があります。通常、発信の TCP 接続には明示的なファイアウォール設定は必要ありません。

データベースサイズ

Zabbix の設定データが必要とするディスク容量は一定であり、ほとんど増加しません。

Zabbix データベースのサイズは主に、保存される履歴データ量を定義する以下の変数に依存します。

- 1 秒あたりに処理される値の数

これは、Zabbix サーバーが 1 秒ごとに受信する新しい値の平均数です。
たとえば、監視対象のアイテムが 3000 個あり、更新間隔が 60 秒の場合、1 秒あたりの値の数は 3000/60 = **50** と計算されます。

これは、1 秒ごとに 50 個の新しい値が Zabbix データベースに追加されることを意味します。

- 履歴に対する housekeeper 設定

Zabbix は、通常は数週間または数か月の一定期間、値を保持します。
新しい値ごとに、データとインデックスのために一定量のディスク容量が必要です。

したがって、30 日分の履歴を保持し、1 秒あたり 50 個の値を受信する場合、値の総数はおよそ (**30***24*3600)* **50** = 129.600.000、つまり約 1 億 3000 万件になります。

使用するデータベースエンジンや受信する値の種類（浮動小数点、整数、文字列、ログファイルなど）によって、1 つの値を保持するために必要なディスク容量は 40 バイトから数百バイトまで変動します。
通常、数値アイテム[2] では 1 値あたり約 90 バイトです。
この場合、1 億 3000 万件の値には 130M * 90 bytes = **10.9GB** のディスク容量が必要になります。

Note:
text/log アイテムの値のサイズを正確に予測することはできませんが、1 値あたり約 500 バイトを見込めます。

- トレンドに対する housekeeper 設定

Zabbix は、各アイテムについて 1 時間ごとの最大値/最小値/平均値/件数のセットをテーブル **trends** に保持します。
このデータはトレンド表示や長期間のグラフに使用されます。
この 1 時間という期間は変更できません。

Zabbix データベースでは、データベースの種類に応じて、各集計値に約 90 バイトが必要です。
トレンドデータを 5 年間保持したいとします。
3000 個のアイテムの値には、1 年あたり 3000*24*365* **90** = **2.2GB**、5 年間では **11GB** が必要になります。

- イベントに対する housekeeper 設定

各 Zabbix イベントには、およそ 250 バイトのディスク容量が必要です[1]。
Zabbix が 1 日に生成するイベント数を見積もるのは困難です。
最悪のケースでは、Zabbix が 1 秒あたり 1 イベントを生成すると仮定できます。

復旧した各イベントについては、event_recovery レコードが作成されます。
通常、ほとんどのイベントは復旧されるため、1 イベントにつき 1 つの event_recovery レコードがあると考えられます。
つまり、イベントごとに追加で 80 バイトが必要です。

イベントには任意でタグを付けることができ、各タグレコードにはおよそ 100 バイトのディスク容量が必要です[1]。
イベントあたりのタグ数（#tags）は設定に依存します。
したがって、各イベントには追加で #tags * 100 bytes のディスク容量が必要になります。

つまり、3 年分のイベントを保持したい場合、必要なディスク容量は 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B となります[2]。

Note:
[1] 非 ASCII のイベント名、タグ、値がある場合はさらに増えます。
 [2] このサイズの概算は MySQL に基づいており、他のデータベースでは異なる場合があります。

この表には、Zabbix システムに必要なディスク容量を計算するための式が示されています。

Parameter	必要なディスク容量の計算式（バイト単位）
Zabbix configuration	固定サイズ。通常は 10MB 以下。
History	days*(items/refresh rate)*24*3600*bytes items：アイテム数 days：履歴を保持する日数 refresh rate：アイテムの平均更新間隔 bytes：1 つの値を保持するために必要なバイト数。データベースエンジンに依存し、通常は ~90 bytes。
Trends	days*(items/3600)*24*3600*bytes items：アイテム数 days：履歴を保持する日数 bytes：1 つのトレンドを保持するために必要なバイト数。データベースエンジンに依存し、通常は ~90 bytes。

Parameter	必要なディスク容量の計算式（バイト単位）
Events	days*events*24*3600*bytes events：1 秒あたりのイベント数。最悪のケースでは 1 秒あたり 1 イベント。 days：履歴を保持する日数 bytes：1 つのイベントを保持するために必要なバイト数。データベースエンジンに依存し、通常は ~330 + イベントあたりの平均タグ数 * 100 bytes。

したがって、必要な総ディスク容量は次のように計算できます。

Configuration + History + Trends + Events

ディスク容量は、Zabbix のインストール直後にすぐ使用されるわけではありません。
データベースサイズは増加していき、その後、housekeeper 設定に応じた時点で増加が止まります。

時刻同期

Zabbix が動作しているサーバで正確なシステム時刻を得ることは非常に重要です。ntpdはホストの時刻を他のマシンの時刻と同期させる最も有名なデーモンです。Zabbix のコンポーネントが動作している全てのシステムで時刻を同期させることを強く推奨します。

ネットワーク要件

次のリストは、コンポーネントごとのオープンポートのデフォルト設定です。

コンポーネント	ポート
Web インターフェース	80 (http), 443 (https)
Zabbix サーバー	10051 (アクティブなプロキシ/エージェントで使用)
アクティブプロキシ	10051
パッシブプロキシ	10051
エージェント 2	10050
トラッパー	
Java ゲートウェイ	10052
Web サービス	10053

Note:
Zabbix との外部通信を可能にするには、ファイアウォールでポート番号を開く必要があります。通常、発信 TCP 接続には明示的なファイアウォール設定は必要ありません。

3 ソースコードからのインストール

Zabbix の最新バージョンはソースからコンパイルすることで入手できます。

Zabbix をソースからインストールするためのステップバイステップのチュートリアルはこちらです。

Zabbix デーモンのインストール

1 ソースアーカイブのダウンロード

Zabbix ダウンロードページにアクセスし、ソースアーカイブをダウンロードします。ダウンロード後、次を実行してソースを展開します。

```
tar -zxvf zabbix-7.4.0.tar.gz
```

Note:
コマンドには正しい Zabbix のバージョンを入力してください。ダウンロードしたアーカイブの名前と一致している必要があります。

2 ユーザーアカウントの作成

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

デフォルト設定では、デーモンが'root' として起動された場合、'zabbix' ユーザーアカウントに切り替わります。したがって、このアカウントが存在する必要があります。`zabbix` システムユーザーとグループを作成するには、以下のコマンドを実行します。

RedHat ベースのシステム:

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

Debian ベースのシステム:

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

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

セキュリティに関する推奨事項

Zabbixサーバーとエージェントを同じマシンで実行する場合は、別々のユーザーアカウントで実行することをお勧めします。両方を同じユーザーで実行すると、エージェントがサーバーの設定ファイルにアクセスできるようになるため、データベースのパスワードといった機密情報が Zabbix の管理者レベルのユーザーに公開される可能性があります。

> **Attention:**
> Zabbix を root、bin、または特別な権限を持つその他のアカウントで実行すると、セキュリティリスクが生じます。

ホームディレクトリ (オプション)

Zabbix プロセスにはホームディレクトリは必要ないため、通常は作成しないことをお勧めします。ただし、ホームディレクトリを必要とする機能 (例: MySQL 認証情報を `$HOME/.my.cnf` に保存する) が必要な場合は、次のコマンドを実行してホームディレクトリを作成できます。

RedHat ベースのシステムの場合:

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

Debian ベースのシステムの場合:

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

3 Zabbix データベースの作成

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

データベーススキーマを作成し、データセットを挿入するための SQLスクリプトが提供されています。Zabbix プロキシデータベースにはスキーマのみが必要ですが、Zabbix サーバーデータベースにはスキーマ上のデータセットも必要です。

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

4 ソースの設定

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

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

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

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

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

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

Zabbix サーバーおよびエージェント用にソースを設定するには、次のように実行できます。

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

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

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

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

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

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

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

または、Zabbix agent 2 の場合は次を実行します。

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

Note:
Zabbix agent 2 をビルドするには、サポートされている Go のバージョン がインストールされている必要があります。

コンパイルオプションに関する注意:

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

Attention:
ライブラリの不足やその他の問題により `./configure` が失敗した場合は、詳細なエラー情報について `config.log` ファイルを確認してください。
たとえば、`libssl` が不足している場合、直ちに表示されるエラーメッセージは誤解を招く可能性があります。

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

この場合、`config.log` には実際の原因が示されます。

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

あわせて参照:

- 暗号化サポート付きで Zabbix をコンパイルする
- 既知のコンパイルの問題

5 全体の作成とインストール

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

```
$ make dbschema
```

```
make install
```

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

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

Note:
/usr/local とは異なる場所を指定するには、ソースを構成する前の手順で--prefix キーを使用します (例: --prefix=/home/zabbix)。この場合、デーモンのバイナリは <prefix>/sbin にインストールされ、ユーティリティは <prefix>/bin にインストールされます。man ページは <prefix>/share の下にインストールされます。

6 設定ファイルの確認と編集

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

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

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

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

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

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

- Zabbix プロキシをインストールしている場合は、プロキシ設定ファイル**/usr/local/etc/zabbix_proxy.conf** を編集します

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

Note:
SQLite では、データベースファイルへのフルパスを指定する必要があります。DB ユーザーとパスワードは必要ありません。

7 デーモンの起動

サーバー側で zabbix_server を起動します。

```
zabbix_server
```

Note:
システムで 36MB(またはそれ以上) の共有メモリの割り当てが許可されていることを確認してください。許可されていないと、サーバーが起動せず、サーバーログファイル内に”Cannot allocate shared memory for <type of cache>.” というメッセージが表示されます。これは FreeBSD、Solaris 8 で発生する可能性があります。

すべての監視対象マシンで zabbix_agentd を実行します。

```
zabbix_agentd
```

Note:
システムで 2MB の共有メモリの割り当てが許可されていることを確認してください。許可されていないと、エージェントが起動せず、エージェントログファイル内に”Cannot allocate shared memory for collector.” というメッセージが表示されることがあります。これは Solaris 8 で発生する可能性があります。

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

```
zabbix_proxy
```

Zabbix Web インターフェースのインストール

PHP ファイルのコピー

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

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

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

HTML ルートの代わりにサブディレクトリを使用することをお勧めします。サブディレクトリを作成し、そこに Zabbix フロントエンドファイルをコピーするには、次のコマンドの <htdocs> を実際のディレクトリを置き換えて実行します。

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

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

Web インターフェースのインストール

Zabbix Web インターフェースのインストールウィザードの詳細については、Web インターフェースのインストールページを参照してください。

Zabbix エージェント 2 のロード可能プラグインのインストール

Zabbix エージェント 2 のロード可能プラグインのインストールは、組み込みプラグインで対応していない監視対象（例: MongoDB サーバーまたはクラスター、PostgreSQL およびその派生版など）を監視したい場合にのみ必要です。
ロード可能プラグイン および組み込みプラグイン の完全な一覧を参照してください。

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

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

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

```
make
```

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

プラグインの実行ファイルは、Zabbix エージェント 2 からロード可能である限り、任意の場所に配置できます。
プラグイン設定ファイルで、プラグインバイナリへのパスを指定してください。たとえば、PostgreSQL プラグインの postgresql.conf では次のように指定します。

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

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

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

プラグインの設定の詳細については、セットアップ に進んでください。

Zabbix が提供するロード可能プラグインでは、以下のビルドターゲットを持つシンプルな makefile が使用されています。

- `make` - プラグインをビルドします
- `make clean` - プラグインのビルドによって作成されたすべてのファイルを削除します
- `make check` - セルフテストを実行します（実際の監視対象が必要です。例: PostgreSQL データベース）
- `make style` - `golangci-lint` を使用して Go コードのスタイルをチェックします
- `make format` - `go fmt` を使用して Go コードを整形します
- `make dist` - すべての依存関係を含むソースアーカイブを作成します

Java ゲートウェイのインストール

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

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

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

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

Java ゲートウェイをコンパイルし、JAR ファイルとしてパッケージ化するには、`make` を実行します。この手順では、パス上に `javac` および `jar` 実行ファイルが必要であることに注意してください。

```
make
```

これで、src/zabbix_java/bin に zabbix-java-gateway-$VERSION.jar ファイルが作成されます。配布ディレクトリ内の src/zabbix_java から Java ゲートウェイを実行することに問題がなければ、Java ゲートウェイの設定および実行手順に進むことができます。そうでない場合は、十分な権限があることを確認して `make install` を実行してください。

```
make install
```

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

Zabbix web サービスのインストール

Zabbix web サービスのインストールは、定期レポートを使用する場合にのみ必要です。

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

Zabbix web サービスをコンパイルするには、`./configure` スクリプトを `--enable-webservice` オプション付きで実行します。

> **Note:**
> Zabbix web サービスをビルドするには、サポートされている Go のバージョンがインストールされている必要があります。

web サービスがインストールされているマシンで zabbix_web_service を実行します。

```
zabbix_web_service
```

定期レポート生成の設定の詳細については、セットアップに進んでください。

1 Windows での Zabbix エージェントのビルド

概要

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

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

Zabbix エージェントのビルドには以下が必要です:

- C コンパイラ (Visual Studio 2022 のビルドツールに含まれています)
- OpenSSL (Zabbix の暗号化機能用)
- PCRE2 (Perl 互換正規表現。Zabbix の正規表現パターンマッチング機能用)

Zabbix エージェントは、以下のいずれかの方法でビルドできます:

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

監視のニーズによっては、追加のライブラリが必要になる場合があります。詳細については、要件を参照してください。

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

> - コマンドを実行するには、x64 Native Tools コマンドプロンプト (Build Tools for Visual Studio 2022 に含まれています) を使用してください。コマンドプロンプトは、保護されたフォルダーへの書き込み権限を持つユーザーによって起動されます。
> - すべてのソースファイルとビルドフォルダー用の作業ディレクトリを `C:\Zabbix` に作成することをお勧めします。ただし、コンパイル済みのコンポーネントは `C:\Program Files\Zabbix\x64` にインストールする必要があります。

vcpkg を使用した Zabbix エージェントのビルド

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

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

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

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

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

3. Zabbix ソースアーカイブ をダウンロードし、`C:\Zabbix\zabbix-7.4.0` に展開します。

4. Zabbix のビルドディレクトリ (C:\Zabbix\zabbix-7.4.0\build\win32\project) に移動し、以下の build.bat スクリプトを作成します。
OpenSSL と PCRE2 がインストールされているディレクトリを正しく指定してください。

```
:: Set vcpkg installation path:
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: Run the build process:
nmake -f Makefile CPU=AMD64 ^
    PCRE2INCDIR="%vcpkg%\include" ^
    PCRE2LIBDIR="%vcpkg%\lib" ^
    TLS=openssl ^
    TLSINCDIR="%vcpkg%\include" ^
    TLSLIBDIR="%vcpkg%\lib" ^
    LIBS="$(LIBS) Crypt32.lib" ^
    all
```

5. スクリプトを実行して Zabbix エージェントをコンパイルします。

```
build.bat
```

コンパイル後、Zabbix コンポーネントのバイナリは C:\Zabbix\zabbix-7.4.0\bin\win64 に配置されます。
Zabbix エージェントの設定ファイルは C:\Zabbix\zabbix-7.4.0\conf にあります。

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

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-7.4.0\bin\win64\zabbix_agent.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-7.4.0\conf\zabbix_agent.win.conf C:\Zabbix\agent\

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

Zabbix エージェントを手動でビルドする

> **Attention:**
> この Zabbix エージェントのビルド方法は、ビルド環境を完全に制御する必要があるユーザー、またはusing vcpkgが利用できない制限された環境にいるユーザーに適しています。

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

ビルドツールのインストール

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

- C コンパイラ（Microsoft Visual C++）
- NMake コマンドラインツール
- x64 Native Tools Command Prompt

OpenSSL のインストール

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

1. Strawberry Perl をダウンロードしてインストールします（MSI インストーラーとして提供されています）。インストール中に、インストール先フォルダとして C:\Zabbix\Strawberry を指定してください。

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

```
cpanm Text::Template
```

3. Strawberry Perl のインストール時に、Netwide Assembler（NASM。OpenSSL のコンパイルに必要）がコンパイルされていることを確認します:

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

NASM がコンパイルされていない場合は、手動でインストールしてください。詳細については、NASM documentation を参照してください。

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

5. 展開したディレクトリに移動し、OpenSSL を設定します。例:

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

Attention:
Windows で Zabbix エージェントをコンパイルする際に、OpenSSL 用のカスタムディレクトリ（例: C:\zabbix または C:\openssl-64bit）を選択する場合は、このディレクトリに対する非管理者ユーザーの書き込みアクセス権を必ず取り消してください。そうしないと、エージェントが権限のないユーザーによって変更可能なパスから SSL 設定を読み込むことになり、潜在的なセキュリティ脆弱性につながります。

- no-shared オプションを指定すると、libcrypto.lib と libssl.lib の OpenSSL 静的ライブラリが自己完結型になり、Zabbix バイナリに外部 DLL を必要とせず OpenSSL が含まれます。これは、OpenSSL ライブラリがなくても Zabbix バイナリを他の Windows マシンにコピーできることを意味します。ただし、新しい OpenSSL のバグ修正版がリリースされた場合は、Zabbix エージェントを再コンパイルする必要があります。
- no-shared オプションを指定しない場合、Zabbix は実行時に OpenSSL DLL に依存します。これは、OpenSSL の更新時に Zabbix エージェントの再コンパイルが不要になる可能性があることを意味します。ただし、他のマシンにコピーする場合は、OpenSSL DLL も一緒にコピーする必要があります。

その他の OpenSSL 設定オプションの詳細については、OpenSSL documentation を参照してください。

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

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

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

7. OpenSSL をインストールします:

```
nmake install
```

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

PCRE のコンパイル

1. PCRE または PCRE2 ライブラリをダウンロードします (https://pcre.org/)。
2. ディレクトリ *E:\pcre2-10.39* に解凍します。
3. CMake をhttps://cmake.org/download/からインストールします。インストール中に次を選択し、cmake\bin がパス上にあることを確認してください (テスト済みバージョン 3.9.4)。
4. 新しい空のビルドディレクトリ、できればソースディレクトリのサブディレクトリを作成します。例: *E:\pcre2-10.39\build*。
5. コマンドラインウィンドウを開きます。例えば x64 Native Tools Command Prompt for VS 2017 を使用し、そのシェル環境から cmake-gui を実行します。エラーが発生する可能性があるため、Windows のスタートメニューから Cmake を起動しないでください。
6. ソースディレクトリとビルドディレクトリにそれぞれ *E:\pcre2-10.39* と *E:\pcre2-10.39\build* を入力します。
7. ”Configure” ボタンをクリックします。
8. このプロジェクトのジェネレーターを指定する場合は、”NMake Makefiles” を選択します。
9. 新しい空のインストールディレクトリを作成します。例: *E:\pcre2-10.39-install*。
10. GUI にはいくつかの設定オプションがリストされます。次のオプションが選択されていることを確認してください。
    - **PCRE_SUPPORT_UNICODE_PROPERTIES** ON

- **PCRE_SUPPORT_UTF** ON
- **CMAKE_INSTALL_PREFIX** *E:\pcre2-10.39-install*

11. もう一度”Configure” をクリックします。隣接する”Generate” ボタンがアクティブになるはずです。
12. ”Generate” をクリックします。
13. エラーが発生した場合は、CMake ビルドプロセスを繰り返す前に CMake キャッシュを削除することをお勧めします。CMake GUI では、”File > Delete Cache” を選択してキャッシュを削除できます。
14. ビルドディレクトリには、使用可能なビルドシステム *Makefile* が含まれているはずです。
15. コマンドラインウィンドウを開きます。例えば x64 Native Tools Command Prompt for VS 2017 を使用し、上記の *Makefile* に移動します。
16. NMake コマンドを実行する。

```
E:\pcre2-10.39\build> nmake install
```

Zabbix エージェントのコンパイル

1. Zabbix ソースアーカイブをダウンロードし、C:\Zabbix\zabbix-7.4.0 に展開します。

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

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

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

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

```
cd C:\Zabbix\zabbix-7.4.0\build\win32\project

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

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

コンパイル後、Zabbix コンポーネントのバイナリは C:\Zabbix\zabbix-7.4.0\bin\win64 に配置されます。
Zabbix エージェントの設定ファイルは C:\Zabbix\zabbix-7.4.0\conf にあります。

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

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-7.4.0\bin\win64\zabbix_agentd.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-7.4.0\conf\zabbix_agentd.win.conf C:\Zabbix\agent\

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

2 Windows での Zabbix エージェント 2 のビルド

概要

このセクションでは、Zabbix エージェント 2 (Windows) をソースコードからビルドする方法を説明します。

vcpkg を使用した Zabbix エージェント 2 のビルド

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

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

2. Go をダウンロードしてインストールします（MSI インストーラーを利用できます）。
インストール時には、インストール先フォルダーとして C:\Zabbix\Go を指定してください。

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

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

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

4. vcpkg を初期化し、Zabbix エージェント 2 のビルドに必要な依存関係をインストールします（これにはしばらく時間がかかる場合があります）。

```
cd C:\Zabbix

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

#### For 64-bit builds:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### For 32-bit builds:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
vcpkg install --triplet x86-mingw-static --x-install-root=x86
```

5. Zabbix ソースアーカイブ をダウンロードし、C:\Zabbix\zabbix-7.4.0 に展開します。

6. Zabbix のビルドディレクトリ (C:\Zabbix\zabbix-7.4.0\build\mingw) に移動し、以下の build.bat スクリプトを作成します。

- 64 ビットビルド用:

```
:: Add MinGW and Go to the system `PATH` variable for the current session:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Set vcpkg installation path:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Set linker flags for Crypt32 library:
SET CGO_LDFLAGS="-lCrypt32"

:: Run the build process:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- 32 ビットビルド用:

```
:: Add MinGW and Go to the system `PATH` variable for the current session:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Set vcpkg installation path:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Set linker flags for Crypt32 library:
SET CGO_LDFLAGS="-lCrypt32"

:: Run the build process:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
```

```
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. スクリプトを実行して Zabbix エージェント 2 をコンパイルします。

```
build.bat
```

コンパイル後、Zabbix エージェント 2 のバイナリは C:\Zabbix\zabbix-7.4.0\bin\win64（64 ビットビルドの場合）または C:\Zabbix\zabbix-7.4.0\bin\win32（32 ビットビルドの場合）に配置されます。
Zabbix エージェント 2 の設定ファイルは C:\Zabbix\zabbix-7.4.0\src\go\conf にあります。

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

```
mkdir C:\Zabbix\agent2

#### For 64-bit builds:
copy C:\Zabbix\zabbix-7.4.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### For 32-bit builds:
copy C:\Zabbix\zabbix-7.4.0\bin\win32\zabbix_agent2.exe C:\Zabbix\agent2\

copy C:\Zabbix\zabbix-7.4.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-7.4.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

必要に応じて、Zabbix エージェント 2 のロード可能プラグイン のコンパイルに進んでください。

Zabbix エージェント 2 のロード可能なプラグインのコンパイル

1. お使いの Zabbix エージェント 2 のバージョンに一致する Zabbix プラグインのソース (例: zabbix-agent2-plugin-postgresql-7.4.0.) をダウンロードし、C:\Zabbix に展開します。

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

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

```
cd C:\Zabbix\zabbix-agent2-plugin-ember-plus-7.4.0

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

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

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

プラグインの実行ファイルは、Zabbix エージェント 2 からロード可能である限り、任意の場所に配置できます。プラグイン設定ファイルで、プラグインバイナリへのパスを指定してください。たとえば、PostgreSQL プラグインの postgresql.conf では次のように指定します。

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

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

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

プラグインの設定の詳細については、セットアップ に進んでください。

Zabbix エージェント 2 を手動でビルドする

> **Attention:**
> この Zabbix エージェント 2 のビルド方法は、ビルド環境を完全に制御する必要があるユーザー、またはvcpkg の使用が不可能な制限された環境にいるユーザーに適しています。

このセクションでは、必要なビルドツールと依存関係のインストール、およびその後のエージェントのコンパイルを含む、Zabbix エージェント 2 を手動でビルドする手順について説明します。

MinGW コンパイラのインストール

1. SJLJ (set jump/long jump) 例外処理と Windows スレッドを備えた MinGW-w64 をダウンロードします (例: *x86_64-8.1.0-release-win32-sjlj-rt_v6-rev0.7z*)
2. 解凍して、*c:\mingw* へ移動します。
3. 環境変数の設定をします。

```
@echo off
set PATH=%PATH%;c:\mingw\bin
cmd
```

コンパイル時に、MinGW が提供する MSYS ターミナルの代わりに Windows プロンプトを使用します。

OpenSSL のインストール

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

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

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

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

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

- 64 ビットビルドの場合:

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

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

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

make
make install
```

- 32 ビットビルドの場合:

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

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

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

make
make install
```

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

- no-shared オプションを指定すると、libcrypto.lib および libssl.lib の OpenSSL 静的ライブラリが自己完結型となるため、Zabbix バイナリには外部 DLL を必要とせずに OpenSSL が含まれます。これは、OpenSSL ライブラリがなくても Zabbix バイナリを他の Windows マシンにコピーできることを意味します。ただし、新しい OpenSSL のバグ修正版がリリースされた場合は、Zabbix エージェントを再コンパイルする必要があります。
- no-shared オプションを指定しない場合、Zabbix は実行時に OpenSSL DLL に依存します。これは、OpenSSL の更新時に Zabbix エージェントの再コンパイルが不要になる可能性があることを意味します。ただし、他のマシンにコピーする際には、OpenSSL DLL もあわせてコピーする必要があります。

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

4. スクリプトを実行して OpenSSL を設定およびインストールします（これにはしばらく時間がかかる場合があります）。

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

PCRE2 のインストール

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

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

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

3. PCRE2 を設定します。

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

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

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

4. PCRE2 をインストールします。

```
make install
```

Zabbix エージェント 2 のコンパイル

1. Zabbix ソースアーカイブをダウンロードし、C:\Zabbix\zabbix-7.4.0 に展開します。

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

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

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

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

- 64 ビットビルドの場合:

```
cd C:\Zabbix\zabbix-7.4.0\build\mingw
set PATH=C:\Zabbix\mingw64\bin;%PATH%
mklink /D C:\Zabbix\x64\OpenSSL-Win64-350-static\lib C:\Zabbix\x64\OpenSSL-Win64-350-static\lib64

#### With TLS support:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### Without TLS support:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- 32 ビットビルドの場合:

```
cd C:\Zabbix\zabbix-7.4.0\build\mingw
set PATH=C:\Zabbix\mingw32\bin;%PATH%
```

```
#### With TLS support:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win64-350-static"

#### Without TLS support:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

コンパイル後、Zabbix エージェント 2 のバイナリは C:\Zabbix\zabbix-7.4.0\bin\win64 に配置されます（32 ビットビルドの場合は C:\Zabbix\zabbix-7.4.0\bin\win32）。
Zabbix エージェント 2 の設定ファイルは C:\Zabbix\zabbix-7.4.0\src\go\conf にあります。

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

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-7.4.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-7.4.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-7.4.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

必要に応じて、Zabbix エージェント 2 のロード可能プラグインのコンパイルに進んでください。

3 macOS での **Zabbix** エージェントのビルド

概要

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

前提条件

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

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

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

次に Automake と pkg-config をインストールします。

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

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

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

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

共有ライブラリを使用したエージェントバイナリのビルド

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

```
brew install pcre2
```

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

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

Zabbix ソースをダウンロード。

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

TLS なしでエージェントをビルド。

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

```
make
make install
```

OpenSSL を使用してエージェントをビルド。

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

GnuTLS を使用してエージェントをビルド。

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

TLS を使用しない静的ライブラリを使用したエージェントバイナリのビルド

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

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

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

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

Zabbix ソースとビルドエージェントをダウンロードします。

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

OpenSSL と静的ライブラリを使用してエージェントバイナリをビルドする

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

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

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

static-libs-source で静的ライブラリをビルドします。

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

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

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

OpenSSL をダウンロードしてビルドする。

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

Zabbix ソースとビルドエージェントをダウンロードする。

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

GnuTLS と静的ライブラリを使用してエージェントバイナリをビルドする

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

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

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

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

static-libs-source で静的ライブラリをビルドします。

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

Nettle をダウンロードしてビルドします。

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

GnuTLS をダウンロードしてビルドします。

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

Zabbix ソースとビルドエージェントをダウンロードをします。

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

```
make
make install
```

4 パッケージインストール

概要

公式の Zabbix 7.4 パッケージは Zabbix の Web サイト で提供されており、そこでお使いのオペレーティングシステムと Zabbix コンポーネントを選択すると、環境に適したインストール手順を生成できます。重要な追加情報については、このページのパッケージインストールに関する注意事項およびInstallation and setup セクションも参照してください。

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

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

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

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

必要に応じて、Zabbix エージェント/agent 2、Zabbix get、Zabbix sender 用の個別パッケージを Zabbix Official Repository で利用できます。

Zabbix は、Linux 以外のオペレーティングシステム向けに、事前コンパイル済みの Zabbix エージェントバイナリも提供しています。以下を参照してください。

- Windows agent installation from MSI
- macOS agent installation from PKG
- Legacy binaries（HP-UX、NetBSD、Tru64、旧バージョンの SLES など、古い/あまり一般的でないシステム向け）

パッケージインストールに関する注意事項

以下の注意事項はすべてのシステムに適用されます。

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

以下の注意事項は RHEL およびその派生ディストリビューションに適用されます。

- EL9 向けの EPEL リポジトリを有効にしている場合、このリポジトリでも Zabbix パッケージが提供されるため、公式の Zabbix パッケージをインストールする前に、パッケージ解決の対象から除外する必要があります。詳細はAccidental installation of EPEL Zabbix packages を参照してください。
- Red Hat UBI 環境に Zabbix パッケージをインストールする場合は、Zabbix packages for RHEL on Red Hat UBI environments を参照してください。
- ICMP ping アイテム を使用する場合、`fping` 用のパッケージは Zabbix Official Repository でも提供されています。

SELinux の設定

Zabbix は、ソケットベースのプロセス間通信を使用します。
Security-Enhanced Linux（SELinux）が有効になっているシステムでは、Zabbix が SocketDir ディレクトリ内で UNIX ドメインソケットを作成/使用できるようにするため、SELinux ルールの追加が必要になる場合があります。
ソケットファイルは、Zabbix サーバー（alerter、preprocessing、IPMI）および Zabbix プロキシ（IPMI）で使用され、プロセスの実行中に存在します。

SELinux が enforcing モードで有効になっている場合、Zabbix Web インターフェースとサーバー間の通信を有効にするには、次のコマンドを実行します。

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

```
setsebool -P httpd_can_connect_zabbix on
```

データベースにネットワーク経由でアクセスする場合（PostgreSQL の `localhost` を含む）は、Zabbix Web インターフェースからデータベースへの接続も許可します。

```
setsebool -P httpd_can_network_connect_db on
```

RHEL 7 より前の場合:

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

SELinux 設定を適用した後、Apache を再起動します。

```
systemctl restart httpd
```

必要に応じて、Zabbix Official Repository から事前定義済みの `zabbix-selinux-policy` パッケージをインストールすることもできます。
このパッケージは、サポート対象のすべての OS バージョン向けに提供されており、Zabbix の導入を簡素化し、設定の複雑さを理由にユーザーが SELinux を無効化してしまうことを防ぎます。

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

`zabbix-selinux-policy` パッケージには基本的な SELinux ポリシーが含まれており、Zabbix がソケットを作成および使用できるようにし、HTTPd から PostgreSQL（Web インターフェースで使用）への接続を有効にします。

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

```
module zabbix_policy 1.2;

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

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

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

Debuginfo パッケージ

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

`zabbix-debuginfo` リポジトリを有効にするには、次のようにします。

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

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo - $basearch
baseurl=http://repo.zabbix.com/zabbix/7.4/stable/rhel/7/$basearch/debuginfo/
```

```
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
gpgcheck=1
```

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

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/7.4/stable/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/7.4/stable/sles/15/x86_64/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

有効化したら、パッケージをインストールします。

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

```
dnf install zabbix-debuginfo
```

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

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

4 MSI で Windows エージェントのインストール

概要

Windows 版 Zabbix エージェントは、ダウンロード可能な Windows MSI インストーラーパッケージ (32bit 版と 64bit 版) を使用してインストールできます。

32bit 版のパッケージは 64bit 版の Windows にインストールすることはできません。

MSI インストールの最小要件は次のとおりです。

- **Zabbix** エージェントの場合**:** Windows XP 64-bit または Windows Server 2003
- **Zabbix** エージェント **2** の場合**:** Windows 10 32-bit または Windows Server 2016

32bit パッケージは、64bit システムにインストールできません。

パッケージは以下を含みます。

- TLS サポート (TLS 構成はオプションです)
- Zabbix getおよびZabbix senderユーティリティ (Zabbix エージェント/エージェント 2 と一緒または別々にインストールできます)

> **Attention:**
> Zabbix エージェント 2 パッケージには、ダウンロードしてインストールを個別に必要とする必要があるロード可能なプラグイン (MongoDB、PostgreSQL、MSSQL) は含まれていません。

インストールは、セットアップウィザードまたはコマンドラインを使用して実行できます。

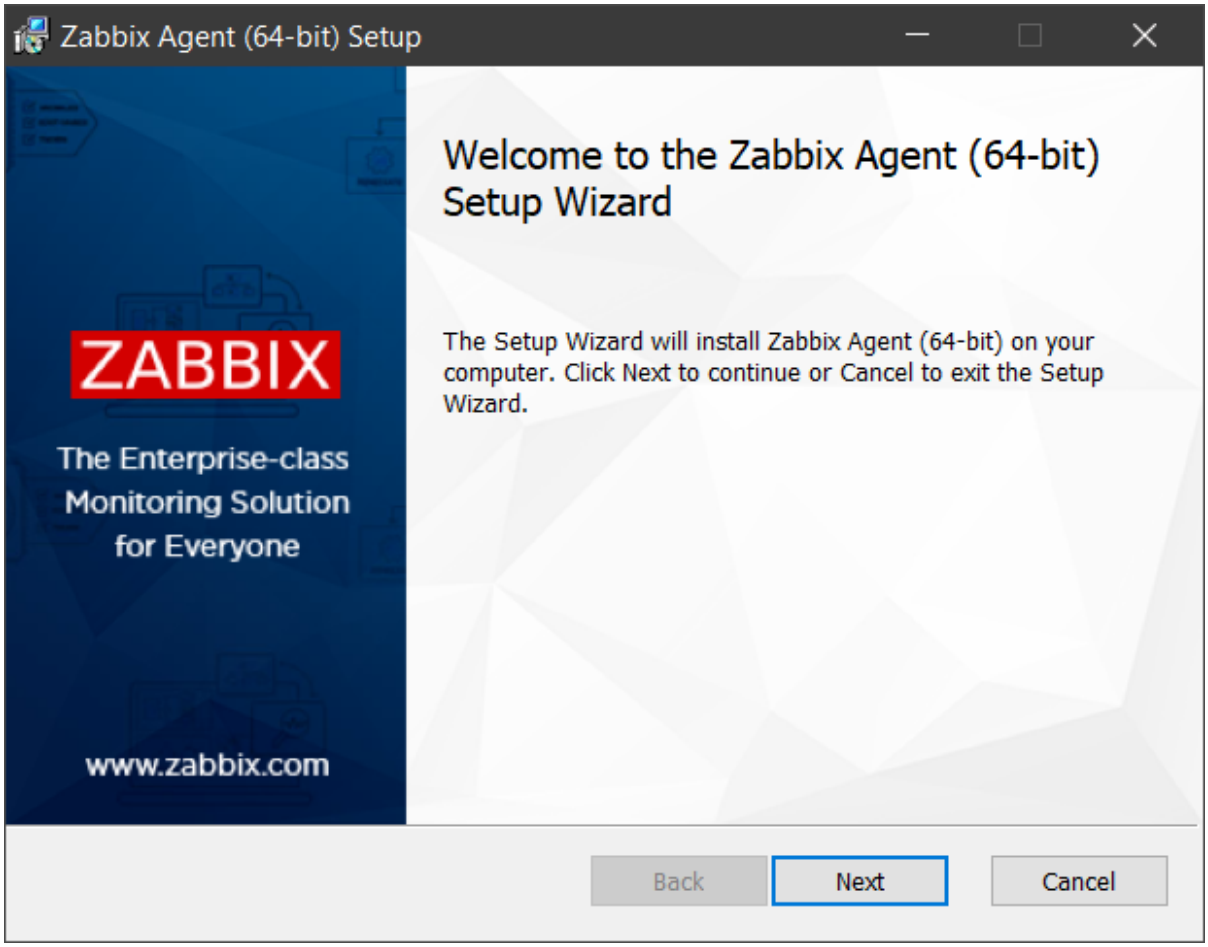
MSI パッケージを使用したインストールは完全にサポートされていますが、適切なエラー処理のため、少なくともMicrosoft .NET Framework 2をインストールすることをお勧めします。

> **Attention:**
> インストーラーが提供するデフォルトのインストール場所を使用することをお勧めします。必要な権限なしでカスタムロケーションを使用すると、インストールのセキュリティが損なわれる場合があります。

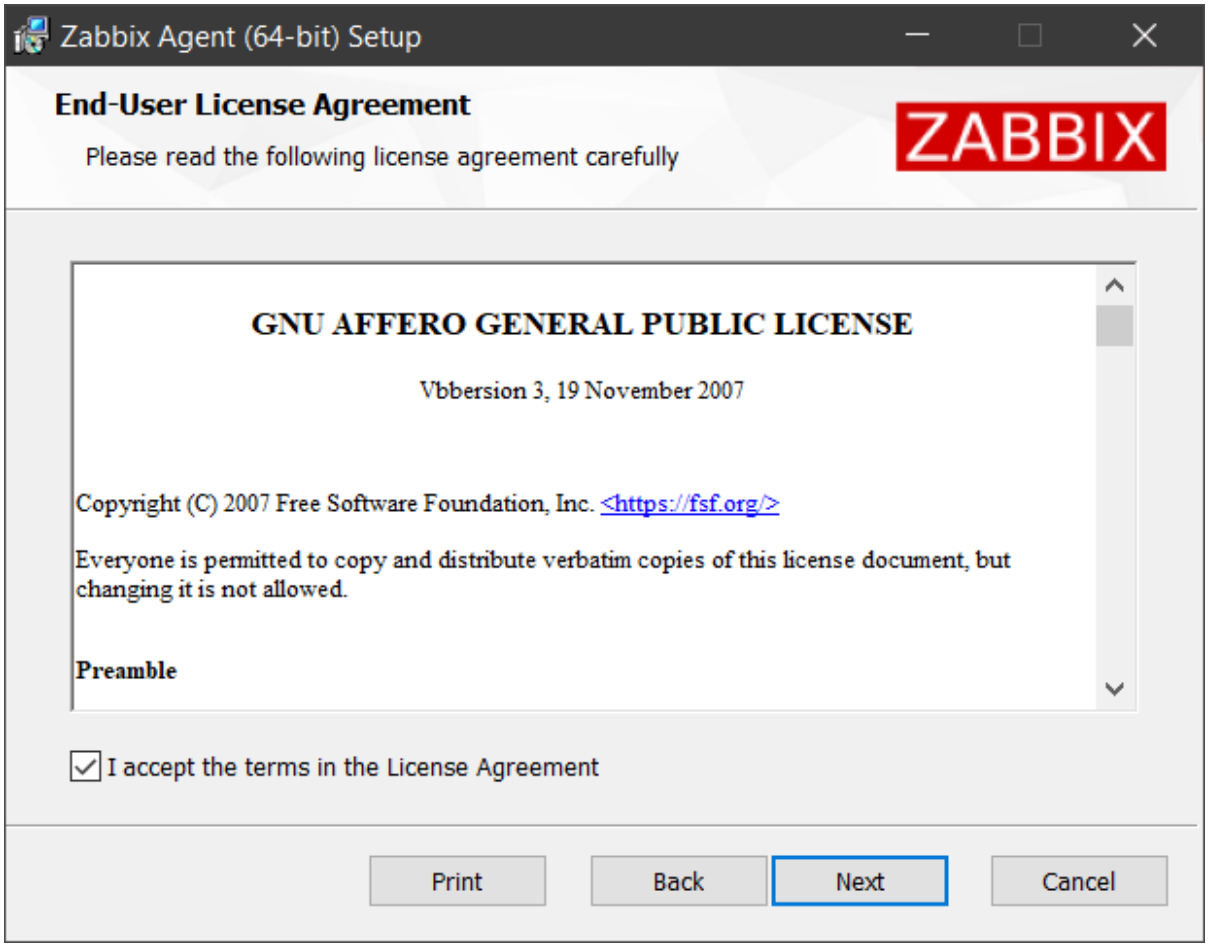
セットアップウィザードからのインストール

以下のインストール手順は、Zabbix エージェントと Zabbix エージェント 2 の両方に適用されます。

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

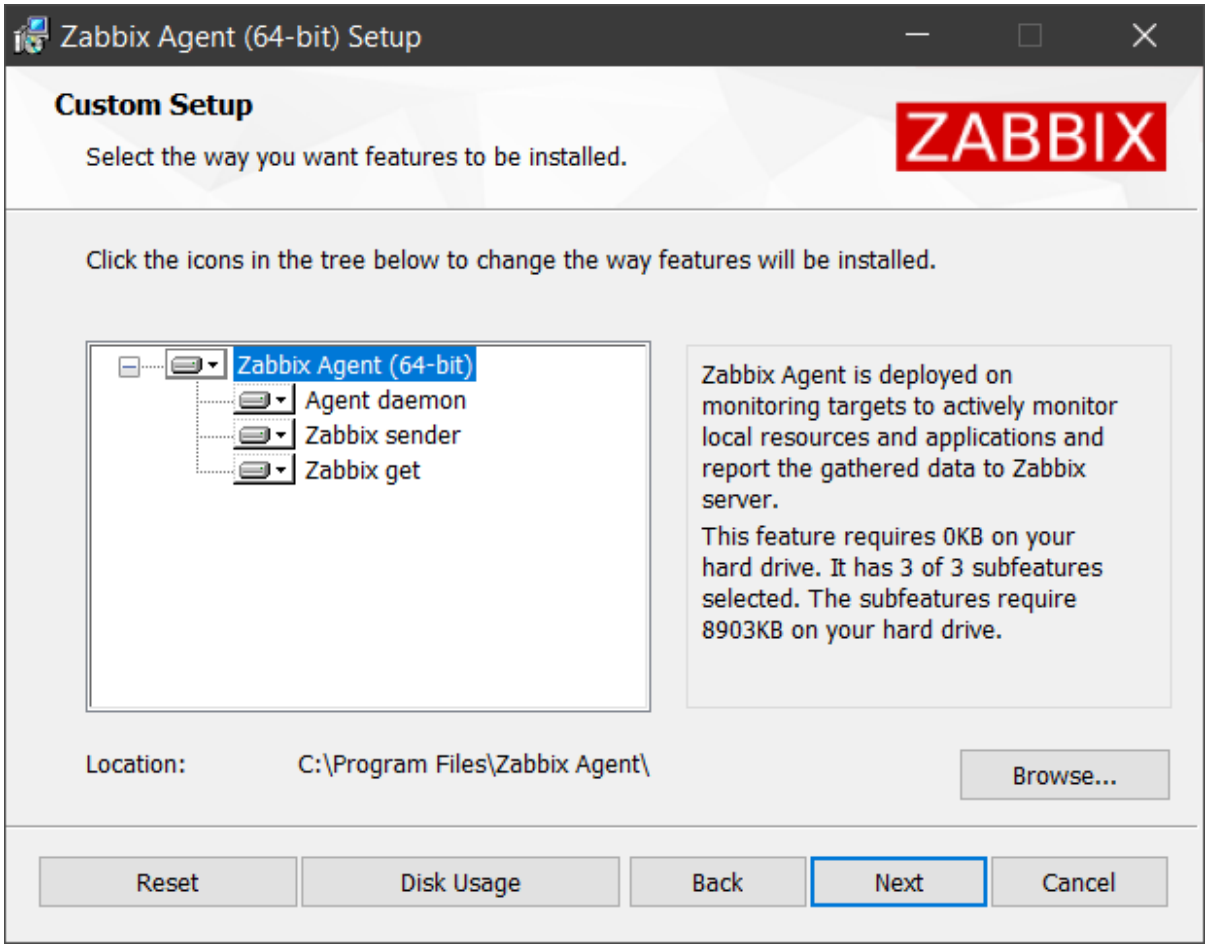


2. エンドユーザー使用許諾契約書に同意します。



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

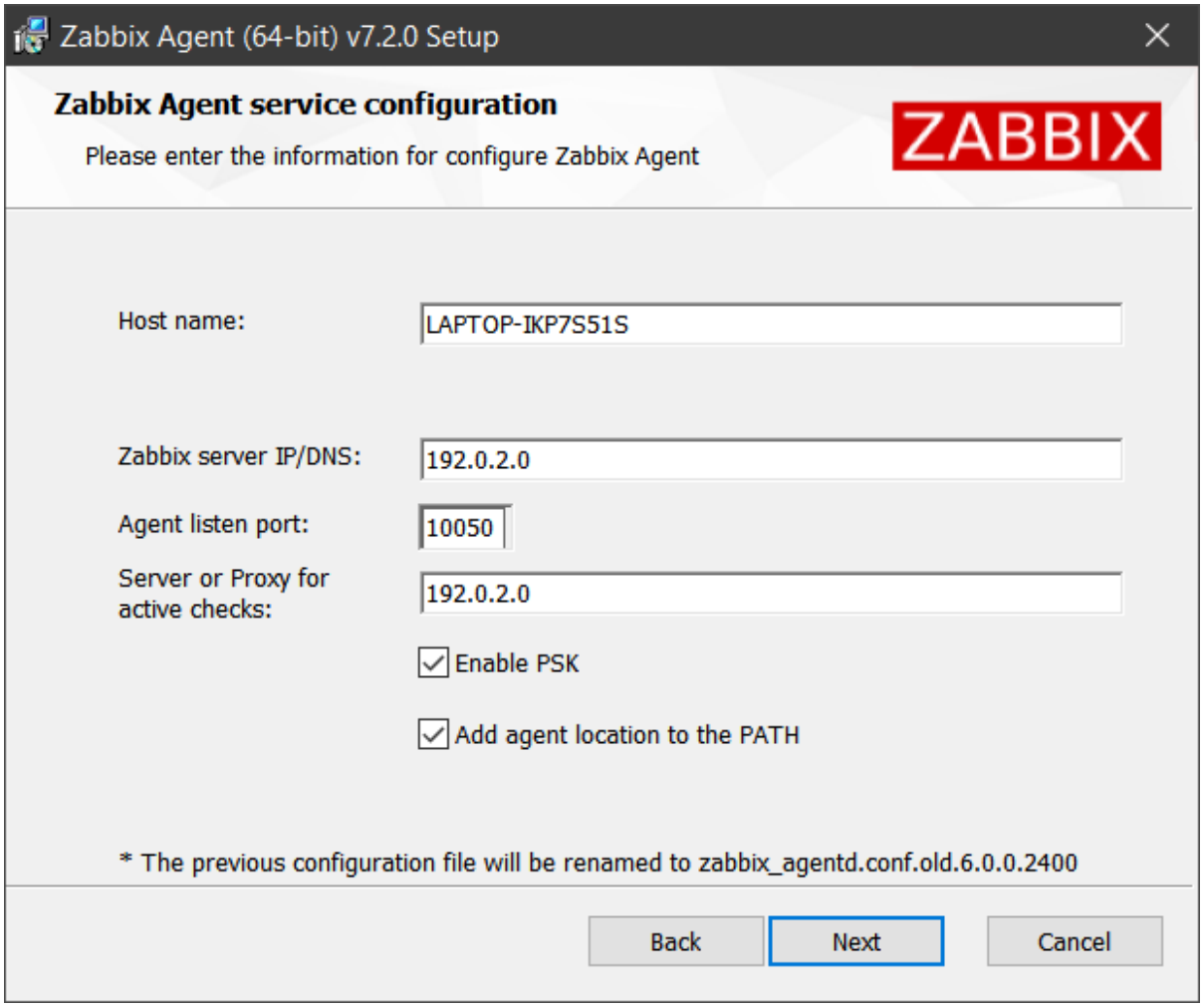
Attention:
インストーラーが提供するデフォルトのインストール先を使用することを推奨します。必要な権限のないカスタムの場所を使用すると、インストールのセキュリティが損なわれる可能性があります。



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

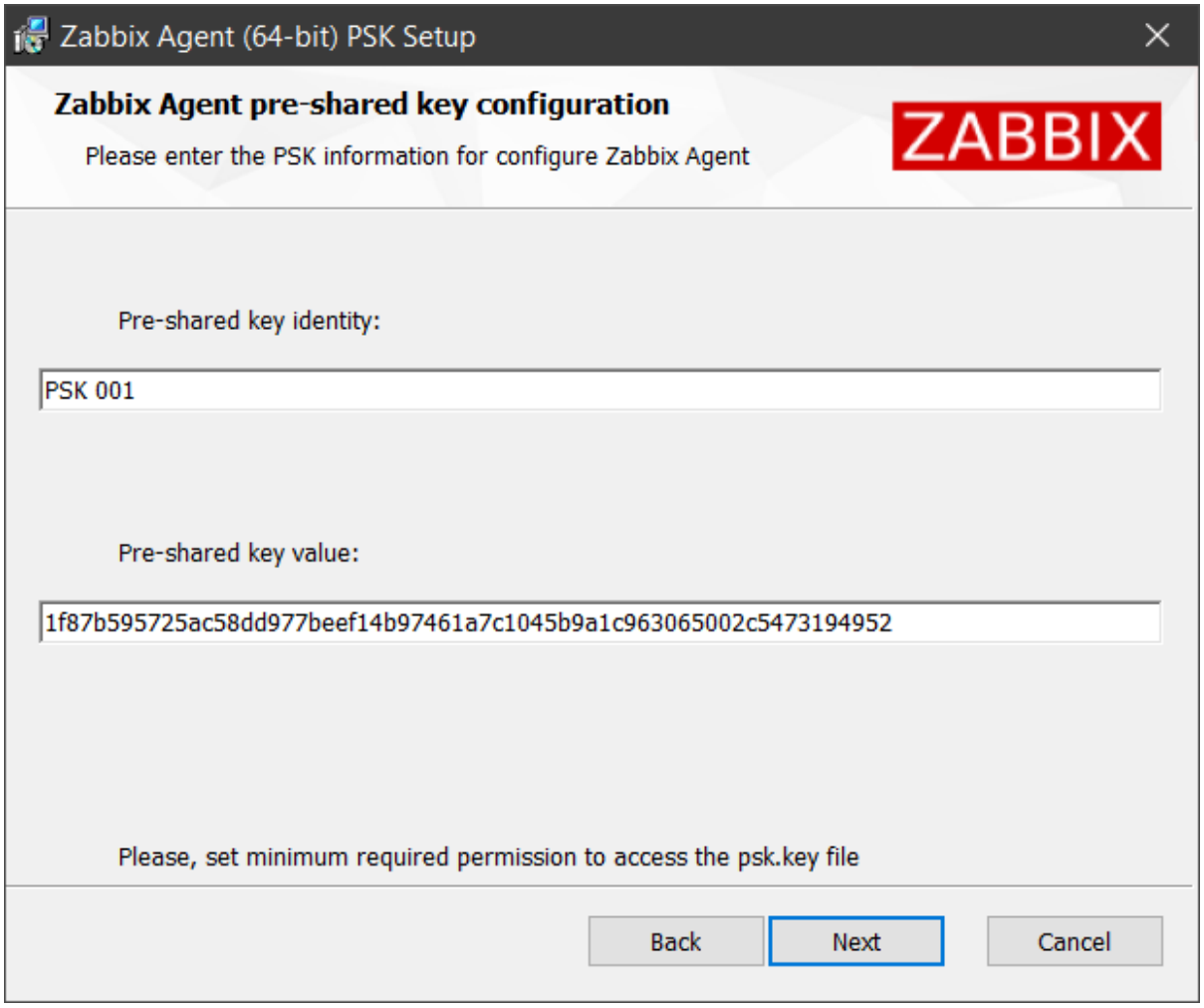
Parameter	Description
Host name	Zabbix エージェントをインストールするマシンのホスト名です。Hostname パラメータを設定します。
Zabbix server IP/DNS	カンマ区切りの IP アドレス一覧（CIDR 表記も可）または Zabbix サーバーもしくは Zabbix プロキシの DNS 名です。このパラメータは必須です。Server パラメータを設定します。
Agent listen port	エージェントがサーバーからの接続を待ち受けるポートです。ListenPort パラメータを設定します。
Server or Proxy for active checks	アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定です。サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、必要に応じてコロンで区切ったポート番号です。ServerActive パラメータを設定します。
Enable PSK	チェックボックスをオンにすると、事前共有鍵の使用による TLS サポートを有効にします。TLSConnect および TLSAccept パラメータを psk に設定します。
Add agent location to the PATH	チェックボックスをオンにすると、Zabbix エージェントの場所をシステムの PATH 変数に追加します。

Note:
既存の Zabbix エージェントが検出された場合、その設定ファイルのパラメータ値が表示されます。さらに、インストール中に既存の設定ファイルはリネームされ、新しい設定ファイルが作成されます。



5. 前の手順で *Enable PSK* チェックボックスをオンにした場合は、PSK パラメータを設定します。これらのパラメータも Zabbix エージェントの設定ファイルに設定されます。

Parameter	Description
Pre-shared key identity	事前共有鍵の識別文字列です。TLSPSKIdentity パラメータを設定します。
Pre-shared key value	事前共有鍵の文字列値です。鍵を含む psk.key ファイルを作成し、TLSPSKFile パラメータを鍵の場所（デフォルト: `C:\Program Files\Zabbix Agent\psk.key`）に設定します。ファイルのセキュリティ設定を調整し、Zabbix エージェント（またはエージェントを実行するユーザー）のみが事前共有鍵ファイルを読み取れるようにして、事前共有鍵ファイルへのアクセスを制限することを推奨します。

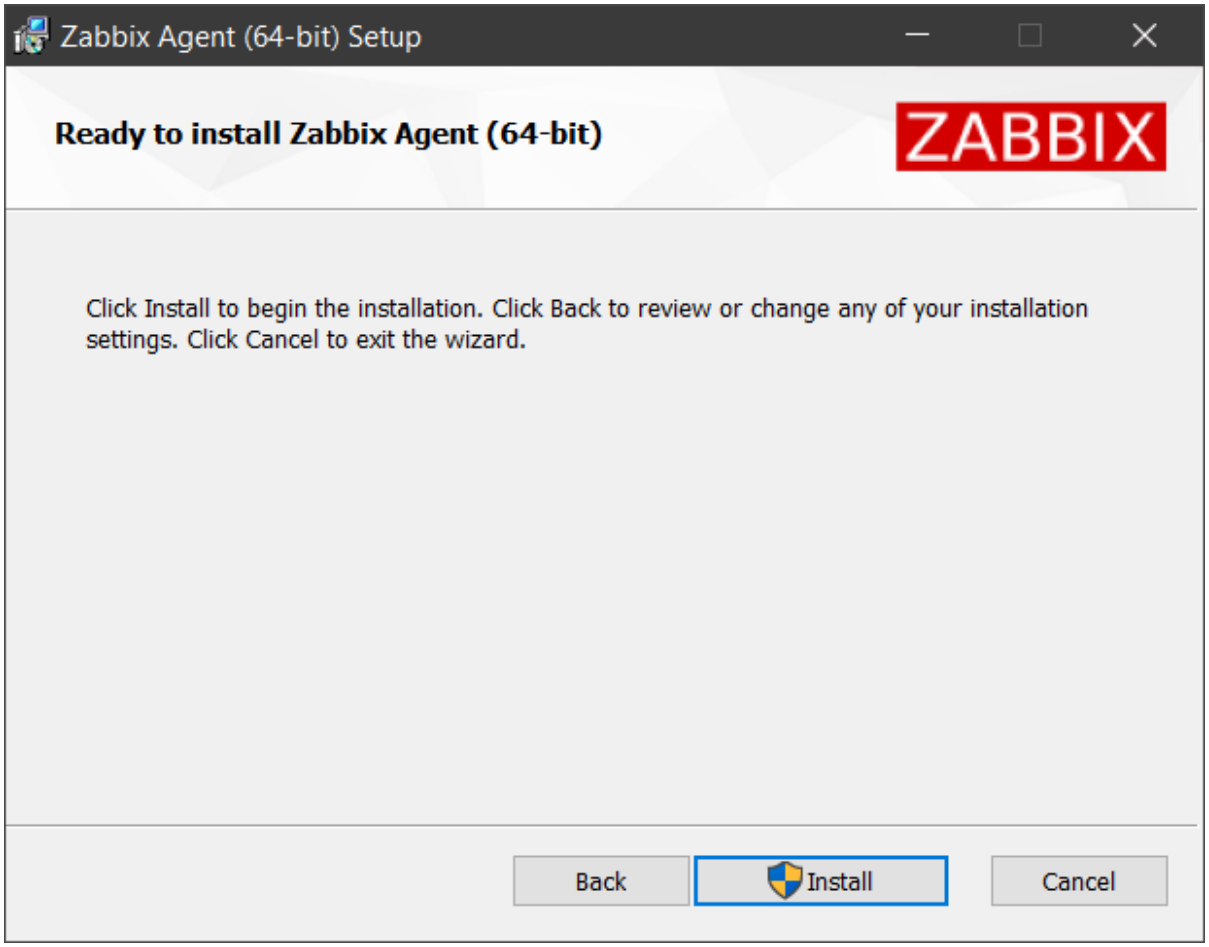


6. *Install* をクリックしてインストールを開始します。

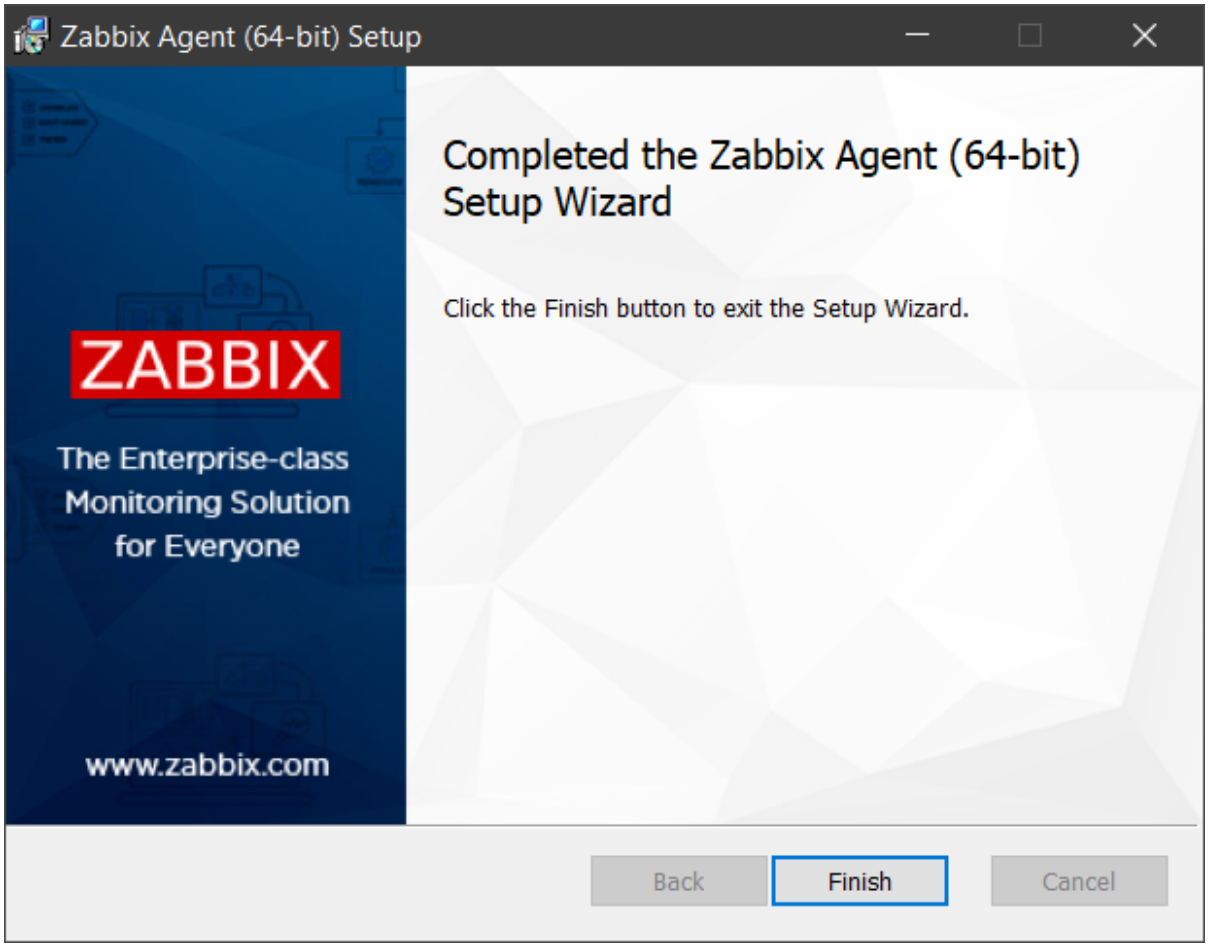
選択したすべての Zabbix コンポーネントと Zabbix エージェントの設定ファイルが、指定した場所（デフォルト: `C:\Program Files\Zabbix Agent`）にインストールされます。Zabbix エージェント 2 についても同様ですが、その組み込みプラグイン用の追加設定ファイルは `zabbix_agent2.d\plugins.d` サブフォルダにインストールされます。

さらに、zabbix_agentd.exe（または zabbix_agent2.exe）は、遅延自動開始の Windows サービスとして設定されます（Windows Vista/Server 2008 より前の Windows バージョンでは自動開始）。

インストール中に別バージョンの Zabbix エージェントが実行中の場合は、アプリケーションを閉じて再起動を試みるか、そのまま開いたままにするかを選択するよう求められます。後者の場合は再起動が必要になります。



7. *Finish* ボタンをクリックしてセットアップウィザードを終了します。



コマンドラインからのインストール

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

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.4.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

この方法では、パラメータを使用して無人インストールやカスタム設定を行うことができます。

サポートされるパラメータ

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

Note:
Zabbix エージェント/agent2 のパラメータは、インストール時に設定ファイルへ設定されます。パラメータ名をクリックすると、Zabbix agent (Windows) ページで詳細な説明と設定例を確認できます。Zabbix エージェント 2 については、Zabbix agent 2 (Windows) ページを参照してください。

Parameter	Description
ADDDEFAULT	デフォルト設定でインストールするコンポーネントのカンマ区切りリスト。詳細は ADDDEFAULT property を参照してください。 指定可能な値: AgentProgram, GetProgram, SenderProgram, ALL 例: ADDDEFAULT=AgentProgram,GetProgram
ADDLOCAL	ローカルにインストールするコンポーネントのカンマ区切りリスト。詳細は ADDLOCAL property を参照してください。 指定可能な値: AgentProgram, GetProgram, SenderProgram, ALL 例: ADDLOCAL=AgentProgram,SenderProgram
ALLOWDENYKEY	Zabbix エージェントのチェックを制限するための、セミコロン区切りの AllowKey または DenyKey パラメータのリスト。必要に応じて、区切り文字 (\;) をエスケープするためにバックスラッシュを使用してください。エージェント設定ファイル内のAllowKey およびDenyKey パラメータを設定します。 例: ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"
CONF	Zabbix エージェント用のテンプレート設定ファイルへのフルパス名。インストール時に、このファイルがエージェント設定ファイルになります。このファイルには少なくともServer およびLogFile パラメータが含まれている必要があります。 例: CONF="C:\full\path\to\example.conf"
DONOTSTART	DONOTSTART=1 を使用すると、MSI インストーラーが Zabbix エージェントサービスを起動しないようにします。 Zabbix 7.4.6 以降の MSI インストーラーパッケージでサポートされています。
ENABLEPATH	ENABLEPATH=1 を使用すると、Zabbix エージェントの場所をシステムの PATH 変数に追加します。
ENABLEPERSISTENTBUFFER	Zabbix エージェント 2 のみ。アクティブアイテム用のローカル永続ストレージの使用を有効にします。
HOSTINTERFACE	ホストインターフェースを定義する任意のパラメータです。
HOSTMETADATA	ホストメタデータを定義する任意のパラメータです。
HOSTMETADATAITEM	ホストメタデータの取得に使用するアイテムを定義する任意のパラメータです。
HOSTNAME	ホスト名を定義する任意のパラメータです。
HOSTNAMEITEM	ホスト名の取得に使用するアイテムを定義する任意のパラメータです。 Zabbix 7.4.6 以降の MSI インストーラーパッケージでサポートされています。
INCLUDE	Zabbix エージェント設定ファイルに含める、個別ファイルまたはディレクトリ内のすべてのファイルのセミコロン区切りリスト。
INSTALLFOLDER	Zabbix コンポーネントおよび Zabbix エージェント設定ファイルをインストールするフォルダへのフルパス名。Zabbix エージェント 2 では、組み込みプラグイン用の追加設定ファイルが zabbix_agent2.d\plugins.d サブフォルダにインストールされます。 例: INSTALLFOLDER="C:\Program Files\Zabbix Agent"
LISTENIP	エージェントが待ち受ける IP アドレスのカンマ区切りリスト。
LISTENPORT	エージェントはこのポートでサーバーからの接続を待ち受けます。
LOGFILE	Zabbix エージェントログファイルの名前。
LOGTYPE	ログ出力の種類。
NONMSICONFNAME	Zabbix エージェント用のカスタム設定ファイルへのフルパス名。インストール時に、このファイル内に存在する有効なエージェント設定パラメータ（この表に記載されているものに限る）が、新しく作成されるエージェント設定ファイルに書き込まれます。このファイルには少なくともServer パラメータが含まれている必要があります。 例: NONMSICONFNAME="C:\full\path\to\example.conf"
PERSISTENTBUFFERFILE	Zabbix エージェント 2 のみ。Zabbix エージェント 2 が SQLite データベースを保持するファイル。

Parameter	Description
PERSISTENTBUFFERPERIOD	Zabbix エージェント 2 のみ。サーバーまたはプロキシへの接続がない場合にデータを保存しておく期間。
SERVER	Zabbix サーバーまたは Zabbix プロキシの IP アドレス（CIDR 表記を任意で使用可能）または DNS 名のカンマ区切りリスト。このパラメータは、STARTAGENTS が 0 に設定されている場合を除き、必須です。
SERVERACTIVE	アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレスまたはクラスタ設定。
SKIP	SKIP=fw を使用すると、MSI インストーラーが Zabbix エージェント用の Windows Firewall 例外ルールを追加しないようにします。
SOURCEIP	Zabbix サーバーまたは Zabbix プロキシへの送信接続、または一部のアイテム（web.page.get、net.tcp.port など）の実行時に接続を行うための送信元 IP アドレス。 Zabbix 7.4.6 以降の MSI インストーラーパッケージでサポートされています。
STARTAGENTS	パッシブチェックを処理する zabbix_agentd の事前 fork されたインスタンス数。 0 に設定すると、パッシブチェックは無効になり、エージェントはいかなる TCP ポートでも待ち受けを行いません。 Zabbix 7.4.6 以降の MSI インストーラーパッケージでサポートされています。
STARTUPTYPE	Zabbix エージェントサービスの起動タイプ。指定可能な値: **automatic** - Windows 起動時にサービスを自動的に開始します。 **delayed** - *(デフォルト)* 自動起動するサービスの起動完了後に、サービスの開始を遅延させます (Windows Vista/Server 2008 以降で利用可能)。 **manual** - サービスを手動で開始します（ユーザーまたはアプリケーションによる）。 **disabled** - ユーザーまたはアプリケーションが開始できないようにサービスを無効化します。 例: STARTUPTYPE=disabled
STATUSPORT	Zabbix エージェント 2 のみ。設定すると、エージェントはこのポートで HTTP ステータスリクエストを待ち受けます (http://localhost:<port>/status)。
TIMEOUT	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。
TLSACCEPT	受け入れる受信接続（パッシブチェックに使用）。psk に設定すると、TLSCONNECT も psk に設定されます（別途指定されていない場合）。
TLSCAFILE	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルへのフルパス名。
TLSCERTFILE	エージェント証明書または証明書チェーンを含むファイルへのフルパス名。
TLSCONNECT	エージェントが Zabbix サーバーまたはプロキシへどのように接続するか（アクティブチェックに使用）。psk に設定すると、TLSACCEPT も psk に設定されます（別途指定されていない場合）。
TLSCRLFILE	失効した証明書を含むファイルへのフルパス名。
TLSKEYFILE	Zabbix エージェントの秘密鍵を含むファイルへのフルパス名。
TLSPSKFILE	Zabbix エージェントの事前共有鍵を含むファイルへのフルパス名。TLSPSKFILE と TLSPSKVALUE の両方が設定されている場合、TLSPSKVALUE の値が TLSPSKFILE で指定されたファイルに書き込まれます。事前共有鍵ファイルへのアクセスは、Zabbix エージェント（またはエージェントを実行するユーザー）のみが読み取れるよう、ファイルのセキュリティ設定を調整して制限することを 推奨します。
TLSPSKIDENTITY	事前共有鍵の識別文字列。
TLSPSKVALUE	事前共有鍵の文字列値。TLSPSKFILE と TLSPSKVALUE の両方が設定されている場合、TLSPSKVALUE の値が TLSPSKFILE で指定されたファイルに書き込まれます。 例: TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
TLSSERVERCERTISSUER	許可されるサーバー（プロキシ）証明書の発行者。
TLSSERVERCERTSUBJECT	許可されるサーバー（プロキシ）証明書のサブジェクト。
UNSAFEUSERPARAMETERS	ユーザー定義パラメータの引数にすべての文字を渡せるようにします。 Zabbix 7.4.6 以降の MSI インストーラーパッケージでサポートされています。

例

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

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.4.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.0^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
```

```
TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
ENABLEPATH=1^
ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

次の例では、新しいバージョンの Zabbix エージェントをインストールし、テンプレート設定ファイル(CONF="C:\agent-template.conf")を使用します。インストール中に、このファイルはエージェント設定ファイルになります。古い設定ファイルからパラメータを引き継ぐには、パラメータプレースホルダー（例: [AllowDenyKey]）を使用します。

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.4.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

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

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

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-7.4.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

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

Zabbix agent 2 のロード可能なプラグイン

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

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

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

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

セットアップウィザードからのインストール

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

2. エンドユーザー使用許諾契約書に同意します。

3. インストールする Zabbix エージェント 2 のロード可能プラグインを選択します。

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

4. *Install* をクリックして、インストールを開始します。

選択したすべての Zabbix エージェント 2 ロード可能プラグインは、指定した場所（デフォルト: C:\Program Files\Zabbix Agent 2）にインストールされ、設定ファイルは zabbix_agent2.d サブフォルダにインストールされます。

5. *Finish* ボタンをクリックして、セットアップウィザードを終了します。

コマンドラインからのインストール

Zabbix エージェント 2 のロード可能なプラグインは、msiexec を使用して MSI インストーラーを実行することで、コマンドラインからインストールできます。例:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent2_plugins-7.4.0-windows-amd64.msi" /qn+
```

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

Parameter	Description
ADDDEFAULT	デフォルト設定でインストールするコンポーネントのカンマ区切りリストです。詳細は ADDDEFAULT property を参照してください。 指定可能な値: ALL, EmberplusPlugin, MongodbPlugin, MssqlPlugin, NvidiagpuPlugin, PostgresqlPlugin 例: ADDDEFAULT=MongodbPlugin,PostgresqlPlugin
ADDLOCAL	ローカルにインストールするコンポーネントのカンマ区切りリストです。詳細は ADDLOCAL property を参照してください。 指定可能な値: ALL, EmberplusPlugin, MongodbPlugin, MssqlPlugin, NvidiagpuPlugin, PostgresqlPlugin 例: ADDLOCAL=MongodbPlugin,MssqlPlugin
INSTALLFOLDER	Zabbix コンポーネントをインストールするフォルダの完全パス名です。設定ファイルは zabbix_agent2.d サブフォルダにインストールされます。 例: INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"

5 PKG で MacOS エージェントのインストール

概要

Zabbix Mac OS エージェントは、ダウンロードで入手可能な PKG インストーラーパッケージからインストールできます。暗号化ありまたは暗号化なしのバージョンが利用可能です。

エージェントのインストール

エージェントは、グラフィカルユーザーインターフェースまたはコマンドラインを使用してインストールできます。例えば、次のように実行します。

```
sudo installer -pkg zabbix_agent-7.4.0-macos-arm64-openssl.pkg -target /
```

コマンドでは、正しい Zabbix パッケージのバージョンを使用してください。
ダウンロードしたパッケージ名と一致している必要があります。

エージェント実行

エージェントは、インストールまたは再起動後に自動的に開始されます。

必要に応じて設定ファイル/usr/local/etc/zabbix/zabbix_agentd.conf を編集してください。

エージェントを手動で開始するには、次のコマンドを実行します。

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

エージェントを手動で停止するには、次のコマンドを実行します。

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

アップグレード時、既存の設定ファイルは上書きされません。代わりに、新しい zabbix_agentd.conf.NEW ファイルが作成され、必要に応じて既存の設定ファイルを確認および更新するために使用します。設定ファイルを手動で変更した後は、忘れずにエージェントを再始動してください。

エージェントのトラブルシューティングと削除

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

Zabbix エージェントが実行されているかの確認:

```
ps aux | grep zabbix_agentd
```

Zabbix エージェントがパッケージからインストールされているかの確認:

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

インストーラーパッケージからインストールされたファイルの確認 (このビューには最初の/が表示されないことに注意してください):

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

launchctl で起動された Zabbix エージェントの停止:

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

インストーラーパッケージでインストールされたファイル (設定とログ含む) の削除:

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

Zabbix エージェントがインストールされていることを忘れさせる:

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

6 不安定版リリース

概要

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

まず、最新の zabbix-release パッケージをインストールするか、更新します。システムで rc パッケージを有効にするには、次の手順を実行します。

Red Hat Enterprise Linux

/etc/yum.repos.d/zabbix.repo ファイルを開き、zabbix-unstable リポジトリの enabled=1 を設定します。

```
[zabbix-unstable]
name=Zabbix Official Repository (unstable) - $basearch
baseurl=https://repo.zabbix.com/zabbix/7.4/unstable/rhel/8/$basearch/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
```

Debian/Ubuntu

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

```
#### Zabbix unstable repository
deb https://repo.zabbix.com/zabbix/7.4/unstable/debian bullseye main
deb-src https://repo.zabbix.com/zabbix/7.4/unstable/debian bullseye main
```

SUSE

/etc/zypp/repos.d/zabbix.repo ファイルを開き、zabbix-unstable リポジトリの enable=1 を設定します。

```
[zabbix-unstable]
name=Zabbix Official Repository
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/7.4/unstable/sles/15/x86_64/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/7.4/unstable/sles/15/x86_64/repodata/repomd.xml.key
enabled=1
update=1
```

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

Zabbix は公式に以下を提供しています。

- 各 Zabbix コンポーネントごとに個別の Docker イメージ。これにより、ポータブルで自己完結型のコンテナとして実行できます。
- Docker で複数コンテナの Zabbix コンポーネントを定義および実行するための Compose ファイル。

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

ソースファイル

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

Docker　Zabbix は、さまざまな OS ベースイメージに基づくイメージを提供しています。特定の Zabbix コンポーネントでサポートされているベースオペレーティングシステムイメージの一覧を取得するには、Docker Hub にある各コンポーネントの説明を参照してください。すべての Zabbix イメージは、ベースイメージが更新された場合に最新イメージを再ビルドするよう設定されています。

インストール

Zabbix コンポーネントのイメージを取得するには、次を実行します。

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

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

このコマンドは、Alpine Linux OS ベースの最新の安定版 Zabbix コンポーネントバージョンを取得します。別のオペレーティングシステムをベースにしたイメージ、または特定の Zabbix メジャー/マイナーバージョンのイメージを取得するには、リポジトリ名にtags を追加できます。

Docker Hub では、次のリポジトリを利用できます。

Component		Docker repository
Zabbix エージェント		zabbix/zabbix-agent
Zabbix サーバー		
	MySQL サポート付き	zabbix/zabbix-server-mysql
	PostgreSQL サポート付き	zabbix/zabbix-server-pgsql
Zabbix Web インターフェース		
	MySQL サポート付き Apache2 Web サーバーベース	zabbix/zabbix-web-apache-mysql
	PostgreSQL サポート付き Apache2 Web サーバーベース	zabbix/zabbix-web-apache-pgsql
	MySQL サポート付き Nginx Web サーバーベース	zabbix/zabbix-web-nginx-mysql
	PostgreSQL サポート付き Nginx Web サーバーベース	zabbix/zabbix-web-nginx-pgsql
Zabbix プロキシ		
	SQLite3 サポート付き	zabbix/zabbix-proxy-sqlite3
	MySQL サポート付き	zabbix/zabbix-proxy-mysql
Zabbix Java ゲートウェイ		zabbix/zabbix-java-gateway

Note:
SNMP トラップのサポートは、別リポジトリ zabbix/zabbix-snmptraps で提供されています。これは Zabbix サーバーおよび Zabbix プロキシとリンクできます。

タグ

公式の Zabbix コンポーネントイメージには、以下のタグを含めることができます。

Tag	Description	Example
latest	Alpine Linux イメージをベースにした Zabbix コンポーネントの最新の安定版。	zabbix-agent:latest
<OS>-trunk	特定のオペレーティングシステム上で現在開発中の Zabbix バージョンの最新ナイトリービルド。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	特定のオペレーティングシステム上の Zabbix コンポーネントの最新の安定版。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu	zabbix-agent:ol-latest
<OS>-X.X-latest	特定のメジャーバージョンおよびオペレーティングシステム向け Zabbix コンポーネントの最新マイナーバージョン。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu **X.X** - Zabbix のメジャーバージョン（例: *6.0*、*7.0*、*7.4*）。	zabbix-agent:alpine-7.4-latest
<OS>-X.X.*	特定のメジャーバージョンおよびオペレーティングシステム向け Zabbix コンポーネントの特定のマイナーバージョン。 **<OS>** - ベースとなるオペレーティングシステム。サポートされる値: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019（エージェントのみ）; *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022（エージェントのみ）; *ubuntu* - Ubuntu **X.X** - Zabbix のメジャーバージョン（例: *6.0*、*7.0*、*7.4*）。 * - Zabbix のマイナーバージョン	zabbix-agent:alpine-7.4.1

初期設定

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

Note:
Zabbix コンポーネント間の通信を有効にするため、Zabbix サーバー（trapper）用の 10051/TCP、Zabbix エージェント用の 10050/TCP、SNMP トラップ用の 162/UDP、Zabbix Web インターフェース用の 80/TCP などの一部のポートがホストマシンに公開されます。Zabbix コンポーネントで使用されるデフォルトポートの完全な一覧は、Requirementsページで確認できます。Zabbix サーバーおよびエージェントでは、デフォルトポートは ZBX_LISTENPORT 環境変数 を設定することで変更できます。

環境変数

すべての Zabbix コンポーネントイメージは、設定を制御するための環境変数を提供しています。サポートされている環境変数は、コンポーネントリポジトリに一覧表示されています。

これらの環境変数は Zabbix 設定ファイルのオプションに対応していますが、命名方法が異なります。例えば、ZBX_LOGSLOWQUERIES は、Zabbix サーバー または Zabbix プロキシ 設定ファイル内の LogSlowQueries に相当します。

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

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

Variable	Components	Default value	Description
DB_SERVER_HOST	サーバー プロキシ Web インターフェース	MYSQL の場合は mysql-server PostgreSQL の場合は postgres-server	MySQL または PostgreSQL サーバーの IP アドレスまたは DNS 名。
DB_SERVER_PORT	サーバー プロキシ Web インターフェース	MYSQL の場合は 3306 PostgreSQL の場合は 5432	MySQL または PostgreSQL サーバーのポート。
MYSQL_USER	サーバー プロキシ Web インターフェース	zabbix	MySQL データベースユーザー。
MYSQL_PASSWORD	サーバー プロキシ Web インターフェース	zabbix	MySQL データベースパスワード。
MYSQL_DATABASE	サーバー プロキシ Web インターフェース	Zabbix サーバーの場合は zabbix Zabbix プロキシの場合は zabbix_proxy	Zabbix データベース名。
POSTGRES_USER	サーバー Web インターフェース	zabbix	PostgreSQL データベースユーザー。
POSTGRES_PASSWORD	サーバー Web インターフェース	zabbix	PostgreSQL データベースパスワード。
POSTGRES_DB	サーバー Web インターフェース	Zabbix サーバーの場合は zabbix Zabbix プロキシの場合は zabbix_proxy	Zabbix データベース名。
PHP_TZ	Web インターフェース	Europe/Riga	PHP 形式のタイムゾーン。サポートされているタイムゾーンの完全な一覧は php.net で確認できます。
ZBX_SERVER_NAME	Web インターフェース	Zabbix Docker	Web インターフェースの縦型メニューで、Zabbix ロゴの下に表示される Zabbix インストール名。
ZBX_JAVAGATEWAY_ENABLE	サーバー プロキシ	false	Java 関連のチェックを収集するために、Zabbix Java ゲートウェイとの通信を有効にします。
ZBX_ENABLE_SNMP_TRAPS	サーバー プロキシ	false	SNMP トラップ機能を有効にします。これには **zabbix-snmptraps** インスタンスと、共有ボリューム */var/lib/zabbix/snmptraps* を Zabbix サーバーまたは Zabbix プロキシに設定する必要があります。

ボリューム

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

ボリューム	説明
Zabbix エージェント	
/etc/zabbix/zabbix_agentd.d	このボリュームを使用すると、**.conf* ファイルを追加でき、UserParameter 機能を使用して Zabbix エージェントを拡張できます
/var/lib/zabbix/modules	このボリュームを使用すると、追加モジュールをロードでき、LoadModule 機能を使用して Zabbix エージェントを拡張できます
/var/lib/zabbix/enc	このボリュームは TLS 関連ファイルの保存に使用されます。これらのファイル名は、環境変数 ZBX_TLSCAFILE、ZBX_TLSCRLFILE、ZBX_TLSKEY_FILE、ZBX_TLSPSKFILE を使用して指定します
Zabbix サーバー	
/usr/lib/zabbix/alertscripts	このボリュームはカスタムアラートスクリプトに使用されます。zabbix_server.conf の AlertScriptsPath パラメータです
/usr/lib/zabbix/externalscripts	このボリュームは外部チェック で使用されます。zabbix_server.conf の ExternalScripts パラメータです
/var/lib/zabbix/modules	このボリュームを使用すると、追加モジュールをロードでき、LoadModule 機能を使用して Zabbix サーバーを拡張できます
/var/lib/zabbix/enc	このボリュームは TLS 関連ファイルの保存に使用されます。これらのファイル名は、環境変数 ZBX_TLSCAFILE、ZBX_TLSCRLFILE、ZBX_TLSKEY_FILE、ZBX_TLSPSKFILE を使用して指定します
/var/lib/zabbix/ssl/certs	このボリュームは、クライアント認証用の SSL クライアント証明書ファイルの保存場所として使用されます。zabbix_server.conf の SSLCertLocation パラメータです
/var/lib/zabbix/ssl/keys	このボリュームは、クライアント認証用の SSL 秘密鍵ファイルの保存場所として使用されます。zabbix_server.conf の SSLKeyLocation パラメータです
/var/lib/zabbix/ssl/ssl_ca	このボリュームは、SSL サーバー証明書検証用の認証局 (CA) ファイルの保存場所として使用されます。zabbix_server.conf の SSLCALocation パラメータです
/var/lib/zabbix/snmptraps	このボリュームは snmptraps.log ファイルの保存場所として使用されます。このボリュームは zabbix-snmptraps コンテナと共有でき、新しい Zabbix サーバーインスタンスの作成時に Docker の volumes_from オプションを使用して継承できます。SNMP トラップ処理機能は、共有ボリュームを使用し、環境変数 ZBX_ENABLE_SNMP_TRAPS を'true' に切り替えることで有効にできます
/var/lib/zabbix/mibs	このボリュームを使用すると、新しい MIB ファイルを追加できます。サブディレクトリはサポートされていないため、すべての MIB は /var/lib/zabbix/mibs に配置する必要があります
Zabbix プロキシ	
/usr/lib/zabbix/externalscripts	このボリュームは外部チェック で使用されます。zabbix_proxy.conf の ExternalScripts パラメータです
/var/lib/zabbix/db_data	このボリュームを使用すると、データベースファイルを外部デバイスに保存できます。SQLite3 を使用する Zabbix プロキシでのみサポートされます
/var/lib/zabbix/modules	このボリュームを使用すると、追加モジュールをロードでき、LoadModule 機能を使用して Zabbix サーバーを拡張できます
/var/lib/zabbix/enc	このボリュームは TLS 関連ファイルの保存に使用されます。これらのファイル名は、環境変数 ZBX_TLSCAFILE、ZBX_TLSCRLFILE、ZBX_TLSKEY_FILE、ZBX_TLSPSKFILE を使用して指定します
/var/lib/zabbix/ssl/certs	このボリュームは、クライアント認証用の SSL クライアント証明書ファイルの保存場所として使用されます。zabbix_proxy.conf の SSLCertLocation パラメータです
/var/lib/zabbix/ssl/keys	このボリュームは、クライアント認証用の SSL 秘密鍵ファイルの保存場所として使用されます。zabbix_proxy.conf の SSLKeyLocation パラメータです

ボリューム		説明
	/var/lib/zabbix/ssl/ssl_ca	このボリュームは、SSL サーバー証明書検証用の認証局 (CA) ファイルの保存場所として使用されます。zabbix_proxy.conf の SSLCALocation パラメータです
	/var/lib/zabbix/snmptraps	このボリュームは snmptraps.log ファイルの保存場所として使用されます。このボリュームは zabbix-snmptraps コンテナと共有でき、新しい Zabbix サーバーインスタンスの作成時に Docker の volumes_from オプションを使用して継承できます。SNMP トラップ処理機能は、共有ボリュームを使用し、環境変数 ZBX_ENABLE_SNMP_TRAPS を'true' に切り替えることで有効にできます
	/var/lib/zabbix/mibs	このボリュームを使用すると、新しい MIB ファイルを追加できます。サブディレクトリはサポートされていないため、すべての MIB は /var/lib/zabbix/mibs に配置する必要があります
Apache2 Web サーバーをベースとした **Zabbix Web** インターフェース		
	/etc/ssl/apache2	このボリュームを使用すると、Zabbix Web インターフェースで HTTPS を有効にできます。このボリュームには、Apache2 の SSL 接続用に準備された ssl.crt と ssl.key の 2 つのファイルが含まれている必要があります

ボリューム	説明
Nginx Web サーバーをベースとした Zabbix Web インターフェース	
/etc/ssl/nginx	このボリュームを使用すると、Zabbix Web インターフェースで HTTPS を有効にできます。このボリュームには、Nginx の SSL 接続用に準備された `ssl.crt`、`ssl.key`、`dhparam.pem` が含まれている必要があります
Zabbix snmptraps	
/var/lib/zabbix/snmptraps	このボリュームには、受信した SNMP トラップを記録した `snmptraps.log` ログファイルが含まれます
/var/lib/zabbix/mibs	このボリュームを使用すると、新しい MIB ファイルを追加できます。サブディレクトリはサポートされていないため、すべての MIB は `/var/lib/zabbix/mibs` に配置する必要があります

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

例

例 **1**

この例では、MySQL データベース対応の Zabbix サーバー、Nginx Web サーバーベースの Zabbix Web インターフェース、および Zabbix Java ゲートウェイを実行する方法を示します。

1. Zabbix コンポーネントコンテナ専用のネットワークを作成します。

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

2. 空の MySQL サーバーインスタンスを起動します。

> **Attention:**
> MySQL 8.4 以降では、`mysql_native_password` の代わりに `caching_sha2_password` を使用する必要があります。

```
docker run --name mysql-server -t \
      -e MYSQL_DATABASE="zabbix" \
      -e MYSQL_USER="zabbix" \
      -e MYSQL_PASSWORD="zabbix_pwd" \
      -e MYSQL_ROOT_PASSWORD="root_pwd" \
      --network=zabbix-net \
```

```
      --restart unless-stopped \
      -d mysql:8.0-oracle \
      --character-set-server=utf8 --collation-server=utf8_bin \
      --default-authentication-plugin=mysql_native_password
```

3. Zabbix Java ゲートウェイインスタンスを起動します。

```
docker run --name zabbix-java-gateway -t \
      --network=zabbix-net \
      --restart unless-stopped \
      -d zabbix/zabbix-java-gateway:alpine-7.4-latest
```

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

```
docker run --name zabbix-server-mysql -t \
      -e DB_SERVER_HOST="mysql-server" \
      -e MYSQL_DATABASE="zabbix" \
      -e MYSQL_USER="zabbix" \
      -e MYSQL_PASSWORD="zabbix_pwd" \
      -e MYSQL_ROOT_PASSWORD="root_pwd" \
      -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
      --network=zabbix-net \
      -p 10051:10051 \
      --restart unless-stopped \
      -d zabbix/zabbix-server-mysql:alpine-7.4-latest
```

5. Zabbix Web インターフェースを起動し、作成した MySQL サーバーインスタンスおよび Zabbix サーバーインスタンスにリンクします。

```
docker run --name zabbix-web-nginx-mysql -t \
      -e ZBX_SERVER_HOST="zabbix-server-mysql" \
      -e DB_SERVER_HOST="mysql-server" \
      -e MYSQL_DATABASE="zabbix" \
      -e MYSQL_USER="zabbix" \
      -e MYSQL_PASSWORD="zabbix_pwd" \
      -e MYSQL_ROOT_PASSWORD="root_pwd" \
      --network=zabbix-net \
      -p 80:8080 \
      --restart unless-stopped \
      -d zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest
```

例 **2**

この例では、PostgreSQL データベース対応の Zabbix サーバー、Nginx Web サーバーベースの Zabbix Web インターフェース、および SNMP トラップ機能を実行する方法を示します。

1. Zabbix コンポーネントコンテナ専用のネットワークを作成します。

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

2. 空の PostgreSQL サーバーインスタンスを起動します。

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

3. Zabbix snmptraps インスタンスを起動します。

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

```
      --restart unless-stopped \
      -d zabbix/zabbix-snmptraps:alpine-7.4-latest
```

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

```
docker run --name zabbix-server-pgsql -t \
      -e DB_SERVER_HOST="postgres-server" \
      -e POSTGRES_USER="zabbix" \
      -e POSTGRES_PASSWORD="zabbix_pwd" \
      -e POSTGRES_DB="zabbix" \
      -e ZBX_ENABLE_SNMP_TRAPS="true" \
      --network=zabbix-net \
      -p 10051:10051 \
      --volumes-from zabbix-snmptraps \
      --restart unless-stopped \
      -d zabbix/zabbix-server-pgsql:alpine-7.4-latest
```

5. Zabbix Web インターフェースを起動し、作成した PostgreSQL サーバーインスタンスおよび Zabbix サーバーインスタンスにリンクします。

```
docker run --name zabbix-web-nginx-pgsql -t \
      -e ZBX_SERVER_HOST="zabbix-server-pgsql" \
      -e DB_SERVER_HOST="postgres-server" \
      -e POSTGRES_USER="zabbix" \
      -e POSTGRES_PASSWORD="zabbix_pwd" \
      -e POSTGRES_DB="zabbix" \
      --network=zabbix-net \
      -p 443:8443 \
      -p 80:8080 \
      -v /etc/ssl/nginx:/etc/ssl/nginx:ro \
      --restart unless-stopped \
      -d zabbix/zabbix-web-nginx-pgsql:alpine-7.4-latest
```

例 **3**

この例では、Red Hat 8 上で podman を使用して、MySQL データベース対応の Zabbix サーバー、Nginx Web サーバーベースの Zabbix Web インターフェース、および Zabbix Java ゲートウェイを実行する方法を示します。

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

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

2. （任意）zabbix pod 内で Zabbix エージェントコンテナを起動します。

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

3. ホスト上に ./mysql/ ディレクトリを作成し、Oracle MySQL サーバー 8.0 を起動します。

> **Attention:**
> MySQL 8.4 以降では、mysql_native_password の代わりに caching_sha2_password を使用する必要があります。

```
podman run --name mysql-server -t \
      -e MYSQL_DATABASE="zabbix" \
      -e MYSQL_USER="zabbix" \
      -e MYSQL_PASSWORD="zabbix_pwd" \
      -e MYSQL_ROOT_PASSWORD="root_pwd" \
      -v ./mysql/:/var/lib/mysql/:Z \
      --restart=always \
      --pod=zabbix \
      -d mysql:8.0 \
      --character-set-server=utf8 --collation-server=utf8_bin \
      --default-authentication-plugin=mysql_native_password
```

4. Zabbix サーバーコンテナを起動します。

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

5. Zabbix Java Gateway コンテナを起動します。

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

6. Zabbix Web インターフェースコンテナを起動します。

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

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

Docker Compose　または、Zabbix は Docker Compose プラグインを使用してインストールすることもできます。複数コンテナの Zabbix コンポーネントを定義して実行するための Compose ファイルは、GitHub 上の公式のZabbix Docker リポジトリで利用できます。

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

これらの compose ファイルは例として追加されていますが、多くの構成が含まれています。たとえば、MySQL と SQLite3 の両方をサポートするプロキシが含まれています。

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

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

必要なバージョンに切り替えます。

```
git checkout 7.4
```

Compose 設定ファイルを使用してコンテナを作成し、起動します。

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

上記のコマンド内の docker-compose_v3_alpine_mysql_latest.yaml を、必要な設定ファイルに置き換えてください。

利用可能なオプションは次のとおりです。

File name	Description
docker-compose_v3_alpine_mysql_latest.yaml	Alpine Linux 上で MySQL データベースをサポートする最新バージョンの Zabbix 7.4 コンポーネントを実行します。

File name	Description
docker-compose_v3_alpine_mysql_local.yaml	最新バージョンの Zabbix 7.4 をローカルでビルドし、Alpine Linux 上で MySQL データベースをサポートする Zabbix コンポーネントを実行します。
docker-compose_v3_alpine_pgsql_latest.yaml	Alpine Linux 上で PostgreSQL データベースをサポートする最新バージョンの Zabbix 7.4 コンポーネントを実行します。
docker-compose_v3_alpine_pgsql_local.yaml	最新バージョンの Zabbix 7.4 をローカルでビルドし、Alpine Linux 上で PostgreSQL データベースをサポートする Zabbix コンポーネントを実行します。
docker-compose_v3_ol_mysql_latest.yaml	は、Oracle Linux 上で MySQL データベースをサポートする最新バージョンの Zabbix 7.4 コンポーネントを実行します。
docker-compose_v3_ol_mysql_local.yaml	は、最新バージョンの Zabbix 7.4 をローカルでビルドし、Oracle Linux 上で MySQL データベースをサポートする Zabbix コンポーネントを実行します。
docker-compose_v3_ol_pgsql_latest.yaml	は、Oracle Linux 上で PostgreSQL データベースをサポートする最新バージョンの Zabbix 7.4 コンポーネントを実行します。
docker-compose_v3_ol_pgsql_local.yaml	は、最新バージョンの Zabbix 7.4 をローカルでビルドし、Oracle Linux 上で PostgreSQL データベースをサポートする Zabbix コンポーネントを実行します。
docker-compose_v3_ubuntu_mysql_latest.yaml	Ubuntu 24.04 (noble) 上で MySQL データベースをサポートする最新バージョンの Zabbix 7.4 コンポーネントを実行します。
docker-compose_v3_ubuntu_mysql_local.yaml	最新バージョンの Zabbix 7.4 をローカルでビルドし、Ubuntu 24.04 (noble) 上で MySQL データベースをサポートする Zabbix コンポーネントを実行します。
docker-compose_v3_ubuntu_pgsql_latest.yaml	Ubuntu 24.04 (noble) 上で PostgreSQL データベースをサポートする最新バージョンの Zabbix 7.4 コンポーネントを実行します。
docker-compose_v3_ubuntu_pgsql_local.yaml	最新バージョンの Zabbix 7.4 をローカルでビルドし、Ubuntu 24.04 (noble) 上で PostgreSQL データベースをサポートする Zabbix コンポーネントを実行します。

ストレージ

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

/etc/localtime および/etc/timezone ファイルに対して読み取り専用モードのボリュームもあります。

環境変数

変数ファイルは、.env_< コンポーネントの種類 > という名前構造を持ち、*env_vars*ディレクトリにあります。変数の名前と利用可能な選択の詳細については、環境変数を参照してください。

例

例 **1**

```
git checkout 7.4
docker compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up -d
```

このコマンドは、各 Zabbix コンポーネント向けの最新の Zabbix 7.4 イメージをダウンロードし、デタッチモードで実行します。

> **Attention:**
> compose ファイルとともに、github.com の公式 Zabbix リポジトリから .env_<type of component> ファイルをダウンロードすることを忘れないでください。

例 **2**

```
git checkout 7.4
docker compose -f ./docker-compose_v3_ubuntu_mysql_local.yaml up -d
```

このコマンドは、ベースイメージ Ubuntu 24.04（noble）をダウンロードし、その後 Zabbix 7.4 コンポーネントをローカルでビルドして、デタッチモードで実行します。

6 Web インターフェースのインストール

このセクションでは、Zabbix Web インターフェースをインストールするための手順を順を追って説明します。
Zabbix Web インターフェースは PHP で記述されているため、実行するには PHP をサポートする Web サーバーが必要です。

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

ようこそ画面

ブラウザで Zabbix Web インターフェースの URL を開きます。パッケージから Zabbix をインストールした場合、URL は次のとおりです。

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

Web インターフェースのインストールウィザードの最初の画面が表示されます。

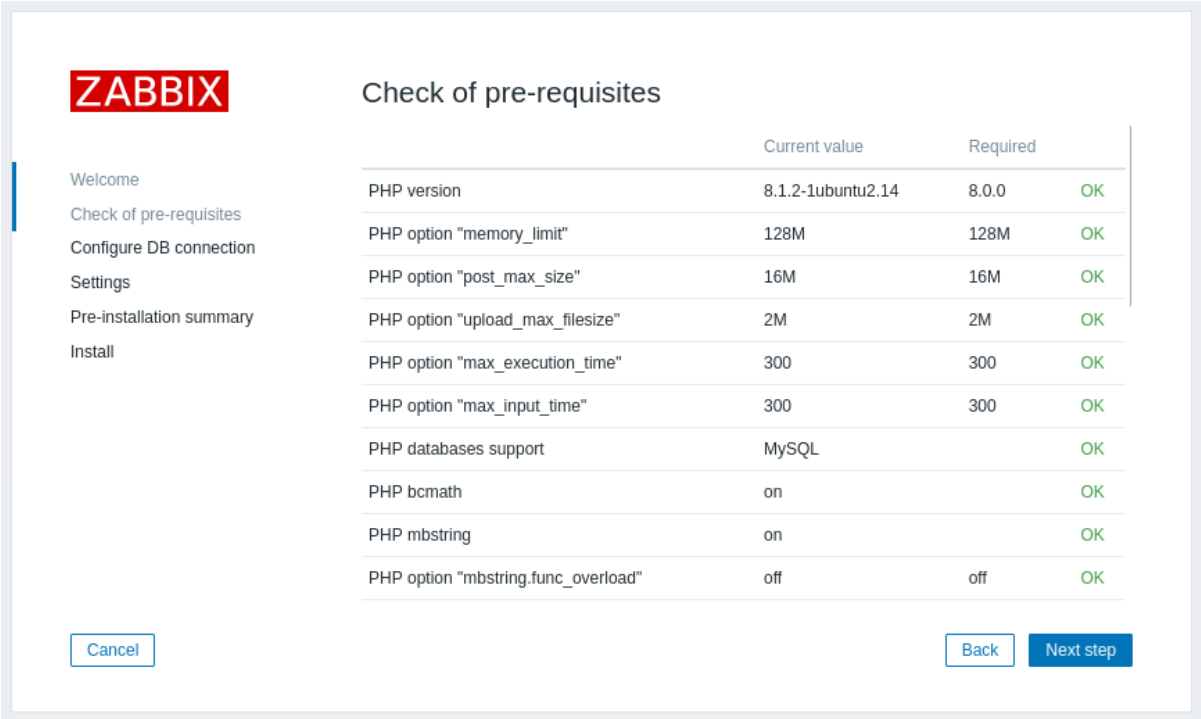
Default language ドロップダウンメニューを使用してシステムのデフォルト言語を変更し、選択した言語でインストールプロセスを続行します（任意）。詳細については、追加の Web インターフェース言語のインストールを参照してください。

言語を *English (en_US)* に設定すると、Web インターフェースでも米国式の時刻/日付形式が有効になることに注意してください。



前提条件の確認

Zabbix Web インターフェースの必須前提条件がすべて満たされていることを確認してください。



前提条件	最小値	説明
PHP version	8.0.0	
PHP option "memory_limit"	128MB	php.ini 内: memory_limit = 128M
PHP option "post_max_size"	16MB	php.ini 内: post_max_size = 16M
PHP option "upload_max_filesize"	2MB	php.ini 内: upload_max_filesize = 2M
PHP option "max_execution_time"	300 秒	php.ini 内: max_execution_time = 300（値 0 および -1 も使用可能）
PHP option "max_input_time"	300 秒	php.ini 内: max_input_time = 300（値 0 および -1 も使用可能）
PHP databases support	次のいずれか: MySQL、PostgreSQL	必須およびオプションのすべての PHP 拡張機能の一覧については、Requirementsを参照してください。オプションの前提条件を満たしていない場合は、赤い *Warning* ステータスで表示されますが、それらが満たされていなくてもセットアップ処理は続行できます。
PHP bcmath	有効である必要があります	
PHP mbstring	有効である必要があります	
PHP option "mbstring.func_overload"	無効である必要があります	php.ini 内: mbstring.func_overload = 0
PHP option "session.auto_start"	無効である必要があります	php.ini 内: session.auto_start = 0
PHP option "arg_separator.output"	&	php.ini 内: arg_separator.output = "&"（値"&" も使用可能）

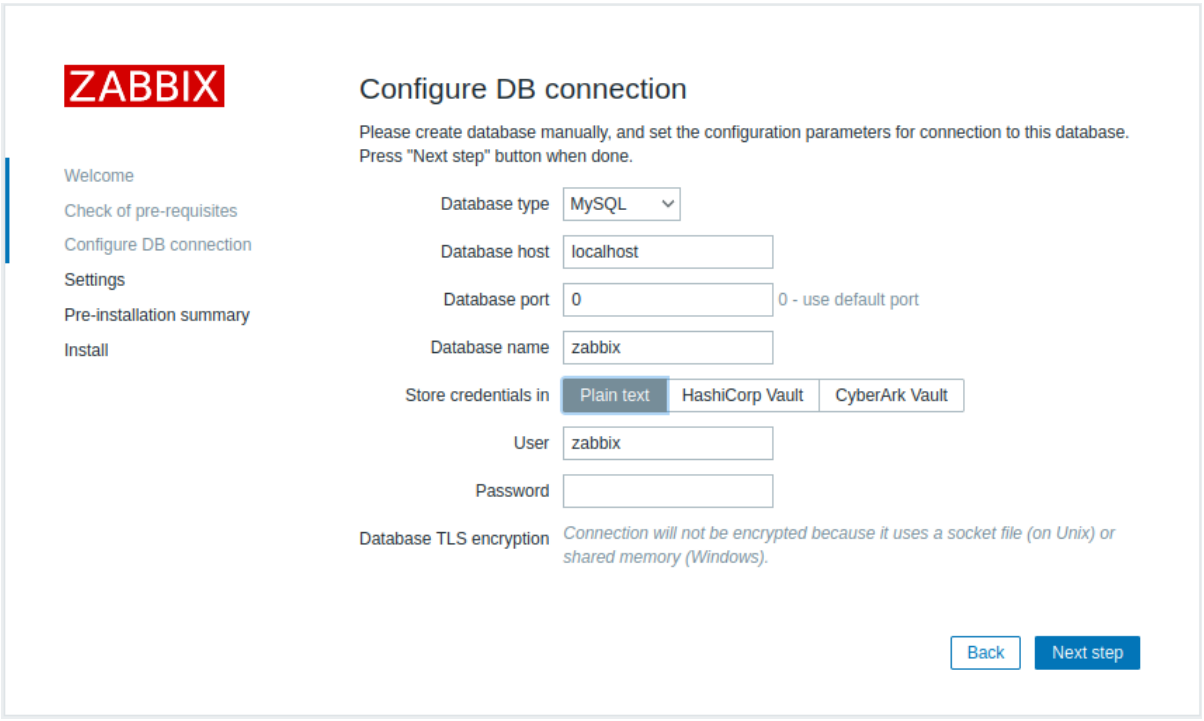
Attention:
Apache ユーザーまたはユーザーグループを変更する必要がある場合は、セッションフォルダーの権限を確認してください。そうしないと、Zabbix セットアップを続行できない可能性があります。

DB 接続の設定

データベースへの接続に必要な詳細を入力します。Zabbix データベースは事前に作成されている必要があります。

MySQL では、`localhost` を入力するか、*Database host* フィールドを空のままにすると、デフォルトの Unix ソケット経由で接続されます。セットアップフォームには個別の *Database socket* フィールドがないため、カスタムソケットを使用する場合は、Zabbix サーバーの設定で構成してください（例: zabbix_server.conf のDBSocket=）。これにより、Web インターフェースの設定がサーバーとデータベース間の接続設定と一致した状態に保たれます。

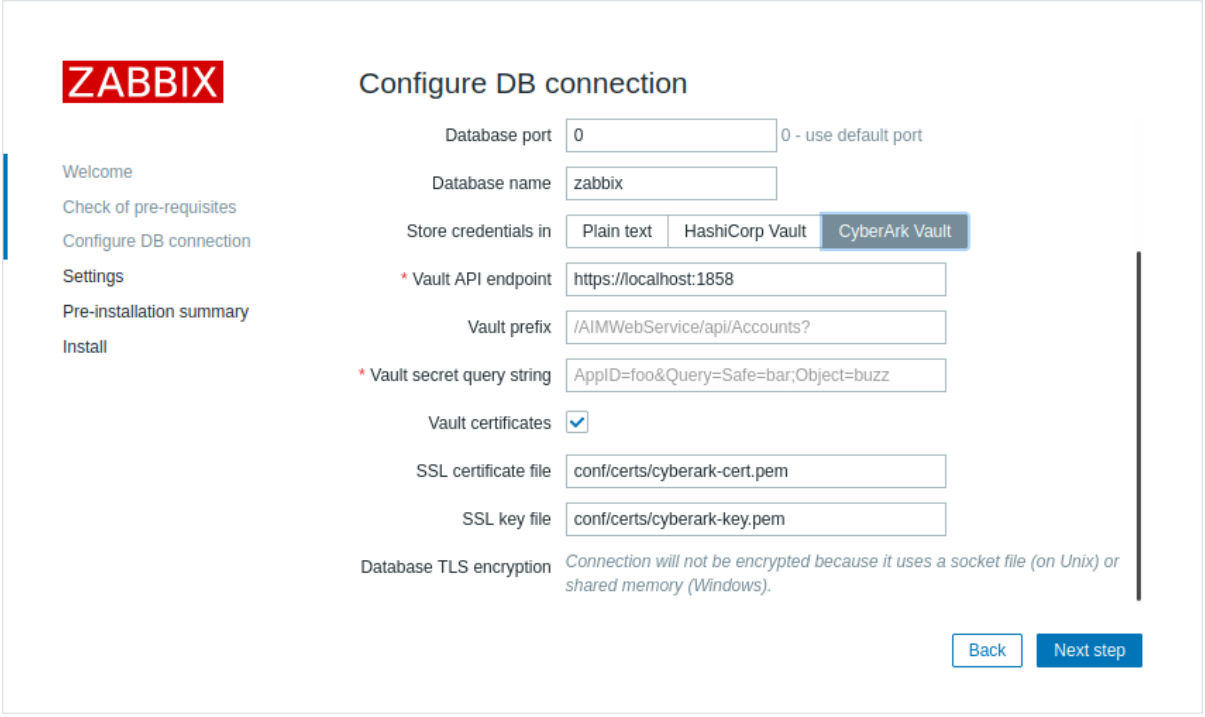
PostgreSQL では、*Database host* フィールドを空のままにすると、デフォルトの Unix ドメインソケットが使用されます。ソケットパスを入力した場合（例: `/var/run/pgbouncer`）、代わりにその Unix ドメインソケットが使用されます。



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

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

- HashiCorp Vault の場合: Vault API endpoint、vault prefix、secret path、authentication token
- CyberArk Vault の場合: Vault API endpoint、vault prefix、secret query string、certificates
  Vault certificates チェックボックスをオンにすると、SSL certificate file と SSL key file のパスを指定するための 2 つの新しいフィールドが表示されます。

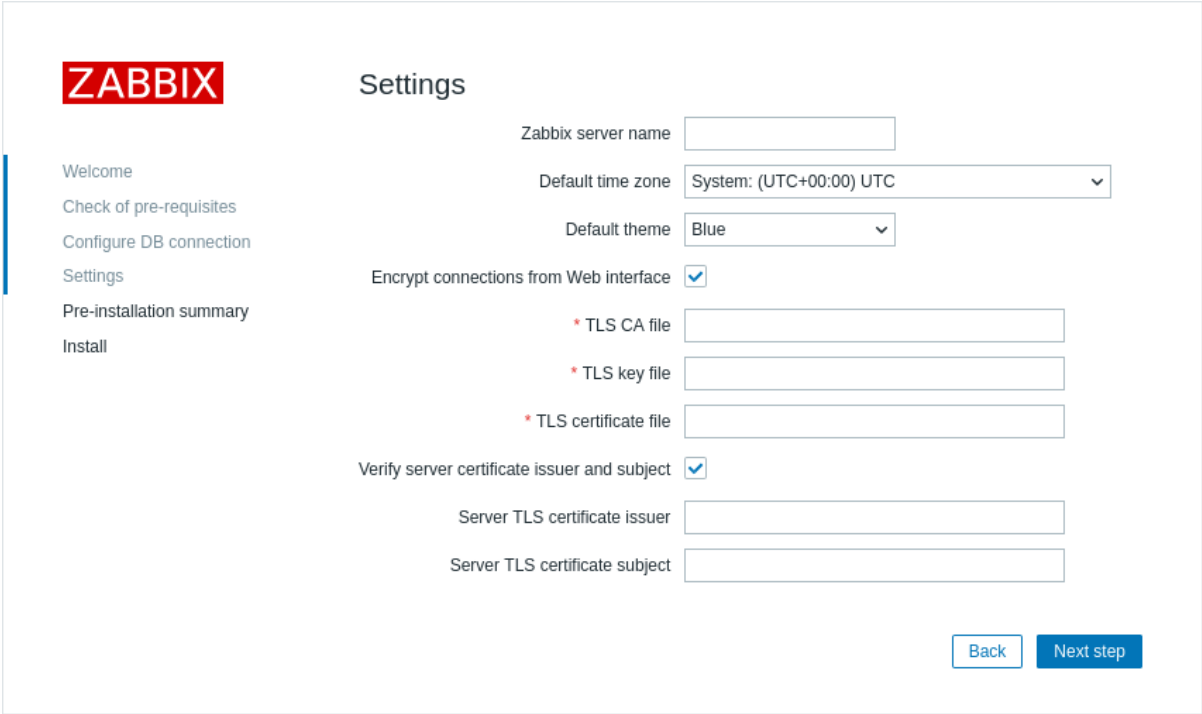


設定

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

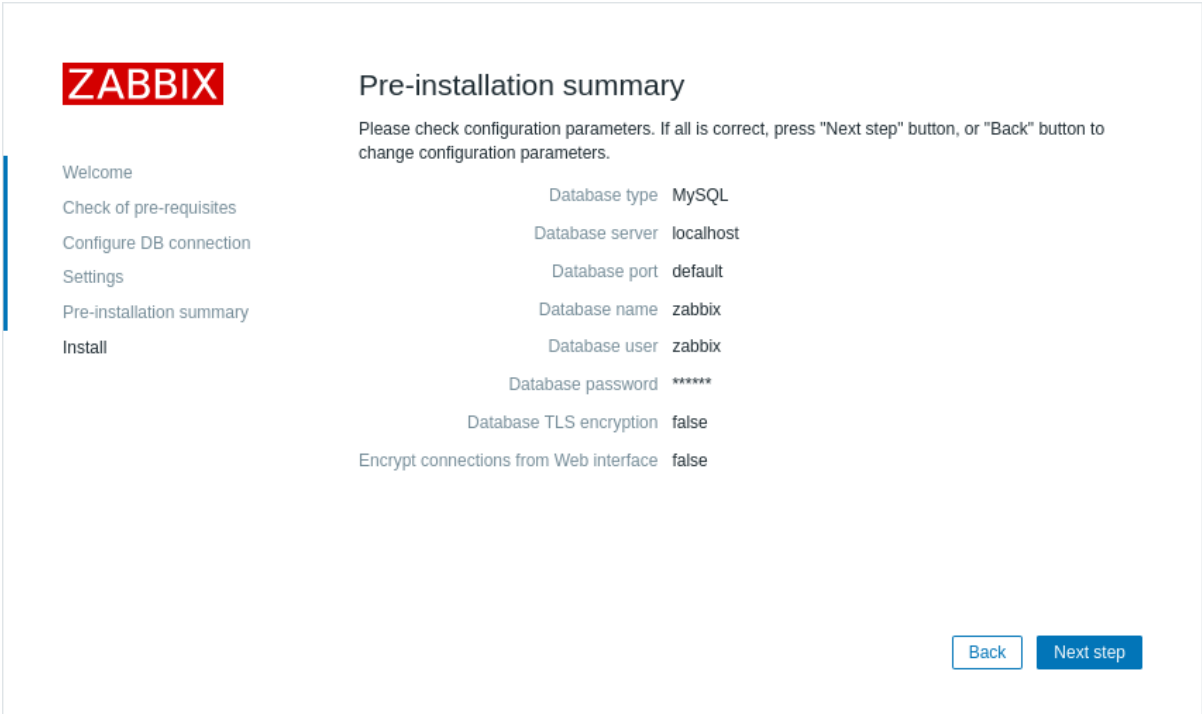
デフォルトのタイムゾーンと、Web インターフェースのテーマを設定します。

Web interface からの接続を暗号化オプションをオンにすると、Zabbix サーバーと Web インターフェース間のTLS 接続の設定のための追加フィールドがフォームに表示されます。

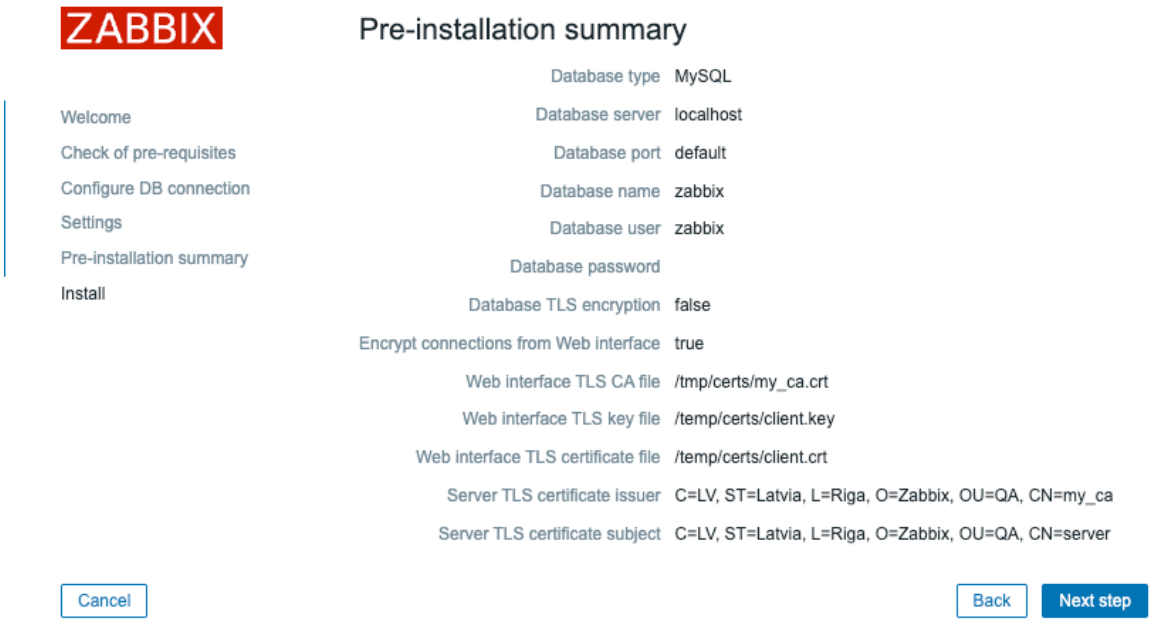


インストール前の概要

設定の概要を確認します。

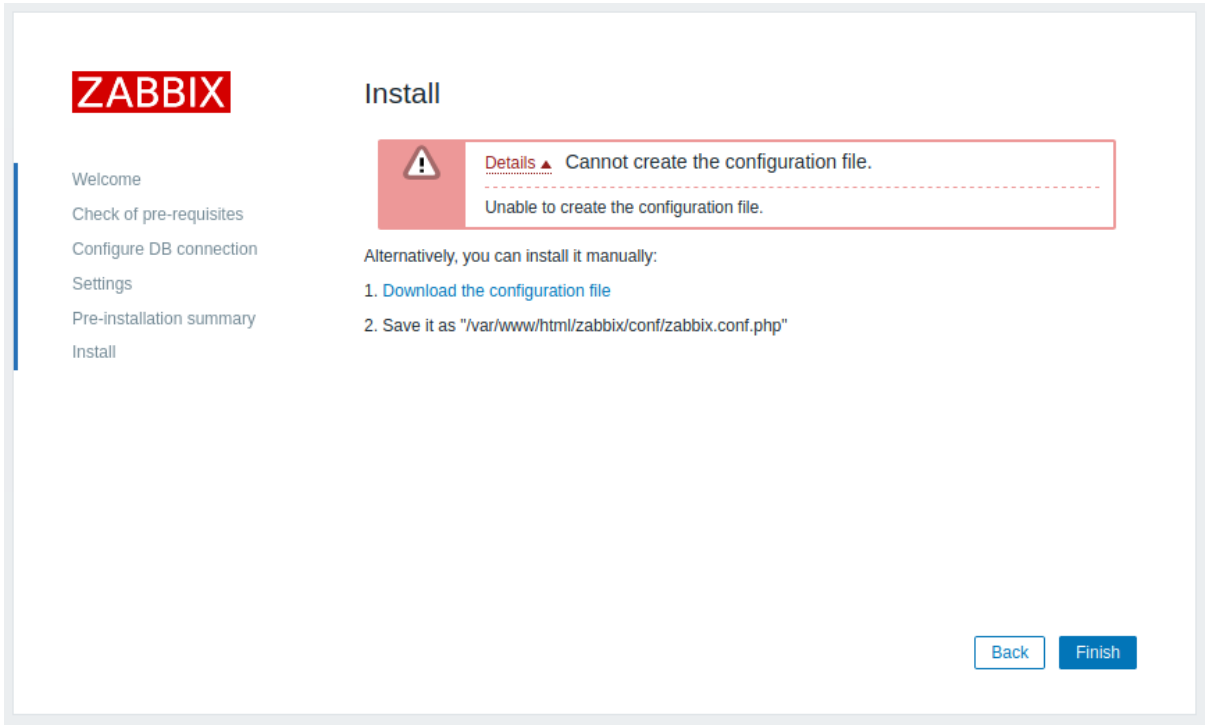


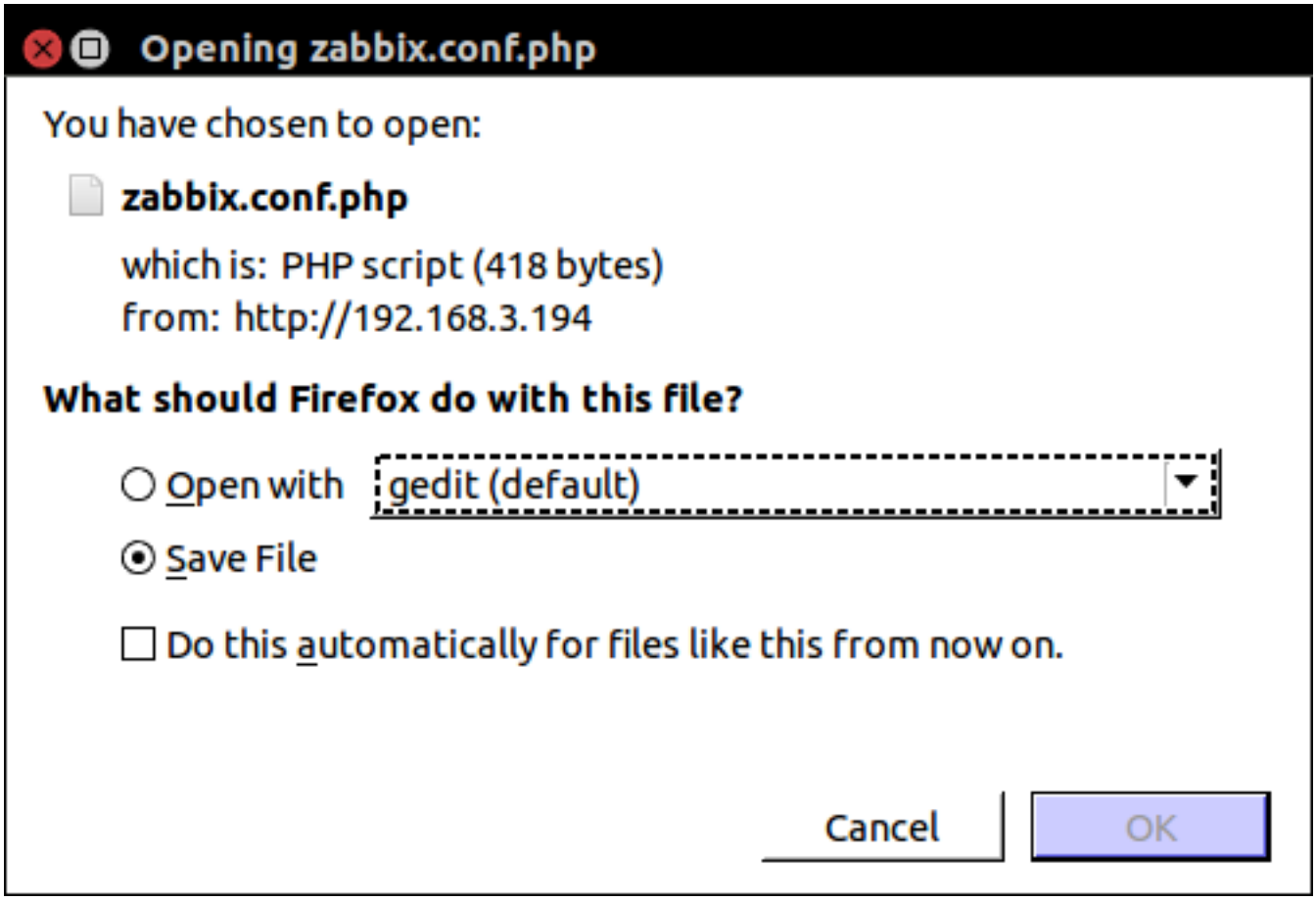
TLS 設定が追加されている場合、このサブページにデータが表示されます。



インストール

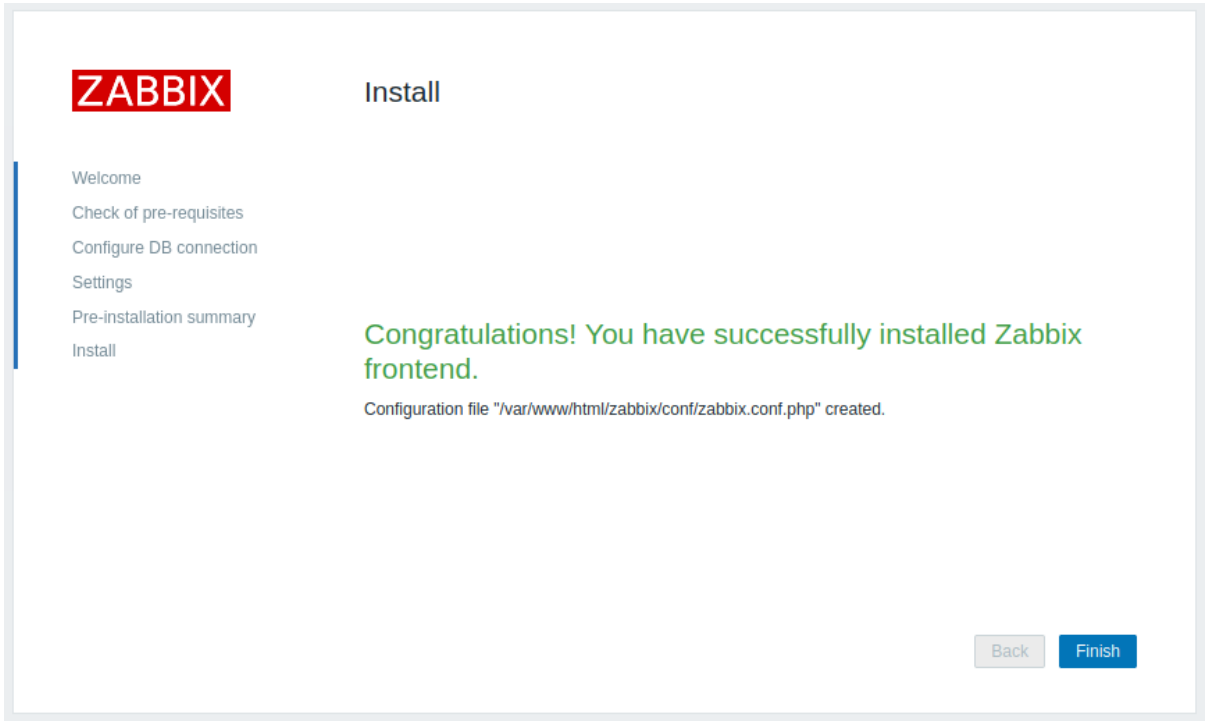
ソースから Zabbix をインストールする場合は、設定ファイルをダウンロードし、Zabbix の PHP ファイルをコピーした Web サーバーの HTML ドキュメントサブディレクトリ配下の conf/ に配置します。





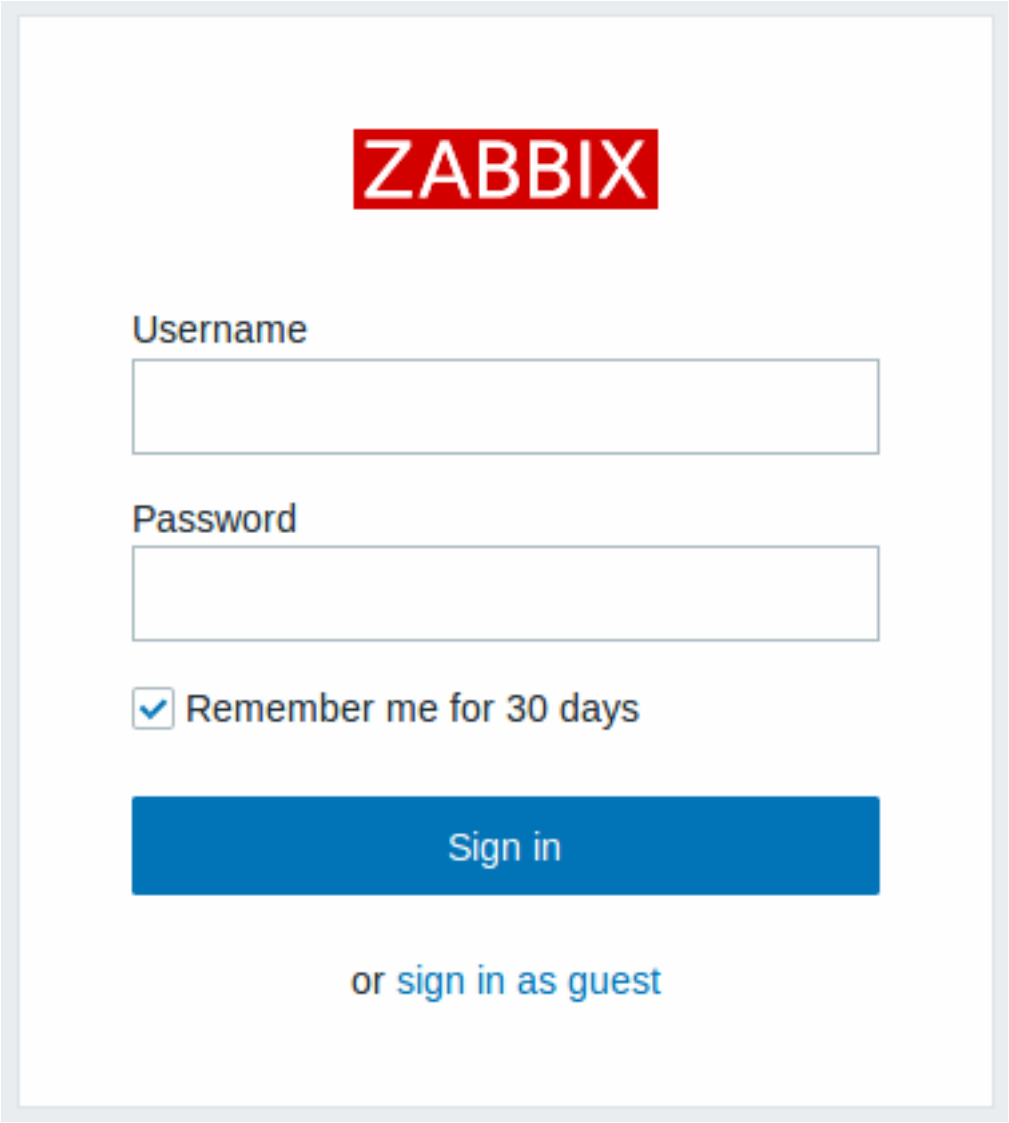
Note:
Web サーバーのユーザーに conf/ ディレクトリへの書き込み権限がある場合、設定ファイルは自動的に保存され、すぐに次の手順へ進むことができます。

インストールを完了します。



ログイン

Zabbix Web インターフェースの準備ができました。デフォルトのユーザー名は **Admin**、パスワードは **zabbix** です。



Zabbix の利用開始 に進んでください。

7 アップグレード手順

概要

このセクションでは、Zabbix **7.4** のアップグレード情報を提供します。

- Red Hat Enterprise Linux またはDebian/Ubuntu 向けのパッケージを使用する方法
- ソース を使用する方法

関連する手順:

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

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

Zabbix プロキシはアップグレードすることを強くお勧めします。Zabbix サーバーは、サーバーと同じメジャーバージョンのプロキシを完全にサポートします。Zabbix サーバーは、一つ前の LTS リリースバージョンより古いプロキシもサポートしますが、機能が制限されています (データ収集、リモートコマンドの実行、即時のアイテム値のチェック)。設定の更新も無効になり、旧式のプロキシは古い設定でのみ機能します。

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

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

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

Zabbix プロキシが SQLite3 を使用しており、起動時に既存のデータベースファイルのバージョンが必要なバージョンより古いことを検出した場合、データベースファイルは自動的に削除され、新しいファイルが作成されることに注意してください。したがって、SQLite データベースファイルに保存されていた履歴データは失われます。Zabbix プロキシのバージョンがデータベースファイルのバージョンより古い場合、Zabbix はエラーをログに記録して終了します。

データベースのサイズによっては、バージョン 7.4 へのデータベースアップグレードに長時間かかる場合があります。

Zabbix **7.2**.x、**7.0**.x、**6.4**.x、**6.2**.x、**6.0**.x、**5.4**.x、
5.2.x、**5.0**.x、**4.4**.x、**4.2**.x、**4.0**.x、**3.4**.x、**3.2**.x、
3.0.x、**2.4**.x、**2.2**.x、および **2.0**.x からは、Zabbix 7.4.x への直接アップグレードが可能です。
それ以前のバージョンからアップグレードする場合は、Zabbix 2.0 およびそれ以前のドキュメントを参照してください。

Note:
アップグレード後、アップグレードした Zabbix バージョンに外部ソフトウェアが対応していない場合、Zabbix の一部のサードパーティ製ソフトウェア連携に影響が生じる可能性があることにご注意ください。

以下のアップグレードに関する注意事項を利用できます。

アップグレード元	完全なアップグレード注意事項を読む	バージョン間の最も重要な変更点
7.2.x	対象: Zabbix 7.4	PCRE ライブラリのサポートが廃止されました。
7.0.x LTS	対象: Zabbix 7.2 Zabbix 7.4	Oracle DB のサポートが廃止されました。
6.4.x	対象: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小 PHP バージョンが 7.4.0 から 8.0.0 に引き上げられました。 エージェント、HTTP エージェント、および SNMP walk[oid] チェック用の非同期ポーラー。 プロキシ用の個別のデータベーステーブル。 Windows エージェント設定ファイルのデフォルトの場所が変更されました。 Oracle DB は非推奨になりました。 古い数値（float）値タイプが廃止されました。
6.2.x	対象: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小 MySQL バージョンが 8.0.0 から 8.0.30 に引き上げられました。 Zabbix サーバー/プロキシには'libevent_pthreads' ライブラリが必要です。 アップグレード後の初回起動時に、SQLite3 を使用する Zabbix プロキシは古いバージョンのデータベース（すべての履歴を含む）を自動的に削除し、新しいデータベースを作成します。
6.0.x LTS	対象: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小 PHP バージョンが 7.2.5 から 7.4.0 に引き上げられました。 サービス監視が大幅に再設計されました。 決定論的トリガーはアップグレード中に作成する必要があります。 MySQL/MariaDB でバイナリログが有効な場合、これにはスーパーユーザー権限、または変数/設定パラメータ *log_bin_trust_function_creators = 1* の設定が必要です。変数の設定方法については、データベース作成スクリプトを参照してください。

アップグレード元	完全なアップグレード注意事項を読む	バージョン間の最も重要な変更点
5.4.x	対象: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小データベースバージョンが引き上げられました。 データベースが古い場合、サーバー/プロキシは起動しません。 データベース構造の変更により監査ログレコードが失われます。
5.2.x	対象: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小データベースバージョンが引き上げられました。 集計アイテムは独立したタイプとして削除されました。
5.0.x LTS	対象: Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小 PHP バージョンが 7.2.0 から 7.2.5 に引き上げられました。 パスワードのハッシュアルゴリズムが MD5 から bcrypt に変更されました。
4.4.x	対象: Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	IBM DB2 のサポートが廃止されました。 必要な最小 PHP バージョンが 5.4.0 から 7.2.0 に引き上げられました。 必要な最小データベースバージョンが引き上げられました。 Zabbix PHP ファイルディレクトリが変更されました。
4.2.x	対象: Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	Jabber および Ez Texting メディアタイプが削除されました。
4.0.x LTS	対象: Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	古いプロキシは、アップグレードされたサーバーにデータを報告できなくなりました。 新しいエージェントは、古い Zabbix サーバーでは動作できなくなります。

アップグレード元	完全なアップグレード注意事項を読む	バージョン間の最も重要な変更点
3.4.x	対象: Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	'libpthread' および'zlib' ライブラリが必須になりました。 プレーンテキストプロトコルのサポートが廃止され、ヘッダーが必須になりました。 1.4 より前のバージョンの Zabbix エージェントはサポートされなくなりました。 パッシブプロキシ設定の Server パラメータが必須になりました。
3.2.x	対象: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	Zabbix サーバー/Web インターフェースのバックエンドデータベースとしての SQLite サポートが廃止されました。 POSIX 拡張の代わりに Perl Compatible Regular Expressions (PCRE) がサポートされます。 Zabbix サーバーには'libpcre' および'libevent' ライブラリが必須です。 ユーザーパラメータ、リモートコマンド、および'nowait' フラグのない system.run[] アイテム、ならびに Zabbix サーバーで実行されるスクリプトに対して、終了コードチェックが追加されました。 新機能をサポートするために、Zabbix Java ゲートウェイをアップグレードする必要があります。
3.0.x LTS	対象: Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	履歴テーブルのサイズによっては、データベースのアップグレードに時間がかかる場合があります。
2.4.x	対象: Zabbix 3.0 Zabbix 3.2 Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4	必要な最小 PHP バージョンが 5.3.0 から 5.4.0 に引き上げられました。 LogFile エージェントパラメータを指定する必要があります。

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

1 ソースからのアップグレード

概要

このセクションでは、公式の Zabbix ソースを使用して、Zabbix **7.2**.x から Zabbix **7.4**.x へ正常にアップグレードするために必要な手順を説明します。

Warning:
アップグレードの前に、必ず関連するアップグレードに関する注意事項を確認してください。

7.4 の要件も確認しておくことをお勧めします。

Note:
アップグレード中は、2 つの SSH セッションを並行して実行すると便利な場合があります。1 つのセッションでアップグレード手順を実行し、もう 1 つのセッションでサーバー/プロキシのログを監視します。たとえば、2 つ目の SSH セッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログファイルエントリや発生した可能性のあるエラーをリアルタイムで確認できます。これは本番環境のインスタンスでは特に重要になる場合があります。

サーバーのアップグレードプロセス

1 Zabbix プロセスの停止

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

2 既存 Zabbix データベースのバックアップ

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

3 設定ファイル、PHP ファイル、Zabbix バイナリのバックアップ

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

4 新しいサーバーバイナリのインストール

手順を使用して、ソースから Zabbix サーバーをコンパイルします。

5 サーバー設定パラメータの確認

設定パラメータの変更が必要かどうかを確認するため、アップグレードに関する注意事項を必ず確認してください。

新しいオプションパラメータについては、新機能ページを参照してください。

6 新しい Zabbix バイナリの開始

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

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

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

サーバーを起動する前に：

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

7 新しい Web インターフェースのインストール

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

8 Web ブラウザのクッキーとキャッシュのクリア

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

プロキシのアップグレードプロセス

1 プロキシのを停止

Zabbix プロキシを停止します。

2 設定ファイルと Zabbix プロキシバイナリのバックアップ

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

3 新しいプロキシバイナリのインストール

手順を参照して、ソースから Zabbix プロキシをコンパイルします。

4 プロキシ設定パラメータの確認

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

新しいオプションパラメータについては、新機能ページを参照してください。

5 新しい Zabbix プロキシの起動

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

Zabbix プロキシはデータベースを自動的にアップグレードします。データベースのアップグレードは、Zabbix サーバーを起動するときと同様に行われます。

エージェントのアップグレードプロセス

Attention:
エージェントのアップグレードは必須ではありません。新しい機能にアクセスする必要がある場合にのみ、エージェントをアップグレードする必要があります。

このセクションで説明するアップグレード手順は、Zabbix エージェントと Zabbix エージェント 2 の両方をアップグレードするために使用できます。

1 エージェントの停止

Zabbix エージェントを停止します。

2 設定ファイルと Zabbix エージェントバイナリのバックアップ

Zabbix エージェントバイナリと設定ファイルのバックアップコピーを作成します。

3 新しいエージェントバイナリのインストール

手順を参照して、ソースから Zabbix エージェントをコンパイルします。

またはZabbix ダウンロードページからコンパイル済みの Zabbix エージェントをダウンロードすることもできます。

4 エージェント設定パラメータの確認

設定パラメータの変更が必要かどうかを確認するため、アップグレードに関する注意事項を必ず確認してください。

新しいオプションパラメータについては、新機能ページを参照してください。

5 新しい Zabbix エージェントの起動

新しい Zabbix エージェントを起動し、ログファイルをチェックして、エージェントが正常に起動したかどうかを確認します。

マイナーバージョン間のアップグレード

7.4.x のマイナーバージョン間（たとえば 7.4.1 から 7.4.3 へのアップグレード）では、メジャーバージョン間のアップグレード時と同じ手順をサーバー/プロキシ/エージェントに対して実行する必要があります。
唯一の違いは、マイナーバージョン間のアップグレードではデータベースに変更が加えられないことです。

2 パッケージからのアップグレード

概要

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

- Red Hat Enterprise Linux
- Debian/Ubuntu

OS リポジトリからの Zabbix パッケージ

多くの場合、OS ディストリビューション（特に Debian ベースのディストリビューション）は独自の Zabbix パッケージを提供します。
これらのパッケージは Zabbix でサポートされていないことに注意してください。たいていの場合、これらのパッケージは古く、最新の機能とバグ修正が含まれていません。repo.zabbix.comのパッケージのみが公式にサポートされています。

OS ディストリビューションが提供するパッケージからアップグレードする場合（またはある時点でインストールした場合）、次の手順に従って公式の Zabbix パッケージに切り替えます。

1. 最初に古いパッケージをアンインストールしてください。
2. アンインストール後にファイルが残っていないか確認してください。
3. Zabbix が提供するインストール手順に従って公式パッケージをインストールします。

インストール済みのパッケージが破損する可能性があるため、直接更新しないでください。

1 Red Hat Enterprise Linux

概要

このセクションでは、Red Hat Enterprise Linux またはその派生ディストリビューションである AlmaLinux、CentOS Stream、Oracle Linux、Rocky Linux 向けの公式 Zabbix パッケージを使用して、Zabbix **7.2.x** から最新バージョンの Zabbix **7.4.x** にアップグレードする手順を説明します。

Warning:
アップグレードを開始する前に、該当するアップグレードに関する注意事項を確認し、お使いのシステムが Zabbix 7.4 の要件を満たしていることを確認してください。

Note:
アップグレード中は、SSH セッションを 2 つ並行して実行することを検討してください。1 つはアップグレード手順の実行用、もう 1 つはサーバー/プロキシのログ監視用です。
たとえば、2 つ目のセッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログエントリや発生した可能性のあるエラーをリアルタイムで確認できます。
これは本番環境のインスタンスでは特に重要になる場合があります。

Zabbix 7.4.x のマイナーバージョン間（たとえば 7.4.1 から 7.4.3 への）アップグレード手順については、マイナーバージョン間のアップグレードを参照してください。

アップグレード手順

1 Zabbix プロセスの停止

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

```
systemctl stop zabbix-server
```

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

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

2 Zabbix データベースのバックアップ

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

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

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

設定ファイルの場合は、次を実行します。

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

PHP ファイルと Zabbix バイナリの場合は、次を実行します。

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

4 リポジトリ設定パッケージの更新

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

RHEL 10 では、次を実行します。

```
rpm -Uvh https://repo.zabbix.com/zabbix/7.4/release/rhel/10/noarch/zabbix-release-latest.el10.noarch.rpm
```

RHEL 9 では、次を実行します。

```
rpm -Uvh https://repo.zabbix.com/zabbix/7.4/release/rhel/9/noarch/zabbix-release-latest.el9.noarch.rpm
```

Note:
古い RHEL バージョンまたはその派生ディストリビューションでは、上記のリンクを Zabbix repository にある正しいリンクに置き換えてください。ただし、これらのバージョン向けパッケージには、すべての Zabbix コンポーネントが含まれていない場合があります。これらのコンポーネントをパッケージからアップグレードするには、OS のアップグレードを検討してください。含まれているコンポーネントの一覧については、Zabbix packages を参照してください。

次に、dnf パッケージマネージャーのキャッシュをクリーンアップします（以前のインストールや更新時にダウンロードされたヘッダー、メタデータ、パッケージファイルを含む）。

```
dnf clean all
```

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

関連項目: RHEL でリポジトリ設定パッケージを更新する方法については、Known issues を参照してください。

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

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

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

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

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

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

6 コンポーネントの設定パラメータを確認する

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

新しいオプションパラメータについては、新機能ページを参照してください。

7 Zabbix プロセスの起動

アップグレードされた Zabbix コンポーネントを起動します。

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

8 Web ブラウザのクッキーとキャッシュのクリア

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

マイナーバージョン間のアップグレード

Zabbix 7.4.x のマイナーバージョン間（例えば、7.4.1 から 7.4.3 へ）でアップグレードすることができます。

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

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

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

2 Debian/Ubuntu

概要

このセクションでは、Debian/Ubuntu 向けの公式 Zabbix パッケージを使用して、Zabbix **7.2.x** から最新バージョンの Zabbix **7.4.x** にアップグレードする手順を説明します。

> **Warning:**
> アップグレードを開始する前に、関連するアップグレードに関する注意事項を確認し、お使いのシステムが Zabbix 7.4 の要件を満たしていることを確認してください。

Note:
アップグレード中は、2 つの SSH セッションを並行して実行することを検討してください。1 つはアップグレード手順の実行用、もう 1 つはサーバー/プロキシのログ監視用です。たとえば、2 つ目のセッションで `tail -f zabbix_server.log` または `tail -f zabbix_proxy.log` を実行すると、最新のログエントリや発生した可能性のあるエラーをリアルタイムで確認できます。これは本番環境のインスタンスでは特に重要になる場合があります。

Zabbix 7.4.x のマイナーバージョン間（たとえば 7.4.1 から 7.4.3 など）のアップグレード手順については、マイナーバージョン間のアップグレードを参照してください。

アップグレード手順

1 Zabbix プロセスの停止

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

```
systemctl stop zabbix-server
```

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

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

2 Zabbix データベースのバックアップ

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

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

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

設定ファイルの場合は、次を実行します。

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

PHP ファイルと Zabbix バイナリの場合は、次を実行します。

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

4 リポジトリ設定パッケージの更新

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

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

新しいパッケージをダウンロードする前に、アップグレード中にパッケージマネージャーが古いバージョンを再利用するのを防ぐため、作業ディレクトリから古い Zabbix パッケージを手動で削除する必要がある場合もあります（例: `rm zabbix-release_latest+debian12_all.deb`）。

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

Debian 12 では、次を実行します。

```
wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian12_all.deb
```

Note:
古い Debian バージョンでは、上記のリンクを Zabbix repository の正しいリンクに置き換えてください。ただし、これらのバージョン向けパッケージには、すべての Zabbix コンポーネントが含まれていない場合があります。これらのコンポーネントをパッケージからアップグレードするには、OS のアップグレードを検討してください。含まれるコンポーネントの一覧は、Zabbix packages を参照してください。

Ubuntu 24.04 では、次を実行します。

```
wget https://repo.zabbix.com/zabbix/7.4/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
```

Ubuntu 22.04 では、次を実行します。

```
wget https://repo.zabbix.com/zabbix/7.4/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu22.04_all.deb
```

> **Note:**
> 古い Ubuntu バージョンでは、上記のリンクを Zabbix repository の正しいリンクに置き換えてください。ただし、これらのバージョン向けパッケージには、すべての Zabbix コンポーネントが含まれていない場合があります。これらのコンポーネントをパッケージからアップグレードするには、OS のアップグレードを検討してください。含まれるコンポーネントの一覧は、Zabbix packages を参照してください。

Zabbix リポジトリ設定に関するプロンプトが表示される場合があります。

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

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

次に、リポジトリ情報を更新します。

```
apt update
```

5 Zabbix コンポーネントのアップグレード

Zabbix コンポーネントをアップグレードするには、次を実行します。

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

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

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

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

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

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

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

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

6 コンポーネントの設定パラメータを確認する

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

新しいオプションパラメータについては、新機能ページを参照してください。

7 Zabbix プロセスの起動

アップグレードされた Zabbix コンポーネントを起動します。

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

8 Web ブラウザのクッキーとキャッシュのクリア

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

マイナーバージョン間のアップグレード

Zabbix 7.4.x のマイナーバージョン間（例えば 7.4.1 から 7.4.3）でアップグレードできます。

まず、リポジトリ情報を更新します:

```
apt update
```

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

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

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

3 コンテナからのアップグレード

概要

このセクションでは、Zabbix **7.4**.x コンテナへの正常なアップグレードに必要な手順について説明します。

個々の Zabbix コンポーネントイメージおよび Docker compose filesをアップグレードするための個別の手順が用意されています。

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

Attention:
アップグレードを開始する前に、アップグレード作業のために、ユーザーがデータベースに対する必要な権限を持っていることを確認してください。

Zabbix 6.0 以前からアップグレードする場合、アップグレード中に決定論的トリガーを作成する必要があります。MySQL/MariaDB でバイナリログが有効になっている場合、これにはスーパーユーザー権限、または変数/設定パラメータ *log_bin_trust_function_creators = 1* の設定が必要です。この変数の設定方法については、Database creation scriptsを参照してください。
コンソールから実行した場合、この変数は一時的にのみ設定され、Docker が再起動されると失われることに注意してください。この場合は、SQL サービスは実行したままにし、'docker compose down zabbix-server' を実行して zabbix-server サービスのみを停止し、その後'docker compose up -d zabbix-server' を実行してください。
または、この変数を設定ファイルで設定することもできます。

データベースのサイズによっては、バージョン 7.4 へのアップグレードにかなり長い時間がかかる場合があります。

Zabbix イメージのアップグレード

以下の手順を使用して、任意の Zabbix コンポーネントをアップグレードできます。`zabbix-server-mysql` は、必要なコンポーネントのイメージ名に置き換えてください。

1. 現在のイメージバージョンを確認します。

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

2. 必要なイメージバージョンを取得します。例:

```
docker pull zabbix/zabbix-server-mysql:alpine-7.4-latest
```

zabbix/zabbix-server-mysql:alpine-7.4-latest は、Alpine Linux ベースで MySQL をサポートする Zabbix サーバー 7.4 の最新リリース済みマイナーバージョンを取得します。必要な Docker リポジトリ名とタグの組み合わせに置き換えてください。利用可能なオプションの一覧については、コンテナからのインストール を参照してください。

3. コンテナを停止します。

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

4. コンテナを削除します。

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

5. docker run コマンドを実行し、必要な環境変数および/またはマウントポイントを指定する追加引数を続けて、更新済みコンテナを起動します。

設定例

MySQL を使用する Zabbix サーバー:

```
docker run --name zabbix-server-mysql -t \
    -e DB_SERVER_HOST="mysql-server" \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-7.4-latest
```

PostgreSQL を使用する Zabbix サーバー:

```
docker run --name zabbix-server-pgsql -t \
    -e DB_SERVER_HOST="postgres-server" \
    -e POSTGRES_USER="zabbix" \
    -e POSTGRES_PASSWORD="zabbix_pwd" \
    -e POSTGRES_DB="zabbix" \
    -e ZBX_ENABLE_SNMP_TRAPS="true" \
    --network=zabbix-net \
    -p 10051:10051 \
    --volumes-from zabbix-snmptraps \
    --restart unless-stopped \
    -d zabbix/zabbix-server-pgsql:alpine-7.4-latest
```

MySQL サーバー:

```
docker run --name mysql-server -t \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    --network=zabbix-net \
    --restart unless-stopped \
    -d mysql:8.0-oracle \
    --character-set-server=utf8mb4--collation-server=utf8mb4_bin \
```

他の Zabbix コンポーネントの例を含む、さらに多くの設定例は、コンテナからのインストール ページで確認できます。

6. 更新を確認します。

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

Compose ファイル

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

1. 現在のイメージバージョンを確認します。

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

2. GitHub のrepository から最新の更新を取得し、必要なブランチに切り替えます。

```
git pull
git checkout 7.4
```

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

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

4. 更新を確認します。

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

サポートされている環境変数やボリュームのマウントポイントの一覧を含む詳細については、コンテナからのインストール を参照してください。

8 既知の問題

参照: コンパイルの問題

7.4.4 の既知の問題

以下の理由により、このバージョンへのアップグレードは推奨されません。

- Zabbix エージェント 2 の MySQL プラグインを使用している場合、CPU 使用率が突然急上昇する問題があるため（ZBX-27156 を参照）
- 未定義のインデックスエラーにより、Zabbix アクティブエージェントのアイテムのグラフに”Undefined array key” 警告が表示されるため（ZBX-27153 を参照）

アップグレード

アップグレードを成功させるための SQL モード設定

MySQL/MariaDB の `sql_mode` 設定には、”STRICT_TRANS_TABLES” モードが設定されている必要があります。これが設定されていない場合、Zabbix データベースのアップグレードは失敗します (参照: ZBX-19435)。

MariaDB 10.2.1 以前でのアップグレード

データベーステーブルが MariaDB 10.2.1 以前で作成された場合、Zabbix のアップグレードが失敗することがあります。これは、これらのバージョンではデフォルトの行形式がコンパクトであるためです。行形式をダイナミックに変更することで、この問題を修正できます (参照: ZBX-17690)。

テンプレート

デュアルスタック（IPv4/IPv6）環境におけるテンプレートの互換性

デュアルスタック環境（IPv4 と IPv6 の両方をサポートするよう設定されたシステム）では、ホスト名 `localhost` は通常、IPv4 と IPv6 の両方のアドレスに解決されます。
多くのオペレーティングシステムや DNS リゾルバーでは IPv4 よりも IPv6 が優先されることが一般的であるため、監視対象のサービスが IPv4 のみで待ち受けるよう設定されている場合、Zabbix テンプレートが正しく動作しないことがあります。

IPv6 アドレスで待ち受けるよう設定されていないサービスにはアクセスできなくなる可能性があり、その結果、監視に失敗します。ユーザーが IPv4 用のアクセスを正しく設定していても、IPv6 がデフォルトで優先される動作により、接続の問題が発生する場合があります。

この問題の回避策として、サービス（Nginx、Apache、PostgreSQL など）が IPv4 と IPv6 の両方のアドレスで待ち受けるよう設定されており、Zabbix サーバー/エージェントから IPv6 経由でのアクセスが許可されていることを確認してください。
さらに、Zabbix のテンプレートおよび設定では、IPv4 と IPv6 の両方との互換性を確保するため、`127.0.0.1` ではなく `localhost` を明示的に使用してください。

たとえば、PostgreSQL by Zabbix agent 2 テンプレートで PostgreSQL を監視する場合、`zbx_monitor` ユーザーの接続を許可するために `pg_hba.conf` ファイルを編集する必要があることがあります。
デュアルスタック環境で IPv6 が優先される場合（システムが localhost を `::1` に解決する場合）、`localhost` を設定していても IPv4 のエントリ（`127.0.0.1/32`）しか追加していないと、一致する IPv6 エントリが存在しないため接続は失敗します。

以下の `pg_hba.conf` ファイルの例では、`zbx_monitor` ユーザーがローカルマシンから任意のデータベースへ、IPv4 と IPv6 の両方のアドレスを使用して、異なる認証方式で接続できるようにしています。

```
### TYPE     DATABASE     USER            ADDRESS          METHOD
  host     all          zbx_monitor     localhost        trust
  host     all          zbx_monitor     127.0.0.1/32     md5
  host     all          zbx_monitor     ::1/128          scram-sha-256
```

必要に応じて、接続文字列用の PostgreSQL by Zabbix agent 2 テンプレートマクロを設定する際に、IPv4 アドレス（127.0.0.1）を直接使用することもできます。

EPEL Zabbix パッケージの誤インストール

EPEL リポジトリがインストールされ、有効になっている場合、Zabbix パッケージのインストール時に、公式の Zabbix パッケージではなく EPEL 版が取得されることがあります。これを解決するには、次の手順を実行してください。

1. EPEL からインストールされた Zabbix パッケージを削除します。

```
dnf remove zabbix-server-mysql
```

2. /etc/yum.repos.d/epel.repo ファイルに次の行を追加して、EPEL から Zabbix パッケージを除外します。

```
[epel]
...
excludepkgs=zabbix*
```

3. 公式の Zabbix サーバーパッケージを再インストールします。

```
dnf install zabbix-server-mysql
```

インストール時､公式の Zabbix パッケージにはバージョン文字列に release という語が含まれています(例: 7.0.0-release1.el8)。これにより、EPEL パッケージと区別できます。

Red Hat UBI 環境の RHEL 用 Zabbix パッケージ

Red Hat Universal Base Image環境で Red Hat Enterprise Linux パッケージから Zabbix をインストールする場合は、必要なリポジトリと依存関係にアクセスできることを確認してください。Zabbix パッケージは libOpenIPMI.so および libOpenIPMIposix.so ライブラリに依存していますが、これらは UBI システムで有効になっているデフォルトのパッケージマネージャーリポジトリのどのパッケージでも提供されていないため、インストールが失敗します。

libOpenIPMI.so および libOpenIPMIposix.so ライブラリは、redhat-#-for-<arch>-appstream-rpms リポジトリによって提供される OpenIPMI-libs パッケージで使用できます。このリポジトリへのアクセスはサブスクリプションによって管理され UBI 環境の場合、RHEL ホストのリポジトリ構成およびシークレットディレクトリをコンテナーファイルシステムの名前空間にマウントすることによって伝播されます。

詳細についてはZBX-24291を参照してください。

RHEL パッケージの署名キーの有効期限切れ

Red Hat Enterprise Linux またはその派生ディストリビューションで Zabbix をアップグレードする際に、Zabbix repository のパッケージで署名キーの有効期限切れの問題が発生する場合があります。署名キーの有効期限が切れると、パッケージ署名を検証しようとした際に、証明書またはキーがすでに有効ではないことを示すエラーが表示されます。例:

```
error: Verifying a signature using certificate D9AA84C2B617479C6E4FCF4D19F2475308EFA7DD (Zabbix LLC (Jul 2
  1. Certificate 19F2475308EFA7DD invalid: certificate is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
  2. Key 19F2475308EFA7DD invalid: key is not alive
      because: The primary key is not live
      because: Expired on 2024-07-04T11:41:23Z
```

このような問題を解決するには、使用している RHEL のバリアントに対応する最新の zabbix-release パッケージを手動で再インストールしてください（以下のリンクは、Zabbix repository にある正しいリンクに置き換えてください)。

たとえば、**RHEL 9** では次を実行します。

```
rpm -Uvh https://repo.zabbix.com/zabbix/7.4/release/rhel/9/noarch/zabbix-release-latest.el9.noarch.rpm
```

その後、リポジトリ情報を更新します。

```
dnf update
```

詳細については、ZBX-24761 を参照してください。

Timescale DB: 大量のパーティションがある場合の高メモリ使用量

PostgreSQL バージョン 9.6-12 では、多数のパーティションを持つテーブルを更新する際に、メモリを過剰に使用します。
この問題は、trends が比較的小さい（例: 1 日）チャンクに分割されている場合に、TimescaleDB を使用するシステムで Zabbix が trends を更新すると発生します。
その結果、デフォルトのハウスキーピング設定では trends テーブル内に数百のチャンクが存在することになり、PostgreSQL がメモリ不足に陥る可能性が高い状態になります。

この問題は、TimescaleDB を使用する新規インストールについては Zabbix 5.0.1 以降で解決されていますが、それ以前に Zabbix で TimescaleDB を設定していた場合は、移行に関する注意事項について ZBX-16347 を参照してください。

Timescale DB 2.5.0: 整数を含むテーブルで圧縮ポリシーが失敗する可能性

この問題は、TimescaleDB 2.5.0/2.5.1 を使用している場合に発生します。
この問題は TimescaleDB 2.5.2 以降で解決されています。

詳細については、TimescaleDB Issue #3773 を参照してください。

MariaDB でのデータベース TLS 接続

MariaDB が使用されている場合、DBTLSConnectパラメーターの'verify_ca' オプションではデータベース TLS 接続はサポートされません。

MySQL/MariaDB でのデッドロックの可能性

負荷が高く、複数の LLD 処理が実行されている場合、行ロック処理に関する InnoDB エラーが原因でデッドロックが発生する可能性があります (上流のバグ参照))。このエラーは、MySQL 8.0.29 で修正されましたが、MariaDB では修正されていません。詳細は、ZBX-21506を参照してください。

グローバルイベント相関

最初のイベントと 2 番目のイベントの時間間隔が非常に小さい場合 (0.5 秒以下)、イベントは正しく相関しないことがあります。

PostgreSQL 11 以前における数値データ型 (float) の範囲

PostgreSQL 11 以前のバージョンでは、浮動小数点値の範囲は -1.34E-154 から 1.34E+154 程度までしかサポートされていません。

NetBSD 8.0 以降

NetBSD バージョン 8.X および 9.X では、起動時に Zabbix のプロセスがランダムにクラッシュする場合があります。これは、デフォルトのスタックサイズ (4MB) が小さすぎることが原因です。以下のコマンドを実行してスタックサイズを増やす必要があります。

```
ulimit -s 10240
```

詳細については、関連する問題レポートZBX-18275を参照してください。

Zabbix エージェント 2 の正規表現の制限

Zabbix エージェント 2 は、標準の Go 正規表現ライブラリの制限により、正規表現の先読みと後読みをサポートしていません。

IPMI チェック

Debian 9 (stretch) 以前、および Ubuntu 16.04 (xenial) 以前のバージョンでは、標準の OpenIPMI ライブラリパッケージでは IPMI チェックが動作しません。この問題を修正するには、ZBX-6139の説明に従って OpenSSL を有効にし、OpenIPMI ライブラリを再コンパイルしてください。

IPMI — 信頼されていないホストにより OpenIPMI がクラッシュする可能性があります

Zabbix が IPMI データのポーリングに使用する OpenIPMI ライブラリには、信頼されていないデバイスからの特別に細工された応答によって引き起こされる不具合があります。
信頼されていない IPMI デバイスは、OpenIPMI ライブラリをクラッシュさせるように細工されたデータを送信する可能性があり、その結果、IPMI ポーリングを実行する Zabbix サーバーのプロセスが終了することがあります。

SSH チェック

- libssh2 ライブラリがパッケージからインストールされている場合、Debian、Ubuntu などの一部の Linux ディストリビューションは、暗号化された秘密鍵 (パスフレーズ付き) をサポートしません。詳細については、ZBX-4850を参照してください。
- OpenSSH 8 を使用していくつかの Linux ディストリビューションで LIBSSH 0.9.x を使用する場合、SSH チェックは「SSH Server からデータを読み取ることができない」と報告する場合があります。これは、libssh の問題 (詳細レポート) が原因です。このエラーは安定版の libssh 0.9.5 リリースで修正される予定です。詳細については、ZBX-17756を参照してください。
- パイプ"|" を使用した SSH スクリプトで「SSH サーバーからデータを読み取れません」というエラーが発生する可能性があります。この場合 libssh ライブラリのバージョンをアップグレードすることをお勧めします。詳細については、ZBX-21337を参照してください。

ODBC チェック

- MySQL unixODBC ドライバーは、MariaDB コネクタライブラリに対してコンパイルされた Zabbix サーバーまたは Zabbix プロキシと共に使用しないでください。アップストリームのバグ のため、可能であればドライバーと同じコネクタを使用しないこともお勧めします。推奨セットアップ:

  PostgreSQL, SQLite または Oracle connector → MariaDB または MySQL unixODBC driver MariaDB connector → MariaDB unixODBC driver MySQL connector → MySQL unixODBC driver

詳細および利用可能な回避策についてはZBX-7665 を参照してください。

- Microsoft SQL Server からクエリされた XML データは、Linux および UNIX システムでさまざまな方法で切り捨てられる場合があります。
- さまざまなバージョンの Linux 用 Oracle Instant Client を使用して Oracle データベースを監視するために ODBC チェックを使用すると、Zabbix サーバーがクラッシュすることが確認されています。[ZBX-18402](https://support.zabbix.com/browse/ZBX-18402)ZBX-20803も参照してください。
- FreeTDS UnixODBC ドライバーを使用している場合は、'SET NOCOUNT ON' ステートメントを SQL クエリの先頭に追加する必要があります (例:`'SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = .... '`)。そうしないと、Zabbix のデータベースモニタアイテムは"SQL クエリが空の結果を返しました" というエラーで情報の取得に失敗します。\詳細についてはZBX-19917 を参照してください。

アイテム内のリクエストメソッドパラメーターが正しくない

HTTP チェックでのみ使用されるリクエストメソッドパラメーターが、4.0 以前の Zabbix からアップグレードした結果、すべてのアイテムにデフォルト値ではない'1' が設定されることがあります。この問題の解決方法の詳細については、ZBX-19308を参照してください。

Web 監視と HTTP エージェント

いくつかの Linux ディストリビューション上で、Web シナリオや HTTP エージェント内で「SSL ピア検証」を有効にしていた時にupstream bugによってメモリリークが発生します。ZBX-10486を参照してより詳細な情報と回避策を確認してください。

シンプルチェック

v3.10 より前のバージョンの **fping** には、重複するエコーリプレイパケットを適切に処理できないバグがあります。これにより、`icmpping`、`icmppingloss`、`icmppingsec` の各アイテムで予期しない結果が発生する可能性があります。最新バージョンの **fping** を使用することをお勧めします。詳細については、ZBX-11726を参照してください。

rootless コンテナでの fping 実行に関するエラー

コンテナが rootless モードまたは特定の制限のある環境で実行されている状態で ICMP チェックを実行する際に、`fping: Operation not permitted` やすべてのリソースへのすべてのパケットが失われるなど、fping 実行に関連するエラーが発生する可能性があります。

この問題を解決するには、"docker run" または"podman run" コマンドに`--cap-add=net_raw` を追加します。

さらに、非ルート環境で fping を実行するには、sysctl の変更が必要になる場合があります。例:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

この"1995" は zabbix GID です。詳細については、ZBX-22833を参照してください。

SNMP チェック

OpenBSD オペレーティングシステムをご利用の場合、バージョン 5.7.3 までの Net-SNMP ライブラリに存在する UAF(Use After Free) バグにより、Zabbix サーバー設定ファイルで SourceIP パラメーターが設定されていると Zabbix サーバーがクラッシュする可能性があります。回避策として、SourceIP パラメーターを設定しないでください。Linux でも同じ問題が発生しますが、Zabbix サーバーが動作を停止することはありません。OpenBSD の net-snmp パッケージにはローカルパッチが適用されており、OpenBSD 6.3 でリリースされる予定です。

SNMP データのスパイク

SNMP データにスパイクが発生していることが確認されています。主電源の電圧スパイクなどの特定の物理的要因に関連している可能性があります。詳細については、ZBX-14318を参照してください。

SNMP トラップ

SNMP トラップの処理で必要な"net-snmp-perl" パッケージは、RHEL 8.0-8.2 で削除され、RHEL 8.3 で追加されました。

RHEL 8.0-8.2 を使用している場合、RHEL 8.3 にアップグレードするのが最適な解決策です。

ZBX-17192を参照してください。

RHEL 7 での alerter プロセスのクラッシュ

RHEL 7 上で Zabbix サーバーの alerter プロセスがクラッシュすることが確認されています。詳細はZBX-10461を参照してください。

Zabbix エージェント 2 (6.0.5 以前) のアップグレード

Zabbix エージェント 2 (バージョン 6.0.5 以前) をパッケージからアップグレードすると、プラグイン関連のファイル競合エラーが発生する場合があります。エラーを修正するには、(必要に応じて) エージェント 2 の構成をバックアップし、エージェント 2 をアンインストールして、新たにインストールします。

RHEL ベースのシステムでは、次を実行します:

dnf remove zabbix-agent2 dnf install zabbix-agent2

Debian ベースのシステムでは、次のコマンドを実行します:

apt remove zabbix-agent2 apt install zabbix-agent2

詳細については、ZBX-23250を参照してください。

Web インターフェースの言語設定の切り替え

Web インターフェースの言語設定が、明確なロジックなしに切り替わる場合があることが確認されています。例えば、一部のページ (またはページの一部) がある言語で表示され、他のページ (またはページの一部) が別の言語で表示されることがあります。この問題は、複数のユーザーがいて、一部のユーザーがあるロケールを使用し、他のユーザーが別のロケールを使用している場合に発生する可能性があります。

この問題の既知の回避策として、PHP と Apache のマルチスレッドを無効にすることが挙げられます。

この問題は、PHP におけるロケール設定の仕組みに関係しています。ロケール情報はスレッドごとではなくプロセスごとに保持されます。そのため、マルチスレッド環境で、同じ Apache プロセスで複数のプロジェクトが実行されている場合、別のスレッドでロケールが変更され、Zabbix スレッドでのデータ処理方法が変更される可能性があります。

詳細については、関連する問題レポートをご覧ください:

- ZBX-10911 (Web インターフェースのロケール切り替えに関する問題)
- ZBX-16297 (BC Math 関数の `bcdiv` 関数を使用したグラフ内の数値処理に関する問題)

グラフ

グラフ（クラシック）の問題

クラシックグラフで問題が発生した場合は、GD ライブラリ（libgd）をバージョン 2.3.3-13 以降に、PHP をバージョン 8.0.19、8.1.33、8.2.29、8.3.25、8.4.12 以降に更新することを推奨します。

サマータイム

サマータイム（DST）の変更により、X 軸ラベルの表示に不規則性（日付の重複、日付の欠落など）が生じます。

合計集計

グラフで 1 時間未満の期間に対して合計集計を使用すると、データがトレンドから取得される場合に、グラフに誤った（乗算された）値が表示されます。

テキストの重なり

一部の Web インターフェース言語（例: 日本語）では、ローカルフォントが原因でグラフの凡例内のテキストが重なることがあります。これを回避するには、PHP GD 拡張モジュールのバージョン 2.3.0 以降を使用してください。

ログファイルの監視

アイテム `log[]` および `logrt[]` は、ファイルシステムが 100% 使用されていてログファイルに追加される場合、ログファイルを先頭から繰り返し再読み込みします (詳細については、ZBX-10884を参照してください)。

MySQL のスロークエリ

Zabbix サーバーはアイテムの存在しない値がある場合、スロー SELECT クエリを生成します。この問題は、MySQL 5.6/5.7 バージョンで発生することが知られています (拡張された議論については、ZBX-10652を参照してください)。そして特定の場合、後の MySQL バージョンでも発生する場合があります。これに対する回避策は、MySQL の`index_condition_pushdown`または`firep_ordering_index`オプティマイザーを無効にすることです。ただし、この回避策は、スロークエリに関連するすべての問題を修正しない場合があることに注意してください。

リンクからの永続的なフィルター設定

フィルター設定（期間セレクターを含む）を含む Zabbix Web インターフェースページへのリンクを開くと、そのフィルターはユーザー用としてデータベースに自動的に保存され、そのページで以前に保存されていたフィルターおよび/または期間セレクターの設定を置き換えます。
これらの設定は、ユーザーが手動で更新またはリセットするまで有効なままです。

SNMPv3 トラップにおける IPv6 アドレスの問題

net-snmp のバグにより、SNMP トラップで SNMPv3 を使用すると、IPv6 アドレスが正しく表示されない場合があります。詳細と回避策については、ZBX-14541を参照してください。

ログイン失敗情報の長い IPv6 IP アドレスが切り詰められる

ログイン失敗メッセージには、データベースフィールドの文字数制限である最初の 39 文字のみが表示されます。そのため、39 文字を超える IPv6 IP アドレスは不完全な形で表示されます。

Windows での Zabbix エージェントのチェック

Zabbix エージェント設定ファイル (zabbix_agentd.conf) の `Server` パラメーターに存在しない DNS エントリがあると、Windows 上で Zabbix エージェントの応答時間が長くなる可能性があります。これは、Windows DNS キャッシュデーモンが IPv4 アドレスの否定応答をキャッシュしないために発生します。ただし、IPv6 アドレスの場合は否定応答がキャッシュされるため、回避策としてホストで IPv4 を無効にすることが考えられます。

YAML エクスポート/インポート

YAML のエクスポート/インポートには、いくつかの既知の問題があります。

- エラーメッセージが翻訳できない。
- .yaml ファイルの拡張子を持つ有効な JSON がインポートできないことがある。
- 引用符で囲まれていない日付は、自動的に Unix タイムスタンプに変換される。

SUSE での NGINX および php-fpm 使用時のセットアップウィザード

SUSE で NGINX + php-fpm を使用している場合、フロントエンドのセットアップウィザードは設定ファイルを保存できません。これは、/usr/lib/systemd/system/php-fpm.service ユニットの設定により、Zabbix が/etc に書き込みできないことが原因です (PHP 7.4で導入されました)。

回避策は 2 つあります。

- php-fpm systemd ユニットのProtectSystemオプションを'full' ではなく'true' に設定する。
- /etc/zabbix/web/zabbix.conf.php ファイルを手動で保存する。

認証ヘッダーの転送

Apache または NGINX が、API リクエスト内の認証ヘッダーを Zabbix に転送できない場合があります。これにより、Zabbix API や Okta を使用した SAML といったシングルサインオン (SSO) サービスの使用時に認証の問題が発生する可能性があります。

この問題を解決するには、Web サーバーの設定を更新してください。

Apache をリバースプロキシ (CGI 以外の設定) として使用する場合は、(RHEL ベースシステムの場合) `/etc/httpd/conf/httpd.conf` または (Debian/Ubuntu の場合) `/etc/apache2/apache2.conf` に次のディレクティブを追加してください。

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Apache がリクエストを処理するためにスクリプトを直接実行する場合 (例: mod_cgi を使用)、代わりに次のディレクティブを追加してください。

```
CGIPassAuth On
```

一方、**NGINX** は Authorization ヘッダーを自動的に処理します。ただし、NGINX がリバースプロキシとして動作している場合は、`/etc/nginx/nginx.conf` (Zabbix フロントエンドの場所) に以下のディレクティブを追加することで、Authorization ヘッダーを明示的に転送できます。

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

設定を更新したら、Web サーバーを再起動してください。

詳細については、以下を参照してください:

- ZBX-22952
- Apache 2.4 + PHP-FPM および Authorization ヘッダー
- SetEnvIfNoCaseおよびCGIPassAuthディレクティブ
- NGINX リバースプロキシ

Ubuntu 20 の Zabbix Web サービスの Chromium

ほとんどの場合、Zabbix Web サービスは Chromium で動作しますが、Ubuntu 20.04 で Chromium を使用すると、以下のエラーが発生します:

```
Cannot fetch data: chrome failed to start:cmd_run.go:994:
WARNING: cannot create user data directory: cannot create
"/var/lib/zabbix/snap/chromium/1564": mkdir /var/lib/zabbix: permission denied
Sorry, home directories outside of /home are not currently supported. See https://forum.snapcraft.io/t/112
```

このエラーは、/var/lib/zabbix がユーザー'zabbix' のホームディレクトリとして使用されるために発生します。

MySQL カスタムエラーコード

Zabbix はバックエンドデータベースにアクセスできないことを検出すると、通知を送信し、接続を試行し続けます。特定のデータベースエンジンでは固有のエラーコードが認識されます。MySQL で認識されるエラーコードは次のとおりです:

- CR_CONN_HOST_ERROR
- CR_SERVER_GONE_ERROR
- CR_CONNECTION_ERROR
- CR_SERVER_LOST
- CR_UNKNOWN_HOST
- ER_SERVER_SHUTDOWN
- ER_ACCESS_DENIED_ERROR
- ER_ILLEGAL_GRANT_FOR_TABLE
- ER_TABLEACCESS_DENIED_ERROR
- ER_UNKNOWN_ERROR

さらに、Azure 上の MySQL インストールで Zabbix を使用している場合、不明瞭なエラーメッセージ *[9002] Some errors occurred* が Zabbix ログに表示されることがあります。この一般的なエラーテキストは、データベースから Zabbix サーバーまたはプロキシに送信されます。エラーの原因について詳しくは、Azure ログを確認してください。

PCRE2 への移行後の無効な正規表現

Zabbix 6.0 では、PCRE2 のサポートが追加されました。PCRE は現在もサポートされていますが、RHEL 7 以降、SLES (全バージョン)、Debian 9 以降、Ubuntu 16.04 以降の Zabbix インストールパッケージは、PCRE2 を使用するように更新されました。PCRE2 への切り替えは多くの利点を提供する一方で、特定の既存の PCRE Regexp パターンが無効になったり、異なる動作をしたりする可能性があります。特に、パターン *^[\w-\.]* に影響します。この regexp をセマンティクスに影響することなく再度有効にするために、式を *^[-\w\.]* に変更します。これは、PCRE2 がダッシュ記号を区切り文字として扱い、文字クラス内に範囲を作成するために発生します。

Geomap ウィジェットエラー

NGINX を使用して古いバージョンの Zabbix からアップグレードし、アップグレード中に新しい NGINX 設定ファイルに切り替えなかった場合、Geomap ウィジェットの地図が正しく読み込まれないことがあります。

この問題を解決するには、古い設定ファイルを破棄し、現在のバージョンのパッケージの設定ファイルを使用し、*e. Zabbix* フロントエンドの *PHP* の設定セクションのdownload instructionsに記載されている方法で再設定することができます。

または、既存の NGINX 設定ファイル (通常、*/etc/zabbix/nginx.conf*) を手動で編集することができます。これを行うには、ファイルを開き以下のブロックを探します:

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

そして、このブロックを次のように置き換えます:

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

Webhook 呼び出し間で共有されるグローバル変数の使用例

グローバル変数 (例: `var`、`let` または `const` 宣言なしで `global = 1` を実行する) は異なる Webhook 呼び出し間で共有されるため、次のコードを実行すると、タグ値カウンターが徐々に増加します:

```
try
{
   aa = aa + 1;
```

```
}
catch(e)
{
   aa = 0;
}

result = {
        'tags': {
            'endpoint': aa
        }
    };
return JSON.stringify(result);
```

各スクリプトが独自のデータで動作し、同時呼び出し間での競合を抑止する為には、グローバル変数ではなくローカル変数を使用ください。

Windows のプロセッサグループ

Microsoft のドキュメントによると、論理プロセッサ数が 64 未満のシステムでは、プロセッサグループは常に 1 つで、Group 0 になります。ただし、Zabbix ユーザーからは、論理プロセッサ数が 64 以下のシステムで 2 つのプロセッサグループが存在するというまれなバグ ZBX-20260 が報告されています。その結果、2 つあるプロセッサグループのうち 1 つに対してしか"\Processor(n)" パフォーマンスカウンターが存在しない状態になっていました。このバグの実際の根本原因は不明です。ただし、これと似た事例が stackoverflow.com で説明されており、そこでの根本原因は BIOS と Windows の相互運用にありました。

utf8mb4 照合順序におけるフィルタリングの制限

特定の Unicode 文字（例: 𝒸、𝓈）を含むエンティティに適用した場合、フィルター（例: データ収集 > メンテナンス）が正しく機能しないことがあります。この問題は、MySQL または MariaDB データベースのデフォルトの utf8mb4_bin 照合順序における Unicode 文字の並べ替えおよび比較の処理方法に起因します。

この制限に対処するため、ユーザーはデータベース列の照合順序を、utf8mb4_0900_bin、utf8mb4_0900_ai_ci、utf8mb4_unicode_520_ci などの代替値に変更できます。ただし、照合順序を変更すると、空白の扱いや、他の文字の並べ替えおよびフィルタリングにおいて予期しない動作が発生する可能性がある点に注意してください。

照合順序の変更の詳細については、MySQL documentation または MariaDB documentation を参照してください。照合順序の違いの詳細については、MySQL documentation の Unicode Character Sets を参照してください。

MariaDB 10.5.1–10.5.9 における UI 要素へのアクセス

Super Admin 以外のロールで Zabbix Web インターフェースにアクセスすると、次のメッセージが表示される場合があります: 「System error occurred. Please contact Zabbix administrator.」。この問題は、MariaDB versions 10.5.1 から 10.5.9 を使用しているインストール環境に影響します。

この問題を回避するには、MariaDB を 10.5.9 より新しいバージョンに更新してください。詳細については、ZBX-25746 を参照してください。

tcmalloc を使用した過剰なメモリ使用量のプロファイリング

Zabbix のインストール環境でメモリを使いすぎている疑いがある場合は、tcmalloc のメモリプロファイリング機能を使用して、Zabbix サーバー/プロキシのメモリ消費を調査できます。

1. Zabbix をソースからインストールする際に、追加のフラグを設定します。

```
export CFLAGS="-std=gnu99 -g -O0"
```

-std=gnu99 フラグは、Zabbix サーバー、Zabbix プロキシ、または Zabbix エージェントのビルドに必要です。
-g フラグは追加のデバッグ情報を付加し、-O0 は最適化を無効にします。最適化は tcmalloc のプロファイリングに干渉する可能性があります。

2. Zabbix サーバーを起動する前に、以下の環境変数を設定します。
これらの変数は、メモリ使用量をどのように追跡し、レポートするかを tcmalloc に指示します。

```
LD_PRELOAD="/usr/lib/aarch64-linux-gnu/libtcmalloc.so" \
HEAPPROFILE=./heap_profile \
HEAP_PROFILE_ALLOCATION_INTERVAL=0 \
HEAP_PROFILE_INUSE_INTERVAL=4294967296 \
HEAPPROFILESIGNAL=5 \
MALLOCSTATS=1 \
./sbin/zabbix_server -f -c /etc/zabbix/zabbix_server.conf
```

3. 対象プロセスにシグナル 5 を送信して、プロファイルダンプをトリガーします。
1234 は実際のプロセス ID（PID）に置き換えてください。

```
kill -5 1234
```

4. 生成されたプロファイルを表示します。

```
pprof-symbolize -text ./sbin/zabbix_server ./heap_profile.0001.heap

Using local file ./sbin/zabbix_server.
Using local file ./heap_profile.0001.heap.
Total: 1078.1 MB
  1076.8  99.9%  99.9%   1076.8  99.9% zbx_malloc2
     1.0   0.1% 100.0%      1.0   0.1% __GI___strdup
     0.2   0.0% 100.0%      0.2   0.0% CRYPTO_zalloc@@OPENSSL_3.0.0
     0.1   0.0% 100.0%      0.1   0.0% OPENSSL_LH_insert@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% zbx_realloc2
     0.0   0.0% 100.0%      0.1   0.0% PKCS7_decrypt@@OPENSSL_3.0.0
     0.0   0.0% 100.0%      0.0   0.0% find_best_tree_node
     0.0   0.0% 100.0%      0.0   0.0% CRYPTO_strndup@@OPENSSL_3.0.0
     ...
     0.0   0.0% 100.0%      0.0   0.0% preprocessing_flush_value
     0.0   0.0% 100.0%   1074.0  99.6% preprocessor_add_request
```

この例では、zbx_malloc2 がほぼすべてのメモリ割り当ての原因となっています。

関連情報:

- 関連する問題レポートについては、ZBX-25050 および ZBX-25584 を参照してください。
- コンパイルオプション（-std=gnu99、-g、-O0 など）については、GCC Option Summary を参照してください。
- tcmalloc プロファイリング用の環境変数については、Gperftools Heap Profiler のドキュメントを参照してください。

マルチプライマリモードでの MySQL 8.0 Group Replication

MySQL 8.0 Group Replication をマルチプライマリモードで使用している場合、トランザクションのコミット中に次のようなエラーが発生することがあります。

```
1531697:20250128:064734.697 query [txnlev:1] [update alerts set status=1,retries=0,error='' where alertid=
1531697:20250128:064734.713 query [txnlev:1] [commit;]
1531697:20250128:064734.753 [Z3005] query failed: [3101] Plugin instructed the server to rollback the curr
```

このエラーは、外部キー制約を伴うロールバック処理の問題によって発生していると考えられます。

関連情報:

- 関連する問題報告については、ZBX-26060 を参照してください。
- アップストリームの問題については、MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" を参照してください。

1 コンパイルの問題

これらは、ソースからの Zabbix コンパイルに関する既知の問題です。他のすべてのケースについては、既知の問題を参照してください。

非標準の場所にあるライブラリ

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

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

これは、システムにインストールされている唯一の libcurl である場合に機能しますが、たとえば、パッケージマネージャーによって標準の場所に別の libcurl がインストールされている場合は機能しません。これは、Zabbix に新しいバージョンのライブラリと他のアプリケーションに古いバージョンが必要な場合で起きます。

したがって、非標準の場所でコンポーネントを指定すると、同じコンポーネントが標準の場所に存在する場合、常に機能するとは限りません。

たとえば、libcurl パッケージがまだインストールされている状態で/usr/local に新しくインストールされた libcurl を使用する場合、zabbix は間違ったものをピックアップしてコンパイルが失敗する可能性があります。

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

ここで、関数 curl_easy_header() は古い/usr/lib/x86_64-linux-gnu/libcurl.so では利用できませんが、新しい/usr/local/lib/libcurl.so では利用可能です。

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

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

一部のシステムでは-Wl,--no-as-needed オプションが必要かもしれないことに注意してください（参照: debian ベースシステムのデフォルトリンクオプション）。

一部のシステムでスタックサイズが小さすぎる

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

次の例では、スレッドごとのスタックサイズを 512 KB に設定しています。

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

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

2 エスケープ関連のアップグレードの問題

アップグレード後に関数パラメーターの文字数制限超過

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

7.0 より前の Zabbix バージョンからのアップグレード中にバックスラッシュが追加されることで、パラメーターが長くなり、長さが最大データサイズの 255 文字を超えてトリガー関数が壊れる可能性があります。

データベースのアップグレード中に発生する可能性のある問題、その影響、その認識方法と修正方法を説明するために、3 つのアイテムと 2 つのトリガーを含む設定例を見てみましょう。

セットアップ

1. Zabbix 6.0 バージョン。
2. 末尾に'a' があり、114 個のバックスラッシュを含む /tmp/ONE ファイル:

```
\\\
```

```
2485502:20250228:115442.236 DBpatch_6050165(): cannot save in DB function parameter: resulting size 477 is
functionid:33792 function:'find'
used on host: 'Zabbix server'
  in trigger: 'TRIG_228'.
Current parameter value:
'$,1m:now,"regexp","\\\
```

これで両方のトリガーが以前と同じように動作します - Trig_228 のみが発砲します。

計算されたアイテムには問題はありませんが、一貫性のために更新することもできます:

Calc_228:

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

Calc_232:

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

9 テンプレートの変更

このページでは、Zabbix に同梱されている標準テンプレートに対するすべての変更を一覧表示しています。

Zabbix をアップグレードしても、既存のテンプレートはカスタム変更が上書きされるのを防ぐため、自動的には更新されません。テンプレートをアップグレードする場合、または新しいテンプレートを追加する場合は、テンプレートのアップグレード を参照してください。

Note:
Zabbix 6.0 以降、すべてのテンプレートは更新された形式に従っているため、6.0 より前のテンプレートのインポートに影響する可能性があることにご注意ください。詳細については、6.0 におけるテンプレートの変更 を参照してください。

7.4.7 での変更点　更新されたテンプレート

- Ciena 3906 by SNMP は、ファイルシステムおよび CPU 負荷のアイテムを含むように更新されました。
- Vyatta Virtual Router by SNMP は、新しい OID アイテムおよびダッシュボードの機能強化を含むように更新されました。

7.4.6 での変更点　新しいテンプレート

- Ribbon SBC Edge by HTTP: HTTP 経由で Ribbon SBC Edge（旧 SWe Lite）デバイスを監視するためのテンプレート。
- Ribbon SBC SWe Core by HTTP: HTTP 経由で SBC SWe Core デバイスを監視するためのテンプレート、および Ribbon SBC SWe CE by HTTP: HTTP 経由で Ribbon SBC SWe Call Engine（CE）インスタンスを監視するためのテンプレート。
- VeloCloud SD-WAN Edge by HTTP: HTTP 経由で VeloCloud SD-WAN Edge デバイスを監視するためのテンプレート。

更新されたテンプレート

- MySQL by Zabbix agent、MySQL by Zabbix agent 2、MySQL by Zabbix agent active、MySQL by Zabbix agent 2 active、および MySQL by ODBC は、`SHOW SLAVE STATUS`（旧構文）と `SHOW REPLICA STATUS`（新構文）の両方をサポートするよう更新されました。
- 以前は *VMWare SD-WAN VeloCloud by HTTP* だった VeloCloud SD-WAN by HTTP は、最新の VeloCloud SD-WAN プラットフォームに合わせて名称が変更され、更新されました。

7.4.4 での変更点　新しいテンプレート

- Stormshield SNS by SNMP: SNMP 経由で Stormshield Network Security（SNS）デバイスを監視するためのテンプレート。

7.4.3 での変更点　新しいテンプレート

- Aruba CX 8300s by SNMP: Aruba CX 8300 スイッチシリーズの SNMP ベースの監視を提供するテンプレート。
- Ciena 3906 by SNMP: Ciena 3906 デバイスを監視するためのテンプレート。
- Vyatta Virtual Router by SNMP: Vyatta 1908e 仮想ルーターを監視するためのテンプレート。

更新されたテンプレート

- Huawei OceanStor Dorado by SNMP は、対応デバイスの詳細を追加して更新されました。
- Zabbix server health および Zabbix proxy health のテンプレートセットは、前処理スループットの監視機能を拡張して更新されました。

7.4.2 での変更点　新しいテンプレート

- AWS by HTTP テンプレートセットに、テンプレート *AWS Backup Vault by HTTP* が追加されました。

7.4.1 での変更点　新しいテンプレート

- Cisco Secure Firewall Threat Defense by HTTP：REST API を使用する Cisco Secure Firewall Threat Defense デバイスの監視機能を提供するテンプレート。

更新されたテンプレート

- VMware および VMware FQDN のテンプレートセット内のテンプレート *VMware Guest* が、新しいアイテム *Boot time* で更新されました。このアイテムは、vMotion などの仮想マシン移行後の誤検知を回避するため、トリガー *VM has been restarted* において *Uptime of guest OS* を置き換えるようになりました。

10 7.4.0 へアップグレード時の注意点

Zabbix 7.2.x から Zabbix 7.4.0 へのアップグレードに関する注意点です。

すべての注意点は以下のように分類されています。

- 重大な変更 - 既存の環境を壊す可能性のある変更、およびアップグレードプロセスに関連する重要な情報
- その他 - Zabbix の機能の変更に関する残りのすべての情報

参照:

- Zabbix 7.4.0 より前のバージョンからのアップグレードに関する情報は、アップグレード手順を参照してください。
- 高可用性 (HA) クラスター内のサーバーのアップグレード手順は、HA クラスターのアップグレードを参照してください。

アップグレードプロセス

MySQL/MariaDB で Zabbix サーバーのアップグレードを正常に完了するには、バイナリロギングが有効で、スーパーユーザー権限がなく、MySQL 構成ファイルに `log_bin_trust_function_creators = 1` が設定されていない場合、MySQL で `GLOBAL log_bin_trust_function_creators = 1` を設定する必要がある場合があります。

MySQL コンソールを使用して変数を設定するには、次を実行します。

```
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
```

アップグレードが正常に完了したら、このオプションを無効にすることができます。

```
mysql> SET GLOBAL log_bin_trust_function_creators = 0;
```

重大な変更　PCRE ライブラリの廃止

PCRE（または PCRE1）ライブラリは廃止されました。Zabbix は現在、PCRE2 でコンパイルされています。

自身のユーザーメディアの管理

現在、すべてのユーザーはデフォルトで自身のメディアを管理できます。

ユーザー自身のメディア詳細を変更する権限は、ユーザーロールに基づいて付与（または取り消し）できます（*Create and edit own media* オプションを参照）。

さらに、他のユーザーのメディア詳細を変更する Super admin ユーザーの権限についても、ユーザーロールに基づいて付与/取り消しできます（*Create and edit user media* オプションを参照）。

古いバージョンからアップグレードする場合、*Default access to new actions* が有効になっているすべてのロールで、両方のオプションが有効になります。*Default access to new actions* が有効でない場合、アップグレード後に管理者およびスーパー管理者がメディアを編集する権限を失う可能性があることに注意してください。

履歴関数パラメータにおけるバックスラッシュのエスケープ

Zabbix 7.0.0 以降、履歴関数の文字列パラメータにおいて、バックスラッシュの適切なエスケープが追加されました。

7.0 より前の Zabbix バージョンからのアップグレード時には追加のバックスラッシュが挿入されるため、パラメータが長くなり、パラメータ長が最大データサイズである 255 文字を超えると、トリガー関数が壊れる可能性があります。

この問題を回避するため、アップグレード前に長いパラメータを手動でユーザーマクロへ移動することを推奨します。

7.0 より前の Zabbix バージョンからのアップグレードでは、変換後のパラメータ長に対する追加チェックも導入されています。変換後の長さが最大サイズを超える場合、そのようなパラメータはアップグレードされず、ログに警告が出力され、一覧にあるパラメータを手動で修正するようユーザーに求めます。

あわせてエスケープ関連のアップグレード時の問題 も参照してください。

MSSQL Zabbix エージェント 2 プラグインの更新

テンプレート MSSQL by Zabbix agent 2 は、検出されたクォーラムメンバーを名前で含めるまたは除外するためのフィルター、および空のクラスターを除外するためにクラスター名でフィルタリングできるサービスフィルターを追加して更新されました。テンプレートをエラーなく動作させるには、MSSQL Zabbix エージェント 2 プラグイン を 7.4.0 以上のバージョンに更新する必要があります。

必要な libssh2 の最小バージョン

必要な libssh2 の最小バージョンが、1.0.0 から 1.8.0 に引き上げられました。

検出されたホスト上のホストプロトタイプ

現在の設定に、他のホストプロトタイプを含むテンプレートを使用するホストプロトタイプが含まれている場合、Zabbix 7.4 へのアップグレードにより、検出されたすべてのホストに、それらのテンプレートで定義されたホストプロトタイプが設定されます。
この動作が意図したものでない場合は、アップグレード前に、該当するテンプレートからホストプロトタイプを手動で削除してください。

アップグレード中にホストプロトタイプが追加される検出済みホストを特定するには、Zabbix データベースで次の SQL クエリを実行できます。

```
SELECT h.hostid,ht.templateid
FROM hosts_templates ht
JOIN hosts h ON ht.hostid=h.hostid
WHERE h.flags=4
  AND EXISTS (
    SELECT NULL
    FROM items i,host_discovery hd
    WHERE i.hostid=ht.templateid
    AND hd.parent_itemid=i.itemid
    )
ORDER BY hostid;
```

データベース接続パラメータ

DBPort と DBSocket は、現在では相互排他的です。データベース接続パラメータを指定する場合は、DBPort または DBSocket のいずれか一方を定義するか、デフォルト値を使用するために両方とも未定義のままにすることができます。

その他　有効な OID がない未サポートの SNMP walk/ディスカバリアイテム

SNMP walk アイテムでは、OID または OID インスタンスが 1 つも存在しない場合、エラーを返し、アイテムは未サポートになります。存在しない OID/インスタンスの詳細は、DebugLevel=5 でログに記録されます。以前は、このような場合は空文字列を返していました。

同様に、SNMP discovery アイテムでも、OID または OID インスタンスが 1 つも存在しない場合、エラーを返し、アイテムは未サポートになります。以前は、このような場合は空の配列を返していました。

空文字列/配列が返されるのは、OID/インスタンスが存在していても、それに対応するデータがない場合のみになったことに注意してください。

履歴キャッシュの管理

一部のデータ収集シナリオでは、特定のアイテムが一時的にサーバー/プロキシの履歴キャッシュをブロックすることがあります。これにより、履歴データのデータベースへの書き込みが遅延し、システムの動作が遅くなる可能性があります。履歴キャッシュの管理を支援するため、以下の改善が導入されました。

- 手動でのキャッシュクリア**:** 新しい history_cache_clear=target ランタイムコマンドにより、Zabbix server/proxy で、特定のアイテムの履歴キャッシュを ID で指定して手動でクリアできるようになりました。
- 自動キャッシュクリア**:** アイテムを無効化すると、そのアイテムは直ちに履歴キャッシュから削除されます（ただし、ログ用に最後の値は保持されます）。同様に、ホストを無効化すると、そのホストのすべてのアイテムが履歴キャッシュから削除されます（ただし、それぞれの最後の値は保持されます）。
- キャッシュ診断ログ**:** 履歴キャッシュがいっぱいになると、Zabbix サーバー/プロキシはDebugLevel=3 から履歴キャッシュの診断情報をログに記録するようになりました（以前は DebugLevel=4）。このログには、履歴キャッシュ内で最も多くの値を保持しているアイテムが含まれます。

システムの長期的な安定性を確保するため、データ収集が利用可能なリソース（データベース性能、キャッシュサイズ、収集間隔、ログアイテムのパラメータなど）と適切にバランスしていることを確認してください。Zabbix の履歴キャッシュは、内部アイテムzabbix[wcache] を使用して監視できます。また、Zabbix server/proxy の履歴キャッシュサイズを増やすことも検討してください。

ユーザー通知用の独立したメニューセクション

ユーザーメディアの視認性を向上させるため、ユーザー通知は ユーザー設定の下にある独自のメニューセクションを持つようになりました。

新しい「通知」セクションには、以前はユーザープロファイルセクションの一部であった メディアと *Web* インターフェース通知の 2 つのタブが含まれています。

マップ内のネストされたホストグループの情報を修正

マップ内で、ネストされたホストグループの情報が正しく表示されるようになりました。例えば、以下のとおりです。

- ホストグループラベルに、ネストされたホストグループ内のすべてのホストの障害サマリーが表示されるようになりました。
- 「ホストグループ要素」表示では、ネストされたホストグループ内の各ホストごとに個別のマップ要素が表示されるようになりました。
- マップラベルに、ネストされたホストグループに含まれるすべての障害のサマリーが表示されるようになりました。

変換された設定用データベーステーブル

`settings` テーブルは、グローバル設定パラメータの保存において、`config` テーブルに代わって使用されるようになりました。
新しいテーブルでは、各パラメータごとに列を持つ単一行に保存するのではなく、キーと値の形式が使用されます。
これがデータ交換にどのように影響するかの例については、サーバー-プロキシ間データ交換プロトコル（`"settings"` オブジェクト）を参照してください。

サーバーおよびプロキシの最大キャッシュサイズを拡大

一時的な問題（例：設定、データベース、またはネットワークの問題）が発生した際のキャッシュ枯渇を遅らせ、動作を維持できるようにするため、最大キャッシュサイズが 2 GB から 16 GB に引き上げられました。

- サーバー: HistoryCacheSize、HistoryIndexCacheSize、およびTrendCacheSize の最大値が引き上げられました。
- プロキシ: HistoryCacheSize およびHistoryIndexCacheSize の最大値が引き上げられました。

デフォルトのユーザーセッション保存期間の短縮

ユーザーセッションデータのデフォルトの保存期間が、365 日から 31 日に短縮されました。この変更はhk_sessionsパラメータに影響し、デフォルト値は 365d ではなく 31d になりました。

サポートされる Go の最小バージョン

サポートされる Go の最小バージョンが、1.21 から 1.23 に引き上げられました。

以前に、1.23 より古い Go バージョンを使用して、ソースから Zabbix エージェント 2、エージェント 2 のロード可能プラグイン、またはWeb サービスをビルドしていた場合は、最新のセキュリティ更新とバグ修正を受け取るために、サポートされる Go バージョンを使用してこれらのコンポーネントを再ビルドすることを推奨します。古い Go バージョンでビルドされたコンポーネントも引き続き動作しますが、それらをアップグレードするには、より新しい Go 環境が必要になります。

ホストウィザード向けテンプレートのアップグレード

新しいホストウィザードでは、Zabbix で監視対象（デバイス、アプリケーション、サービスなど）を設定するための、ガイド付きの段階的なインターフェースが導入されています。
これにより、テンプレートの選択、Zabbix エージェントのインストール、ホストインターフェースの追加などの主要な手順をユーザーに案内し、新規または既存のホストの設定を簡素化します。

以前の Zabbix バージョンからアップグレードした後は、ホストウィザードで動作するようにテンプレートをアップグレードする必要があります。
手順については、テンプレートのアップグレードを参照してください。

11 7.4.x のアップグレードに関する注意事項

このページでは、Zabbix メジャーバージョンのマイナーリリースに関するアップグレード時の注意事項をまとめて提供します。

メジャーバージョンのアップグレードに関する注意事項も参照してください。

7.4.9 のアップグレードに関する注意事項　このマイナーバージョンには、アップグレードに関する注意事項はありません。

7.4.8 へのアップグレードに関する注意事項　iframe を使用して HTML として表示されるアイテム値

アイテム履歴ウィジェット内のアイテムデータは、HTML 形式のテキストとして表示される場合、iframe 内に分離されるようになりました。

更新された SAML PHP Toolkit ライブラリ

SAML PHP Toolkit ライブラリ（Zabbix に同梱）が、バージョン 4.0.0 から 4.3.1 に更新されました。

7.4.7 のアップグレードに関する注意事項　このマイナーバージョンには、アップグレードに関する注意事項はありません。

7.4.6 へのアップグレードに関する注意事項　ログファイルの権限制限

Zabbix コンポーネントのログファイルは、現在、ファイル所有者のみに読み書き権限を付与して作成されます。さらに、これらのファイルは所有グループによる読み取りも可能です。その他のすべての権限は拒否されます。

Linux では、この変更により権限は 0666 から 0640 になりました。この変更は、Zabbix サーバー、プロキシ、およびエージェントのログファイルに影響します。Zabbix エージェント 2 のログファイルの権限も、0644 から 0640 に更新されました（others の読み取り権限を削除）。

さらに、いくつかのその他のファイルも現在は 0640 の権限で作成されます。

- Zabbix プロセスの UNIX pid ファイル
- データエクスポートファイル
- `icmpping*` チェック用の一時ファイル
- ログ永続ファイル

Zabbix エージェント 2 が処理する MySQL レプリケーションステータスのクエリ

Zabbix エージェント 2 のMySQL プラグイン は、`SHOW SLAVE STATUS` と `SHOW REPLICA STATUS` の両方をサポートするようになりました。このプラグインは、サーバーのバージョンに応じて適切なステートメントを自動的に選択して実行し、通常の JSON 形式で結果を返します。

`mysql.replication.discovery` や`mysql.replication.get_slave_status` などのエージェント 2 アイテムは、`SHOW SLAVE STATUS` または `SHOW REPLICA STATUS` のいずれかを使用するサーバーで動作します。既存のエージェント 2 アイテムやエージェント 2 テンプレートを変更する必要はありません。

サポートされる Go の最小バージョン

サポートされる Go の最小バージョンが、1.23 から 1.24.10 に引き上げられました。

以前に、1.24.10 より古い Go バージョンを使用してソースから Zabbix エージェント 2、エージェント 2 のロード可能なプラグイン、またはWeb サービスをビルドしていた場合は、最新のセキュリティ更新とバグ修正を受け取るために、サポートされる Go バージョンを使用してこれらのコンポーネントを再ビルドすることを推奨します。古い Go バージョンでビルドされたコンポーネントも引き続き動作しますが、それらをアップグレードするには、より新しい Go 環境が必要になります。

Web サービス設定にファイルまたはフォルダを含める

Include パラメータが Zabbix Web サービス設定ファイルに追加されました。このパラメータを使用すると、個別のファイル、またはディレクトリ内のすべてのファイルを設定に含めることができます。

7.4.5 のアップグレードに関する注意事項　このマイナーバージョンには、アップグレードに関する注意事項はありません。

7.4.4 へアップグレード時の注意点　注意**!** グラフと Zabbix エージェント 2 の MySQL プラグインに関連するエラーのため、このバージョンへのアップグレードはお勧めできません。

7.4.3 へアップグレード時の注意点　PHP cURL の最小バージョン

PHP 拡張機能cURL の最小サポートバージョンは 7.19.4 になりました。

7.4.1 へアップグレード時の注意点　エージェント 2 ランタイム制御出力ストリーム

Zabbix エージェント 2 のランタイム制御コマンド (`zabbix_agent2 -R <option>`) は、出力を `stderr`(標準エラー) ではなく `stdout`(標準出力) に書き込むようになりました。以前の動作に依存しているスクリプトは更新してください。

5. クイックスタート

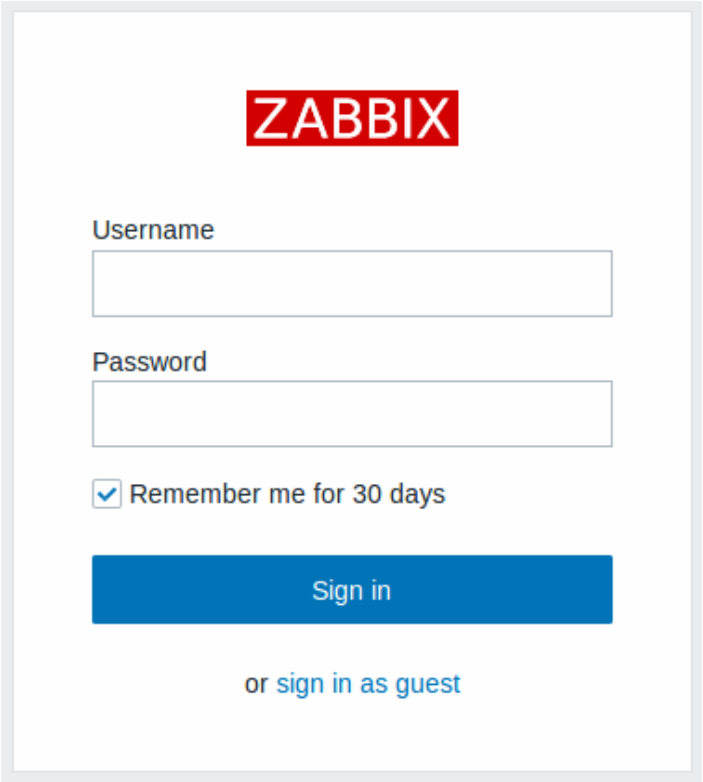
サイドバーから、参照したい項目を選択してください。

1 ログインとユーザー設定

概要

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

ログイン



これは Zabbix のようこそ画面です。Zabbix スーパーユーザーとしてログインするには、ユーザー名 **Admin** とパスワード **zabbix** を入力します。すべてのメニューセクションへのアクセスが許可されます。

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

永続ログイン

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

30 日間記憶するが有効の場合:

- セッションは 30 日間アクティブなままになります。
- 自動ログアウトが上書きされ、期間が終了するまでログイン状態が維持されます。
- 30 日以内に再度アクセスすると、認証情報を再入力することなく自動的にログインされます。

30 日間記憶するが無効の場合:

- 以前に有効にした自動ログインはすべてクリアされます。
- 設定された自動ログアウト期間に従ってセッションが期限切れになります。

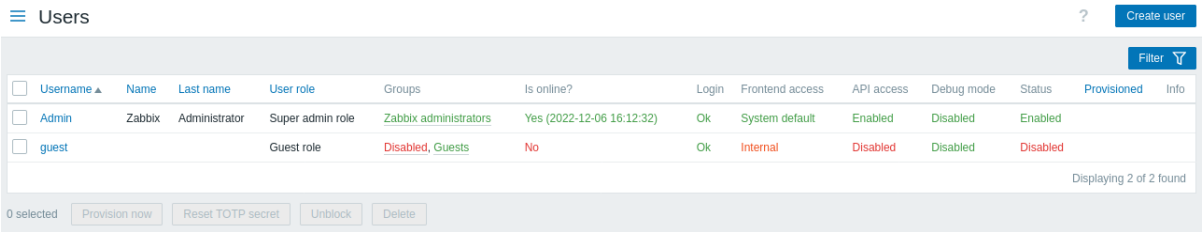
ブルートフォース攻撃からの保護

ログイン試行が 5 回連続して失敗した場合、Zabbix インターフェースはブルートフォース攻撃と辞書攻撃を防ぐために 30 秒間一時停止します。

失敗したログイン試行の IP アドレスは、ログイン成功時に表示されます。

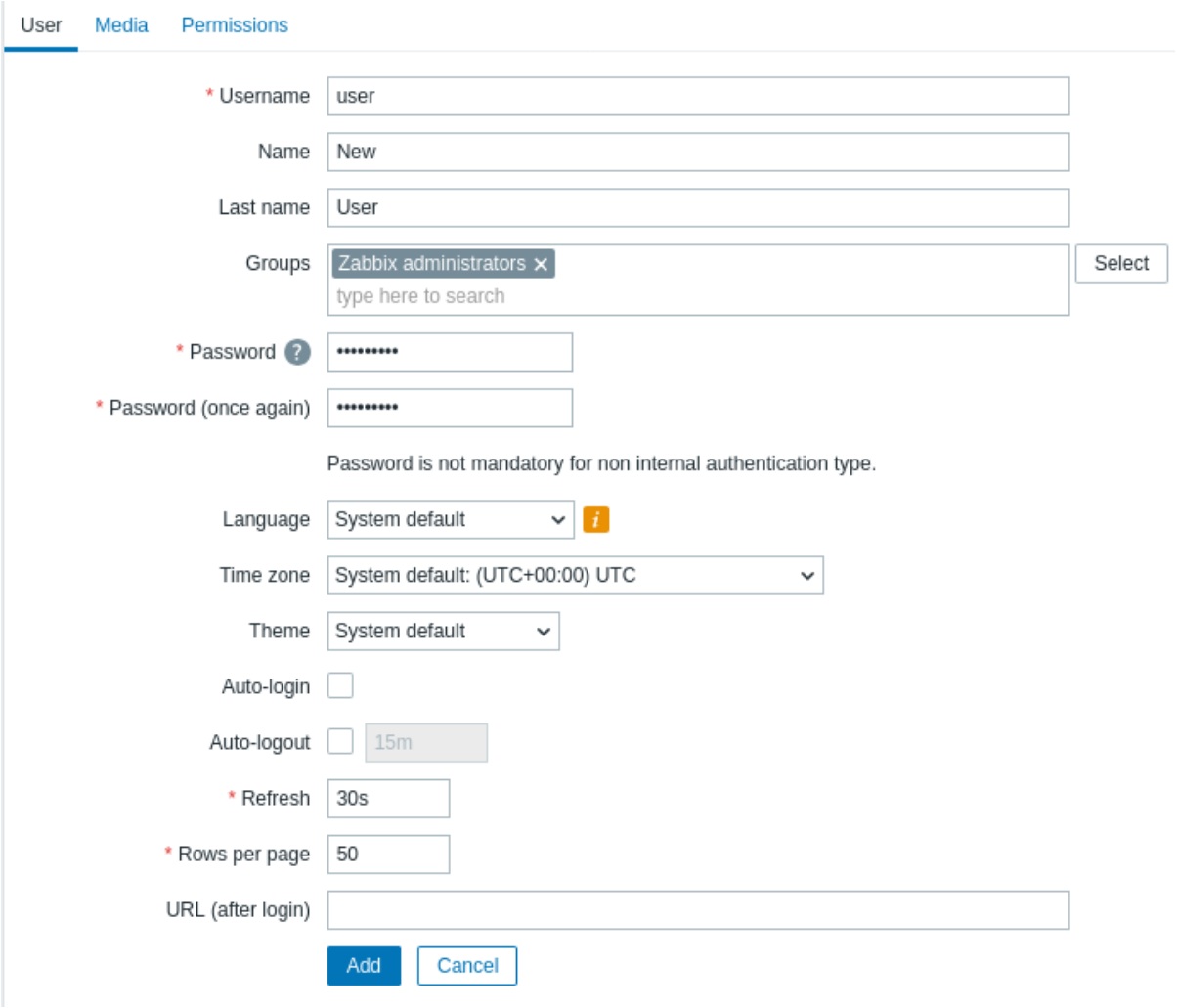
ユーザーの追加

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



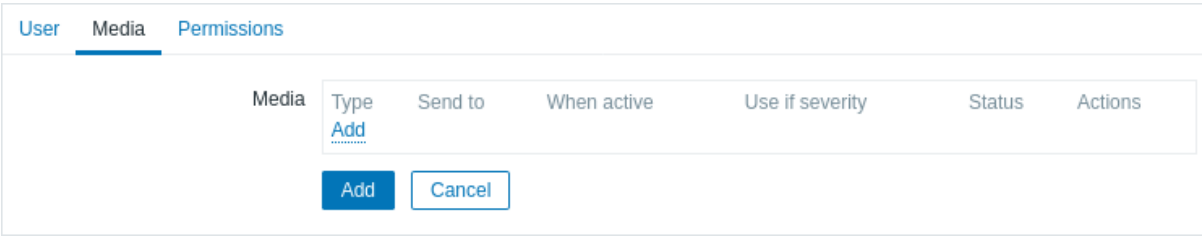
新しいユーザーを追加するには、右上隅の *Create user* を選択します。

新規ユーザーフォームでは、既存のユーザーグループ のいずれか（例: 'Zabbix administrators'）にユーザーを追加してください。



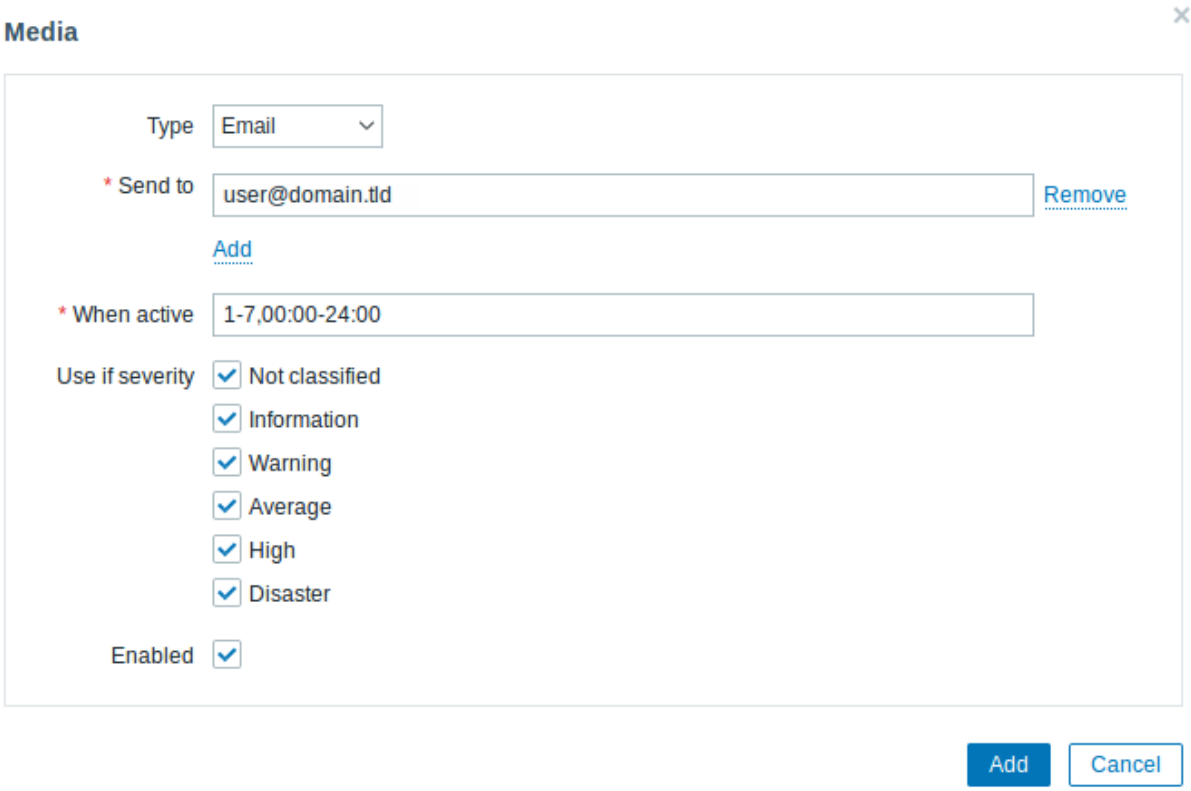
必須入力フィールドには、赤いアスタリスクが付いています。この設定フォームの入力フィールドの詳細については、ユーザー設定ページを参照してください。

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



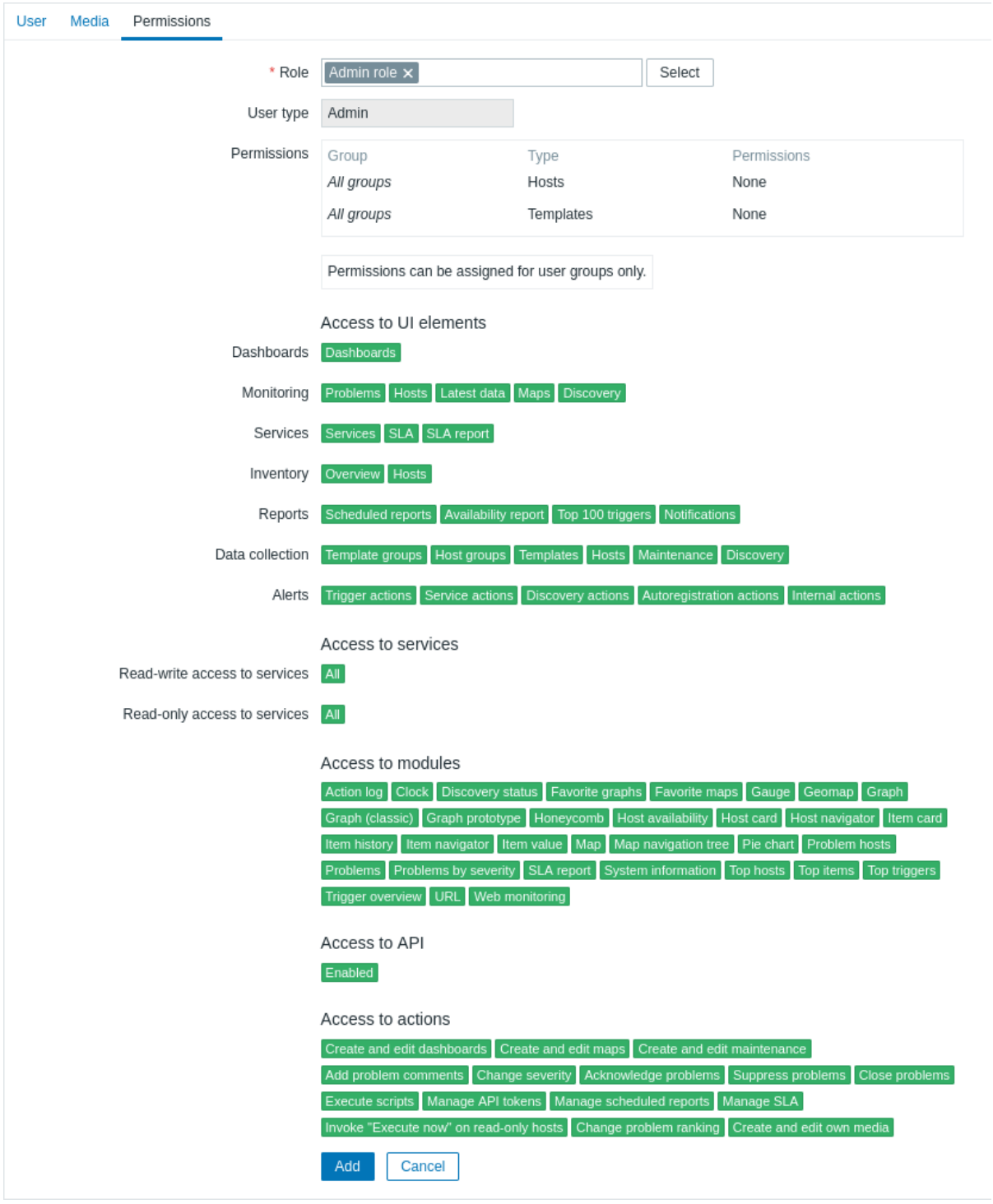
ポップアップで、ユーザーのメールアドレスを入力します。

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



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

Permissions タブには必須フィールド *Role* があります。ロールによって、ユーザーが表示できる Web インターフェース要素と、実行を許可される操作が決まります。Select を押して、一覧からいずれかのロールを選択します。例えば、*Admin role* を選択すると、Administration を除くすべての Zabbix Web インターフェースセクションへのアクセスが許可されます。後で権限を変更したり、さらに多くのユーザーロールを作成したりできます。ロールを選択すると、同じタブに権限が表示されます。

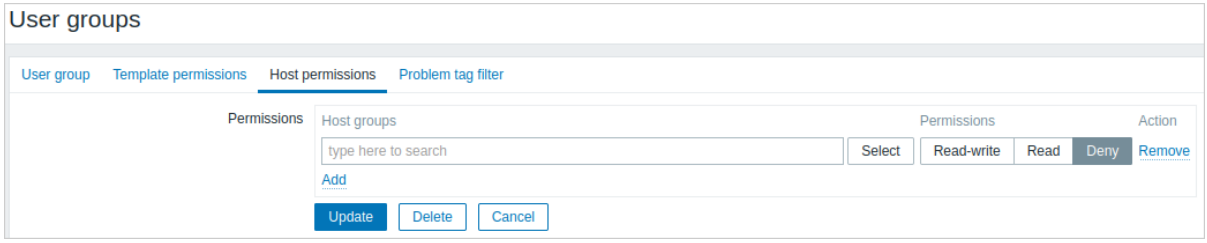


ユーザープロパティフォームで *Add* をクリックして、ユーザーを保存します。新しいユーザーがユーザー一覧に表示されます。



権限の追加

デフォルトでは、新しいユーザーにはホストとテンプレートにアクセスするための権限がありません。権限を付与するにはグループ列のグループをクリックします (この場合 - 'Zabbix administrators')。ユーザーグループプロパティフォームでホスト権限タブに移動し、ホストグループに権限を付与します。ホストグループ選択フィールドを表示するには Add をクリックして下さい。



次に、フィールドの横にある選択をクリックして、ホストグループのリストを表示します。このユーザーは *Linux servers* グループへの読み取り専用アクセス権を持っているため、リスト内の適切なチェックボックスをオンにし、選択をクリックして確定します。



読み取りボタンをクリックしてアクセス許可レベルを設定し、更新をクリックしてユーザーグループ構成に加えた変更を保存します。

テンプレートに権限を付与するには、テンプレート権限タブに切り替えて、テンプレートグループを指定する必要があります。この手順は、ホストグループに権限を割り当てる手順と同じです。テンプレートの概要は、このクイックスタートの新規テンプレートセクションで参照できます。

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

以上で新しいユーザーの資格情報を使用してログインを試みることができます。

2 新規ホスト

概要

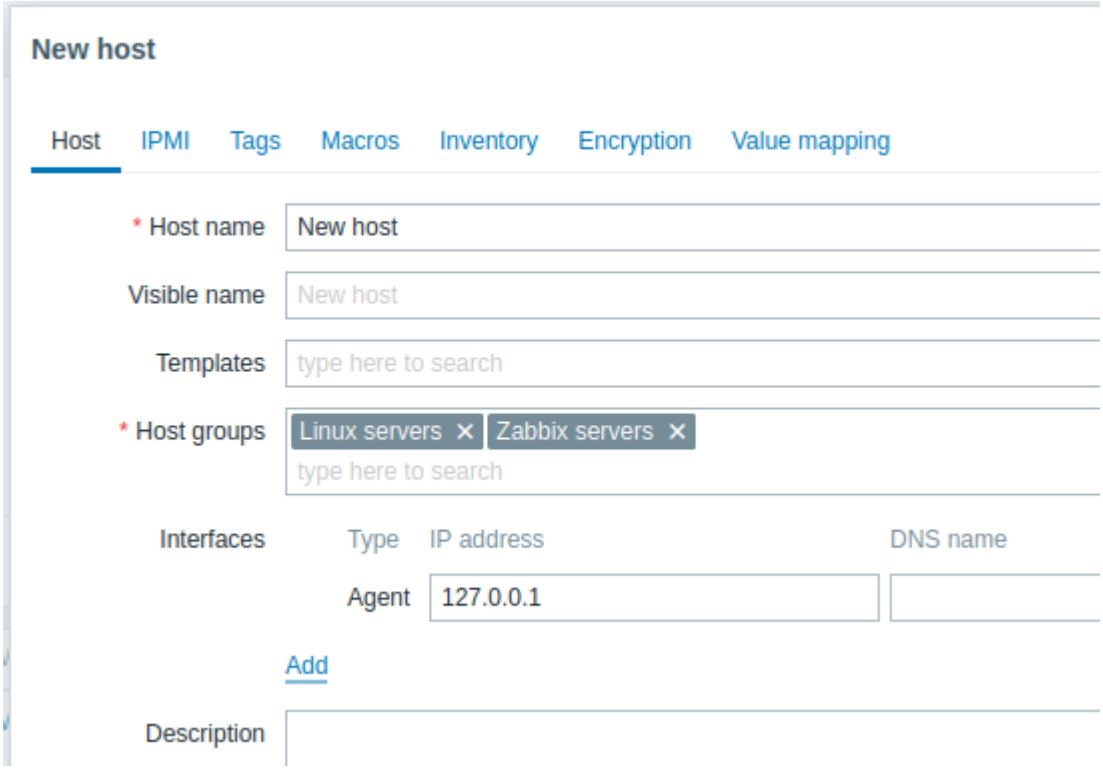
このセクションでは、新しいホストを設定する方法を学習します。

Zabbix のホストとは、監視するネットワークエンティティ (物理、仮想) です。Zabbix で” ホスト” とすることができるものは非常に柔軟です。物理サーバー、ネットワークスイッチ、仮想マシン、または一部のアプリケーションもホストとすることができます。

ホストの追加

Zabbix で設定されたホストに関する情報は、*Data collection* > *Hosts* および *Monitoring* > *Hosts* メニューセクションで確認できます。すでに「Zabbix server」という事前定義済みのホストが 1 つありますが、ここでは別のホストを追加する方法を学びます。

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



必須入力フィールドには、赤いアスタリスクが付いています。

ここで入力する最低限必要な項目は次のとおりです。

Host name

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

Host groups

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

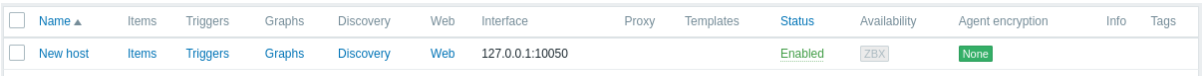
Note:
すべてのアクセス権限は個々のホストではなくホストグループに割り当てられます。そのため、ホストは少なくとも 1 つのグループに属している必要があります。

Interfaces: IP address

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

その他のオプション は、デフォルト値のまま変更せずにおくことができます。

完了したら、*Add* をクリックします。新しいホストがホスト一覧に表示されるはずです。



Availability 列には、各インターフェースごとのホストの可用性を示すインジケーターが表示されます。ここでは Zabbix エージェントインターフェースを定義しているため、エージェント可用性アイコン（'ZBX' と表示されたもの）を使ってホストの可用性を確認できます。

- ZBX - ホストの状態はまだ確立されていません。まだメトリクスチェックは実行されていません
- ZBX - ホストは利用可能で、メトリクスチェックは成功しています
- ZBX - ホストは利用不可で、メトリクスチェックは失敗しています（エラーメッセージを確認するには、アイコンの上にマウスカーソルを移動します）。通信に何らかの問題がある可能性があり、インターフェース認証情報の誤りが原因である場合もあります。Zabbix サーバーが稼働していることを確認し、しばらくしてからページを再読み込みしてみてください。

概要

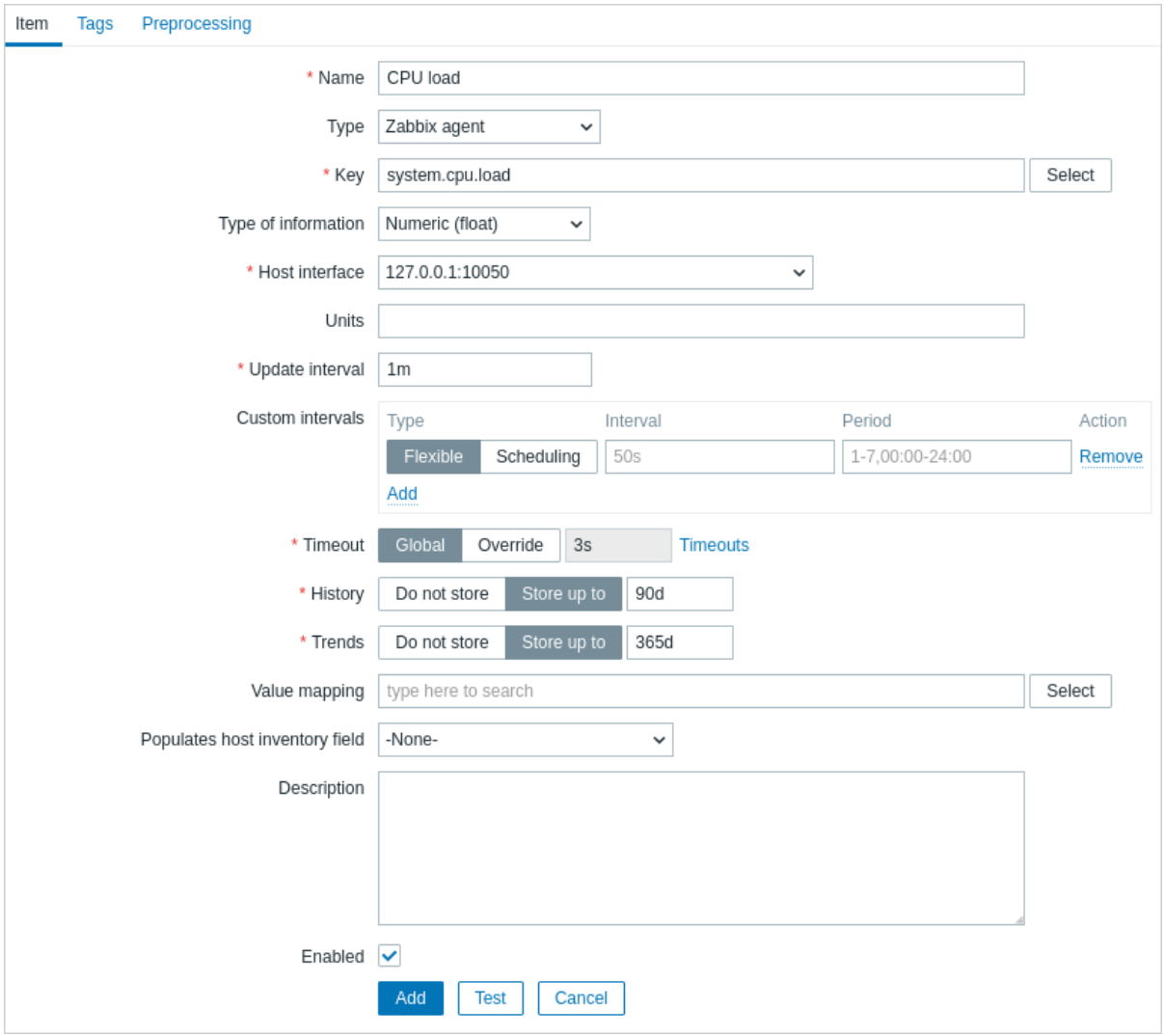
このセクションでは、どうやってアイテムを設定するかを説明します。

アイテムとは、Zabbix でデータを収集する基礎となるものです。アイテムなくしてデータは存在しえません。アイテムこそがメトリックの指標となり、ホストから収集するデータの種類を定義します。

アイテムの追加

すべてのアイテムはホストに関連付けられています。そのため、サンプルのアイテムを設定するには、データ収集 > ホストに移動し、作成した「New host」を見つけます。

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



必須の入力項目には、赤いアスタリスクが付いています。

このサンプルアイテムで入力する重要な情報は次のとおりです。

名前

- 値として *CPU load* を入力します。これは、一覧などに表示されるアイテム名になります。

キー

- 値として *system.cpu.load* を手動で入力します。これは、収集される情報の種類を識別するアイテムの技術的な名前です。このキーは、Zabbix エージェントに用意されている定義済みのキーの 1 つです。

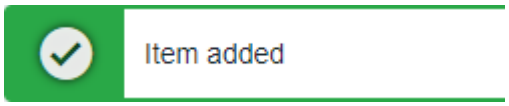
データ型

- この属性は、想定されるデータの形式を定義します。*system.cpu.load* キーの場合、この項目は自動的に 数値 (浮動小数) に設定されます。

Note:
アイテムの履歴の保存日数を、7 日または 14 日に減らすことも検討してください。これは、大量の履歴値を保持することによるデータベースの負荷を軽減するための良い方法です。

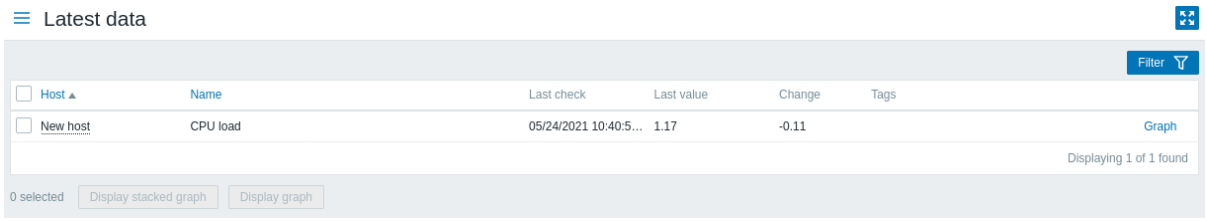
その他のオプションは、デフォルト値のまま変更せずに使用できます。

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



データの閲覧

アイテムを定義すると、それが実際にデータを収集しているかどうかが気になるかもしれません。確認するには、監視 > 最新データと移動し、フィルターで'New host' を選択し、適用をクリックします。



ただし、最初のデータが到着するまでに最大 60 秒かかる場合があります。サーバーが設定変更を読み取り、実行する新しいアイテムを取得する頻度の設定があり、そのデフォルトの設定値です。

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

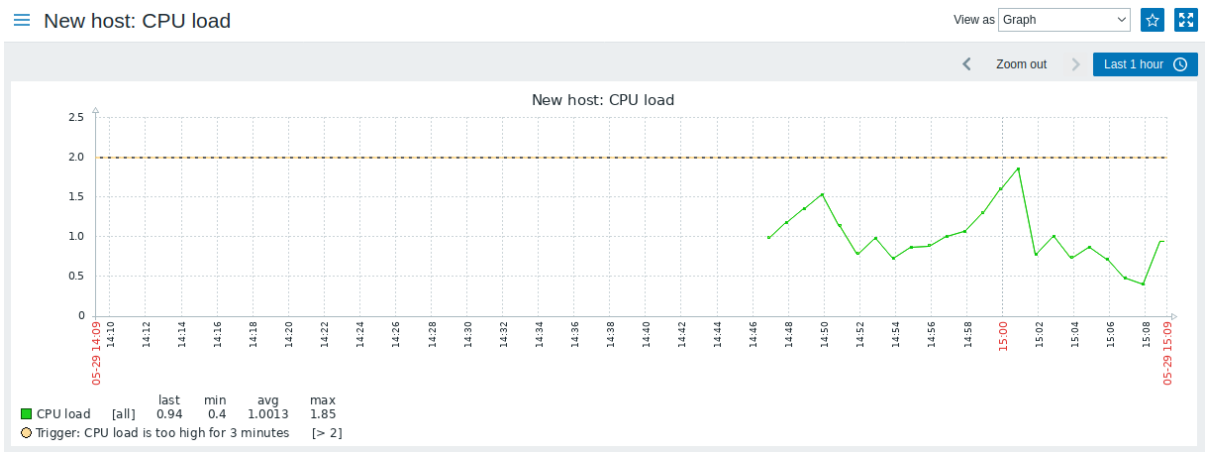
スクリーンショットのようなアイテムに関する情報が表示されない場合は、次のことを確認してください。

- アイテムの' キー' と' データ型' フィールドをスクリーンショットとまったく同じように入力したか。
- エージェントとサーバーの両方とも実行中か。
- ホストのステータスは' 有効' で、可用性アイコンは緑色か。
- ホストフィルターで選択されたホストは正しいか。
- アイテムは有効になっているか。

グラフ

アイテムがしばらくの間実行されてるのなら、その様子を視覚的に確認することが出来るかもしれません。シンプルグラフは監視してる全ての数値アイテムに対し、追加の設定なしに利用できます。これらのグラフは実行時に生成されます。

グラフを見るには、監視データ > 最新データに移動し、アイテムの横にある' グラフ' のリンクをクリックします。



4 新規トリガー

概要

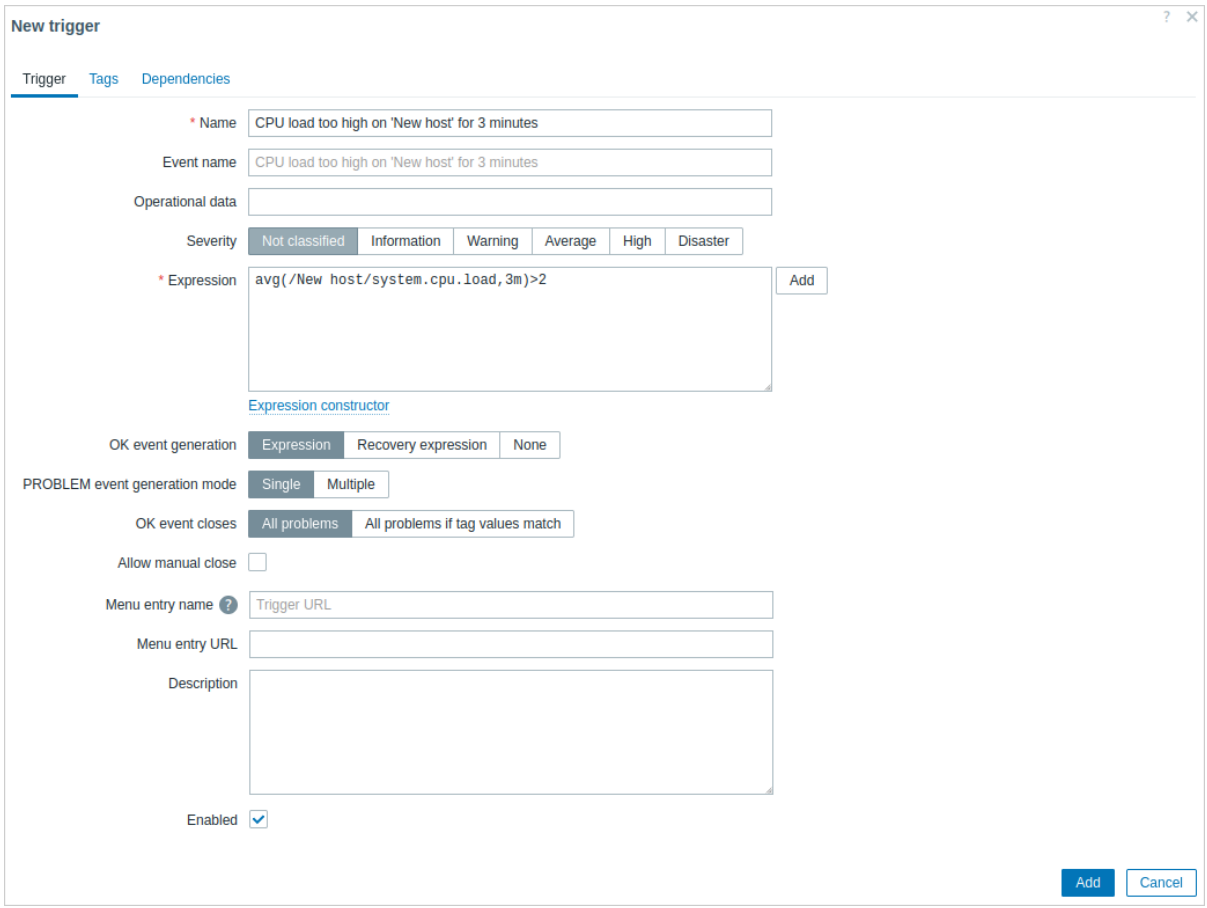
このセクションでは、トリガーの設定方法を学びます。

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

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

トリガーの追加

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



このトリガーで、ここに入力する必要がある重要な情報は次のとおりです。

名前

- 値として *'New host' の CPU* 負荷が *3* 分間高すぎますと入力します。
  これは、一覧などに表示されるトリガー名になります。

条件式

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

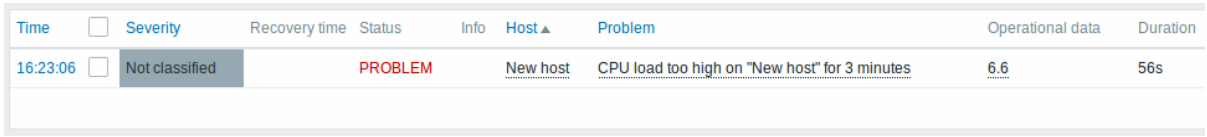
これはトリガー条件式です。最後の記号に至るまで、条件式が正しく入力されていることを確認してください。ここでは、アイテムキー (system.cpu.load) を使用してアイテムを参照しています。この条件式は基本的に、3 分間の CPU 負荷の平均値が 2 を超えたときに障害のしきい値を超えることを意味します。トリガー条件式の構文 について詳しく学ぶことができます。

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

トリガーステータスの表示

トリガーを定義すると、ステータスを確認したい場合があります。

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



ステータス列の点滅は、過去 30 分間に発生したトリガーステータスの最近の変更を示します。

概要

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

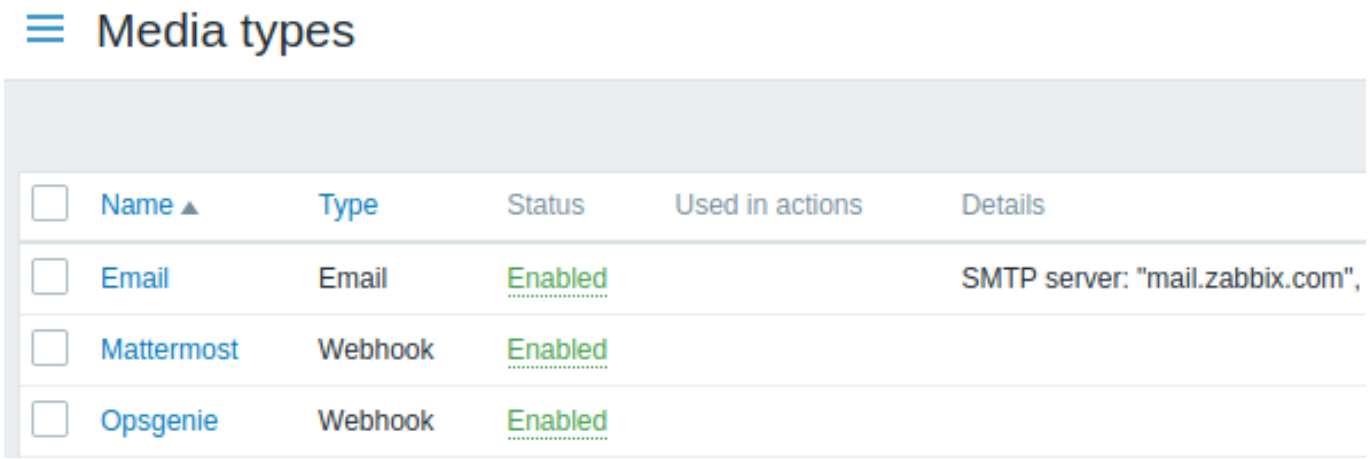
データを収集するアイテムと、問題の発生時に「発火」するよう設計されたトリガーがあるため、Zabbix の Web インターフェースを常に確認していない場合でも重要なイベントを知らせるためのアラートの仕組みがあると便利です。

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

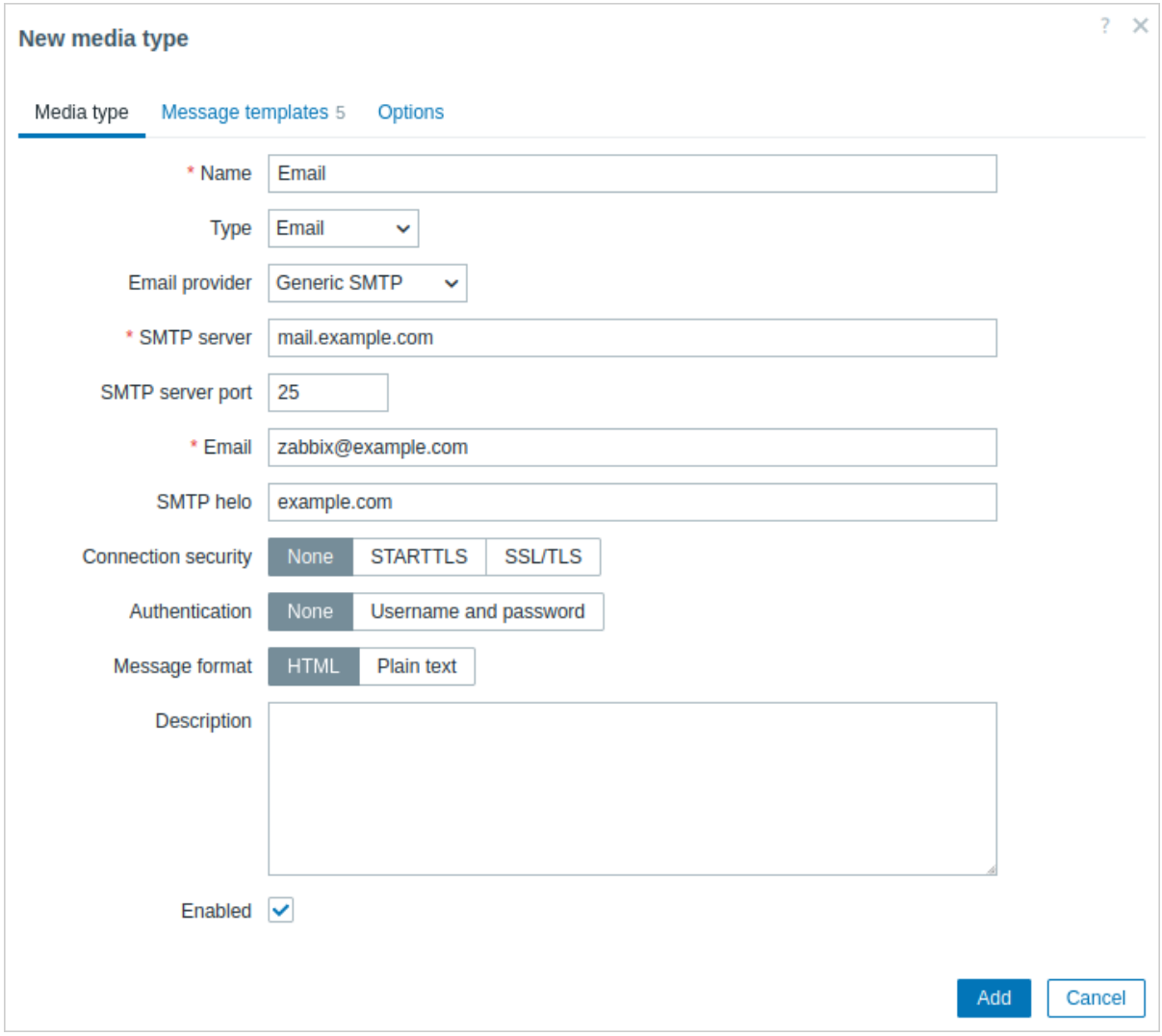
メール設定

Zabbix には、初期状態でいくつかの定義済み通知配信方法があります。Email はその 1 つです。

メール設定を構成するには、*Alerts* > *Media types* に移動し、定義済みメディアタイプの一覧で *Email* をクリックします。



メール設定の構成フォームが開きます。



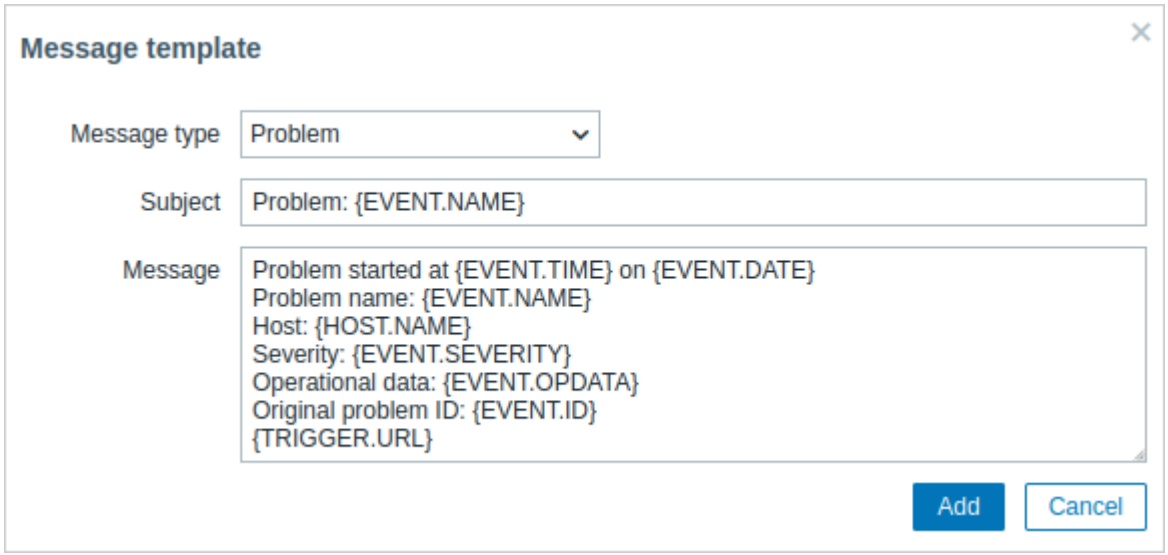
必須入力フィールドには、すべて赤いアスタリスクが付いています。

Media type タブで、*SMTP server*、*SMTP helo*、および *Email* の値を環境に応じて適切に設定します。

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

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

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

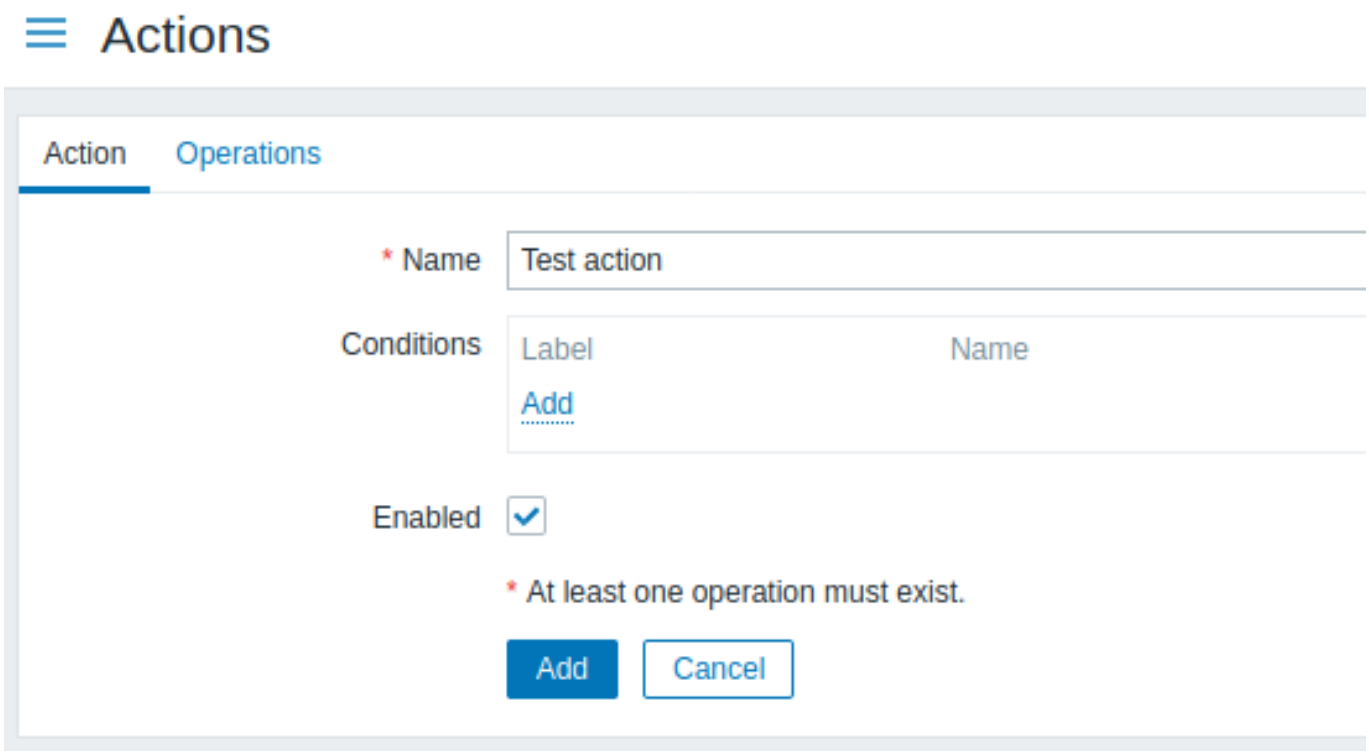


準備ができたら *Add* をクリックし、フォームを保存します。

これで、*Email* を動作するメディアタイプとして設定できました。ただし、このメディアタイプは、特定の配信先アドレスを定義してユーザーに関連付ける必要があります（新しいユーザーの設定で行ったように）。そうしないと使用されません。

新規アクション

Zabbix のアクションの動作の一つに、通知の配信があります。通知を設定するには、アラート > アクション > トリガーアクションと進み、アクションの作成をクリックします。

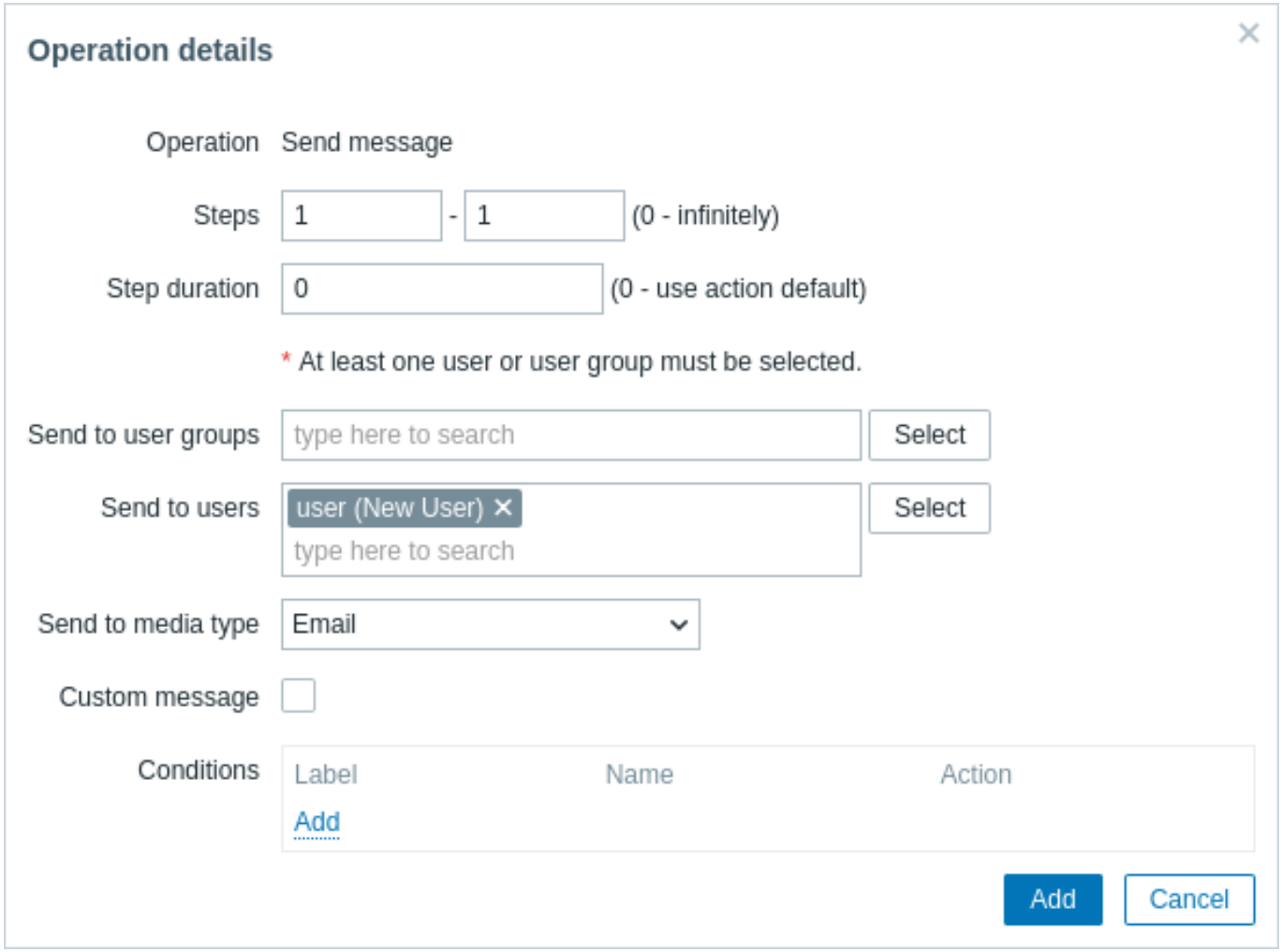


入力が必須な項目には赤いアスタリスクが表示されます。

この入力フォームでは、アクションの名前を入力します。

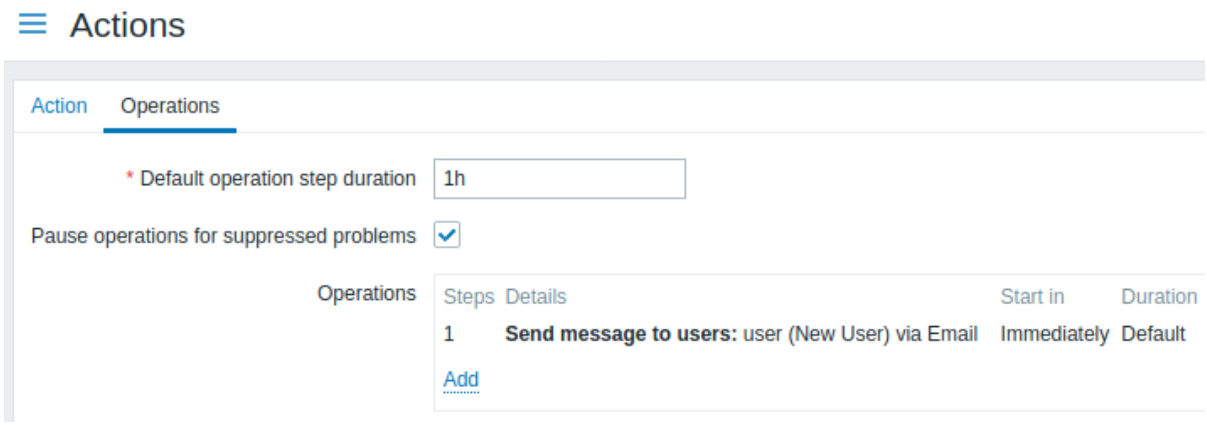
最も単純なケースでは、特定の条件を設定しない限り、トリガーが’ 正常’ から’ 障害’ に変化した際にアクションが実行されます。

アクションは何をすべきかの定義が必要で、実行内容タブで行われます。操作ブロック内の追加をクリックすると、新しい操作フォームが表示されます。



入力が必須な項目には赤いアスタリスクが表示されます。

ユーザーに送信内の選択をクリックし、定義済みのユーザー ('user') を選択します。送信メディアタイプの値に'Email' を設定します。選択後、追加をクリックすると動作が追加されます。



以上が単純なアクションの設定となります。アクションフォームの追加をクリックしてください。

通知受信

通知の配信の設定が出来たら、実際に通知を受け取ってみましょう。トリガーが" 発砲" され、障害通知を受け取れるように、わざとホストの CPU 負荷を上げます。

ホストのコンソールを開き、以下を実行してください。

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

これらのプロセスは一つでも複数でも実行できます。

次に監視データ > 最新データへ行き、'CPU Load' の値がどの程度増加したかを確認します。トリガーが発砲するには、3 分間の動作で'CPU Load' の値が'2' を超える必要があることを覚えておきましょう。一度でもそうなれば:

- 監視データ > 障害で、障害ステータスが点滅するトリガーが表示されます。
- あなたの E メールアドレスに障害発生通知が届いています。

Attention:
もし通知が上手く動作してない場合、
- もう一度 E メールの設定とアクションの設定が正しいかを確認してみてください。
- ユーザーの追加にて説明したように、作成したユーザーがそのイベントを生成したホストに関して、少なくとも読み取りの権限を持っていることを確認してください。'Zabbix Administrators' のユーザーグループにこのユーザーは所属しており、ホストが所属している'Linux servers' ホストグループに対して、少なくとも読み取り権限を持っている必要があります。
- 更にレポート > アクションログでアクションログを確認します。

6 新規テンプレート

概要

このセクションでは、テンプレートの設定について学びます。

これまでアイテムやトリガー、ホストの障害通知の設定方法を学びました。

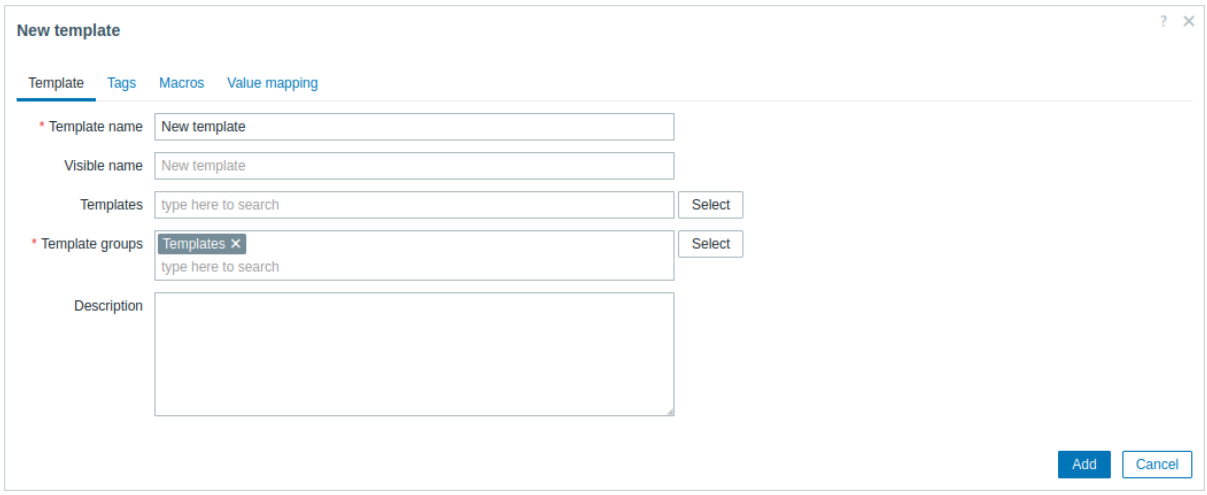
これらの手順は柔軟性に富んではいますが、例えば千台のホストが必要な場合には多くの手順を踏む必要があるため、自動化が出来れば便利でしょう。

そこで役に立つのがテンプレートです。これは有用なアイテムやトリガー、その他エンティティをグループ化して、一度の手順でホストに何度も再使用できるようにする為のものです。

テンプレートがホストにリンクされると、ホストはそのテンプレートの全容を継承します。したがって、基本的には前もって準備されたチェックの数々を迅速に適用することができます。

テンプレートの追加

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



必須の入力フィールドには、赤いアスタリスクが付いています。

ここで入力が必要なパラメータは次のとおりです。

テンプレート名

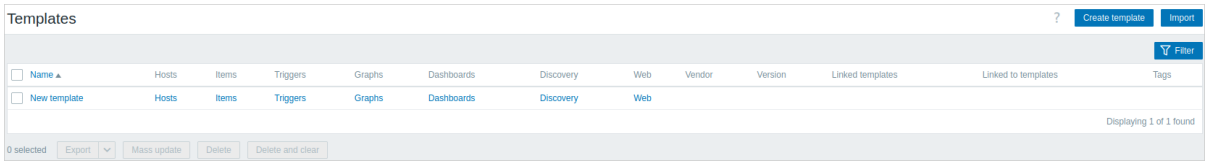
- テンプレート名を入力します。英数字、スペース、アンダースコアが使用できます。

テンプレートグループ

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

Note:
テンプレートグループへのアクセス権は、ホスト権限と同様に、ユーザーグループ の設定にある テンプレート権限タブで割り当てられます。すべてのアクセス権は個々のテンプレートではなくグループに対して割り当てられるため、テンプレートを少なくとも 1 つのグループに含めることが必須です。

完了したら、追加をクリックします。新しいテンプレートがテンプレートの一覧に表示されるはずです。また、フィルター を使用してテンプレートを見つけることもできます。



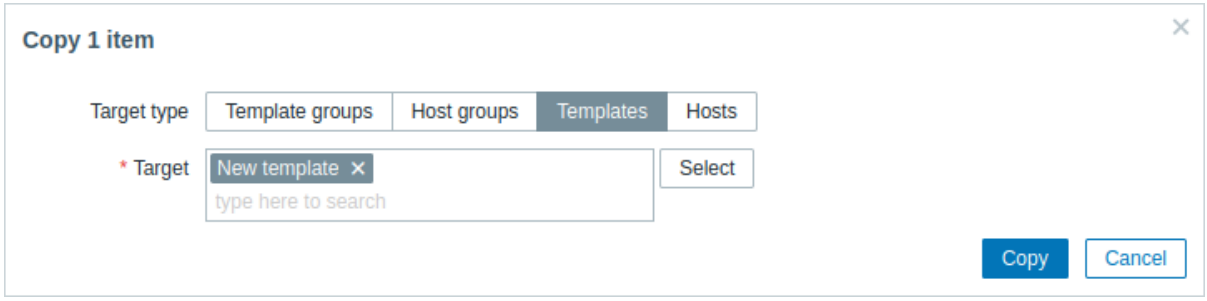
ご覧のとおり、テンプレートは作成されていますが、まだ中身は何もありません。アイテム、トリガー、その他のエンティティは含まれていません。

テンプレートにアイテムを追加

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

それから:

- リストの'CPU Load' アイテムのチェックボックスにチェックを入れます。
- リストの下にあるコピーをクリックします。
- テンプレートタブを選択します。
- アイテムをコピーするテンプレートを選びます。



記入が必須の項目には、赤いアスタリスクが表記されます。

- コピーをクリックします。

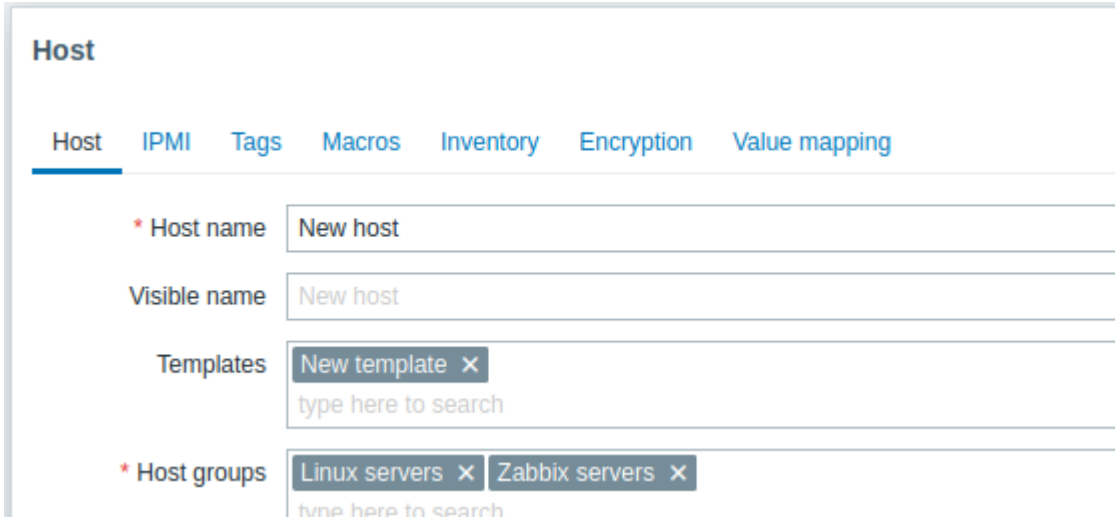
この段階でデータ収集 > テンプレートに行くと、'New template' に新しいアイテムが追加されているはずです。

この段階では 1 つのアイテムだけにしておきますが、同様にほかのアイテムやトリガー等を、希望の目的 (OS の監視、単一のアプリケーションの監視等) に対する動作の完全なセットを完成させるまでテンプレートに追加する事が出来ます。

テンプレートをホストにリンク

準備が出来たテンプレートがあれば、後はホストに追加するだけです。データ収集 > ホストに移動し、'New host' をクリックして設定フォームを開き、テンプレートのフィールドがある事を確認します。

テンプレートのフィールドに *New template* を入力します。作成したテンプレートの名前がドロップダウンリストに表示されているはずなので、スクロールして選択します。テンプレートフィールドに表示されている事を確認します。



フォーム内にある更新をクリックし、変更を保存します。これで全てのエンティティを保有するテンプレートがホストに追加されます。

この手順で他のどのホストにも適用する事が出来ます。アイテムやトリガー等の動作の変更は、テンプレートがリンクされているホストに伝搬されます。

事前に定義されたテンプレートをホストにリンク

お気づきかもしれませんが、Zabbix には様々な OS、デバイス、アプリケーション用の事前定義されたテンプレートが用意されています。適切なテンプレートをホストにリンクする事ですぐに監視を開始することが出来ますが、これらのテンプレートは環境に合わせて微調整する必要があることに注意してください。一部のチェックは不要かもしれまんせんし、ポーリング間隔が短すぎるかもしれません。

テンプレートの詳細については、こちらをご覧ください。

6. Zabbix アプライアンス

概要　Zabbix アプライアンスは、Zabbix サーバーとフロントエンドを手動で設定したり、Zabbix 用の既存のサーバーを再利用したりする代わりに、即座に展開する方法を提供します。

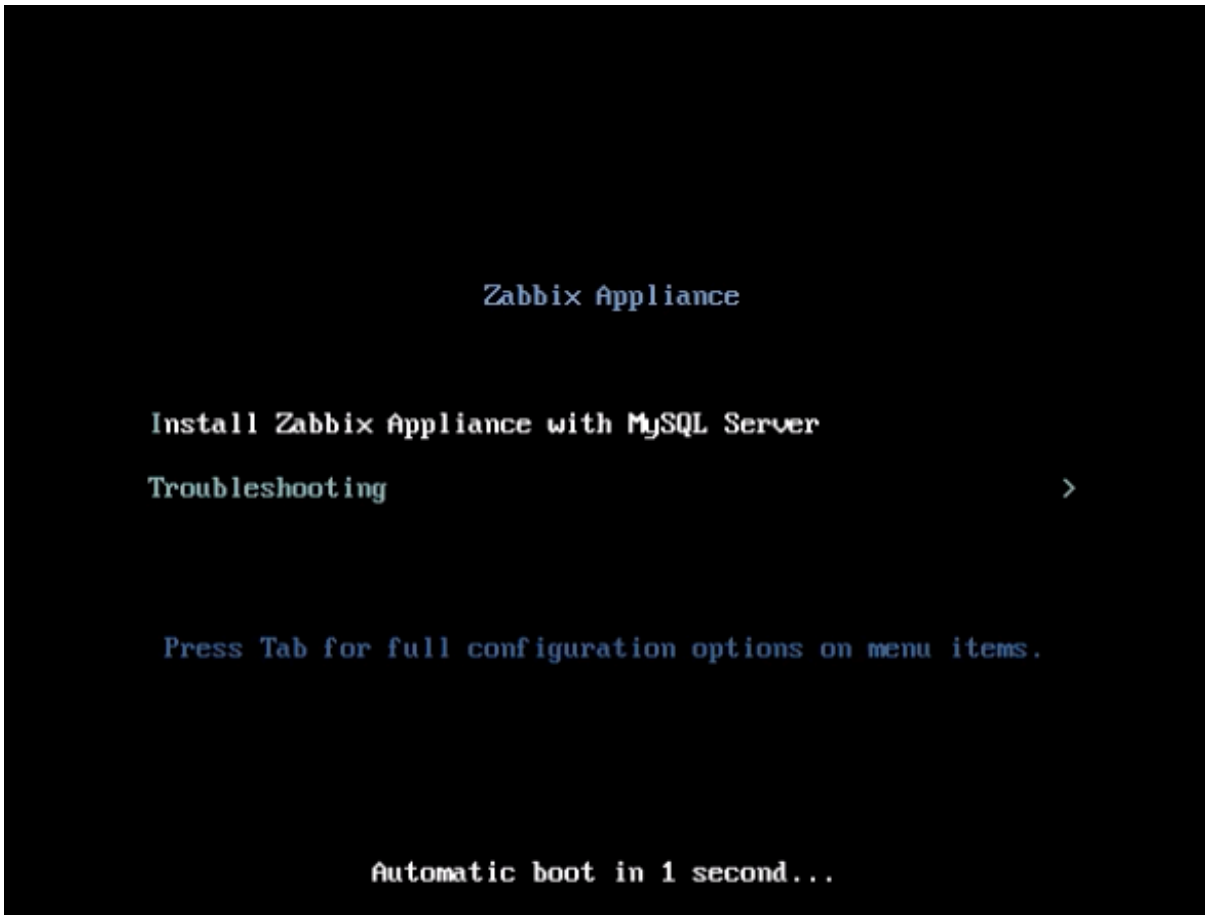
このアプライアンスは AlmaLinux 8 (x86_64) に基づいており、MySQL 上で実行される設定済みの Zabbix サーバーと Nginx Web サーバー上で実行されるフロントエンドが含まれています。

Attention:
このアプライアンスは、Zabbix を評価するために設計されています。本番環境での使用は推奨していません。

アプライアンスイメージは、次の形式でダウンロードできます。

- インストール CD/DVD (.iso)
- VMware (.vmx) - 注意点
- オープン仮想化フォーマット (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - 注意点
- KVM、Parallels、QEMU、USB スティック、VirtualBox、Xen (.raw) - 注意点
- KVM、QEMU (.qcow2)

Zabbix インストール CD/DVD のブートメニュー:



クイックスタート　前提条件

1. ホストマシンに、仮想マシンのシステム要件を満たすのに十分なリソースがあることを確認してください。

- *RAM*: 4 GB
- ディスク容量: 仮想マシンに最低 8GB
- *CPU*: 最低 2 コア

2. まだインストールされていない場合は、アプライアンスイメージを起動するための仮想化ソフトウェアをインストールします (例: VirtualBox)。
3. 仮想化ソフトウェアでサポートされている形式でアプライアンスをダウンロードします。
4. ネットワーク設定をチェックして、ホストマシンで DHCP が有効になっていることを確認します。

インストール

1. ダウンロードしたイメージからアプライアンス仮想マシンを起動します。
2. ホストマシン上のブラウザからアクセスできるように仮想マシンのネットワーク設定を構成します。これは *Bridged mode* を有効にすることで実現できます。
3. デフォルトのシステム認証情報を使用して仮想マシンにログインします。
4. IP アドレスを取得するには、仮想マシンで次のコマンドを実行します:

```
ip addr show
```

5. ホストマシンでブラウザを開き、アプライアンスが DHCP 経由で受信した IP アドレスを指定します。
6. デフォルトの認証情報を使用して Zabbix にログインし、監視を開始します。

設定　このセクションでは、頻繁に必要となるデフォルトの設定と、利用可能なカスタマイズオプションについて説明します。

ユーザー認証情報

システム

- ユーザー名: root
- パスワード: zabbix

Zabbix フロントエンド

- ユーザー名: Admin
- パスワード: zabbix

ログイン後、ユーザープロファイル設定でデフォルトのパスワードを変更したり、新しいユーザーを作成してデフォルトのパスワードを削除したりできます。

データベース

すべてのデータベースユーザーのパスワードは、インストールプロセス中にランダムに生成されます。データベースには次のユーザーが定義されています。

Root:

- ユーザー名: root
- パスワード: パスワードは`/root/.my.cnf` ファイルに保存されます。root アカウントではパスワードを入力する必要はありません。

Zabbix サーバー*:*

- ユーザー名: zabbix_srv
- パスワード: パスワードは`/etc/zabbix/zabbix\_server.conf` に保存されます。

Zabbix フロントエンド*:*

- ユーザー名: zabbix_web
- パスワード: パスワードは`/etc/zabbix/web/zabbix.conf.php` に保存されます。

データベースユーザーのパスワードを変更するには、MySQL および対応する設定ファイルでパスワードを変更します。

フロントエンドのアクセス

Zabbix フロントエンドには、*http://*< 仮想マシンの *IP*> からアクセスできます。

デフォルトでは、どこからでもアクセスが許可されます。アクセスを制限するには`/etc/nginx/conf.d/zabbix.conf` を変更します。編集したファイルを保存したら、root ユーザーとして SSH 経由でログインし、次のコマンドを実行して Nginx を再起動します。

```
systemctl restart nginx
```

静的 IP アドレス

デフォルトでは、アプライアンスは DHCP を使用して IP アドレスを取得します。静的 IP アドレスを指定するには:

- root ユーザーとしてログインします。
- 次のコマンドを実行し、値をカスタム IP アドレスに置き換えます。

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.10/24    # アプライアンスのIPアドレス/CIDRプレフィッ
nmcli connection modify eth0 ipv4.gateway 192.168.1.1     # ゲートウェイのIPアドレス
nmcli connection modify eth0 ipv4.dns 8.8.8.8    # DNSサーバーのIPアドレス
nmcli connection modify eth0 ipv4.method manual
systemctl restart network
```

あるいは、ファイル/etc/NetworkManager/system-connections/eth0.nmconnection を開いて次の行を追加することもできます。

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

編集したファイルを保存した後、systemctl restart network コマンドを実行します。

ファイアウォールの構成

ファイアウォール設定を管理するために、アプライアンスは事前定義されたルールを持つ iptables を使用します。

- SSH (22 TCP) ポートを開放
- Zabbix エージェント (10050 TCP) と Zabbix トラッパー (10051 TCP) ポートを開放
- HTTP (80 TCP) と HTTPS (443 TCP) ポートを開放
- SNMP trap (162 UDP) ポートを開放
- NTP (123 UDP) ポートの発信接続を開放
- ICMP パケットを 1 秒あたり 5 パケットに制限
- 他のすべての着信接続はドロップ

追加のポートを開くには、/etc/sysconfig/iptables ファイルを変更し、ファイアウォールルールをリロードします。

```
systemctl reload iptables
```

リポジトリ

Zabbix アプライアンスは、Zabbix のリポジトリにある *zabbix-release* パッケージを使用します。
リポジトリは /etc/yum.repos.d/* ディレクトリで設定されています。

タイムゾーン

デフォルトではアプライアンスはシステムクロックに UTC を使用します。タイムゾーンを変更するには適切なファイルを/usr/share/zoneinfo から/etc/localtime にコピーします。例:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

> **Note:**
> Zabbixフロントエンドのタイムゾーンは個別に設定され、フロントエンドの設定で変更できます。Zabbix フロントエンドのデフォルトのタイムゾーンは Europe/Riga です。

ファイルの場所

- 設定ファイルは/etc/zabbix にあります。
- Zabbix サーバー、プロキシ、エージェントのログファイルは/var/log/zabbix にあります。
- Zabbix フロントエンドは/usr/share/zabbix にあります。
- ユーザー *zabbix* のホームディレクトリは/var/lib/zabbix です。

システムサービス

Systemd サービスが利用可能です。Zabbix サービスのリストを表示するには、仮想マシンで次のコマンドを実行します。

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

イメージフォーマット固有の注意点　VMware

vmdk 形式のイメージは、VMware Player、Server および Workstation 製品で直接使用できます。ESX、ESXi および vSphere で使用するには、VMware vCenter コンバーター (authentication required for download) を使用して変換する必要があります。VMware vCenter コンバーターを使用する場合、ハイブリッドネットワークアダプタで問題が発生する可能性があります。その場合は、変換プロセス中に E1000 アダプタを指定してみてください。あるいは、変換の完了後、既存のアダプタを削除して E1000 アダプタを追加することもできます。

HDD/フラッシュイメージ（raw）

イメージを起動するには、次を実行します。

```
dd if=./zabbix_appliance_7.4.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

/dev/sdc は、お使いのフラッシュ/HDD ディスクデバイスのパスに置き換えてください。

Hyper-V

アプライアンスが Hyper-V で起動しない場合は、**Ctrl+Alt+F2** を押して TTY セッションを切り替えてみてください。

トラブルシューティング　Web インターフェースへのログイン時に、エラーメッセージ *Access denied for user 'replace_user'@'localhost' (using password: YES)* が表示される場合、インストールがまだ進行中である可能性があります。



数分待ってもエラーが解消されない場合、またはその他の予期しない動作が見られる場合は、インストールプロセスが正常に完了していない可能性があります。
この場合は、現在のアプライアンスを削除し、同じインストール手順に従って再デプロイすることをお勧めします。
通常、この手順で問題は解決します。

なお、破損したインストールを手動で修復しようとすることは推奨されません。さらに問題が複雑になる可能性があります。

7. 設定

サイドバーから、参照したい項目を選択してください。

1 テンプレートの設定

概要

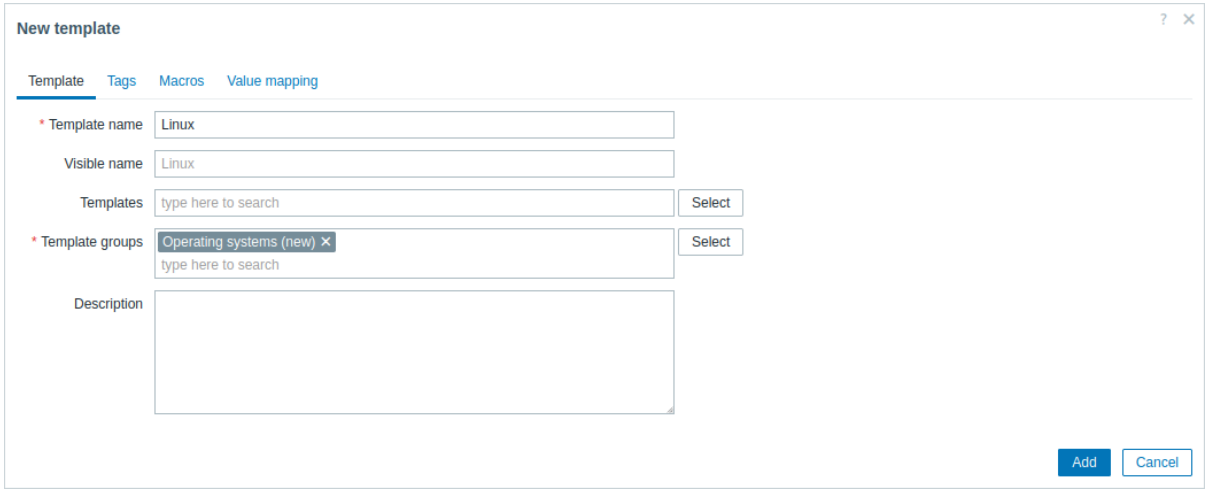
テンプレートを設定するには、最初に一般的なパラメーターを定義してテンプレートを作成し、次にエンティティ (アイテム、トリガー、グラフなど) をテンプレートに追加する必要があります。

テンプレートの作成

テンプレートを作成するには、次の手順を実行します。

1. データ収集 → テンプレートに移動します。
2. テンプレートの作成をクリックします。
3. テンプレート属性を編集します。

テンプレートタブには、テンプレートの一般属性が含まれています。

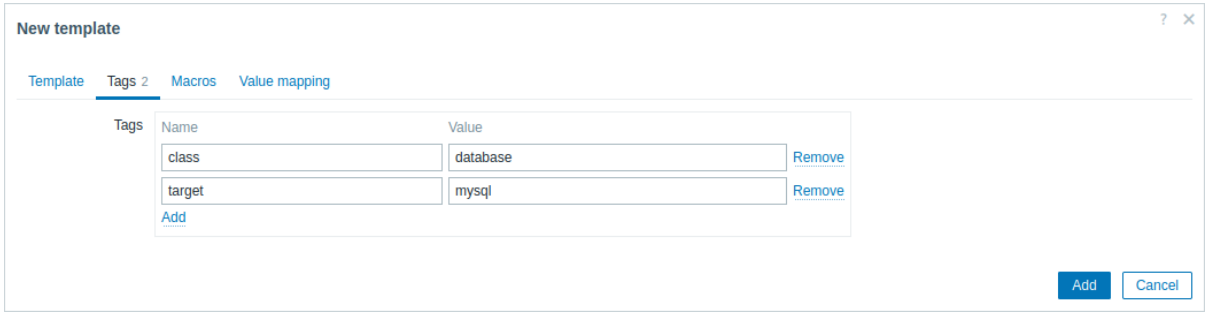


必須の入力フィールドには、すべて赤いアスタリスクが付いています。

テンプレート属性:

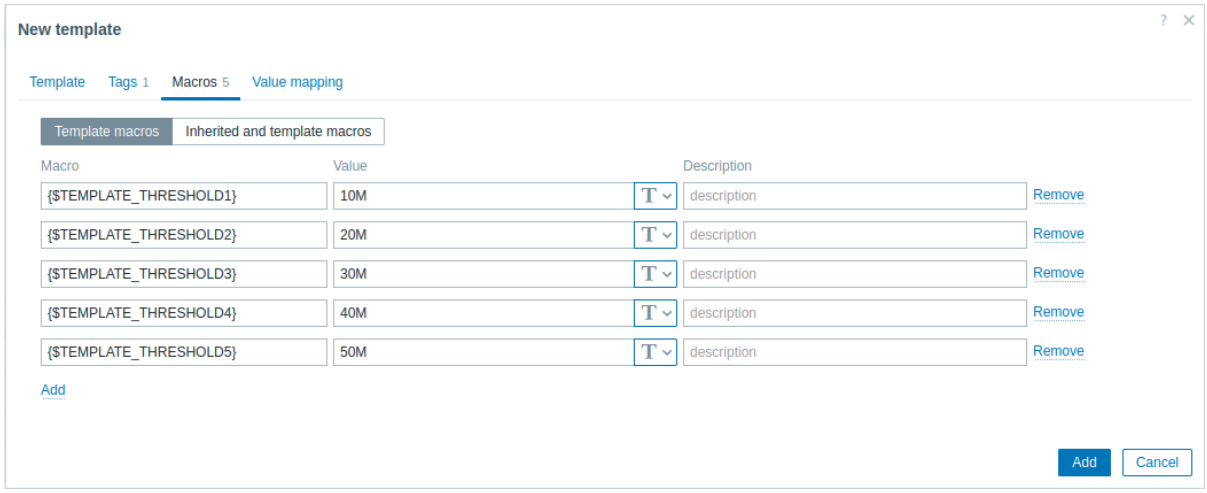
パラメータ	説明
テンプレート名	一意のテンプレート名。 英数字、スペース、ドット、ダッシュ、アンダースコアを使用できます。 先頭および末尾のスペースは使用できません。
表示名	この名前を設定すると、リスト、マップなどで表示される名前になります。
テンプレート	このテンプレートに 1 つ以上のテンプレートをリンクします。リンクされたテンプレートから、すべてのエンティティ（アイテム、トリガーなど）が継承されます。 新しいテンプレートをリンクするには、テンプレートフィールドにその名前を入力します。一致するテンプレートのドロップダウンリストが表示されます。あるいは、選択をクリックして テンプレートポップアップを開くこともできます。このポップアップには、初期状態ではテンプレートは表示されません。リストを表示するには、テンプレートグループフィールドにテンプレートグループ名を入力して候補リストから選択するか、テンプレートグループフィールドの横にある 選択ボタンを押して テンプレートグループポップアップを開きます。テンプレートグループを選択すると、テンプレートポップアップが更新され、そのグループに属するテンプレートが表示されます。選択したテンプレートは、設定の保存または更新時にリンクされます。 テンプレートのリンクを解除するには、テンプレートブロック内の次の 2 つのオプションのいずれかを使用します。 リンク解除 - テンプレートのリンクを解除しますが、そのエンティティ（アイテム、トリガーなど）は保持します。 リンク解除して削除 - テンプレートのリンクを解除し、そのすべてのエンティティ（アイテム、トリガーなど）を削除します。
テンプレートグループ	テンプレートが属するテンプレートグループ。
説明	テンプレートの説明。
ベンダーとバージョン	テンプレートのベンダーとバージョン。既存のテンプレート（Zabbix が提供する標準テンプレート、インポートされたテンプレート、またはTemplate API を通じて変更されたテンプレート）を更新する場合にのみ表示されます。テンプレート設定にその情報が含まれている必要があります。 Zabbix Web インターフェースでは変更できません。 標準テンプレートでは、バージョンは次の形式で表示されます: Zabbix のメジャーバージョン、区切り文字（”-”）、リビジョン番号（テンプレートの新しいバージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされます）。例: 7.0-0、7.0-5、7.4-0、7.4-3。

タグタブでは、テンプレートレベルのタグを定義できます。このテンプレートにリンクされたホストのすべての障害には、ここで入力した値のタグが付けられます。

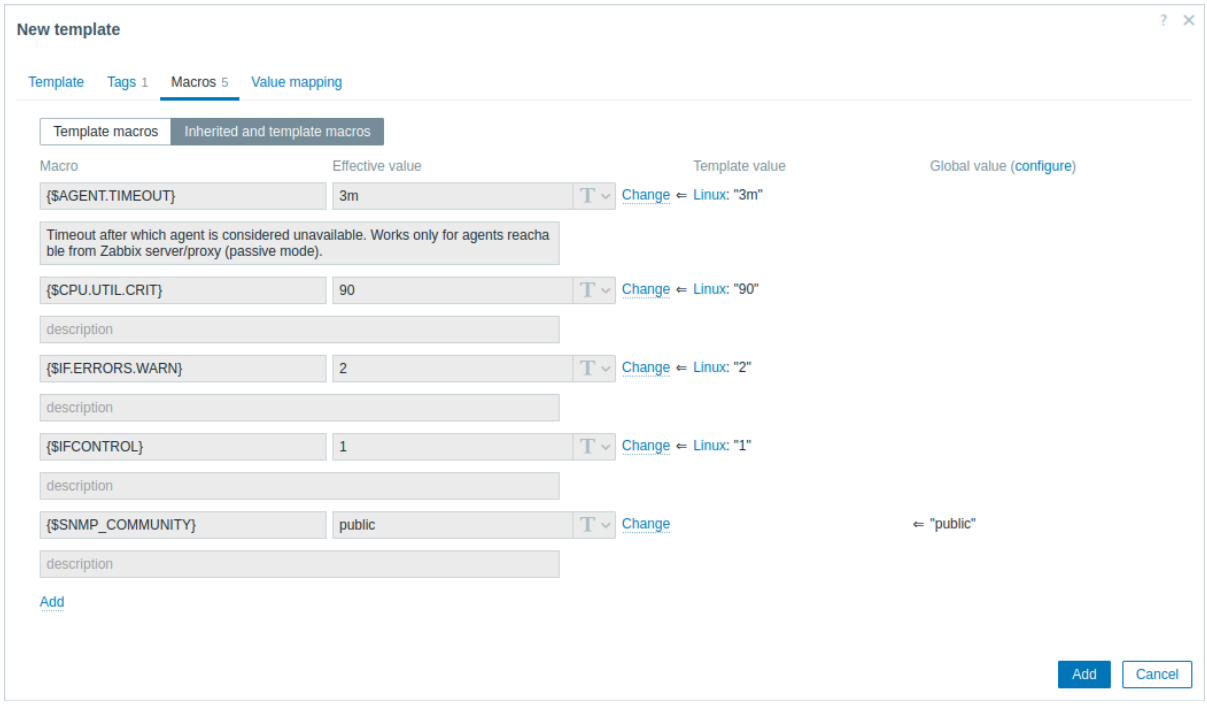


タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。

マクロタブでは、テンプレートレベルのユーザーマクロを名前と値のペアとして定義できます。マクロ値は、プレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できます。説明の追加もサポートされています。



継承されたマクロとテンプレートマクロオプションを選択すると、リンクされたテンプレートおよびテンプレートが継承するグローバルマクロのマクロと、それらのマクロが解決される値も表示できます。



利便性のために、それぞれのテンプレートへのリンクと、グローバルマクロ設定へのリンクも提供されています。また、リンクされたテンプレートマクロまたはグローバルマクロをテンプレートレベルで編集することもでき、実質的にそのマクロのコピーをテンプレート上に作成できます。

値のマッピングタブでは、値のマッピングでアイテムデータの人間にとって分かりやすい表現を設定できます。

ボタン:

Add	テンプレートを追加します。追加したテンプレートは一覧に表示されます。
Update	既存のテンプレートのプロパティを更新します。
Clone	現在のテンプレートのプロパティに基づいて別のテンプレートを作成します。これには、リンクされたテンプレートから継承されたエンティティ（アイテム、トリガーなど）と、現在のテンプレートに直接関連付けられたエンティティの両方が含まれます。ただし、複製されたテンプレートを元のテンプレートと区別できるようにするため、現在のテンプレートのベンダーとバージョンは含まれません。
Delete	テンプレートを削除します。テンプレートのエンティティ（アイテム、トリガーなど）は、リンクされたホストに残ります。
Delete and clear	テンプレートと、そのすべてのエンティティをリンクされたホストから削除します。
Cancel	テンプレートプロパティの編集をキャンセルします。

アイテム、トリガー、グラフの追加

Attention:
アイテムは最初にテンプレートへ追加する必要があります。対応するアイテムがない場合、トリガーとグラフは追加できません。

テンプレートにアイテムを追加する方法は 2 つあります:

1. 新しいアイテムを作成するには、アイテムの作成 のガイドラインに従ってください。

2. 既存のアイテムをテンプレートに追加するには:

- データ収集 → ホスト（または テンプレート）に移動します。
- 対象のホスト/テンプレートの行で アイテムをクリックします。
- テンプレートに追加したいアイテムのチェックボックスを選択します。
- アイテム一覧の下にある コピーをクリックします。
- アイテムのコピー先となるテンプレート（またはテンプレートグループ）を選択し、コピーをクリックします。
  選択したすべてのアイテムがテンプレートにコピーされます。

トリガーとグラフの追加も同様の方法で行います（それぞれトリガー一覧およびグラフ一覧から追加します）。ただし、必要なアイテムが先に追加されている場合にのみ追加できる点に注意してください。

ダッシュボードの追加

データ収集 → テンプレートでテンプレートにダッシュボードを追加するには、次の手順を実行します。

1. テンプレートの行で ダッシュボードをクリックします。
2. ダッシュボードの設定 のガイドラインに従ってダッシュボードを設定します。

Attention:
テンプレートダッシュボード上でウィジェットを設定する場合（グローバルダッシュボードではなく）、ホスト関連のパラメータは利用できず、一部のパラメータはラベルが異なります。
これは、テンプレートダッシュボードが、そのテンプレートにリンクされているホストのデータのみを表示するためです。
例えば、障害 ウィジェットでは、ホストグループ、除外するホストグループ、ホストの各パラメータは利用できません。
また、ホストの可用性 ウィジェットでは、ホストグループパラメータは利用できず、メンテナンス中のホストを表示パラメータは メンテナンス中のデータを表示に名称変更されています。
テンプレートダッシュボードのウィジェットで利用可能なパラメータの詳細については、各ダッシュボードウィジェットの個別パラメータを参照してください。

Note:
テンプレートダッシュボードから作成されたホストダッシュボードへのアクセス方法の詳細については、ホストダッシュボード セクションを参照してください。

ローレベルディスカバリの設定

マニュアルのローレベルディスカバリセクションを参照

Web シナリオの追加

データ収集 → テンプレートでテンプレートに Web シナリオを追加するには、次の手順を実行します。

1. テンプレートの行で *Web* をクリックします。
2. 通常のWeb シナリオの設定 方法に従って、Web シナリオを設定します。

2 テンプレートグループの設定

Overview

Template groups are used for the logical grouping of templates and assigning user permissions to them.

Each template must have at least one template group assigned. A template may belong to multiple template groups, and each template group may contain multiple templates.

Note that in Zabbix, all permissions are based on groups: user groups, host groups, and template groups. So, even if a single user needs access to a single template, it is granted by adding the user to a user group that has permission to access the template group containing that template.

設定

> **Attention:**
> テンプレートグループを作成できるのは、Super admin ユーザーのみです。

Zabbix Web インターフェースでテンプレートグループを作成する方法は 2 つあります。

1 つ目の方法:

- データ収集 → テンプレートグループに移動します
- 画面右上の テンプレートグループの作成をクリックします
- フォームにグループ名を入力します



2 つ目の方法: テンプレートを設定する際に、テンプレートグループ入力フィールドに存在しないグループ名を入力します。

テンプレートグループが作成されると、データ収集 → テンプレートグループの一覧でテンプレート名をクリックして、グループ名の編集、グループの複製、またはグループの削除を行うことができます。

テンプレートグループを削除しても、削除されるのは論理グループのみであり、そのグループ内のテンプレートは削除されません。既存のテンプレートにとって唯一のグループであるテンプレートグループは削除できません。

テンプレートサブグループの作成

テンプレートサブグループ（またはネストされたテンプレートグループ）は、それを含む親テンプレートグループの子です。

サブグループは、グループ名の入力フィールドで親グループとの関係を示すためにスラッシュ'/' を使用して作成します。例えば、次のようになります。

- `Linux servers/Databases` と入力すると、親グループ `Linux servers` のサブグループ `Linux servers/Databases` が作成されます。
- `Linux servers/Databases/MySQL/Tokyo` と入力すると、ネストされた親グループ `Linux servers`、`Linux servers/Databases`、`Linux servers/Databases/MySQL` 内に対応するサブグループが作成されます。

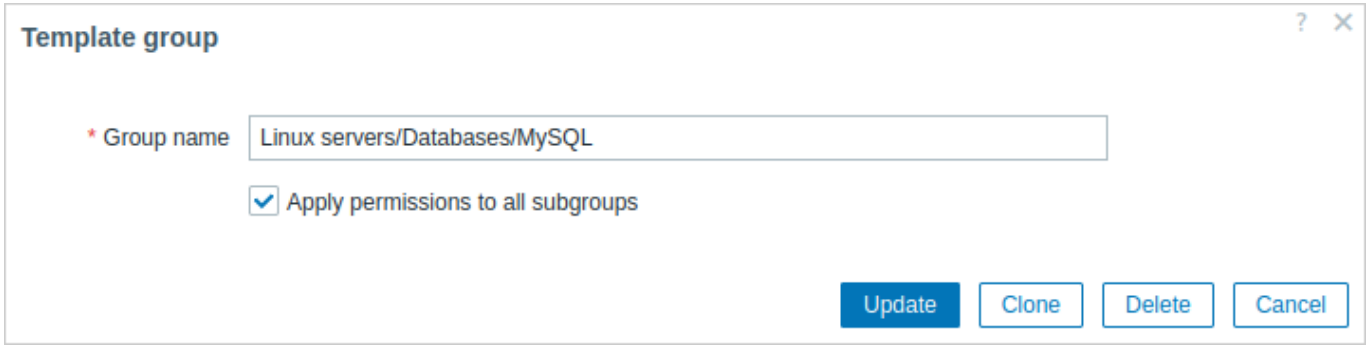
サブグループの作成時に、先頭または末尾にスラッシュを使用したり、スラッシュを連続して使用したりすることはできません。'/' のエスケープはサポートされていません。

サブグループを作成する前に、親テンプレートグループを作成しておく必要はありません。サブグループ（例えば Linux servers/Databases）の作成から始めることも、親テンプレートグループ（この例では Linux servers）の作成から始めることもできます。サブグループの作成から始めた場合、親テンプレートグループが自動的に作成されることは ありません。

テンプレートグループに対する権限

- 既存の親テンプレートグループにサブグループを作成する場合（例えば、Linux servers がすでに存在する状態で Linux servers/Databases を作成する場合）、そのサブグループに対するユーザーグループの権限は親から継承されます。
- 既存のサブグループに対して親テンプレートグループを作成する場合（例えば、Linux servers/Databases がすでに存在する状態で Linux servers を作成する場合）、親には権限は設定されません。

任意のテンプレートグループを編集する際には、追加オプションとして すべてのサブグループに権限を適用を設定することもできます。



このチェックボックスをオンにして 更新をクリックすると、編集中のテンプレートグループの現在および今後作成されるすべてのサブグループに、同じ権限レベルが適用されます。

したがって、編集中のテンプレートグループのサブグループに対して、いずれかのユーザーグループに異なる権限が付与されている場合、このチェックボックスをオンにすると、現在および今後作成されるすべてのサブグループに、編集中のグループと同じユーザー権限が付与されます。

このオプションはデータベースには保存されず、既存の権限を上書きすることに注意してください。このオプションによって行われた変更は、手動でのみ元に戻すことができます。

3 リンク作成/リンク削除

概要

リンクは、テンプレートをホストに適用するプロセスです。一方、リンク解除は、ホストからテンプレートとの関連付けを削除します。

テンプレートのリンク

テンプレートをホストにリンクするには、次の手順を実行します。

1. データ収集 → ホストに移動します。
2. 対象のホストをクリックします。
3. テンプレートフィールドにテンプレート名の入力を開始します。一致するテンプレートの一覧が表示されるので、スクロールして選択します。
   または、フィールドの横にある 選択をクリックし、ポップアップウィンドウの一覧から 1 つまたは複数のテンプレートを選択することもできます。
4. ホスト属性フォームで 追加/更新をクリックします。

これで、ホストはテンプレートのすべてのエンティティを持つようになります。
これには、アイテム、トリガー、グラフ、ローレベルディスカバリルール、Web シナリオ、およびダッシュボードが含まれます。

Attention:
同じホストに複数のテンプレートをリンクする場合、それらのテンプレートに同じアイテムキーを持つアイテムが含まれていると、リンクは失敗します。
また、トリガーやグラフはアイテムを使用するため、同一のアイテムキーを使用している場合、それらも複数のテンプレートから 1 つのホストにリンクすることはできません。

テンプレートからエンティティ（アイテム、トリガーなど）が追加されると、次のようになります。

- ホスト上に以前から存在していた同一のエンティティは、テンプレートのエンティティとして更新され、そのエンティティに対してホストレベルで行われていた既存のカスタマイズはすべて失われます。
- テンプレートのエンティティが追加されます。

- テンプレートのリンク前にホスト上にのみ存在していた直接リンクされたエンティティは、そのまま変更されません。

一覧では、テンプレート由来のすべてのエンティティの先頭にテンプレート名が付加され、そのエンティティが特定のテンプレートに属していることを示します。
テンプレート名自体（灰色のテキスト）はリンクになっており、テンプレートレベルでそれらのエンティティ一覧にアクセスできます。

Note:
外部チェック、HTTP エージェントチェック、シンプルチェック、SSH チェック、Telnet チェック など、一部のアイテムではホストインターフェースは任意です。
テンプレートをリンクする時点でホストにインターフェースが定義されていない場合、これらのアイテムはインターフェースなしで追加されます。
後からホストインターフェースを追加しても、既存のアイテムには自動的に割り当てられません。
新しく追加したホストインターフェースをすべてのテンプレートアイテムに一括で割り当てるには、テンプレートをホストからリンク解除してから、再度リンクしてください。
アイテムの履歴を保持するには、リンク解除オプションを使用し、リンク解除して削除は使用しないでください。

テンプレート名の接頭辞が付いていないエンティティがある場合、それはテンプレートによって追加されたものではなく、以前からホスト上に存在していたことを意味します。

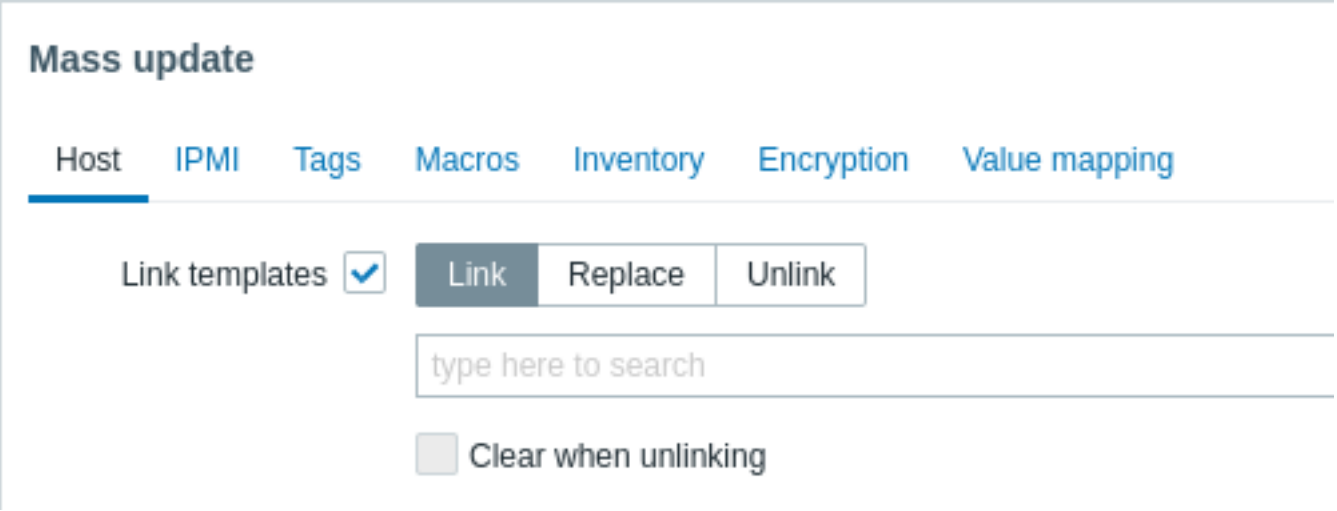
エンティティの一意性の基準

テンプレートからエンティティ（アイテム、トリガーなど）を追加する際には、それらのエンティティのうちどれがすでにホスト上に存在していて更新が必要なのか、またどのエンティティが異なるのかを把握することが重要です。同一性/相違を判断するための一意性の基準は次のとおりです。

- アイテムの場合 - アイテムキー
- トリガーの場合 - トリガー名と条件式
- カスタムグラフの場合 - グラフ名とそのアイテム群

複数のホストへのテンプレートのリンク

多数のホストのテンプレートのリンク設定を更新するには、データ収集 → ホストでいくつかのホストのチェックボックスをオンにしてから、一覧の下にある一括更新をクリックし、次にテンプレートをリンクを選択します。



追加のテンプレートをリンクするには、一致するテンプレートが候補として表示されるドロップダウンが現れるまで、オートコンプリート欄にテンプレート名を入力し始めます。リンクするテンプレートを選択するには、下にスクロールするだけです。

置換オプションでは、ホストに以前リンクされていたテンプレートのリンクを解除しながら、新しいテンプレートをリンクできます。
リンク解除オプションでは、どのテンプレートのリンクを解除するかを指定できます。リンク解除時に削除オプションでは、以前リンクされていたテンプレートのリンクを解除するだけでなく、それらから継承されたすべてのエンティティ（アイテム、トリガーなど）も削除できます。

Note:
Zabbix には、あらかじめ定義された多数のテンプレートが用意されています。これらは参考として利用できますが、本番環境でそのまま使用する場合は注意してください。アイテム数が多すぎたり、データのポーリング頻度が高すぎたりする可能性があります。使用する場合は、実際の要件に合わせて適切に調整してください。

リンクされたエンティティの編集

テンプレートからリンクされたアイテムまたはトリガーを編集しようとすると、多くの主要なオプションが編集不可になっていることに気付く場合があります。
これは、テンプレートの考え方が、テンプレートレベルで一括して編集を行うことにあるため、理にかなっています。
ただし、たとえば個々のホストでアイテムを有効化/無効化したり、更新間隔、履歴の保持期間、その他いくつかのパラメータを設定したりすることは可能です。

エンティティを完全に編集したい場合は、テンプレートレベルで編集する必要があります（テンプレートレベルへのショートカットはフォーム名に表示されます）。ただし、これらの変更は、このテンプレートがリンクされているすべてのホストに影響することに注意してください。

> **Attention:**
> テンプレートレベルでエンティティに対して行われたカスタマイズは、ホストレベルでそのエンティティに対して以前行われたカスタマイズを上書きします。

テンプレートのリンク解除

ホストからテンプレートのリンクを解除するには、次の手順を実行します。

1. データ収集 → ホストに移動します。
2. 対象のホストをクリックし、テンプレートフィールドを見つけます。
3. テンプレートのリンクを解除するには、対象のテンプレートの横にある リンク解除または リンク解除してクリアをクリックします。
4. ホスト属性フォームで 更新をクリックします。

リンク解除オプションを選択すると、テンプレートとの関連付けのみが削除され、そのすべてのエンティティはホストに残ります。これには、アイテム、トリガー、グラフ、ローレベルディスカバリルール、Web シナリオが含まれますが、ダッシュボードは含まれません。
なお、リンクされたテンプレートから継承された値マップおよびタグも削除されます。

リンク解除してクリアオプションを選択すると、テンプレートとの関連付けと、そのすべてのエンティティ（アイテム、トリガーなど）の両方が削除されます。

4 ネスト

概要

ネストは、1 つのテンプレートが 1 つ以上の他のテンプレートを包含する方法です。

さまざまなサービス、アプリケーションなどの個別のテンプレートでエンティティを分離することは理にかなっていますが、最終的に大量のテンプレートが作成され、またそのすべてを大量の数のホストにリンクする必要になる可能性があります。単純化するために、複数のテンプレートを 1 つのテンプレートにリンクすることができます。

ネストの利点は、ホストにリンクするテンプレートが 1 つだけで済み、ホストはその 1 つのテンプレートにリンクされているテンプレートからすべてのエンティティを自動的に継承することです。たとえば、テンプレート *T1* と *T2* を *T3* にリンクすると、*T1* と *T2* のエンティティが *T3* に追加されますが、その逆はありません。*T1* を *T2* と *T3* にリンクすると、*T1* のエンティティが *T2* と *T3* に追加されます。

ネストされたテンプレートの設定

テンプレートをリンクするには、既存のテンプレートを取得するか、新しいテンプレートを作成する必要があります。そして次に:

1. テンプレート設定フォームを開きます。
2. テンプレートフィールドを見つけます。
3. 選択をクリックして、テンプレートポップアップウィンドウを開きます。
4. ポップアップウィンドウで必要なテンプレートを選択し、選択をクリックしてテンプレートをリストに追加します。
5. テンプレート設定フォームで追加または更新をクリックします。

したがって、設定されたテンプレートのすべてのエンティティと、リンクされたテンプレートのすべてのエンティティがテンプレート設定に表示されます。これには、アイテム、トリガー、グラフ、ローレベルディスカバリルール、Web シナリオが含まれますが、ダッシュボードは含まれません。ただし、リンクされたテンプレートダッシュボードはホストによって継承されます。

リンクされているテンプレートのリンクを解除するには、テンプレート設定フォームでリンクを削除またはリンクと保存データを削除をクリックし、更新をクリックします。

リンクを削除オプションは、リンクされたテンプレートとの関連付けを削除しますが、そのエンティティ (アイテム、トリガーなど) は削除されません。

リンクと保存データを削除オプションは、リンクされたテンプレートとの関連付けと、そのすべてのエンティティ (アイテム、トリガーなど) の両方が削除されます。

5 一括アップデート

概要

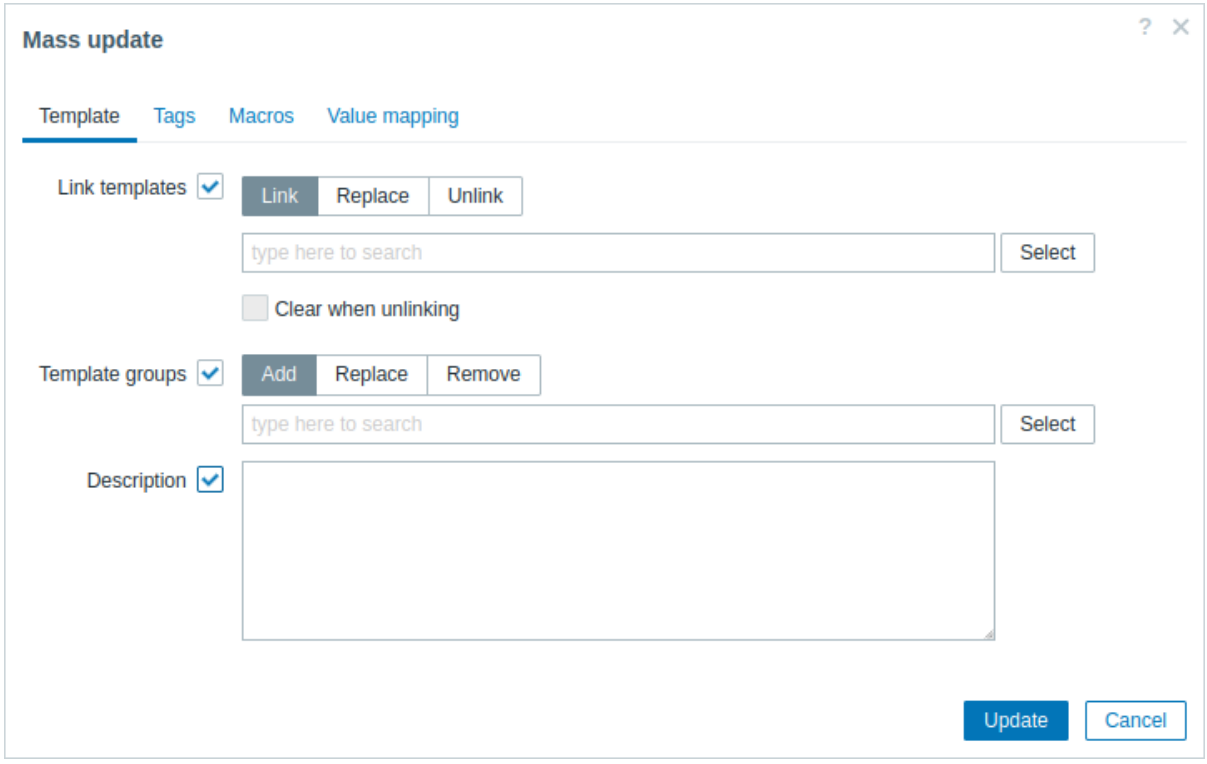
いくつかのテンプレートの属性を一度に変更したい場合があります。個々のテンプレートを開いて編集する代わりに、一括更新機能を使用できます。

一括更新の使用

複数のテンプレートを一括更新するには、次の手順を実行します。

1. テンプレート一覧で、更新したいテンプレートの前にあるチェックボックスをオンにします。
2. 一覧の下にある 一括更新をクリックします。
3. 必要な属性のタブ（テンプレート、タグ、マクロ、または 値のマッピング）に移動します。
4. 更新する属性のチェックボックスをオンにし、新しい値を入力します。

テンプレートタブには、テンプレートの一括更新に関する一般的なオプションが含まれています。



テンプレートのリンクの更新では、対応するボタンを選択すると、以下のオプションを利用できます。

- リンク - 追加でリンクするテンプレートを指定します。
- 置換 - リンクするテンプレートを指定すると同時に、以前にリンクされていたテンプレートのリンクを解除します。
- リンク解除 - リンクを解除するテンプレートを指定します。

リンク/リンク解除するテンプレートを指定するには、自動補完フィールドにテンプレート名を入力し始めてください。一致するテンプレートを候補として表示するドロップダウンが表示されます。必要なテンプレートを選択するには、下にスクロールしてください。

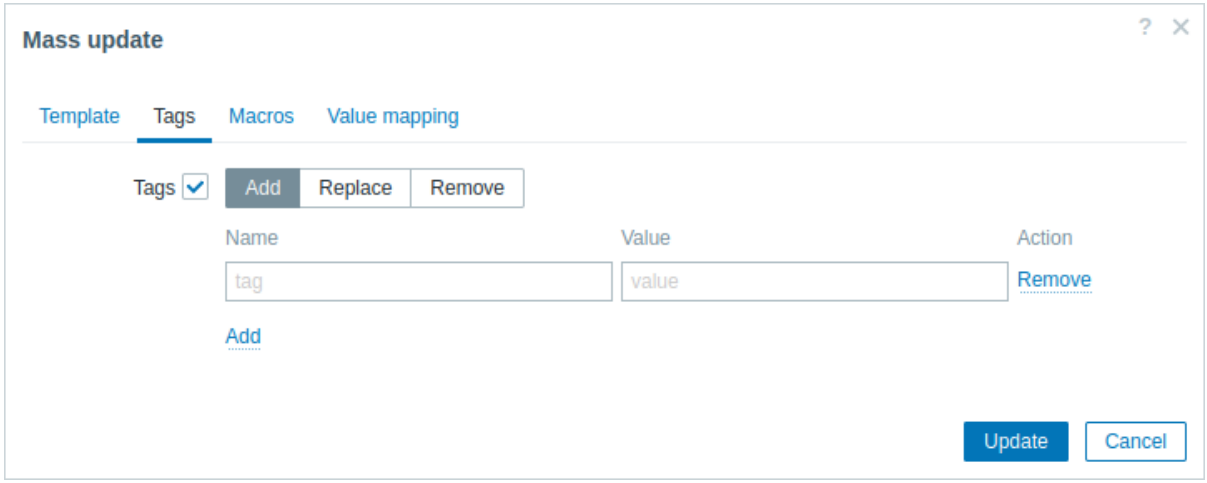
リンク解除時に削除オプションを使用すると、以前にリンクされていたテンプレートのリンクを解除できるだけでなく、それらから継承されたすべての要素（アイテム、トリガー、グラフなど）も削除できます。

テンプレートグループの更新では、対応するボタンを選択すると、以下のオプションを利用できます。

- 追加 - 既存のテンプレートグループから追加のテンプレートグループを指定するか、テンプレート用にまったく新しいテンプレートグループを入力できます。
- 置換 - テンプレートを既存のすべてのテンプレートグループから削除し、このフィールドで指定したテンプレートグループ（既存または新規）に置き換えます。
- 削除 - テンプレートから特定のテンプレートグループを削除します。

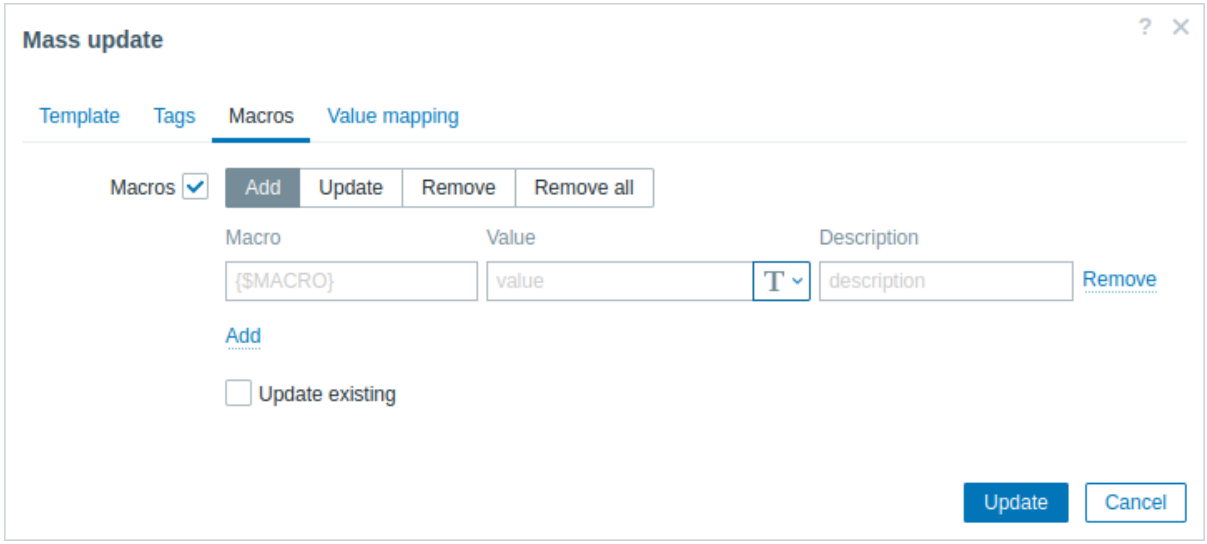
これらのフィールドは自動補完に対応しています。入力を開始すると、一致するテンプレートグループのドロップダウンが表示されます。テンプレートグループが新規の場合もドロップダウンに表示され、文字列の後ろに *(new)* と表示されます。選択するには、下にスクロールしてください。

タグタブでは、テンプレートレベルのタグを一括更新できます。



タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。同じ名前で値が異なるタグは「重複」とは見なされず、同じテンプレートに追加できることに注意してください。

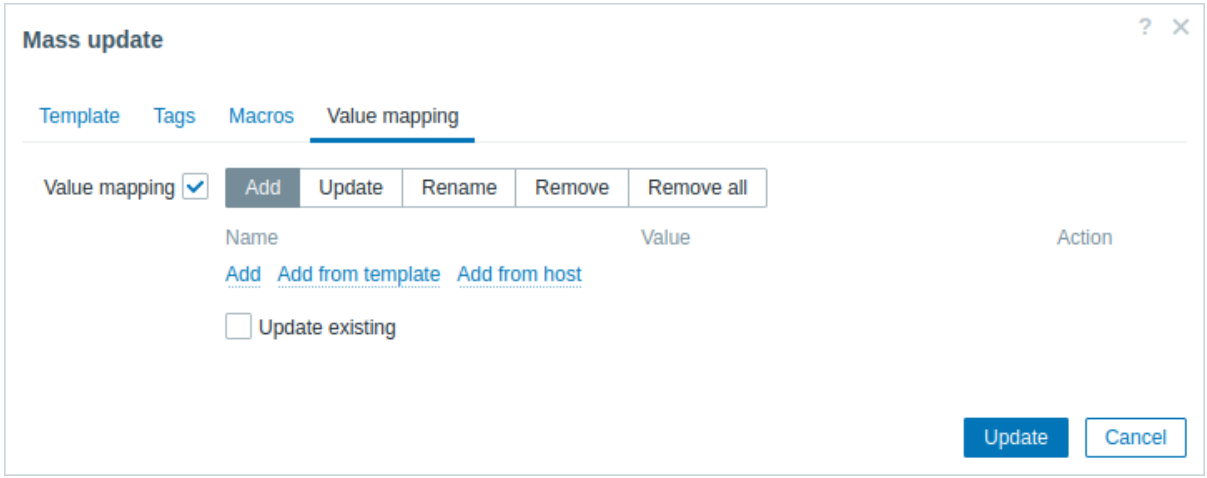
マクロタブでは、テンプレートレベルのマクロを一括更新できます。



マクロ更新で各ボタンを選択した場合、以下のオプションを利用できます。

- 追加 - テンプレートに追加のユーザーマクロを指定できます。
  既存を更新チェックボックスをオンにすると、指定したマクロ名の値、タイプ、説明が更新されます。
  オフの場合、その名前のマクロがテンプレートにすでに存在していても更新されません。
- 更新 - この一覧で指定したマクロの値、タイプ、説明を置き換えます。
  不足分を追加チェックボックスをオンにすると、テンプレート上に以前存在しなかったマクロは新しいマクロとして追加されます。
  オフの場合、テンプレート上にすでに存在するマクロのみが更新されます。
- 削除 - 指定したマクロをテンプレートから削除します。
  選択したものを除くチェックボックスをオンにすると、一覧で指定したもの以外のすべてのマクロが削除されます。
  オフの場合、一覧で指定したマクロのみが削除されます。
- すべて削除 - テンプレートからすべてのユーザーマクロを削除します。
  すべてのマクロを削除することを確認しますチェックボックスがオンになっていない場合、すべてのマクロを削除する確認を求める新しいポップアップウィンドウが開きます。

値のマッピングタブでは、値のマッピングを一括更新できます。



値のマッピングの更新では、次のオプションを持つボタンを利用できます。

- 追加 - テンプレートに値のマッピングを追加します。既存のものを更新をチェックすると、この名前を持つ値のマッピングのすべてのプロパティが更新されます。チェックしない場合、同じ名前の値のマッピングがすでに存在していても更新されません。
- 更新 - 既存の値のマッピングを更新します。不足分を追加をチェックすると、以前はテンプレート上に存在しなかった値のマッピングが新しい値のマッピングとして追加されます。チェックしない場合は、テンプレート上にすでに存在する値のマッピングのみが更新されます。
- 名前変更 - 既存の値のマッピングに新しい名前を付けます。
- 削除 - 指定した値のマッピングをテンプレートから削除します。選択したもの以外をチェックすると、指定したものを除くすべての値のマッピングが削除されます。
- すべて削除 - テンプレートからすべての値のマッピングを削除します。すべての値のマッピングを削除することを確認しますチェックボックスがチェックされていない場合、削除の確認を求める新しいポップアップウィンドウが開きます。

値のマッピングの追加/更新操作では、テンプレートから追加および ホストから追加オプションを利用できます。
これらを使用すると、それぞれテンプレートまたはホストから値のマッピングを選択できます。

必要な変更をすべて完了したら、更新をクリックします。
選択したすべてのテンプレートの属性が、それに応じて更新されます。

1 ホストとホストグループ

「ホスト」とは何ですか？

Zabbix における「ホスト」とは、物理または仮想デバイス、アプリケーション、サービス、または監視対象パラメータの論理的に関連するその他の集合を指します。

ホストの作成は、Zabbix における最初の監視作業の 1 つです。たとえば、サーバー「x」のいくつかのパラメータを監視したい場合、まず「Server X」などの名前のホストを作成し、その後でそのホストに監視アイテムを追加できます。

ホストはホストグループに整理されます。

次に進んでください:

- 新規または既存のホストを作成および設定するための、ガイド付きのステップバイステップのインターフェースについては、ホストウィザード を参照してください。
- ホストを作成および設定する従来の方法については、ホストの設定 を参照してください。

1 ホストウィザード

概要

ホストウィザードは、Zabbix で監視対象（デバイス、アプリケーション、サービスなど）を設定するための、ガイド付きのステップバイステップのインターフェースです。このウィザードでは、次の手順に沿って設定を進めます。

- テンプレートの選択
- ホストの作成または選択
- Zabbix エージェントまたは agent 2 のインストール
- ホストインターフェースの追加
- 監視対象または Zabbix ホストへの追加設定の適用（テンプレートで必要な場合）

Zabbix Web インターフェースでホストウィザードにアクセスするには、次の操作を行います。

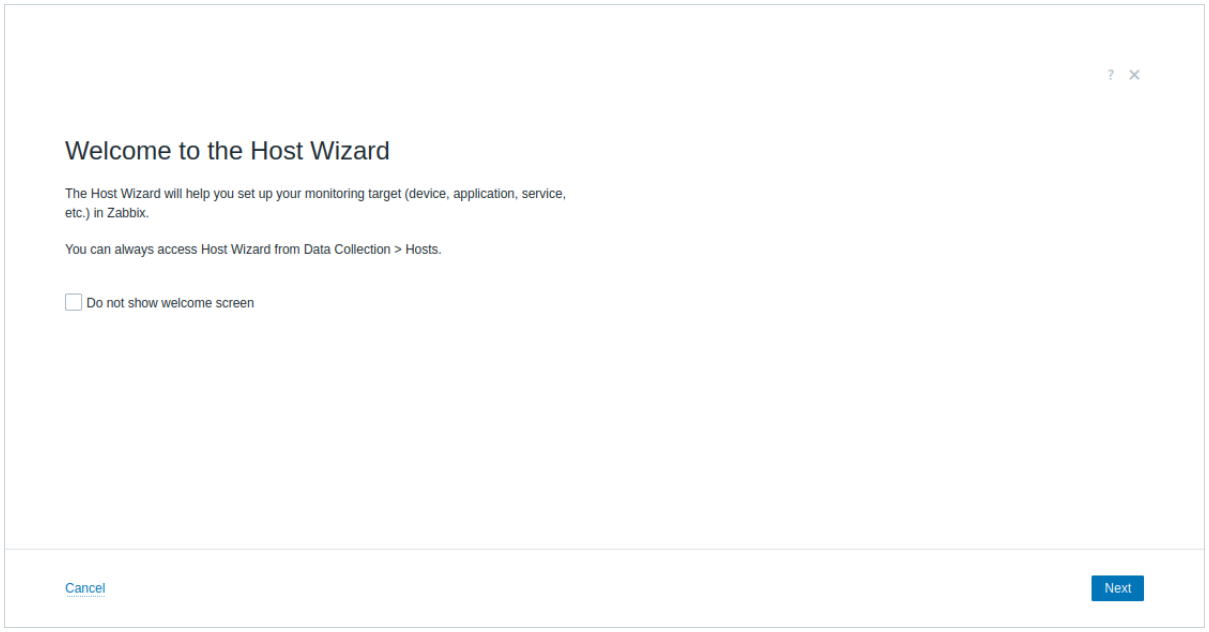
- *Data collection* > *Hosts* に移動します
- 画面右上の *Host Wizard* をクリックします
- ウィザードの指示に従います

また、各種 Web インターフェースセクションで既存のホスト名（またはその横にある三点アイコン）をクリックし、ホストメニューからホストウィザードにアクセスして、既存のホストを編集することもできます。既存のホストを編集する場合、選択したテンプレートで必要な設定がすでに含まれていれば、一部の手順はスキップされることがあります。

または、従来の方法でホストの作成と設定を行うこともできます。

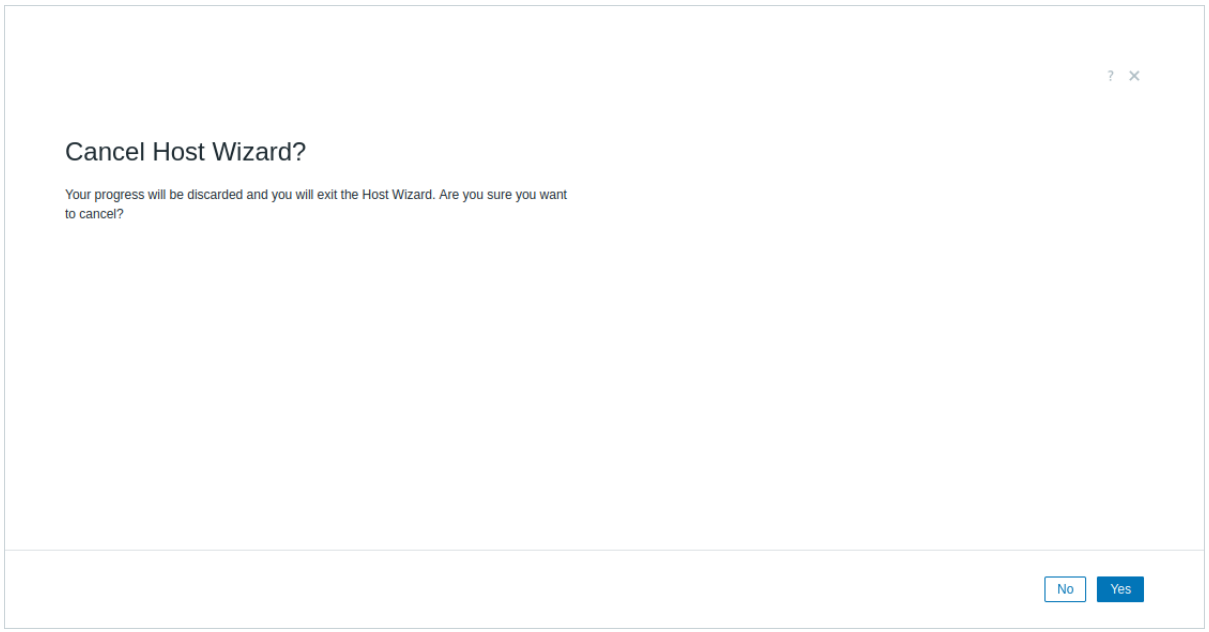
ホストウィザードへようこそ

ホストウィザードを開始すると、ようこそ画面が表示されます。



今後のセッションでこの画面をスキップするには、*Do not show welcome screen* チェックボックスをオンにして、*Next* をクリックします。

次のホストウィザードの手順で設定を開始した後、閉じるアイコンをクリックするか ESC を押してウィザードを閉じようとすると、確認画面が表示されます。



Yes をクリックすると、進行状況を保存せずにホストウィザードを終了します。
No、ESC、または閉じるアイコンをクリックすると、最後の手順に戻ります。

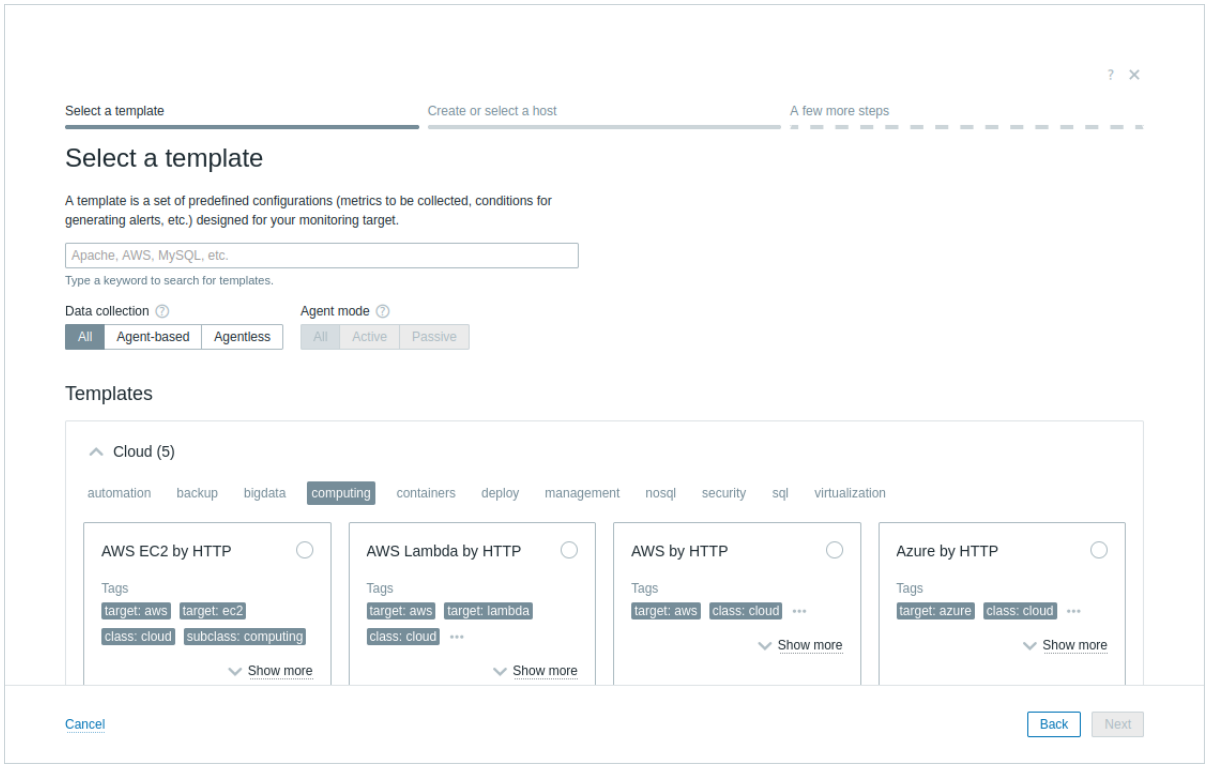
テンプレートを選択

監視対象の設定の最初のステップは、テンプレートを選択することです。テンプレートとは、監視対象向けに設計された、事前定義済みの設定一式（収集するメトリクス、アラートを生成する条件など）です。

Zabbix の標準搭載テンプレートでは、オペレーティングシステム、アプリケーション、データベース、ネットワーク機器、サービスなどに対応した、さまざまな事前定義済みの監視設定が提供されています。

このステップでは、次のことができます。

- テンプレートのtagに基づいて、クラス（Cloud、Database、Network など）およびサブクラス（例: automation、backup）ごとにテンプレートを参照できます。
- テンプレート名、またはテンプレートのタグ名と値に含まれるキーワードでテンプレートを検索できます。
- データ収集方式（エージェントベースまたはエージェントレス）でテンプレートを絞り込めます。エージェントベースのテンプレートには、少なくとも 1 つのZabbix エージェントアイテムが含まれます。
- エージェントモード（アクティブまたはパッシブ）でテンプレートを絞り込めます。アクティブテンプレートには少なくとも 1 つの Zabbix agent (active) アイテムが含まれ、パッシブテンプレートには少なくとも 1 つのパッシブアイテムが含まれます。詳細は、Passive and active agent checksを参照してください。
- Host Wizard を使用して既存のホストを設定する場合は、ホストにすでにリンクされているテンプレートで絞り込むこともできます。



Attention:
テンプレートを選択すると、次のメッセージが表示される場合があります: *Some templates (n) are incompatible with the Host Wizard. See how to update them. Custom templates are not supported.* これは、一部の標準搭載テンプレートがまだ Host Wizard と互換性がなく、アップグレードが必要であることを示しています。詳細は、template upgradeの手順を参照してください。

選択したテンプレートによって、Host Wizard の次のステップが決まります。
たとえば、データ収集に Zabbix エージェントを使用する *MySQL by Zabbix agent* テンプレートを選択した場合、ウィザードはエージェントのインストール手順へ案内します。

ホストを作成または選択

テンプレートは、監視対象を表す Zabbix 内のエンティティであるホストにリンクする必要があります。リンクされると、ホストは監視に必要なすべてのテンプレートエンティティ（収集するメトリクスであるアイテムや、アラートを生成する条件であるトリガーなど）を受け取ります。

各ホストは、少なくとも 1 つのホストグループにも属している必要があります。ホストグループは、ホストの整理や、それらに対するユーザー権限の割り当てに使用されます。

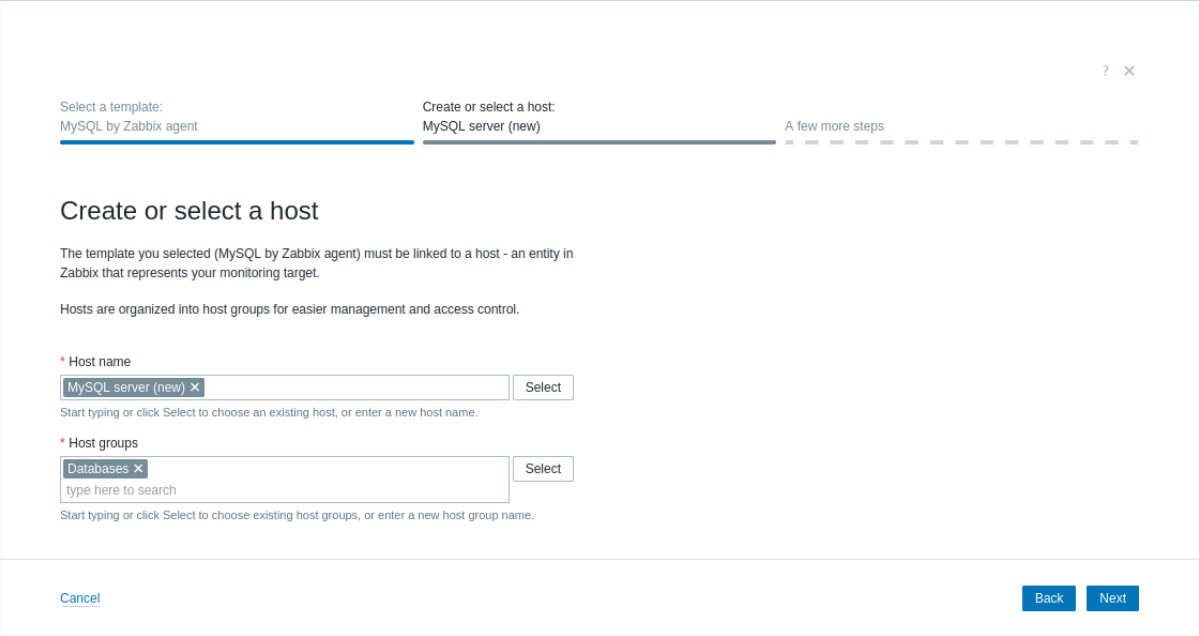
このステップでは、次のことができます。

- ホスト名とホストグループ名の両方を入力して、新しいホストとホストグループを作成する。
- ホスト名を入力して新しいホストを作成し、その新しいホストを割り当てる既存のホストグループを選択する。

- 既存のホストを選択する。この場合、現在所属しているホストグループから削除することなく、このホストを追加のホストグループに割り当てることもできます。

Note:
既存のホストを選択しても、後続のステップで明示的に記載されていない限り、その現在の設定が削除または上書きされることはありません。検出されたホストは選択できません。

たとえば、監視したいローカルにインストールされた MySQL サーバーを表すために、*MySQL server* という名前の新しいホストを作成できます。前のステップで選択した *MySQL by Zabbix agent* テンプレートは、このホストにリンクされます。さらに、既存の *Databases* ホストグループを選択することも、新しく作成することもできます（例: *MySQL servers*）。これにより、このホストを他のデータベース関連ホストと整理し、後で権限管理を簡素化できます。



Zabbix エージェントのインストール

エージェントベースのテンプレート（*MySQL by Zabbix agent* など）では、Zabbix エージェント またはagent 2 のインストールが必要です。これは、監視対象に配置されるプロセスであり、ローカルリソースやアプリケーションを能動的に監視します（テンプレートで必要な場合）。

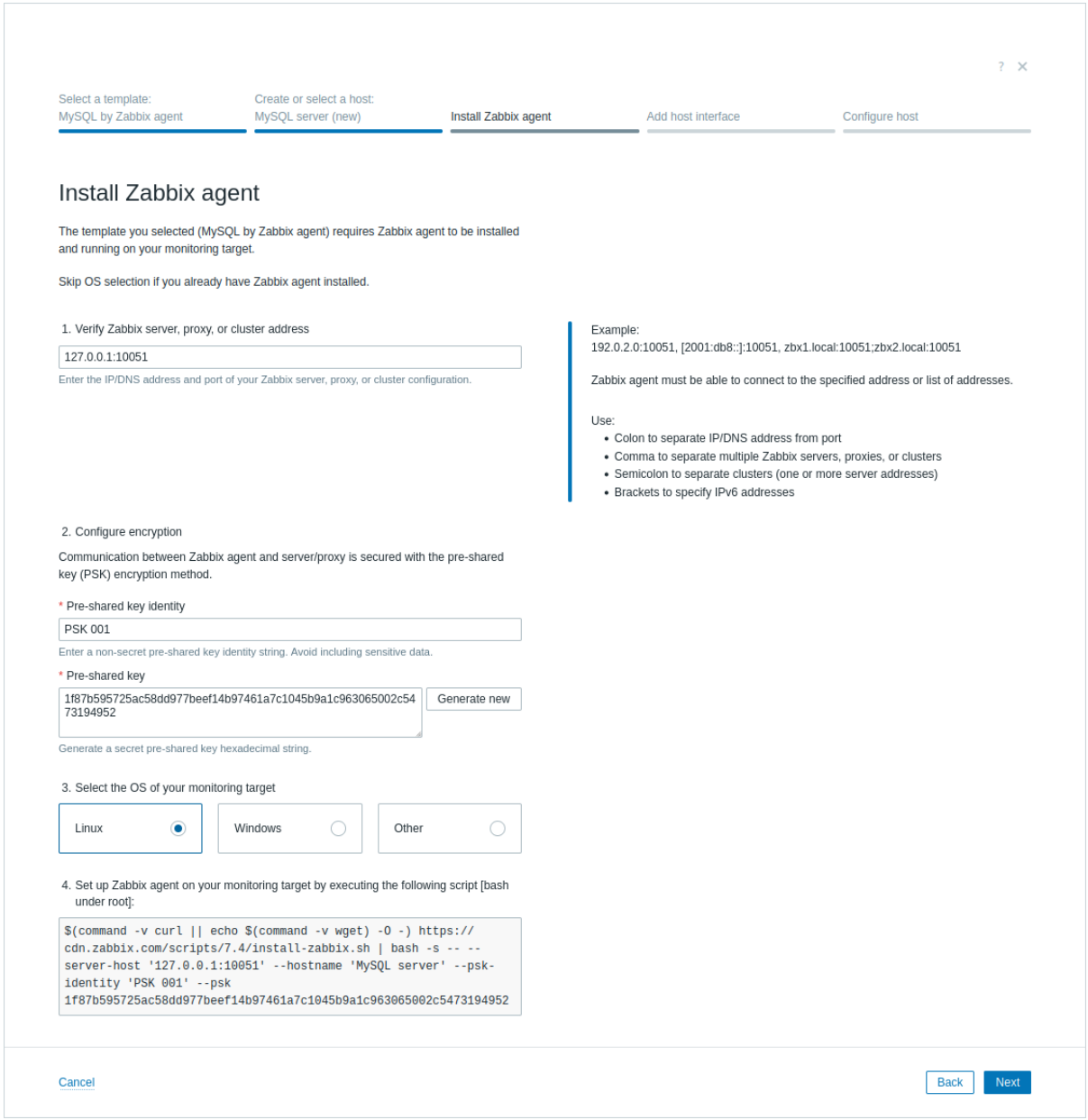
このステップでは、次の操作を行います。

- Zabbix サーバー、プロキシ、またはクラスタのアドレス（例: 192.0.2.0:10051）を確認します。この値は、Zabbix エージェントのServer およびServerActive パラメータの設定に使用されます。
- 新しい一意の、かつ秘密情報ではない事前共有鍵 ID（例: *PSK 001* または *mysql-agent-psk1*）を入力し、事前共有鍵を生成します。これらの値は、Zabbix ホスト（例: *MySQL server*）および Zabbix エージェントの事前共有鍵（PSK） 暗号化の設定に使用されます。

Attention:
既存のホストを編集する場合、PSK 設定によって、そのホスト上の既存の暗号化設定がすべて上書きされます。

- 監視対象をホストしているマシンのオペレーティングシステムを選択します。
- 提供されたスクリプトをそのシステム上で実行するか、インストール手順に従って、そのシステムに Zabbix エージェントをインストールします。

たとえば、*Pre-shared key identity* に *PSK 001* を設定し、新しい *Pre-shared key* を生成できます。次に、オペレーティングシステムとして *Linux* を選択し、そのシステム上で提供されたスクリプトを実行できます。Zabbix エージェントのインストール後、ホストウィザードに戻ります。

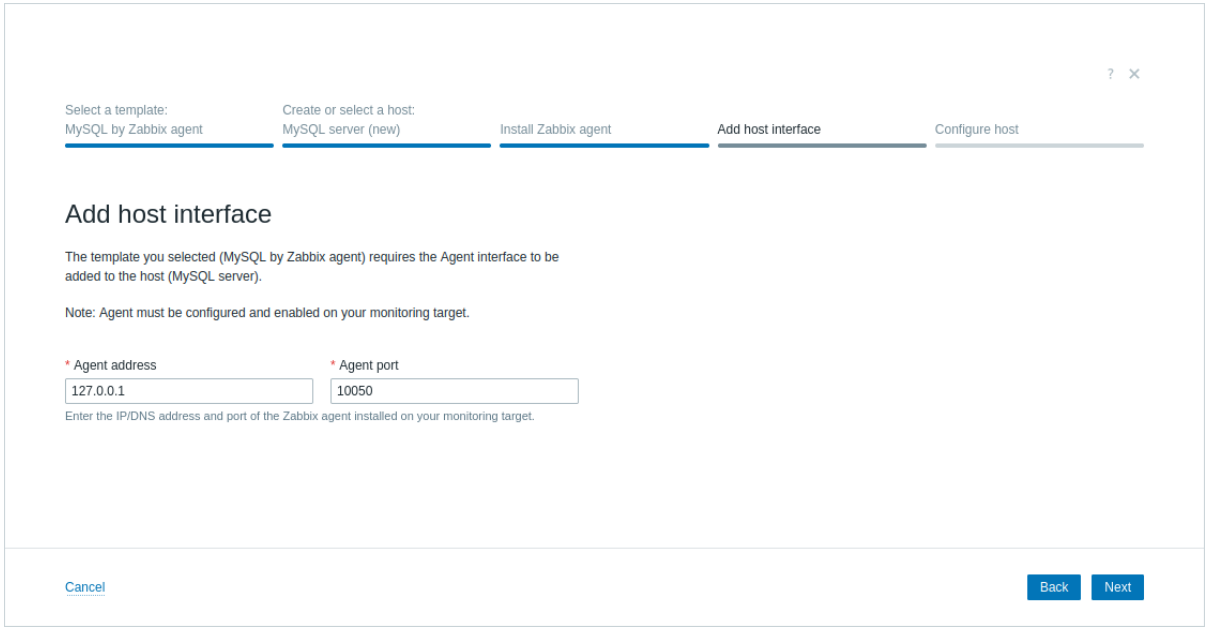


ホストインターフェースの追加

ホストインターフェースは、Zabbix サーバーがネットワーク経由で監視対象にどのように接続するかを定義します。
選択したテンプレートで必要なデータ収集方法に応じて、IP アドレス、DNS 名、ポート、インターフェースタイプ（エージェント、SNMP、JMX、または IPMI）などの接続パラメータを定義します。

このステップでは、選択したテンプレートで必要なホストインターフェースの詳細を入力します。

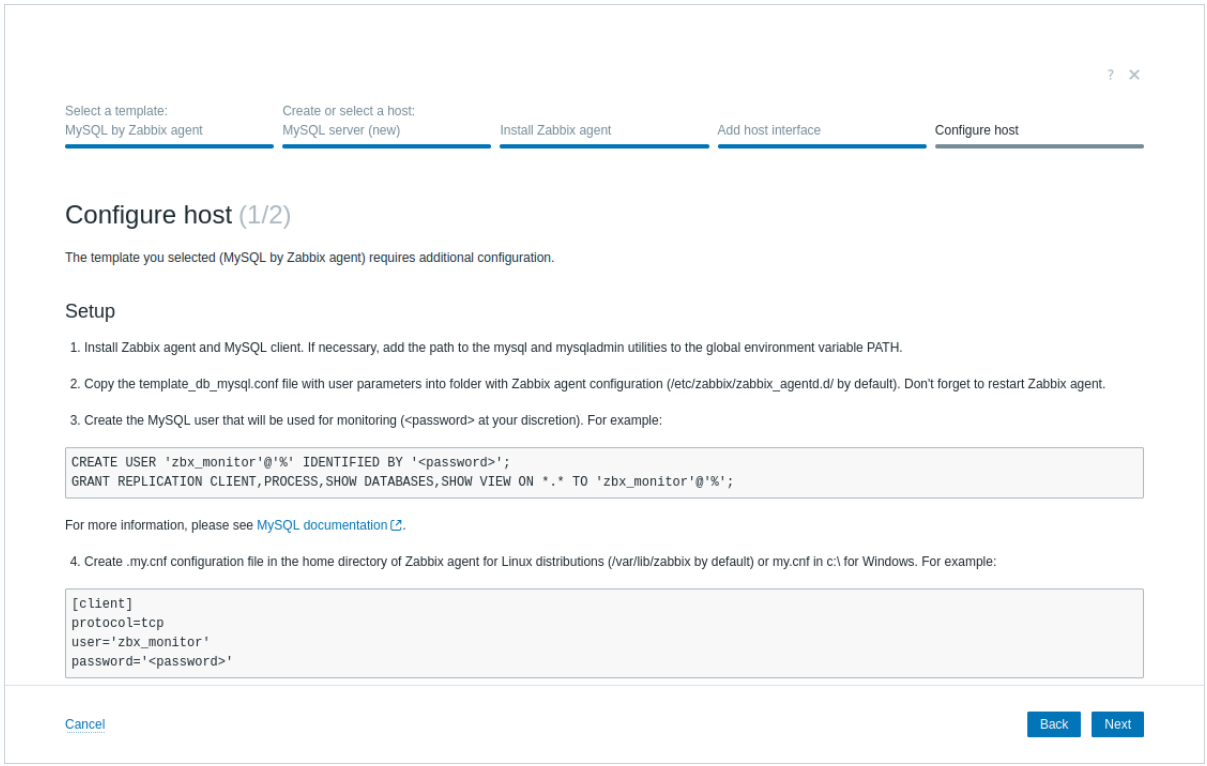
例えば、Zabbix サーバー、Zabbix エージェント、および MySQL サーバーがすべて同じマシン上でローカルに実行されている場合は、デフォルトのエージェントアドレス（127.0.0.1）とポート（10050）を使用できます。



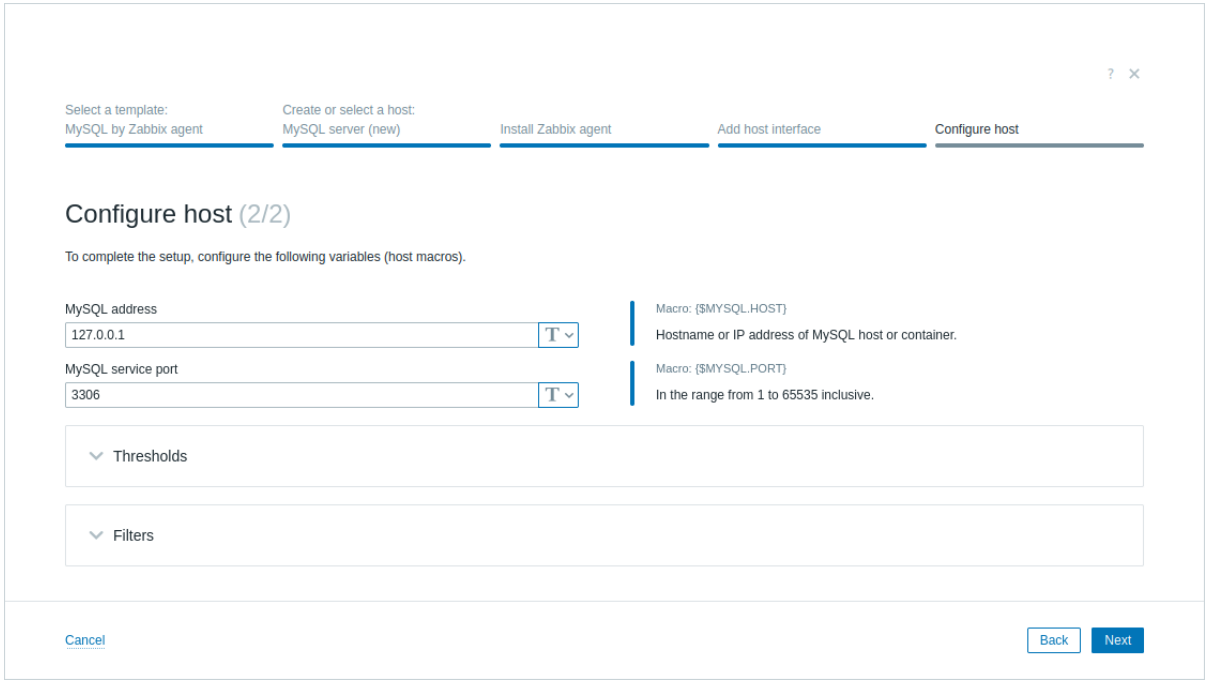
ホストの設定

一部のテンプレートでは、ホストを作成する前に追加の設定が必要です。これには次のようなものがあります。

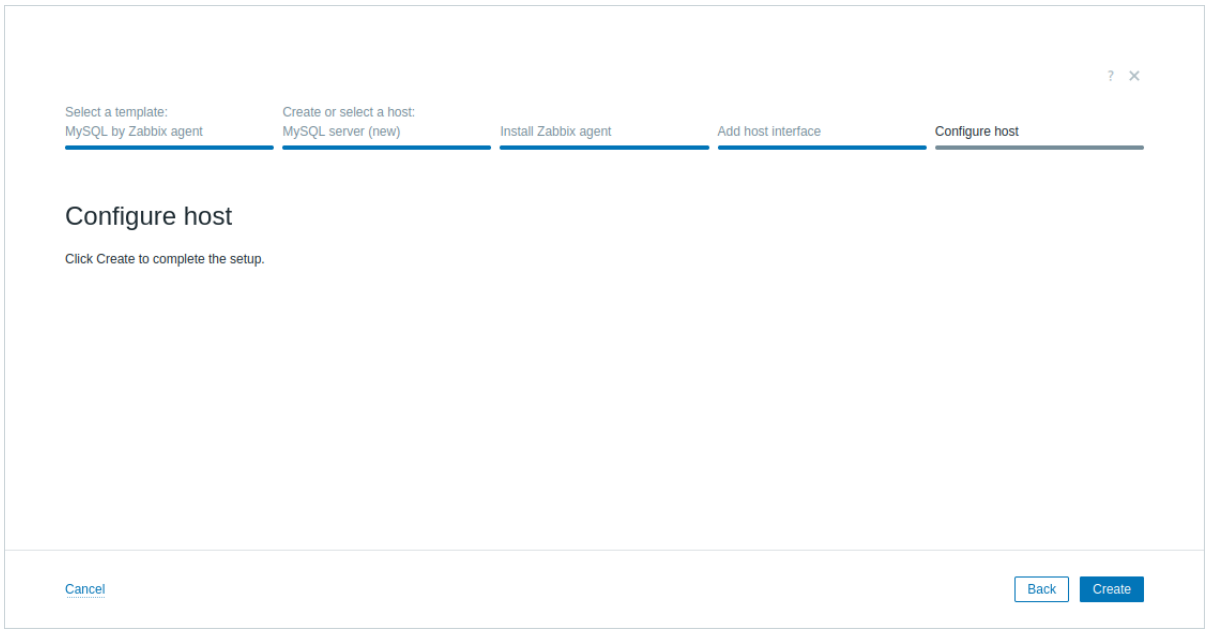
- 監視対象の手動設定（例: 特定のサービスの有効化、サービスユーザーの作成、または権限の付与）:



- ホストマクロの定義—アイテムの動作、接続設定、認証情報、その他のパラメータを制御する変数:



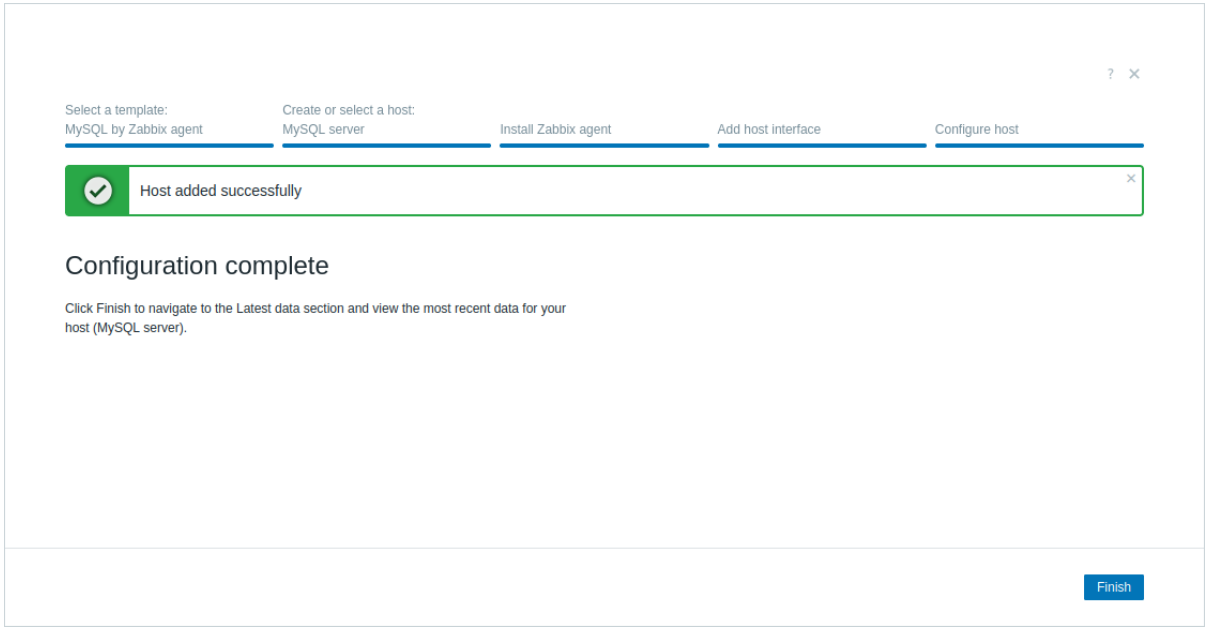
必要な設定手順をすべて完了したら、*Create* をクリックしてホストを Zabbix に追加します（既存のホストの設定を更新する場合は *Update* をクリックします）。



設定完了

この時点で、Zabbix はすでに対象を監視しています。

新しいホストの場合は、*Finish* をクリックして最新データセクションに移動し、ホストの最新データを表示します。
既存のホストの場合は、*Finish* をクリックしてホストウィザードを閉じ、開いた元の画面に戻ります。



2 ホストの設定

概要

Zabbix の Web インターフェースでホストを設定するには、次の手順を実行します。

- データ収集 > ホストまたは 監視 > ホストに移動します
- 画面右上の ホストの作成をクリックします（既存のホストを編集する場合はホスト名をクリックします）
- フォームにホストのパラメータを入力します

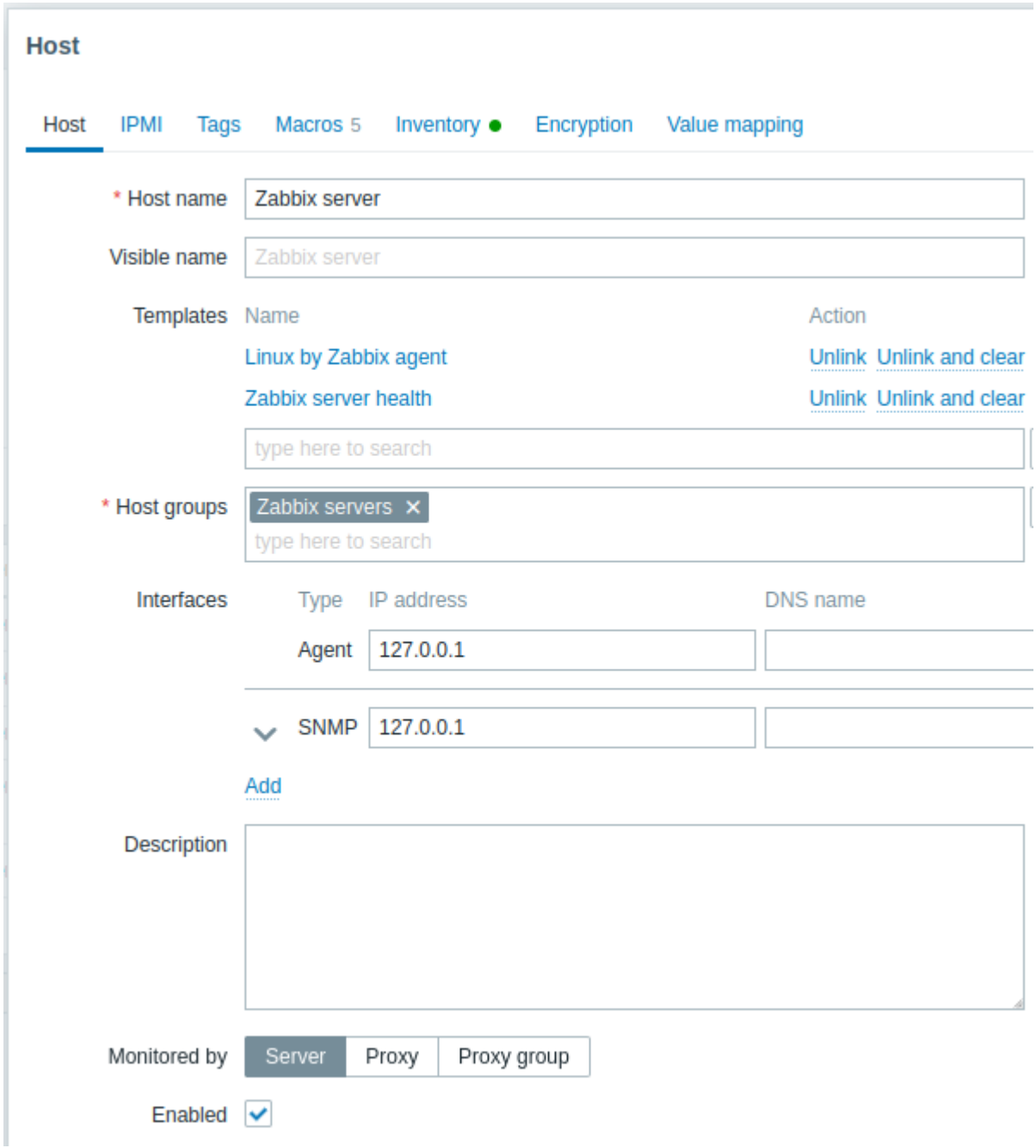
既存のホストの設定フォームにある 複製ボタンを使用して、新しいホストを作成することもできます。このホストには、リンクされたテンプレート、それらのテンプレートに含まれるエンティティ（アイテム、トリガーなど）、および既存のホストに直接関連付けられたエンティティを含め、既存のホストのすべてのプロパティが引き継がれます。

ホストを複製した場合、すべてのテンプレートエンティティはテンプレート上の元の状態のまま保持されることに注意してください。既存のホストレベルでそれらのエンティティに加えられた変更（たとえば、変更されたアイテムの監視間隔、修正された正規表現、またはローレベルディスカバリルールに追加されたプロトタイプ）は、新しいホストには複製されません。代わりに、それらはテンプレート上の状態になります。

または、ホストウィザード を使用して、ガイド付きの段階的なインターフェースでホストを設定することもできます。

設定

ホストタブには、ホストの一般的な属性が含まれています。



必須入力フィールドには、すべて赤いアスタリスクが付いています。

Parameter	Description
Host name	一意のホスト名を入力します。英数字、スペース、ドット、ダッシュ、アンダースコアが使用できます。ただし、先頭および末尾のスペースは使用できません。 *Note:* 設定対象のホスト上で Zabbix エージェントが動作している場合、エージェントの設定ファイルのパラメータ *Hostname* は、ここで入力したホスト名と同じ値である必要があります。パラメータ内の名前は、アクティブチェックの処理に必要です。
Visible name	ホストの一意の表示名を入力します。この名前を設定すると、技術的なホスト名の代わりに、リストやマップなどにはこの名前が表示されます。この属性は UTF-8 をサポートしています。

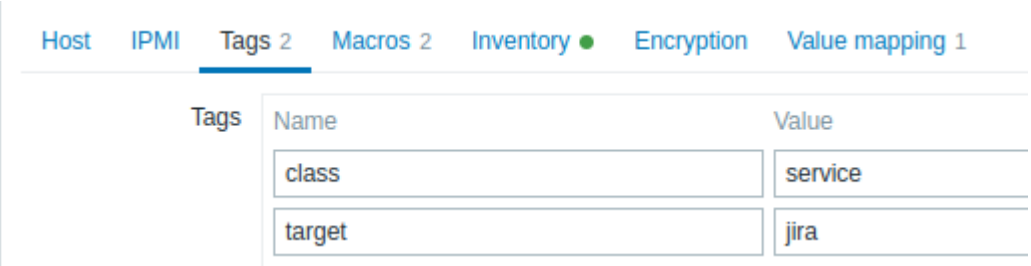
Parameter	Description
Templates	ホストにテンプレートをリンクします。すべてのエンティティ（アイテム、トリガー、その他）はテンプレートから継承されます。 新しいテンプレートをリンクするには、テキスト入力フィールドにテンプレート名の入力を開始します。一致するテンプレートの一覧が表示されるので、スクロールして選択します。あるいは、フィールドの横にある *Select* をクリックし、ポップアップウィンドウの一覧からテンプレートを選択することもできます。選択したすべてのテンプレートは、ホスト設定フォームを保存または更新したときにホストへリンクされます。 テンプレートのリンクを解除するには、*Linked templates* ブロック内の次の 2 つのオプションのいずれかを使用します。 *Unlink* - テンプレートのリンクを解除しますが、そのエンティティ（アイテム、トリガー、その他）は保持します。 *Unlink and clear* - テンプレートのリンクを解除し、そのすべてのエンティティ（アイテム、トリガー、その他）を削除します。 一覧に表示されるテンプレート名はクリック可能なリンクで、テンプレート設定フォームに移動します。
Host groups	ホストが属するホストグループを選択します。ホストは少なくとも 1 つのホストグループに属している必要があります。存在しないグループ名を追加することで、新しいグループを作成してホストにリンクできます。
Interfaces	ホストでは複数のホストインターフェースタイプがサポートされています: *Agent*、*SNMP*、*JMX*、*IPMI*。 デフォルトではインターフェースは定義されていません。新しいインターフェースを追加するには、*Interfaces* ブロック内の *Add* をクリックし、インターフェースタイプを選択して、*IP/DNS*、*Connect to*、*Port* の情報を入力します。 *Note:* いずれかのアイテムで使用されているインターフェースは削除できず、それらの *Remove* リンクはグレー表示になります。 SNMP インターフェースの”IP” または”DNS” は、SNMP トラップにも使用されます。 マッチング時には、ホストインターフェースで選択された”IP” または”DNS” のみが使用されます。 SNMP インターフェース（v1、v2、v3）の設定に関する追加の詳細は、SNMP 監視の設定を参照してください。
IP address	ホストの IP アドレス（任意）。
DNS name	ホストの DNS 名（任意）。
Connect to	対応するボタンをクリックすると、Zabbix サーバーがエージェントからデータを取得する際に何を使用するかを指定できます。 **IP** - ホストの IP アドレスに接続します（推奨） **DNS** - ホストの DNS 名に接続します
Port	TCP/UDP ポート番号。デフォルト値は、Zabbix エージェントは 10050、SNMP エージェントは 161、JMX は 12345、IPMI は 623 です。
Default	ラジオボタンをチェックしてデフォルトインターフェースを設定します。
Description	ホストの説明を入力します。
Monitored by	ホストの監視元を選択します。 **Server** - ホストは Zabbix サーバーによって監視されます。 **Proxy** - ホストはスタンドアロンのプロキシによって監視されます。 **Proxy group** - ホストはプロキシグループによって監視されます。
Proxy	割り当てられたプロキシ名が表示されます（Zabbix サーバーが選択したプロキシグループからプロキシを割り当てた場合のみ）。 このフィールドは、ホストがプロキシグループによって監視されている場合にのみ表示されます。

Parameter	Description
Enabled	チェックボックスがオンの場合、ホストは有効化され、監視可能な状態になります。 チェックボックスがオフの場合、ホストは無効化され、監視されません。 Zabbix サーバー/プロキシによって開始されるパッシブデータ要求（たとえば、Zabbix エージェント、SNMP エージェント、シンプルチェック）では、設定の同期後に監視が無効になります。ホストにリンクされたトリガーとアクションも、設定が再読み込みされた後にのみ無効になります。 Zabbix エージェントのアクティブチェックでは、Zabbix エージェントがホストの無効化情報を受信してからの時間枠内（約 5 秒）で監視が停止します。この短い間隔の間、ホストは引き続きアクティブチェック用のデータをローカルで収集し、サーバー/プロキシへの送信を試みますが、ホストは *Disabled* としてマークされているため、サーバー/プロキシはそのデータを拒否します。 ホストを無効化すると、そのアイテムは履歴キャッシュから即座に削除されます（ただし、ログ用に保持される最新値は除きます）。

IPMI タブには、IPMI 管理属性が含まれています。

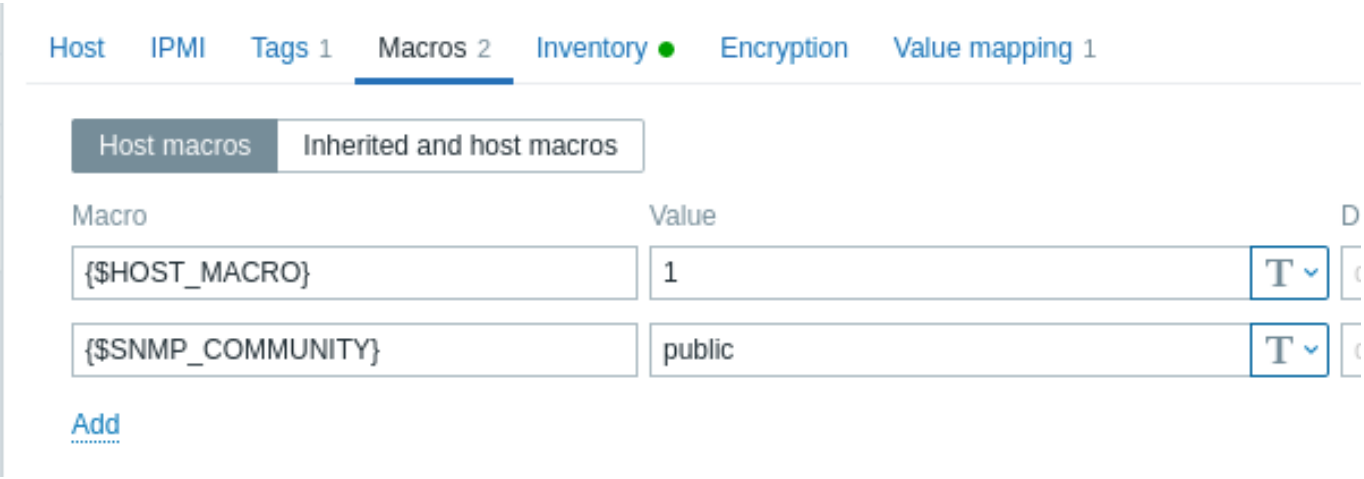
Parameter	Description
Authentication algorithm	認証アルゴリズムを選択します。
Privilege level	特権レベルを選択します。
Username	認証用のユーザー名です。ユーザーマクロを使用できます。
Password	認証用のパスワードです。ユーザーマクロを使用できます。

Tags タブでは、ホストレベルのタグを定義できます。このホストのすべての障害には、ここで入力した値のタグが付けられます。

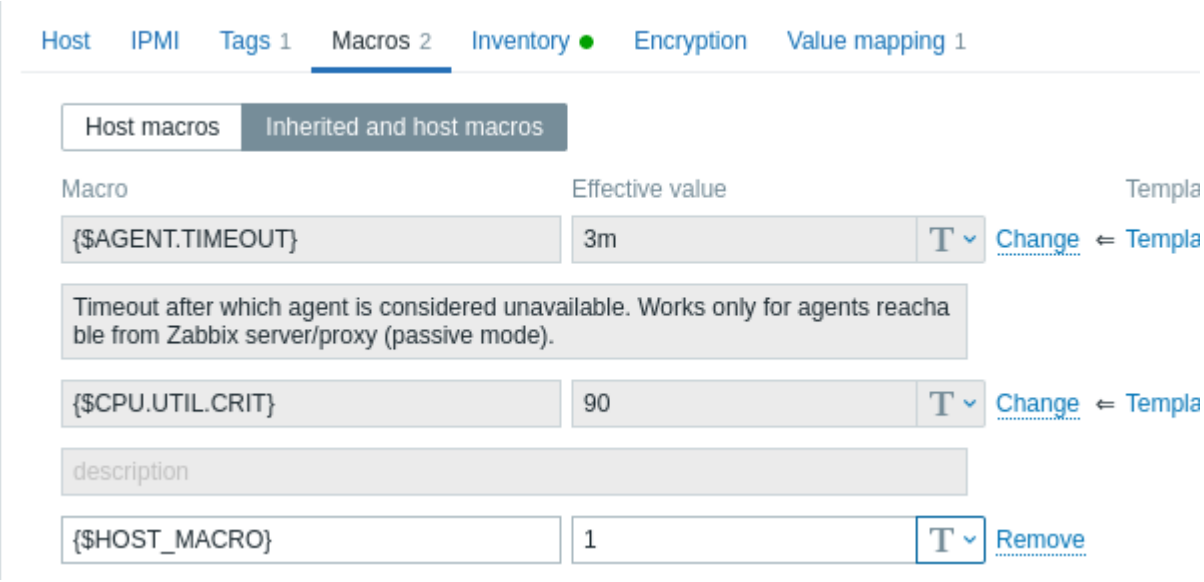


タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。

Macros タブでは、ホストレベルのユーザーマクロを名前と値のペアとして定義できます。マクロ値はプレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できることに注意してください。説明の追加もサポートされています。

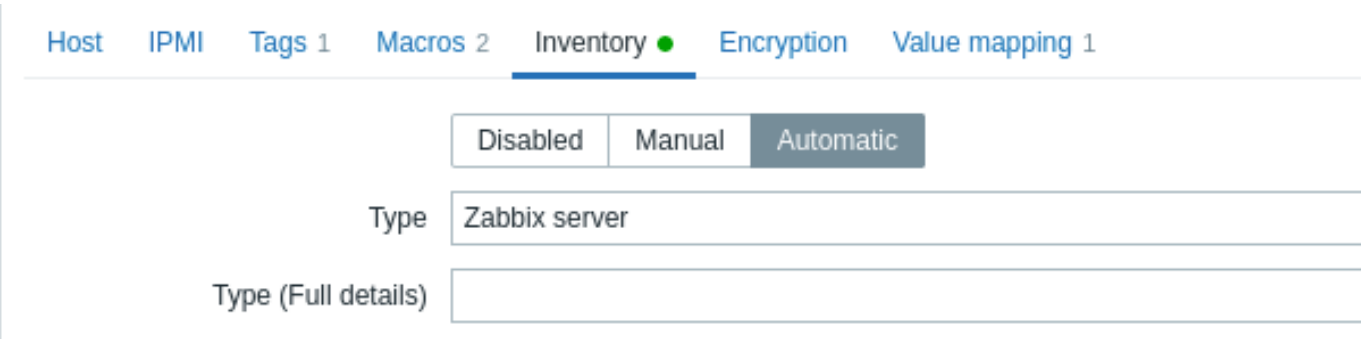


また、*Inherited and host macros* オプションを選択すると、テンプレートレベルおよびグローバルのユーザーマクロもここで表示できます。ここには、ホストに対して定義されたすべてのユーザーマクロが、解決後の値およびその由来とともに表示されます。



利便性のため、対応するテンプレートおよびグローバルマクロ設定へのリンクも提供されています。また、ホストレベルでテンプレート/グローバルマクロを編集することも可能で、その場合は実質的にそのマクロのコピーがホスト上に作成されます。

Inventory タブでは、ホストのインベントリ情報を手動で入力できます。また、*Automatic* インベントリ登録を有効にするか、このホストのインベントリ登録を無効にするかを選択することもできます。



インベントリが有効（手動または自動）な場合、タブ名の横に緑色の点が表示されます。

暗号化

暗号化タブでは、ホストと暗号化された接続をするように要求できます。

パラメーター	説明
ホストへの接続	Zabbix サーバーまたはプロキシがホスト上の Zabbix エージェントに接続する方法: 暗号化なし (デフォルト)、PSK(事前共有キー) または証明書を使用。
ホストからの接続	ホスト (Zabbix エージェントおよび Zabbix sender) から許可される接続の種類を選択します。複数の接続タイプを同時に選択できます (テストや他の接続タイプへの切り替えに便利です)。デフォルトは” 暗号化なし” です。
発行者	許可された証明書の発行者。証明書は最初に CA (証明機関) で検証されます。証明書が有効で、CA によって署名されている場合は、発行者フィールドを使用して、許可される CA をさらに制限できます。このフィールドは、Zabbix インストールで複数の CA からの証明書を使用する場合に使用することを目的としています。このフィールドが空の場合、どの CA でも受け入れられます。
サブジェクト	許可された証明書の件名。証明書は最初に CA で検証されます。証明書が有効で、CA によって署名されている場合、サブジェクトフィールドを使用してサブジェクト文字列の値を 1 つだけ許可できます。このフィールドが空の場合、構成された CA によって署名された有効な証明書はすべて受け入れられます。
PSK アイデンティティ	事前共有キー ID 文字列。 PSK ID には機密情報を入れないでください。これは、受信者にどの PSK を使用するかを通知するために、ネットワーク経由で暗号化されずに送信されます。
PSK	事前共有キー (16 進文字列)。最大長:Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 桁の 16 進数 (256 バイトの PSK)、Zabbix が mbed TLS(PolarSSL) ライブラリを使用する場合は 64 桁の 16 進数 (32 バイトの PSK)。例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

値のマッピング

値のマッピングタブでは、値のマッピングを使用してアイテムの値を分かりやすい表現で表示できるよう設定できます。

3 ホストグループの設定

概要

ホストグループは、ホストを論理的にグループ化し、それらに対するユーザー権限を割り当てるために使用されます。

各ホストには、少なくとも 1 つのホストグループが割り当てられている必要があります。1 つのホストは複数のホストグループに属することができ、各ホストグループには複数のホストを含めることができます。

Zabbix では、すべての権限はグループに基づいていることに注意してください: ユーザーグループ、ホストグループ、およびテンプレートグループです。したがって、1 人のユーザーが 1 つのホストのみにアクセスする必要がある場合でも、そのホストを含むホストグループへのアクセス権を持つユーザーグループにそのユーザーを追加することで権限が付与されます。

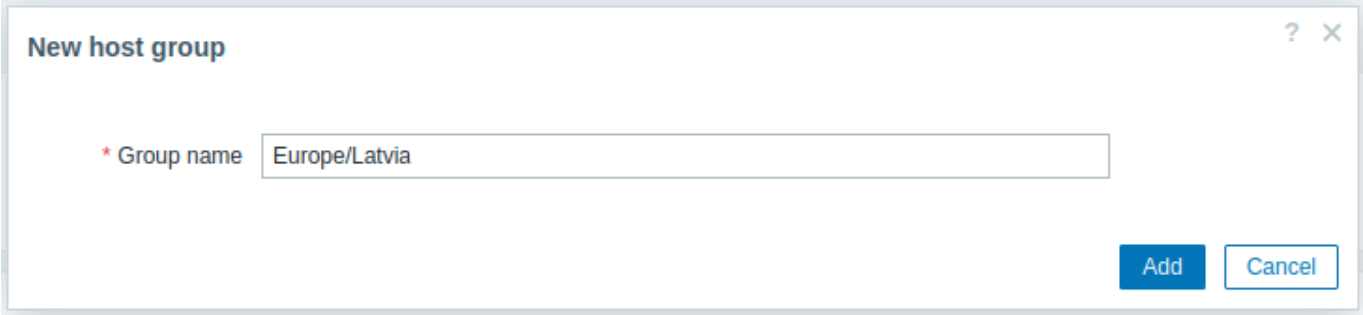
設定

Attention:
ホストグループを作成できるのは、Super admin ユーザーのみです。

Zabbix の Web インターフェースでホストグループを作成する方法は 2 つあります。

1 つ目の方法:

- データ収集 → ホストグループに移動します
- 画面右上の ホストグループの作成をクリックします
- フォームにグループ名を入力します



2 つ目の方法: ホストの設定時に、ホストグループ入力フィールドに存在しないグループ名を入力します。

ホストグループが作成されると、データ収集 → ホストグループの一覧でグループ名をクリックして、グループ名の編集、グループの複製、またはグループの削除を行えます。

ホストグループを削除しても、削除されるのは論理グループのみであり、グループ内のホストは削除されません。既存のホストにとって唯一のグループであるホストグループは削除できません。

ホストサブグループの作成

ホストサブグループ（またはネストされたホストグループ）は、それを含む親ホストグループの子です。

サブグループは、グループ名の入力フィールドで親グループとの関係を示すためにスラッシュ'/' を使用して作成します。例えば、次のとおりです。

- Europe/Latvia と入力すると、親グループ Europe のサブグループ Europe/Latvia が作成されます。
- Europe/Latvia/Riga/Zabbix servers と入力すると、ネストされた親グループ Europe、Europe/Latvia、Europe/Latvia/Riga 内に対応するサブグループが作成されます。

サブグループの作成時に、先頭または末尾にスラッシュを使用したり、スラッシュを連続して使用したりすることはできません。'/' のエスケープはサポートされていません。

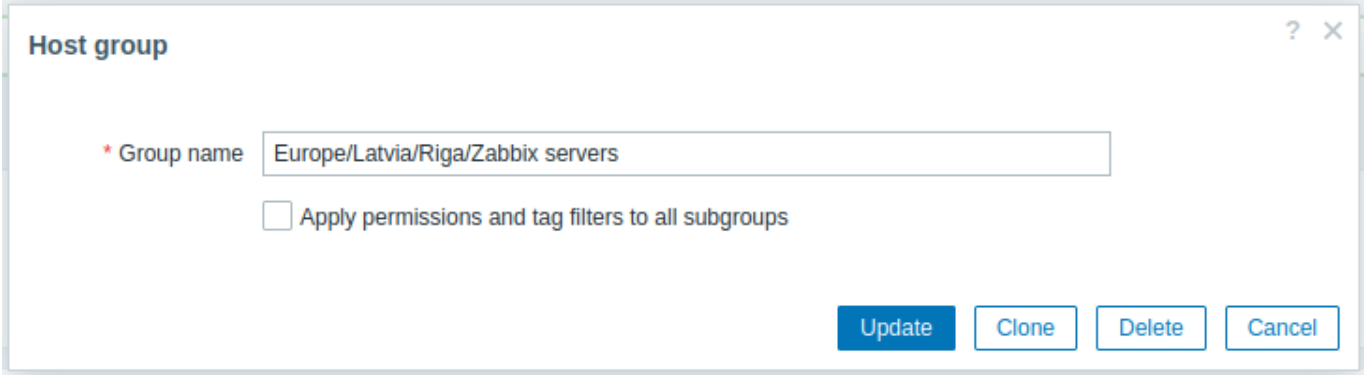
サブグループを作成する前に、親ホストグループを作成しておく必要はありません。サブグループ（例えば Europe/Latvia）の作成から始めることも、親ホストグループ（この例では Europe）の作成から始めることもできます。サブグループの作成から始めた場合、親ホストグループは自動的には作成されません。

ホストグループに対する権限

- 既存の親ホストグループにサブグループを作成する場合（例えば、Europe がすでに存在する状態で Europe/Latvia を作成する場合）、サブグループに対するユーザーグループの権限は親から継承されます。

- 既存のサブグループに対して親ホストグループを作成する場合（例えば、Europe/Latvia がすでに存在する状態で Europe を作成する場合）、親に対する権限は設定されません。

ホストグループを編集する際には、追加オプションとして すべてのサブグループに権限とタグフィルターを適用を設定することもできます。



このチェックボックスをオンにして 更新をクリックすると、編集中のホストグループの現在および将来のすべてのサブグループに、同じ権限レベルとタグフィルターが適用されます。

したがって、編集中のホストグループのサブグループに対して、いずれかのユーザーグループに異なる権限が付与されている場合、このチェックボックスをオンにすると、現在および将来のすべてのサブグループに、編集中のグループと同じユーザー権限およびタグベースの権限が付与されます。

このオプションはデータベースには保存されず、既存の権限を上書きすることに注意してください。このオプションによって行われた変更は、手動でのみ元に戻すことができます。

4 インベントリ

概要

Zabbix でネットワークデバイスのインベントリを管理することができます。

Zabbix フロントエンドには特別なインベントリメニューがあります。しかし、初期状態ではそこにデータは表示されず、データを入力する場所でもありません。インベントリデータの構築は、ホストを設定するときに手動で行うか、自動配置オプションを使用して自動的に行われます。

インベントリの構築

手動モード

ホストを設定するときに、インベントリタブで、デバイスのタイプ、シリアル番号、場所、担当者、URL などの詳細(インベントリ情報に入力されるデータ) を入力できます。

ホストインベントリ情報に URL が含まれており、その URL が'http' または'https' で始まる場合、インベントリセクションにクリック可能なリンクが表示されます。

自動モード

ホストのインベントリも自動的に入力できます。これを機能させるには、インベントリタブでホストのインベントリモードを設定するときに自動に設定する必要があります。

次に、ホストアイテムを設定して、ホストインベントリフィールドに値を付けるようにアイテム設定のそれぞれの属性を持つ宛先フィールドを示すことができます (呼び出されたアイテムはホストインベントリフィールドに入力されます)。

インベントリデータの自動収集に特に有効なアイテムを収集することができます。

- system.hw.chassis[full|type|vendor|model|serial] - デフォルトは [full] で、ルート権限が必要
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - デフォルトは [all,full]
- system.hw.devices[pci|usb] - デフォルトは [pci]
- system.hw.macaddr[interface,short|full] - デフォルトは [all,full]、インターフェースは正規表現
- system.sw.arch
- system.sw.os[name|short|full] - デフォルトは [name]
- system.sw.packages[package,manager,short|full] - デフォルトは [all,all,full]

インベントリモードの選択

インベントリモードは、ホスト設定フォームで選択することができます。

新しいホストのデフォルトのインベントリモードは、管理 → 一般設定 → その他のデフォルトのホストインベントリモード設定に基づいて選択されます。

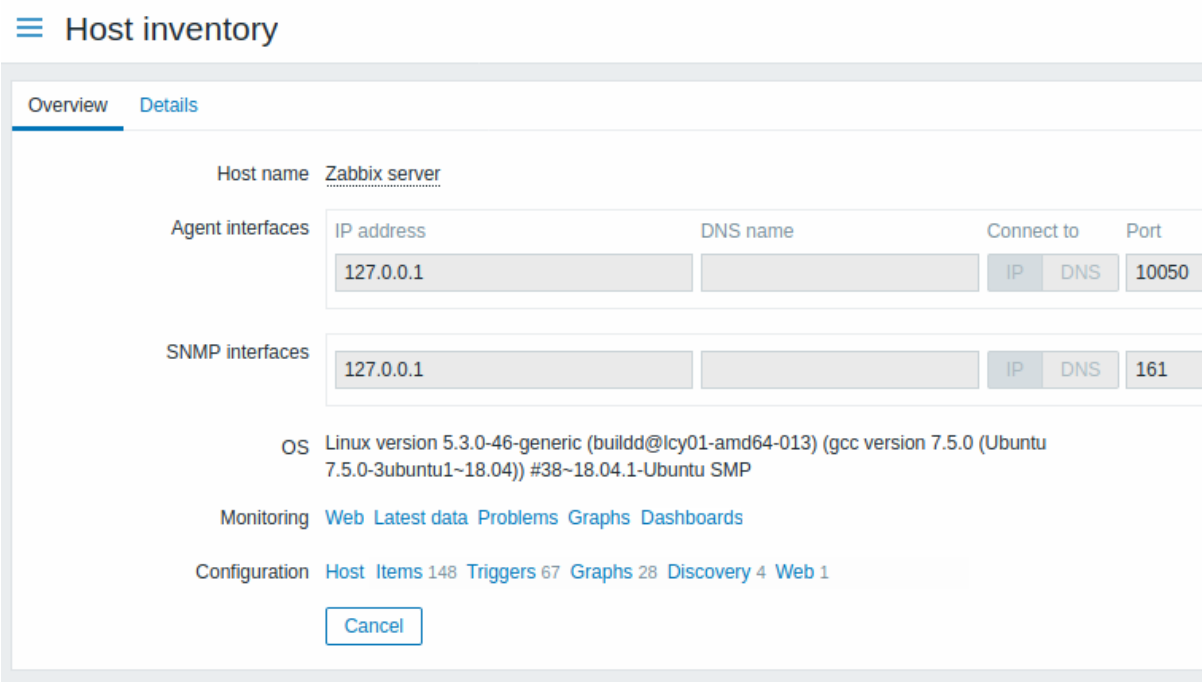
ネットワークディスカバリまたは自動登録アクションによって追加されたホストの場合、ホストインベントリモードの設定操作で手動モードまたは自動モードを選択できます。この操作はデフォルトのホストインベントリモード設定を上書きします。

インベントリ概要

既存のすべてのインベントリデータの詳細は、インベントリメニューに表示されます。

インベントリ → 概要で、インベントリの様々なフィールドによるホスト数を取得できます。

インベントリ → ホストで、インベントリ情報を持つすべてのホストを見ることができます。ホスト名をクリックすると、インベントリの詳細が表示されます。



概要タブは以下を表示します。

パラメーター	説明
ホスト名	ホストの名前。 名前をクリックすると、ホストに定義されているスクリプトを含むメニューが開きます。 ホストがメンテナンス中の場合、ホスト名はオレンジ色のアイコンとともに表示されます。
表示名	ホストの表示名 (定義されている場合)。
ホスト (エージェント、*SNMP*、*JMX*、*IPMI) *インターフェース	ホストに設定されたインターフェースの詳細を提供します。
OS	ホストのオペレーティングシステムのインベントリフィールド (定義されている場合)。
ハードウェア	ホストのハードウェアのインベントリフィールド (定義されている場合)。
ソフトウェア	ホストのソフトウェアのインベントリフィールド (定義されている場合)。
説明	ホストの説明。
監視データ	このホストのデータを含む監視セクションへのリンク: *Web*、最新データ、障害、グラフ、ダッシュボード。
設定	このホストの設定セクションへのリンク: ホスト、アイテム、トリガー、グラフ、ディスカバリ、*Web* 設定されたエンティティの数は、各リンクの後に表示されます。

詳細タブには、入力されている (空ではない) すべてのインベントリフィールドが表示されます。

インベントリマクロ

通知用にホストインベントリマクロ {INVENTORY.} が用意されています。例えば、

”{INVENTORY.LOCATION1} のサーバーに障害が発生しました。責任者は {INVENTORY.CONTACT1}、電話番号は {INVENTORY.POC.PRIMARY.PHONE.A1}。”

詳細は、サポートしているマクロページをご覧ください。

5 一括更新

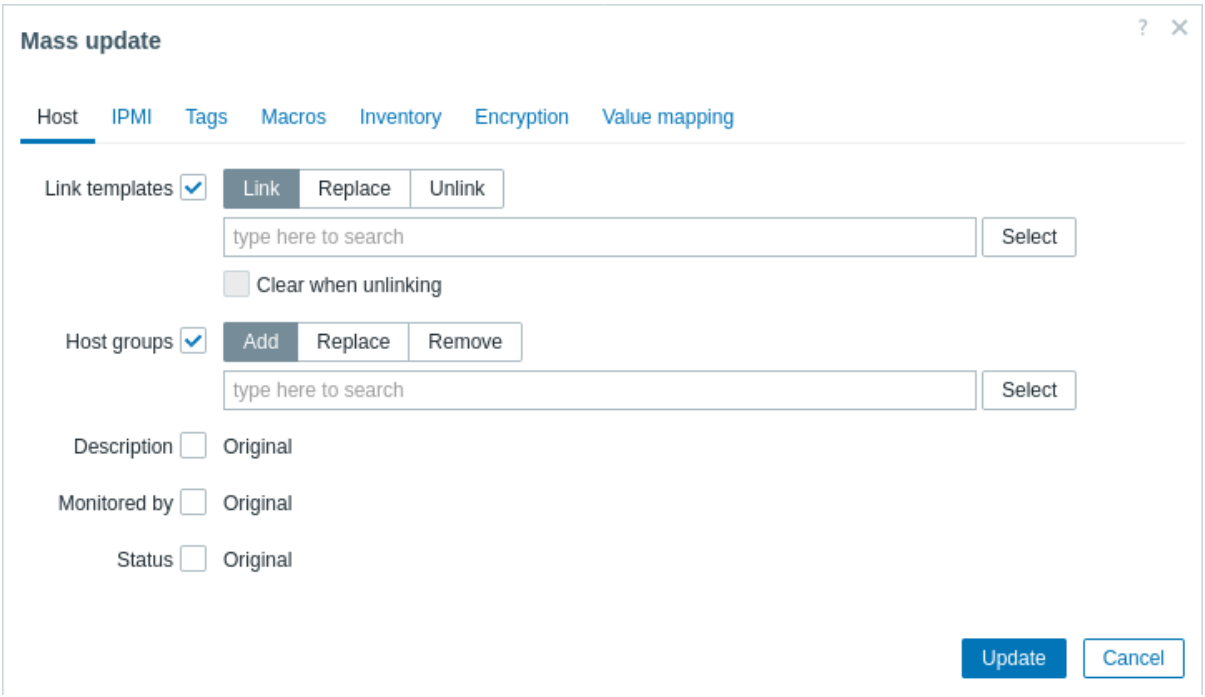
概要

一度に複数のホストの属性を変更したい場合があります。個々のホストを開いて編集する代わりに一括更新機能を使います。

一括更新の使用

複数のホストを一括更新するには、次の手順を実行します。

- ホストリストで、更新したいホストの前にあるチェックボックスを選択します
- リストの下にある 一括更新をクリックします
- 必要な属性のタブ（ホスト、*IPMI*、タグ、マクロ、インベントリ、暗号化、または 値マッピング）に移動します
- 更新する属性のチェックボックスを選択し、新しい値を入力します



テンプレートのリンク更新で該当するボタンを選択した場合、次のオプションを利用できます。

- リンク - 追加でリンクするテンプレートを指定します
- 置換 - リンクするテンプレートを指定し、以前からホストにリンクされていたテンプレートをすべて解除します
- リンク解除 - リンク解除するテンプレートを指定します

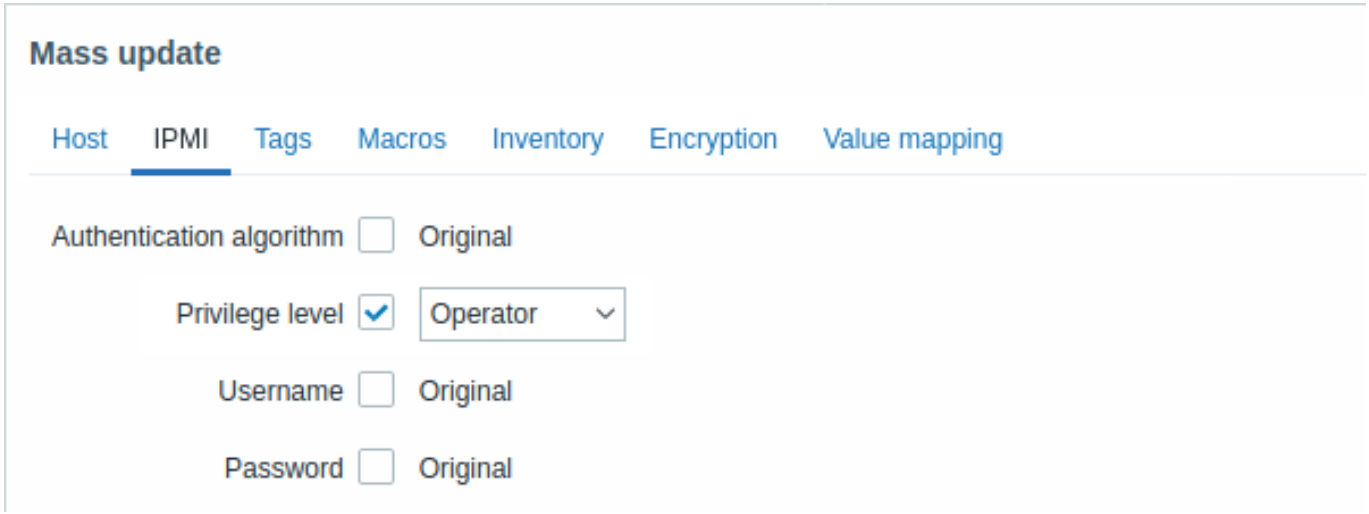
リンク/リンク解除するテンプレートを指定するには、自動補完フィールドにテンプレート名を入力し始めてください。一致するテンプレートを候補として表示するドロップダウンが現れます。必要なテンプレートを選択するには、スクロールダウンしてください。

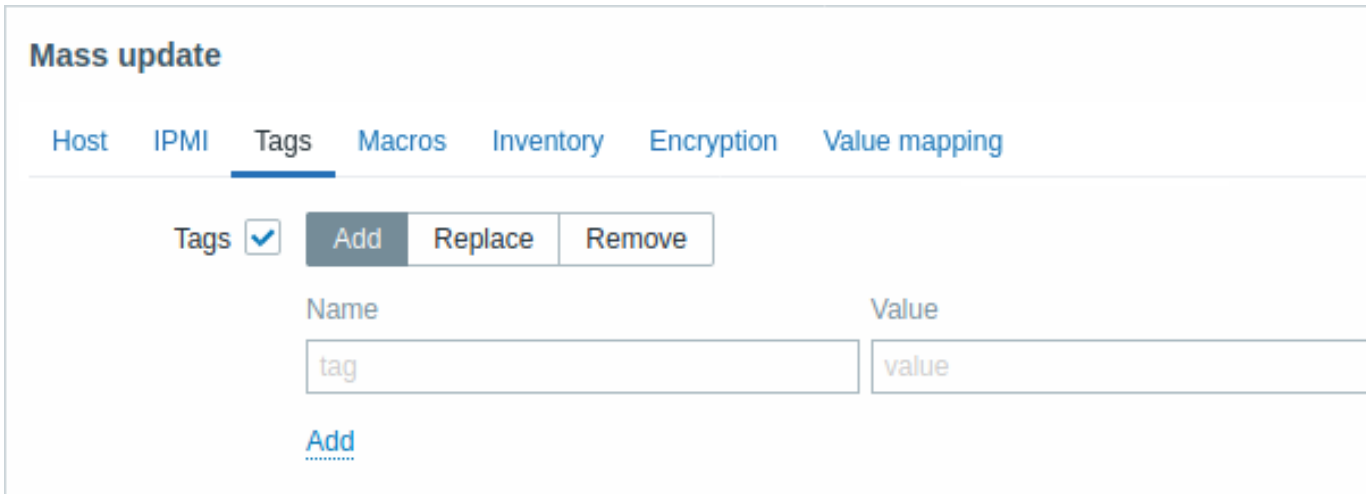
リンク解除時にクリアオプションを使用すると、以前リンクされていたテンプレートのリンクを解除するだけでなく、それらから継承されたすべての要素（アイテム、トリガーなど）も削除できます。

ホストグループの更新で該当するボタンを選択した場合、次のオプションを利用できます。

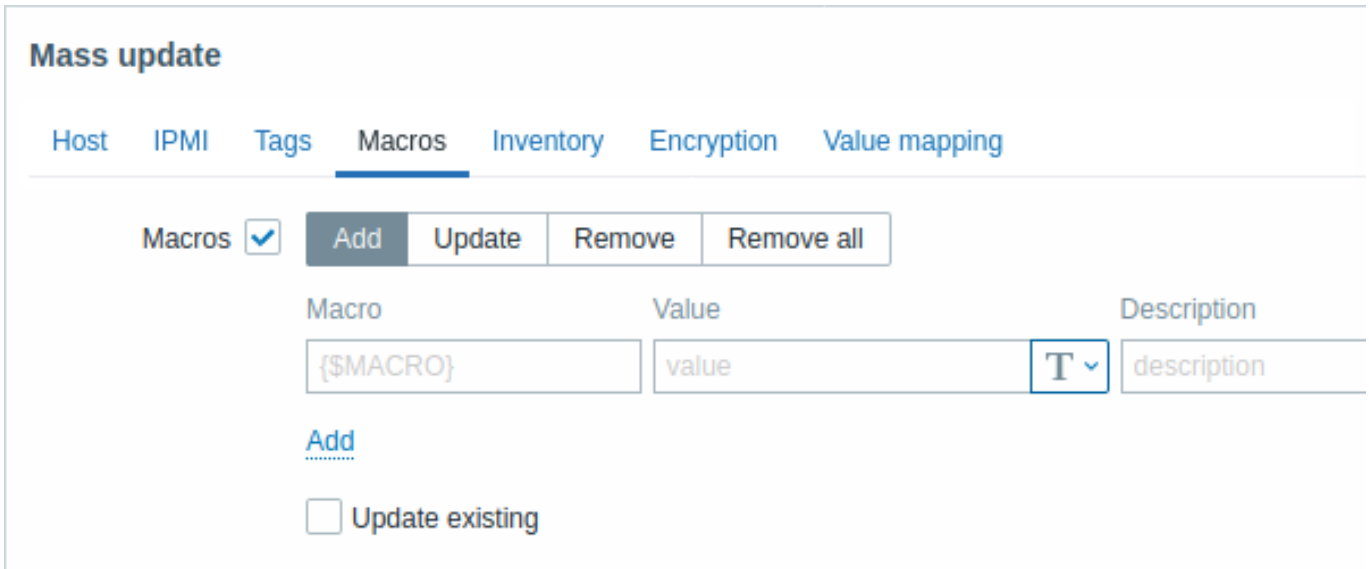
- 追加 - 既存のホストグループから追加のホストグループを指定するか、ホスト用にまったく新しいホストグループを入力できます
- 置換 - ホストを既存のすべてのホストグループから削除し、このフィールドで指定したホストグループ（既存または新規）に置き換えます
- 削除 - ホストから特定のホストグループを削除します

これらのフィールドは自動補完に対応しています。入力を開始すると、一致するホストグループのドロップダウンが表示されます。ホストグループが新規の場合もドロップダウンに表示され、文字列の後ろに *(new)* と示されます。必要なものを選択するには、スクロールダウンしてください。





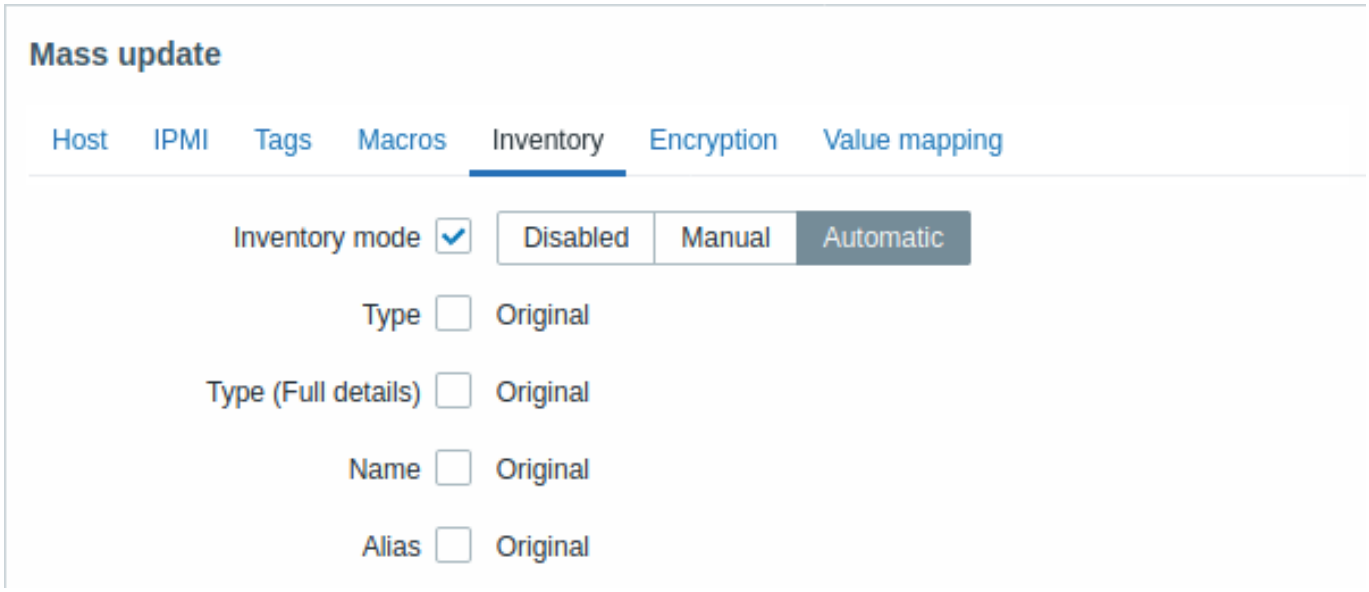
タグでは、ユーザーマクロ、{INVENTORY.*} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、{HOST.ID} マクロがサポートされています。同じ名前で値が異なるタグは「重複」とは見なされず、同じホストに追加できます。



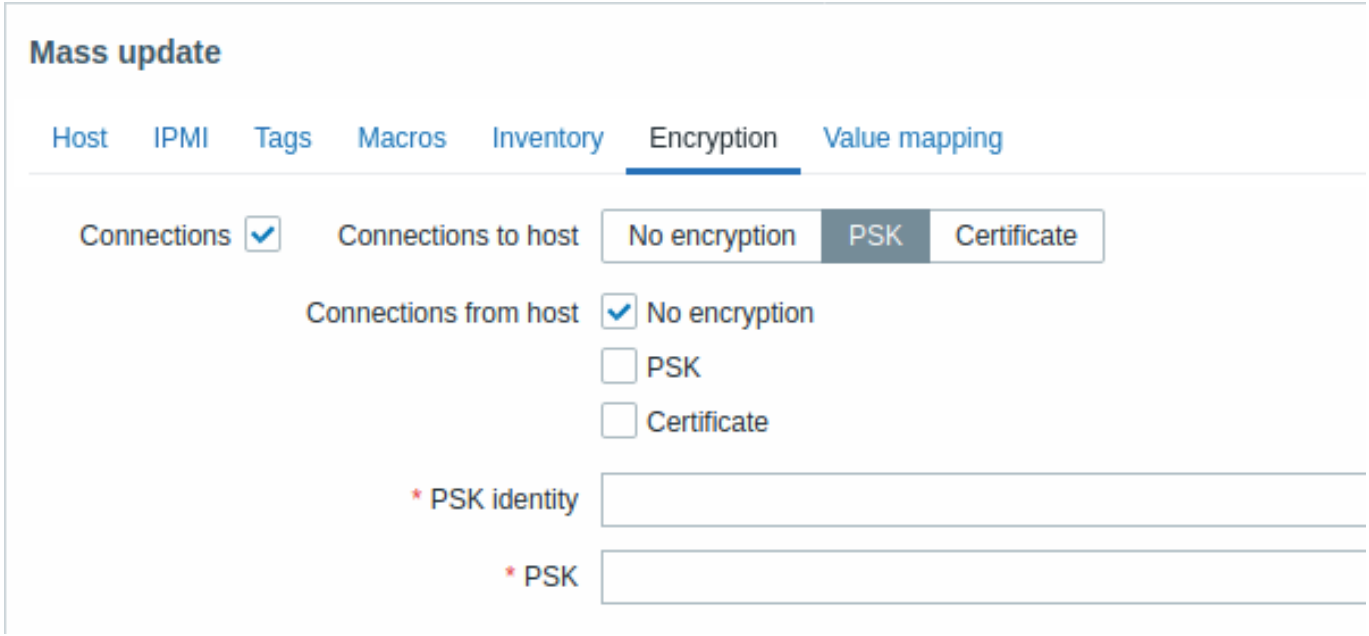
マクロ更新で該当するボタンを選択した場合、次のオプションを利用できます。

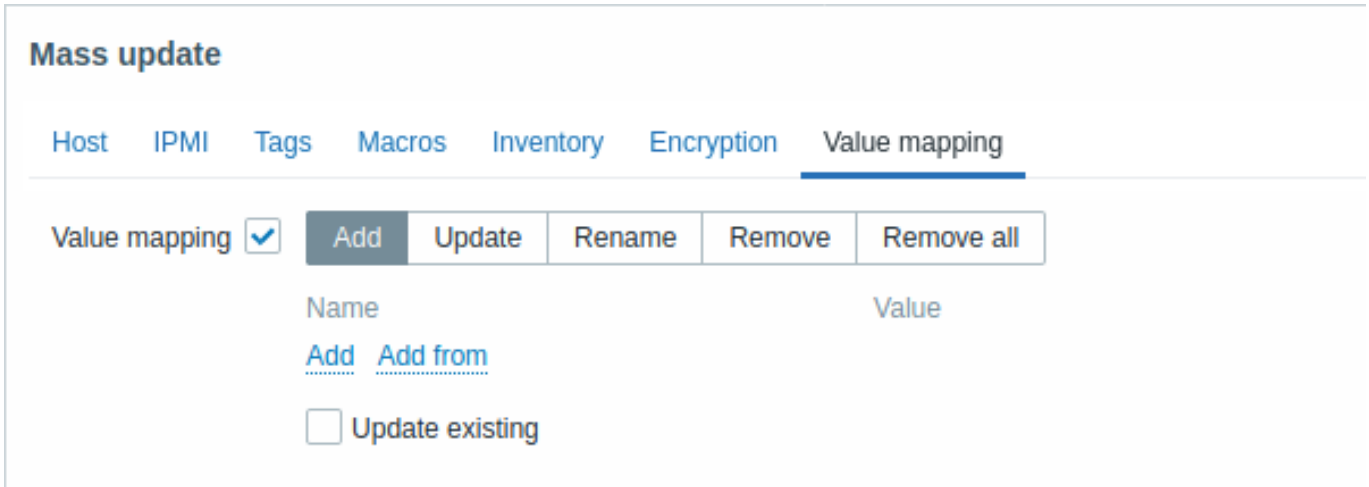
- 追加 - ホストに追加のユーザーマクロを指定できます。既存を更新チェックボックスをオンにすると、指定したマクロ名の値、タイプ、説明が更新されます。オフの場合、その名前のマクロがホスト上にすでに存在していても更新されません。
- 更新 - このリストで指定したマクロの値、タイプ、説明を置き換えます。不足分を追加チェックボックスをオンにすると、以前ホスト上に存在しなかったマクロは新しいマクロとして追加されます。オフの場合、ホスト上にすでに存在するマクロのみが更新されます。
- 削除 - 指定したマクロをホストから削除します。選択したものを除くボックスをオンにすると、リストで指定したもの以外のすべてのマクロが削除されます。オフの場合、リストで指定したマクロのみが削除されます。

- すべて削除 - ホストからすべてのユーザーマクロを削除します。すべてのマクロを削除することを確認しますチェックボックスがオンになっていない場合、すべてのマクロの削除確認を求める新しいポップアップウィンドウが開きます。



インベントリフィールドを一括更新できるようにするには、インベントリモードを「手動」または「自動」に設定する必要があります。





値マップ更新では、次のオプションを持つボタンを利用できます。

- 追加 - ホストに値マップを追加します。既存を更新を選択すると、この名前の値マップのすべてのプロパティが更新されます。

そうでない場合、同じ名前の値マップがすでに存在していても更新されません。
- 更新 - 既存の値マップを更新します。不足分を追加を選択すると、以前ホスト上に存在しなかった値マップは新しい値マップとして追加されます。そうでない場合、ホスト上にすでに存在する値マップのみが更新されます。
- 名前変更 - 既存の値マップに新しい名前を付けます
- 削除 - 指定した値マップをホストから削除します。選択したものを除くを選択すると、指定したものを 除くすべての値マップが削除されます。
- すべて削除 - ホストからすべての値マップを削除します。すべての値マップを削除することを確認しますチェックボックスがオンになっていない場合、削除確認を求める新しいポップアップウィンドウが開きます。

必要な変更をすべて完了したら、更新をクリックします。選択したすべてのホストの属性が、それに応じて更新されます。

2 アイテム

概要

アイテムは個別の指標です。

アイテムはデータ収集に使用されます。ホストを設定したら、実際のデータを取得するためのアイテムを追加する必要があります。多くのアイテムをすばやく追加する方法は、事前定義されたテンプレートの 1 つをホストにアタッチすることです。ただし、システムのパフォーマンスを最適化するには、必要なだけのアイテムと頻度の監視を行うようにテンプレートを微調整する必要があります。

ホストからどのようなデータを収集するかを指定するには、アイテムキーを使用します。たとえば、キー名 **system.cpu.load** のアイテムはプロセッサ負荷データを収集し、キー名 **net.if.in** のアイテムは受信トラフィック情報を収集します。

キー名の後の角括弧内に追加のパラメーターを指定できます。たとえば、system.cpu.load**[avg5]** は過去 5 分間のプロセッサ負荷平均を返し、net.if.in**[eth0]** はインターフェース”eth0” の受信トラフィックを表示します。

> **Note:**
> サポートされているすべてのアイテムタイプとアイテムキーについては、アイテムタイプの個々のセクションを参照してください。

アイテムの作成と設定に進んでください。

1 アイテムの作成

概要

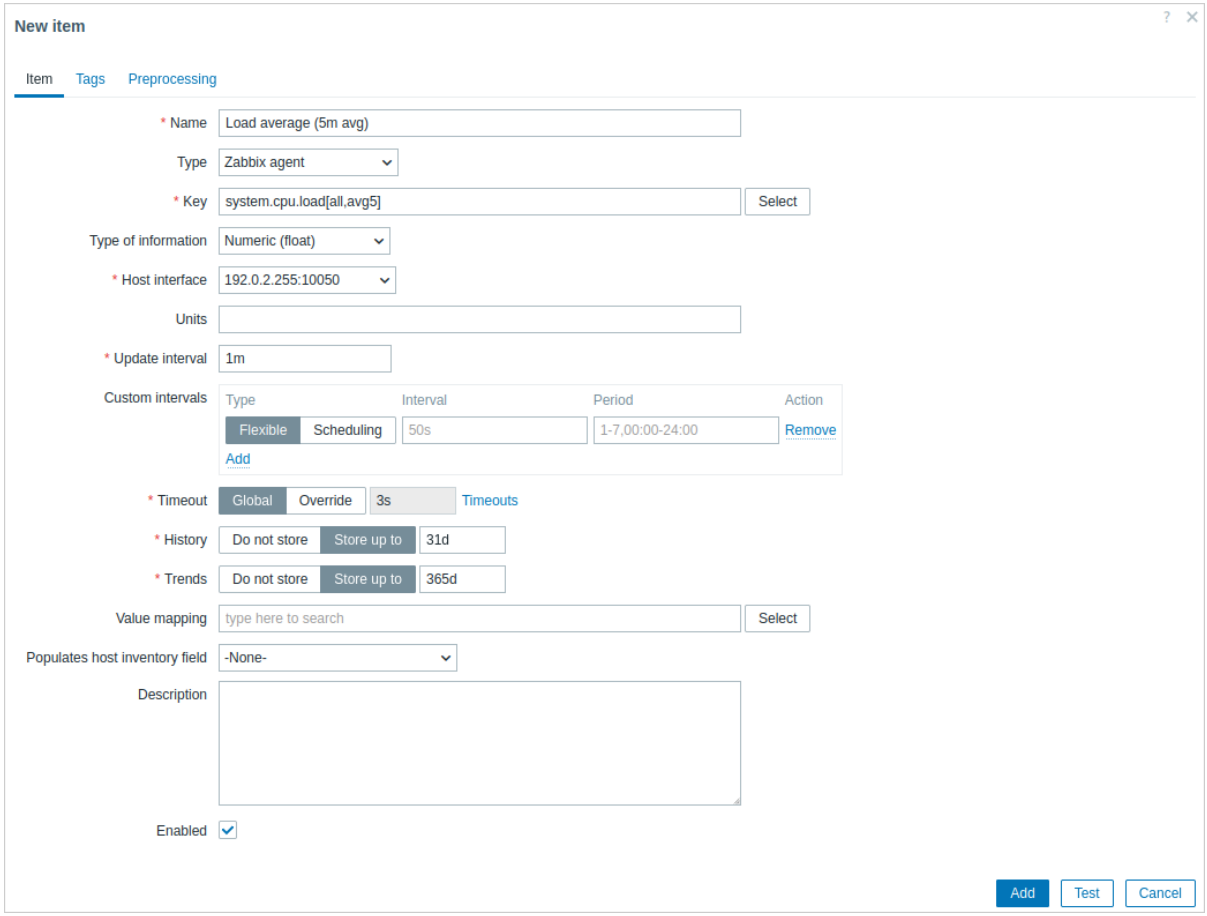
Zabbix Web インターフェースでアイテムを作成するには、次の手順を実行します。

- データ収集 > ホストに移動します
- ホストの行にある アイテムをクリックします
- 画面右上の アイテムの作成をクリックします
- フォームにアイテムのパラメータを入力します

既存のアイテムを開き、複製ボタンを押してから別の名前で保存することでも、アイテムを作成できます。

設定

アイテムタブには、アイテムの一般的な属性が含まれます。



必須入力フィールドには、すべて赤いアスタリスクが付いています。

Parameter	Description
Name	アイテム名。 ユーザーマクロを使用できます。
Type	アイテムのタイプ。各item typeセクションを参照してください。
Key	アイテムキー（最大 2048 文字）。 サポートされているitem keysは、各アイテムタイプのセクションで確認できます。 キーは、1 つのホスト内で一意でなければなりません。 キータイプが「Zabbix agent」、「Zabbix agent (active)」、または「Simple check」の場合、キー値は Zabbix agent または Zabbix server でサポートされている必要があります。 あわせて、正しいkey formatも参照してください。
Type of information	必要に応じて変換を行った後、データベースに保存されるデータの種類です。 **Numeric (unsigned)** - 64 ビット符号なし整数。 整数アイテムで浮動小数点値を受信した場合、小数部は切り捨てられることに注意してください。 **Numeric (float)** - 64 ビット浮動小数点数。 このタイプでは、およそ 15 桁の精度と、およそ -1.79E+308 から 1.79E+308 までの範囲を扱えます（PostgreSQL 11 and earlier versionsを除く）。 科学表記の値の受信もサポートされています。例: 1.23E+7、1e308、1.1E-4。 **Character** - 短いテキストデータ。 **Log** - ログ関連のプロパティ（timestamp、source、severity、logeventid）を任意で含む長いテキストデータ。 **Text** - 長いテキストデータ。text data limitsも参照してください。 **Binary** - バイナリ数値（依存アイテムでのみサポート）。 バイナリデータ型の値を返すアイテムは、計算アイテムの数式やトリガー式ではサポートされません。 1 つの特定の形式でのみデータを返すアイテムキーについては、対応する情報のタイプが自動的に選択されます。
Host interface	ホストインターフェースを選択します。このフィールドは、ホストレベルでアイテムを編集する場合に使用できます。

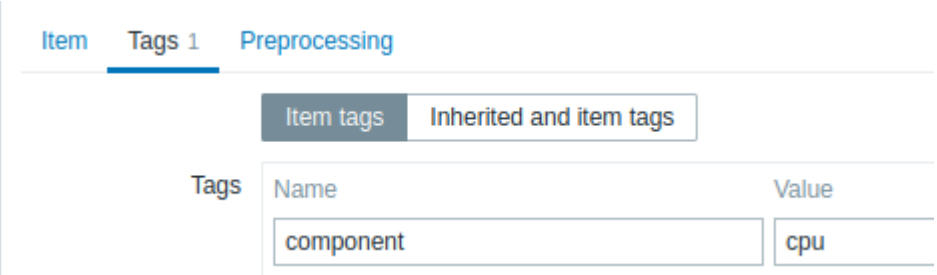
Parameter	Description
Units	単位記号が設定されている場合、Zabbix は受信したアイテム値に後処理を適用し、指定した単位の接尾辞を付けて表示します。 特別な書式設定に対応した単位記号（および受信したアイテム値 → 表示値の例）は次のとおりです。 **B** - バイト (1024 → 1 KB) **Bps** - 1 秒あたりのバイト数 (1024 → 1 KBps) **s** - 秒。最大 3 つまでの最も大きい非ゼロの時間単位で表示されます (881764 → 10d 4h 56m) **uptime** - 経過時間。hh:mm:ss または N days, hh:mm:ss 形式で表示されます (881764 → 10 days, 04:56:04) **unixtime** - Unix タイムスタンプ。yyyy.mm.dd hh:mm:ss 形式で表示されます (881764 → 1970-01-11 04:56:04 AM)。正しく書式設定するには、受信したアイテム値が *Numeric (unsigned)* である必要があります。 その他の単位（Hz、W など）では、受信値が 1000 を超える場合、1000 で割られ、対応する接頭辞付きで表示されます (5000 → 5 KHz、881764 → 881.76 KW)。 単位記号の先頭に ！ を付けると、単位変換と接頭辞付与は無効になります (1024 !B → 1024 B、61 !s → 61 s)。 単位記号と単位変換の詳細およびその他の例については、Item value suffixesを参照してください。
Update interval	このアイテムの新しい値を N 秒ごとに取得します。設定可能な更新間隔の最大値は 86400 秒（1 日）です。 Time suffixesを使用できます。例: 30s、1m、2h、1d。 ユーザーマクロを使用できます。 1 つのマクロでフィールド全体を埋める必要があります。1 つのフィールド内で複数のマクロを使用したり、マクロとテキストを混在させたりすることはできません。 注: 更新間隔を「0」に設定できるのは、0 以外の値を持つカスタム間隔が存在する場合のみです。「0」に設定され、かつ 0 以外の値を持つカスタム間隔（柔軟またはスケジュール）が存在する場合、アイテムはそのカスタム間隔の期間中にポーリングされます。 注: アイテムが有効になった後、または更新間隔を変更した後の最初のアイテムポーリングは、設定値より早く発生する場合があります。 新しいアイテムは、Scheduling または Flexible update interval があり、かつ *Update interval* が 0 に設定されていない限り、作成後 60 秒以内にチェックされます。 既存のパッシブアイテムは、*Execute now* button を押すことで、直ちに値をポーリングできます。
Custom intervals	アイテムのチェックに対してカスタムルールを作成できます。 **Flexible** - *Update interval* に対する例外（異なる頻度の間隔）を作成します。 **Scheduling** - カスタムのポーリングスケジュールを作成します。 詳細はCustom intervalsを参照してください。 *Interval* フィールドではTime suffixesを使用できます。例: 30s、1m、2h、1d。 ユーザーマクロを使用できます。 1 つのマクロでフィールド全体を埋める必要があります。1 つのフィールド内で複数のマクロを使用したり、マクロとテキストを混在させたりすることはできません。
Timeout	アイテムチェックのタイムアウトを設定します（supportedアイテムタイプで使用可能）。タイムアウトオプションを選択します。 **Global** - プロキシ/グローバルタイムアウトが使用されます（グレー表示された *Timeout* フィールドに表示されます）。 **Override** - カスタムタイムアウトが使用されます（*Timeout* フィールドで設定。許容範囲: 1 - 600s）。Time suffixes（例: 30s、1m）およびユーザーマクロを使用できます。 *Timeouts* リンクをクリックすると、プロキシのタイムアウト、または（プロキシを使用しない場合は）globalタイムアウトを設定できます。*Timeouts* リンクは、*Administration* > *General* または *Administration* > *Proxies* の Web インターフェースセクションに対する権限を持つ *Super admin* タイプのユーザーにのみ表示される点に注意してください。

Parameter	Description
History	次のいずれかを選択します。 **Do not store** - アイテムの履歴は保存されません。依存アイテムのみで履歴を保持する必要がある場合、マスターアイテムに有用です。 この設定は、グローバル housekeeper settingsでは上書きできません。 **Store up to** - データベース内に詳細な履歴を保持する期間を指定します（1 時間～25 年）。古いデータは housekeeper によって削除されます。秒単位で保存されます。 Time suffixesを使用できます。例: 2h、1d。ユーザーマクロを使用できます。 *Store up to* の値は、*Administration* > *Housekeeping* でグローバルに上書きできます。  情報アイコンが表示されます。その上にマウスカーソルを置くと、たとえば *Overridden by global housekeeper settings (1d)* のような警告メッセージが表示されます。 データベース内の値履歴のサイズを減らすため、記録された値は可能な限り短い期間だけ保持することを推奨します。長期間の値履歴を保存する代わりに、より長期間のトレンドデータを保存できます。 History and trendsも参照してください。
Trends	次のいずれかを選択します。 **Do not store** - トレンドは保存されません。 この設定は、グローバル housekeeper settingsでは上書きできません。 **Store up to** - 集計済み履歴（1 時間ごとの最小値、最大値、平均値、件数）をデータベース内に保持する期間を指定します（1 日～25 年）。古いデータは housekeeper によって削除されます。秒単位で保存されます。 Time suffixesを使用できます。例: 24h、1d。ユーザーマクロを使用できます。 *Store up to* の値は、*Administration* > *Housekeeping* でグローバルに上書きできます。  情報アイコンが表示されます。その上にマウスカーソルを置くと、たとえば *Overridden by global housekeeper settings (7d)* のような警告メッセージが表示されます。 注: トレンドの保持は、数値以外のデータ（character、log、text）では使用できません。 History and trendsも参照してください。
Value mapping	このアイテムに値のマッピングを適用します。Value mappingは受信値自体を変更せず、表示のためだけに使用されます。 *Numeric(unsigned)*、*Numeric(float)*、*Character* アイテムで機能します。 たとえば、「Windows service states」です。
Log time format	**Log** タイプのアイテムでのみ使用できます。サポートされるプレースホルダー: **y**: 年 *(1970-2038)*。 **M**: 月 *(01-12)*。 **d**: 日 *(01-31)*。 **h**: 時 *(00-23)*。 **m**: 分 *(00-59)*。 **s**: 秒 *(00-59)*。 空白のままにすると、タイムスタンプは Unix time で 0 に設定され、1970 年 1 月 1 日を表します。 たとえば、次の Zabbix agent ログファイルの行を考えてみます。 ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” この行は、PID 用の 6 文字分の位置で始まり、その後に日付、時刻、メッセージの残りが続きます。 この行のログ時刻形式は”ppppp:yyyyMMdd:hhmmss” になります。 ”p” と”:” はプレースホルダーであり、”yMdhms” 以外であれば任意の文字を使用できる点に注意してください。
Populates host inventory field	アイテムの値を格納するホストインベントリフィールドを選択できます。これは、そのホストで自動inventory登録が有効になっている場合に機能します。 *Type of information* が「Log」に設定されている場合、このフィールドは使用できません。
Description	アイテムの説明を入力します。ユーザーマクロを使用できます。
Enabled	チェックボックスをオンにするとアイテムが有効になり、処理されます。 アイテムを無効にすると、履歴キャッシュから直ちに削除されます（最後の値を除く。ログ用に保持されます）。
Latest data	リンクをクリックすると、そのアイテムの最新データを表示します。 このリンクは、既存のアイテムを編集している場合にのみ使用できます。

Note:
アイテムタイプ固有のフィールドは、対応するページで説明されています。

Note:
ホストレベルで既存のテンプレートレベルのアイテムを編集する場合、いくつかのフィールドは読み取り専用になります。フォームヘッダー内のリンクを使用してテンプレートレベルに移動し、そこで編集できます。ただし、テンプレートレベルでの変更は、そのテンプレートがリンクされているすべてのホストのアイテムに反映されることに注意してください。

Tags タブでは、アイテムレベルのtagsを定義できます。



アイテムの値の保存前処理

保存前処理タブでは、受け取った値に対して変換するルールを定義することができます。

アイテムのテスト

Attention:
アイテムのテストを実行するには、サーバーとプロキシのシステム時刻が同期されていることを確認してください。サーバー時刻が遅れている場合、アイテムのテストで「The task has been expired.」というエラーメッセージが返されることがあります。ただし、サーバーとプロキシで異なるタイムゾーンが設定されていても、テスト結果には影響しません。

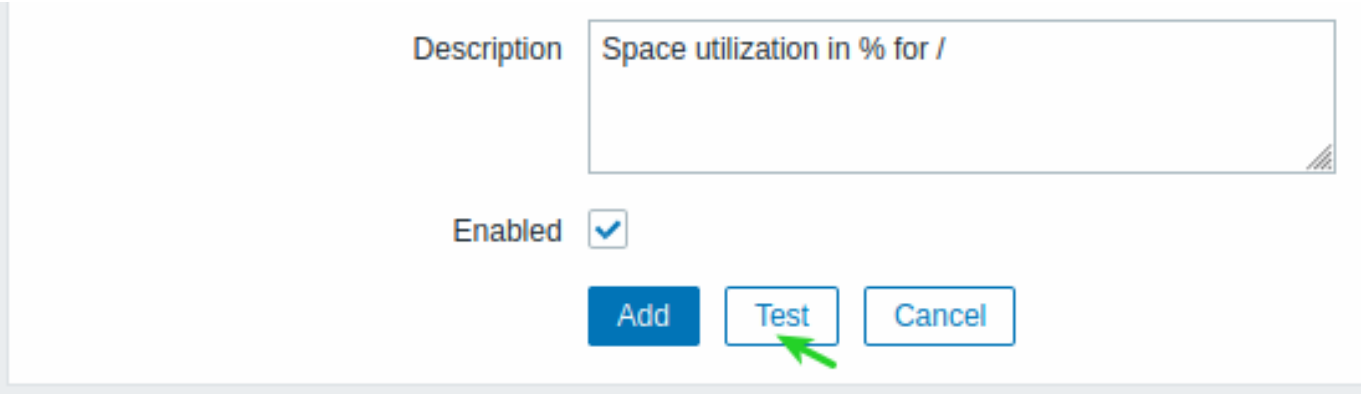
アイテムをテストし、正しく設定されていれば実際の値を取得できます。テストは、アイテムを保存する前でも実行できます。

テストは、ホストおよびテンプレートのアイテム、アイテムのプロトタイプ、ローレベルディスカバリルールで利用できます。アクティブアイテムではテストは利用できません。

アイテムのテストは、以下のパッシブアイテムタイプで利用できます。

- Zabbix エージェント
- SNMP エージェント (v1, v2, v3)
- IPMI エージェント
- SSH チェック
- Telnet チェック
- JMX エージェント
- シンプルチェック (`icmpping*`、`vmware.*` アイテムを除く)
- Zabbix 内部
- 計算アイテム
- 外部チェック
- データベースモニタ
- HTTP エージェント
- スクリプト
- ブラウザ

アイテムをテストするには、アイテム設定フォームの下部にある *Test* ボタンをクリックします。テストできないアイテム（アクティブチェックや、除外されるシンプルチェックなど）では、*Test* ボタンは無効になります。

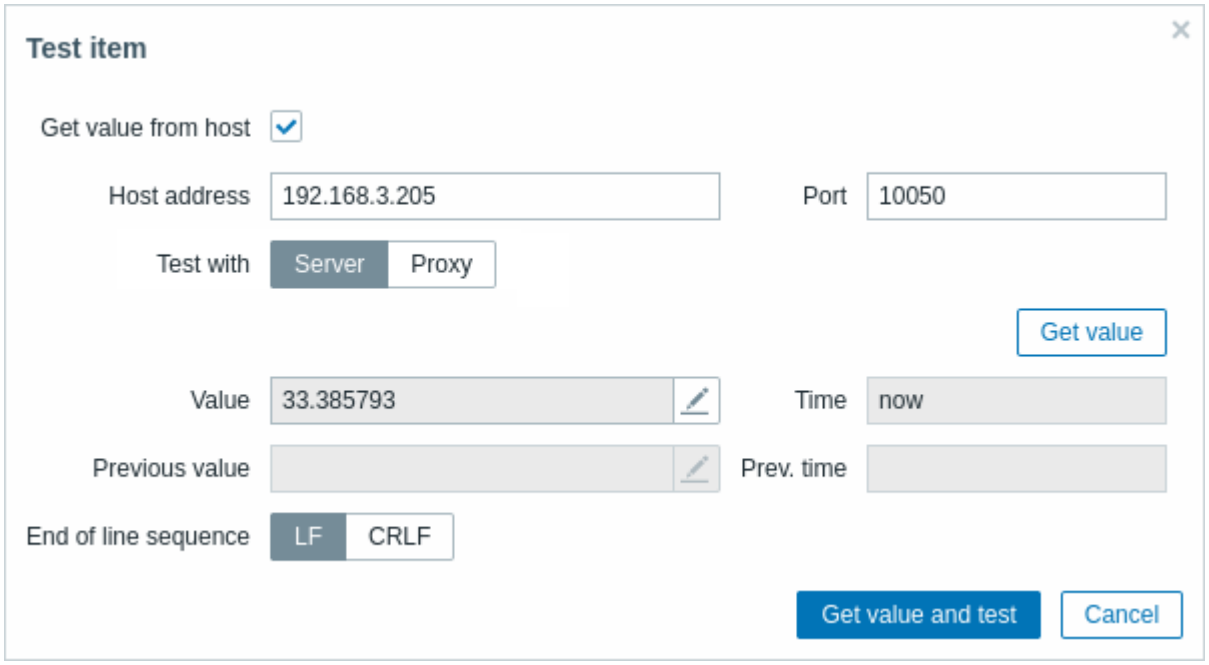


アイテムのテストフォームには、必要なホストパラメータ（ホストアドレス、ポート、サーバー/プロキシでテストするか（プロキシ名））およびアイテム固有の詳細（SNMPv2 コミュニティや SNMPv3 セキュリティ認証情報など）のフィールドがあります。これらのフィールドはコンテキストに応じて変化します。

- 可能な場合は値が事前入力されます。たとえば、エージェントを必要とするアイテムでは、ホストの選択されたエージェントインターフェースの情報が使用されます。
- テンプレートのアイテムでは、値を手動で入力する必要があります。
- プレーンテキストのマクロ値は展開されます。
- 値（または値の一部）がシークレットマクロまたは Vault マクロであるフィールドは空欄となり、手動で入力する必要があります。アイテムのいずれかのパラメータにシークレットマクロ値が含まれている場合、次の警告メッセージが表示されます: 「Item contains user-defined macros with secret values. Values of these macros should be entered manually.」
- アイテムタイプのコンテキスト上不要な場合、フィールドは無効になります（例: 計算アイテムでは、ホストアドレスフィールドとプロキシフィールドは無効になります）

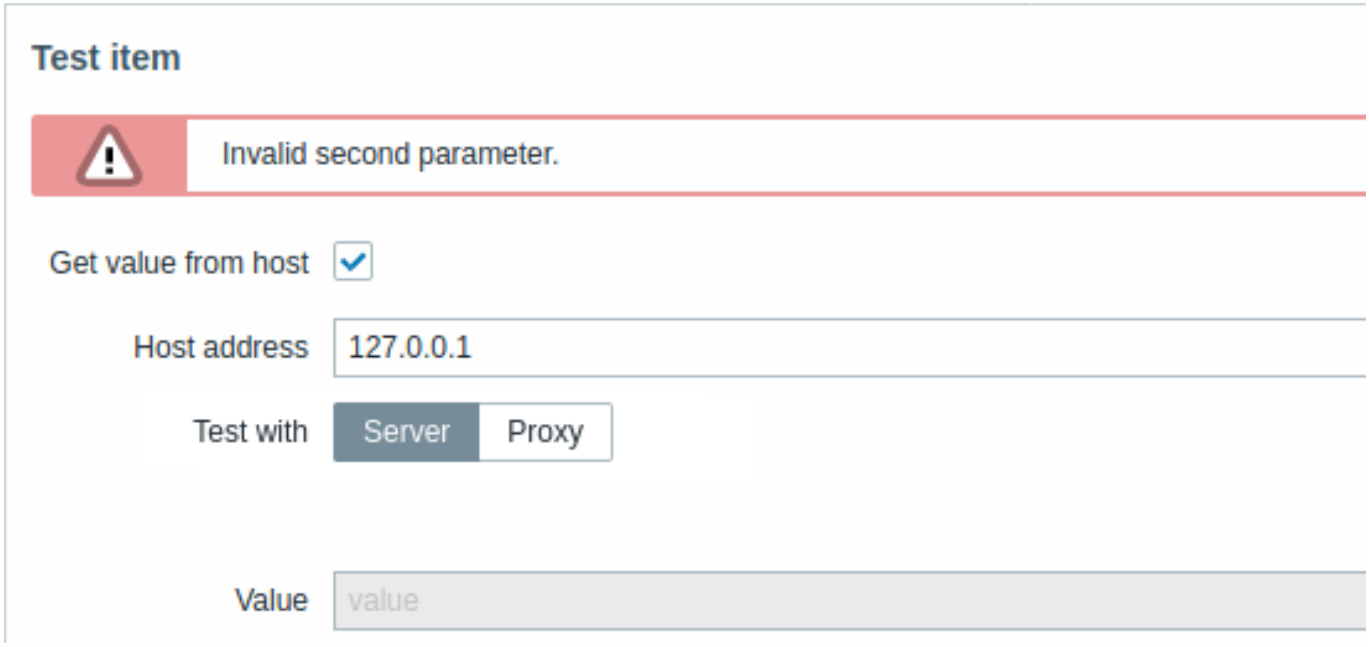
アイテムをテストするには、*Get value* をクリックします。値の取得に成功すると、その値が *Value* フィールドに入力され、現在の値（存在する場合）は *Previous value* フィールドに移動されます。同時に *Prev. time* フィールド、つまり 2 つの値（クリック）間の時間差が計算され、取得した値に”\n\r” が検出された場合は EOL シーケンスの検出が試行され、CRLF に切り替えられます。

前処理をテストするには、*Get value and test* をクリックします。



ホストから取得した値およびテスト結果は、Web インターフェースに送信される際に最大 512KB に切り詰められます。結果が切り詰められた場合は、警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも、Zabbix サーバーでは引き続き完全に処理されることに注意してください。

設定が正しくない場合は、考えられる原因を説明するエラーメッセージが表示されます。



ホストから正常に取得した値は、前処理ステップのテストにも使用できます。

フォームのボタン

フォームの下部にあるボタンで、いくつかの操作を行うことができます。

Add	アイテムの追加。このボタンは新規のアイテムの時のみ使用できます。
Update	アイテムのプロパティの更新。
Clone	現在のアイテムプロパティをベースにして別のアイテムを作成。
Execute now	アイテムの値を取得。アイテムのタイプがパッシブなものだけ対応しています。(詳細参照) 注意値の取得時には設定キャッシュは更新されません。つまり直近のアイテムの設定変更が反映されません。
Test	値を取得できるかのテスト実行。
Delete and clear	アイテムのヒストリとトレンドを削除。
Delete	アイテムの削除。
Cancel	アイテムのプロパティ変更をキャンセル。

テキストデータの制限

テキストデータの制限はデータベースバックエンドによって異なります。テキスト値は、データベースに保存する前に、データベース値の型の制限に一致するように切り詰められます。

データベース	情報の種類		
	Character	**Log**	**Text**
MySQL	255 文字	65536 バイト	65536 バイト
PostgreSQL	255 文字	65536 文字	65536 文字
SQLite (Zabbix プロキシのみ)	255 文字	65536 文字	65536 文字

カスタムスクリプトの制限

使用可能なカスタムスクリプトの長さは、使用するデータベースによって異なります。

データベース	文字数の上限	バイト数の上限
MySQL	65535	65535
PostgreSQL	65535	制限なし
SQLite (Zabbix プロキシのみ)	65535	制限なし

アイテムのタイムアウト

アイテムのタイムアウトは、チェックを失敗として中止するまでに Zabbix がどれだけ待機するかを指定します。

タイムアウトに達すると、データ取得が完了していなくてもチェックは中止されます。データが部分的に受信された場合、アイテムはサポートされていない状態となり、エラーメッセージがログに記録されます（たとえば、SNMP チェックで複数の OID のうち 1 つについてのみ正常にデータが収集された場合）。

多くのアイテムタイプでは、柔軟なアイテムタイムアウトを設定できます。

- 個々のアイテムごと
- アイテムタイプごと（プロキシレベル）
- アイテムタイプごと（グローバルレベル）

個々のアイテムごとのカスタムタイムアウトは、他のアイテムのタイムアウトを短く保ったまま、特定のアイテムに対してより長いタイムアウトを指定したい場合に便利です。

タイムアウトの優先順位

1. 個々のアイテムのタイムアウトは、他のすべてのタイムアウトより優先されます。
2. プロキシレベルのタイムアウトは、グローバルタイムアウトより優先されます。

柔軟なタイムアウトが設定されているチェックでは、Zabbix サーバー またはプロキシ の設定にあるタイムアウトは一切関係しません。

柔軟なタイムアウトのサポート

柔軟なアイテムのタイムアウトは、以下のアイテムタイプでサポートされています。

- Zabbix エージェント（パッシブチェックとアクティブチェックの両方）
- SNMP エージェント（従来の SNMP チェック[1] を除く）
- シンプルチェック（icmpping* および VMware アイテム[2] を除く）
- SSH エージェント
- Telnet エージェント
- 外部チェック
- データベースモニタ
- スクリプト
- HTTP エージェント
- ブラウザ

[1] 従来の SNMP チェック（単一 OID のポーリング）では、サーバー またはプロキシ のタイムアウト設定が適用されます。
 [2] icmpping* アイテムでは、タイムアウト値はアイテムキー内で直接指定されます。VMware 監視アイテムでは、サーバー またはプロキシ の設定にある独自の VMwareTimeout パラメータが使用されます。

サポートされていないアイテム

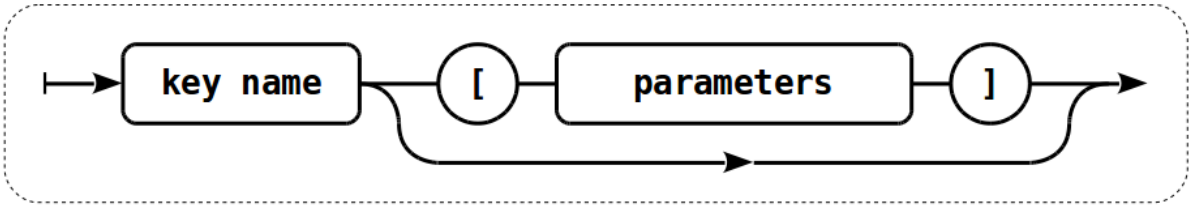
サポートされていないアイテムは、取得不可ステータスとして報告され、標準の 更新間隔 で再チェックされます。

何らかの理由 (接続エラー、アイテムを処理するように設定されたポーラーがないなど) で値を取得できない場合、アイテムはサポートされません。また、ヒストリデータを受け取らないアイテムは、取得不可ステータスのままになります。これは想定された動作です。アイテムは、新しい有効なヒストリデータを受け取った場合にのみ、通常ステータスに移行します。

障害と同様に、サポートされていないアイテムは、そのアイテムのヒストリデータがもう利用できない場合であっても、新しいデータを受け取った場合にのみ再評価されます。つまり、アイテムとトリガーは、新しいデータを受け取った場合にのみ状態を変更します。ただし、トリガー式に日付と時刻やnodata()関数が含まれている場合、トリガーは 30 秒ごとに再計算されます (詳細については、トリガーの計算時間を参照してください)。

1 アイテムキーのフォーマット

キーパラメーターを含むアイテムキーのフォーマットは、構文規則に従う必要があります。次の図は、サポートされている構文を示しています。各ポイントで許可される要素と文字は、矢印に従って決定できます。矢印の先にあるブロックに到達できる場合は許可され、到達できない場合は許可されません。



有効なアイテムキーを作成するには、キー名を指定し、次にパラメーターの有無を選択します。

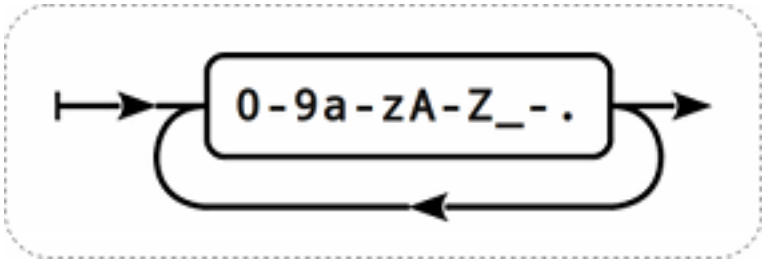
キー名

キー名自体には、使用できる文字の範囲が限られており、その文字が相互に続くだけです。許容される文字は、

`0-9a-zA-Z_-.`

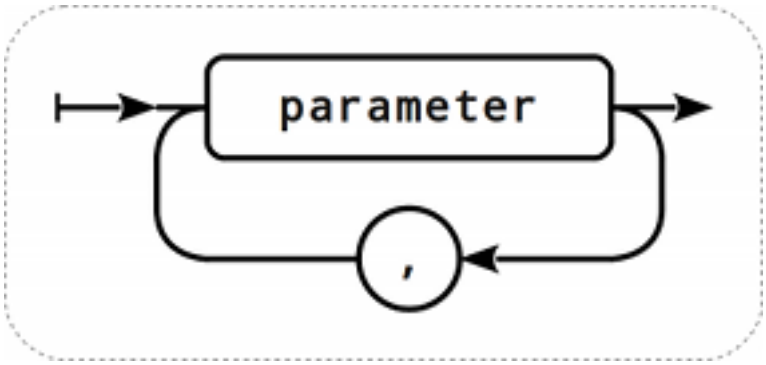
つまり、以下の文字種を指します。

- すべての数字
- すべての小文字
- すべての大文字
- アンダースコア
- ダッシュ
- ドット

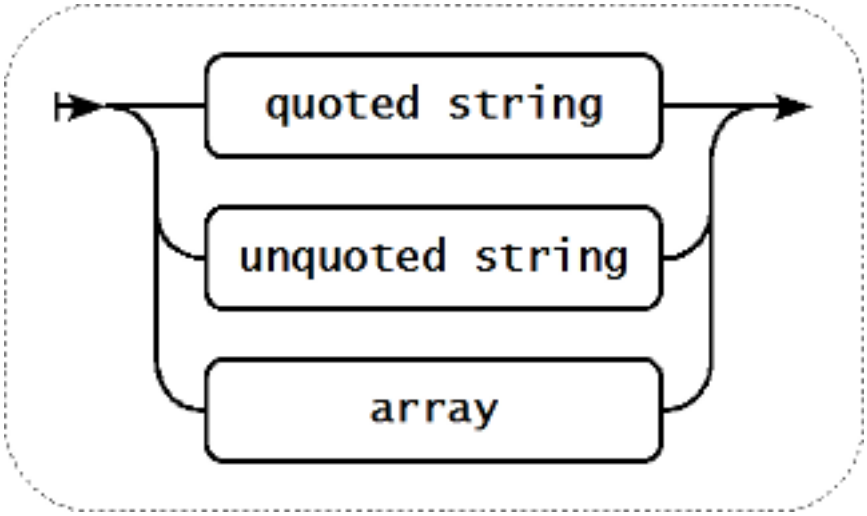


キーパラメーター

アイテムキーは、カンマ区切りで複数のパラメーターを持つことができます。



各キーパラメーターは、引用符で囲まれた文字列、引用符で囲まれていない文字列、または配列のいずれかになります。

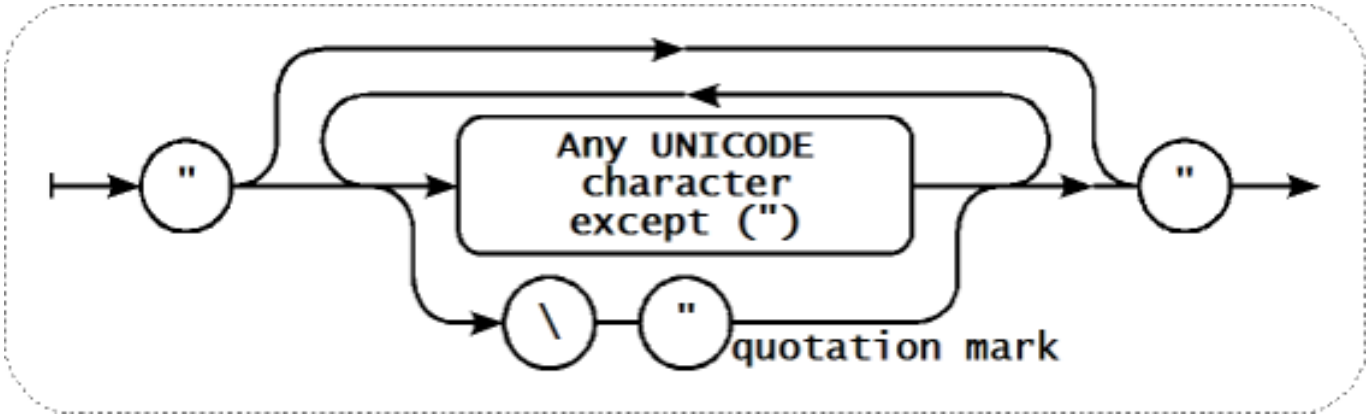


また、パラメーターを空にして、デフォルト値を使用することも可能です。その場合、さらにパラメーターを指定する場合は、適切な数のカンマを追加する必要があります。例えば、アイテムキー **icmpping[„200„500]** は、個々の ping の間隔が 200 ミリ秒、ping のタイムアウトが 500 ミリ秒、その他のパラメーターはデフォルトのままです。

パラメーターにマクロを含めることができます。マクロはユーザーマクロまたは組み込みマクロの一部です。アイテムキーパラメーターでサポートされている特定の組み込みマクロを確認するには、サポートされているマクロページで” アイテムキーパラメーター” を検索してください。

パラメーター - 二重引用符 (”) で囲まれた文字列

キーパラメーターが二重引用符で囲まれた文字列の場合、任意の Unicode 文字が使用可能です。キーパラメーターが二重引用符を含む場合、このパラメーターを二重引用符で囲み、パラメーター文字列の一部である各二重引用符はバックスラッシュ (\) でエスケープする必要があります。キーパラメーターがカンマを含む場合、このパラメーターは二重引用符で囲む必要があります。

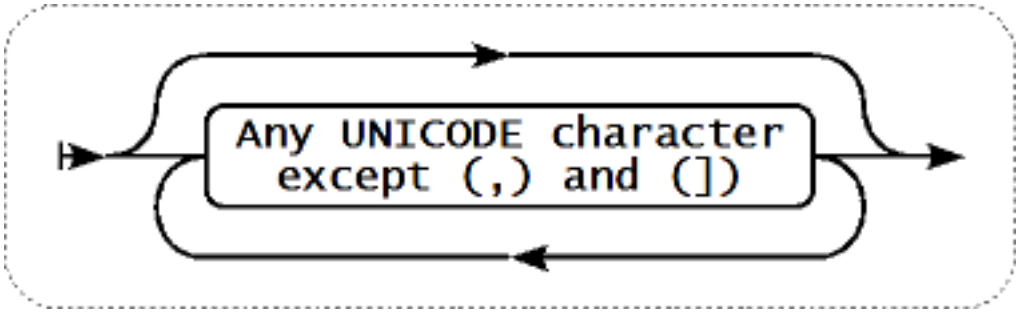


Warning:
アイテムのキーパラメーターを引用符で囲むには、二重引用符 (”) のみを使用してください。一重引用符 (’) はサポートされていません。

Attention:
`[a,[b,[c,d]],e]` のようなマルチレベルパラメーター配列は許可されていません。

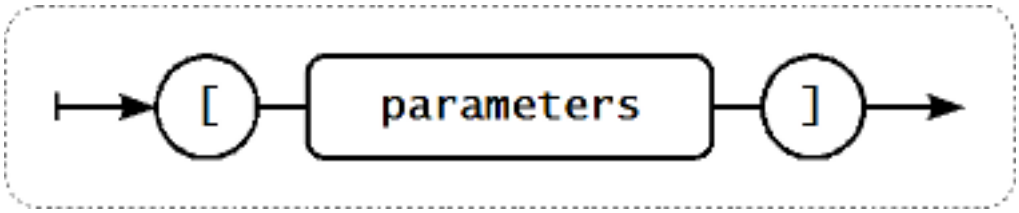
パラメーター - 二重引用符 (”) がない文字列

キーパラメーターに二重引用符がない文字列の場合、カンマと右大括弧 (]) を除く全ての Unicode 文字が使用可能です。二重引用符で囲まれていないパラメータで、左大括弧 ([) で始まる文字列は使用できません。



パラメーター - 配列

キーパラメーターが配列の場合、これも角括弧で囲まれ、個々のパラメーターは複数のパラメーターを指定する規則と構文に従います。



2 監視間隔のカスタマイズ

概要

アイテムがチェックされる時間帯について、カスタムルールを作成することができます。デフォルトの更新間隔を再定義できる例外設定と、特定の時間または一連の時間にアイテムチェックを実行することができる定期設定の 2 つの方法があります。

例外設定

例外設定は、特定の時間帯のデフォルトの更新間隔を再定義することができます。例外設定は、監視間隔と期間で定義されます。

- 監視間隔 - 指定された期間の更新間隔です。
- 期間 - 例外設定が有効な期間 (詳細については、日時フォーマットを参照してください)。

複数の例外設定の監視間隔が重なった場合、最も小さい監視間隔の値が重なる期間に使用されます。重なる設定の監視間隔の最小値が'0' である場合、ポーリングは行われません。例外設定の期間外では、デフォルトの監視間隔が使用されます。

例外設定の監視間隔が期間の長さと等しい場合、アイテムは一度だけチェックされることに注意してください。例外設定の監視間隔が期間より大きい場合、アイテムは一度だけチェックされるか、全くチェックされないかもしれません (したがって、このような構成は望ましくありません)。もし監視間隔が期間より短い場合、そのアイテムは少なくとも 1 回チェックされます。

例外設定の監視間隔が'0' に設定されている場合、そのアイテムは例外設定の期間中はポーリングされません。期間が終了するとデフォルトの監視間隔に従ってポーリングが再開されます。たとえば:

監視間隔	期間	説明
10	1-5,09:00-18:00	アイテムは勤務時間中に 10 秒ごとにチェックする。
0	1-7,00:00-7:00	アイテムは夜にチェックを行わない。
0	7-7,00:00-24:00	アイテムは日曜日にはチェックを行わない。
60	1-7,12:00-12:01	アイテムは毎日 12:00 にチェックする。これは、スケジュールされたチェック用に回避策として用意されたものであることに注意してください。このようなチェックには定期設定を使用してください。

定期設定

定期設定は、特定の時間にアイテムのチェックを実行するために使用します。例外設定は、デフォルトのアイテムの監視間隔を再定義するよう設計されていますが、定期設定は、並行して実行される独立したチェックスケジュールを指定するために使用されます。

定期設定の監視間隔は次のように定義されます: `md<filter>wd<filter>h<filter>m<filter>s<filter>`

- **md** - 日付
- **wd** - 曜日
- **h** - 時
- **m** - 分
- **s** – 秒

`<filter>` は、その接頭辞の値 (日付、時、分、秒) を指定するために用いられ、次のように定義されます: `[<from>[-<to>]][/<step>][,<filter`

- `<from>` と `<to>` はマッチする値の範囲 (含む) を定義します。`<to>` が省略された場合、フィルターは `<from> - <from>` の範囲でマッチします。`<from>` が省略された場合、フィルタはすべての可能な値にマッチします。
- `<step>` は、範囲内の数値をどのようにスキップするかを指定します。デフォルトでは `<step>` の値は 1 で、これは定義された範囲のすべての値にマッチすることを意味します。

フィルタの定義は任意であるが、少なくとも 1 つのフィルタが使用されなければなりません。フィルタは範囲か *<step>* 値を定義しなければなりません。

空のフィルタは、下位レベルのフィルタが定義されていない場合は'0' に、それ以外の場合は可能なすべての値にマッチします。例えば、時間フィルタが省略された場合は、'0' の時間だけがマッチし、分と秒のフィルタも空の場合は、すべての時間の値にマッチします。

`<from>` と `<to>` それぞれのフィルタプレフィックスに有効な値は:

接頭辞	説明	*<from>*	*<to>*
md	日付	1-31	1-31
wd	曜日	1-7	1-7
h	時間	0-23	0-23
m	分	0-59	0-59
s	秒	0-59	0-59

`<from>` の値は `<to>` の値より小さいか等しくなければなりません。`<step>` の値は、1 以上かつ `<to> - <from>` 以下の値でなければなりません。

一桁の月日、時間、分、秒の値には、先頭に 0 を付けることができます。例えば、md01-31 と h/02 は有効な間隔ですが、md01-031 と wd01-07 は有効ではありません。

Zabbix の Web インターフェースでは、複数の定期間隔を別々の行に入力することができます。Zabbix API では、セミコロン; をセパレーターとして 1 つの文字列に連結されます。

ある時刻が複数の区間に一致する場合、一度だけ実行されます。例えば、wd1h9;h9 は月曜日の午前 9 時に一度だけ実行されます。

監視間隔	実行される日時
m0-59	毎分
h9-17/2	9:00 から 2 時間ごと (9:00、11:00、...)
m0,30 or m/30	各時の 0 分と 30 分 (hh:00 と hh:30)
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	0 分から 5 分ごと
wd1-5h9	月曜から金曜までの 9:00
wd1-5h9-18	月曜から金曜までの 9:00、10:00、...、18:00
h9,10,11 or h9-11	毎日の 9:00、10:00、11:00
md1h9m30	毎月 1 日の 9:30
md1wd1h9m30	毎月 1 日かつ月曜日の 9:30
h9m/30	毎日の 9:00、9:30
h9m0-59/30	毎日の 9:00、9:30
h9,10m/30	毎日の 9:00、9:30、10:00、10:30
h9-10m30	毎日の 9:30、10:30
h9m10-40/30	毎日の 9:10、9:40
h9,10m10-40/30	毎日の 9:10、9:40、10:10、10:40
h9-10m10-40/30	毎日の 9:10、9:40、10:10、10:40
h9m10-40	毎日の 9:10、9:11、9:12、...、9:40
h9m10-40/1	毎日の 9:10、9:11、9:12、...、9:40
h9-12,15	毎日の 9:00、10:00、11:00、12:00、15:00
h9-12,15m0	毎日の 9:00、10:00、11:00、12:00、15:00
h9-12,15m0s30	毎日の 9:00:30、10:00:30、11:00:30、12:00:30、15:00:30
h9-12s30	毎日の 9:00:30、9:01:30、9:02:30、...、12:58:30、12:59:30
h9m/30;h10 (*API* 用の文法)	毎日の 9:00、9:30、10:00
h9m/30 h10 (*Web* インターフェース内での別の行として追加)	毎日の 9:00、9:30、10:00

プロキシとエージェントのタイムゾーンの調整

Zabbix プロキシとエージェントは、スケジュール間隔を処理するときにローカルタイムゾーンを使用することに注意してください。

このため、Zabbix プロキシまたはエージェントのアクティブアイテムによって監視されるアイテムにスケジュール間隔を適用する場合は、それぞれのプロキシまたはエージェントのタイムゾーンを Zabbix サーバーと同じに設定することをお勧めします。

Zabbix プロキシまたはエージェントのタイムゾーンは、systemd ユニットファイルの環境変数 TZ を使用して設定できます。

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

2 アイテムの値の保存前処理

概要

前処理では、受信したアイテムの値をデータベースに保存する前に変換を適用できます。これらの変換/前処理ステップは、Zabbix サーバーまたはプロキシ（アイテムがプロキシによって監視されている場合）によって実行されます。

この機能は、例えば次のようなさまざまな用途に対応しています。

- バイトをビットに変換する（例: ネットワークトラフィックの値に「8」を掛ける）;
- 増加し続ける値について 1 秒あたりの統計を計算する;
- 正規表現を適用して値を抽出または変更する;
- 値に対してカスタムスクリプトを実行する;
- 変更されていない値を破棄してデータベーストレージを最適化する。

1 つのアイテムに対して、1 つ以上の前処理ステップを設定できます。これらのステップは、設定された順序で実行されます。

Attention:
前処理ステップが失敗すると、アイテムはサポートされていない状態になります。これは、*Custom on fail* エラーハンドリング（ほとんどの変換で利用可能）を使用することで回避でき、値を破棄したりカスタム値を設定したりできます。

ログアイテムの場合、ログメタデータ（値を含まない）によって常にアイテムのサポートされていない状態がリセットされ、再びサポートされる状態になります。これは、最初のエラーがエージェントからログ値を受信した後に発生した場合でも同様です。

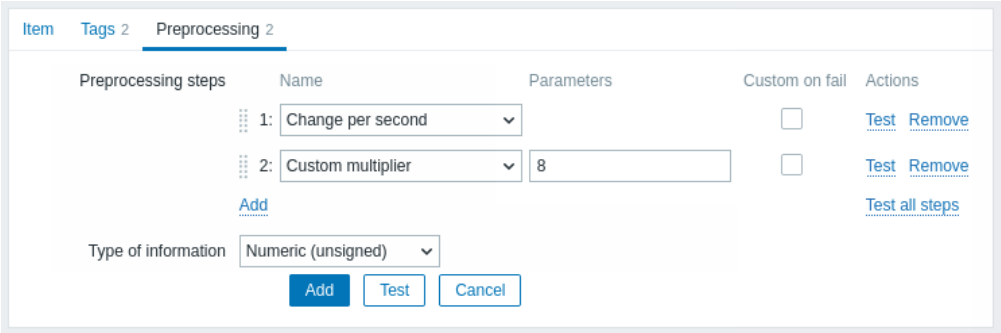
前処理に渡されるすべての値は、最初は文字列として扱われます。目的の値の型（アイテム設定で定義）への変換は、前処理パイプラインの最後に実行されます。ただし、特定の前処理ステップでは、必要に応じてそれより前に変換が行われる場合があります。詳細な技術情報については、前処理の詳細を参照してください。

前処理の設定が期待どおりに動作することを確認するには、テストできます。

関連項目: 前処理の例

設定

前処理ステップは、アイテムの設定フォームの前処理タブで定義します。



追加をクリックして、サポートされている変換を選択します。

少なくとも 1 つの前処理ステップが定義されている場合、情報の型フィールドがタブの下部に表示されます。必要に応じて、前処理タブを離れることなく情報の型を変更できます。パラメータの詳細な説明については、アイテムの作成を参照してください。

サポートされている変換

サポートされているすべての変換を以下に示します。変換名をクリックすると、その詳細を確認できます。

Name	Description	Type
正規表現	値を正規表現に一致させ、必要な出力に置き換えます。	テキス
置換	検索文字列を見つけて、別の文字列（または何もない状態）に置き換えます。	
トリム	値の先頭と末尾から指定した文字を削除します。	
右トリム	値の末尾から指定した文字を削除します。	
左トリム	値の先頭から指定した文字を削除します。	
XML XPath	XPath 機能を使用して、XML データから値またはフラグメントを抽出します。	構造化
JSON Path	JSON データから、JSONPath functionality を使用して値またはフラグメントを抽出します。	
CSV から JSON	CSV ファイルデータを JSON 形式に変換します。	
XML から JSON	XML 形式のデータを JSON に変換します。	
SNMP walk 値	指定した OID/MIB 名で値を抽出し、書式設定オプションを適用します。	SNMP
SNMP walk から JSON	SNMP 値を JSON に変換します。	
SNMP get 値	SNMP get 値に書式設定オプションを適用します。	
カスタム乗数	値に指定した整数または浮動小数点数を乗算します。	算術
単純変化	現在の値と前回の値の差を計算します。	変化
1 秒あたりの変化	値の変化速度（現在の値と前回の値の差）を 1 秒あたりで計算します。	
ブール値から 10 進数	値をブール形式から 10 進数に変換します。	数値体
8 進数から 10 進数	値を 8 進数形式から 10 進数に変換します。	
16 進数から 10 進数	値を 16 進数形式から 10 進数に変換します。	
JavaScript	JavaScript コードを入力します。	カスタ
範囲内	値が入るべき範囲を定義します。	検証
正規表現に一致する	値が一致しなければならない正規表現を指定します。	
正規表現に一致しない	値が一致してはならない正規表現を指定します。	
JSON 内のエラーを確認	JSONPath にあるアプリケーションレベルのエラーメッセージを確認します。	
XML 内のエラーを確認	XPath にあるアプリケーションレベルのエラーメッセージを確認します。	
正規表現を使用してエラーを確認	正規表現を使用してアプリケーションレベルのエラーメッセージを確認します。	
未サポート値を確認	アイテム値を取得できなかったかどうかを確認します。	

Name	Description	Type
未変更の場合は破棄	値が変更されていない場合は破棄します。	スロッ
ハートビート付きで未変更の場合は破棄	定義された期間内に値が変更されていない場合は破棄します。	
Prometheus パターン	次のクエリを使用して、Prometheus メトリクスから必要なデータを抽出します。	Promet
Prometheus から JSON	必要な Prometheus メトリクスを JSON に変換します。	

Change および *Throttling* の前処理ステップでは、新しい値を必要に応じて計算/比較するために、Zabbix が最後の値を記憶しておく必要があることに注意してください。これらの前回値は、前処理マネージャーによって処理されます。Zabbix サーバーまたはプロキシが再起動された場合、あるいは前処理ステップに何らかの変更が加えられた場合、対応するアイテムの最後の値はリセットされ、その結果、次のようになります。

- *Simple change*、*Change per second* ステップでは、変化を計算するための前回値がないため、次の値は無視されます。
- *Discard unchanged*、*Discard unchanged with heartbeat* ステップでは、破棄ルールにより本来破棄されるべき場合であっても、次の値が破棄されることはありません。

正規表現

値を正規表現に照合し、必要な出力に置き換えます。

パラメータ:

- **pattern** - 正規表現。
- **output** - 出力書式テンプレート。\N（N=1...9）エスケープシーケンスは、N 番目に一致したグループに置き換えられます。\0 エスケープシーケンスは、一致したテキストに置き換えられます。

コメント:

- 入力値の照合に失敗すると、アイテムは未サポートになります。
- 正規表現では、\N シーケンスを使用して最大 10 個のキャプチャグループを抽出できます。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートになりません。
- 既存の例については、正規表現 セクションを参照してください。

置換

検索文字列を見つけて、別の文字列（または何もない文字列）に置き換えます。

パラメータ:

- 検索文字列 - 検索して置換する文字列です。大文字と小文字は区別されます（必須）。
- 置換文字列 - 検索文字列を置き換える文字列です。
  置換文字列は空にすることもでき、その場合、見つかった検索文字列を実質的に削除できます。

コメント:

- 検索文字列のすべての出現箇所が置換されます。
- エスケープシーケンスを使用して、改行、キャリッジリターン、タブ、スペース”\n \r \t \s” を検索または置換できます。バックスラッシュは”\\” としてエスケープでき、エスケープシーケンスは”\\n” のようにエスケープできます。
- 改行、キャリッジリターン、タブのエスケープは、ローレベルディスカバリ中に自動的に行われます。

{#trim} のトリム

値の先頭と末尾から指定した文字を削除します。

右トリム

値の末尾から指定した文字を削除します。

左トリム

値の先頭から指定した文字を削除します。

XML XPath

XPath 機能を使用して、XML データから値またはフラグメントを抽出します。

コメント:

- このオプションを機能させるには、Zabbix サーバー（または Zabbix プロキシ）が libxml サポート付きでコンパイルされている必要があります。
- 名前空間はサポートされていません。

- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかです。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

例:

```
number(/document/item/value) #<document><item><value>10</value></item></document> から '10' を抽出します
number(/document/item/@attribute) #<document><item attribute="10"></item></document> から '10' を抽出します
/document/item #<document><item><value>10</value></item></document> から '<item><value>10</value></item>' を
```

JSON Path

JSONPath functionality を使用して、JSON データから値またはフラグメントを抽出します。

Custom on fail チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

CSV から JSON へ

CSV ファイルのデータを JSON 形式に変換します。

詳細については、CSV から JSON への前処理を参照してください。

XML から JSON へ

XML 形式のデータを JSON に変換します。

詳細については、シリアライズルールを参照してください。

Custom on fail チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

SNMP walk 値

指定した OID/MIB 名で値を抽出し、書式設定オプションを適用します:

- **Unchanged** - 16 進文字列をエスケープされていない 16 進文字列として返します（注意: 表示ヒントは引き続き適用されます）。
- **UTF-8 from hex-STRING** - 16 進文字列を UTF-8 文字列に変換します。
- **MAC from hex-STRING** - 16 進文字列が MAC アドレスとして正しいか検証し、適切な MAC アドレス文字列として返します（' ' は ':' に置き換えられます）。
- **Integer from BITS** - 16 進文字の並びで表現されたビット文字列（例: ”1A 2B 3C 4D”）の先頭 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、それ以降のバイトは無視されます。

Custom on fail チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、そのアイテムは未サポートにはなりません。

SNMP walk から JSON へ

SNMP 値を JSON に変換します。

JSON 内のフィールド名と、それに対応する SNMP OID パスを指定します。フィールド値には、指定した SNMP OID パス内の値が設定されます。

コメント:

- *SNMP walk value* ステップと同様の値のフォーマットオプションを利用できます。
- この前処理ステップは、SNMP OID discovery に使用できます。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄する、または指定した値を設定するオプションが選択されていれば、そのアイテムは未サポートにはなりません。

SNMP 取得値

SNMP 取得値に書式設定オプションを適用します。

- **Hex-STRING** から **UTF-8** - hex-string を UTF-8 文字列に変換します。
- **Hex-STRING** から **MAC** - hex-string が MAC アドレスとして有効か検証し、適切な MAC アドレス文字列を返します（' ' は ':' に置き換えられます）。
- **BITS** から整数 - 16 進文字の並びで表現されたビット文字列（例: ”1A 2B 3C 4D”）の先頭 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、それ以降のバイトは無視されます。

Custom on fail チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

カスタム乗数

値を、指定した整数値または浮動小数点値で乗算します。

コメント:

- このオプションを使用すると、KB、MBps などで受信した値を B、Bps に変換できます。そうしないと、Zabbix はprefixes（K、M、G など）を正しく設定できません。
- アイテムの情報の型が *Numeric (unsigned)* の場合、小数部を含む受信値はカスタム乗数が適用される前に切り捨てられることに注意してください（つまり、'0.9' は'0' になります）。
- 情報の型が *Numeric (unsigned)* に設定されているアイテムで、カスタム乗数を使用するか、保存する値として *Change per second* を使用し、その結果として計算値が実際には浮動小数点数になる場合でも、計算値は小数部を切り捨てて整数として保存されることで、正しい値として受け入れられます。
- サポート対象: たとえば 1e+70 のような科学表記、ユーザーマクロおよび LLD マクロ、ならびに {#MACRO}e+10、{$MACRO1}e+{$MACRO2} のようにマクロを含む文字列。マクロは整数または浮動小数点数に解決される必要があります。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかです。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、そのアイテムは未サポートにはなりません。

単純な変化量

現在の値と前回の値の差を計算します。

コメント:

- このステップは、継続的に増加する値の測定に役立ちます。
- **value**-**prev_value** として評価されます。ここで、*value* は現在の値、*prev_value* は前回受信した値です。
- 1 つのアイテムにつき、変更操作（「単純な変化量」または「1 秒あたりの変化量」）は 1 つだけ許可されます。
- 現在の値が前回の値より小さい場合、Zabbix はその差を破棄し（何も保存せず）、次の値を待ちます。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。
  前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

1 秒あたりの変化量

値の変化量（現在の値と前回の値の差）の 1 秒あたりの速度を計算します。

コメント:

- このステップは、継続的に増加する値の 1 秒あたりの速度を計算するのに便利です。
- この計算では浮動小数点数が生成される可能性があるため、受信する生の値が整数であっても、'Type of information' は *Numeric (float)* に設定することを推奨します。これは、小数部分が重要になる小さな数値で特に重要です。一方、浮動小数点値が大きく、'float' フィールド長を超えて値全体が失われる可能性がある場合は、*Numeric (unsigned)* を使用し、小数部分のみを切り捨てることを推奨します。
- (**value**-**prev_value**)/(**time**-**prev_time**) として評価されます。ここで、*value* は現在の値、*prev_value* は前回受信した値、*time* は現在のタイムスタンプ、*prev_time* は前回の値のタイムスタンプです。
- 1 つのアイテムにつき、変更操作（"Simple change" または"Change per second"）は 1 つだけ使用できます。
- 現在の値が前回の値より小さい場合、Zabbix はその差分を破棄し（何も保存せず）、次の値を待ちます。これは、たとえば 32 ビット SNMP カウンターのラップアラウンド（オーバーフロー）に対して正しく動作させるのに役立ちます。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかです。前処理ステップが失敗した場合でも、値を破棄する、または指定した値を設定するオプションが選択されていれば、そのアイテムは未サポートにはなりません。

ブール値を 10 進数へ

値をブール形式から 10 進数に変換します。

コメント:

- テキスト表現は 0 または 1 に変換されます。
  したがって、'TRUE' は 1 として保存され、'FALSE' は 0 として保存されます。
  すべての値は大文字と小文字を区別せずに照合されます。
  現在認識される値は、*TRUE* の場合は true, t, yes, y, on, up, running, enabled, available, ok, master、*FALSE* の場合は false, f, no, n, off, down, unused, disabled, unavailable, err, slave です。
  さらに、0 以外の数値は TRUE と見なされ、0 は FALSE と見なされます。

- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。
  前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

8 進数から 10 進数へ

値を 8 進数形式から 10 進数に変換します。

失敗時のカスタム処理チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理のステップが失敗した場合でも、値を破棄する、または指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

16 進数から 10 進数へ

値を 16 進数形式から 10 進数に変換します。

失敗時のカスタムチェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

JavaScript

パラメーターフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターに、JavaScript コードを入力します。

> **Attention:**
> 前処理の JavaScript では、宣言されていない代入を使用しないでください。ローカル変数の宣言には `var` を使用してください。

コメント:

- 使用可能な JavaScript の長さは、使用しているデータベース によって異なります。
- 詳細については、JavaScript による前処理 を参照してください。

範囲内

最小値/最大値（両端を含む）を指定して、値が入るべき範囲を定義します。

コメント:

- 数値を使用できます（桁数は任意、小数部は任意、指数部は任意、負の値も可）。
- 最小値は最大値より小さくなければなりません。
- 少なくとも 1 つの値が存在している必要があります。
- ユーザーマクロおよびローレベルディスカバリマクロを使用できます。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。

前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、そのアイテムは未サポートにはなりません。

正規表現に一致

値が一致しなければならない正規表現を指定します。

Custom on fail チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、そのアイテムは未サポートにはなりません。

正規表現 {#doesnotmatchregexp} に一致しない

値が一致してはならない正規表現を指定します。

失敗時のカスタムチェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

JSON 内のエラーをチェック

JSONPath にあるアプリケーションレベルのエラーメッセージをチェックします。成功し、かつメッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値を使用して処理を続行します。

コメント:

- これらの外部サービスエラーは、前処理ステップの情報を追加せずに、そのままユーザーに報告されます。

- 無効な JSON の解析に失敗した場合、エラーは報告されません。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄する、または指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

XML 内のエラーを確認

XPath にあるアプリケーションレベルのエラーメッセージを確認します。成功し、かつメッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。

コメント:

- これらの外部サービスエラーは、前処理ステップの情報を追加せずに、そのままユーザーに報告されます。
- 無効な XML の解析に失敗した場合、エラーは報告されません。
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。前処理ステップが失敗した場合でも、値を破棄する、または指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

正規表現を使用したエラーチェック

正規表現を使用して、アプリケーションレベルのエラーメッセージをチェックします。
成功し、かつメッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。

パラメータ:

- **pattern** - 正規表現;
- **output** - 出力書式テンプレート。
  \N（N=1...9）エスケープシーケンスは、N 番目に一致したグループに置き換えられます。
  \0 エスケープシーケンスは、一致したテキストに置き換えられます。

コメント:

- これらの外部サービスエラーは、前処理ステップの情報を追加せずに、そのままユーザーに報告されます;
- *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定する、のいずれかを選択できます。
  前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートにはなりません。

未サポートの値をチェック

アイテムの値を取得できなかったかどうかを確認します。
返されたエラーメッセージを調べて、失敗をどのように処理するかを指定します。

パラメータ:

- **scope** - エラー処理の範囲を選択します:
any error - 任意のエラー;
error matches - pattern で指定した正規表現に一致するエラーのみ;
error does not match - pattern で指定した正規表現に一致しないエラーのみ
- **pattern** - エラーを照合するための正規表現です。
  scope パラメータで *any error* が選択されている場合、このフィールドは表示されません。
  表示される場合、このフィールドは必須です。

コメント:

- 通常、値が存在しない、または値の取得に失敗すると、アイテムは未サポートになります。
  この前処理ステップを使用すると、この動作を変更できます。
  Custom on fail チェックボックスをオンにすると（この前処理ステップでは常にオンかつグレーアウトされています）、カスタムのエラー処理オプションを指定できます。具体的には、値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
  前処理ステップが失敗した場合でも、値を破棄するオプションまたは指定した値を設定するオプションが選択されていれば、アイテムは未サポートになりません。
- この前処理ステップは、アイテムの値を取得できなかったかどうかのみを確認します。
  たとえば、取得した値の型（例: string）がアイテムの情報の型（例: numeric）と一致するかどうかは確認しません。詳細はPreprocessing examples を参照してください。
  型の不一致がある場合、すべての前処理ステップの実行後にアイテムが未サポートになることがあります。
  型の不一致を確認するには、たとえば *Custom multiplier* 前処理ステップを使用できます。詳細はPreprocessing examples を参照してください。
- 正規表現のキャプチャグループは、*Set value to* または *Set error to* フィールドで使用できます。
  N 番目に一致したグループを取得するには \N（N=1...9）を使用し、一致したテキスト全体を取得するには \0 を使用します。
- これらのステップは常に最初の前処理ステップとして実行され、アイテムへの変更を保存した後は、他のすべてのステップより上に配置されます。

- 複数の *Check for not supported value* ステップを、指定した順序で使用できます。
  any error 用のステップは、このグループ内の最後のステップとして自動的に配置されます。

未変更の値を破棄

値が変更されていない場合、その値を破棄します。

コメント:

- 値が破棄された場合、その値はデータベースに保存されず、Zabbix サーバーはその値を受信したことを認識しません。その結果、トリガー条件式は評価されず、関連するトリガーに対して障害の作成/復旧は行われません。関数は、実際にデータベースに保存されたデータのみに基づいて動作します。トレンドはデータベース内のデータに基づいて作成されるため、1 時間の間に保存された値がない場合、その時間のトレンドデータも存在しません。
- アイテムごとに指定できるスロットリングオプションは 1 つだけです。

ハートビート付きで未変更値を破棄

定義された期間内（秒単位）に値が変化していない場合、その値を破棄します。

コメント:

- 秒数を指定する正の整数値をサポートしています（最小値は 1 秒）。
- 時間のサフィックスを使用できます（例: 30s、1m、2h、1d）。
- ユーザーマクロおよびローレベルディスカバリマクロを使用できます。
- 値が破棄された場合、その値はデータベースに保存されず、Zabbix サーバーはその値を受信したことを認識しません。トリガー条件式は評価されないため、関連するトリガーに対して障害の生成や復旧は行われません。関数は、実際にデータベースに保存されたデータに基づいてのみ動作します。トレンドはデータベース内のデータに基づいて作成されるため、1 時間の間に保存された値がない場合、その 1 時間のトレンドデータも存在しません。
- アイテムごとに指定できるスロットリングオプションは 1 つだけです。

Prometheus パターン

以下のクエリを使用して、Prometheus メトリクスから必要なデータを抽出します。

詳細については、Prometheus チェックを参照してください。

Prometheus から JSON へ

必要な Prometheus メトリクスを JSON に変換します。

詳細については、Prometheus チェックを参照してください。

マクロのサポート

ユーザーマクロおよびコンテキスト付きユーザーマクロは、以下でサポートされています。

- JavaScript コードを含む、前処理ステップのパラメータ
- カスタムエラー処理のパラメータ（*Set value to* フィールドおよび *Set error to* フィールド）

Note:
マクロがその値に置き換えられる際、マクロのコンテキストは無視されます。マクロ値はコード内にそのまま挿入されるため、JavaScript コード内に値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があることにご注意ください。

テスト

前処理のテストを参照してください。

1 保存前処理のテスト

テスト

前処理ステップのテストは、複雑な前処理パイプラインが期待どおりの結果を生成することを、アイテムの値が受信されて前処理されるのを待たずに確認するのに役立ちます。

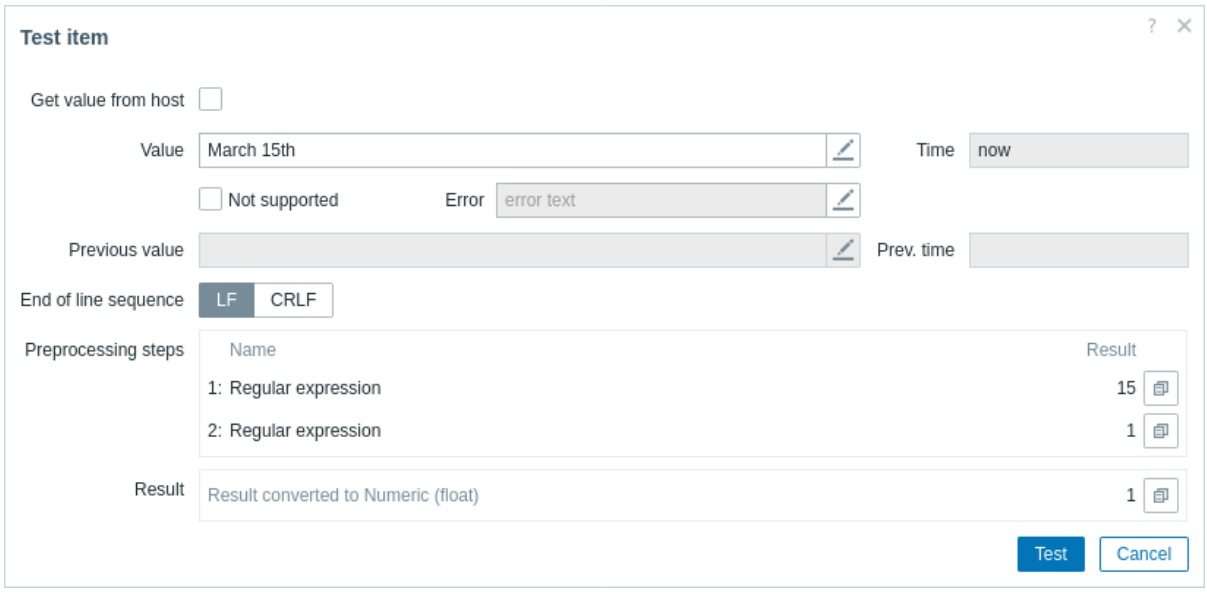


次の内容をテストできます。

- 仮想的な値に対して
- ホストからの実際の値に対して

各前処理ステップは個別にテストでき、すべてのステップをまとめてテストすることもできます。Actions ブロックでそれぞれ *Test* または *Test all steps* ボタンをクリックすると、テスト用ウィンドウが開きます。

仮想値のテスト



パラメータ	説明
ホストから値を取得	仮想値をテストする場合は、このチェックボックスをオフのままにします。 あわせて参照: 実際の値のテスト。
値	テストする入力値を入力します。 パラメーターフィールド内をクリックするか、表示/編集ボタン をクリックすると、値またはコードブロックを入力するためのテキストエリアウィンドウが開きます。
未サポート	未サポートの値をテストするには、このチェックボックスをオンにします。 このオプションは、前処理ステップ 未サポート値をチェックのテストに役立ちます。
エラー	エラーテキストを入力します。 このフィールドは、ホストから値を取得がオフで、かつ 未サポートがオンの場合に有効になります。 ホストから値を取得がオンの場合、このフィールドにはホストからの実際のエラーメッセージが自動入力されます（読み取り専用）。
時刻	入力値の時刻が表示されます: `now`（読み取り専用）。
前回の値	比較対象となる前回の入力値を入力します。 差分および スロットリングの前処理ステップでのみ使用されます。
前回の時刻	比較対象となる前回の入力値の時刻を入力します。 差分および スロットリングの前処理ステップでのみ使用されます。 デフォルト値は、アイテムの「更新間隔」フィールドの値に基づきます（たとえば'1m' の場合、このフィールドには `now-1m` が入力されます）。何も指定されていない場合、またはユーザーにホストへのアクセス権がない場合のデフォルトは `now-30s` です。
マクロ	マクロが使用されている場合、それらが値とともに一覧表示されます。値はテスト目的で編集できますが、変更はテストのコンテキスト内でのみ保存されます。
行末シーケンス	複数行の入力値に対する行末シーケンスを選択します: **LF** - LF（ラインフィード）シーケンス **CRLF** - CRLF（キャリッジリターン + ラインフィード）シーケンス。

パラメータ	説明
前処理ステップ	前処理ステップが一覧表示され、テストボタンをクリックすると各ステップのテスト結果が表示されます。 テスト結果は、Web インターフェースに送信される際に最大 512KB に切り詰められます。テスト結果はコピーできます（切り詰め後の 512KB まで）。結果が切り詰められた場合は警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも、Zabbix サーバーでは引き続き完全に処理されることに注意してください。 テスト中にステップが失敗した場合は、エラーアイコンが表示されます。エラーの説明はマウスオーバーで表示されます。 ステップに対して”Custom on fail” が指定され、そのアクションが実行された場合は、前処理テストステップの行の直後に新しい行が表示され、実行されたアクションとその結果（エラーまたは値）が示されます。
結果	すべてのステップをまとめてテストした場合（すべてのステップをテストボタンをクリックした場合）は、前処理ステップの最終結果が常に表示されます。 アイテムの値の型への変換タイプも表示されます。たとえば `Result converted to Numeric (unsigned)` のように表示されます。 テスト結果は、Web インターフェースに送信される際に最大 512KB に切り詰められます。最終結果はコピーできます（切り詰め後の 512KB まで）。結果が切り詰められた場合は警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも、Zabbix サーバーでは引き続き完全に処理されることに注意してください。

テストをクリックすると、各前処理ステップ後の結果を確認できます。

テスト値は、個別ステップのテストでも全ステップのテストでも、テストセッション間で保持されます。これにより、ユーザーは前処理ステップやアイテム設定を変更した後、情報を再入力することなくテストウィンドウに戻ることができます。ただし、ページを更新すると値は失われます。

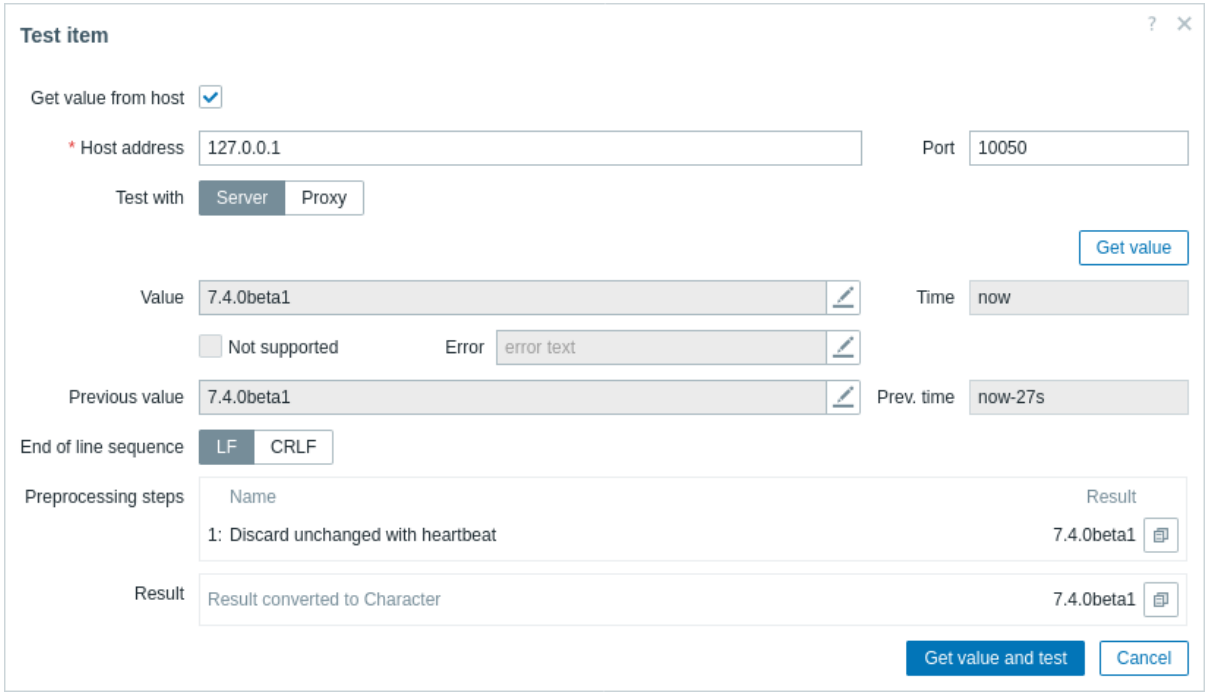
テストは Zabbix サーバーによって実行されます。Web インターフェースは対応するリクエストをサーバーに送信し、結果を待機します。リクエストには、入力値と前処理ステップ（展開されたユーザーマクロを含む）が含まれます。差分および スロットリングステップでは、任意で前回の値と時刻を指定できます。サーバーは各前処理ステップの結果を返します。

技術的なエラーまたは入力検証エラーはすべて、テストウィンドウ上部のエラーボックスに表示されます。

実際の値のテスト

実際の値に対して前処理をテストするには、次のようにします。

- ホストから値を取得チェックボックスをオンにします
- ホストのパラメータ（ホストアドレス、ポート、プロキシ名/プロキシなし）およびアイテム固有の詳細（SNMPv2 コミュニティや SNMPv3 セキュリティ認証情報など）を入力または確認します。これらのフィールドはコンテキストに応じて変化します。
  - 可能な場合は値が事前入力されます。例えば、エージェントを必要とするアイテムでは、ホストの選択されたエージェントインターフェースの情報が使用されます
  - テンプレートのアイテムでは、値を手動で入力する必要があります
  - プレーンテキストのマクロ値は展開されます
  - フィールドの値（または値の一部）がシークレットまたは Vault マクロである場合、そのフィールドは空になり、手動で入力する必要があります。アイテムのいずれかのパラメータにシークレットマクロ値が含まれている場合は、次の警告メッセージが表示されます: 「アイテムにはシークレット値を持つユーザー定義マクロが含まれています。これらのマクロの値は手動で入力する必要があります。」
  - アイテムタイプのコンテキストで不要な場合、フィールドは無効になります（例: 計算アイテムではホストアドレスおよびプロキシのフィールドは無効になります）
- 値を取得してテストをクリックして前処理をテストします



アイテム設定フォームで値のマッピング（「Show value」フィールド）を指定している場合、アイテムテストダイアログには最終結果の後に、`Result with value map applied` という名前の別の行が表示されます。

ホストから実際の値を取得する場合に固有のパラメータ:

Parameter	Description
Get value from host	このチェックボックスをオンにすると、ホストから実際の値を取得します。
Host address	ホストアドレスを入力します。 このフィールドには、アイテムのホストインターフェースのアドレスが自動的に入力されます。
Port	ホストのポートを入力します。 このフィールドには、アイテムのホストインターフェースのポートが自動的に入力されます。
Additional fields for SNMP interfaces (SNMP version, SNMP community, Context name, etc.)	SNMP インターフェース（v1、v2、v3）の設定に関する追加の詳細は、SNMP 監視の設定を参照してください。 これらのフィールドには、アイテムのホストインターフェースから自動的に値が入力されます。
Proxy	ホストがプロキシによって監視されている場合は、プロキシを指定します。 このフィールドには、ホストのプロキシ（存在する場合）が自動的に入力されます。
Value	ホストから取得した値です。 パラメータフィールド内、または表示/編集ボタン をクリックすると、値またはコードブロックのテキストエリアウィンドウが開きます。 値は最大 512KB までに切り詰められますが、これは Web インターフェース上のみです。結果が切り詰められた場合は、警告アイコンが表示されます。警告の説明はマウスオーバーで表示されます。512KB を超えるデータも、Zabbix サーバーでは引き続き完全に処理されることに注意してください。

その他のパラメータについては、上記の仮想的な値のテストを参照してください。

2 保存前処理の詳細

概要

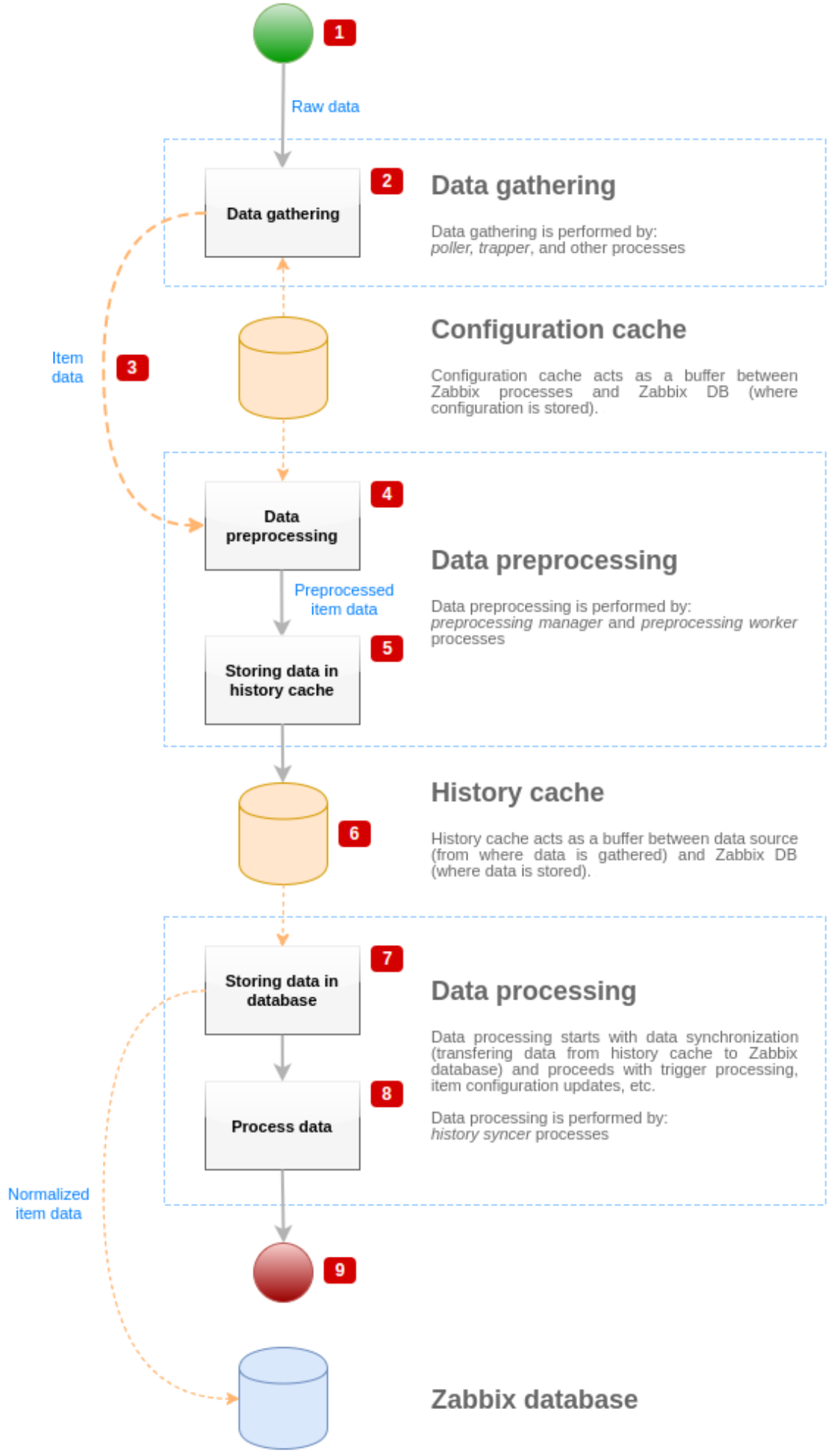
このセクションでは、アイテム値の前処理の詳細について説明します。アイテム値の前処理では、受信したアイテム値に対して変換ルールを定義し、実行できます。

前処理は、前処理ステップを実行する前処理ワーカーとともに、preprocessing manager プロセスによって管理されます。前処理が設定されたすべての値（Zabbix 7.4.1 より前では、すべての値）は、さまざまなデータ収集プロセスから受信された後、履歴キャッシュに追加される前に preprocessing manager を通過します。データ収集プロセス（poller、trapper など）と前処理プロセスの間で

は、ソケットベースの IPC 通信が使用されます。前処理ステップは、Zabbix サーバー、または Zabbix プロキシ（プロキシによって監視されるアイテムの場合）のいずれかによって実行されます。

アイテム値の処理

データソースから Zabbix データベースまでのデータフローを視覚化するために、以下の簡略化した図を使用できます。


1
Raw data
Data gathering
2
Data gathering
Data gathering is performed by:
poller, trapper, and other processes
Item data
3
Configuration cache
Configuration cache acts as a buffer between Zabbix processes and Zabbix DB (where configuration is stored).
Data preprocessing
4
Data preprocessing
Data preprocessing is performed by:
preprocessing manager and preprocessing worker processes
Preprocessed item data
Storing data in history cache
5
History cache
6
History cache acts as a buffer between data source (from where data is gathered) and Zabbix DB (where data is stored).
Storing data in database
7
Data processing
Data processing starts with data synchronization (transfering data from history cache to Zabbix database) and proceeds with trigger processing, item configuration updates, etc.
Process data
8
Data processing is performed by:
history syncer processes
Normalized item data
9
Zabbix database

上の図は、アイテム値の処理に関連するプロセス、オブジェクト、およびアクションのみを簡略化した形で示しています。この図には、条件による方向の変化、エラー処理、ループは示されていません。また、前処理マネージャーのローカルデータキャッシュも、データフローに直接影響しないため示されていません。この図の目的は、アイテム値の処理に関与するプロセスと、それらがどのように相互作用するかを示すことです。

- データ収集は、データソースからの生データから始まります。この時点では、データには ID、タイムスタンプ、および値のみが含まれます（複数の値を含む場合もあります）。
- どの種類のデータ収集プロセスを使用する場合でも、アクティブチェック、パッシブチェック、trapper アイテムなどに対する考え方は同じです。変わるのはデータ形式と通信の開始側だけです（データ収集プロセスが接続とデータを待機するか、またはデータ収集プロセスが通信を開始してデータを要求するかのいずれかです）。生データは検証され、アイテム設定が設定キャッシュから取得されます（データに設定情報が付加されます）。
- ソケットベースの IPC メカニズムを使用して、データ収集プロセスから前処理マネージャーへデータが渡されます。この時点で、データ収集プロセスは前処理マネージャーからの応答を待たずにデータ収集を継続します。
- データの前処理が実行されます。これには、前処理ステップの実行と依存アイテムの処理が含まれます。

Note:
いずれかの前処理ステップが失敗した場合、前処理の実行中にアイテムの状態が NOT SUPPORTED に変わることがあります。

- 前処理マネージャーのローカルデータキャッシュにある履歴データは、履歴キャッシュにフラッシュされます。
- この時点で、履歴キャッシュの次回同期までデータフローは停止します（履歴同期プロセスがデータ同期を実行する時点）。
- 同期プロセスは、データを Zabbix データベースに保存する前のデータ正規化から始まります。データ正規化では、目的のアイテム型（アイテム設定で定義された型）への変換が実行されます。これには、それらの型で許可される定義済みサイズに基づくテキストデータの切り詰めも含まれます（string では HISTORY_STR_VALUE_LEN、text では HISTORY_TEXT_VALUE_LEN、log 値では HISTORY_LOG_VALUE_LEN）。正規化が完了すると、データは Zabbix データベースに送信されます。

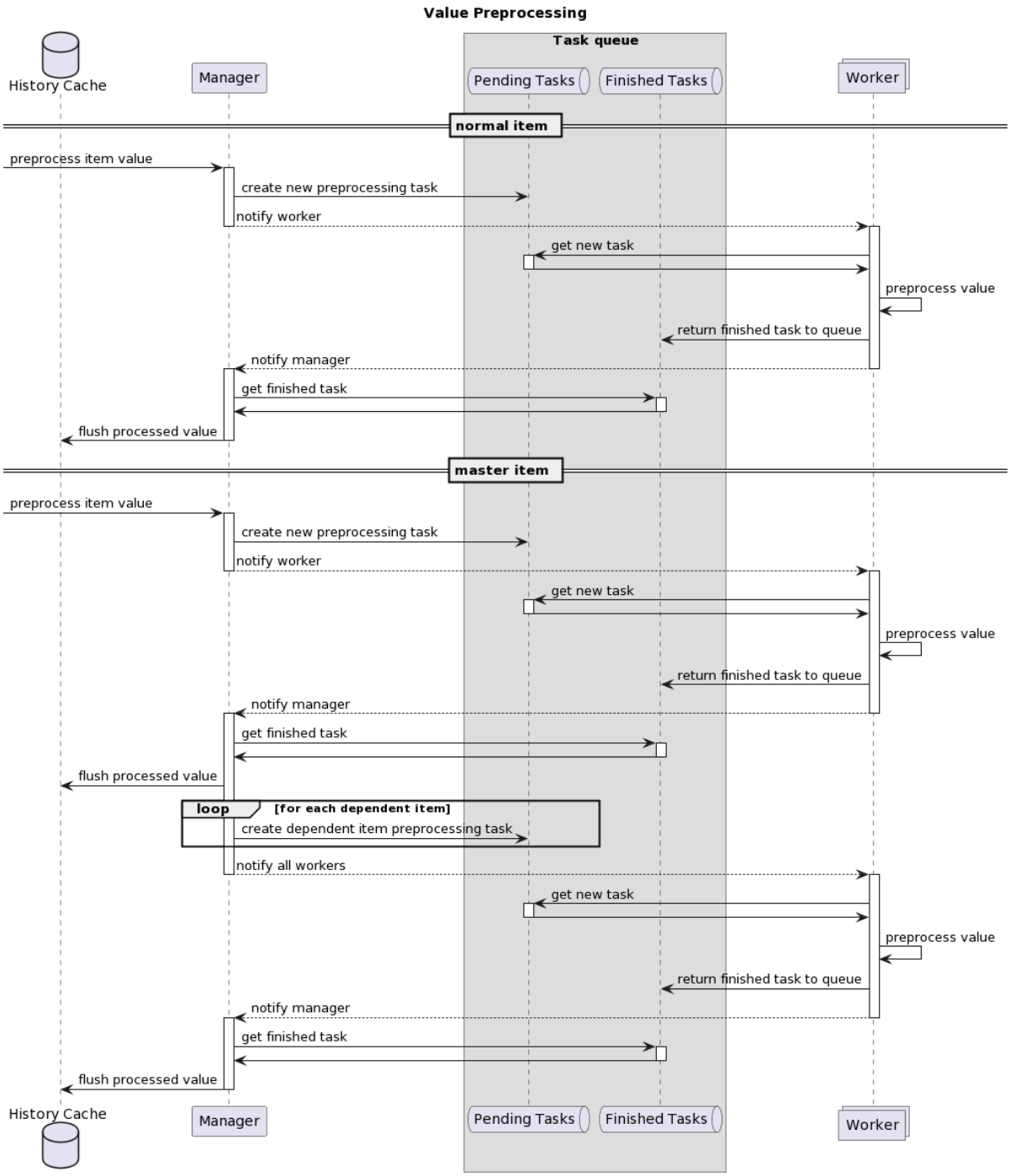
Note:
データ正規化に失敗した場合、アイテムの状態が NOT SUPPORTED に変わることがあります（たとえば、テキスト値を数値に変換できない場合）。

- 収集されたデータが処理されます。トリガーがチェックされ、アイテムが NOT SUPPORTED になった場合はアイテム設定が更新されるなどの処理が行われます。
- これは、アイテム値の処理という観点では、データフローの終点と見なされます。

アイテム値の前処理

データの前処理は、次の手順で実行されます。

- アイテムに前処理も依存アイテムも設定されていない場合、その値は履歴キャッシュに追加されるか、LLD マネージャーに送信されます。それ以外の場合、アイテム値は UNIX ソケットベースの IPC メカニズムを使用して前処理マネージャーに渡されます（Zabbix 7.4.1 より前では、すべての値が履歴キャッシュに追加されるか LLD マネージャーに送信される前に、前処理マネージャーを経由します）。
- 前処理タスクが作成され、キューに追加されます。その後、新しいタスクについて前処理ワーカーに通知されます。
- この時点で、少なくとも 1 つの未使用（つまり、どのタスクも実行していない）前処理ワーカーが存在するまで、データフローは停止します。
- 前処理ワーカーが利用可能になると、キューから次のタスクを取得します。
- 前処理が完了すると（前処理ステップの実行が失敗した場合も成功した場合も）、前処理済みの値が完了済みタスクキューに追加され、マネージャーに新しい完了済みタスクが通知されます。
- 前処理マネージャーは結果を目的の形式（アイテムの値の型で定義）に変換し、それを履歴キャッシュに追加するか、LLD マネージャーに送信します。
- 処理対象のアイテムに依存アイテムがある場合、前処理済みのマスターアイテム値とともに依存アイテムが前処理キューに追加されます。依存アイテムは通常の値前処理リクエストをバイパスしてキューに入れられますが、これは値が設定されており、かつ NOT SUPPORTED 状態ではないマスターアイテムに対してのみ行われます。



図では、前処理キャッシュを省略しているため、マスターアイテムの前処理はやや簡略化されていることに注意してください。

前処理キュー

前処理キューは、次のように構成されています。

- 保留中のタスクのリスト:
  - 受信した順序で、値の前処理リクエストから直接作成されたタスク
- 即時タスクのリスト（保留中のタスクより先に処理されます）:
  - テストタスク（Web インターフェースによるアイテム/前処理のテストリクエストへの応答として作成されるタスク）
  - 依存アイテムのタスク
  - シーケンスタスク（厳密な順序で実行する必要があるタスク）:
    - 最後の値を使用する前処理ステップを持つもの:
      - 変化
      - スロットリング
      - JavaScript（バイトコードキャッシュ）
    - 依存アイテムの前処理キャッシュ
- 完了したタスクのリスト

前処理キャッシュ

前処理キャッシュは、類似した前処理ステップを持つ複数の依存アイテムの前処理パフォーマンスを向上させるために導入されました（これは一般的な LLD の結果です）。

キャッシュは、1 つの依存アイテムを前処理し、その内部前処理データの一部を残りの依存アイテムで再利用することで実現されます。前処理キャッシュは、以下のタイプの最初の前処理ステップでのみサポートされます。

- Prometheus パターン（メトリクスによって入力にインデックスを付ける）
- JSONPath（データをオブジェクトツリーに解析し、最初の式 `[?(@.path == "value")]` にインデックスを付ける）

前処理ワーカー

Zabbix サーバーの設定ファイルでは、前処理ワーカースレッド数を設定できます。前処理ワーカーの事前起動インスタンス数を設定するには、StartPreprocessors 設定パラメータを使用します。この値は、少なくとも利用可能な CPU コア数と同じにする必要があります。

前処理タスクが CPU 負荷型ではなく、頻繁なネットワークリクエストを伴う場合は、追加のワーカーを設定することを推奨します。前処理ワーカーの最適な数は、「前処理可能な」アイテム数（何らかの前処理ステップの実行が必要なアイテム）、データ収集プロセス数、アイテム前処理の平均ステップ数など、多くの要因によって決まります。ワーカー数が不足すると、メモリ使用量が高くなる可能性があります。Zabbix 環境で過剰なメモリ使用量をトラブルシュートする方法については、tcmalloc を使用した過剰なメモリ使用量のプロファイリング を参照してください。

ただし、大きな XML/JSON チャンクの解析のような重い前処理処理がないと仮定すると、前処理ワーカー数はデータ収集プロセスの総数と同じにできます。こうすることで、通常は（収集プロセスからのデータが一括で到着する場合を除き）収集データを処理するための未使用の前処理ワーカーが少なくとも 1 つ存在することになります。

Warning:
データ収集プロセス（poller、unreachable poller、ODBC poller、HTTP poller、Java poller、pinger、trapper、proxypoller）が多すぎる場合、さらに IPMI manager、SNMP trapper、前処理ワーカーが加わることで、前処理マネージャーに対するプロセスごとのファイルディスクリプタ上限を使い果たす可能性があります。

 プロセスごとのファイルディスクリプタ上限を使い果たすと、Zabbix サーバーは停止します。通常は起動直後に発生しますが、場合によってはもう少し時間がかかることもあります。このような問題を回避するには、Zabbix サーバー設定ファイル を見直し、同時チェック数とプロセス数を最適化してください。また、必要に応じて、システムの上限を確認および調整し、ファイルディスクリプタ上限が十分に高く設定されていることを確認してください。

値処理パイプライン

アイテムの値処理は、複数のプロセスによって複数のステップ（またはフェーズ）で実行されます。これにより、次のようなことが発生する可能性があります。

- 依存アイテムは値を受信できる一方で、マスター値自体は受信できない場合があります。これは、次のユースケースで実現できます。
  - マスターアイテムの値の型は `UINT`（トラッパーアイテムを使用可能）、依存アイテムの値の型は `TEXT` です。
  - マスターアイテムと依存アイテムの両方で、前処理ステップは不要です。
  - テキスト値（たとえば”abc”）をマスターアイテムに渡します。
  - 実行する前処理ステップがないため、前処理マネージャーは、マスターアイテムが NOT SUPPORTED 状態ではなく、かつ値が設定されていることを確認し（どちらも真）、マスターアイテムと同じ値で依存アイテムをキューに入れます（前処理ステップがないため）。
  - マスターアイテムと依存アイテムの両方が履歴同期フェーズに到達すると、値変換エラーによりマスターアイテムは NOT SUPPORTED になります（テキストデータは符号なし整数に変換できないため）。

その結果、依存アイテムは値を受信しますが、マスターアイテムの状態は NOT SUPPORTED に変わります。

- 依存アイテムが、マスターアイテムの履歴に存在しない値を受信する場合があります。このユースケースは前のものと非常によく似ていますが、異なるのはマスターアイテムの型だけです。たとえば、マスターアイテムに `CHAR` 型を使用した場合、マスターアイテムの値は履歴同期フェーズで切り詰められますが、依存アイテムはマスターアイテムの初期値（切り詰め前の値）から値を受信します。

3 保存前処理の例

概要

このセクションでは、保存前処理のステップを使用していくつかの実用的なタスクを実行する例を紹介します。

VMware イベントログレコードのフィルタリング

この例では、正規表現に一致前処理ステップを使用して、VMware イベントログから不要なイベントをフィルタリングします。

1. 稼働中の VMware Hypervisor ホストで、vmware.eventlogアイテムが存在し、正常に動作していることを確認します。
イベントログアイテムは、ホスト作成時にVMwareテンプレートがリンクされている場合、すでにハイパーバイザー上に存在している可能性があることに注意してください。

2. VMware Hypervisor ホストで、ログタイプの依存アイテムを作成し、イベントログアイテムをそのマスターアイテムとして設定します。

3. 依存アイテムの前処理タブで、追加をクリックして前処理ステップを作成し、ドロップダウンから正規表現に一致を選択します。
次に、以下のいずれかのパターンを指定します。

- すべてのログイベントをフィルタリングする場合:

```
.* logged in .*
```

- "User" の後にあるユーザー名を含む行をフィルタリングする場合:

```
\bUser\s+\K\S+
```

Attention:
正規表現に一致しない場合、依存アイテムは対応するエラーメッセージとともに未サポートになります。
これを回避するには、失敗時のカスタム処理チェックボックスをオンにし、値を破棄する、またはカスタム値を設定するなどのオプションを選択してください。
破棄された値はデータベースに保存されないため、トリガーは評価されず、トレンドデータも生成されないことに注意してください。

または、正規表現前処理ステップを使用して、一致するグループを抽出し、出力を制御することもできます。

- "logged in" を含むログイベント全体を抽出して出力するには、次のパラメータを指定します:

```
Pattern: .*logged in.*
Output: \0
```

- "User" の後に続くユーザー名を抽出して出力するには、次のように指定します:

```
Pattern: User (.*?)(?=\ )
Output: \1
```

取得した値の型の確認

この例では、取得したアイテムの値の型が数値かどうかを確認するために、前処理ステップのCustom multiplier を使用します。

アイテムの *Preprocessing* タブで、前処理ステップ *Custom multiplier* を選択し、次のパラメータを指定します（取得した値に 1 を掛けます）。

```
1
```

Attention:
前処理に失敗した場合（例: 入力が数値ではない場合）、アイテムは対応するエラーメッセージとともに未サポートになります。
これを回避するには、*Custom on fail* チェックボックスをオンにし、値を破棄する、またはカスタム値を設定するなどのオプションを選択してください。
discarded 値はデータベースに保存されないため、トリガーは評価されず、トレンドデータも生成されないことに注意してください。

未サポート値のチェック

この例では、アイテムの値を取得できなかったかどうかを確認するために、前処理ステップの未サポート値のチェックを使用します。

Zabbix サーバー/プロキシの poller プロセスがアイテムの値を収集しようとすると、次のいずれかになります。

- 有効な結果を返す。
- 最初は有効に見える結果を返すが、後で未サポートになる可能性がある（例: 前処理後の値の型の不一致による）。
- 値の収集中にエラーを返し、その結果アイテムが未サポートになる。一般的な原因は次のとおりです。
  - 不明なアイテムキー（Zabbix エージェント、Simple check、または Zabbix internal アイテム）
  - 不明な OID（SNMP エージェント）、不明なセンサー（IPMI エージェント）、または JMX メトリクスがない（JMX エージェント）
  - トラップファイルを読み取れない（SNMP trap）
  - スクリプトが見つからない（External check）
  - そのような URL がない（HTTP エージェント、Browser）
  - ログイン失敗（SSH エージェント、TELNET エージェント）

- 数式の構文が無効（Calculated）、JavaScript の構文エラー（Script）、または無効な SQL（Database monitor）

アイテム値の収集エラーを検出して処理するには、前処理ステップの未サポート値のチェックを使用できます。このステップは常に最初に実行され、前処理の開始前に発生したエラーのみを検出することに注意してください。

アイテムの前処理タブで、前処理ステップとして未サポート値のチェックを選択し、次のいずれかのパラメータを指定します。

- 任意のエラーの場合:

```
Parameter: any error
```

- ”cannot connect” を含むエラーの場合:

```
Parameter: error matches
Pattern: (?i)cannot connect
```

その後、失敗時のカスタム処理オプションを使用して、値（この場合はエラー）を破棄するか、カスタム値を設定するか、カスタムエラーメッセージを返します。破棄された値はデータベースに保存されないため、トリガーは評価されず、トレンドデータも生成されないことに注意してください。

4 JSONPath 機能

概要

このセクションでは、アイテムの値の前処理ステップ内でサポートされる JSONPath 機能の概要を説明します。

JSONPath は、ドットで区切られたセグメントで構成されます。セグメントは、JSON 値の名前を表す単純な単語、ワイルドカード文字（*）、または角括弧で囲まれたより複雑な構文の形式を取ることができます。角括弧で囲まれたセグメントの前のドットは任意であり、省略できます。

JSONPath example	Description
`$.object.name`	`object.name` の内容を返します。
`$.object['name']`	`object.name` の内容を返します。
`$.object.['name']`	`object.name` の内容を返します。
`$["object"]['name']`	`object.name` の内容を返します。
`$.['object'].["name"]`	`object.name` の内容を返します。
`$.object.history.length()`	`object.history` 配列の要素数を返します。
`$[?(@.name == 'Object')].price.first()`	名前が”Object” の最初のオブジェクトから `price` プロパティの値を返します。
`$[?(@.name == 'Object')].history.first().length()`	名前が”Object” の最初のオブジェクトから history 配列の要素数を返します。
`$[?(@.price > 10)].length()`	価格が 10 より大きいオブジェクトの数を返します。

参照: JSONPath で LLD マクロ値の特殊文字をエスケープする。

サポートされているセグメント

Segment	Description
`<name>`	名前でオブジェクトのプロパティに一致します。
`*`	すべてのオブジェクトプロパティに一致します。
`['<name>']`	名前でオブジェクトのプロパティに一致します。
`['<name>', '<name>', ...]`	列挙された名前のいずれかに一致するオブジェクトプロパティに一致します。
`[<index>]`	インデックスで配列要素に一致します。
`[<number>, <number>, ...]`	列挙されたインデックスのいずれかに一致する配列要素に一致します。
`[*]`	すべてのオブジェクトプロパティまたは配列要素に一致します。
`[<start>:<end>]`	定義された範囲で配列要素に一致します: **<start>** - 一致対象の最初のインデックス（含む）。指定しない場合は、先頭からすべての配列要素に一致します。負の値の場合は、配列の末尾からの開始オフセットを指定します。 **<end>** - 一致対象の最後のインデックス（含まない）。指定しない場合は、末尾まですべての配列要素に一致します。負の値の場合は、配列の末尾からの開始オフセットを指定します。
`[?(<expression>)]`	フィルター式を適用してオブジェクト/配列要素に一致します。

一致するセグメントをその祖先を無視して検索するには（detached segment）、先頭に 2 つのドット（`..`）を付ける必要があります。たとえば、`$..name` または `$..['name']` は、すべての name プロパティの値を返します。

一致した要素名は、JSONPath にチルダ（`~`）接尾辞を追加することで抽出できます。これにより、一致したオブジェクトの名前、または一致した配列アイテムのインデックスが文字列形式で返されます。出力形式は他の JSONPath クエリと同じルールに従います。確定パスの結果はそのまま返され、不確定パスの結果は配列で返されます。ただし、確定パスに一致する要素の名前を抽出することにはほとんど意味がありません。なぜなら、その名前はすでに分かっているためです。

フィルター式

フィルター式は、中置記法による算術式です。

サポートされるオペランド:

Operand	Description

`"<text>"`
`'<text>'`
	テキスト定数。
	
	例:
	`'value: \\'1\\''`
	`"value: '1'"`

`<number>`	科学表記をサポートする数値定数。 例: 123
`<jsonpath starting with $>`	入力ドキュメントのルートノードからの JSONPath で参照される値。確定パスのみサポートされます。 例: `$.object.name`
`<jsonpath starting with @>`	現在のオブジェクト/要素からの JSONPath で参照される値。確定パスのみサポートされます。 例: `@.name`

サポートされる演算子:

Operator	Type	Description	Result
-	Binary	減算	数値
+	Binary	加算	数値
/	Binary	除算	数値
*	Binary	乗算	数値
==	Binary	等価	ブール値 (1/0)
!=	Binary	不等価	ブール値 (1/0)
	Binary	より小さい	ブール値 (1/0)
<=	Binary	以下	ブール値 (1/0)
>	Binary	より大きい	ブール値 (1/0)
>=	Binary	以上	ブール値 (1/0)
=~	Binary	正規表現に一致	ブール値 (1/0)
!	Unary	ブール否定	ブール値 (1/0)
||	Binary	ブール OR	ブール値 (1/0)
&&	Binary	ブール AND	ブール値 (1/0)

関数

関数は JSONPath の末尾で使用できます。先行する関数が後続の関数で受け入れられる値を返す場合、複数の関数を連結できます。

サポートされている関数:

関数	説明	入力	出力
`avg`	入力配列内の数値の平均値	数値の配列	数値
`min`	入力配列内の数値の最小値	数値の配列	数値
`max`	入力配列内の数値の最大値	数値の配列	数値
`sum`	入力配列内の数値の合計	数値の配列	数値
`length`	入力配列内の要素数	配列	数値

関数	説明	入力	出力
`first`	配列の最初の要素	配列	入力配列の内容に応じた JSON 構造（オブジェクト、配列、値）

JSONPath 集計関数は、引用符で囲まれた数値を受け入れます。これらの値は、集計が必要な場合に文字列から数値型へ自動的に変換されます。互換性のない入力が与えられた場合、関数はエラーを生成します。

出力値

JSONPath は、確定パスと不確定パスに分けることができます。
確定パスが返せるのは、null または単一の一致のみです。
不確定パスは複数の一致を返すことができます。これには、分離されたパス、複数の名前/インデックスのリスト、配列スライス、または式セグメントを含む JSONPath が該当します。
ただし、関数が使用される場合、JSONPath は確定になります。関数は常に単一の値を出力するためです。

確定パスは、参照しているオブジェクト/配列/値を返します。
一方、不確定パスは、一致したオブジェクト/配列/値の配列を返します。

> **Attention:**
> JSONPath クエリ結果のプロパティ順序は、内部最適化の方法により、元の JSON のプロパティ順序と一致しない場合があります。
> たとえば、JSONPath `$.books[1]["author", "title"]` は `["title", "author"]` を返すことがあります。
> 元のプロパティ順序を保持することが重要な場合は、クエリ後の別の処理方法を検討してください。

パスの書式設定ルール

空白文字（スペース、タブ文字）は、たとえば `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )` のように、ブラケット記法のセグメントや式で使用できます。

文字列は、シングルクォート（'）またはダブルクォート（"）で囲む必要があります。文字列内では、シングルクォートまたはダブルクォート（どちらで囲んでいるかに応じて）およびバックスラッシュ（\）は、バックスラッシュ文字（\）でエスケープします。

例

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sayings of the Century",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Sword of Honour",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "fiction",
      "author": "Herman Melville",
      "title": "Moby Dick",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "fiction",
      "author": "J. R. R. Tolkien",
      "title": "The Lord of the Rings",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
```

```
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": " 地元の町での翌日配達",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "A5 形式での書籍の印刷および製本",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": " さまざまな修復方法",
      "active": false,
      "methods": [
        {
          "description": " 化学洗浄",
          "price": 46
        },
        {
          "description": " 湿気で損傷したページの圧着",
          "price": 24.5
        },
        {
          "description": " 破れた本の再製本",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "no \"filters\""
  },
  "closed message": " 店舗は閉店しています",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	no ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”no \”filters\”” }
`$.books[1].title`	definite	Sword of Honour
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]

JSONPath	Type	Result
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sayings of the Century”]
`$.books[1]['author', "title"]`	indefinite	[”Sword of Honour”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] 注: このクエリは、クエリ内で算術演算を使用できることを示しています。これは `$.books[?(@.id == 2)].title` に簡略化できます。
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sayings of the Century”, ”Moby Dick”, ”The Lord of the Rings”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sayings of the Century”, ”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”The Lord of the Rings”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sayings of the Century”, ”The Lord of the Rings”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”The Lord of the Rings”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”The Lord of the Rings”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Result

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sayings of the Century”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Sword of Honour”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”fiction”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”fiction”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”The Lord of the Rings”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”A5 形式での書籍の印刷および製本”, ” 破れた本の再製本”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour
`$..tags.first().length()`	definite	5 注: `$..tags` は不定パスであるため、一致した要素の配列、つまり `[["a", "b", "c", "d", "e" ]]` を返します。`first()` は最初の要素、つまり `["a", "b", "c", "d", "e"]` を返します。`length()` はその要素の長さ、つまり 5 を計算します。
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	注: 一致しないクエリは、definite パスと indefinite パスの両方で NULL を返します。
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] 注: 真偽値の比較ではテキスト定数を使用する必要があります。
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] 注: 真偽値の比較ではテキスト定数を使用する必要があります。

`$.services[?(@.servicegroup=="1002")].first()`	definite	{
		”servicegroup”: 1002,
		”active”: false,
		”description”: ”Description 2”
		}

1 JSONPath の LLD マクロ値からの特殊文字のエスケープ

ローレベルディスカバリマクロが JSONPath の保存前処理で使用され、その値が解決される場合、次の特殊文字のエスケープ規則が適用されます。

- バックスラッシュ (\) と二重引用符 (") 文字のみがエスケープ対象として考慮されます。
- 解決されたマクロ値にこれらの文字が含まれている場合、それぞれの文字はバックスラッシュでエスケープされます。
- すでにバックスラッシュでエスケープされている場合は、エスケープとみなされず、バックスラッシュとそれに続く特殊文字の両方がさらにエスケープされます。

例:

JSONPath	LLD マクロの値	置換後
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]
	c:\temp	$.[?(@.value == "c:\\temp")]
	a\\b	$.[?(@.value == "a\\\\b")]

式で使用する場合、特殊文字を含む可能性のあるマクロを二重引用符で囲む必要があります。

JSONPath	LLD マクロの値	置換後	結果
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	不正な **JSONPath** の式

特殊文字を含むマクロをパスで使用する場合は、角括弧と二重引用符で囲む必要があります。

JSONPath	LLD マクロの値	置換後	結果
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	不正な **JSONPath** の式

5 JavaScript の保存前処理

概要

このセクションでは、JavaScript による保存前処理について詳しく説明します。

JavaScript の保存前処理

JavaScript の保存前処理は、JavaScript の関数を'value' という 1 つのパラメーターとユーザー提供の関数本体で呼び出すことで行われます。保存前処理のステップの結果は、この関数から返される値です。例えば、華氏から摂氏への変換を行う場合、次のように入力します。

```
return (value - 32) * 5 / 9
```

JavaScript の保存前処理パラメーターは、サーバーによって JavaScript 関数にラップされます。

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

入力パラメーター'value' は常に文字列として渡されます。戻り値はいくつかの例外とともに自動的に toString() メソッドで文字列に変換されます (失敗した場合は、エラーが文字列として返されます)。

- 未定義の値を返すとエラーになります。
- null 値を返すと、入力値が破棄されます。これは' 失敗時のカスタマイズ' の' 値を破棄' と似た動作になります。

エラーは、値/オブジェクト (通常は文字列かエラーオブジェクト) をスローすることによって返されます。

例:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

各スクリプトには 10 秒の実行タイムアウトがあり、超えるとエラーが返されます (スクリプトによって、タイムアウトがトリガーするのに時間がかかる場合があります)。512 メガバイトのヒープ制限が施行されています。

JavaScript の保存前処理のステップのバイトコードはキャッシュされ、次回ステップが適用されたときに再利用されます。アイテムの保存前処理ステップに変更を加えると、キャッシュされたスクリプトがリセットされ、後で再コンパイルされます。

連続したランタイムエラー (3 回連続) が発生すると、1 つのスクリプトによって次のスクリプトの実行環境が破壊される可能性を軽減するために、エンジンが再初期化されます (このアクションは、デバッグレベル 4 以上でログに記録されます)。

JavaScript の保存前処理は、Duktape JavaScript エンジンを使用して実装されています。

参照: 追加の JavaScript オブジェクトとグローバル関数

スクリプトでのマクロの使用

JavaScript コードでは、ユーザーマクロ（およびローレベルディスカバリのコンテキストにおけるLLD マクロ）を使用できます。スクリプトにマクロが含まれている場合、これらのマクロは特定の前処理ステップを実行する前にサーバー/プロキシによって展開されます。Web インターフェースで前処理ステップをテストする場合、マクロ値は取得されず、手動で入力する必要があることに注意してください。

> **Note:**
> マクロがその値に置き換えられる際、コンテキストは無視されます。マクロ値はそのままコードに挿入されるため、JavaScript コード内に値を配置する前に追加のエスケープを行うことはできません。このため、場合によっては JavaScript エラーが発生する可能性があることに注意してください。

以下の例では、受信した値が {$THRESHOLD} マクロ値を超えている場合、しきい値が返されます（存在する場合）。

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

例

次の例は、JavaScript の保存前処理の使用方法を示しています。各例には、簡単な説明、JavaScript 保存前処理パラメーターの関数本体、および保存前処理のステップの結果 (関数によって返される値) が含まれています。

例 1: 数値の変換 (科学的記数法から整数へ)

数値”2.62128e+07” を科学的記数法から整数に変換します。

```
return (Number(value))
```

関数によって返される値: 26212800

例 2: 数値の変換 (2 進数から 10 進数へ)

2 進数”11010010” を 10 進数に変換します。

```
return(parseInt(value,2))
```

関数によって返される値: 210

例 3: 数値を丸める

数値”18.2345” を 2 桁に丸めます。

```
return(Math.round(value* 100) / 100)
```

関数によって返される値: 18.23

例 4: 文字列内の文字数を数える

文字列”Zabbix” 内の文字数を数えます。

```
return (value.length)
```

関数によって返される値: 6

例 5: 残り時間を取得する

証明書の有効期限 (Feb 12 12:33:56 2022 GMT) までの残り時間 (秒単位) を取得します。

```
var split = value.split(' '),
    MONTHS_LIST = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
    month_index = ('0' + (MONTHS_LIST.indexOf(split[0]) + 1)).slice(-2),
    ISOdate = split[3] + '-' + month_index + '-' + split[1] + 'T' + split[2],
    now = Date.now();
```

```
return parseInt((Date.parse(ISOdate) - now) / 1000);
```

関数によって返される値: 44380233

例 6: JSON プロパティの削除

キー"data_size" または"index_size" を持つプロパティを削除して、JSON データ構造を変更します。

```
var obj=JSON.parse(value);

for (i = 0; i < Object.keys(obj).length; i++) {
    delete obj[i]["data_size"];
    delete obj[i]["index_size"];
}

return JSON.stringify(obj)
```

関数によって受け入れられる値:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

関数によって返される値:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

例 7: Apache ステータスを JSON に変換する

Zabbix エージェントアイテムweb.page.get(例: web.page.get[http://127.0.0.1:80/server-status?auto]) から受信した値を JSON オブジェクトに変換します。

```
// Convert Apache status to JSON

// Split the value into substrings and put these substrings into an array
var lines = value.split('\n');

// Create an empty object "output"
var output = {};

// Create an object "workers" with predefined properties
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Add the substrings from the "lines" array to the "output" object as properties (key-value pairs)
for (var i = 0; i < lines.length; i++) {
    var line = lines[i].match(/([A-z0-9 ]+): (.*)/);
```

```
    if (line !== null) {
        output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);
    }
}

// Multiversion metrics
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Parse "Scoreboard" property to get the worker count
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Add worker data to the "output" object
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// Return JSON string
return JSON.stringify(output);
```

関数によって受け入れられる値:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2023 11:08:39 GMT
Server: Apache/2.4.52 (Ubuntu)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 405
Content-Type: text/plain; charset=ISO-8859-1

127.0.0.1
ServerVersion: Apache/2.4.52 (Ubuntu)
ServerMPM: prefork
Server Built: 2023-03-08T17:32:01
CurrentTime: Monday, 27-Mar-2023 14:08:39 EEST
RestartTime: Monday, 27-Mar-2023 12:19:59 EEST
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 6520
ServerUptime: 1 hour 48 minutes 40 seconds
Load1: 0.56
Load5: 0.33
Load15: 0.28
Total Accesses: 2476
Total kBytes: 8370
Total Duration: 52718
CPUUser: 8.16
CPUSystem: 3.44
CPUChildrenUser: 0
CPUChildrenSystem: 0
CPULoad: .177914
Uptime: 6520
ReqPerSec: .379755
BytesPerSec: 3461.58
```

```
BytesPerReq: 3461.58
DurationPerReq: 21.2916
BusyWorkers: 2
IdleWorkers: 6
Scoreboard: ____KW__.....................................................................................
```

関数によって返される値:


```
{
    "Date": "Mon, 27 Mar 2023 11:08:39 GMT",
    "Server": "Apache/2.4.52 (Ubuntu)",
    "Vary": "Accept-Encoding",
    "Encoding": "gzip",
    "Length": 405,
    "Type": "text/plain; charset=ISO-8859-1",
    "ServerVersion": "Apache/2.4.52 (Ubuntu)",
    "ServerMPM": "prefork",
    "Server Built": "2023-03-08T17:32:01",
    "CurrentTime": "Monday, 27-Mar-2023 14:08:39 EEST",
    "RestartTime": "Monday, 27-Mar-2023 12:19:59 EEST",
    "ParentServerConfigGeneration": 1,
    "ParentServerMPMGeneration": 0,
    "ServerUptimeSeconds": 6520,
    "ServerUptime": "1 hour 48 minutes 40 seconds",
    "Load1": 0.56,
    "Load5": 0.33,
    "Load15": 0.28,
    "Total Accesses": 2476,
    "Total kBytes": 8370,
    "Total Duration": 52718,
    "CPUUser": 8.16,
    "CPUSystem": 3.44,
    "CPUChildrenUser": 0,
    "CPUChildrenSystem": 0,
    "CPULoad": 0.177914,
    "Uptime": 6520,
    "ReqPerSec": 0.379755,
    "BytesPerSec": 1314.55,
    "BytesPerReq": 3461.58,
    "DurationPerReq": 21.2916,
    "BusyWorkers": 2,
    "IdleWorkers": 6,
    "Scoreboard": "____KW__.....................................................................................
    "Workers": {
        "waiting": 6,
        "starting": 0,
        "reading": 0,
        "sending": 1,
        "keepalive": 1,
        "dnslookup": 0,
        "closing": 0,
        "logging": 0,
        "finishing": 0,
        "cleanup": 0,
        "slot": 142
    }
}
```


1 追加の JavaScript オブジェクト

概要

このセクションでは、Duktape で実装された JavaScript 言語に対する Zabbix の追加機能と、サポートされているグローバル

JavaScript 関数について説明します。

Attention:
前処理の JavaScript では、宣言されていない代入を使用しないでください。ローカル変数の宣言には var を使用してください。

組込オブジェクト

Zabbix

Zabbix オブジェクトは、Zabbix の内部機能とのやり取りを提供します。

Method	Description
log(loglevel, message)	<message> を <loglevel> のログレベルを使用して Zabbix ログに書き込みます（設定ファイルの DebugLevel パラメータを参照）。

例:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

以下のエイリアスを使用できます。

Alias	Alias to
console.log(object)	Zabbix.log(4, JSON.stringify(object))
console.warn(object)	Zabbix.log(3, JSON.stringify(object))
console.error(object)	Zabbix.log(2, JSON.stringify(object))

Attention:
記録されるすべてのメッセージの合計サイズは、スクリプト実行ごとに 8 MB に制限されています。

Method	Description
sleep(delay)	JavaScript の実行を delay ミリ秒だけ遅延させます。

例（実行を 15 秒遅延）:

```
Zabbix.sleep(15000)
```

HttpRequest

このオブジェクトは cURL ハンドルをカプセル化し、シンプルな HTTP リクエストを実行できるようにします。
エラーは例外としてスローされます。

Attention:
複数の HttpRequest オブジェクトの初期化は、スクリプト実行ごとに 10 個までに制限されています。

Method	Description
addHeader(value)	HTTP ヘッダーフィールドを追加します。このフィールドは、clearHeader() メソッドでクリアされるまで、以降のすべてのリクエストで使用されます。 1 つの HttpRequest オブジェクトに追加できるヘッダーフィールドの合計長は、128 Kbytes に制限されています（特殊文字およびヘッダー名を含む）。
clearHeader()	HTTP ヘッダーをクリアします。ヘッダーフィールドが設定されていない場合、HttpRequest は、送信されるデータが JSON 形式であれば Content-Type を application/json に設定し、それ以外の場合は text/plain に設定します。
connect(url)	URL に HTTP CONNECT リクエストを送信し、レスポンスを返します。
customRequest(method, url, data)	最初のパラメータで任意の HTTP メソッドを指定できます。オプションの *data* ペイロード付きで、そのメソッドのリクエストを URL に送信し、レスポンスを返します。
delete(url, data)	オプションの *data* ペイロード付きで URL に HTTP DELETE リクエストを送信し、レスポンスを返します。

Method	Description
getHeaders(<asArray>)	受信した HTTP ヘッダーフィールドのオブジェクトを返します。 asArray パラメータには”true”（例: getHeaders(true)）、”false”、または未定義を設定できます。”true” に設定した場合、受信した HTTP ヘッダーフィールドの値は配列として返されます。これは、同じ名前の複数のヘッダーのフィールド値を取得する際に使用します。 設定しない場合、または”false” に設定した場合、受信した HTTP ヘッダーフィールドの値は文字列として返されます。
get(url, data)	オプションの *data* ペイロード付きで URL に HTTP GET リクエストを送信し、レスポンスを返します。
head(url)	URL に HTTP HEAD リクエストを送信し、レスポンスを返します。
options(url)	URL に HTTP OPTIONS リクエストを送信し、レスポンスを返します。
patch(url, data)	オプションの *data* ペイロード付きで URL に HTTP PATCH リクエストを送信し、レスポンスを返します。
put(url, data)	オプションの *data* ペイロード付きで URL に HTTP PUT リクエストを送信し、レスポンスを返します。
post(url, data)	オプションの *data* ペイロード付きで URL に HTTP POST リクエストを送信し、レスポンスを返します。
getStatus()	最後の HTTP リクエストのステータスコードを返します。
setProxy(proxy)	HTTP プロキシを”proxy” の値に設定します。このパラメータが空の場合、プロキシは使用されません。
setHttpAuth(bitmask, username, password)	有効にする HTTP 認証方式（HTTPAUTH_BASIC、HTTPAUTH_DIGEST、HTTPAUTH_NEGOTIATE、HTTPAUTH_NTLM、HTTPAUTH_NONE）を’bitmask’ パラメータで設定します。 HTTPAUTH_NONE フラグを使用すると、HTTP 認証を無効にできます。 例: request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password) request.setHttpAuth(HTTPAUTH_NONE)
trace(url, data)	オプションの *data* ペイロード付きで URL に HTTP TRACE リクエストを送信し、レスポンスを返します。

例:

```
try {
    Zabbix.log(4, 'jira webhook script value='+value);

    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new HttpRequest(),
    fields = {},
    resp;

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic '+params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
    fields.issuetype = {"id": params.issue_id};
    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: '+req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
```

```
} catch (error) {
    Zabbix.log(4, 'jira issue creation failed json : '+JSON.stringify({"fields": fields}));
    Zabbix.log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);
```

XML

XML オブジェクトを使用すると、アイテムおよびローレベルディスカバリの前処理、ならびに webhook で XML データを処理できます。

Attention:
XML オブジェクトを使用するには、サーバー/プロキシが libxml2 サポート付きでコンパイルされている必要があります。

Method	Description
`XML.query(data, expression)`	XPath を使用してノードの内容を取得します。ノードが見つからない場合は null を返します。 **expression** - XPath 式。 **data** - 文字列としての XML データ。
`XML.toJson(data)`	XML 形式のデータを JSON に変換します。
`XML.fromJson(object)`	JSON 形式のデータを XML に変換します。

例:

入力:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

出力:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

シリアライズルール

XML から JSON への変換は、以下のルールに従って処理されます（JSON から XML への変換では、逆のルールが適用されます）。

1. XML 属性は、名前の先頭に'@' を付けたキーに変換されます。

例:

入力:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

出力:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. 自己終了要素（<foo/>）は、値が'null' のものとして変換されます。

例:

入力:

```
<xml>
  <foo/>
</xml>
```

出力:

```
{
  "xml": {
    "foo": null
  }
}
```

3. 空の属性（値が""）は、値が空文字列 ('') のものとして変換されます。

例:

入力:

```
<xml>
  <foo bar="" />
</xml>
```

出力:

```
{
  "xml": {
    "foo": {
      "@bar": ""
    }
  }
}
```

4. 同じ要素名を持つ複数の子ノードは、単一のキーに変換され、その値は値の配列になります。

例:

入力:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

出力:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. テキスト要素に属性も子要素もない場合、文字列として変換されます。

例:

入力:

```
<xml>
    <foo>BAZ</foo>
</xml>
```

出力:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. テキスト要素に子要素はないが属性がある場合、テキスト内容はキー'#text' を持つ要素に変換され、その内容が値になります。属性は、シリアライズルール 1 で説明したとおりに変換されます。

例:

入力:

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

出力:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

グローバル JavaScript 関数

Duktape により、追加のグローバル JavaScript 関数が実装されています。

- btoa(data) - データを base64 文字列にエンコードします。
- atob(base64_string) - base64 文字列を Uint8Array バッファとしてデコードします。

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    // Note that decoding logic depends on the data format of the buffer.
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- md5(data) - データの MD5 ハッシュを計算します。
- sha256(data) - データの SHA256 ハッシュを計算します。
- hmac('<hash type>',key,data) - HMAC ハッシュを 16 進形式の文字列として返します。hash type としては md5 および sha256 がサポートされます。key および data パラメータはバイナリデータをサポートします。

  例:

  - hmac('md5',key,data)
  - hmac('sha256',key,data)

- sign(hash,key,data) - 計算された署名（SHA-256 を使用した RSA 署名）を文字列として返します。各引数は以下のとおりです:
 hash - sha256 のみ使用可能です。それ以外の場合はエラーがスローされます。
 key - 秘密鍵です。PKCS#1 または PKCS#8 標準に対応している必要があります。鍵は以下のさまざまな形式で指定できます:

- 改行の代わりにスペースを使用
- 改行の代わりにエスケープ済みまたは非エスケープの \n を使用
- 改行を含まない 1 行の文字列
- JSON 形式の文字列

鍵は、ユーザーマクロ/シークレットマクロ/vault から読み込むこともできます。

data - 署名対象のデータです。文字列（バイナリデータもサポート）またはバッファ（Uint8Array/ArrayBuffer）を指定できます。

例:

- `sign('sha256',key,data)`

署名の計算には OpenSSL または GnuTLS が使用されます。Zabbix がこれらの暗号化ライブラリのいずれも使用せずにビルドされている場合は、エラーがスローされます（'missing OpenSSL or GnuTLS library'）。

2 ブラウザアイテムの JavaScript オブジェクト

概要

このセクションでは、ブラウザアイテム スクリプトで使用するために、Duktape で実装された JavaScript 言語に対する Zabbix の追加機能について説明します。
これらの追加機能は、追加の JavaScript オブジェクト ページで説明されている JavaScript オブジェクトを補完するものです。

Browser

`Browser` オブジェクトは WebDriver セッションを管理し、作成時にセッションを初期化し、破棄時にセッションを終了します。1 つのスクリプトで最大 4 つの `Browser` オブジェクトをサポートできます。

`Browser` オブジェクトを構築するには、`new Browser(options)` 構文を使用します。`options` (*JSON object*) パラメータはブラウザオプションを指定し、通常は WebDriver options メソッドの結果です（例: `Browser.chromeOptions()`）。

`Browser` オブジェクトでは、以下のメソッドがサポートされています。

Method	Description
`navigate(url)`	指定した URL に移動します。 パラメータ: `url` - (string) 移動先の URL。
`getUrl()`	開いているページの URL 文字列を返します。
`getPageSource()`	開いているページのページソース文字列を返します。
`findElement(strategy, selector)`	開いているページ内の 1 つの要素を持つ`Element` オブジェクトを返します (`strategy` と `selector` に一致する要素がない場合は `null` を返します)。 パラメータ: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Location strategy; `selector` - (string) 指定したロケーション戦略を使用する要素セレクター。
`findElements(strategy, target)`	開いているページ内の複数の要素を持つ`Element` オブジェクトの配列を返します（ロケーション戦略と target に一致する要素がない場合は空の配列を返します)。 パラメータ: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Location strategy; `target` - (string) 指定したロケーション戦略を使用する要素セレクター。
`getCookies()`	`Cookie` オブジェクトの配列を返します。
`addCookie(cookie)`	Cookie を設定します。 パラメータ: `cookie` - (`Cookie` object) 設定する Cookie。
`getScreenshot()`	ブラウザのビューポートの画像を表す base64 エンコード文字列を返します。

Method	Description
`setScreenSize(x,y)`	ブラウザのビューポートサイズを設定します。 パラメータ: `x` - (string) ピクセル単位の幅; `y` - (string) ピクセル単位の高さ。
`setScriptTimeout(timeout)`	スクリプト読み込みタイムアウトを設定します。 パラメータ: `timeout` - (integer) ミリ秒単位のタイムアウト値。
`setSessionTimeout(timeout)`	セッション（ページ読み込み）タイムアウトを設定します。 パラメータ: `timeout` - (integer) ミリ秒単位のタイムアウト値。
`setElementWaitTimeout(timeout)`	要素ロケーション戦略の（暗黙的）タイムアウトを設定します。 パラメータ: `timeout` - (integer) ミリ秒単位のタイムアウト値。
`collectPerfEntries(mark)`	`getResult()` メソッドで取得するパフォーマンスエントリを収集します。 パラメータ: `mark` - (string, optional) パフォーマンススナップショットのマーク。
`getRawPerfEntries()`	パフォーマンスエントリオブジェクトの配列を返します。
`getResult()`	ブラウザセッションの統計情報（エラー情報、パフォーマンススナップショットなど）を含む`Result` オブジェクトを返します。
`getError()`	ブラウザエラーを含む `BrowserError` オブジェクトを返します（ブラウザエラーがない場合は `null` を返します）。
`setError(message)`	`Result` オブジェクトに含めるカスタムエラーメッセージを設定します。 パラメータ: `message` - (string) エラーメッセージ。
`discardError()`	`Result` オブジェクトで返されるエラーを破棄します。
`getAlert()`	ブラウザアラートを含む`Alert` オブジェクトを返します（ブラウザアラートがない場合は `null` を返します）。
`chromeOptions()`	事前定義された Chrome ブラウザオプションを含む `chromeOptions` オブジェクトを返します。
`firefoxOptions()`	事前定義された Firefox ブラウザオプションを含む `firefoxOptions` オブジェクトを返します。
`safariOptions()`	事前定義された Safari ブラウザオプションを含む `safariOptions` オブジェクトを返します。
`edgeOptions()`	事前定義された Edge ブラウザオプションを含む `edgeOptions` オブジェクトを返します。
`switchFrame(target)`	指定したフレームに切り替えます。 パラメータ: `target` - (browser element or integer, optional) 対象フレーム。要素でフレームを選択するには、その要素を渡します。インデックスでフレームを選択するには、その番号を渡します。空のままにすると、トップレベルのブラウジングコンテキストに切り替わります。

すべての Browser メソッドは、以下のエラーをスローする可能性があります。

- `BrowserError` - `Error` オブジェクトから派生したもので、`Browser` コンストラクタが失敗した場合にスローされます。追加の `browser` プロパティを含み、この `BrowserError` をスローした `Browser` オブジェクトを保持します。
- `WebdriverError` - `BrowserError` から派生したもので、`BrowserError` オブジェクトと同じプロパティを含みます。これらは、WebDriver レスポンス内のエラーに応答して生成されたエラーであることを示します。

Element

`Element` オブジェクトは、`Browser` オブジェクトの `findElement()`/`findElements()` メソッドによって返され、直接構築することはできません。

`Element` オブジェクトは Web ページ内の要素を表し、それを操作するためのメソッドを提供します。

Element オブジェクトでは、以下のメソッドがサポートされています。

Method	Description
getAttribute(name)	要素の属性の属性値文字列を返します（指定した属性が見つからない場合は null を返します）。 パラメータ: name - (string) 属性名。
getProperty(name)	要素のプロパティのプロパティ値文字列を返します（指定したプロパティが見つからない場合は null を返します）。 パラメータ: name - (string) プロパティ名。
getText()	要素のテキストの文字列値を返します。
click()	要素をクリックします。
clear()	編集可能な要素の内容をクリアします。
sendKeys(keys)	キーを送信します。 パラメータ: keys - (string) 送信するキー。

Cookie

Cookie オブジェクトは、Browser オブジェクトの getCookies() メソッドによって返され、addCookie() メソッドに渡されます。

Cookie オブジェクトにはメソッドはありませんが、以下のプロパティを含めることができます。

プロパティ	Type	説明
name	string	Cookie の名前。
value	string	Cookie の値。
path	string	Cookie が有効なパス。 Cookie の追加時に省略した場合、デフォルトは "/" です。
domain	string	Cookie を参照できるドメイン。 Cookie の追加時に省略した場合、デフォルトはセッションの現在の閲覧コンテキストのアクティブなドキュメントの URL ドメインです。
secure	boolean	Cookie がセキュアであるかどうかを示します。 Cookie の追加時に省略した場合、デフォルトは false です。
httpOnly	boolean	Cookie が HTTP 専用であるかどうかを示します。 Cookie の追加時に省略した場合、デフォルトは false です。
expiry	integer	Cookie の有効期限（Unix エポックからの秒数）。Cookie の追加時に省略する場合は、設定してはいけません。
sameSite	string	Cookie の sameSite 属性で、Cookie をファーストパーティまたは same-site コンテキストに制限するかどうかを制御します。 "Lax" または "Strict" のいずれかを設定できます。 Cookie の追加時に省略した場合、デフォルトは "None" です。

Alert

Alert オブジェクトは Web ページのアラートを表し、Browser オブジェクトの getAlert() メソッドによって返され、直接構築することはできません。

Alert オブジェクトには、アラートのテキストを含む text プロパティがあります（アラートがない場合は null）。

Alert オブジェクトでは、以下のメソッドがサポートされています。

Method	Description
accept()	アラートを受け入れます。
dismiss()	アラートを閉じます。

結果

Result オブジェクトにはセッション統計が含まれ、Browser オブジェクトの getResult() メソッドによって返されます。

通常、Result オブジェクトは文字列化されてスクリプトから返され、その後、前処理によって依存アイテムの値に解析されます。

Result オブジェクトにはメソッドはありませんが、以下のプロパティを含めることができます。

プロパティ	型	説明
duration	string	セッション作成から結果取得までのセッション継続時間。
error	object	エラー情報。
http_status	integer	WebDriver によって返された HTTP ステータス（WebDriver エラーがない場合は 0）。
error_code	string	WebDriver によって返されたエラー（WebDriver エラーがない場合は空文字列）。
message	string	WebDriver のエラーメッセージ（WebDriver エラーがない場合は空文字列）。
performance_data	object	パフォーマンス統計。
summary	object	パフォーマンスの概要。
navigation	object	ナビゲーションの概要。
resource	object	リソースの概要。
details	array of objects	ナビゲーションが発生した可能性のある各操作後のパフォーマンス統計。
mark	string	(任意) `collectPerfEntries()` メソッドで指定されたパフォーマンススナップショットのマーク。
navigation	object	ナビゲーション統計。
resource	object	このステップのリソース概要。
user	array of objects	mark/measure 型統計の配列。
marks	array of objects	マークされたパフォーマンススナップショットのインデックス。
name	string	パフォーマンススナップショットのマーク名。
index	integer	details 配列内のパフォーマンススナップショットのインデックス。

6 CSV から JSON への保存前処理

概要

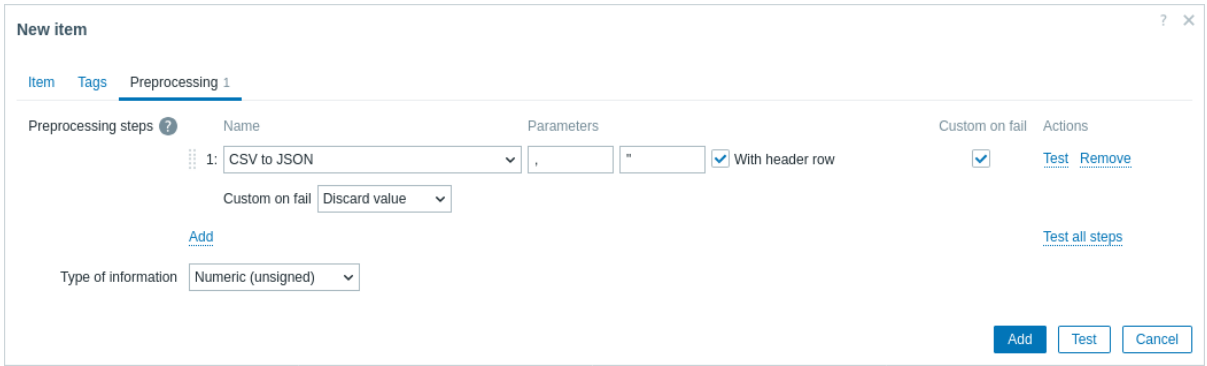
この保存前処理で、CSV ファイルデータを JSON 形式に変換することが可能です。サポートしているのは、

- アイテム (アイテムのプロトタイプ)
- ローレベルディスカバリルール

設定

CSV から JSON への保存前処理のステップを設定する場合:

- アイテム/ディスカバリルールの保存前処理タブにアクセスします。
- 追加をクリックします。
- *CSV から JSON* オプションを選択します。



1 番目のパラメーターカスタムデリミタを設定することができます。CSV 入力の最初の行が”Sep=” で始まり、UTF-8 文字が 1 つ続く場合、最初のパラメーターが設定されていない場合、その文字がデリミタとして使用されることに注意ください。1 番目のパラメーターが設定されておらず、かつ”Sep=” 行からデリミタが取得できない場合は、カンマがセパレーターとして使用されます。

オプションの 2 番目のパラメーターには引用記号を設定することができます。

ヘッダ行を含むのチェックボックスがチェックされている場合、ヘッダー行の値は列名として解釈されます (詳しくはヘッダー処理を参照してください)。

失敗時のカスタマイズのチェックボックスがチェックされている場合、保存前処理のステップが失敗してもそのアイテムがサポートされなくなることはありません。さらに、カスタムエラー処理オプションを設定することができます: 値を破棄、値を設定、エラーを設定。

ヘッダー処理

CSV ファイルのヘッダー行は、2 種類の方法で処理することができます。

- ヘッダ行を含むチェックボックスがチェックされている場合 - ヘッダー行の値はカラム名として解釈されます。この場合、カラム名は一意でなければならず、データ行はヘッダー行より多くのカラムを含んではなりません。
- ヘッダ行を含むチェックボックスがチェックされていない場合 - ヘッダー行はデータとして解釈されます。カラム名は自動的に生成されます (1,2,3,4...)。

CSV ファイルの例:

```
Nr,Item name,Key,Qty
1,active agent item,agent.hostname,33
"2","passive agent item","agent.version","44"
3,"active,passive agent items",agent.ping,55
```

Note:
入力の引用フィールド内の引用文字は、その前に別の引用文字を置くことでエスケープしなければいけません。

ヘッダー行の処理

ヘッダー行が検出される場合の JSON 出力:

```
[
   {
      "Nr":"1",
      "Item name":"active agent item",
      "Key":"agent.hostname",
      "Qty":"33"
   },
   {
      "Nr":"2",
      "Item name":"passive agent item",
      "Key":"agent.version",
      "Qty":"44"
   },
   {
      "Nr":"3",
      "Item name":"active,passive agent items",
      "Key":"agent.ping",
      "Qty":"55"
   }
]
```

ヘッダー行の処理なし

ヘッダー行が検出できない場合の JSON 出力:

```
[
   {
      "1":"Nr",
      "2":"Item name",
      "3":"Key"
      "4":"Qty"
   },
   {
      "1":"1",
      "2":"active agent item",
      "3":"agent.hostname"
      "4":"33"
   },
   {
      "1":"2",
```

```
        "2":"passive agent item",
        "3":"agent.version"
        "4":"44"
    },
    {
        "1":"3",
        "2":"active,passive agent items",
        "3":"agent.ping"
        "4":"55"
    }
]
```

3 アイテムのタイプ

概要

アイテムのタイプでは、システムからデータを取得するためのさまざまな方法をカバーします。各アイテムのタイプには、サポートされるアイテムキーと必須パラメーターのセットがあります。

現在、Zabbix が提供しているアイテムのタイプは以下の通りです。

- Zabbix エージェント
- SNMP エージェント
- SNMP トラップ
- IPMI
- シンプルチェック
  - VMware 監視
- ログ監視
- 計算アイテム
  - アグリゲート
- Zabbix インターナル
- SSH
- Telnet
- 外部チェック
- Zabbix トラッパー
- JMX
- データベースモニタ
- 依存アイテム
- HTTP
- Prometheus
- スクリプト
- ブラウザアイテム

すべてのアイテムのタイプの詳細は、このセクションのサブページに含まれています。アイテムのタイプはデータ収集のための多くのオプションを提供しますが、ユーザーパラメーターまたはローダブルモジュールを使用することもできます。

一部のチェックは Zabbix サーバーのみで実行されます (エージェントレス監視) が、Zabbix エージェント、または Zabbix Java ゲートウェイ (JMX 監視) が必要なものもあります。

> **Attention:**
> 特定のアイテムのタイプで特定のインターフェースが必要な場合 (IPMI チェックにはホスト上の IPMI インターフェースが必要なように)、そのインターフェースはホストの定義に存在する必要があります。

ホストの定義には複数のインターフェースを設定することができます: Zabbix エージェント、SNMP エージェント、JMX、IPMI)。アイテムが複数のインターフェースを使用できる場合、利用可能なホストインターフェースを検索 (エージェント → SNMP → JMX → IPMI の順) し、最初にリンクされる適切なインターフェースが表示されます。

テキストを返すすべてのアイテム (文字、ログ、情報のテキストタイプ) は空白のみを返し、(該当する場合) 戻り値を空の文字列に設定できます (2.0 以降サポート)。

1 Zabbix エージェント

概要

このセクションでは、データ収集のために Zabbix エージェントとの通信を使用するアイテムキーの詳細を説明します。

エージェントチェックには、パッシブチェックとアクティブチェックがあります。アイテムを設定する際に、必要なタイプを選択できます。

- *Zabbix* エージェント - パッシブチェック用
- *Zabbix* エージェント *(active)* - アクティブチェック用

Zabbix エージェントでサポートされているすべてのアイテムキーは、新世代の Zabbix エージェント 2 でもサポートされていることに注意してください。エージェント 2 でのみ使用できる追加のアイテムキーも参照してください。

サポートされているアイテムキー

Zabbix エージェントで使用できるアイテムキーを以下に示します。

アイテムキーは、パラメータおよび追加情報を省略して一覧表示しています。詳細は、アイテムキーをクリックして確認してください。

Item key	Description	Item group
kernel.maxfiles	OS でサポートされるオープン可能なファイルの最大数。	Kernel
kernel.maxproc	OS でサポートされるプロセスの最大数。	
kernel.openfiles	現在オープンされているファイルディスクリプタの数。	
log	ログファイルの監視。	Log monitoring
log.count	監視対象のログファイル内で一致した行数。	
logrt	ローテーションされるログファイルの監視。	
logrt.count	ローテーションされる監視対象ログファイル内で一致した行数。	
modbus.get	Modbus データを読み取ります。	Modbus
net.dns	DNS サービスの状態を確認します。	Network
net.dns.perf	DNS サービスのパフォーマンスを確認します。	
net.dns.record	DNS クエリを実行します。	
net.if.collisions	衝突ウィンドウ外で発生したコリジョン数。	
net.if.discovery	ネットワークインターフェースの一覧。	
net.if.in	ネットワークインターフェースの受信トラフィック統計。	
net.if.out	ネットワークインターフェースの送信トラフィック統計。	
net.if.total	ネットワークインターフェースの受信および送信トラフィック統計の合計。	
net.tcp.listen	この TCP ポートが LISTEN 状態かどうかを確認します。	
net.tcp.port	指定したポートへの TCP 接続が可能かどうかを確認します。	
net.tcp.service	サービスが稼働しており、TCP 接続を受け付けているかどうかを確認します。	
net.tcp.service.perf	TCP サービスのパフォーマンスを確認します。	
net.tcp.socket.count	パラメータに一致する TCP ソケット数を返します。	
net.udp.listen	この UDP ポートが LISTEN 状態かどうかを確認します。	
net.udp.service	サービスが稼働しており、UDP リクエストに応答しているかどうかを確認します。	
net.udp.service.perf	UDP サービスのパフォーマンスを確認します。	
net.udp.socket.count	パラメータに一致する UDP ソケット数を返します。	
proc.cpu.util	プロセスの CPU 使用率。	Processes
proc.get	OS プロセスとそのパラメータの一覧。	
proc.mem	プロセスが使用しているメモリ量（バイト単位）。	
proc.num	プロセス数。	
sensor	ハードウェアセンサーの読み取り値。	Sensors
system.boottime	システムの起動時刻。	System
system.cpu.discovery	検出された CPU/CPU コアの一覧。	
system.cpu.intr	デバイス割り込み。	
system.cpu.load	CPU 負荷。	
system.cpu.num	CPU 数。	
system.cpu.switches	コンテキストスイッチ数。	
system.cpu.util	CPU 使用率。	
system.hostname	システムのホスト名。	
system.hw.chassis	シャーシ情報。	
system.hw.cpu	CPU 情報。	
system.hw.devices	PCI または USB デバイスの一覧。	
system.hw.macaddr	MAC アドレスの一覧。	
system.localtime	システム時刻。	
system.run	ホスト上で指定したコマンドを実行します。	
system.stat	システム統計。	
system.sw.arch	ソフトウェアアーキテクチャ情報。	

Item key	Description	Item group
system.sw.os	オペレーティングシステム情報。	
system.sw.os.get	オペレーティングシステムの詳細情報（バージョン、タイプ、ディストリビューション名、マイナーバージョン、メジャーバージョンなど）。	
system.sw.packages	インストール済みパッケージの一覧。	
system.sw.packages.get	インストール済みパッケージの詳細一覧。	
system.swap.in	スワップイン（デバイスからメモリへ）の統計。	
system.swap.out	スワップアウト（メモリからデバイスへ）の統計。	
system.swap.size	スワップ領域のサイズ（バイト単位、または全体に対する割合）。	
system.uname	システムの識別情報。	
system.uptime	システムの稼働時間（秒単位）。	
system.users.num	ログイン中のユーザー数。	
vfs.dev.discovery	ブロックデバイスとそのタイプの一覧。	Virtual file systems
vfs.dev.read	ディスク読み取り統計。	
vfs.dev.write	ディスク書き込み統計。	
vfs.dir.count	ディレクトリエントリ数。	
vfs.dir.get	ディレクトリエントリの一覧。	
vfs.dir.size	ディレクトリサイズ。	
vfs.file.cksum	UNIX cksum アルゴリズムで計算されたファイルチェックサム。	
vfs.file.contents	ファイルの内容を取得します。	
vfs.file.exists	ファイルが存在するかどうかを確認します。	
vfs.file.get	ファイルに関する情報を返します。	
vfs.file.md5sum	ファイルの MD5 チェックサム。	
vfs.file.owner	ファイルの所有者を取得します。	
vfs.file.permissions	UNIX パーミッションの 8 進数を含む 4 桁の文字列を返します。	
vfs.file.regexp	ファイル内の文字列を取得します。	
vfs.file.regmatch	ファイル内の文字列を検索します。	
vfs.file.size	ファイルサイズ。	
vfs.file.time	ファイル時刻情報。	
vfs.fs.discovery	マウントされたファイルシステムとそのタイプおよびマウントオプションの一覧。	
vfs.fs.get	マウントされたファイルシステムとそのタイプ、利用可能なディスク容量、inode 統計、およびマウントオプションの一覧。	
vfs.fs.inode	inode 数、またはその割合。	
vfs.fs.size	ディスク容量（バイト単位、または全体に対する割合）。	
vm.memory.size	メモリサイズ（バイト単位、または全体に対する割合）。	Virtual memory
web.page.get	Web ページの内容を取得します。	Web monitoring
web.page.perf	Web ページ全体の読み込み時間。	
web.page.regexp	Web ページ上の文字列を検索します。	
agent.hostmetadata	エージェントのホストメタデータ。	Zabbix
agent.hostname	エージェントのホスト名。	
agent.ping	エージェントの可用性チェック。	
agent.variant	Zabbix エージェントのバリアント（Zabbix エージェントまたは Zabbix エージェント 2）。	
agent.version	Zabbix エージェントのバージョン。	
zabbix.stats	Zabbix サーバーまたはプロキシの内部メトリクス一式を返します。ip と port が指定されている場合はリモートインスタンスから、そうでない場合はローカルインスタンスからメトリクスを取得します。	
zabbix.stats	Zabbix サーバーまたはプロキシで遅延しているキュー内の監視対象アイテム数を返します。ip と port が指定されている場合はリモートインスタンスから、そうでない場合はローカルインスタンスからメトリクスを取得します。	

サポートされているプラットフォーム

アイテムの詳細で別途指定がない限り、エージェントのアイテム（およびすべてのパラメータ）は以下でサポートされています。

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**

- **MacOS X**
- **OpenBSD**
- **NetBSD**

多くのエージェントアイテムは **Windows** でもサポートされています。
詳細については、Windows エージェントアイテム のページを参照してください。

アイテムキーの詳細

山かっこなしのパラメータは必須です。
山かっこ **< >** で示されたパラメータは任意です。

kernel.maxfiles

 OS がサポートする、開くことができるファイルの最大数。
 戻り値: 整数。
 サポートされているプラットフォーム: Linux、FreeBSD、MacOS X、OpenBSD、NetBSD。

kernel.maxproc

 OS がサポートするプロセスの最大数。
 戻り値: 整数。
 サポートされているプラットフォーム: Linux 2.6 以降、FreeBSD、Solaris、MacOS X、OpenBSD、NetBSD。

kernel.openfiles

 現在オープンされているファイルディスクリプターの数。
 戻り値: 整数。
 サポートされているプラットフォーム: Linux（このアイテムは他の UNIX 系プラットフォームでも動作する場合があります）。

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 ログファイルの監視。
 戻り値: *Log*。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **file** - ログファイルのフルパスとファイル名。
- **regexp** - 必要なパターンを記述する正規表現。
- **encoding** - コードページの識別子。
- **maxlines** - エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行の最大数。このパラメータは、zabbix_agentd.conf の'MaxLinesPerSecond' の値を上書きします。
- **mode** - 指定可能な値: *all* (デフォルト) または *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。
- **output** - オプションの出力整形テンプレート。**\0** エスケープシーケンスは一致したテキスト部分（一致開始位置の最初の文字から一致終了位置の文字まで）に置き換えられ、**\N**（N=1...9）エスケープシーケンスは N 番目の一致グループに置き換えられます（N がキャプチャされたグループ数を超える場合は空文字列になります）。
- **maxdelay** - 最大遅延（秒）。型: float。値: 0 -（デフォルト）ログファイルの行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行を解析できるようにするため、古い行を無視します。使用する前にmaxdelay に関する注意事項を読んでください。
- **options** - 追加オプション:
mtime-noreread - 一意でないレコードについて、ファイルサイズが変化した場合にのみ再読み込みします（更新時刻の変更は無視します）。（このパラメータは 5.0.2 以降非推奨です。現在は mtime が無視されるためです。）
- **persistent dir**（Unix システム上の zabbix_agentd のみ。Zabbix agent 2 ではサポートされません）- 永続ファイルを保存するディレクトリの絶対パス名。persistent files に関する追加の注意事項も参照してください。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります。
- ファイルが存在しない場合、または権限によりアクセスできない場合、このアイテムは未サポートになります。
- output を空のままにすると、一致したテキストを含む行全体が返されます。なお、グローバル正規表現タイプのうち'Result is TRUE' を除くすべては、常に一致した行全体を返し、output パラメータは無視されます。
- output パラメータを使用した内容の抽出は、エージェント上で行われます。

例:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

ログレコードから数値を抽出するために output パラメータを使用する例:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] # このアイテムはロ
```

サーバーへ送信する前にログレコードを書き換えるために output パラメータを使用する例:

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

log.count[file,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 監視対象のログファイル内で一致した行数です。
 戻り値: *Integer*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ログファイルの完全なパスとファイル名。
- **regexp** - 必要なパターンを記述する正規表現。
- **encoding** - コードページの識別子。
- **maxproclines** - エージェントが 1 秒あたりに解析する新しい行の最大数です（10000 を超えることはできません）。デフォルト値はzabbix_agentd.conf の 10*'MaxLinesPerSecond' です。
- **mode** - 使用可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。
- **maxdelay** - 最大遅延（秒）。型: float。値: 0 -（デフォルト）ログファイルの行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行を解析できるようにするため、古い行を無視します。使用する前にmaxdelay に関する注意事項を読んでください。
- **options** - 追加オプション:
mtime-noreread - 一意でないレコードについて、ファイルサイズが変化した場合にのみ再読み込みします（更新時刻の変更は無視します）。（このパラメータは 5.0.2 以降非推奨です。現在は mtime が無視されるためです。）
- **persistent dir**（Unix システム上の zabbix_agentd のみ。Zabbix agent 2 ではサポートされません）- 永続ファイルを保存するディレクトリの絶対パス名。persistent files に関する追加の注意事項も参照してください。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります。
- 一致した行は、エージェントによる前回のログチェック以降の新しい行の中でカウントされるため、アイテムの更新間隔に依存します。
- ファイルが存在しない場合、または権限によりアクセスできない場合、このアイテムは未サポートになります。

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 ローテーションされるログファイルの監視。
 戻り値: *Log*。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **file regexp** - ファイルへの絶対パス。ファイル名は通常の正規表現で指定します。正規表現はファイル名にのみ適用され、名前全体に一致する必要はないことに注意してください（例: /path/to/agent は zabbix_agentd.log に一致します）。
- **regexp** - 必要な内容パターンを記述する通常の正規表現。
- **encoding** - コードページの識別子。
- **maxlines** - エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行の最大数。このパラメータは、zabbix_agentd.conf の'MaxLinesPerSecond' の値を上書きします。
- **mode** - 指定可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。
- **output** - オプションの出力整形テンプレート。**\0** エスケープシーケンスは一致したテキスト部分（一致開始位置の最初の文字から一致終了位置の文字まで）に置き換えられ、**\N**（N=1...9）エスケープシーケンスは N 番目の一致グループに置き換えられます（N がキャプチャされたグループ数を超える場合は空文字列になります）。
- **maxdelay** - 最大遅延（秒）。型: float。値: 0 -（デフォルト）ログファイルの行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行を解析できるよう、古い行を無視します。使用する前にmaxdelay の注意事項を読んでください。
- **options** - ログファイルローテーションの種類およびその他のオプション。指定可能な値:
rotate（デフォルト）、
copytruncate - *copytruncate* は *maxdelay* と同時に使用できないことに注意してください。この場合、*maxdelay* は 0 にするか指定しない必要があります。copytruncate の注意事項を参照してください。
mtime-reread - 一意でないレコード。更新時刻またはサイズが変化した場合に再読み込みします（`mtime-*` オプションが明示的に設定されていない場合のデフォルト）。
mtime-noreread - 一意でないレコード。サイズが変化した場合のみ再読み込みします（更新時刻の変化は無視します）。
- **persistent dir**（Unix システム上の zabbix_agentd のみ。Zabbix agent 2 では未サポート）- 永続ファイルを保存するディレクトリの絶対パス名。persistent files に関する追加の注意事項も参照してください。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります。
- ログローテーションはファイルの最終更新時刻に基づきます。
- logrt は、現在アクティブな 1 つのログファイルと、ローテーションされた複数の一致する非アクティブファイルで動作するよう設計されていることに注意してください。たとえば、あるディレクトリに多数のアクティブなログファイルがある場合は、それぞれに対して個別の logrt アイテムを作成する必要があります。そうしないと、1 つの logrt アイテムがあまりに多くのファイルを処理してしまい、メモリ枯渇や監視のクラッシュにつながる可能性があります。
- `output` を空のままにすると、一致したテキストを含む行全体が返されます。なお、グローバル正規表現タイプのうち'Result is TRUE' 以外は常に一致した行全体を返し、`output` パラメータは無視されます。
- `output` パラメータを使用した内容抽出はエージェント上で行われます。

- file regexp パラメータでは、ログディレクトリのパスとログファイルの正規表現を正しいディレクトリ区切り文字で区切る必要があります。
  - Windows では、区切り文字はバックスラッシュ (\) でなければなりません。ログディレクトリのパスとログファイルの正規表現を区切る位置を除き、他の位置ではスラッシュ (/) も許容される場合があります（以下の例を参照）。
  - その他のシステムでは、スラッシュ (/) でなければなりません。

Windows の例:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、"c:/dir1/dir2/dir3" 内にある、
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、ネットワーク共
```

その他のシステムの例:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] # このアイテムは "logfile1" のようなファイルに一致します（".
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] # このアイテムは "logfile_abc_1" 
```

output パラメータを使用してログレコードから数値を抽出する例:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] # このアイ
```

output パラメータを使用して、サーバーへ送信する前にログレコードを書き換える例:

```
logrt[/app1/^test.*log$,"([0-9 :-]+) task run ([0-9.]+) sec, processed ([0-9]+) records, ([0-9]+) errors",
```

logrt.count[file regexp,<regexp>,<encoding>,<maxproclines>,<mode>,<maxdelay>,<options>,<persistent dir>]

 ローテーションされる監視対象のログファイル内で一致した行数です。
 戻り値: *Integer*。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **file regexp** - ファイルへの絶対パス。ファイル名は正規表現で指定します。この正規表現はファイル名にのみ適用され、名前全体に一致する必要はありません（例: /path/to/agent は zabbix_agentd.log に一致します）。
- **regexp** - 必要なパターンを記述する正規表現。
- **encoding** - コードページの識別子。
- **maxproclines** - エージェントが 1 秒あたりに解析する新規行の最大数（10000 を超えることはできません）。デフォルト値はzabbix_agentd.conf の 10*'MaxLinesPerSecond' です。
- **mode** - 指定可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。
- **maxdelay** - 最大遅延（秒）。型: float。値: 0 -（デフォルト）ログファイルの行を無視しない; > 0.0 - "maxdelay" 秒以内に最新の行を解析できるよう、古い行を無視します。使用する前にmaxdelay に関する注意事項を読んでください。
- **options** - ログファイルローテーションの種類およびその他のオプション。指定可能な値:
rotate（デフォルト）、
copytruncate - *copytruncate* は *maxdelay* と同時に使用できないことに注意してください。この場合、*maxdelay* は 0 にするか、省略する必要があります。copytruncate に関する注意事項を参照してください。
mtime-reread - 一意でないレコード。更新時刻またはサイズが変更された場合に再読み込みします（デフォルト）、
mtime-noreread - 一意でないレコード。サイズが変更された場合にのみ再読み込みします（更新時刻の変更は無視します）。
- **persistent dir**（Unix システム上の zabbix_agentd のみ。Zabbix agent 2 では未サポート）- 永続ファイルを保存するディレクトリの絶対パス名。persistent files に関する追加の注意事項も参照してください。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります。
- 一致した行は、エージェントによる前回のログチェック以降の新しい行の中でカウントされるため、アイテムの更新間隔に依存します。
- ログローテーションは、ファイルの最終更新時刻に基づきます。
- file regexp パラメータでは、ログディレクトリのパスとログファイルの正規表現を、正しいディレクトリ区切り文字で区切る必要があります。
  - Windows では、区切り文字はバックスラッシュ (\) でなければなりません。その他の位置ではスラッシュ (/) も許容される場合がありますが、ログディレクトリのパスとログファイルの正規表現を区切る位置では使用できません（以下の例を参照）。
  - その他のシステムでは、スラッシュ (/) でなければなりません。

Windows の例:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、"c:/dir1/dir2/dir3" にあ
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] # このアイテムは、ネット
```

その他のシステムの例:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] # このアイテムは、"/home/zabbix/logs" ディレクト
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] # このアイテムは、"/home/use
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Modbus データを読み取ります。
 戻り値: *JSON* 文字列。
 対応プラットフォーム: Linux、Windows。

パラメータ:

- **endpoint** - `protocol://connection_string` として定義されるエンドポイント。
- **slave id** - スレーブ ID。
- **function** - Modbus 機能。
- **address** - 最初のレジスタ、コイル、または入力のアドレス。
- **count** - 読み取るレコード数。
- **type** - データ型。
- **endianness** - エンディアン設定。
- **offset** - 'address' から開始して、結果が破棄されるレジスタ数。

Windows では、このアイテムは **endpoint** パラメータ内の UNC パスに従います。これはセキュリティリスクとなる可能性があります。

パラメータの詳細な説明を参照してください。

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS サービスのステータスをチェックします。
 戻り値: 0 - DNS 名前解決に失敗した（DNS サーバーが応答しなかった、またはエラーを返した）; 1 - DNS 名前解決に成功した。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **ip**（Windows では Zabbix エージェント 2 を使用している場合を除き無視されます）- DNS サーバーの IP アドレス（デフォルトの DNS サーバーを使用する場合は空のままにします）;
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ（デフォルトは *SOA*）;
- **timeout**（Windows では Zabbix エージェント 2 を使用している場合を除き無視されます）- リクエストのタイムアウト（秒単位、デフォルトは 1 秒）;
- **count**（Windows では Zabbix エージェント 2 を使用している場合を除き無視されます）- リクエストの試行回数（デフォルトは 2）;
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp*（デフォルト）または *tcp*。

コメント:

- `type` に指定可能な値は次のとおりです: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS*（Windows 上の Zabbix エージェント、およびすべての OS 上の Zabbix エージェント 2 ではサポートされません）, *HINFO*, *MINFO*, *TXT*, *SRV*。
- 逆引き DNS ルックアップ（`type` を *PTR* に設定した場合）では、DNS 名を逆順形式と非逆順形式の両方で指定できます（以下の例を参照）。PTR レコードが要求された場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていないため、代わりに IDNA エンコードされた名前を使用してください。

例:

```
net.dns[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS サービスのパフォーマンスをチェックします。
 戻り値: *Float* (0 - サービス停止; 秒 - サービスからの応答を待機した秒数)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **ip** (Windows では Zabbix エージェント 2 を使用しない限り無視されます) - DNS サーバーの IP アドレス (デフォルトの DNS サーバーを使用する場合は空のままにします);
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ (デフォルトは *SOA*);
- **timeout** (Windows では Zabbix エージェント 2 を使用しない限り無視されます) - リクエストのタイムアウト (秒単位、デフォルトは 1 秒);

- **count** (Windows では Zabbix エージェント 2 を使用しない限り無視されます) - リクエストの試行回数 (デフォルトは 2);
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp* (デフォルト) または *tcp*。

コメント:

- type に指定可能な値は次のとおりです:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (Windows 上の Zabbix エージェント、すべての OS 上の Zabbix エージェント 2 ではサポートされません), *HINFO*, *MINFO*, *TXT*, *SRV*。
- 逆引き DNS ルックアップ (type を *PTR* に設定した場合) では、DNS 名を逆順形式と非逆順形式の両方で指定できます (以下の例を参照)。PTR レコードが要求された場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。
- DNS サーバーがエラーコード (`NXDOMAIN` や `SERVFAIL` など) を返した場合、このアイテムは 0 ではなく応答時間を返します。

例:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 DNS クエリを実行します。

Zabbix エージェント 2 では、追加のレコードタイプや DNS 監視のより詳細な制御などの追加機能を提供するnet.dns.get も利用できます。戻り値: 必要な種類の情報を含む文字列。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **ip** (Windows では Zabbix エージェント 2 を使用しない限り無視されます) - DNS サーバーの IP アドレス (デフォルトの DNS サーバーを使用する場合は空のままにします);
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ (デフォルトは *SOA*);
- **timeout** (Windows では Zabbix エージェント 2 を使用しない限り無視されます) - リクエストのタイムアウト時間 (秒単位、デフォルトは 1 秒);
- **count** (Windows では Zabbix エージェント 2 を使用しない限り無視されます) - リクエストの試行回数 (デフォルトは 2);
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp* (デフォルト) または *tcp*。

コメント:

- type に指定可能な値は次のとおりです:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (Windows 上の Zabbix エージェント、すべての OS 上の Zabbix エージェント 2 ではサポートされません), *HINFO*, *MINFO*, *TXT*, *SRV*。
- 逆引き DNS ルックアップ (type を *PTR* に設定した場合) では、逆順形式または非逆順形式の DNS 名を指定できます (以下の例を参照)。PTR レコードが要求された場合、DNS 名は実際には IP アドレスであることに注意してください。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。

例:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 ウィンドウ外コリジョンの数。
 戻り値: *Integer*。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、AIX、MacOS X、OpenBSD、NetBSD。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名

net.if.discovery

 ネットワークインターフェースの一覧です。ローレベルディスカバリに使用されます。
 戻り値: *JSON string*。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD、Windows。

net.if.in[if,<mode>]

 ネットワークインターフェース上の受信トラフィック統計。
 戻り値: 整数。
 対応プラットフォーム: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名 (Unix)、ネットワークインターフェースの完全な説明または IPv4 アドレス、または中括弧で囲まれている場合はネットワークインターフェース GUID (Windows)。
- **mode** - 指定可能な値:
bytes - バイト数 (デフォルト)
packets - パケット数
errors - エラー数
dropped - ドロップされたパケット数
overruns (fifo) - FIFO バッファエラー数
frame - パケットフレーミングエラー数
compressed - デバイスドライバーが受信した圧縮パケット数
multicast - デバイスドライバーが受信したマルチキャストフレーム数

コメント:

- バイト/秒の統計を取得するには、このキーを *1* 秒あたりの変化量前処理ステップとともに使用できます。
- *dropped* モードは Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD でのみサポートされています。
- *overruns*, *frame*, *compressed*, *multicast* モードは Linux でのみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 ネットワークインターフェース上の送信トラフィック統計です。
 戻り値: 整数。
 対応プラットフォーム: Linux、FreeBSD、Solaris[5]、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名 (Unix)、ネットワークインターフェースの完全な説明または IPv4 アドレス、あるいは中括弧で囲まれたネットワークインターフェース GUID (Windows)。
- **mode** - 指定可能な値:
bytes - バイト数 (デフォルト)
packets - パケット数
errors - エラー数
dropped - ドロップされたパケット数
overruns (fifo) - FIFO バッファエラー数
collisions (colls) - インターフェース上で検出された衝突数
carrier - デバイスドライバーによって検出されたキャリア損失数
compressed - デバイスドライバーによって送信された圧縮パケット数

コメント:

- バイト/秒の統計を取得するために、このキーを *1* 秒あたりの変化量前処理ステップとともに使用できます。
- *dropped* モードは Linux、HP-UX でのみサポートされています。
- *overruns*、*collision*、*carrier*、*compressed* モードは Linux でのみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 ネットワークインターフェースにおける受信および送信トラフィック統計の合計です。
 戻り値: *Integer*。
 対応プラットフォーム: Linux、FreeBSD、Solaris[5]、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。NetBSD では root 権限が必要です。

パラメータ:

- **if** - ネットワークインターフェース名 (Unix)、ネットワークインターフェースの完全な説明または IPv4 アドレス、あるいは波かっこで囲んだネットワークインターフェース GUID (Windows)
- **mode** - 指定可能な値:
bytes - バイト数 (デフォルト)
packets - パケット数
errors - エラー数
dropped - ドロップされたパケット数
overruns (fifo) - FIFO バッファエラー数
collisions (colls) - インターフェース上で検出された衝突数
compressed - デバイスドライバーによって送受信された圧縮パケット数

コメント:

- バイト/秒の統計を取得するには、このキーを *1* 秒あたりの変化量の前処理ステップとともに使用できます。
- *dropped* モードは Linux、HP-UX でのみサポートされています。ドロップされたパケットは、お使いのプラットフォームで `net.if.in` と `net.if.out` の両方が dropped パケットに対して機能する場合にのみサポートされます。
- *overruns*、*collision*、*compressed* モードは Linux でのみサポートされています。
- HP-UX では、このアイテムはループバックインターフェース (例: lo0) の詳細を提供しません。

例:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 この TCP ポートが LISTEN 状態にあるかどうかを確認します。
 戻り値: 0 - LISTEN 状態ではない; 1 - LISTEN 状態である。
 対応プラットフォーム: Linux、FreeBSD、Solaris、MacOS X、Windows。

パラメータ:

- **port** - TCP ポート番号

Linux カーネル 2.6.14 以降では、可能な場合、待ち受け中の TCP ソケットに関する情報はカーネルの NETLINK インターフェースから取得されます。それ以外の場合、情報は /proc/net/tcp および /roc/net/tcp6 ファイルから取得されます。

例:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 指定したポートへの TCP 接続が可能かどうかを確認します。
 戻り値: 0 - 接続不可; 1 - 接続可能。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **ip** - IP アドレスまたは DNS 名（デフォルトは 127.0.0.1）;
- **port** - ポート番号。

コメント:

- 単純な TCP パフォーマンステストには `net.tcp.service.perf[tcp,<ip>,<port>]` を使用してください。
- これらのチェックにより、システムデーモンのログファイルに追加のメッセージが記録される場合があります（通常は SMTP および SSH セッションが記録されます）。

例:

```
net.tcp.port[,80] # このアイテムは、ポート 80 で稼働している Web サーバーの可用性をテストするために使用できます
```

net.tcp.service[service,<ip>,<port>]

 サービスが実行中で、TCP 接続を受け付けているかを確認します。
 戻り値: 0 - サービス停止; 1 - サービス稼働中。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ssh*、*ldap*、*smtp*、*ftp*、*http*、*pop*、*nntp*、*imap*、*tcp*、*https*、または *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

コメント:

- これらのチェックにより、システムデーモンのログファイルに追加のメッセージが記録される場合があります (通常は SMTP および SSH セッションが記録されます)。
- 暗号化プロトコル (993 番ポートの IMAP や 995 番ポートの POP など) のチェックは、現在サポートされていません。回避策として、このようなチェックには `net.tcp.port[]` を使用してください。
- Windows での LDAP および HTTPS のチェックは、Zabbix エージェント 2 でのみサポートされています。
- telnet チェックでは、ログインプロンプト (末尾の':') を探します。

例:

```
net.tcp.service[ftp,,45] # このアイテムは TCP ポート 45 上の FTP サーバーの可用性をテストするために使用できます
```

net.tcp.service.perf[service,<ip>,<port>]

 TCP サービスのパフォーマンスをチェックします。
 戻り値: *Float* (0 - サービス停止; 秒 - サービスからの応答を待機した秒数)。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **service** - *ssh*、*ldap*、*smtp*、*ftp*、*http*、*pop*、*nntp*、*imap*、*tcp*、*https*、または *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

コメント:

- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには net.tcp.service.perf[tcp,<ip>,<port>] を使用してください。
- telnet チェックでは、ログインプロンプト (末尾の':') を探します。

例:

```
net.tcp.service.perf[ssh] # このアイテムは SSH サーバーからの初期応答速度のテストに使用できます
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 パラメータに一致する TCP ソケットの数を返します。
 戻り値: 整数。
 対応プラットフォーム: Linux、Windows。

パラメータ:

- **laddr** - ローカルの IPv4/6 アドレスまたは CIDR サブネット。
- **lport** - ローカルのポート番号またはサービス名。
- **raddr** - リモートの IPv4/6 アドレスまたは CIDR サブネット。
- **rport** - リモートのポート番号またはサービス名。
- **state** - 接続状態 (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*)。

例:

```
net.tcp.socket.count[,80,,,established] # ローカル TCP ポート 80 への接続のうち、established 状態にある接続数
```

net.udp.listen[port]

 この UDP ポートが LISTEN 状態にあるかどうかを確認します。
 戻り値: 0 - LISTEN 状態ではない; 1 - LISTEN 状態である。
 サポートされるプラットフォーム: Linux、FreeBSD、Solaris、MacOS X。

パラメータ:

- **port** - UDP ポート番号

例:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 サービスが実行中で、UDP リクエストに応答しているかを確認します。
 戻り値: 0 - サービス停止; 1 - サービス稼働中。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトは 127.0.0.1);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

例:

```
net.udp.service[ntp,,45] # このアイテムは、UDP ポート 45 での NTP サービスの可用性をテストするために使用できます
```

net.udp.service.perf[service,<ip>,<port>]

 UDP サービスのパフォーマンスをチェックします。
 戻り値: *Float*（0 - サービス停止中、秒 - サービスからの応答を待機した秒数)。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **service** - *ntp*（詳細を参照）;
- **ip** - IP アドレスまたは DNS 名（デフォルトは 127.0.0.1）;
- **port** - ポート番号（デフォルトでは標準のサービスポート番号が使用されます)。

例:

```
net.udp.service.perf[ntp] # このアイテムは NTP サービスからの応答時間のテストに使用できます
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 パラメータに一致する UDP ソケットの数を返します。
 戻り値: 整数。
 対応プラットフォーム: Linux、Windows。

パラメータ:

- **laddr** - ローカル IPv4/6 アドレスまたは CIDR サブネット。
- **lport** - ローカルポート番号またはサービス名。

- **raddr** - リモート IPv4/6 アドレスまたは CIDR サブネット。
- **rport** - リモートポート番号またはサービス名。
- **state** - 接続状態 (*established*、*unconn*)。

例:

```
net.udp.socket.count[,,,,established] # 接続状態の UDP ソケット数を返します
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 プロセスの CPU 使用率（%）です。
 戻り値: *Float*。
 対応プラットフォーム: Linux、Solaris[6]。

パラメータ:

- **name** - プロセス名（デフォルトは すべてのプロセス）;
- **user** - ユーザー名（デフォルトは すべてのユーザー）;
- **type** - CPU 使用率の種類: *total*（デフォルト）、*user*、または *system*;
- **cmdline** - コマンドラインでフィルタリングします（正規表現です）;
- **mode** - データ収集モード: *avg1*（デフォルト）、*avg5*、または *avg15*;
- **zone** - 対象ゾーン: *current*（デフォルト）または *all*。このパラメータは Solaris でのみサポートされています。

コメント:

- 返される値は、単一の CPU コアの使用率（%）に基づきます。たとえば、2 つのコアを完全に使用しているプロセスの CPU 使用率は 200% になります。
- プロセスの CPU 使用率データは、collector によって収集されます。この collector は、(name、user、command line の組み合わせで）最大 1024 個の一意なクエリをサポートします。過去 24 時間アクセスされていないクエリは collector から削除されます。
- zone パラメータを *current*（またはデフォルト値）に設定した場合、zone サポートのない Solaris 上でエージェントがコンパイルされ、その後 zone をサポートする新しい Solaris 上で実行されていると、エージェントは NOTSUPPORTED を返します（エージェントは結果を現在の zone のみに制限できません）。ただし、この場合は *all* がサポートされます。

例:

```
proc.cpu.util[,root] #CPU utilization of all processes running under the "root" user
proc.cpu.util[zabbix_server,zabbix] #CPU utilization of all zabbix_server processes running under the zabb
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 OS プロセスとそのパラメータの一覧です。ローレベルディスカバリに使用できます。
 戻り値: *JSON string*。
 Supported platforms: Linux, FreeBSD, Windows, OpenBSD, NetBSD。

パラメータ:

- **name** - プロセス名（デフォルトは すべてのプロセス）;
- **user** - ユーザー名（デフォルトは すべてのユーザー）;
- **cmdline** - コマンドラインでフィルタリングします（正規expressionです）。このパラメータは Windows ではサポートされません。その他のプラットフォームでは、mode が'summary' に設定されている場合はサポートされません。
- **mode** - 指定可能な値:
process（デフォルト）, *thread*（NetBSD ではサポートされません）, *summary*。各 mode および OS で返されるprocess parametersの一覧を参照してください。

コメント:

- 値を取得できない場合、たとえばエラー（プロセスがすでに終了している、権限不足、システムコールの失敗）により、-1 が返されます。
- `name` および `cmdline` パラメータでプロセスを選択する際のnotesを参照してください（Linux 固有）。

例:

```
proc.get[zabbix_server,zabbix,,process] #zabbix ユーザーで実行されているすべての zabbix_server プロセスの一覧、PID
proc.get[java,,,thread] # すべての Java プロセスの一覧、スレッドごとに 1 エントリを返す
proc.get[,zabbix,,summary] #zabbix ユーザーで実行されている各種類のプロセスの集計データ、プロセス名ごとに 1 エントリを返
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 プロセスが使用しているメモリ量（バイト単位）。
 戻り値: `mode` が *max*、*min*、*sum* の場合は *Integer*、`mode` が *avg* の場合は *Float*
 対応プラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD、NetBSD。

パラメータ:

- **name** - プロセス名（デフォルトは すべてのプロセス）;
- **user** - ユーザー名（デフォルトは すべてのユーザー）;
- **mode** - 指定可能な値: *avg*、*max*、*min*、または *sum*（デフォルト）;

- **cmdline** - コマンドラインでフィルタリングします（正規表現です）；
- **memtype** - プロセスが使用するメモリの種類

コメント:

- memtype パラメータは、Linux、FreeBSD、Solaris[6]、AIX でのみサポートされています。
- 複数のプロセスが共有メモリを使用している場合、プロセスが使用するメモリの合計は、非現実的な大きな値になることがあります。

name および cmdline パラメータによるプロセス選択については、注意事項（Linux 固有）を参照してください。

 このアイテムがコマンドラインから呼び出され、コマンドラインパラメータを含む場合（例: エージェントのテストモード zabbix_agentd -t proc.mem[,,,apache2] を使用）、エージェント自身もカウントするため、1 つ余分なプロセスがカウントされます。

例:

```
proc.mem[,root] #the memory used by all processes running under the "root" user
proc.mem[zabbix_server,zabbix] #the memory used by all zabbix_server processes running under the zabbix us
proc.mem[,oracle,max,oracleZABBIX] #the memory used by the most memory-hungry process running under Oracle
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 プロセス数。
 戻り値: *Integer*。
 対応プラットフォーム: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows。

パラメータ:

- **name** - プロセス名（デフォルトは すべてのプロセス）；
- **user** - ユーザー名（デフォルトは すべてのユーザー）；
- **state** - 指定可能な値:
all（デフォルト）,
disk - 割り込み不可のスリープ,
run - 実行中,
sleep - 割り込み可能なスリープ,
trace - 停止中,
zomb - ゾンビ;
- **cmdline** - コマンドラインでフィルタリングします（正規表現です）；
- **zone** - 対象ゾーン: *current*（デフォルト）または *all*。このパラメータは Solaris でのみサポートされています。

コメント:

- *disk* および *trace* の状態パラメータは、Linux、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- Windows では、name および user パラメータのみサポートされています。
- このアイテムがコマンドラインから呼び出され、コマンドラインパラメータを含む場合（例: エージェントのテストモードを使用する場合: zabbix_agentd -t proc.num[,,,apache2]）、エージェント自身もカウントするため、プロセスが 1 つ余分にカウントされます。
- zone パラメータを *current*（またはデフォルト）に設定した場合、ゾーンサポートなしでコンパイルされたエージェントを、ゾーンがサポートされている新しい Solaris 上で実行すると、エージェントは NOTSUPPORTED を返します（エージェントは結果を現在のゾーンのみに制限できません）。ただし、この場合は *all* がサポートされます。
- name および cmdline パラメータを使用したプロセスの選択については、注意事項（Linux 固有）を参照してください。

例:

```
proc.num[,mysql] #the number of processes running under the mysql user
proc.num[apache2,www-data] #the number of apache2 processes running under the www-data user
proc.num[,oracle,sleep,oracleZABBIX] #the number of processes in sleep state running under Oracle having c
```

sensor[device,sensor,<mode>]

 ハードウェアセンサーの読み取り。
 戻り値: *Float*。
 対応プラットフォーム: Linux、OpenBSD。

パラメータ:

- **device** - デバイス名。
- **sensor** - センサー名。
- **mode** - 指定可能な値: *avg*、*max*、または *min*。

コメント:

- Linux 2.4 では /proc/sys/dev/sensors を読み取ります。
- Linux 2.6+ では /sys/class/hwmon を読み取ります。Linux におけるsensorアイテムのより詳細な説明も参照してください。
- OpenBSD では *hw.sensors* MIB を読み取ります。
- Linux 2.4 では、mode を指定した場合、device と sensor は正規表現として扱われます。
- Linux 2.6+ および OpenBSD、または mode を省略した場合、device と sensor はそのままの文字列として扱われます。

例:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #the temperature of one CPU
sensor["cpu[0-2]$",temp,avg] #the average temperature of the first three CPUs
```

system.boottime

 システムの起動時刻。
 戻り値: 整数 (*Unix* タイムスタンプ)。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、MacOS X、OpenBSD、NetBSD。

system.cpu.discovery

 検出された CPU/CPU コアの一覧です。
ローレベルディスカバリに使用されます。
 戻り値: *JSON string*。
 サポートされているプラットフォームを参照してください。

system.cpu.intr

 デバイス割り込み数。
 戻り値: 整数。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD、NetBSD。

system.cpu.load[<cpu>,<mode>]

 CPU 負荷。
 戻り値: *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **cpu** - 指定可能な値: *all*（デフォルト）または *percpu*（オンライン CPU 数で割った合計負荷）;
- **mode** - 指定可能な値: *avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*。

例:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 CPU の数。
 戻り値: 整数。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。

パラメータ:

- **type** - 指定可能な値: *online*（デフォルト）または *max*

max タイプのパラメータは、Linux、FreeBSD、Solaris、MacOS X でのみサポートされています。

例:

```
system.cpu.num
```

system.cpu.switches

 コンテキストスイッチの回数。
 戻り値: 整数。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD、NetBSD。

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 CPU 使用率（パーセンテージ）。
 戻り値: *Float*。
 サポートされているプラットフォーム: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows。

パラメータ:

- **cpu** - *<CPU number>* または *all*（デフォルト）;
- **type** - 指定可能な値: *user*（デフォルト）、*idle*、*nice*、*system*、*iowait*、*interrupt*、*softirq*、*steal*、*spin*（OpenBSD のみ）、*guest*（Linux カーネル 2.6.24 以降）、または *guest_nice*（Linux カーネル 2.6.33 以降）;
- **mode** - 指定可能な値: *avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*;
- **logical or physical** - 指定可能な値: *logical*（デフォルト）または *physical*。このパラメータは AIX でのみサポートされています。

コメント:

- *nice* タイプのパラメータは、Linux、FreeBSD、HP-UX、OpenBSD、NetBSD でのみサポートされています。
- *iowait* タイプのパラメータは、Linux 2.6 以降、Solaris、AIX でのみサポートされています。
- *interrupt* タイプのパラメータは、Linux 2.6 以降、FreeBSD、OpenBSD でのみサポートされています。
- *softirq*、*steal*、*guest*、*guest_nice* タイプのパラメータは、Linux 2.6 以降でのみサポートされています。
- *avg5* および *avg15* モードのパラメータは、Linux、FreeBSD、Solaris、HP-UX、AIX、OpenBSD、NetBSD でサポートされています。

例:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 システムのホスト名。
 戻り値: 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **type** - 指定可能な値: *netbios* (Windows でのデフォルト)、*host* (Linux でのデフォルト)、*shorthost* (最初のドットより前のホスト名の部分を返し、ドットを含まない名前の場合は文字列全体を返す)、*fqdn* (完全修飾ドメイン名を返す);
- **transform** - 指定可能な値: *none* (デフォルト) または *lower* (小文字に変換)。

値は、uname() システム API の出力から nodename を取得して得られます。

戻り値の例:

```
system.hostname → linux-w7x1
system.hostname → example.com
system.hostname[shorthost] → example
system.hostname → WIN-SERV2008-I6
system.hostname[host] → Win-Serv2008-I6LonG
system.hostname[host,lower] → win-serv2008-i6long
system.hostname[fqdn,lower] → blog.zabbix.com
```

system.hw.chassis[<info>]

 シャーシ情報。
 戻り値: 文字列。
 サポートされるプラットフォーム: Linux。

パラメータ:

- **info** - 指定可能な値: *full* (デフォルト)、*model*、*serial*、*type*、または *vendor*

コメント:

- このアイテムキーは、SMBIOS テーブルが利用可能であることに依存します。
- sysfs から DMI テーブルの読み取りを試み、sysfs へのアクセスに失敗した場合は、メモリから直接読み取ろうとします。
- 値は sysfs またはメモリを読み取って取得されるため、**root** 権限が必要です。

例:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 CPU 情報です。
 戻り値: 文字列または 整数。
 対応プラットフォーム: Linux。

パラメータ:

- **cpu** - <*CPU* 番号 > または *all*（デフォルト）;
- **info** - 指定可能な値: *full*（デフォルト）、*curfreq*、*maxfreq*、*model* または *vendor*。

コメント:

- /proc/cpuinfo および /sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq から情報を収集します。
- CPU 番号と *curfreq* または *maxfreq* を指定した場合、数値（Hz）が返されます。

例:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 PCI または USB デバイスの一覧です。
 戻り値: *Text*。
 対応プラットフォーム: Linux。

パラメータ:

- **type** - *pci*（デフォルト）または *usb*

lspci または lsusb ユーティリティのいずれかの出力を返します（いずれもパラメータなしで実行されます）。

例:

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

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

 MAC アドレスの一覧。
 戻り値: 文字列。
 サポートされるプラットフォーム: Linux。

パラメータ:

- **interface** - *all* (デフォルト) または正規表現
- **format** - *full* (デフォルト) または *short*

コメント:

- 指定された `interface` 正規表現 に一致する名前のインターフェースの MAC アドレスを一覧表示します（*all* の場合はすべてのインターフェースを一覧表示します）。
- `format` に *short* を指定した場合、インターフェース名および同一の MAC アドレスは一覧表示されません。

例:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 システム時刻です。
 戻り値: `type` が *utc* の場合は *Integer*、`type` が *local* の場合は *String*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **type** - 指定可能な値: *utc* - （デフォルト）エポック（1970 年 1 月 1 日 00:00:00 UTC）からの経過時間を秒単位で表したもの、または *local* - 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm' 形式の時刻

Zabbix エージェントではパッシブチェックとして使用する必要があります。Zabbix エージェント 2 ではアクティブチェックとして設定することもできます。

例:

```
system.localtime[local] # このキーを使用してアイテムを作成し、それを使って *Clock* ダッシュボードウィジェットにホスト時刻
```

system.run[command,<mode>]

 ホスト上で指定したコマンドを実行します。
 戻り値: コマンドの結果のテキスト、または `mode` が *nowait* の場合は 1（コマンドの結果に関係なく）。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **command** - 実行するコマンド;
- **mode** - 指定可能な値: *wait* - 実行終了まで待機（デフォルト）、または *nowait* - 待機しない。

コメント:

- このアイテムはデフォルトで無効になっています。有効化する方法はこちらを参照してください。
- アイテムの戻り値は、コマンドによって生成された標準出力と標準エラー出力を合わせたものです。終了コードのチェックは実行されません。
- 正しく処理されるためには、コマンドの戻り値は `text` データ型である必要があります。空の結果も許可されます。
- 戻り値は 16MB に制限されます（切り捨てられる末尾の空白を含む）。また、データベースの制限も適用されます。
- あわせて参照: コマンド実行。

例:

```
system.run[ls -l /] # ルートディレクトリの詳細なファイル一覧を返す
```

system.stat[resource,<type>]

 システム統計情報です。
 戻り値: *Integer* または *float*。
 サポートされているプラットフォーム: AIX。

パラメータ:

- **ent** - このパーティションに割り当てられている受給可能なプロセッサーユニット数 (float);
- **kthr,<type>** - カーネルスレッドの状態に関する情報:
r - 実行可能なカーネルスレッドの平均数 (float)
b - Virtual Memory Manager の待機キューに置かれたカーネルスレッドの平均数 (float)
- **memory,<type>** - 仮想メモリおよび実メモリの使用状況に関する情報:
avm - アクティブな仮想ページ数 (integer)
fre - フリーリストのサイズ (integer)
- **page,<type>** - ページフォールトおよびページングアクティビティに関する情報:
fi - 1 秒あたりのファイルページイン数 (float)
fo - 1 秒あたりのファイルページアウト数 (float)
pi - ページング領域からページインされたページ数 (float)
po - ページング領域へページアウトされたページ数 (float)
fr - 解放されたページ数 (ページ置換) (float)
sr - ページ置換アルゴリズムによってスキャンされたページ数 (float)

- **faults,<type>** - トラップおよび割り込み率:
in - デバイス割り込み (float)
sy - システムコール (float)
cs - カーネルスレッドのコンテキストスイッチ (float)
- **cpu,<type>** - プロセッサー時間の使用率の内訳:
us - ユーザー時間 (float)
sy - システム時間 (float)
id - アイドル時間 (float)
wa - システムに未処理のディスク/NFS I/O リクエストがある間のアイドル時間 (float)
pc - 消費された物理プロセッサー数 (float)
ec - 消費された割り当て容量の割合 (float)
lbusy - ユーザーレベルおよびシステムレベルでの実行中に発生した論理プロセッサー使用率の割合を示します (float)
app - 共有プール内で利用可能な物理プロセッサーを示します (float)
- **disk,<type>** - ディスク統計情報:
bps - ドライブに対して転送されたデータ量 (読み取りまたは書き込み) を 1 秒あたりのバイト数で示します (integer)
tps - 物理ディスク/テープに対して発行された 1 秒あたりの転送回数を示します (float)

コメント:

- これらのアイテムには、以下の制限があることに注意してください:
 `system.stat[cpu,app]` - AIX LPAR のタイプが”Shared” の場合のみサポートされます。
 `system.stat[cpu,ec]` - AIX LPAR のタイプが”Shared” および”Dedicated” の場合にサポートされます (”Dedicated” は常に 100 (percent) を返します)。
 `system.stat[cpu,lbusy]` - AIX LPAR のタイプが”Shared” の場合のみサポートされます。
 `system.stat[cpu,pc]` - AIX LPAR のタイプが”Shared” および”Dedicated” の場合にサポートされます。
 `system.stat[ent]` - AIX LPAR のタイプが”Shared” および”Dedicated” の場合にサポートされます。

system.sw.arch

 ソフトウェアアーキテクチャ情報。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

この情報は `uname()` 関数から取得されます。

例:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 オペレーティングシステムの情報です。
 戻り値: *String*。
 サポートされているプラットフォーム: Linux、Windows。

パラメータ:

- **info** - 指定可能な値: *full*（デフォルト）、*short*、または *name*

info は以下から取得されます（すべてのディストリビューションですべてのファイルやオプションが存在するわけではないことに注意してください）:

- Linux では `/proc/version`（*full*）;
- Linux では `/proc/version_signature`（*short*）;
- Linux では、これをサポートするシステム上の `/etc/os-release` の PRETTY_NAME パラメータ、または `/etc/issue.net`（*name*）;
- Windows では `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` レジストリキー。

例:

```
system.sw.os[short] → Ubuntu 2.6.35-28.50-generic 2.6.35.11
system.sw.os[full] → [s|Windows 10 Enterprise 22621.1.asd64fre.ni_release.220506-1250 Build 22621.963]
```

system.sw.os.get

 オペレーティングシステムに関する詳細情報（バージョン、種類、ディストリビューション名、マイナーバージョン、メジャーバージョンなど）。
 戻り値: *JSON string*。
 サポートされているプラットフォーム: Linux、Windows。

system.sw.packages[<regexp>,<manager>,<format>]

 インストール済みパッケージの一覧です。
 戻り値: *Text*。
 対応プラットフォーム: Linux。

パラメータ:

- **regexp** - *all* (デフォルト) または正規表現。
- **manager** - *all* (デフォルト) またはパッケージマネージャー。
- **format** - *full* (デフォルト) または *short*。

コメント:

- 指定した正規表現に名前が一致するインストール済みパッケージを（アルファベット順に）一覧表示します（*all* を指定するとすべて表示します）。
- 対応するパッケージマネージャー(実行コマンド):
dpkg (dpkg --get-selections)
pkgtool (ls /var/log/packages)
rpm (rpm -qa)
pacman (pacman -Q)
portage
- `format` に *full* を指定した場合、パッケージはパッケージマネージャーごとにグループ化されます（各マネージャーは、角括弧内の名前で始まる別々の行に表示されます）。

- `format` に *short* を指定した場合、パッケージはグループ化されず、1 行に一覧表示されます。

例:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 インストール済みパッケージの詳細な一覧です。
 戻り値: *JSON string*。
 対応プラットフォーム: Linux。

パラメータ:

- **regexp** - *all*（デフォルト）または通常のexpression。
- **manager** - *all*（デフォルト）またはパッケージマネージャー（指定可能な値: *rpm*、*dpkg*、*pkgtool*、*pacman*、または *portage*）。

コメント:

- 指定された正規表現に名前が一致するインストール済みパッケージについて、整形されていない JSON を返します。
- 出力はオブジェクトの配列で、各オブジェクトには次のキーが含まれます: name、manager、version、size、architecture、buildtime、installtime（詳細はこちらを参照）。

system.swap.in[<device>,<type>]

 スワップイン（デバイスからメモリへの読み込み）の統計情報です。
 戻り値: *Integer*。
 サポートされているプラットフォーム: Linux、FreeBSD、OpenBSD。

パラメータ:

- **device** - スワップに使用されるデバイス（Linux のみ）または *all*（デフォルト）を指定します。
- **type** - 指定可能な値: *count*（スワップイン回数。Linux 以外のプラットフォームでのデフォルト）、*sectors*（スワップインされたセクター数）、または *pages*（スワップインされたページ数。Linux でのデフォルト）。

コメント:

- この情報の取得元は次のとおりです:
/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4)
/proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)
- *pages* は、device が指定されていない場合にのみ機能します。
- *sectors* 型のパラメータは Linux でのみサポートされています。

例:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 スワップアウト（メモリからデバイスへの書き出し）の統計情報です。
 戻り値: *Integer*。
 対応プラットフォーム: Linux、FreeBSD、OpenBSD。

パラメータ:

- **device** - スワップに使用するデバイス（Linux のみ）または *all*（デフォルト）を指定します。
- **type** - 指定可能な値: *count*（スワップアウト回数。Linux 以外のプラットフォームでのデフォルト）、*sectors*（スワップアウトされたセクター数）、または *pages*（スワップアウトされたページ数。Linux でのデフォルト）。

コメント:

- この情報の取得元は次のとおりです:
/proc/swaps, /proc/partitions, /proc/stat (Linux 2.4)
/proc/swaps, /proc/diskstats, /proc/vmstat (Linux 2.6)
- *pages* は、device が指定されていない場合にのみ動作します。
- *sectors* 型パラメータは Linux でのみサポートされています。

例:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 スワップ領域のサイズを、バイト単位または合計に対する割合で返します。
 戻り値: バイトの場合は *Integer*、割合の場合は *Float*。
 対応プラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD、Windows。

パラメータ:

- **device** - スワップに使用するデバイス（FreeBSD のみ）または *all*（デフォルト）を指定します。
- **type** - 指定可能な値: *free*（空きスワップ領域、デフォルト）、*pfree*（空きスワップ領域、割合）、*pused*（使用済みスワップ領域、割合）、*total*（スワップ領域の合計）、または *used*（使用済みスワップ領域）。

コメント:

- Windows では、スワップサイズが 0 の場合、*pfree* と *pused* はサポートされません。
- device が指定されていない場合、Zabbix エージェントはスワップデバイス（ファイル）のみを考慮し、物理メモリは無視します。たとえば Solaris システムでは、`swap -s` コマンドには物理メモリの一部とスワップデバイスが含まれます（`swap -l` とは異なります）。

例:

```
system.swap.size[,pfree] → 空きスワップ領域の割合
```

system.uname

 システムの識別情報。
 戻り値: 文字列。
 サポートされているプラットフォームを参照してください。

コメント:

- UNIX では、このアイテムの値は uname() システムコールによって取得されます。
- Windows では、このアイテムは OS アーキテクチャを返します。一方、UNIX では CPU アーキテクチャを返します。

例:

```
system.uname → FreeBSD localhost 4.2-RELEASE FreeBSD 4.2-RELEASE #0: Mon Nov i386
system.uname → Windows ZABBIX-WIN 6.0.6001 Microsoft® Windows Server® 2008 Standard Service Pack 1 x86
```

system.uptime

 システムの稼働時間（秒単位）。
 戻り値: *Integer*。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、AIX、MacOS X、OpenBSD、NetBSD、Windows。

アイテムの設定では、読みやすい値を取得するために、単位として **s** または **uptime** を使用します。

system.users.num

 ログインしているユーザー数。
 戻り値: 整数。
 サポートされているプラットフォームを参照してください。

値の取得には、エージェント側で **who** コマンドが使用されます。

vfs.dev.discovery

 ブロックデバイスとそのタイプの一覧です。
ローレベルディスカバリに使用されます。
 戻り値: *JSON string*。
 サポートされているプラットフォーム: Linux。

vfs.dev.read[<device>,<type>,<mode>]

 ディスクの読み取り統計情報です。
 戻り値: `type` が *sectors*、*operations*、*bytes* の場合は *Integer*、`type` が *sps*、*ops*、*bps* の場合は *Float*。
 対応プラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD。

パラメータ:

- **device** - ディスクデバイス（デフォルトは *all* [3]）;
- **type** - 使用可能な値: *sectors*、*operations*、*bytes*、*sps*、*ops*、または *bps*（*sps*、*ops*、*bps* はそれぞれ 1 秒あたりの sectors、operations、bytes を表します）;
- **mode** - 使用可能な値: *avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*。このパラメータは、`type` が sps、ops、bps の場合にのみサポートされます。

コメント:

- 更新間隔が 3 時間以上[2] の場合、このアイテムは常に'0' を返します。
- *sectors* および *sps* の type パラメータは Linux でのみサポートされます。
- *ops* の type パラメータは Linux および FreeBSD でのみサポートされます。
- *bps* の type パラメータは FreeBSD でのみサポートされます。
- *bytes* の type パラメータは FreeBSD、Solaris、AIX、OpenBSD でのみサポートされます。
- mode パラメータは Linux、FreeBSD でのみサポートされます。
- 相対デバイス名（例: sda）および省略可能な /dev/ プレフィックス付きの名前（例: /dev/sda）を使用できます。
- LVM 論理ボリュームがサポートされます。
- OS ごとの'type' パラメータのデフォルト値:
AIX - operations
FreeBSD - bps
Linux - sps
OpenBSD - operations
Solaris - bytes
- 対応プラットフォームにおける *sps*、*ops*、*bps* は 1024 デバイスまでに制限されます（個別デバイス 1023 個と *all* 用 1 個）。

例:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 ディスクの書き込み統計情報です。
 戻り値: type が *sectors*、*operations*、*bytes* の場合は *Integer*、type が *sps*、*ops*、*bps* の場合は *Float*。
 対応プラットフォーム: Linux、FreeBSD、Solaris、AIX、OpenBSD。

パラメータ:

- **device** - ディスクデバイス（デフォルトは *all* [3]）;
- **type** - 指定可能な値: *sectors*、*operations*、*bytes*、*sps*、*ops*、または *bps*（*sps*、*ops*、*bps* はそれぞれ 1 秒あたりの sectors、operations、bytes を表します）;
- **mode** - 指定可能な値: *avg1*（1 分平均、デフォルト）、*avg5*、または *avg15*。このパラメータは、type が sps、ops、bps の場合にのみサポートされます。

コメント:

- 更新間隔が 3 時間以上[2] の場合、このアイテムは常に'0' を返します。
- *sectors* および *sps* の type パラメータは Linux でのみサポートされます。
- *ops* の type パラメータは Linux および FreeBSD でのみサポートされます。
- *bps* の type パラメータは FreeBSD でのみサポートされます。
- *bytes* の type パラメータは FreeBSD、Solaris、AIX、OpenBSD でのみサポートされます。
- mode パラメータは Linux、FreeBSD でのみサポートされます。
- 相対デバイス名（例: sda）および省略可能な /dev/ プレフィックス付きの名前（例: /dev/sda）を使用できます。
- LVM 論理ボリュームがサポートされます。
- OS ごとの'type' パラメータのデフォルト値:
AIX - operations
FreeBSD - bps
Linux - sps
OpenBSD - operations
Solaris - bytes
- サポート対象プラットフォーム上の *sps*、*ops*、*bps* は 1024 デバイス（個別 1023 と *all* 用 1 つ）に制限されます。

例:

```
vfs.dev.write[,operations]
```

vfs.dir.count[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 ディレクトリエントリ数です。
 戻り値: *Integer*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **dir** - ディレクトリへの絶対パス。
- **regex incl** - 含めるエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合はすべてを含めます（デフォルト値）。
- **regex excl** - 除外するエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合は何も除外しません（デフォルト値）。
- **types incl** - カウントするディレクトリエントリのタイプ。指定可能な値: *file* - 通常ファイル、*dir* - サブディレクトリ、*sym* - シンボリックリンク、*sock* - ソケット、*bdev* - ブロックデバイス、*cdev* - キャラクタデバイス、*fifo* - FIFO、*dev* - "bdev,cdev" と同義、*all* - すべてのタイプ（デフォルト）、すなわち"file,dir,sym,sock,bdev,cdev,fifo"。複数のタイプはカンマで区切り、引用符で囲む必要があります。
- **types excl** - カウントしないディレクトリエントリのタイプ（types incl を参照）。あるエントリタイプが types incl と types excl の両方に含まれている場合、そのタイプのディレクトリエントリはカウントされません。
- **max depth** - 走査するサブディレクトリの最大深度:
-1（デフォルト）- 無制限、
0 - サブディレクトリへは降りません。
- **min size** - カウント対象となるファイルの最小サイズ（バイト単位）。これより小さいファイルはカウントされません。メモリサフィックスを使用できます。
- **max size** - カウント対象となるファイルの最大サイズ（バイト単位）。これより大きいファイルはカウントされません。メモリサフィックスを使用できます。
- **min age** - カウント対象となるディレクトリエントリの最小経過時間（秒単位）。これより新しいエントリはカウントされません。時間サフィックスを使用できます。
- **max age** - カウント対象となるディレクトリエントリの最大経過時間（秒単位）。これより古い、または同じだけ古いエントリはカウントされません（更新時刻）。時間サフィックスを使用できます。
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。そのディレクトリ内のすべての内容が除外されます（regex_excl とは対照的です）。

コメント:

- Windows では、このアイテムは UNC パスをたどります。これはセキュリティリスクになる可能性があります。
- %APP_HOME%、$HOME、%TEMP% などの環境変数はサポートされていません。
- 疑似ディレクトリ"." および".." は決してカウントされません。
- ディレクトリ走査ではシンボリックリンクは決してたどられません。
- エントリ数の計算時には regex incl と regex excl の両方がファイルとディレクトリに適用されますが、走査するサブディレクトリを選択する際には無視されます（regex incl が "(?i)^.+\.zip$" で max depth が設定されていない場合、すべてのサブディレクトリが走査されますが、zip タイプのファイルのみがカウントされます）。

- 実行時間はエージェントの設定にあるタイムアウト値（3 秒）によって制限されます。大規模なディレクトリ走査はそれより長くかかる場合があるため、データは返されず、このアイテムは未サポートになります。部分的なカウントは返されません。
- サイズでフィルタリングする場合、意味のあるサイズを持つのは通常ファイルだけです。Linux および BSD では、ディレクトリにもゼロ以外のサイズがあります（通常は数 KB）。デバイスのサイズはゼロです。たとえば **/dev/sda1** のサイズは対応するパーティションサイズを反映しません。したがって、`<min_size>` と `<max_size>` を使用する場合は、予期しない結果を避けるために `<types_incl>` を*"file"* に指定することを推奨します。

例:

```
vfs.dir.count[/dev] #/dev 内のデバイス数を監視します（Linux）
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] # 一時ディレクトリ内のファイル数を監視します
```

vfs.dir.get[dir,<regex incl>,<regex excl>,<types incl>,<types excl>,<max depth>,<min size>,<max size>,<min age>,<max age>,<regex excl dir>]

 ディレクトリエントリの一覧です。
 戻り値: *JSON string*。
 対応プラットフォームを参照してください。

パラメータ:

- **dir** - ディレクトリへの絶対パス。
- **regex incl** - 含めるエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合はすべてを含みます（デフォルト値）。
- **regex excl** - 除外するエンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合は何も除外しません（デフォルト値）。
- **types incl** - 一覧表示するディレクトリエントリのタイプ。指定可能な値: *file* - 通常ファイル、*dir* - サブディレクトリ、*sym* - シンボリックリンク、*sock* - ソケット、*bdev* - ブロックデバイス、*cdev* - キャラクタデバイス、*fifo* - FIFO、*dev* - "bdev,cdev" と同義、*all* - すべてのタイプ（デフォルト）、すなわち"file,dir,sym,sock,bdev,cdev,fifo"。
  複数のタイプはカンマで区切り、引用符で囲む必要があります。
- **types excl** - 一覧表示しないディレクトリエントリのタイプ（`types incl` を参照）。
  あるエントリタイプが `types incl` と `types excl` の両方に含まれている場合、そのタイプのディレクトリエントリは一覧表示されません。
- **max depth** - 走査するサブディレクトリの最大深度:
-1（デフォルト）- 無制限、
0 - サブディレクトリへは降りません。
- **min size** - 一覧表示するファイルの最小サイズ（バイト単位）。
  これより小さいファイルは一覧表示されません。
  メモリサフィックスを使用できます。
- **max size** - 一覧表示するファイルの最大サイズ（バイト単位）。
  これより大きいファイルは一覧表示されません。
  メモリサフィックスを使用できます。
- **min age** - 一覧表示するディレクトリエントリの最小経過時間（秒単位）。
  これより新しいエントリは一覧表示されません。
  時間サフィックスを使用できます。
- **max age** - 一覧表示するディレクトリエントリの最大経過時間（秒単位）。
  これより古い、または同じだけ古いエントリは一覧表示されません（更新時刻）。
  時間サフィックスを使用できます。
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。
  そのディレクトリのすべての内容が除外されます（`regex excl` とは対照的です）。

コメント:

- Windows では、このアイテムは UNC パスをたどります。これはセキュリティリスクになる可能性があります。
- %APP_HOME%、$HOME、%TEMP% などの環境変数はサポートされていません。
- 疑似ディレクトリ"." および".." は一覧表示されません。
- ディレクトリ走査ではシンボリックリンクはたどられません。
- エントリ一覧の生成時には `regex incl` と `regex excl` の両方がファイルとディレクトリに適用されますが、走査対象のサブディレクトリを選択する際には無視されます（たとえば `regex incl` が "(?i)^.+\.zip$" で `max depth` が設定されていない場合、すべてのサブディレクトリが走査されますが、カウントされるのは zip タイプのファイルのみです）。
- 実行時間は、エージェントの設定にある timeout 値によって制限されます。
  大規模なディレクトリ走査はそれより長くかかる可能性があるため、データは返されず、このアイテムは未サポートになります。
  部分的な一覧は返されません。
- サイズでフィルタリングする場合、意味のあるサイズを持つのは通常ファイルのみです。
  Linux および BSD では、ディレクトリも 0 以外のサイズを持ちます（通常は数 KB）。
  デバイスのサイズは 0 です。たとえば **/dev/sda1** のサイズは対応するパーティションサイズを反映しません。
  そのため、`min size` と `max size` を使用する場合は、予期しない結果を避けるために `types incl` を*"file"* に指定することを推奨します。

例:

```
vfs.dir.get[/dev] #/dev 内のデバイス一覧を取得します（Linux）
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] # 一時ディレクトリ内のファイル一覧を取得します
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 ディレクトリのサイズ（バイト単位）。
 戻り値: *Integer*。
 対応プラットフォーム: Linux、Windows。このアイテムは、他の UNIX 系プラットフォームでも動作する場合があります。

パラメータ:

- **dir** - ディレクトリへの絶対パス。
- **regex incl** - 含める対象エンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合はすべてを含めます（デフォルト値）。
- **regex excl** - 除外する対象エンティティ（ファイル、ディレクトリ、シンボリックリンク）の名前パターンを記述する正規表現。空の場合は何も除外しません（デフォルト値）。
- **mode** - 指定可能な値: *apparent*（デフォルト）- ディスク使用量ではなく見かけ上のファイルサイズを取得します（`du -sb dir` と同様の動作）、*disk* - ディスク使用量を取得します（`du -s -B1 dir` と同様の動作）。du コマンドとは異なり、vfs.dir.size アイテムはディレクトリサイズの計算時に隠しファイルも考慮します（dir 内で `du -sb .[^.]* *` を実行した場合と同様の動作）。
- **max depth** - 走査するサブディレクトリの最大深度: **-1**（デフォルト）- 無制限、**0** - サブディレクトリには降りません。
- **regex excl dir** - 除外するディレクトリの名前パターンを記述する正規表現。このディレクトリ内のすべての内容が除外されます（`regex excl` とは異なります）

コメント:

- Windows では、このアイテムは UNC パスをたどります。これはセキュリティリスクとなる可能性があります。
- 少なくとも *zabbix* ユーザーに読み取り権限があるディレクトリのみが計算対象となります。読み取り権限のみのディレクトリでは、ディレクトリ自体のサイズのみが計算されます。読み取り権限と実行権限があるディレクトリでは、内容を含めて計算されます。
- 大きなディレクトリや低速なドライブでは、このアイテムはエージェント およびサーバー/プロキシ 設定ファイルの Timeout 設定によりタイムアウトする場合があります。必要に応じてタイムアウト値を増やしてください。
- ファイルサイズの上限は、large file support に依存します。

例:

```
vfs.dir.size[/tmp,log] #/tmp 内で名前に 'log' を含むすべてのファイルのサイズを計算します
vfs.dir.size[/tmp,log,^.+\.old$] #/tmp 内で名前に 'log' を含み、名前の末尾が '.old' のファイルを除外したすべてのファイ
```

vfs.file.cksum[file,<mode>]

 UNIX cksum アルゴリズムで計算されたファイルのチェックサムです。
 戻り値: `mode` が *crc32* の場合は *Integer*、`mode` が *md5*、*sha256* の場合は *String*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス。
- **mode** - *crc32*（デフォルト）、*md5*、または *sha256*。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの上限は、large file support に依存します。

例:

```
vfs.file.cksum[/etc/passwd]
```

戻り値の例（それぞれ crc32/md5/sha256）:

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 ファイルの内容を取得します[7]。
 戻り値: テキスト。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス。
- **encoding** - コードページの識別子。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- 戻り値は 16MB に制限されます（切り捨てられる末尾の空白文字を含む）。また、データベースの制限も適用されます。
- ファイルが空である場合、または LF/CR 文字のみを含む場合は、空文字列が返されます。
- バイトオーダーマーク（BOM）は出力に含まれません。

例:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 ファイルが存在するかどうかを確認します。
 戻り値: 0 - 見つからない; 1 - 指定したタイプのファイルが存在する。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス。
- **types incl** - 含めるファイルタイプのリスト。指定可能な値: *file*（通常ファイル。types_excl が設定されていない場合のデフォルト）、*dir*（ディレクトリ）、*sym*（シンボリックリンク）、*sock*（ソケット）、*bdev*（ブロックデバイス）、*cdev*（キャラクタデバイス）、*fifo*（FIFO）、*dev*（"bdev,cdev" と同義）、*all*（上記すべてのタイプ。types_excl が設定されている場合のデフォルト）。
- **types excl** - 除外するファイルタイプのリスト。指定可能な値については types_incl を参照してください（デフォルトでは何も除外されません）

コメント:

- Windows では、このアイテムは UNC パスをたどります。これはセキュリティリスクとなる可能性があります。
- 複数のタイプを指定する場合はカンマで区切り、セット全体を引用符"" で囲む必要があります。
- 同じタイプが <types_incl> と <types_excl> の両方に含まれている場合、そのタイプのファイルは除外されます。
- ファイルサイズの上限は、large file support に依存します。

例:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 ファイルに関する情報を返します。
 戻り値: *JSON* 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- UNIX 系システムでサポートされるファイルタイプ: 通常ファイル、ディレクトリ、シンボリックリンク、ソケット、ブロックデバイス、キャラクタデバイス、FIFO。
- ファイルサイズの上限は、large file support に依存します。

例:

```
vfs.file.get[/etc/passwd] # /etc/passwd ファイルに関する情報（タイプ、ユーザー、パーミッション、SID、uid など）を含む
```

vfs.file.md5sum[file]

 ファイルの MD5 チェックサム。
 戻り値: 文字列（ファイルの MD5 ハッシュ）。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの上限は、large file supportに依存します。

例:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

戻り値の例:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 ファイルの所有者を取得します。
 戻り値: 文字列。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス。
- **ownertype** - *user*（デフォルト）または *group*（Unix のみ）。
- **resulttype** - *name*（デフォルト）または *id*。id の場合、Unix では uid/gid、Windows では SID を返します。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの制限は、large file support に依存します。

例:

```
vfs.file.owner[/tmp/zabbix_server.log] #return the file owner of /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #return the file owner ID of /tmp/zabbix_server.log
```

vfs.file.permissions[file]

 UNIX パーミッションの 8 進数を含む 4 桁の文字列を返します。
 戻り値: 文字列。
 サポートされているプラットフォーム: Linux。このアイテムは、他の UNIX 系プラットフォームでも動作する場合があります。

パラメータ:

- **file** - ファイルへのフルパス

ファイルサイズの上限は、large file support に依存します。

例:

```
vfs.file.permissions[/etc/passwd] # /etc/passwd のパーミッションを返します。例: '0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 ファイル内の文字列を取得します[7]。
 戻り値: 一致した文字列を含む行、またはオプションの output パラメータで指定された値。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス。
- **regexp** - 必要なパターンを記述する正規表現。
- **encoding** - コードページの識別子。
- **start line** - 検索を開始する最初の行番号（デフォルトではファイルの先頭行）。
- **end line** - 検索する最後の行番号（デフォルトではファイルの最終行）。
- **output** - オプションの出力書式テンプレート。
  \0 エスケープシーケンスは、一致したテキスト部分（一致開始位置の最初の文字から一致終了位置の文字まで）に置き換えられます。一方、**\N**（N=1...9）エスケープシーケンスは、N 番目の一致グループに置き換えられます（N がキャプチャされたグループ数を超える場合は空文字列）。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの上限は、large file supportに依存します。
- 最初に一致した行のみが返されます。
- 式に一致する行がない場合は、空文字列が返されます。
- バイトオーダーマーク（BOM）は出力から除外されます。
- output パラメータを使用した内容の抽出はエージェント上で行われます。

例:

```
vfs.file.regexp[/etc/passwd,zabbix]
vfs.file.regexp[/path/to/some/file,"([0-9]+)$",,3,5,\1]
vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1] → ユーザー *zabbix* の ID を取得
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 ファイル内の文字列を検索します[7]。
 戻り値: 0 - 一致が見つからない; 1 - 見つかった。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス;
- **regexp** - 必要なパターンを記述する正規表現;
- **encoding** - コードページの識別子;
- **start line** - 検索を開始する最初の行番号 (デフォルトではファイルの先頭行);
- **end line** - 検索する最後の行番号 (デフォルトではファイルの最終行)。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの上限は、large file supportに依存します。
- バイトオーダーマーク (BOM) は無視されます。

例:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 ファイルサイズ（バイト単位）。
 戻り値: *Integer*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス。
- **mode** - 指定可能な値: *bytes*（デフォルト）または *lines*（空行もカウントされます）。

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルには、ユーザー *zabbix* の読み取り権限が必要です。
- ファイルサイズの上限は、大きなファイルのサポートに依存します。

例:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 ファイルの時刻情報です。
 戻り値: *Integer*（Unix タイムスタンプ）。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **file** - ファイルへのフルパス;
- **mode** - 指定可能な値:
modify（デフォルト）- ファイル内容が最後に変更された時刻,
access - ファイルが最後に読み取られた時刻,
change - ファイルのプロパティが最後に変更された時刻

コメント:

- Windows では、このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- ファイルサイズの制限は、large file support に依存します。

例:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 マウントされているファイルシステムの一覧と、そのタイプおよびマウントオプションです。ローレベルディスカバリに使用されます。
 戻り値: *JSON string*。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。

vfs.fs.get

 マウントされているファイルシステムの一覧を、そのタイプ、利用可能なディスク容量、inode 統計、およびマウントオプションとともに取得します。ローレベルディスカバリに使用できます。
 戻り値: *JSON string*。
 サポートされているプラットフォーム: Linux、FreeBSD、Solaris、HP-UX、AIX、MacOS X、OpenBSD、NetBSD、Windows。

コメント:

- inode 数が 0 のファイルシステムも報告されます。これは、動的 inode を持つファイルシステム（例: btrfs）で発生する場合があります。
- 参照: マウントされたファイルシステムのディスカバリ。

vfs.fs.inode[fs,<mode>]

 inode の数または割合。
 戻り値: 数の場合は *Integer*、割合の場合は *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **fs** - ファイルシステム。
- **mode** - 指定可能な値: *total* (デフォルト)、*free*、*used*、*pfree* (空き、割合)、または *pused* (使用中、割合)。

inode 数が 0 の場合、これは動的 inode を持つファイルシステム (例: btrfs) で発生することがあり、その場合 pfree/pused の値はそれぞれ”100” および”0” として報告されます。

例:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 ディスク容量を、バイト単位または全体に対する割合（パーセンテージ）で返します。
 戻り値: バイトの場合は *Integer*、割合の場合は *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **fs** - ファイルシステム。
- **mode** - 指定可能な値: *total*（デフォルト）、*free*、*used*、*pfree*（空き容量、パーセンテージ）、または *pused*（使用量、パーセンテージ）。

コメント:

- ファイルシステムがマウントされていない場合、マウントポイントが存在するローカルファイルシステムのサイズを返します。
- ファイルシステムの予約領域は考慮され、*free* モード使用時の空き容量には含まれません。

例:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 メモリサイズを、バイト単位または総量に対する割合で返します。
 戻り値: バイトの場合は *Integer*、割合の場合は *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **mode** - 指定可能な値: *total* (デフォルト), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (使用量、割合), *available*, または *pavailable* (利用可能量、割合)。

コメント:

- このアイテムは 3 つのカテゴリのパラメータを受け付けます:
  - *total* - メモリの総量;
  - プラットフォーム固有のメモリタイプ: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - 使用中および利用可能なメモリ量に関するユーザーレベルの推定値: *used*, *pused*, *available*, *pavailable*。
- *active* モードパラメータは、FreeBSD、HP-UX、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *anon*、*exec*、*file* モードパラメータは、NetBSD でのみサポートされています。
- *buffers* モードパラメータは、Linux、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- *cached* モードパラメータは、Linux、FreeBSD、AIX、OpenBSD、NetBSD でのみサポートされています。
- *inactive*、*wired* モードパラメータは、FreeBSD、MacOS X、OpenBSD、NetBSD でのみサポートされています。
- *pinned* モードパラメータは、AIX でのみサポートされています。
- *shared* モードパラメータは、Linux 2.4、FreeBSD、OpenBSD、NetBSD でのみサポートされています。
- このアイテムについては、追加の詳細も参照してください。

例:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>]

 Web ページの内容を取得します。
 戻り値: Web ページのソースをテキストとして返します（ヘッダーを含む）。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（scheme://host:port/path の形式。必須なのは *host* のみ）。使用可能な URL スキーム: *http*, *https*[4]。スキームが指定されていない場合は *http* として扱われます。URL が指定されている場合、path と port は空でなければなりません。認証が必要なサーバーに接続する際のユーザー名/パスワードの指定（例: http://user:password@www.example.com）は、cURL サポートがある場合にのみ可能です [4]。ホスト名では Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは /）;
- **port** - ポート番号（HTTP のデフォルトは 80）

コメント:

- host で指定されたリソースが存在しないか利用できない場合、このアイテムは未サポートになります。
- host にはホスト名、ドメイン名、IPv4 アドレス、または IPv6 アドレスを指定できます。ただし、IPv6 アドレスの場合は、Zabbix エージェントが IPv6 サポートを有効にしてコンパイルされている必要があります。

例:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
```

web.page.perf[host,<path>,<port>]

 完全な Web ページの読み込み時間（秒）。
 戻り値: *Float*。
 サポートされているプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（scheme://host:port/path の形式。必須なのは *host* のみ）。使用可能な URL スキーム: *http*、*https*[4]。スキームが指定されていない場合は *http* として扱われます。URL が指定されている場合、path と port は空でなければなりません。認証が必要なサーバーへの接続時にユーザー名/パスワードを指定すること（例: http://user:password@www.example.com）は、cURL サポートがある場合にのみ可能です [4]。ホスト名では Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは /）;
- **port** - ポート番号（HTTP のデフォルトは 80）

コメント:

- host で指定されたリソースが存在しないか利用できない場合、このアイテムは未サポートになります。
- host にはホスト名、ドメイン名、IPv4 アドレス、または IPv6 アドレスを指定できます。ただし、IPv6 アドレスを使用する場合、Zabbix エージェントは IPv6 サポートを有効にしてコンパイルされている必要があります。

例:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Web ページ上の文字列を検索します。
 戻り値: 一致した文字列、またはオプションの output パラメータで指定した値。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **host** - ホスト名または URL（scheme://host:port/path の形式。必須なのは *host* のみ）。使用可能な URL スキーム: *http*、*https*[4]。スキームが指定されていない場合は *http* として扱われます。URL が指定されている場合、path と port は空でなければなりません。認証が必要なサーバーへの接続時にユーザー名/パスワードを指定すること（例: http://user:password@www.example.com）は、cURL サポートがある場合にのみ可能です [4]。ホスト名では Punycode がサポートされています。
- **path** - HTML ドキュメントへのパス（デフォルトは /）;
- **port** - ポート番号（HTTP のデフォルトは 80）
- **regexp** - 必要なパターンを記述する正規表現;
- **length** - 返す文字数の最大値;
- **output** - オプションの出力書式テンプレート。**\0** エスケープシーケンスは一致したテキスト部分（マッチ開始位置の最初の文字からマッチ終了位置の文字まで）に置き換えられ、**\N**（N=1...9）エスケープシーケンスは N 番目の一致グループ（N がキャプチャされたグループ数を超える場合は空文字列）に置き換えられます。

コメント:

- host で指定したリソースが存在しないか利用できない場合、このアイテムは未サポートになります。

- host にはホスト名、ドメイン名、IPv4 アドレス、または IPv6 アドレスを指定できます。ただし、IPv6 アドレスの場合、Zabbix エージェントは IPv6 サポートを有効にしてコンパイルされている必要があります。
- output パラメータを使用したコンテンツ抽出はエージェント上で行われます。

例:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 エージェントのホストメタデータ。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

HostMetadata またはHostMetadataItem パラメータの値を返します。どちらも定義されていない場合は、空文字列を返します。

agent.hostname

 エージェントのホスト名です。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

戻り値:

- パッシブチェックの場合 - エージェント設定ファイルのHostname パラメータに記載された最初のホスト名。
- アクティブチェックの場合 - 現在のホスト名。

agent.ping

 エージェントの可用性チェックです。
 戻り値: 利用不可の場合は何も返さない、利用可能な場合は 1。
 サポートされているプラットフォームを参照してください。

ホストの利用不可を確認するには、**nodata()** トリガー関数を使用してください。

agent.variant

 Zabbix エージェントのバリアント（Zabbix エージェントまたは Zabbix エージェント 2）。
 戻り値: 1 - Zabbix エージェント; 2 - Zabbix エージェント 2。
 サポートされているプラットフォームを参照してください。

agent.version

 Zabbix エージェントのバージョン。
 戻り値: *String*。
 サポートされているプラットフォームを参照してください。

返される値の例:

```
6.0.3
```

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

 Zabbix サーバーまたはプロキシの内部メトリクスのセットを返します。ip と port が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON* 文字列。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **ip** - リモート照会されるサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）
- **port** - リモート照会されるサーバー/プロキシのポート（デフォルトは 10051）

コメント:

- このアイテムは、選択された内部メトリクスのセットを返します。詳細については、Zabbix 統計のリモート監視を参照してください。
- 統計情報のリクエストは、対象インスタンスの'StatsAllowedIP' server/proxy パラメータに一覧表示されているアドレスからのみ受け付けられることに注意してください。

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

 Zabbix サーバーまたはプロキシ上で遅延しているキュー内の監視対象アイテム数を返します。ip と port が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON string*。
 サポートされるプラットフォームを参照してください。

パラメータ:

- **ip** - リモート照会されるサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）
- **port** - リモート照会されるサーバー/プロキシのポート（デフォルトは 10051）
- **queue** - 定数（そのまま使用）
- **from** - 少なくともこの時間だけ遅延（デフォルトは 6 秒）
- **to** - 最大でこの時間まで遅延（デフォルトは無限大）

統計情報のリクエストは、対象インスタンスの'StatsAllowedIP' server/proxy パラメータに記載されたアドレスからのみ受け付けられることに注意してください。

注釈

[1] Linux 固有の注記です。Zabbix エージェントには、ファイルシステム */proc* への読み取り専用アクセス権が必要です。www.grsecurity.org のカーネルパッチは、非特権ユーザーのアクセス権を制限します。

[2] `vfs.dev.read[]`、`vfs.dev.write[]`: アイテムの値に 3 時間を超えてアクセスされない場合、Zabbix エージェントは「古い」デバイス接続を終了します。これは、システムにパスが動的に変化するデバイスがある場合や、デバイスが手動で取り外された場合に発生することがあります。また、これらのアイテムで更新間隔が 3 時間以上に設定されている場合、常に「0」を返すことにも注意してください。

[3] `vfs.dev.read[]`、`vfs.dev.write[]`: 最初のパラメータにデフォルトの *all* を使用した場合、このキーは集計統計を返します。これには、sda、sdb のようなすべてのブロックデバイス、それらのパーティション（sda1、sda2、sdb3...）、それらのブロックデバイス/パーティションに基づく複数デバイス（MD raid）、およびそれらのブロックデバイス/パーティションに基づく論理ボリューム（LVM）が含まれます。このような場合、返される値は絶対値ではなく、相対値（時間とともに変動する値）としてのみ考慮する必要があります。

[4] SSL (HTTPS) は、エージェントが cURL サポート付きでコンパイルされている場合にのみサポートされます。それ以外の場合、このアイテムは未サポートになります。

[5] `bytes` および `errors` の値は、Solaris 10 6/06 までの Solaris システム上のループバックインターフェースではサポートされていません。これは、バイト、エラー、および使用率の統計がカーネルによって保存および/または報告されないためです。ただし、net-snmp 経由で Solaris システムを監視している場合は、値が返されることがあります。これは、net-snmp が 1997 年頃までさかのぼる cmu-snmp のレガシーコードを含んでおり、インターフェース統計からバイト値の読み取りに失敗した際に、インターフェースのパケットカウンタ（ループバックインターフェースには存在します）に任意の値 308 を掛けて返すためです。これは、パケットの平均長が 308 オクテットであると仮定していますが、Solaris システムのループバックインターフェースにおける MTU 制限は 8892 バイトであるため、非常に大まかな推定です。これらの値が正しい、あるいは正確に近いものであると見なすべきではありません。これらは推定値です。Zabbix エージェントはこのような推測処理を行いませんが、net-snmp はこれらのフィールドに対して値を返します。

[6] Solaris で `/proc/pid/psinfo` から取得されるコマンドラインは 80 バイトに制限されており、プロセス開始時点のコマンドラインが含まれます。

[7] `vfs.file.contents[]`、`vfs.file.regexp[]`、`vfs.file.regmatch[]` アイテムは、ファイル内容の取得に使用できます。機密情報を含む特定のファイルへのアクセスを制限したい場合は、それらのファイルを閲覧する権限を持たないユーザーで Zabbix エージェントを実行してください。

コマンドラインユーティリティでの使用

コマンドラインから zabbix_agentd または zabbix_get を使用してアイテムキーをテストまたは使用する場合は、シェル構文も考慮する必要があることに注意してください。

例えば、キーの特定のパラメータを二重引用符で囲む必要がある場合は、二重引用符を明示的にエスケープする必要があります。そうしないと、二重引用符は特殊文字としてシェルによって取り除かれ、Zabbix ユーティリティに渡されません。

例:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

エンコーディング設定

取得したデータが破損しないようにするため、`encoding` パラメータでチェック（例: `vfs.file.contents`）の処理に使用する正しいエンコーディングを指定できます。サポートされているエンコーディングの一覧（コードページ識別子）は、libiconv（GNU Project）のドキュメント、または Microsoft Windows SDK の"Code Page Identifiers"のドキュメントで確認できます。Microsoft は一部のコードページを「managed アプリケーションでのみ利用可能」として扱うことがある点に注意してください。つまり、.NET ランタイム経由でのみ公開され、ネイティブ Win32 API では利用できません。Zabbix エージェントは独自のエンコーディング変換ロジックを実装しているため、ネイティブ Windows 関数で提供されていない場合でも、これらのコードページは Zabbix エージェントでサポートされます。

`encoding` パラメータでエンコーディングが指定されていない場合、以下の解決方法が適用されます。

- エンコーディングが指定されていない場合（または空文字列の場合）、UTF-8 とみなされ、データは「そのまま」処理されます。
- BOM 解析 - アイテム'vfs.file.contents'、'vfs.file.regexp'、'vfs.file.regmatch' に適用されます。ファイルの先頭にあるバイトオーダーマーク（BOM）を使用して、正しいエンコーディングの判定が試みられます。BOM が存在しない場合は、代わりに標準の解決方法（上記参照）が適用されます。

エージェントアイテムのトラブルシューティング

パッシブチェックの場合、サーバーからエージェントへのリクエストが先にタイムアウトし、その結果アイテムが値を取得できなくなることを防ぐため、次の点に注意してください。

- エージェントのバージョンがサーバーのバージョンより古い場合、アイテム設定 の *Timeout* 値（またはグローバルタイムアウト）を、エージェントの設定ファイル にある `Timeout` 値より大きくする必要がある場合があります。
- エージェントのバージョンがサーバーのバージョンより新しい場合、サーバーの設定ファイル にある `Timeout` 値を、エージェントの設定ファイル にある `Timeout` 値より大きくする必要がある場合があります。

1 Zabbix エージェント 2

Zabbix agent 2 は、Unix およびWindows の Zabbix エージェントでサポートされているすべてのアイテムキーをサポートしています。
このページでは、Zabbix agent 2 でのみ使用できる追加のアイテムキーについて、それらが属するプラグインごとにまとめて説明します。

アイテムキーは、パラメータおよび追加情報を省略した形で一覧表示されています。
詳細を確認するには、アイテムキーをクリックしてください。

Item key	Description	Plugin
ceph.df.details	クラスターのデータ使用量とプール間での分布。	Ceph
ceph.osd.stats	集計済みおよび OSD ごとの統計情報。	
ceph.osd.discovery	検出された OSD の一覧。	
ceph.osd.dump	OSD の使用量しきい値とステータス。	
ceph.ping	Ceph への接続を確立できるかどうかをテストします。	
ceph.pool.discovery	検出されたプールの一覧。	
ceph.status	クラスター全体のステータス。	
docker.container_info	コンテナに関する低レベル情報。	Docker
docker.container_stats	コンテナのリソース使用統計。	
docker.containers	コンテナの一覧を返します。	
docker.containers.discovery	コンテナの一覧を返します。ローレベルディスカバリに使用されます。	
docker.data.usage	現在のデータ使用量に関する情報。	
docker.images	イメージの一覧を返します。	
docker.images.discovery	イメージの一覧を返します。ローレベルディスカバリに使用されます。	
docker.info	システム情報。	
docker.ping	Docker デーモンが稼働しているかどうかをテストします。	
ember.get	要求されたデバイスの結果を返します。	Ember+
memcached.ping	接続が有効かどうかをテストします。	Memcached
memcached.stats	STATS コマンドの出力を取得します。	
mongodb.collection.stats	指定したコレクションのさまざまなストレージ統計を返します。	MongoDB
mongodb.collections.discovery	検出されたコレクションの一覧を返します。	
mongodb.collections.usage	コレクションの使用統計を返します。	
mongodb.connpool.stats	現在のデータベースインスタンスから、シャーディングクラスターまたはレプリカセットの他のメンバーへの開いている送信接続に関する情報を返します。	
mongodb.db.stats	指定したデータベースシステムの状態を反映する統計を返します。	
mongodb.db.discovery	検出されたデータベースの一覧を返します。	
mongodb.jumbo_chunks.count	jumbo チャンク数を返します。	
mongodb.oplog.stats	oplog から取得したデータを使用して、レプリカセットのステータスを返します。	
mongodb.ping	接続が有効かどうかをテストします。	
mongodb.rs.config	レプリカセットの現在の設定を返します。	
mongodb.rs.status	メソッドが実行されたメンバーの視点から見たレプリカセットのステータスを返します。	
mongodb.server.status	データベースの状態を返します。	
mongodb.sh.discovery	クラスター内に存在する検出済みシャードの一覧を返します。	
mongodb.version	データベースサーバーのバージョンを返します。	
mqtt.get	指定されたブローカーの特定のトピックまたは複数のトピック（ワイルドカード付き）を購読し、公開メッセージを待機します。	MQTT
mssql.availability.group.get	可用性グループを返します。	MSSQL
mssql.custom.query	カスタムクエリの結果を返します。	
mssql.db.get	利用可能なすべての MSSQL データベースを返します。	
mssql.job.status.get	ジョブのステータスを返します。	
mssql.last.backup.get	すべてのデータベースの最終バックアップ時刻を返します。	

Item key	Description	Plugin
mssql.local.db.get	Always On 可用性グループおよびレプリカ（プライマリまたはセカンダリ）に参加しており、接続先のサーバー上に存在するデータベースを返します。	
mssql.mirroring.get	ミラーリング情報を返します。	
mssql.nonlocal.db.get	Always On 可用性グループおよびレプリカ（プライマリまたはセカンダリ）に参加しており、他のサーバー上に存在するデータベースを返します（このデータベースは、接続先の SQL Server インスタンス上には存在しません）。	
mssql.perfcounter.get	パフォーマンスカウンターを返します。	
mssql.ping	接続が有効かどうかをテストします。	
mssql.quorum.get	クォーラム情報を返します。	
mssql.quorum.member.get	クォーラムメンバーを返します。	
mssql.replica.get	レプリカを返します。	
mssql.version	MSSQL のバージョンを返します。	
mysql.custom.query	カスタムクエリの結果を返します。	MySQL
mysql.db.discovery	MySQL データベースの一覧を返します。	
mysql.db.size	データベースサイズ（バイト単位）。	
mysql.get_status_variables	グローバルステータス変数の値。	
mysql.ping	接続が有効かどうかをテストします。	
mysql.replication.discovery	MySQL レプリケーションの一覧を返します。	
mysql.replication.get_slave_status	レプリケーションのステータス。	
mysql.version	MySQL のバージョン。	
net.dns.get	DNS クエリを実行し、詳細な DNS レコード情報を返します。	Network
nvml.device.count	GPU デバイス数。	NVIDIA GPU
nvml.device.decoder.utilization	GPU デバイスのデコーダー使用率。	
nvml.device.ecc.mode	GPU デバイスの ECC モード。	
nvml.device.encoder.stats.get	GPU デバイスのエンコーダー統計。	
nvml.device.encoder.utilization	GPU デバイスのエンコーダー使用率。	
nvml.device.energy.consumption	GPU デバイスのエネルギー消費量。	
nvml.device.errors.memory	GPU デバイスの ECC メモリエラー情報。	
nvml.device.errors.register	GPU デバイスの ECC レジスターエラー情報。	
nvml.device.fan.speed.avg	GPU デバイスの平均ファン速度。	
nvml.device.get	UUID と名前を含む検出済み GPU。	
nvml.device.graphics.frequency	GPU デバイスのグラフィックスクロック速度。	
nvml.device.memory.bar1.get	GPU デバイスの BAR1 メモリ情報。	
nvml.device.memory.fb.get	GPU デバイスのフレームバッファメモリ情報。	
nvml.device.memory.frequency	GPU デバイスのメモリクロック速度。	
nvml.device.pci.utilization	GPU デバイスの PCI 使用率。	
nvml.device.performance.state	GPU デバイスのパフォーマンス状態。	
nvml.device.power.limit	GPU デバイスの電力上限。	
nvml.device.power.usage	GPU デバイスの電力使用量。	
nvml.device.serial	GPU デバイスのシリアル番号。	
nvml.device.sm.frequency	GPU デバイスのストリーミングマルチプロセッサクロック速度。	
nvml.device.temperature	GPU デバイスの温度。	
nvml.device.utilization	GPU デバイスの使用率統計。	
nvml.device.video.frequency	GPU デバイスのビデオクロック速度。	
nvml.system.driver.version	NVIDIA ドライバーバージョン。	
nvml.version	NVML ライブラリのバージョン。	
oracle.diskgroups.stats	Automatic Storage Management (ASM) ディスクグループの統計を返します。	Oracle
oracle.diskgroups.discovery	ASM ディスクグループの一覧を返します。	
oracle.archive.info	アーカイブログの統計。	
oracle.cdb.info	Container Databases (CDB) の情報。	
oracle.custom.query	カスタムクエリの結果。	
oracle.datafiles.stats	データファイルの統計を返します。	
oracle.db.discovery	データベースの一覧を返します。	
oracle.fra.stats	Fast Recovery Area (FRA) の統計を返します。	
oracle.instance.info	インスタンス統計。	
oracle.pdb.info	Pluggable Databases (PDB) の情報。	
oracle.pdb.discovery	PDB の一覧を返します。	
oracle.pga.stats	Program Global Area (PGA) の統計を返します。	
oracle.ping	Oracle への接続を確立できるかどうかをテストします。	
oracle.proc.stats	プロセス統計を返します。	

Item key	Description	Plugin
oracle.redolog.info	制御ファイルから取得したログファイル情報。	
oracle.sga.stats	System Global Area (SGA) の統計を返します。	
oracle.sessions.stats	セッション統計を返します。	
oracle.sys.metrics	システムメトリクス値のセットを返します。	
oracle.sys.params	システムパラメータ値のセットを返します。	
oracle.ts.stats	表領域の統計を返します。	
oracle.ts.discovery	表領域の一覧を返します。	
oracle.user.info	Oracle ユーザー情報を返します。	
oracle.version	データベースサーバーのバージョンを返します。	
pgsql.autovacuum.count	autovacuum ワーカー数。	PostgreSQL
pgsql.archive	アーカイブ済みファイルに関する情報。	
pgsql.bgwriter	データベースクラスターのチェックポイント総数を、チェックポイントタイプ別に分類して返します。	
pgsql.cache.hit	PostgreSQL バッファキャッシュヒット率。	
pgsql.connections	タイプ別の接続数を返します。	
pgsql.custom.query	カスタムクエリの結果を返します。	
pgsql.db.age	データベース内で最も古い FrozenXID の経過期間。	
pgsql.db.bloating_tables	データベースごとの肥大化テーブル数。	
pgsql.db.discovery	PostgreSQL データベースの一覧。	
pgsql.db.size	データベースサイズ（バイト単位）。	
pgsql.dbstat	データベースごとの統計を収集します。	
pgsql.dbstat.sum	クラスター内のすべてのデータベースについて集計したデータ。	
pgsql.locks	データベースごとの許可済みロックに関する情報。	
pgsql.oldest.xid	最も古い XID の経過期間。	
pgsql.ping	接続が有効かどうかをテストします。	
pgsql.queries	実行時間別のクエリメトリクス。	
pgsql.replication.count	スタンバイサーバー数。	
pgsql.replication.process	各送信プロセスごとの flush lag、write lag、replay lag。	
pgsql.replication.process.discovery	レプリケーションプロセス名のディスカバリ。	
pgsql.replication.recovery_role	リカバリステータス。	
pgsql.replication.status	レプリケーションのステータス。	
pgsql.replication_lag.b	レプリケーション遅延（バイト単位）。	
pgsql.replication_lag.sec	レプリケーション遅延（秒単位）。	
pgsql.uptime	PostgreSQL の稼働時間（ミリ秒単位）。	
pgsql.version	PostgreSQL のバージョンを返します。	
pgsql.wal.stat	WAL 統計。	
redis.config	パターンに一致する Redis インスタンスの設定パラメータを取得します。	Redis
redis.info	INFO コマンドの出力を取得します。	
redis.ping	接続が有効かどうかをテストします。	
redis.slowlog.count	Redis の起動以降のスローログエントリ数。	
smart.attribute.discovery	S.M.A.R.T. デバイス属性の一覧を返します。	S.M.A.R.T.
smart.disk.discovery	S.M.A.R.T. デバイスの一覧を返します。	
smart.disk.get	S.M.A.R.T. デバイスの利用可能なすべてのプロパティを返します。	
systemd.unit.get	systemd ユニットのすべてのプロパティを返します。	Systemd
systemd.unit.info	systemd ユニット情報。	
systemd.unit.discovery	systemd ユニットの一覧とその詳細。	
web.certificate.get	証明書を検証し、証明書の詳細を返します。	Web certificates

関連項目:

- Built-in plugins
- Loadable plugins

アイテムキーの詳細

山かっこなしのパラメータは必須です。
山かっこ **< >** で示されたパラメータは任意です。

ceph.df.details[connString,<user>,<apikey>]

 クラスターのデータ使用量とプール間の分布。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

ceph.osd.stats[connString,<user>,<apikey>]

 集約および OSD ごとの統計情報。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

ceph.osd.discovery[connString,<user>,<apikey>]

 検出された OSD の一覧です。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

ceph.osd.dump[connString,<user>,<apikey>]

 OSD の使用率しきい値とステータス。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

ceph.ping[connString,<user>,<apikey>]

 Ceph への接続を確立できるかどうかをテストします。
 戻り値: *0* - 接続が切断されています（AUTH や設定の問題を含む、何らかのエラーが発生した場合）; *1* - 接続に成功しています。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

ceph.pool.discovery[connString,<user>,<apikey>]

 検出されたプールの一覧です。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

ceph.status[connString,<user>,<apikey>]

 クラスター全体のステータス。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, apikey** - Ceph への接続に使用するユーザー名と API キー。

docker.container_info[<ID>,<info>]

 コンテナに関する低レベル情報です。
 戻り値: ContainerInspect API 呼び出しの出力を JSON としてシリアライズしたもの。

パラメータ:

- **ID** - コンテナの ID または名前。
- **info** - 返される情報量。サポートされる値: *short*（デフォルト）または *full*。

十分な権限を得るために、Agent 2 のユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.container_stats[<ID>]

 コンテナのリソース使用状況の統計情報です。
 戻り値: ContainerStats API 呼び出しの出力と CPU 使用率を JSON としてシリアライズしたもの。

パラメータ:

- **ID** - コンテナの ID または名前。

十分な権限を得るには、Agent 2 ユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.containers

 コンテナの一覧。
 戻り値: ContainerList API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るには、Agent 2 のユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.containers.discovery[<options>]

 コンテナの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **options** - すべてのコンテナを検出するか、実行中のコンテナのみを検出するかを指定します。サポートされる値: *true* - すべてのコンテナを返す; *false* - 実行中のコンテナのみを返す（デフォルト）。

十分な権限を得るために、Agent 2 のユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.data.usage

 現在のデータ使用量に関する情報。
 戻り値: SystemDataUsage API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.images

 イメージの一覧を返します。
 戻り値: ImageList API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るには、Agent 2 のユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.images.discovery

 イメージの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

十分な権限を得るには、Agent 2 のユーザー（'zabbix'）を'docker' group に追加する必要があります。そうしないと、このチェックは失敗します。

docker.info

 システム情報。
 戻り値: SystemInfo API 呼び出しの出力を JSON としてシリアライズしたもの。

十分な権限を得るために、Agent 2 のユーザー（'zabbix'）を'docker' groupに追加する必要があります。そうしないと、このチェックは失敗します。

docker.ping

 Docker デーモンが稼働しているかどうかをテストします。
 戻り値: *1* - 接続は有効です。*0* - 接続は切断されています。

十分な権限を得るために、Agent 2 のユーザー（'zabbix'）を'docker' グループ に追加する必要があります。そうしないと、このチェックは失敗します。

ember.get[<uri>,<path>]

 必要なデバイスの結果を返します。
 戻り値: *JSON object*。

パラメータ:

- **uri** - Ember+ デバイスの URI。デフォルト: 127.0.0.1:9998;
- **path** - デバイスへの OID パス。デフォルトでは空で、ルートコレクションのデータを返します。

memcached.ping[connString,<user>,<password>]

 接続が有効かどうかをテストします。
 戻り値: *1* - 接続は有効です。*0* - 接続は切断されています（AUTH や設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - Memcached のログイン認証情報。

memcached.stats[connString,<user>,<password>,<type>]

 STATS コマンドの出力を取得します。
 戻り値: *JSON* - 出力は JSON としてシリアライズされます。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - Memcached のログイン認証情報。
- **type** - 返される統計の種類: *items*、*sizes*、*slabs*、または *settings*（デフォルトでは空で、一般的な統計を返します）。

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 指定されたコレクションについて、さまざまなストレージ統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。
- **database** - データベース名（デフォルト: admin）。
- **collection** - コレクション名。

mongodb.collections.discovery[connString,<user>,<password>]

 検出されたコレクションの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.collections.usage[connString,<user>,<password>]

 コレクションの使用状況の統計を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.connpool.stats[connString,<user>,<password>]

 現在のデータベースインスタンスから、シャーディングクラスタまたはレプリカセットの他のメンバーへの開いている送信接続に関する情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.db.stats[connString,<user>,<password>,<database>]

 指定されたデータベースシステムの状態を反映する統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。
- **database** - データベース名（デフォルト: admin）。

mongodb.db.discovery[connString,<user>,<password>]

 検出されたデータベースの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 jumbo チャンクの数を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.oplog.stats[connString,<user>,<password>]

 oplog から取得したデータを使用して、レプリカセットのステータスを返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.ping[connString,<user>,<password>]

 接続が有効かどうかをテストします。
 戻り値: *1* - 接続は有効です。*0* - 接続は切断されています（AUTH や設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.rs.config[connString,<user>,<password>]

 レプリカセットの現在の設定を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.rs.status[connString,<user>,<password>]

 メソッドが実行されたメンバーの視点から、レプリカセットのステータスを返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.server.status[connString,<user>,<password>]

 データベースの状態を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.sh.discovery[connString,<user>,<password>]

 クラスター内に存在する検出済みシャードの一覧を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mongodb.version[connString,<user>,<password>]

 データベースサーバーのバージョンを返します。
 戻り値: *String*。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MongoDB のログイン認証情報。

mqtt.get[<broker url>,topic,<user>,<password>]

 指定されたブローカーの特定のトピック、またはトピック群（ワイルドカード使用）を購読し、パブリケーションを待機します。
 戻り値: トピックの内容に応じます。ワイルドカードを使用した場合、トピックの内容を JSON として返します。

パラメータ:

- **broker url** - クエリパラメータを含まない `protocol://host:port` 形式の MQTT ブローカー URL（サポートされるプロトコル: `tcp`、`ssl`、`ws`）。値が指定されていない場合、エージェントは `tcp://localhost:1883` を使用します。プロトコルまたはポートが省略された場合は、デフォルトのプロトコル（`tcp`）またはポート（1883）が使用されます。
- **topic** - MQTT トピック（必須）。ワイルドカード（+、#）をサポートしています。
- **user, password** - 認証資格情報（必要な場合）。

コメント:

- このアイテムはactive check（「Zabbix エージェント（アクティブ）」アイテムタイプ）として設定する必要があります。
- TLS 暗号化証明書は、デフォルトの場所（例: Ubuntu では `/etc/ssl/certs/` ディレクトリ）に保存することで使用できます。TLS には、`tls://` スキームを使用してください。

mssql.availability.group.get[URI,<user>,<password>]

 可用性グループを返します。
 戻り値: *JSON object*。

パラメータ:

- **URI** - MSSQL サーバーの URI（サポートされるスキーマは `sqlserver://` のみです）。埋め込み認証情報は無視されます。URI の一部としてインスタンス名を指定できます。例: `sqlserver://localhost/InstanceName`（ポート指定なし）。ポートが指定されている場合、インスタンス名は無視されます。
- **user, password** - 保護された MSSQL サーバーに送信するユーザー名、パスワード。

詳細については、MSSQL plugin の readme を参照してください。

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON object*。

パラメータ:

- **URI** - MSSQL サーバーの URI（サポートされるスキーマは `sqlserver://` のみです）。埋め込み認証情報は無視されます。URI の一部としてインスタンス名を指定できます。例: `sqlserver://localhost/InstanceName`（ポート指定なし）。ポートが指定されている場合、インスタンス名は無視されます。
- **user, password** - 保護された MSSQL サーバーに送信するユーザー名、パスワード。
- **queryName** - `Plugins.MSSQL.CustomQueriesDir` で設定されたカスタムクエリの名前（.sql 拡張子を除く）。
- **args** - クエリに渡す、カンマ区切りの 1 つまたは複数の引数。

詳細については、MSSQL plugin の readme を参照してください。

mssql.db.get

 利用可能なすべての MSSQL データベースを返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL プラグインの readme を参照してください。

mssql.job.status.get

 ジョブのステータスを返します。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.last.backup.get

 すべてのデータベースの最終バックアップ時刻を返します。
 戻り値: *JSON* オブジェクト。

詳細については、MSSQL plugin の readme を参照してください。

mssql.local.db.get

 Always On 可用性グループおよびレプリカ（プライマリまたはセカンダリ）に参加しており、接続先のサーバー上に存在するデータベースを返します。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.mirroring.get

 ミラーリング情報を返します。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.nonlocal.db.get

 Always On 可用性グループに参加しており、かつ他のサーバー上にあるレプリカ（プライマリまたはセカンダリ）に属するデータベースを返します（このデータベースは、接続先として確立された SQL Server インスタンス上のローカルデータベースではありません）。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.perfcounter.get

 パフォーマンスカウンターを返します。
 戻り値: *JSON object*。

詳細については、MSSQL プラグイン の readme を参照してください。

mssql.ping

 データベースに ping を実行します。接続が正しく設定されているかをテストします。
 戻り値: *1* - 稼働中、*0* - 稼働していない。

詳細については、MSSQL plugin の readme を参照してください。

mssql.quorum.get

 クォーラム情報を返します。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.quorum.member.get

 クォーラムメンバーを返します。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.replica.get

 レプリカを返します。
 戻り値: *JSON object*。

詳細については、MSSQL plugin の readme を参照してください。

mssql.version

 MSSQL のバージョンを返します。
 戻り値: *String*。

詳細については、MSSQL plugin の readme を参照してください。

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - URI またはセッション名;
- **user, password** - MySQL のログイン認証情報;
- **queryName** - カスタムクエリの名前。拡張子を除いた SQL ファイル名と一致している必要があります;
- **args** - クエリに渡す、カンマ区切りの 1 つまたは複数の引数。

詳細については、MySQL plugin の readme の *Custom queries* セクションを参照してください。

mysql.db.discovery[connString,<user>,<password>]

 MySQL データベースの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: 「show databases」SQL クエリの結果を LLD JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MySQL のログイン認証情報。

mysql.db.size[connString,<user>,<password>,<database name>]

 データベースのサイズ（バイト単位）。
 戻り値: 特定のデータベースに対する”select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?” SQL クエリの結果（バイト単位）。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MySQL のログイン認証情報。
- **database name** - データベース名。

mysql.get_status_variables[connString,<user>,<password>]

 グローバルステータス変数の値。
 戻り値: ”show global status” SQL クエリの結果（JSON 形式）。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MySQL のログイン認証情報。

mysql.ping[connString,<user>,<password>]

 接続が有効かどうかをテストします。
 戻り値: *1* - 接続は有効です。*0* - 接続は切断されています（認証エラーや設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **connString** - URI またはセッション名。

- **user, password** - MySQL のログイン認証情報。

mysql.replication.discovery[connString,<user>,<password>]

 MySQL レプリケーションの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: SHOW SLAVE STATUS または SHOW REPLICA STATUS SQL クエリの結果を、LLD JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MySQL のログイン認証情報。

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 レプリケーションのステータス。
 戻り値: SHOW SLAVE STATUS または SHOW REPLICA STATUS SQL クエリの結果を JSON 形式で返します。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MySQL のログイン認証情報。
- **master host** - レプリケーション元のマスターホスト名。一致するものが見つからない場合は、エラーが返されます。このパラメータが指定されていない場合は、すべてのホストが返されます。

mysql.version[connString,<user>,<password>]

 MySQL のバージョン。
 戻り値: *String*（MySQL インスタンスのバージョン）。

パラメータ:

- **connString** - URI またはセッション名。
- **user, password** - MySQL のログイン認証情報。

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

DNS クエリを実行し、詳細な DNS レコード情報を返します。
 このアイテムは、対応するレコードタイプを増やし、フラグをカスタマイズ可能にしたnet.dns.record Zabbix エージェントアイテムの拡張版です。
 戻り値: *JSON object*

パラメータ:

- **ip** - DNS サーバーの IP アドレス（デフォルトの DNS サーバーを使用する場合は空のままにします）;
- **name** - 問い合わせる DNS 名;
- **type** - 問い合わせるレコードタイプ（デフォルトは *SOA*）;
- **timeout** - リクエストのタイムアウト（秒単位、デフォルトは 1 秒）;
- **count** - リクエストの試行回数（デフォルトは 2）;
- **protocol** - DNS クエリの実行に使用するプロトコル: *udp*（デフォルト）または *tcp*;
- **flags** - クエリに渡す、カンマ区切りの 1 つ以上の引数。

コメント:

- type に指定可能な値は次のとおりです: *A*, *NS*, *MD*, *MF*, *CNAME*, *SOA*, *MB*, *MG*, *MR*, *NULL*, *PTR*, *HINFO*, *MINFO*, *MX*, *TXT*, *RP*, *AFSDB*, *X25*, *ISDN*, *RT*, *NSAPPTR*, *SIG*, *KEY*, *PX*, *GPOS*, *AAAA*, *LOC*, *NXT*, *EID*, *NIMLOC*, *SRV*, *ATMA*, *NAPTR*, *KX*, *CERT*, *DNAME*, *OPT*, *APL*, *DS*, *SSHFP*, *IPSECKEY*, *RRSIG*, *NSEC*, *DNSKEY*, *DHCID*, *NSEC3*, *NSEC3PARAM*, *TLSA*, *SMIMEA*, *HIP*, *NINFO*, *RKEY*, *TALINK*, *CDS*, *CDNSKEY*, *OPENPGPKEY*, *CSYNC*, *ZONEMD*, *SVCB*, *HTTPS*, *SPF*, *UINFO*, *UID*, *GID*, *UNSPEC*, *NID*, *L32*, *L64*, *LP*, *EUI48*, *EUI64*, *URI*, *CAA*, *AVC*, *AMTRELAY*。値は大文字のみで指定する必要があることに注意してください。小文字または大文字小文字混在の値はサポートされていません。
- 逆引き DNS ルックアップ（type を *PTR* に設定した場合）では、DNS 名を逆順形式と非逆順形式の両方で指定できます（以下の例を参照）。PTR レコードを要求する場合、DNS 名は実際には IP アドレスであることに注意してください。
- flags に指定可能な値は次のとおりです: *cdflag* または *nocdflag*（デフォルト）、*rdflag*（デフォルト）または *nordflag*、*dnssec* または *nodnssec*（デフォルト）、*nsid* または *nonsid*（デフォルト）、*edns0*（デフォルト）または *noedns0*、*aaflag* または *noaaflag*（デフォルト）、*adflag* または *noadflag*（デフォルト）。flags の *dnssec* および *nsid* は、どちらも *edns0* を必要とするため、*noedns0* と一緒に使用できません。値は小文字のみで指定する必要があることに注意してください。大文字または大文字小文字混在の値はサポートされていません。
- 国際化ドメイン名はサポートされていません。代わりに IDNA エンコードされた名前を使用してください。
- 出力は、指定されたパラメータに基づく DNS レコード情報を含むオブジェクトです（詳細はこちらを参照）。

例:

```
net.dns.get[192.0.2.0,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]
```

```
net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 GPU デバイスの数。
 戻り値: 整数。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのデコーダー使用率（パーセンテージ）。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの ECC モード情報（現在、保留中）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのエンコーダー統計情報。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのエンコーダー使用率（パーセンテージ）。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 ドライバーが最後に再読み込みされて以降の、GPU デバイスの総エネルギー消費量（ミリジュール（mJ））。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの ECC メモリエラー情報（訂正済み、未訂正）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの ECC レジスターエラー情報（訂正済み、未訂正）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの平均ファン速度を最大速度に対する割合（パーセンテージ）で示します。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

nvml.device.get

 UUID と名前を持つ検出済み GPU。
 戻り値: *JSON*。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのグラフィックスクロック速度（MHz）。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのフレームバッファメモリ統計情報（合計、予約済み、空き、使用中）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの BAR1 メモリ統計情報（合計、空き、使用量）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのメモリクロック速度（MHz）。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの PCI 使用率（送信/受信スループット、KBps）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのパフォーマンス状態。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの電力上限（ミリワット単位）。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスの現在の消費電力（ミリワット単位）。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

nvml.device.serial[<deviceUUID>]

 GPU デバイスのシリアル番号。
 戻り値: *String*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのストリーミングマルチプロセッサのクロック速度（MHz）。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU プラグインの README を参照してください。

nvml.device.temperature[<deviceUUID>]

 GPU デバイスの温度（摂氏）。
 戻り値: *Integer*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

nvml.device.utilization[<deviceUUID>]

 GPU デバイスの使用率統計（GPU/メモリ使用率をパーセンテージで表示）。
 戻り値: *JSON*。

パラメータ:

- **deviceUUID** - GPU デバイスの UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

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

 GPU デバイスのビデオクロック速度（MHz）。
 戻り値: 整数。

パラメータ:

- **deviceUUID** - GPU デバイス UUID。

詳細については、NVIDIA GPU plugin の readme を参照してください。

nvml.system.driver.version

 NVIDIA ドライバのバージョン。
 戻り値: *String*。

詳細については、NVIDIA GPU plugin の readme を参照してください。

nvml.version

 NVML ライブラリのバージョン。
 戻り値: *String*。

詳細については、NVIDIA GPU plugin の readme を参照してください。

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Automatic Storage Management (ASM) ディスクグループの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:

    - URI - 指定された名前のセッションが見つからず、かつ Plugins.Oracle.ResolveTNS が false に設定されている場合。
      スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 "(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、ログインオプションとして as sysdba、as sysoper、または as sysasm のみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **diskgroup** - 問い合わせ対象の ASM ディスクグループ名。

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 ASM ディスクグループの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、かつ Plugins.Oracle.ResolveTNS が false に設定されている場合。
      スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 "(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを付加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、ログインオプションとして as sysdba、as sysoper、または as sysasm のみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

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

 アーカイブログの統計情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。
      スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 "(" で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを付加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、as sysdba、as sysoper、または as sysasm のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **destination** - 問い合わせ対象の宛先名。

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

 コンテナデータベース（CDB）の情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、ログインオプションとして `as sysdba`、`as sysoper`、`as sysasm` のみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **database** - 問い合わせ対象のデータベース名。

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 カスタムクエリの結果です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを `user as sysdba` の形式で末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
  Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、`as sysasm` のログインオプションのみがサポートされる点に注意してください。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。
- **queryName** - カスタムクエリ名。拡張子を除いた SQL ファイル名と一致している必要があります。
- **args** - クエリに渡す、カンマ区切りの 1 つまたは複数の引数。

コメント: - 返されたデータは自動的に JSON に変換されます。- クエリから直接 JSON を返すことは避けてください。プラグインが再度 JSON に変換しようとすると、データが破損します。

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 データファイルの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、または `as sysasm` のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

oracle.db.discovery[connString,<user>,<password>,<service>]

 データベースの一覧を返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - プラグイン設定ファイル内にその名前が存在する場合。
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、上記のいずれの条件にも該当しない場合。
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、または `as sysasm` のログインオプションのみがサポートされる点に注意してください。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。

oracle.fra.stats[connString,<user>,<password>,<service>]

 Fast Recovery Area（FRA）の統計情報を返します。
 戻り値: *JSON object*。

パラメータ: - **connString** - 次のいずれかを指定できます:
 - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
 - セッション名 - プラグイン設定ファイル内でその名前が見つかった場合;
 - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
 - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
 - **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字と小文字は区別されず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、または `as sysasm` のログインオプションのみがサポートされる点に注意してください;
 - **password** - Oracle のパスワード;
 - **service** - Oracle のサービス名。

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

 インスタンスの統計情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、または `as sysasm` のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

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

 プラガブル・データベース（PDB）の情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、かつ `Plugins.Oracle.ResolveTNS` が false に設定されている場合。
    スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
  - セッション名 - その名前がプラグイン設定ファイル内に存在する場合。

    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションでは大文字・小文字は区別されず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、`as sysasm` のログインオプションのみがサポートされる点に注意してください。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。
- **database** - 問い合わせ対象のデータベース名。

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 PDB のリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内でその名前が見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、`as sysasm` のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

oracle.pga.stats[connString,<user>,<password>,<service>]

 Program Global Area (PGA) の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます。
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合。
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合。
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、または `as sysasm` のログインオプションのみがサポートされる点に注意してください。
- **password** - Oracle のパスワード。
- **service** - Oracle のサービス名。

oracle.ping[connString,<user>,<password>,<service>]

 Oracle への接続を確立できるかどうかをテストします。
 戻り値: *1* - 接続成功; *0* - 接続切断（AUTH や設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内にその名前が存在する場合;

    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値 (Zabbix 7.4.3 以降でサポート) - 開き括弧 "(" で始まる場合 (先頭の空白は無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、ログインオプションとして `as sysdba`、`as sysoper`、または `as sysasm` のみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

例:

```
oracle.ping[tcp://127.0.0.1:1521,ZABBIX_MON,zabbix,xe]
oracle.ping[localhost,ZABBIX_MON,zabbix,xe]
oracle.ping[zbx_tns_example,ZABBIX_MON,zabbix,xe]
oracle.ping["(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=x
```

oracle.proc.stats[connString,<user>,<password>,<service>]

 プロセスの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内でその名前が見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値 (Zabbix 7.4.3 以降でサポート) - 開き括弧 "(" で始まる場合 (先頭の空白は無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、または `as sysasm` のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

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

 制御ファイルから取得したログファイル情報です。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、上記のいずれの条件にも該当しない場合;
    - TNS 値 (Zabbix 7.4.3 以降でサポート) - 開き括弧 "(" で始まる場合 (先頭の空白は無視されます)。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを付加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、`as sysasm` のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

oracle.sga.stats[connString,<user>,<password>,<service>]

 System Global Area (SGA) の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:

    - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
Zabbix 7.4.2 より前では、ログインオプションとして as sysdba、as sysoper、as sysasm のみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 セッションの統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
Zabbix 7.4.2 より前では、as sysdba、as sysoper、as sysasm のログインオプションのみサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **lockMaxTime** - セッションを長時間ロック状態とみなしてカウントするための、セッションロックの最大継続時間（秒）。デフォルト: 600 秒。

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 システムメトリクス値のセットを返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
    - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
    - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
    - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、上記のいずれの条件にも該当しない場合;
    - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。なお、Zabbix 7.4.2 より前では、as sysdba、as sysoper、as sysasm のログインオプションのみがサポートされています;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **duration** - システムメトリクス値の取得間隔（秒）。指定可能な値: *60* — 長い間隔（デフォルト）、*15* — 短い間隔。

oracle.sys.params[connString,<user>,<password>,<service>]

 システムパラメータ値のセットを返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、ログインオプションとして `as sysdba`、`as sysoper`、または `as sysasm` のみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

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

 表領域の統計情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、常に URI として扱われます;
  - セッション名 - プラグイン設定ファイル内でその名前が見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾に空白を含めてはいけません）。
  Zabbix 7.4.2 より前では、`as sysdba`、`as sysoper`、`as sysasm` のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名;
- **tablespace** - 問い合わせる表領域の名前。
  デフォルト（空のままで `type` が設定されている場合）:
  - ”TEMP”（`type` が”TEMPORARY” に設定されている場合）;
  - ”USERS”（`type` が”PERMANENT” に設定されている場合）。
- **type** - 問い合わせる表領域のタイプ。
  デフォルト（`tablespace` が設定されている場合）: ”PERMANENT”。
- **conname** - 情報が必要なコンテナの名前。

`tablespace`、`type`、または `conname` が省略された場合、このアイテムは一致するすべてのコンテナ（PDB および CDB を含む）の表領域統計情報を返します。

oracle.ts.discovery[connString,<user>,<password>,<service>]

 表領域のリストを返します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、`Plugins.Oracle.ResolveTNS` が false に設定されている場合。スキーマ（例: ”tcp://”）、ポート（例: 1521）、またはその両方を含む場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - プラグイン設定ファイル内にその名前が見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 “(” で始まる場合（先頭の空白は無視されます）。この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle のユーザー名。`user as sysdba` の形式で、ログインオプション `as sysdba`、`as sysoper`、`as sysasm`、`as sysbackup`、`as sysdg`、`as syskm`、または `as sysrac` のいずれかを末尾に追加できます（ログインオプションは大

文字・小文字を区別せず、末尾にスペースを含めてはいけません）。Zabbix 7.4.2 より前では、as sysdba、as sysoper、as sysasm のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle のパスワード;
- **service** - Oracle のサービス名。

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

 Oracle ユーザー情報を返します。
 戻り値: *JSON object*。

パラメータ:

- **connString** - 以下のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。
    スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 "(" で始まる場合（先頭の空白は無視されます）。
    この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを末尾に追加できます（ログインオプションは大文字・小文字を区別せず、末尾にスペースを含めてはいけません）。
  Zabbix 7.4.2 より前では、as sysdba、as sysoper、as sysasm のログインオプションのみサポートされる点に注意してください;
- **password** - Oracle パスワード;
- **service** - Oracle サービス名;
- **username** - 情報が必要なユーザー名。
  小文字のユーザー名はサポートされていません。
  デフォルト: 現在のユーザー。

oracle.version[connString,<user>,<password>,<service>]

 データベースサーバーのバージョンを返します。
 戻り値: *String*。

パラメータ:

- **connString** - 次のいずれかを指定できます:
  - URI - 指定された名前のセッションが見つからず、Plugins.Oracle.ResolveTNS が false に設定されている場合。
    スキーマ（例: "tcp://"）、ポート（例: 1521）、またはその両方が含まれている場合、ResolveTNS オプションは考慮されず、いずれにしても URI として扱われます;
  - セッション名 - その名前がプラグイン設定ファイル内で見つかった場合;
  - TNS キー（Zabbix 7.4.3 以降でサポート）- ResolveTNS オプションが true に設定されており、かつ上記のいずれの条件にも該当しない場合;
  - TNS 値（Zabbix 7.4.3 以降でサポート）- 開き括弧 "(" で始まる場合（先頭の空白は無視されます）。
    この場合、ResolveTNS オプションは考慮されません。
- **user** - Oracle ユーザー名。user as sysdba の形式で、ログインオプション as sysdba、as sysoper、as sysasm、as sysbackup、as sysdg、as syskm、または as sysrac のいずれかを末尾に追加できます（ログインオプションは大文字小文字を区別せず、末尾にスペースを含めてはいけません）。
  Zabbix 7.4.2 より前では、as sysdba、as sysoper、as sysasm のログインオプションのみがサポートされる点に注意してください;
- **password** - Oracle パスワード;
- **service** - Oracle サービス名。

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

 autovacuum ワーカーの数。
 戻り値: *Integer*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 アーカイブされたファイルに関する情報。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 データベースクラスタのチェックポイント総数を、チェックポイントの種類ごとに分類して返します。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 PostgreSQL バッファキャッシュのヒット率。
 戻り値: *Float*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 タイプ別の接続数を返します。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

pgsql.custom.query[uri,<username>,<password>,queryName,<args...>]

 カスタムクエリの結果を返します。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **queryName** - カスタムクエリの名前。拡張子を除いた SQL ファイル名と一致している必要があります。
- **args** - クエリに渡す、カンマ区切りの 1 つまたは複数の引数。

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

 データベースの最も古い FrozenXID の経過期間。
 戻り値: 整数。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合は、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 データベースごとの肥大化したテーブル数。
 戻り値: 整数。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合は、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 PostgreSQL データベースの一覧です。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 データベースのサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 データベースごとの統計情報を収集します。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 クラスター内のすべてのデータベースの集計データ。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 データベースごとに許可されたロックに関する情報です。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 最も古い XID の経過時間。
 戻り値: *Integer*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 接続が有効かどうかをテストします。
 戻り値: *1* - 接続は有効です。*0* - 接続は切断されています（AUTH や設定の問題を含む、何らかのエラーが発生した場合）。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 実行時間ごとのクエリメトリクスを取得します。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。
- **time period** - 遅いクエリ数のカウントに対する実行時間のしきい値（正の整数である必要があります）。

pgsql.replication.count[uri,<username>,<password>]

 スタンバイサーバーの数。
 戻り値: *Integer*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.process[uri,<username>,<password>]

 各送信プロセスごとの flush lag、write lag、replay lag。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.process.discovery[uri,<username>,<password>]

 レプリケーションプロセス名のディスカバリ。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.recovery_role[uri,<username>,<password>]

 リカバリステータス。
 戻り値: *0* - マスターモード; *1* - リカバリがまだ進行中（スタンバイモード）。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。

pgsql.replication.status[uri,<username>,<password>]

 レプリケーションのステータス。
 戻り値: *0* - ストリーミング停止; *1* - ストリーミング稼働中; *2* - マスターモード。

パラメータ:

- **uri** - URI またはセッション名;
- **username, password** - PostgreSQL の認証情報。

pgsql.replication_lag.b[uri,<username>,<password>]

 バイト単位のレプリケーション遅延。
 戻り値: *Integer*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。

pgsql.replication_lag.sec[uri,<username>,<password>]

 レプリケーション遅延（秒単位）。
 戻り値: *Integer*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。

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

 PostgreSQL の稼働時間（ミリ秒単位）。
 戻り値: *Float*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Databaseの値が使用されます）。

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

 PostgreSQL のバージョンを返します。
 戻り値: 文字列。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

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

 WAL の統計情報。
 戻り値: *JSON object*。

パラメータ:

- **uri** - URI またはセッション名。
- **username, password** - PostgreSQL の認証情報。
- **database name** - データベース名（省略した場合、postgresql.conf のPlugins.PostgreSQL.Default.Database の値が使用されます）。

redis.config[connString,<password>,<pattern>]

 パターンに一致する Redis インスタンスの設定パラメータを取得します。
 戻り値: グロブ形式のパターンが使用された場合は *JSON*、パターンにワイルドカード文字が含まれない場合は単一の値。

パラメータ:

- **connString** - URI またはセッション名。
- **password** - Redis のパスワード。
- **pattern** - グロブ形式のパターン（デフォルトは *）。

コメント:

- セキュリティ警告: `redis.config` は Redis の `CONFIG GET` コマンドを実行するため、機密性の高い設定パラメータ（たとえば `requirepass`、`masterauth`、TLS 関連の設定、その他のシークレット）を返す可能性があります。
  デフォルトのパターン（*）または広範なパターンを使用すると、Zabbix でアイテムの値を表示する権限を持つすべてのユーザーに、これらのシークレットが公開される可能性があります。

redis.info[connString,<password>,<section>]

 INFO コマンドの出力を取得します。
 戻り値: *JSON* - 出力は JSON としてシリアライズされます。

パラメータ:

- **connString** - URI またはセッション名。
- **password** - Redis のパスワード。
- **section** - 情報のセクション（デフォルトでは *default*）。

redis.ping[connString,<password>]

 接続が有効かどうかをテストします。
 戻り値: *1* - 接続は有効です。*0* - 接続は切断されています（AUTH や設定の問題を含む、何らかのエラーが発生している場合）。

パラメータ:

- **connString** - URI またはセッション名。
- **password** - Redis のパスワード。

redis.slowlog.count[connString,<password>]

 Redis の起動以降のスローログエントリ数。
 戻り値: *Integer*。

パラメータ:

- **connString** - URI またはセッション名。
- **password** - Redis のパスワード。

smart.attribute.discovery

 S.M.A.R.T. デバイス属性の一覧を返します。
 戻り値: *JSON object*。

コメント:

- 次のマクロとその値が返されます: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH};
- HDD、SSD、NVME のドライブタイプがサポートされています。ドライブは単体でも、RAID に組み込まれていてもかまいません。RAID の場合、{#NAME} には追加情報が付加されます。例: {"{#NAME}": "/dev/sda cciss,2"}。

smart.disk.discovery[<type>]

 S.M.A.R.T. デバイスのリストを返します。
 戻り値: *JSON object*。

パラメータ:

- **type** - ディスクのスキャン対象の値を指定します。指定可能な値: *id*、*name*（デフォルト）。*id* は Windows ではサポートされておらず、選択するとエラーを返します

コメント:

- 次のマクロとその値が返されます: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE};
- HDD、SSD、NVME のドライブタイプがサポートされています。ドライブが RAID に属していない場合、{#RAIDTYPE} は空になります。RAID の場合、{#NAME} には追加情報が付加されます。例: {"{#NAME}": "/dev/sda cciss,2"}。

smart.disk.get[<path>,<raid type>]

 利用可能な S.M.A.R.T. デバイスのすべてのプロパティを返します。
 戻り値: *JSON object*。

パラメータ:

- **path** - ディスクのパス。値として {#PATH} マクロを使用できます。
- **raid_type** - RAID タイプ。値として {#RAID} マクロを使用できます

コメント:

- HDD、SSD、および NVME のドライブタイプをサポートしています。ドライブは単体でも、RAID として構成されていてもかまいません。
- データには smartctl のバージョンと呼び出し引数、および追加のフィールドが含まれます:
disk_name - RAID ディスカバリに必要な追加情報を含む名前を保持します。例: {"disk_name": "/dev/sda cciss,2"}
disk_type - ディスクタイプ HDD、SSD、または NVME を保持します。例: {"disk_type": "ssd"};
- パラメータが指定されていない場合、このアイテムはすべてのディスクに関する情報を返します。

systemd.unit.get[ユニット名,<interface>]

 systemd ユニットのすべてのプロパティを返します。
 戻り値: *JSON object*。

パラメータ:

- ユニット名 - ユニット名（名前をディスカバリするには、アイテムプロトタイプで {#UNIT.NAME} マクロを使用すると便利です）;
- **interface** - ユニットのインターフェースタイプ。指定可能な値: *Unit*（デフォルト）、*Service*、*Socket*、*Device*、*Mount*、*Automount*、*Swap*、*Target*、*Path*。

コメント:

- このアイテムは Linux プラットフォームでのみサポートされています;
- Unit インターフェースの LoadState､ActiveState､UnitFileState は､テキストと整数の両方で返されます: `"ActiveState":{"state":1,"`

systemd.unit.info[unit name,<property>,<interface>]

 systemd ユニットの情報です。
 戻り値: *String*。

パラメータ:

- **unit name** - ユニット名（名前のディスカバリーには、アイテムプロトタイプで {#UNIT.NAME} マクロを使用できます）;
- **property** - ユニットのプロパティ（例: ActiveState（デフォルト）、LoadState、Description）;
- **interface** - ユニットのインターフェースタイプ（例: Unit（デフォルト）、Socket、Service）。

コメント:

- このアイテムは Linux プラットフォームでのみサポートされています;
- このアイテムでは、dbus API で説明されているように、特定のインターフェースタイプから特定のプロパティを取得できます。

例:

```
systemd.unit.info["{#UNIT.NAME}"] #collect active state (active, reloading, inactive, failed, activating,
systemd.unit.info["{#UNIT.NAME}",LoadState] #collect load state info on discovered systemd units
systemd.unit.info[mysqld.service,Id] #retrieve the service technical name (mysqld.service)
systemd.unit.info[mysqld.service,Description] #retrieve the service description (MySQL Server)
```

```
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #retrieve the last time the service entered the act
systemd.unit.info[dbus.socket,NConnections,Socket] #collect the number of connections from this socket uni
```

systemd.unit.discovery[<type>]

 systemd ユニットの一覧とその詳細です。ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **type** - 指定可能な値: *all*, *automount*, *device*, *mount*, *path*, *service*（デフォルト）, *socket*, *swap*, *target*。

このアイテムは Linux プラットフォームでのみサポートされています。

web.certificate.get[hostname,<port>,<address>]

 証明書を検証し、証明書の詳細を返します。
 戻り値: *JSON object*。

パラメータ:

- **hostname** - IP または DNS のいずれかを指定できます。
URL スキーム（*https* のみ）、パス（無視されます）、およびポートを含めることができます。
 第 1 パラメータと第 2 パラメータの両方でポートが指定されている場合、それらの値は一致している必要があります。
address（第 3 パラメータ）が指定されている場合、hostname は SNI およびホスト名検証にのみ使用されます。
- **port** - ポート番号（HTTPS のデフォルトは 443）。
- **address** - IP または DNS のいずれかを指定できます。指定した場合、接続にはこれが使用され、hostname（第 1 パラメータ）は SNI およびホスト検証に使用されます。第 1 パラメータが IP で第 3 パラメータが DNS である場合、第 1 パラメータが接続に使用され、第 3 パラメータが SNI およびホスト検証に使用されます。

コメント:

- ホスト設定で指定された宛先が存在しない、利用できない、または無効な証明書以外の何らかのエラーで TLS ハンドシェイクが失敗した場合、このアイテムは未サポートになります。
- 現在、AIA（Authority Information Access）X.509 拡張、CRL、OCSP（OCSP stapling を含む）、および Certificate Transparency はサポートされていません。
- JSON レスポンスフィールド:
  - *x509*: X.509 証明書の詳細を含みます。
    - *version*: X.509 のバージョン（例: 3）。
    - *serial_number*: 証明書のシリアル番号。
    - *signature_algorithm*: 証明書の署名に使用されたアルゴリズム（例: SHA256-RSA）。
    - *issuer*: 証明書の発行者。
    - *not_before*: 証明書の有効期間の開始日。
    - *not_after*: 証明書の有効期限。
    - *subject*: 証明書のサブジェクト。
    - *public_key_algorithm*: 公開鍵に使用されるアルゴリズム（例: RSA）。
    - *alternative_names*: サブジェクト代替名（存在する場合）。存在しない場合は `null`。
  - result: 検証結果を含みます。
    - *value*: 検証ステータス（可能な値は以下を参照）。
    - *message*: 詳細な検証メッセージ（例: `"certificate verified successfully"`）。
  - *sha1_fingerprint*: 証明書の SHA-1 フィンガープリント。
  - *sha256_fingerprint*: 証明書の SHA-256 フィンガープリント。
- `$.result.value` フィールドは証明書の検証結果を示します。可能な値は次のとおりです:
  - *valid* – 証明書は有効で信頼されています。
  - *valid-but-self-signed* – 証明書は有効ですが自己署名証明書です。つまり、サブジェクトが発行者と一致します。
  - *invalid* – 証明書は、期限切れ、不正なホスト名、不明な署名機関などの問題により無効です。

例:

```
web.certificate.get[example.com,443]
```

JSON レスポンス:

```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
      "value": "Jan 15 00:00:00 2025 GMT",
```

```
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


2 Windows Zabbix エージェント

概要

Windows Zabbix エージェントのアイテムは、2 つのリストで示されています。

- 共通アイテム - UNIX Zabbix エージェントと共通のアイテムキー
- Windows 固有のアイテム - Windows でのみサポートされるアイテムキー

Windows 上の Zabbix エージェントでサポートされるすべてのアイテムキーは、新世代の Zabbix エージェント 2 でもサポートされていることに注意してください。エージェント 2 でのみ使用できる追加のアイテムキーも参照してください。

参照: Windows アイテムに必要な最小権限

共通アイテム

以下の表は、Windows でサポートされ、UNIX Zabbix エージェントと共通の Zabbix エージェントのアイテムを示しています。

- アイテムキーは、UNIX Zabbix エージェントのアイテムの詳細説明へのリンクです
- Windows に関連するアイテムのコメントが含まれています

Item key	Description	Item group
log	ログファイルの監視。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	Log monitoring
log.count	監視対象のログファイル内で一致した行数。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	
logrt	ローテーションされるログファイルの監視。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	
logrt.count	ローテーションされる監視対象ログファイル内で一致した行数。このアイテムは Windows Event Log ではサポートされません。 persistent_dir パラメータは Windows ではサポートされません。	
modbus.get	Modbus データを読み取ります。	Modbus
net.dns	DNS サービスが稼働しているかを確認します。 Zabbix agent 2 を使用している場合を除き、ip、timeout、count パラメータは Windows では無視されます。	Network
net.dns.perf	DNS サービスのパフォーマンスを確認します。 Zabbix agent 2 を使用している場合を除き、ip、timeout、count パラメータは Windows では無視されます。	
net.dns.record	DNS クエリを実行します。 Zabbix agent 2 を使用している場合を除き、ip、timeout、count パラメータは Windows では無視されます。	

Item key	Description	Item group
net.if.discovery	ネットワークインターフェースの一覧。 一部の Windows バージョン（例: Server 2008）では、インターフェース名で非 ASCII 文字をサポートするために最新の更新プログラムのインストールが必要な場合があります。	
net.if.in	ネットワークインターフェースの受信トラフィック統計。 Windows では、利用可能な場合、このアイテムは 64 ビットカウンターから値を取得します。64 ビットのインターフェース統計カウンターは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが利用できない場合、エージェントは 32 ビットカウンターを使用します。 Windows ではマルチバイトのインターフェース名がサポートされます。 Windows のネットワークインターフェースの説明は、net.if.discovery または net.if.list アイテムで取得できます。	
net.if.out	ネットワークインターフェースの送信トラフィック統計。 Windows では、利用可能な場合、このアイテムは 64 ビットカウンターから値を取得します。64 ビットのインターフェース統計カウンターは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが利用できない場合、エージェントは 32 ビットカウンターを使用します。 Windows ではマルチバイトのインターフェース名がサポートされます。 Windows のネットワークインターフェースの説明は、net.if.discovery または net.if.list アイテムで取得できます。	
net.if.total	ネットワークインターフェースの受信および送信トラフィック統計の合計。 Windows では、利用可能な場合、このアイテムは 64 ビットカウンターから値を取得します。64 ビットのインターフェース統計カウンターは Windows Vista および Windows Server 2008 で導入されました。64 ビットカウンターが利用できない場合、エージェントは 32 ビットカウンターを使用します。 Windows のネットワークインターフェースの説明は、net.if.discovery または net.if.list アイテムで取得できます。	
net.tcp.listen	この TCP ポートが LISTEN 状態にあるかを確認します。	
net.tcp.port	指定されたポートへの TCP 接続を確立できるかを確認します。	
net.tcp.service	サービスが稼働しており、TCP 接続を受け付けているかを確認します。 Windows での LDAP および HTTPS の確認は、Zabbix agent 2 でのみサポートされます。	
net.tcp.service.perf	TCP サービスのパフォーマンスを確認します。 Windows での LDAP および HTTPS の確認は、Zabbix agent 2 でのみサポートされます。	
net.tcp.socket.count	パラメータに一致する TCP ソケット数を返します。 このアイテムは Linux では Zabbix agent によりサポートされますが、Windows では 64 ビット Windows 上のZabbix agent 2 でのみサポートされます。	
net.udp.service	サービスが稼働しており、UDP リクエストに応答しているかを確認します。	
net.udp.service.perf	UDP サービスのパフォーマンスを確認します。	
net.udp.socket.count	パラメータに一致する UDP ソケット数を返します。 このアイテムは Linux では Zabbix agent によりサポートされますが、Windows では 64 ビット Windows 上のZabbix agent 2 でのみサポートされます。	
proc.get	OS プロセスとそのパラメータの一覧。 `cmdline` パラメータは Windows ではサポートされません。	Processes
proc.num	プロセス数。 Windows では、`name` および `user` パラメータのみサポートされます。	
system.cpu.discovery	検出された CPU/CPU コアの一覧。	System
system.cpu.load	CPU 負荷。 Zabbix エージェントで collector プロセスが開始されると、次のパフォーマンスカウンターが初期化され、その後このアイテムで使用されます: `\System\Processor Queue Length`	
system.cpu.num	CPU 数。	
system.cpu.util	CPU 使用率。 値は *Processor Time* パフォーマンスカウンターを使用して取得されます。Windows 8 以降では、タスクマネージャーは *Processor Utility* パフォーマンスカウンターに基づく CPU 使用率を表示しますが、それ以前のバージョンでは *Processor Time* カウンターに基づいていました（詳細 を参照）。 Windows でサポートされる `type` パラメータは *system* のみです。	

Item key	Description	Item group
system.hostname	システムのホスト名。 Windows では、この値は GetComputerName()（**netbios** の場合）、GetComputerNameExA()（**fqdn** の場合）、または gethostname()（**host** の場合）関数によって取得されます。 あわせてより詳細な説明 も参照してください。	
system.localtime	システム時刻。	
system.run	ホスト上で指定したコマンドを実行します。	
system.sw.arch	ソフトウェアアーキテクチャ情報。	
system.sw.os	オペレーティングシステム情報。	
system.sw.os.get	オペレーティングシステムに関する詳細情報（バージョン、タイプ、ディストリビューション名、マイナーバージョン、メジャーバージョンなど）。	
system.swap.size	スワップ領域のサイズ（バイト単位、または全体に対する割合）。 pused type パラメータは Linux では Zabbix agent によりサポートされますが、Windows ではZabbix agent 2 でのみサポートされます。 このキーは、仮想化された（VMware ESXi、VirtualBox）Windows プラットフォーム上で誤ったスワップ領域サイズ/割合を報告する場合があることに注意してください。この場合、正しいスワップ領域の割合を取得するために `perf_counter[\700(_Total)\702]` キーを使用できます。	
system.uname	システムの識別情報。 Windows では、このアイテムの値は Win32_OperatingSystem および Win32_Processor WMI クラスから取得されます。OS 名（エディションを含む）は、ユーザーの表示言語に翻訳される場合があります。一部の Windows バージョンでは、商標記号や余分なスペースが含まれることがあります。	
system.uptime	システムの稼働時間（秒）。	
vfs.dir.count	ディレクトリエントリ数。 Windows では、ディレクトリのシンボリックリンクはスキップされ、ハードリンクは 1 回だけカウントされます。	Virtual file systems
vfs.dir.get	ディレクトリエントリの一覧。 Windows では、ディレクトリのシンボリックリンクはスキップされ、ハードリンクは 1 回だけカウントされます。	
vfs.dir.size	ディレクトリサイズ。 Windows では、あらゆるシンボリックリンクがスキップされ、ハードリンクは 1 回だけ考慮されます。	
vfs.file.cksum	UNIX の cksum アルゴリズムで計算されたファイルチェックサム。	
vfs.file.contents	ファイルの内容を取得します。	
vfs.file.exists	ファイルが存在するかを確認します。 Windows では、zabbix_get.exe または agent2 を呼び出すコマンドラインユーティリティを使用する場合、二重引用符はバックスラッシュ'\' でエスケープし、アイテムキー全体を二重引用符で囲む必要があります。 Windows では、存在しないディレクトリ内を検索対象にした場合、このアイテムが unsupported になることがある点に注意してください。例: `vfs.file.exists[C:\no\dir,dir]`（この場合'no' は存在しません）。	
vfs.file.get	ファイルに関する情報を返します。 Windows でサポートされるファイルタイプ: 通常ファイル、ディレクトリ、シンボリックリンク	
vfs.file.md5sum	ファイルの MD5 チェックサム。	
vfs.file.owner	ファイルの所有者を取得します。	
vfs.file.regexp	ファイル内の文字列を取得します。	
vfs.file.regmatch	ファイル内の文字列を検索します。	
vfs.file.size	ファイルサイズ。	
vfs.file.time	ファイルの時刻情報。 Windows XP では `vfs.file.time[file,change]` が `vfs.file.time[file,access]` と等しくなる場合があります。	
vfs.fs.discovery	マウントされたファイルシステムの一覧と、そのタイプおよびマウントオプション。 {#FSLABEL} マクロは Windows でサポートされます。	
vfs.fs.get	マウントされたファイルシステムの一覧と、そのタイプ、利用可能なディスク容量、inode 統計、およびマウントオプション。 {#FSLABEL} マクロは Windows でサポートされます。	
vfs.fs.size	ディスク容量（バイト単位、または全体に対する割合）。	
vm.memory.size	メモリサイズ（バイト単位、または全体に対する割合）。	Virtual memory

Item key	Description	Item group
web.page.get	Web ページの内容を取得します。	Web monitoring
web.page.perf	Web ページ全体の読み込み時間。	
web.page.regexp	Web ページ上の文字列を検索します。	
agent.hostmetadata	エージェントのホストメタデータ。	Zabbix
agent.hostname	エージェントのホスト名。	
agent.ping	エージェントの可用性確認。	
agent.variant	Zabbix エージェントのバリアント（Zabbix agent または Zabbix agent 2）。	
agent.version	Zabbix エージェントのバージョン。	
zabbix.stats	リモートで Zabbix サーバーまたはプロキシの内部メトリクス一式を返します。	
zabbix.stats	リモートで、遅延している Zabbix サーバーまたはプロキシ上のキュー内の監視対象アイテム数を返します。	

Windows 固有のアイテム

この表では、Windows Zabbix エージェントでのみサポートされるアイテムキーの詳細を示します。

Windows 固有のアイテムは、類似するエージェントアイテムのおおよその対応物である場合があります。たとえば、Windows でサポートされる `proc_info` は、Windows ではサポートされない `proc.mem` アイテムにおおよそ対応します。

アイテムキーは、各アイテムキーの詳細説明へのリンクになっています。

Item key	Description	Item group
eventlog	Windows イベントログの監視。	ログ監視
eventlog.count	Windows イベントログ内の行数。	
net.if.list	ネットワークインターフェースの一覧（インターフェースタイプ、ステータス、IPv4 アドレス、説明を含む）。	ネットワーク
perf_counter	任意の Windows パフォーマンスカウンターの値。	パフォーマンスカウンター
perf_counter_en	英語での任意の Windows パフォーマンスカウンターの値。	
perf_instance.discovery	Windows パフォーマンスカウンターのオブジェクトインスタンスの一覧。	
perf_instance_en.discovery	英語のオブジェクト名を使用して検出された、Windows パフォーマンスカウンターのオブジェクトインスタンスの一覧。	
proc_info	特定のプロセスに関する各種情報。	プロセス
registry.data	Windows レジストリキー内の指定された値名のデータを返します。	レジストリ
registry.get	指定されたキーにある Windows レジストリの値またはキーの一覧。	
service.discovery	Windows サービスの一覧。	サービス
service.info	サービスに関する情報。	
services	サービスの一覧。	
vm.vmemory.size	仮想メモリサイズ（バイト単位、または合計に対する割合）。	仮想メモリ
wmi.get	WMI クエリを実行し、最初に選択されたオブジェクトを返します。	WMI
wmi.getall	WMI クエリを実行し、応答全体を返します。	

アイテムキーの詳細

山かっこなしのパラメータは必須です。
山かっこ **< >** で示されたパラメータは任意です。

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 イベントログの監視です。
 戻り値: ログ。

パラメータ:

- **name** - イベントログチャネルの名前（イベントビューアー GUI の ログ名）。
- **regexp** - 必要なパターンを記述する正規表現（大文字と小文字を区別）。
- **severity** - 重大度を記述する正規表現（大文字と小文字を区別しない）。このパラメータは、次の値に基づく正規表現を受け付けます: ”Information”、”Warning”、”Error”、”Critical”、”Verbose”（Windows Vista 以降で動作）。
- **source** - ソース識別子を記述する正規表現（大文字と小文字を区別しない）。
- **eventid** - イベント識別子を記述する正規表現（大文字と小文字を区別）。
- **maxlines** - エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行の最大数。このパラメータは、zabbix_agentd.conf の’MaxLinesPerSecond’ の値を上書きします。

- **mode** - 指定可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響）。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります。
- エージェントは”Forwarded events” ログのイベントを送信できません。
- Windows Eventing 6.0 をサポートしています。
- このアイテムに対してログ以外の情報の型を選択すると、ローカルタイムスタンプに加えて、ログの重大度およびソース情報も失われます。
- ログ監視に関する追加情報も参照してください。

例:

```
eventlog[Application]
eventlog[Microsoft-Windows-Application-Experience/Program-Compatibility-Assistant]
eventlog[Security,,"Failure Audit",,^(529|680)$]
eventlog[System,,"Warning|Error"]
eventlog[System,,,,^1$]
eventlog[Windows PowerShell,,,,,,skip]
eventlog[System,,,,@TWOSHORT] # ここでは `TWOSHORT` という名前のカスタム正規表現を参照しています（*Result is TRUE* 
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Windows イベントログ内の行数。
 戻り値: *Integer*。

パラメータ:

- **name** - イベントログチャネル名（イベントビューアー GUI の *Log Name*）;
- **regexp** - 必要なパターンを記述する正規表現（大文字と小文字を区別）;
- **severity** - 重大度を記述する正規表現（大文字と小文字を区別しない）。このパラメータは、次の値に基づく正規表現を受け付けます: ”Information”、”Warning”、”Error”、”Critical”、”Verbose”（Windows Vista 以降で動作）。
- **source** - ソース識別子を記述する正規表現（大文字と小文字を区別しない）;
- **eventid** - イベント識別子を記述する正規表現（大文字と小文字を区別）;
- **maxproclines** - エージェントが 1 秒あたりに解析する新規行の最大数（10000 を超えることはできません）。デフォルト値はzabbix_agentd.conf の 10*’MaxLinesPerSecond’ です。
- **mode** - 使用可能な値: *all*（デフォルト）または *skip* - 古いデータの処理をスキップします（新規作成されたアイテムにのみ影響します）。

コメント:

- このアイテムはアクティブチェックとして設定する必要があります;
- エージェントは”Forwarded events” ログからイベントを送信できません;
- Windows Eventing 6.0 をサポートしています;
- このアイテムに対して Log 以外の情報のタイプを選択すると、ローカルタイムスタンプに加えて、ログの重大度およびソース情報も失われます;
- ログ監視に関する追加情報も参照してください。

例:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 ネットワークインターフェースの一覧です（インターフェースの種類、ステータス、IPv4 アドレス、説明を含みます）。
 戻り値: *Text*。

コメント:

- マルチバイトのインターフェース名をサポートしています。
- 無効化されたインターフェースは一覧に表示されません。
- 一部のコンポーネントを有効化/無効化すると、Windows のインターフェース名における順序が変わる場合があります。
- 一部の Windows バージョン（たとえば Server 2008）では、インターフェース名で非 ASCII 文字をサポートするために最新の更新プログラムのインストールが必要な場合があります。

perf_counter[counter,<interval>]

 任意の Windows パフォーマンスカウンターの値。
 戻り値: *Integer*、*float*、*string*、または *text*（リクエストに応じて異なります）。

パラメータ:

- **counter** - カウンターへのパス。
- **interval** - 平均値を保存する直近 N 秒。`interval` は 1～900 秒（両端を含む）である必要があり、デフォルト値は 1 です。

コメント:

- このアイテムは UNC パスに従います。これはセキュリティリスクとなる可能性があります。
- `interval` は、複数のサンプルを必要とするカウンター（CPU 使用率など）で使用されるため、このチェックは毎回、直近「interval」秒の平均値を返します。
- 利用可能なカウンターの一覧を取得するには、Performance Monitor を使用できます。
- 参照: Windows パフォーマンスカウンター。

perf_counter_en[counter,<interval>]

 英語の任意の Windows パフォーマンスカウンターの値。
 戻り値: *Integer*、*float*、*string*、または *text*（リクエストに応じて異なります）。

パラメータ:

- **counter** - 英語でのカウンターへのパス;
- **interval** - 平均値を保存する直近 N 秒。`interval` は 1～900 秒の範囲内（両端を含む）である必要があり、デフォルト値は 1 です。

コメント:

- このアイテムは UNC パスに従うため、セキュリティリスクとなる可能性があります;
- `interval` は複数のサンプルを必要とするカウンター（CPU 使用率など）で使用されるため、このチェックは毎回、直近”interval” 秒の平均値を返します;
- このアイテムは **Windows Server 2008/Vista** 以降でのみサポートされています;
- 英語文字列の一覧は､次のレジストリキー `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib` を参照することで確認できます。

perf_instance.discovery[object]

 Windows パフォーマンスカウンターのオブジェクトインスタンスの一覧です。
ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **object** - オブジェクト名（ローカライズ済み）。

perf_instance_en.discovery[object]

 英語のオブジェクト名を使用して検出された、Windows パフォーマンスカウンターのオブジェクトインスタンスの一覧です。
ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

パラメータ:

- **object** - オブジェクト名（英語）。

proc_info[process,<attribute>,<type>]

 特定のプロセスに関するさまざまな情報です。
 戻り値: *Float*。

パラメータ:

- **process** - プロセス名;
- **attribute** - 要求するプロセス属性;
- **type** - 表現タイプ（同じ名前のプロセスが複数存在する場合に意味があります）

コメント:

- 次の `attributes` をサポートしています:
vmsize (デフォルト) - プロセスの仮想メモリサイズ（K バイト）
wkset - プロセスのワーキングセットサイズ（プロセスが使用する物理メモリ量）(K バイト）
pf - ページフォールト数
ktime - プロセスのカーネル時間（ミリ秒）
utime - プロセスのユーザー時間（ミリ秒）
io_read_b - I/O 操作中にプロセスが読み取ったバイト数
io_read_op - プロセスが実行した読み取り操作回数
io_write_b - I/O 操作中にプロセスが書き込んだバイト数
io_write_op - プロセスが実行した書き込み操作回数
io_other_b - 読み取りおよび書き込み以外の操作中にプロセスが転送したバイト数
io_other_op - 読み取りおよび書き込み以外でプロセスが実行した I/O 操作回数
gdiobj - プロセスが使用する GDI オブジェクト数
userobj - プロセスが使用する USER オブジェクト数;
- 有効な `types` は次のとおりです:
avg (デフォルト) - <process> という名前のすべてのプロセスの平均値
min - <process> という名前のすべてのプロセスの最小値
max - <process> という名前のすべてのプロセスの最大値
sum - <process> という名前のすべてのプロセスの値の合計;
- 64 ビットシステムでは、このアイテムを正しく動作させるために 64 ビットの Zabbix エージェントが必要です。

例:

```
proc_info[iexplore.exe,wkset,sum] # すべての Internet Explorer プロセスが使用している物理メモリ量を取得
proc_info[iexplore.exe,pf,avg] #Internet Explorer プロセスの平均ページフォールト数を取得
```

registry.data[key,<value name>]

 Windows レジストリキー内の指定した値名のデータを返します。
 戻り値: *Integer*、*string*、または *text*（値の型に応じます）

パラメータ:

- **key** - ルートキーを含むレジストリキー。ルートの省略形（例: HKLM）を使用できます。
- **value name** - キー内のレジストリ値名（デフォルトでは空文字列""）。値名が指定されていない場合は、既定値が返されます。

コメント:

- サポートされるルートの省略形:
HKCR - HKEY_CLASSES_ROOT
HKCC - HKEY_CURRENT_CONFIG
HKCU - HKEY_CURRENT_USER
HKCULS - HKEY_CURRENT_USER_LOCAL_SETTINGS
HKLM - HKEY_LOCAL_MACHINE
HKPD - HKEY_PERFORMANCE_DATA
HKPN - HKEY_PERFORMANCE_NLSTEXT
HKPT - HKEY_PERFORMANCE_TEXT
HKU - HKEY_USERS
- スペースを含むキーは二重引用符で囲む必要があります。

例:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #return the data of
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #return the data of t
```

registry.get[key,<mode>,<name regexp>]

 指定したキーにある Windows レジストリの値またはキーの一覧。
 戻り値: *JSON object*。

パラメータ:

- **key** - ルートキーを含むレジストリキー。ルートの省略形（例: HKLM）が使用できます（省略形の完全な一覧は registry.data[] のコメントを参照してください）。
- **mode** - 指定可能な値:
values（デフォルト）または *keys*;
- **name regexp** - 名前が正規表現に一致する値のみを検出します（デフォルトではすべての値を検出します）。mode が *values* の場合にのみ使用できます。

スペースを含むキーは二重引用符で囲む必要があります。

例:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
JSON には、キー、最後のサブキー、値の名前、値の型、値のデータの詳細が含まれます。
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] # このキー内のすべての値のデータを
JSON には、キー、最後のサブキー、値の名前、値の型、値のデータの詳細が含まれます。
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] # このキーのすべてのサブキーを返しま
JSON には、キーと最後のサブキーの詳細が含まれます。
```

service.discovery

 Windows サービスの一覧です。
ローレベルディスカバリで使用されます。
 戻り値: *JSON object*。

service.info[service,<param>]

 サービスに関する情報。
 戻り値: param が *state*、*startup* の場合は *Integer*、param が *displayname*、*path*、*user* の場合は *String*、param が *description* の場合は *Text*
state の場合: 0 - 実行中、1 - 一時停止中、2 - 開始保留中、3 - 一時停止保留中、4 - 再開保留中、5 - 停止保留中、6 - 停止、7 - 不明、255 - 該当するサービスなし
startup の場合: 0 - 自動、1 - 自動（遅延開始）、2 - 手動、3 - 無効、4 - 不明、5 - 自動トリガー開始、6 - 自動（遅延）トリガー開始、7 - 手動トリガー開始

パラメータ:

- **service** - 実際のサービス名、または MMC のサービススナップインに表示される表示名。
- **param** - *state*（デフォルト）、*displayname*、*path*、*user*、*startup*、または *description*。

コメント:

- service.info[service,state] や service.info[service] のようなアイテムは、同じ情報を返します。
- param が *state* の場合のみ、このアイテムは存在しないサービスに対しても値（255）を返します。

例:

```
service.info[SNMPTRAP] - SNMPTRAP サービスの状態;
service.info[SNMP Trap] - 同じサービスの状態。ただし、表示名を指定;
service.info[EventLog,startup] - EventLog サービスの起動タイプ
```

services[<type>,<state>,<exclude>]

 サービスの一覧です。
 戻り値: 空の場合は *0*、それ以外は改行で区切られたサービスの一覧を *Text* で返します。

パラメータ:

- **type** - *all* (デフォルト)、*automatic*、*manual*、または *disabled*;
- **state** - *all* (デフォルト)、*stopped*、*started*、*start_pending*、*stop_pending*、*running*、*continue_pending*、*pause_pending*、または *paused*;
- **exclude** - 結果から除外するサービスです。除外するサービスは、スペースを入れずにカンマで区切り、二重引用符で囲んで指定する必要があります。

例:

```
services[,started] # 開始済みサービスの一覧を返します;
services[automatic, stopped] # 実行されているべき停止中サービスの一覧を返します;
services[automatic, stopped, "service1,service2,service3"] #「service1」、「service2」、「service3」という名前のサ
```

vm.vmemory.size[<type>]

 仮想メモリのサイズを、バイト単位または総量に対する割合で返します。
 戻り値: バイトの場合は *Integer*、割合の場合は *float*。

パラメータ:

- **type** - 指定可能な値: *available* (利用可能な仮想メモリ), *pavailable* (利用可能な仮想メモリ、割合), *pused* (使用中の仮想メモリ、割合), *total* (仮想メモリの総量、デフォルト), または *used* (使用中の仮想メモリ)

コメント:

- 仮想メモリ統計の監視は、以下に基づいています:
  - Windows 上の仮想メモリ総量 (物理メモリ総量 + ページファイルサイズ);
  - Zabbix エージェントがコミットできる最大メモリ量;
  - システムまたは Zabbix エージェントの現在のコミット済みメモリ上限のうち、小さい方。

例:

```
vm.vmemory.size[pavailable] # 利用可能な仮想メモリを割合で返す
```

wmi.get[<namespace>,<query>]

 WMI クエリを実行し、最初に選択されたオブジェクトを返します。
 戻り値: *Integer*、*float*、*string* または *text* (リクエストに応じて異なります)。

パラメータ:

- **namespace** - WMI 名前空間。
- **query** - 単一のオブジェクトを返す WMI クエリ。

WMI クエリは WQL を使用して実行されます。

例:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] # 最初の物理ディス
```

wmi.getall[<namespace>,<query>]

 WMI クエリを実行し、応答全体を返します。ローレベルディスカバリで使用できます。
 戻り値: *JSON object*

パラメータ:

- **namespace** - WMI 名前空間。
- **query** - WMI クエリ。

コメント:

- WMI クエリは WQL で実行されます。
- 返された JSON 内のより具体的な値を指定するために、JSONPath の前処理を使用できます。

例:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #returns status inf
```

Windows サービスの監視

このチュートリアルでは、Windows サービスの監視を設定するための手順を順を追って説明します。
Zabbix サーバーおよびエージェントが設定済みで、稼働していることを前提としています。

ステップ 1

サービス名を取得します。

MMC の Services スナップインを開き、サービスのプロパティを表示すると、サービス名を確認できます。
General タブには、「Service name」というフィールドがあります。
その値が、監視用のアイテムを設定する際に使用する名前です。
たとえば、「workstation」サービスを監視したい場合、サービス名は **lanmanworkstation** になります。

ステップ 2

サービスを監視するためのアイテムを設定します。

アイテム `service.info[service,<param>]` は、特定のサービスに関する情報を取得します。必要な情報に応じて、`param` オプションを指定します。指定できる値は、*displayname*、*state*、*path*、*user*、*startup*、*description* です。`param` が指定されていない場合のデフォルト値は *state* です（`service.info[service]`）。

返り値の型は、選択した `param` によって異なります。*state* と *startup* では integer、*displayname*、*path*、*user* では文字列、*description* ではテキストになります。

例:

- キー*:* `service.info[lanmanworkstation]`
- 情報の型*:* 数値（符号なし）

アイテム `service.info[lanmanworkstation]` は、サービスの状態に関する情報を数値として取得します。数値を Web インターフェース上でテキスト表現（「0」を「Running」、「1」を「Paused」など）に対応付けるには、そのアイテムが設定されているホストで値のマッピングを設定できます。これを行うには、テンプレート *Windows services by Zabbix agent* または *Windows services by Zabbix agent active* をホストにリンクするか、これらのテンプレートで設定されている *Windows service state* 値のマッピングに基づく新しい値のマッピングをホスト上で設定します。

なお、上記の両テンプレートには、サービスを自動的に検出するディスカバリルールが設定されています。これを望まない場合は、テンプレートをホストにリンクした後で、ホストレベルでディスカバリルールを無効化できます。

Windows サービスのディスカバリ

ローレベルディスカバリ は、コンピュータ上のさまざまなエンティティに対して、アイテム、トリガー、グラフを自動的に作成する方法を提供します。
Zabbix は、サービスの正確な名前を把握したり、各サービスごとに手動でアイテムを作成したりすることなく、マシン上の Windows サービスの監視を自動的に開始できます。
フィルターを使用すると、対象となるサービスについてのみ、実際のアイテム、トリガー、グラフを生成できます。

2 SNMP エージェント

概要

通常 SNMP が有効になっているプリンター、ネットワークスイッチ、ルーター、UPS などのデバイスでは、完全なオペレーティングシステムや Zabbix エージェントをセットアップしようとするのは現実的でないことが多く、SNMP 監視を使用したい場合があります。

これらのデバイス上の SNMP エージェントが提供するデータを取得できるようにするには、Zabbix サーバーを `--with-net-snmp` フラグを指定して SNMP サポート付きで事前に設定しておく必要があります。また、アイテムの値が正しい形式で表示されるように、MIB ファイルをインストールすることも推奨されます。MIB ファイルがない場合、UTF-8 ではなく HEX で値が表示される、またはその逆になるなど、書式上の問題が発生することがあります。

SNMP チェックは UDP プロトコル上でのみ実行されます。

Zabbix サーバーおよびプロキシデーモンは、不正な SNMP レスポンスを受信した場合、次のような行をログに記録します。

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

これらのログは問題のあるすべてのケースを網羅しているわけではありませんが、結合リクエストを無効にすべき個々の SNMP デバイスを特定するのに役立ちます。

Zabbix サーバー/プロキシは、SNMP `walk` および `get` アイテムに対して最大 5 回まで再試行します。再試行の仕組みは、DNS 名前解決の失敗には適用されません。

従来の SNMP チェック（単一の OID 番号または文字列）では、Zabbix サーバー/プロキシは問い合わせに失敗した後、少なくとも 1 回は再試行します。これは、SNMP ライブラリの再試行メカニズム、または内部の結合処理メカニズムのいずれかによって行われます。

Warning:
SNMPv3 デバイスを監視する場合は、msgAuthoritativeEngineID（snmpEngineID または「Engine ID」とも呼ばれます）が 2 つのデバイス間で共有されないようにしてください。RFC 2571（セクション 3.1.1.1）によると、これは各デバイスで一意でなければなりません。

Warning:
RFC3414 では、SNMPv3 デバイスが engineBoots を永続化することを要求しています。これを行わないデバイスもあり、その場合、再起動後にそれらの SNMP メッセージは古いものとして破棄されます。このような状況では、サーバー/プロキシ上で SNMP キャッシュを手動でクリアする（-R snmp_cache_reload を使用）か、サーバー/プロキシを再起動する必要があります。

SNMP 監視の設定

SNMP を介してデバイスの監視を開始するには、次の手順を実行する必要があります:

ステップ 1

監視したいアイテムの SNMP 文字列（または OID）を確認します。

SNMP 文字列の一覧を取得するには、**snmpwalk** コマンド（net-snmp ソフトウェアの一部で、Zabbix のインストール時にあわせてインストールされているはずです）または同等のツールを使用します。

```
snmpwalk -v 2c -c public <host IP> .
```

ここでの「2c」は SNMP バージョンを表しているため、デバイス上の SNMP バージョン 1 を示す「1」に置き換えることもできます。

これにより、SNMP 文字列とその最新の値の一覧が表示されるはずです。
表示されない場合は、SNMP の「community」が標準の「public」と異なる可能性があります。その場合は、実際に使用されている値を確認する必要があります。

その後、一覧を確認して監視したい文字列を探します。例えば、スイッチのポート 3 で受信したバイト数を監視したい場合は、次の行にある IF-MIB::ifHCInOctets.3 という文字列を使用します。

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

次に、**snmpget** コマンドを使用して、IF-MIB::ifHCInOctets.3 の数値 OID を確認できます。

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

文字列の最後の数字は、監視対象のポート番号であることに注意してください。
あわせて参照: 動的インデックス

これにより、次のような結果が得られます。

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

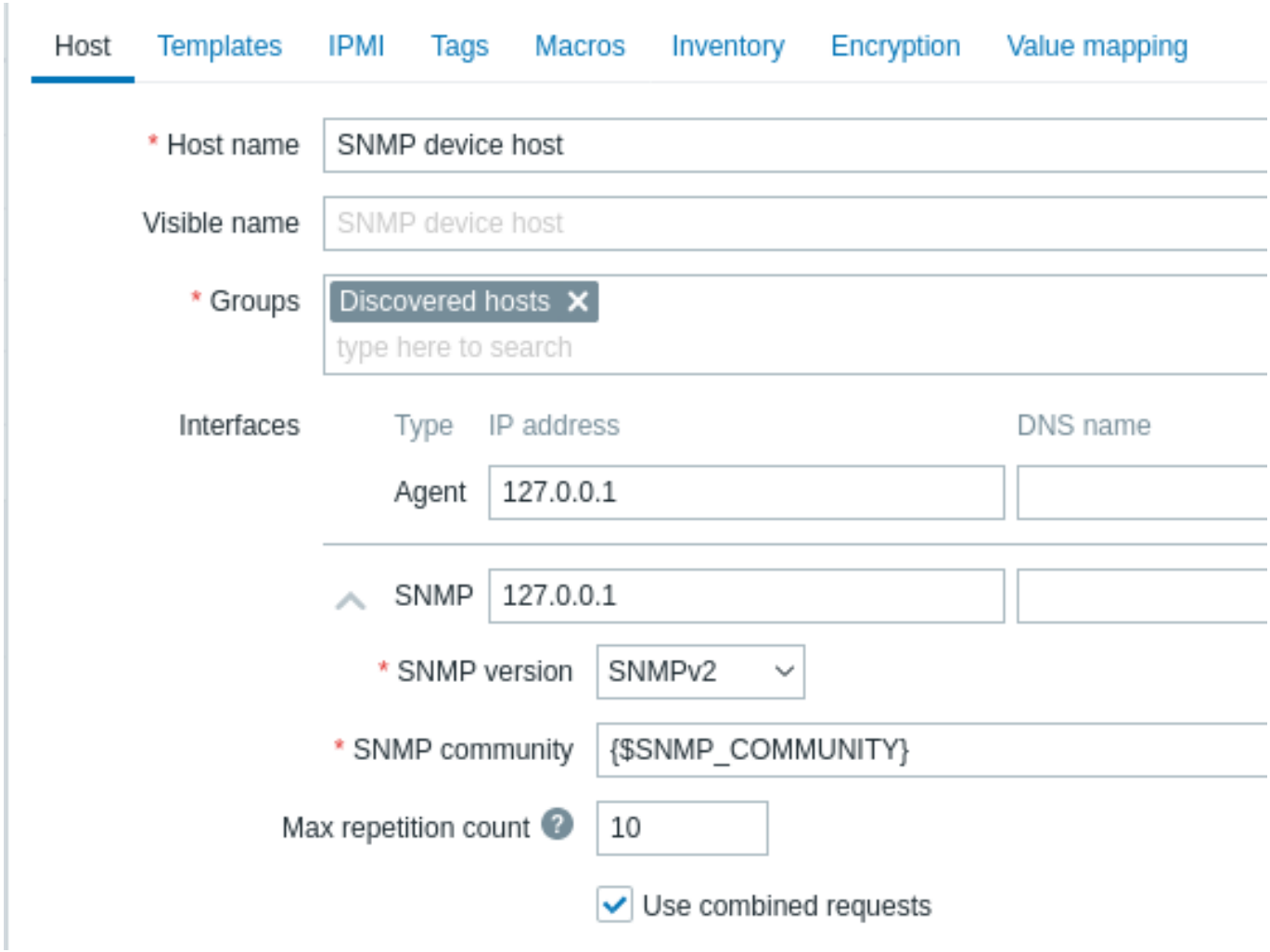
ここでも、OID の最後の数字はポート番号です。

Note:
よく使用される SNMP OID の一部は、Zabbix によって自動的に数値表現へ変換されます。

上記の最後の例では、値の型は”Counter64” で、内部的には ASN_COUNTER64 型に対応します。
サポートされている型の一覧は、ASN_COUNTER、ASN_COUNTER64、ASN_UINTEGER、ASN_UNSIGNED64、ASN_INTEGER、ASN_INTEGER64、ASN_FLOAT、ASN_DOUBLE、ASN_TIMETICKS、ASN_GAUGE、ASN_IPADDRESS、ASN_OCTET_STR、ASN_OBJECT_ID です。
これらの型は、**snmpget** の出力ではおおむね”Counter32”、”Counter64”、”UInteger32”、”INTEGER”、”Float”、”Double”、”Timeticks”、”Gauge32”、”IpAddress”、”OCTET STRING”、”OBJECT IDENTIFIER” に対応しますが、表示ヒントの有無によっては”STRING”、”Hex-STRING”、”OID” など、別の形式で表示されることもあります。

ステップ 2

デバイスに対応するホストを作成します。



ホストに SNMP インターフェースを追加します。

- IP アドレス/DNS 名とポート番号を入力します。
- ドロップダウンから *SNMP version* を選択します。
- 選択した SNMP バージョンに応じてインターフェース認証情報を追加します。
  - SNMPv1、v2 ではコミュニティ（通常は'public'）のみが必要です。
  - SNMPv3 では、より詳細なオプションが必要です（以下を参照）。
- ネイティブ SNMP バルクリクエスト（GetBulkRequest-PDUs）の最大繰り返し値（デフォルト: 10）を指定します。これは、SNMPv2 および v3 の `discovery[]` と `walk[]` アイテムでのみ使用されます。
  この値を高く設定しすぎると、SNMP エージェントのチェックがタイムアウトする可能性があることに注意してください。
- *Use combined requests* チェックボックスをオンにすると、SNMP リクエストの結合処理が可能になります（ネイティブ SNMP バルクリクエストの"walk" および"get" とは無関係です）。

SNMPv3 parameter	説明
Context name	SNMP サブネット上のアイテムを識別するためのコンテキスト名を入力します。 このフィールドではユーザーマクロが展開されます。
Security name	セキュリティ名を入力します。 このフィールドではユーザーマクロが展開されます。
Security level	セキュリティレベルを選択します。 **noAuthNoPriv** - 認証プロトコルもプライバシープロトコルも使用しません **AuthNoPriv** - 認証プロトコルは使用し、プライバシープロトコルは使用しません **AuthPriv** - 認証プロトコルとプライバシープロトコルの両方を使用します
Authentication protocol	認証プロトコルを選択します - *MD5*、*SHA1*、net-snmp 5.8 以降では *SHA224*、*SHA256*、*SHA384* または *SHA512*。
Authentication passphrase	認証パスフレーズを入力します。 このフィールドではユーザーマクロが展開されます。
Privacy protocol	プライバシープロトコルを選択します - *DES*、*AES128*、*AES192*、*AES256*、*AES192C* (Cisco) または *AES256C* (Cisco)。 プライバシープロトコルのサポートに関する注記を参照してください。

SNMPv3 parameter	説明
Privacy passphrase	プライバシーパスフレーズを入力します。 このフィールドではユーザーマクロが展開されます。

SNMPv3 の認証情報（セキュリティ名、認証プロトコル/パスフレーズ、プライバシープロトコル）が誤っている場合:

- Zabbix は net-snmp から ERROR を受信します。ただし、*Privacy passphrase* が誤っている場合は、Zabbix は net-snmp から TIMEOUT エラーを受信します。
- SNMP インターフェースの可用性は赤色（利用不可）に切り替わります。

Note:
Authentication protocol、*Authentication passphrase*、*Privacy protocol*、または *Privacy passphrase* の変更は、*Security name* を変更しない限り、通常は対応する SNMPv3 インターフェースが Zabbix で更新された際に自動的に適用されます。
Security name も変更された場合は、すべてのパラメータが直ちに更新されます。

提供されている SNMP テンプレートのいずれかを使用すると、一連のアイテムが自動的に追加されます。テンプレートを使用する前に、それがホストと互換性があることを確認してください。

Add をクリックしてホストを保存します。

プライバシープロトコルのサポート

使用しているオペレーティングシステムおよび net-snmp の設定によっては、一部のプライバシープロトコルを利用できない場合があります。

- 一部の新しいオペレーティングシステム（例: RHEL9）では、net-snmp パッケージで DES のサポートが廃止されています。
- RHEL 8、CentOS 8、Oracle Linux 8、Debian 12、Ubuntu LTS 22.04、openSUSE Leap 15.5 より古いオペレーティングシステムでは、暗号化プロトコル AES192 以上はそのままではサポートされていません。

net-snmp ライブラリが AES192+ をサポートしているかどうかを確認するには、次のいずれかの方法を使用します。

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

出力に `--enable-blumenthal-aes` が含まれている場合、AES192+ はサポートされています。

なお、net-snmp-config は SNMP の開発パッケージの一部（Debian/Ubuntu では libsnmp-dev、CentOS/RHEL/OL/SUSE では net-snmp-devel）であり、デフォルトではインストールされていない場合があります。

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

出力に `Invalid privacy protocol specified after -3x flag: AES-256` が含まれている場合、AES192+ はサポートされていません。
出力に `No hostname specified.` が含まれている場合、AES192+ はサポートされています。

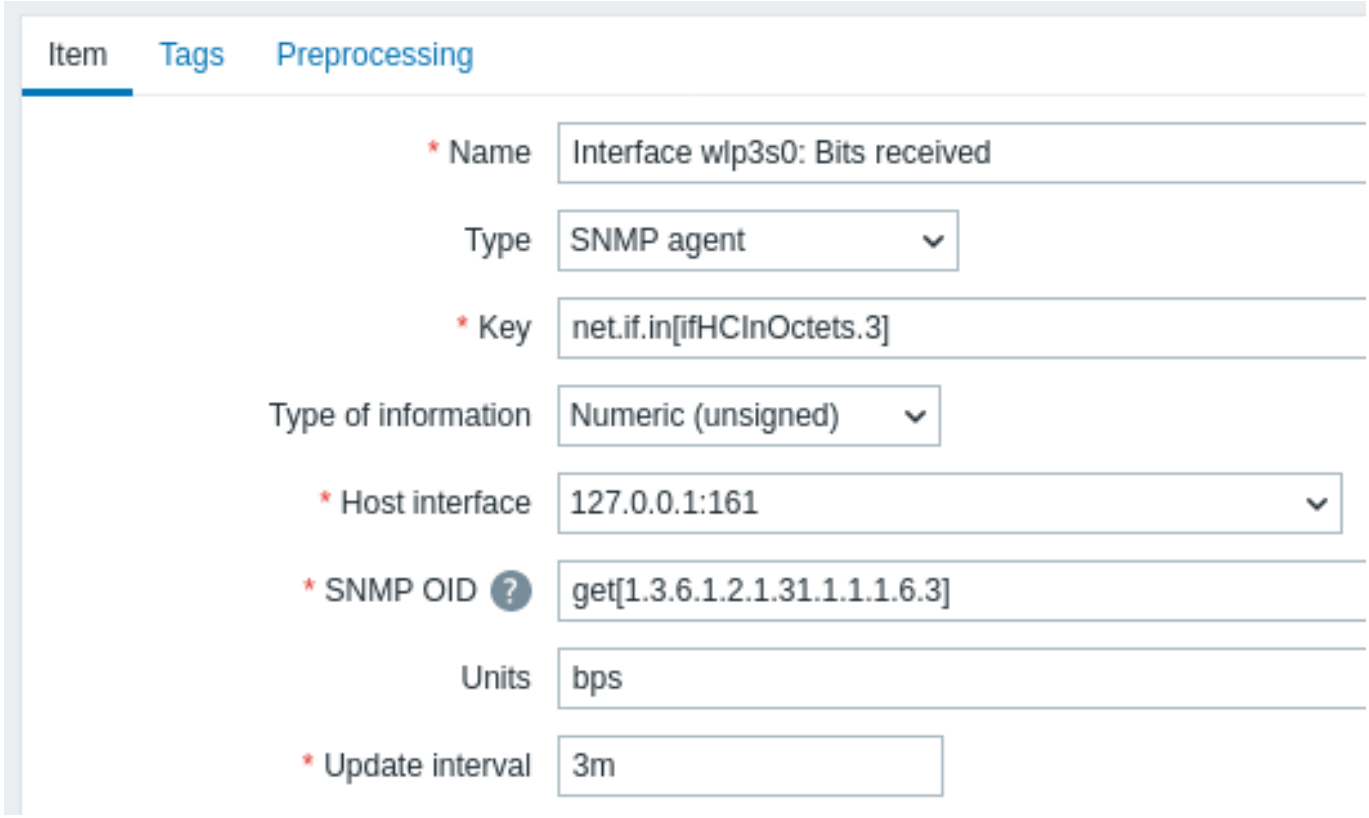
使用している net-snmp ライブラリが AES192 以上のプロトコルをサポートしていない場合は、`--enable-blumenthal-aes` オプションを指定して net-snmp を再コンパイルし、その後 `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-config` オプションを指定して Zabbix server を再コンパイルしてください。

ステップ 3

監視用のアイテムを作成します。

それでは、Zabbix に戻り、先ほど作成した SNMP ホストの アイテムをクリックしてください。ホスト作成時にテンプレートを使用したかどうかによって、ホストに関連付けられた SNMP アイテムの一覧が表示されるか、空の一覧だけが表示されます。ここでは、snmpwalk と snmpget を使って今収集した情報をもとに、自分でアイテムを作成する前提で進めますので、アイテムの作成をクリックしてください。

新しいアイテムのフォームで、必要なパラメータを入力します。



Parameter	Description
Name	アイテム名を入力します。
Type	ここでは **SNMP** エージェントを選択します。
Key	意味のあるキーを入力します。
Host interface	SNMP インターフェースが選択されていることを確認してください。例えば、スイッチやルーターのインターフェースです。

Parameter	Description
SNMP OID	サポートされているいずれかの形式を使用して OID 値を入力します。 **walk[OID1,OID2,...]** - 値のサブツリーを取得します。 例: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`。 このオプションでは、ネイティブ SNMP バルクリクエスト（GetBulkRequest-PDUs）を非同期で使用します。 このアイテムのタイムアウト設定は、アイテム設定 フォームで指定できます。デバイスに到達できない場合の長い遅延を避けるため、低めのタイムアウト値を設定することを検討してください。以前の試行がタイムアウトまたは失敗した場合、最大 5 回まで再試行されるためです（例: タイムアウト 3 秒の場合、待機時間が 15 秒になる可能性があります）。 このアイテムはマスターアイテムとして使用でき、前処理を使ってマスターからデータを抽出する依存アイテムを設定できます。 `walk[OID1,OID2,...]` のように、1 回の snmp walk で複数の OID を指定し、1 つずつ非同期に処理することも可能です。 バルクリクエストで結果が返されない場合は、バルクリクエストを使わずに単一レコードの取得が試行されます。 MIB 名はパラメータとしてサポートされています。したがって、`walk[1.3.6.1.2.1.2.2.1.2]` と `walk[ifDescr]` は同じ出力を返します。 複数の OID/MIB、例えば `walk[ifDescr,ifType,ifPhysAddress]` を指定した場合、出力は連結された一覧になります。 GetBulk リクエストは SNMPv2 および v3 のインターフェースで使用され、SNMPv1 のインターフェースでは GetNext が使用されます。バルクリクエストの max repetitions はインターフェースレベルで設定します。 max repetitions パラメータは、1 回のバルク応答で返される OID の最大数を決定することで、バルクリクエストに影響します。 値を大きくするとバルク応答も大きくなり、必要な通信回数を減らせます。ただし、すべてのデバイスが非常に大きな値をサポートしているとは限らず、問題が発生する可能性があります。 このアイテムは、-Oe -Ot -On パラメータを付けた snmpwalk ユーティリティの出力を返します。 このアイテムは、SNMP ディスカバリ でマスターアイテムとして使用できます。 **get[OID]** - 単一の値を非同期で取得します。 例: `get[1.3.6.1.2.1.31.1.1.1.6.3]` このアイテムのタイムアウト設定は、アイテム設定 フォームで指定できます。デバイスに到達できない場合の長い遅延を避けるため、低めのタイムアウト値を設定することを検討してください。以前の試行がタイムアウトまたは失敗した場合、最大 5 回まで再試行されるためです（例: タイムアウト 3 秒の場合、待機時間が 15 秒になる可能性があります）。 **OID** - （レガシー）単一のテキストまたは数値の OID を入力して、単一の値を同期的に取得します。必要に応じて他の値と組み合わせることもできます。 例: `1.3.6.1.2.1.31.1.1.1.6.3`。 このオプションでは、アイテムチェックのタイムアウトはサーバーの設定ファイルで設定された値と同じになります。 より高いパフォーマンスのために、`walk[OID]` および `get[OID]` アイテムの使用を推奨します。すべての `walk[OID]` および `get[OID]` アイテムは非同期で実行されます。つまり、他のチェックを開始する前に、あるリクエストへの応答を受信する必要はありません。DNS の名前解決も同様に非同期です。 非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPoller で定義）。非同期 SNMP ポーラーの数はStartSNMPPollers パラメータで定義されます。 どの方法で取得した場合でも、ネットワークトラフィック統計については、前処理タブで *1* 秒あたりの変化量ステップを追加する必要がある点に注意してください。そうしないと、最新の変化量ではなく、SNMP デバイスからの累積値を取得することになります。

必須入力フィールドには、赤いアスタリスクが付いています。

次に、アイテムを保存し、SNMP データについて 監視 > 最新データに移動してください。

例 1

一般的な例:

Parameter	Description
OID	1.2.3.45.6.7.8.0（または.1.2.3.45.6.7.8.0）
Key	< トリガーの参照として使用する一意の文字列 > たとえば、”my_param”。

OID は数値形式または文字列形式のいずれでも指定できることに注意してください。ただし、場合によっては文字列 OID を数値表現に変換する必要があります。この目的には、snmpget ユーティリティを使用できます。

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

例 2

稼働時間の監視:

パラメータ	説明
OID	MIB::sysUpTime.0
キー	router.uptime
値の型	Float
単位	uptime
前処理ステップ**:** カスタム乗数	0.01

ネイティブ SNMP バルクリクエスト

walk[OID1,OID2,...] アイテムでは、SNMP バージョン 2/3 で利用可能な、バルクリクエスト（GetBulkRequest-PDUs）向けのネイティブ SNMP 機能を使用できます。

SNMP の GetBulk リクエストは、複数の GetNext リクエストを実行し、その結果を単一のレスポンスで返します。
これは、通常の SNMP アイテムだけでなく、ネットワークの往復回数を最小限に抑えるための SNMP ディスカバリにも使用できます。

SNMP **walk[OID1,OID2,...]** アイテムは、1 回のリクエストでデータを収集するマスターアイテムとして使用でき、レスポンスは必要に応じて前処理を使用して解析する依存アイテムと組み合わせて利用できます。

ネイティブ SNMP バルクリクエストの使用は、SNMP リクエストを結合するオプションとは無関係であることに注意してください。SNMP リクエストの結合は、複数の SNMP リクエストをまとめる Zabbix 独自の方法です（次のセクションを参照）。

SNMP バルクアイテムでは、いずれかのパケットが失われた場合の失敗を回避するため、最大 5 回の再試行が行われます。
`get` および `walk` を使用する SNMP アイテムのタイムアウト（アイテム設定 フォームで設定）は、セッション全体に対して設定されます。
タイムアウトは、データが完全に取得されたかどうかに関係なく適用されます。データが部分的にしか受信されなかった場合（たとえば、複数の OID のうち 1 つについてのみ正常にデータが収集された場合）、そのアイテムは”Only partial data received” というメッセージとともに未サポートになります。
タイムアウトに達すると再試行が行われ、タイムアウトはリセットされ、最後のリクエストが再送されます。これにより、単一のパケットが失われた場合や到着が遅すぎた場合でも、最後のリクエストからセッションを継続できます。
デバイスに到達できない場合の長い遅延を避けるため、低めのタイムアウト値を設定することを検討してください。前の試行がタイムアウトまたは失敗した場合、最大 5 回の再試行が行われるためです（例: タイムアウトが 3 秒の場合、待機時間が 15 秒になる可能性があります）。

組み合わせ処理の内部動作

Zabbix サーバーおよびプロキシは、1 回のリクエストで複数の値を SNMP デバイスに問い合わせることがあります。これは、いくつかの種類の SNMP アイテムに影響します。

- 通常の SNMP アイテム
- 動的インデックス付きSNMP アイテム
- SNMP のローレベルディスカバリルール

同一インターフェース上にあり、パラメータが同一のすべての SNMP アイテムは、同時に問い合わせるようスケジュールされます。最初の 2 種類のアイテムは、poller によって最大 128 アイテムのバッチ単位で処理されます。一方、ローレベルディスカバリルールは、従来どおり個別に処理されます。

より低いレベルでは、値を問い合わせるために実行される操作には 2 種類あります。複数の指定オブジェクトを取得する方法と、OID ツリーを walk する方法です。

「取得」では、最大 128 個の variable binding を持つ GetRequest-PDU が使用されます。「walk」では、SNMPv1 には GetNextRequest-PDU が使用され、SNMPv2 および SNMPv3 には”max-repetitions” フィールドが最大 128 の GetBulkRequest が使用されます。

したがって、各 SNMP アイテム種別における組み合わせ処理の利点は、以下のとおりです。

- 通常の SNMP アイテムは、「取得」の改善による恩恵を受けます。
- 動的インデックス付き SNMP アイテムは、「取得」と「walk」の両方の改善による恩恵を受けます。「取得」はインデックス検証に使用され、「walk」はキャッシュ構築に使用されます。
- SNMP ローレベルディスカバリルールは、「walk」の改善による恩恵を受けます。

ただし、技術的な問題として、すべてのデバイスが 1 回のリクエストで 128 個の値を返せるわけではありません。常に適切な応答を返すものもありますが、潜在的な応答が一定の上限を超えると、”tooBig(1)” エラーを返すか、まったく応答しなくなるものもあります。

特定のデバイスに対して問い合わせるオブジェクト数の最適値を見つけるために、Zabbix は次の戦略を使用します。まず慎重に、1 回のリクエストで 1 つの値を問い合わせることから開始します。それが成功した場合、1 回のリクエストで 2 つの値を問い合わせます。それも成功した場合、1 回のリクエストで 3 つの値を問い合わせ、その後も同様に、問い合わせるオブジェクト数を 1.5 倍しながら増やしていきます。その結果、リクエストサイズは次のような系列になります: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128。

ただし、デバイスが適切な応答を返さなくなった場合（たとえば 42 個の変数に対して）、Zabbix は 2 つのことを行います。

まず、現在のアイテムバッチについては、1 回のリクエスト内のオブジェクト数を半分にし、21 個の変数を問い合わせます。デバイスが稼働していれば、28 個の変数では動作することが確認されており、21 個はそれよりかなり少ないため、大半のケースでこの問い合わせは成功するはずです。それでも失敗する場合、Zabbix は値を 1 つずつ問い合わせる方式にフォールバックします。この時点でもなお失敗する場合、デバイスが確実に応答しておらず、リクエストサイズは問題ではないと判断されます。

次に、後続のアイテムバッチについては、最後に成功した変数数（この例では 28）から開始し、上限に達するまでリクエストサイズを 1 ずつ増やしていきます。たとえば、最大応答サイズが 32 変数であるとすると、後続のリクエストサイズは 29、30、31、32、33 となります。最後のリクエストは失敗し、Zabbix は以後サイズ 33 のリクエストを発行しなくなります。それ以降、このデバイスに対して Zabbix は最大 32 個の変数までしか問い合わせません。

この変数数で大きな問い合わせが失敗する場合、考えられる理由は 2 つあります。デバイスが応答サイズを制限する際に用いる正確な基準は分かりませんが、Zabbix は変数数を使ってそれを近似しようとします。したがって、1 つ目の可能性は、この変数数が一般的なケースにおけるデバイスの実際の応答サイズ上限付近にあることです。つまり、応答が上限未満になることもあれば、上限を超えることもあります。2 つ目の可能性は、いずれかの方向の UDP パケットが単に失われたことです。これらの理由から、Zabbix は問い合わせ失敗時に、デバイスがより余裕を持って応答できる範囲に入るよう最大変数数を減らしますが、それは最大 2 回までに制限されます。

上記の例では、32 変数の問い合わせがたまたま失敗した場合、Zabbix はその数を 31 に減らします。それもたまたま失敗した場合、Zabbix はその数を 30 に減らします。ただし、Zabbix は 30 未満には減らしません。以降の失敗は、デバイスの上限ではなく UDP パケットの損失によるものとみなすためです。

ただし、デバイスが別の理由で組み合わせリクエストを適切に処理できず、上記のヒューリスティックが機能しない場合は、各インターフェースにある「Use combined requests」設定を使用して、そのデバイスに対する組み合わせリクエストを無効にできます。

組み合わせリクエストによって部分的または不正な形式の応答が発生し、その結果、1 秒あたり（delta）の計算が不正確になる場合（たとえば、インターフェースカウンターに見かけ上のスパイクが発生する場合）は、影響を受けるインターフェースで *Use combined requests* を無効にして、アイテムごとの個別問い合わせを強制してください。これにより、誤ったスパイクを防げることがよくあります。または、非同期の `get[]` または `walk[]` アイテムの使用を検討してください。これらは非同期に実行され、インターフェースごとの *Use combined requests* のバッチ処理の対象になりません。そのため、従来の同期 OID チェックの代わりに使用して、組み合わせリクエストに関連する問題を回避できます。影響を受けるデバイスの特定には、概要 セクションに示されているものと同様のサーバー/プロキシログエントリを確認してください。

さらに、インターフェースが頻繁に利用不可になる場合は、リクエスト頻度を下げるために、Zabbix server またはZabbix proxy の設定ファイルで `UnavailableDelay` パラメータを増やす必要がある場合があります。ディスカバリまたは OID walk 中に部分的なデータを受信すると、アイテムが未サポートになることがあります。

1 ダイナミックインデックス

概要

必要なインデックス番号（例えば、ネットワーク・インターフェース）を SNMP OID の中から見つけることができるかもしれませんが
 インデックス番号が常に同じであるとは限りません。

インデックス番号は動的であり、時間の経過とともに変化し、その結果、アイテムが動作しなくなる可能性があります。

このようなシナリオを回避するために、インデックス番号が変化する可能性を考慮した OID を定義することが可能です。

たとえば、次のようなインデックス値を取得する必要があるとします。
 Cisco のデバイスの **GigabitEthernet0/1** インターフェイスに対応する **ifInOctets** に対応する
 インデックス値を取得する必要がある場合、次の OID を使用します。

```
ifInOctets["index", "ifDescr", "GigabitEthernet0/1"]
```

文法

OID の特別な構文が使用されます。

<OID of data>[”index”,”<base OID of index>”,”<string to search for>”]

パラメーター	説明
OID of data	アイテムのデータ検索に使用するメイン OID。
index	処理方法。現在、1 つの方法がサポートされています: **index** - インデックスを検索し、データ OID に追記する。
base OID of index	この OID を検索して、文字列に対応するインデックス値を取得する。
string to search for	検索時に値と完全一致するために使用する文字列。大文字と小文字を区別する。

例

apache プロセスのメモリ使用量を取得します。

この OID 構文を使用する場合、以下のようになります。

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index", "HOST-RESOURCES-MIB::hrSWRunPath","/usr/sbin/apache2"]
```

インデックス番号はここで検索されます。

```
...
HOST-RESOURCES-MIB::hrSWRunPath.5376 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5377 = STRING: "/sbin/getty"
HOST-RESOURCES-MIB::hrSWRunPath.5388 = STRING: "/usr/sbin/apache2"
HOST-RESOURCES-MIB::hrSWRunPath.5389 = STRING: "/sbin/sshd"
...
```

これで、5388 というインデックスができました。
 このインデックスは、データ OID に追加され、目的の値を受け取ります。

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

index lookup のキャッシュ

動的インデックスアイテムが要求された場合、Zabbix はインデックスのベース OID の下にある
 SNMP テーブル全体を取得し、キャッシュします。
 これは、別のアイテムが同じベース OID を後で参照する場合に備えて行われます。
 Zabbix は監視ホストに再度問い合わせる代わりに、キャッシュからインデックスを検索します。
 各ポーラープロセスは別々のキャッシュを使用していることに注意してください。

その後のすべての値取得操作では、見つかったインデックスのみが検証されます。
 インデックスが変更されていない場合、値が要求されます。
 変更されたインデックスに遭遇した各ポーラは、インデックス SNMP テーブルを再度検索し、キャッシュを再構築します。

2 特殊な OID

最もよく使われる SNMP OID のいくつかは、Zabbix によって自動的に数値表現に変換されます。例えば、**ifIndex** は **1.3.6.1.2.1.2.2.1.1** に、**ifIndex.0** は **1.3.6.1.2.1.2.2.1.1.0** に変換されます。

この表は、特殊な OID のリストです。

特殊 OID	識別子	説明
ifIndex	1.3.6.1.2.1.2.2.1.1	A unique value for each interface.
ifDescr	1.3.6.1.2.1.2.2.1.2	A textual string containing information about the interface.This string should include the name of the manufacturer, the product name and the version of the hardware interface.
ifType	1.3.6.1.2.1.2.2.1.3	The type of interface, distinguished according to the physical/link protocol(s) immediately ‘below’ the network layer in the protocol stack.
ifMtu	1.3.6.1.2.1.2.2.1.4	The size of the largest datagram which can be sent / received on the interface, specified in octets.
ifSpeed	1.3.6.1.2.1.2.2.1.5	An estimate of the interface’s current bandwidth in bits per second.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	The interface’s address at the protocol layer immediately ‘below’ the network layer in the protocol stack.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	The current administrative state of the interface.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	The current operational state of the interface.

特殊 OID	識別子	説明
ifInOctets	1.3.6.1.2.1.2.2.1.10	The total number of octets received on the interface, including framing characters.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	The number of subnetwork-unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	The number of non-unicast (i.e., subnetwork- broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
ifInErrors	1.3.6.1.2.1.2.2.1.14	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	The total number of octets transmitted out of the interface, including framing characters.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	The number of outbound packets that could not be transmitted because of errors.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	The length of the output packet queue (in packets).

3 MIB ファイル

はじめに

MIB は Management Information Base の略です。MIB ファイルを使用すると、OID（Object Identifier）のテキスト表現を利用できます。Zabbix で SNMP デバイスを監視する際には生の OID を使用することも可能ですが、テキスト表現を使用したい場合は、MIB ファイルをインストールする必要があります。

例えば、

`ifHCOutOctets`

は、OID

`1.3.6.1.2.1.31.1.1.1.10`

のテキスト表現です。

MIB ファイルのインストール

Debian ベースのシステムの場合:

```
apt install snmp-mibs-downloader
download-mibs
```

RedHat ベースのシステムの場合:

```
dnf install net-snmp-libs
```

MIB ファイルの有効化

RedHat ベースのシステムでは、MIB ファイルはデフォルトで有効になっているはずです。
Debian ベースのシステムでは、/etc/snmp/snmp.conf ファイルを編集し、mibs : と記載された行をコメントアウトする必要があります。

```
# ライセンス上の理由により、snmpパッケージにはMIBファイルが含まれていないため、
# MIBの読み込みはデフォルトで無効になっています。MIBを追加した場合は、
# 以下の行をコメントアウトすることで、読み込みを再度有効にできます。
mibs :
```

MIB ファイルのテスト

SNMP MIB のテストは、snmpwalk ユーティリティを使用して実行できます。インストールされていない場合は、以下の手順を使用してください。

Debian ベースのシステムの場合:

```
apt install snmp
```

RedHat ベースのシステムの場合:

```
dnf install net-snmp-utils
```

その後、ネットワークデバイスを照会した際に、次のコマンドでエラーが発生してはいけません:

```
$ snmpwalk -v 2c -c public <NETWORK DEVICE IP> ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 176137634
IF-MIB::ifInOctets.2 = Counter32: 0
IF-MIB::ifInOctets.3 = Counter32: 240375057
IF-MIB::ifInOctets.4 = Counter32: 220893420
[...]
```

Zabbix で MIB を使用する

最も重要な注意点は、Zabbix のプロセスは MIB ファイルに加えられた変更を認識しないということです。したがって、変更を行うたびに Zabbix サーバーまたはプロキシを再起動する必要があります。例えば、次のようにします。

```
systemctl restart zabbix-server
```

その後、MIB ファイルに加えられた変更が有効になります。

カスタム MIB ファイルの使用

すべての GNU/Linux ディストリビューションには標準の MIB ファイルが付属しています。しかし、一部のデバイスベンダーは独自のものを提供しています。

たとえば、CISCO-SMI MIB ファイルを使用したいとします。以下の手順でダウンロードしてインストールできます。

```
wget ftp://ftp.cisco.com/pub/mibs/v2/CISCO-SMI.my -P /tmp
mkdir -p /usr/local/share/snmp/mibs
grep -q '^mibdirs +/usr/local/share/snmp/mibs' /etc/snmp/snmp.conf 2>/dev/null || echo "mibdirs +/usr/loca
cp /tmp/CISCO-SMI.my /usr/local/share/snmp/mibs
```

これで使用できるようになるはずです。MIB ファイル内のオブジェクト *ciscoProducts* の名前を OID に変換してみてください。

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

OID ではなくエラーが表示される場合は、前述のコマンドですべてエラーが返されていないことを確認してください。

オブジェクト名の変換が成功したので、カスタム MIB ファイルを使用する準備は完了です。クエリで使用されている MIB 名のプレフィックス（*CISCO-SMI::*）に注意してください。これは、コマンドラインツールおよび Zabbix で使用する際に必要になります。

この MIB ファイルを Zabbix で使用する前に、Zabbix サーバー/プロキシを忘れずに再起動してください。

Attention:
MIB ファイルには依存関係がある場合があることに注意してください。つまり、ある MIB が別の MIB を必要とすることがあります。これらの依存関係を満たすには、影響を受ける MIB ファイルをすべてインストールする必要があります。

3 SNMP トラップ

概要

SNMP トラップの受信は、SNMP 対応デバイスへのクエリとは逆です。

この場合、情報は SNMP 対応デバイスから snmptrapd へ送信され、ファイルから Zabbix サーバーまたは Zabbix プロキシによって収集または” トラップ” されます。

通常、何らかの条件が変化するとトラップが送信され、エージェントはポート 162 でサーバーに接続します (クエリに使用されるエージェント側のポート 161 とは対照的です)。トラップを使用すると、クエリ間隔中に発生し、クエリデータによって見逃される可能性のあるいくつかの短い障害を検出できます。

Zabbix での SNMP トラップの受信は、**snmptrapd** と、トラップを Zabbix に渡すためのメカニズムの 1 つ (Bash または Perl スクリプト、SNMPTT) で動作するように設計されています。

Note:
Zabbix の構成後にトラップ監視をセットアップする最も簡単な方法は、Bash スクリプトソリューションを使用することです。これは、最近のディストリビューションでは Perl と SNMPTT が欠落していることが多く、より複雑な構成が必要になるためです。ただし、このソリューションでは、`traphandle` として構成されたスクリプトが使用されます。本番システムでのパフォーマンスを向上させるには、組み込みの Perl ソリューション (`do perl` オプション付きのスクリプトまたは SNMPTT) を使用してください。

トラップを受信するワークフロー:

1. `snmptrapd` がトラップを受信
2. `snmptrapd` はトラップをレシーバースクリプト (Bash、Perl) または SNMPTT に渡す
3. 受信者はトラップを解析、フォーマットし、ファイルに書き込む
4. Zabbix SNMP トラッパーは、トラップファイルを読み取り、解析
5. Zabbix はトラップごとに、受信したトラップアドレスと一致するホストインターフェースを持つすべての”SNMP トラッパー” アイテムを検索します。マッチングでは、ホストインターフェースで選択した”IP” または”DNS” のみが使用されることに注意してください。
6. 見つかったアイテムごとに、トラップは `snmptrap[regexp]` の正規表現と比較されます。トラップは一致したすべてのアイテムの値として設定されます。一致するアイテムが見つからず、snmptrap.fallback アイテムが存在する場合、トラップはその値として設定されます。
7. トラップがアイテムの値として設定されていない場合、Zabbix はデフォルトで一致しないトラップをログに記録します。(これは、管理 > 一般設定 > その他の” マッチしない SNMP トラップをログに記録” で設定されます。)

HA フェイルオーバーに関する注意事項

高可用性 (HA) ノードの切り替え中、Zabbix は最後の ISO 8601 タイムスタンプ内の最後のレコードの後に処理を続行します。同じレコードが見つからない場合は、タイムスタンプのみを使用して最後の位置を識別します。

SNMP トラップの設定

このアイテムタイプには、次のフロントエンド設定が必要です。

1. ホスト用の SNMP インターフェースを作成

- データ収集 > ホストで、ホストを作成/編集し、インターフェースフィールドに、インターフェースタイプ”SNMP” を追加、IP または DNS アドレスを指定します。

 受信した各トラップのアドレスは、すべての SNMP インターフェイスの IP および DNS アドレスと比較され、対応するホストを見つけます。

2. アイテムの設定

- データ収集 > ホストで、必要なアイテムを作成/編集します。
- キーフィールドで、SNMP トラップキーの 1 つを使用します。

キー		
説明	戻り値	コメント
snmptrap[regexp] **regexp** で指定された正規表現に一致するすべての SNMP トラップをキャッチします。regexp が指定されていない場合、すべてのトラップをキャッチします。	SNMP トラップ	このアイテムは、SNMP インターフェースに対してのみ設定できます。 このアイテムキーのパラメーターでは、ユーザーマクロとグローバル正規表現がサポートされています。
snmptrap.fallback		

キー		
そのインターフェースの snmptrap[] アイテムのいずれによってもキャッチされなかったすべての SNMP トラップをキャッチします。	SNMP トラップ	このアイテムは、SNMP インターフェースに対してのみ設定できます。

Note:
現時点では、複数行の正規表現一致はサポートされていません。

- データ型をタイムスタンプが解析されるように” ログ” に設定します。” 数値” などの他の形式も受け入れられますが、カスタムトラップハンドラーが必要になる場合があります。

SNMP トラップ監視の設定

Zabbix サーバー/プロキシの設定

トラップを読み取るには、Zabbix サーバーまたはプロキシが SNMP トラッパープロセスを開始し、SNMPTT または Bash/Perl トラップレシーバーによって書き込まれているトラップファイルを指すように構成されている必要があります。これを行うには設定ファイル (zabbix_server.confまたはzabbix_proxy.conf) を編集します。

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
systemd パラメーター**PrivateTmp**が使用されている場合、このファイルは*/tmp* では機能しません。

Bash トラップレシーバーの設定

要件: snmptrapd のみ。

Bash トラップレシーバースクリプトを使用して、trapper ファイルを使用して snmptrapd から Zabbix サーバーにトラップを渡すことができます。設定するには、snmptrapd 設定ファイル (`snmptrapd.conf`) に `traphandle` オプションを追加します。例を参照してください。

Note:
snmptrapd は設定の変更を取得するために再起動する必要がある場合があります。

Perl トラップレシーバーの設定

要件: Perl、--enable-embedded-perl 付きでコンパイルされた Net-SNMP（Net-SNMP 5.4 以降ではデフォルトで有効）

Perl トラップレシーバー（misc/snmptrap/zabbix_trap_receiver.pl を参照）を使用すると、snmptrapd から Zabbix サーバーへ直接トラップを渡すことができます。設定するには、次のようにします。

- snmptrapd の設定ファイル（snmptrapd.conf）に Perl スクリプトを追加します。例:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- レシーバーを設定します。例:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
設定の変更を反映するために、snmptrapd の再起動が必要になる場合があります。

Note:
スクリプト名が引用符で囲まれていない場合、snmptrapd は次のようなメッセージを出して起動を拒否します。


```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

SNMPTT の設定

まず、snmptrapd が SNMPTT を使用するように設定されている必要があります。

> **Note:**
> 最高のパフォーマンスを得るには、トラップを渡すために **snmptthandler-embedded** を使用するデーモンとして SNMPTT を設定してください。手順については、SNMPTT の設定を参照してください。

SNMPTT がトラップを受信するように設定されたら、`snmptt.ini` を設定します。

1. NET-SNMP パッケージの Perl モジュールの使用を有効にします。

```
net_snmp_perl_enable = 1
```

2. Zabbix が読み取るトラップファイルにトラップを記録します。

```
log_enable = 1
log_file = [TRAP FILE]
```

3. 日時形式を設定します。

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

> **Warning:**
> "net-snmp-perl" パッケージは RHEL 8.0-8.2 で削除され、RHEL 8.3 で再追加されました。詳細については、既知の問題を参照してください。

次に、Zabbix が認識できるようにトラップを整形します（snmptt.conf を編集）。

1. 各 FORMAT 文は"ZBXTRAP [address]" で始まる必要があります。ここで、[address] は Zabbix 上の SNMP インターフェースの IP アドレスおよび DNS アドレスと比較されます。例:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. SNMP トラップ形式の詳細については、以下を参照してください。

> **Attention:**
> 不明なトラップは使用しないでください。Zabbix はそれらを認識できません。不明なトラップは、snmptt.conf で一般イベントを定義することで処理できます。

>
> ```
> EVENT general .* "General event" Normal
> ```

SNMP トラップ形式

カスタマイズされたすべての Perl トラップレシーバーおよび SNMPTT トラップ設定では、トラップを次の形式で整形する必要があります。

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

ここで

- [timestamp] - "%Y-%m-%dT%H:%M:%S%z" 形式のタイムスタンプ
- ZBXTRAP - この行で新しいトラップが開始されることを示すヘッダー
- [address] - このトラップに対応するホストを見つけるために使用される IP アドレス

処理中に、メッセージから"ZBXTRAP" と"[address]" は削除されることに注意してください。トラップがこれ以外の形式になっている場合、Zabbix がトラップを想定外に解析する可能性があります。

トラップの例:

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events" localhost - ZBXTRAP 192.168.1.1 Link
```

これにより、IP=192.168.1.1 の SNMP インターフェースに対して、次のトラップになります。

```
2024-01-11T15:28:47+0200 .1.3.6.1.6.3.1.1.5.3 Normal "Status Events"
localhost - Link down on interface 2. Admin state: 1. Operational state: 2
```

システム要件

Note:
アイテムの値が正しい形式で表示されるようにするため、MIB ファイルをインストールすることを推奨します。MIB ファイルがない場合、UTF-8 ではなく HEX で値が表示される、またはその逆になるなど、書式設定の問題が発生する可能性があります。

ラージファイルサポート

Zabbix は、SNMP トラッパーのラージファイルをサポートしています。Zabbix が読み取れる最大ファイルサイズは 2^63 (8 EiB) です。ファイルシステムによってはファイルサイズに下限があることに注意してください。

ログローテーション

Zabbix はログローテーションの仕組みを提供していません。これはユーザーが処理する必要があります。トラップが失われないようにするため、ログローテーションではまず古いファイルの名前を変更し、その後で削除する必要があります:

1. Zabbix は最後に認識した場所でトラップファイルを開き、手順 3 に進みます
2. Zabbix は、現在開いているファイルの inode 番号を定義されたトラップファイルの inode 番号と比較することで、そのファイルがローテーションされたかどうかを確認します。開いているファイルがない場合、Zabbix は最後の場所をリセットして手順 1 に戻ります。
3. Zabbix は現在開いているファイルからデータを読み取り、新しい場所を設定します。
4. 新しいデータが解析されます。これがローテーションされたファイルであった場合、そのファイルは閉じられ、手順 2 に戻ります。
5. 新しいデータがない場合、Zabbix は 1 秒間スリープして手順 2 に戻ります。

ファイルシステム

Trap ファイルの実装のため、Zabbix はファイルを区別するために inode をサポートする
 ファイルシステムを必要とします (この情報は stat() 呼出で取得します)。

異なる SNMP プロトコルバージョンを使用したセットアップ例

この例では、snmptrapd と Bash の受信スクリプトを使用して、トラップを Zabbix サーバーに渡します。

セットアップ:

1. Zabbix で SNMP trapper を起動し、トラップファイルを設定します。`zabbix_server.conf` に以下を追加します:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Bash スクリプトを `/usr/sbin/zabbix_trap_handler.sh` にダウンロードします:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/Dock
```

必要に応じて、スクリプト内の ZABBIX_TRAPS_FILE 変数を調整してください。デフォルト値を使用するには、まず親ディレクトリを作成します:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. `snmtrapd.conf` に以下を追加します（動作する例も参照してください）

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

Note:
設定変更を反映するために、snmptrapd の再起動が必要になる場合があります。

4. SNMP アイテム TEST を作成します（初期の設定要件に注意してください）:

タイプ: SNMP トラップ
情報の型: ログホストインターフェース: SNMP 127.0.0.1
キー: `snmptrap["linkUp"]`
ログの時間形式: yyyyMMdd.hhmmss

ISO 8601 の日付および時刻形式が使用されていることに注意してください。

5. 次に、選択した SNMP プロトコルバージョン用に `snmptrapd` を設定し、`snmptrap` ユーティリティを使用してテストトラップを送信します。

SNMPv1、SNMPv2

SNMPv1 および SNMPv2 プロトコルは、「コミュニティ文字列」認証に依存しています。以下の例では、コミュニティ文字列として"secret" を使用します。これは SNMP トラップ送信元でも同じ値に設定する必要があります。

本番環境で依然として広く使用されている一方で、SNMPv2 は暗号化も実際の送信元認証も提供しない点に注意してください。データは平文で送信されるため、これらのプロトコルバージョンは、プライベートネットワークのような安全な環境でのみ使用し、パブリックネットワークや第三者のネットワーク上では決して使用しないでください。

SNMP version 1 は 64 ビットカウンターをサポートしておらず、レガシープロトコルと見なされているため、現在ではほとんど使用されていません。

SNMPv1 または SNMPv2 トラップの受信を有効にするには、snmptrapd.conf に以下の行を追加します。"secret" は、SNMP トラップ送信元で設定されている SNMP コミュニティ文字列に置き換えてください:

```
authCommunity log,execute,net secret
```

次に、snmptrap を使用してテストトラップを送信できます。この例では、一般的な"link up" OID を使用します:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 は SNMPv1/v2 のセキュリティ上の問題に対処し、認証と暗号化を提供します。認証方式として MD5 または複数の SHA を、暗号方式として DES/複数の AES を使用できます。

SNMPv3 トラップの受信を有効にするには、snmptrapd.conf に以下の行を追加します:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

Attention:
このユーザーセキュリティモデルでスクリプトの実行を許可する"execute" キーワードに注意してください。

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

Warning:
AES192 や AES256 などの強力な暗号方式を使用したい場合は、バージョン 5.8 以降の net-snmp を使用してください。configure のオプション --enable-blumenthal-aes を指定して再コンパイルが必要になる場合があります。古いバージョンの net-snmp は AES192/AES256 をサポートしていません。あわせて参照: Strong Authentication or Encryption。

検証

どちらの例でも、/var/lib/zabbix/snmptraps/snmptraps.log に次のような行が表示されます:

```
2024-01-30T10:04:23+0200 ZBXTRAP 127.0.0.1
UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Zabbix のアイテム値は次のようになります:

```
2024-01-30 10:04:23 2024-01-30 10:04:21

2024-01-30T10:04:21+0200 UDP: [127.0.0.1]:56585->[127.0.0.1]:162
DISMAN-EVENT-MIB::sysUpTimeInstance = 2538834
SNMPv2-MIB::snmpTrapOID.0 = IF-MIB::linkUp.0
```

Perl を使用した例:

```
2024-01-30T11:42:54+0200 ZBXTRAP 127.0.0.1
PDU INFO:
  receivedfrom                   UDP: [127.0.0.1]:58649->[127.0.0.1]:162
  notificationtype               TRAP
  version                        1
  community                      public
  errorstatus                    0
  transactionid                  1
  requestid                      2101882550
  messageid                      0
```

```
  errorindex                      0
VARBINDS:
  DISMAN-EVENT-MIB::sysUpTimeInstance type=67 value=Timeticks: (457671) 1:16:16.71
  SNMPv2-MIB::snmpTrapOID.0       type=6  value=OID: IF-MIB::linkUp.0
```

参照

- Zabbix blog article on SNMP traps
- Configuring snmptrapd (net-snmp の公式ドキュメント)
- Configuring snmptrapd to receive SNMPv3 notifications (net-snmp の公式ドキュメント)

4 IPMI チェック

概要

Zabbix では、Intelligent Platform Management Interface（IPMI）デバイスの健全性と可用性を監視できます。IPMI チェックを実行するには、事前に Zabbix サーバーを IPMI サポート付きで設定しておく必要があります。

IPMI は、コンピューターシステムのリモート「lights-out」管理または「out-of-band」管理のための標準化されたインターフェースです。これにより、オペレーティングシステムやマシンの電源状態に関係なく、いわゆる「out-of-band」管理カードからハードウェアの状態を直接監視できます。

Zabbix の IPMI 監視は、IPMI をサポートするデバイス（HP iLO、DELL DRAC、IBM RSA、Sun SSP など）でのみ機能します。

IPMI マネージャープロセスは、IPMI ポーラーによる IPMI チェックをスケジュールします。
1 台のホストは、常に同時に 1 つの IPMI ポーラーによってのみポーリングされるため、BMC コントローラーへのオープン接続数が削減されます。したがって、BMC コントローラーの過負荷を心配することなく、IPMI ポーラーの数を安全に増やすことができます。少なくとも 1 つの IPMI ポーラーが起動されると、IPMI マネージャープロセスは自動的に起動します。

IPMI チェックについては、既知の問題も参照してください。

設定

ホストの設定

IPMI チェックを処理するために、ホストを設定する必要があります。
 IPMI インターフェイスをを追加し、それぞれの IP アドレスおよびポート番号を指定し、IPMI 認証パラメーターを定義する必要があります。

詳細は、configuration of hosts を参照してください。

Server の設定

デフォルトでは、Zabbix サーバは IPMI ポーラーを起動するように設定されていないため、IPMI アイテムを追加しても動作しません。
 これを変更するには、Zabbix サーバ設定ファイル (zabbix_server.conf) を root ユーザで開き、
 次の行を探します。

```
# StartIPMIPollers=0
```

この行をアンコメントし、ポーラー数を 3 に設定すると、次のようになります。

```
StartIPMIPollers=3
```

ファイルを保存し、その後 zabbix_server を再起動します。

item の設定

ホストレベルでconfiguring an itemを行う場合:

- *Type* として'IPMI エージェント' を選択します。
- ホスト内で一意である item key (例えば、ipmi.fan.rpm) を入力します。
- *Host interface* には、関連する IPMI インターフェース (IP アドレスおよびポート番号) を選択します。IPMI インターフェースはホスト上に存在する必要があることに注意してください。
- IPMI センサー * (Dell Poweredge の'FAN MOD 1A RPM' など) からメトリックを取得するように指定します。デフォルトでは、センサー ID を指定する必要があります。接頭辞を使用することも可能です。
  - `id:` - センサーの ID を指定します。
  - `name:` - センサーのフルネームを指定します。これは、フルネームを指定することによってのみセンサを区別できる場合に役立ちます。
- それぞれの情報の種類を選択します (この場合は、'Numeric (float)' ディスクリートセンサーの場合は'Numeric (unsigned)')、単位（多くの場合'rpm'）、およびその他の必要な項目属性を選択します。

サポートされているチェック

IPMI エージェントは、IPMI センサー関連の情報を返す組み込みアイテム **ipmi.get** をサポートしており、IPMI センサーのディスカバリ に使用できます。
戻り値: *JSON object*

タイムアウトとセッション終了

IPMI メッセージのタイムアウトおよび再試行回数は、OpenIPMI ライブラリで定義されています。OpenIPMI の現在の設計上、これらの値を Zabbix で設定可能にすることはできず、インターフェースレベルでもアイテムレベルでも設定できません。

LAN 向けの IPMI セッション非アクティブタイムアウトは 60 +/-3 秒です。現在、OpenIPMI では Activate Session コマンドを定期的に送信する仕組みを実装できません。特定の BMC に対して、BMC で設定されたセッションタイムアウトより長い間、Zabbix からの IPMI アイテムチェックが行われなかった場合、タイムアウト経過後の次回の IPMI チェックは、個々のメッセージタイムアウト、再試行、または受信エラーによりタイムアウトします。その後、新しいセッションが開かれ、BMC の完全な再スキャンが開始されます。新しいセッションを管理するために、新しい UDP ポートが開かれる場合があります。非アクティブ状態とは、送信要求と受信応答の両方が存在しない状態として定義されます。BMC の不要な再スキャンを避けたい場合は、IPMI アイテムのポーリング間隔を、BMC で設定された IPMI セッション非アクティブタイムアウトより短く設定することを推奨します。

IPMI ディスクリートセンサーに関する注意事項

ホスト上のセンサーを見つけるには、**DebugLevel=4** を有効にして Zabbix サーバーを起動します。数分待ってから、Zabbix サーバーのログファイルでセンサー検出レコードを確認します。

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:7 id:'CATERR' reading_type:
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'CPU Therm Trip' read
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'System Event Log' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'PhysicalSecurity' re
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'IPMI Watchdog' readi
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'Power Unit Stat' rea
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Ctrl %' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:16 id:'P1 Therm Margin' rea
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 2' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:13 id:'System Fan 3' readin
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'P1 Mem Margin' readi
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'Front Panel Temp' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:15 id:'Baseboard Temp' read
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +5.0V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +3.3V STBY' readi
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:9 id:'BB +3.3V' reading_typ
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.5V P1 DDR3' re
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:17 id:'BB +1.1V P1 Vccp' re
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0 id_sz:14 id:'BB +1.05V PCH' readi
```

IPMI センサーのタイプと状態をデコードするために、IPMI 2.0 specifications のコピーを利用できます（IPMI 仕様については、今後これ以上の更新は予定されていない 点に注意してください）。

最初に確認すべきパラメータは”reading_type” です。仕様書の”Table 42-1, Event/Reading Type Code Ranges” を使用して”reading_type” コードをデコードします。今回の例にあるセンサーの多くは”reading_type:0x1” で、これは”threshold” センサーを意味します。”Table 42-3, Sensor Type Codes” では、”type:0x1” は温度センサー、”type:0x2” は電圧センサー、”type:0x4” はファンなどを意味することが示されています。しきい値センサーは、温度、電圧、毎分回転数のような連続的なパラメータを測定するため、”analog” センサーと呼ばれることもあります。

別の例として、”reading_type:0x3” のセンサーがあります。”Table 42-1, Event/Reading Type Code Ranges” によると、読み取りタイプコード 02h-0Ch は”Generic Discrete” センサーを意味します。ディスクリートセンサーは最大 15 個の可能な状態（言い換えると、最大 15 個の意味のあるビット）を持ちます。たとえば、’CATERR’ センサーの”type:0x7” については、”Table 42-3, Sensor Type Codes” でこのタイプが”Processor” を意味し、各ビットの意味は次のとおりであることが示されています: 00h（最下位ビット）- IERR、01h - Thermal Trip などです。

今回の例には”reading_type:0x6f” のセンサーもいくつかあります。これらのセンサーについては、”Table 42-1, Event/Reading Type Code Ranges” で、ビットの意味をデコードするために”Table 42-3, Sensor Type Codes” を使用するよう案内されています。たとえば、’Power Unit Stat’ センサーは”type:0x9” で、これは”Power Unit” を意味します。オフセット 00h は”PowerOff/Power Down” を意味します。言い換えると、最下位ビットが 1 の場合、サーバーの電源はオフです。このビットをテストするには、マスク’1’ を指定した`bitand` 関数を使用できます。トリガー式は次のようになります。

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

これは、サーバーの電源オフを警告するためのものです。

OpenIPMI-2.0.16, 2.0.17, 2.0.18, 2.0.19 のディスクリートセンサーの名前に関する注意事項

OpenIPMI-2.0.16, 2.0.17, 2.0.18 のディスクリートセンサーの名前には”0” (または他の数字や文字) を
 末尾に付加していることがよくあります。
 例えば、ipmitool と OpenIPMI-2.0.19 では、センサーの名前は”PhysicalSecurity” や”CATERR” のように表示されます。
 OpenIPMI-2.0.16, 2.0.17, 2.0.18 では”PhysicalSecurity0” または”CATERR0” と表示されます。

OpenIPMI-2.0.16、2.0.17、2.0.18 を使用して Zabbix サーバで IPMI item を設定する場合、
 IPMI エージェントアイテムの *IPMI sensor* フィールドにこれらの名前を 0 として使用してください。
 Zabbix サーバを OpenIPMI-2.0.19(またはそれ以降) を使用する新しい Linux ディストリビューションにアップグレードした場合、
 これらの IPMI センサーのアイテムは「NOT SUPPORTED」になります。そのため
 IPMI センサー名を変更し（末尾の’0’ を削除)、再び”Enabled ” になる前に、しばらく待つ必要があります。

しきい値センサーとディスクリートセンサーが同時に利用可能な場合の注意事項

一部の IPMI エージェントでは、同じ名前でしきい値センサーとディスクリートセンサーの両方が提供されます。
常にしきい値センサーが優先されます。

接続終了時の注意事項

IPMI チェックが行われなかった場合 (何らかの理由: ホストの IPMI 項目が全て無効/サポートされていない、
 ホストが無効/削除された、ホストがメンテナンス中であるなど)
 Zabbix server / proxy を起動した時間に応じて 3～4 時間後に Zabbix server / proxy から IPMI 接続が切断されます。

5 シンプルチェック

概要

シンプルチェックは通常、サービスのリモートでのエージェントレス監視に使用されます。

シンプルチェックには Zabbix エージェントは不要であることに注意してください。シンプルチェックの処理（外部接続の実行など）は、Zabbix サーバー/プロキシが担当します。

シンプルチェックの使用例:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
シンプルチェックのアイテム設定にある *User name* フィールドと *Password* フィールド（255 文字まで）は、VMware 監視アイテムで使用され、それ以外では無視されます。

サポートされているチェック

アイテムキーは、オプションのパラメータおよび追加情報を省略して記載されています。アイテムキーをクリックすると、詳細を確認できます。

VMware 監視アイテムキーも参照してください。

Item key	説明
icmpping	ICMP ping によるホストの到達可能性。
icmppingloss	失われたパケットの割合。
icmppingretry	再試行ありの ICMP ping によるホストの到達可能性。
icmppingsec	ICMP ping の応答時間。
net.tcp.service	サービスが稼働しており、TCP 接続を受け付けているかをチェックします。
net.tcp.service.perf	TCP サービスのパフォーマンスをチェックします。
net.udp.service	サービスが稼働しており、UDP リクエストに応答しているかをチェックします。
net.udp.service.perf	UDP サービスのパフォーマンスをチェックします。

アイテムキーの詳細

山かっこなしのパラメータは必須です。山かっこ **< >** で示されたパラメータは任意です。

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 ICMP ping によるホストの到達可能性。
 戻り値: *0* - ICMP ping 失敗; *1* - ICMP ping 成功。

パラメータ:

- **target** - ホストの IP または DNS 名;

- **packets** - パケット数;
- **interval** - 連続するパケット間の時間（ミリ秒）;
- **size** - パケットサイズ（バイト）;
- **timeout** - タイムアウト（ミリ秒）;
- **options** - リダイレクトを許可するために使用します: 空の場合（デフォルト値）、リダイレクトされた応答はターゲットホストがダウンしているものとして扱われます; *allow_redirect* に設定した場合、リダイレクトされた応答はターゲットホストが稼働しているものとして扱われます。

デフォルト値の表も参照してください。

例:

`icmpping[,4] #4つのパケットのうち少なくとも1つが返された場合、このアイテムは1を返します。`

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 失われたパケットの割合。
 戻り値: *Float*。

パラメータ:

- **target** - ホストの IP または DNS 名;
- **packets** - パケット数;
- **interval** - 連続するパケット間の時間（ミリ秒）;
- **size** - パケットサイズ（バイト）;
- **timeout** - タイムアウト（ミリ秒）;
- **options** - リダイレクトを許可するために使用します: 空の場合（デフォルト値）、リダイレクトされた応答はターゲットホストのダウンとして扱われます; *allow_redirect* に設定した場合、リダイレクトされた応答はターゲットホストの稼働として扱われます。

デフォルト値の表も参照してください。

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

 再試行付きの ICMP ping によるホストの到達可能性。最初のパケットが成功した場合は停止し、パケットが失敗した場合は、`retries` パラメータで定義された最大試行回数に達するまで再試行します。このアイテムは、ネットワーク上で送信されるパケット数を減らすのに役立ちます。
 戻り値: *0* - ICMP ping 失敗; *1* - ICMP ping 成功。

パラメータ:

- **target** - ホストの IP または DNS 名;
- **retries** - ターゲットへの ping 試行回数。最初の試行は含みません (0 以上、デフォルトは 1);
- **backoff** - 連続する各リクエストで待機時間に乗算される値 (範囲は 1.0～5.0、デフォルトは 1.0);
- **size** - パケットサイズ (バイト単位);
- **timeout** - タイムアウト (ミリ秒単位);
- **options** - リダイレクトを許可するために使用します。空の場合 (デフォルト値)、リダイレクトされた応答はターゲットホストのダウンとして扱われます。*allow_redirect* に設定した場合、リダイレクトされた応答はターゲットホストの稼働中として扱われます。

デフォルト値 の表も参照してください。

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 ICMP ping の応答時間（秒単位）。
 戻り値: *Float*。

パラメータ:

- **target** - ホストの IP アドレスまたは DNS 名;
- **packets** - パケット数;
- **interval** - 連続するパケット間の時間（ミリ秒）;
- **size** - パケットサイズ（バイト）;
- **timeout** - タイムアウト（ミリ秒）;
- **mode** - 指定可能な値: *min*、*max*、または *avg*（デフォルト）;
- **options** - リダイレクトを許可するために使用します: 空の場合（デフォルト値）、リダイレクトされた応答はターゲットホストがダウンしているものとして扱われます; *allow_redirect* に設定した場合、リダイレクトされた応答はターゲットホストが稼働しているものとして扱われます。

コメント:

- 損失した、またはタイムアウトしたパケットは計算に使用されません;
- ホストが利用不可の場合（タイムアウトに達した場合）、アイテムは 0 を返します;
- 戻り値が 0.0001 秒未満の場合、値は 0.0001 秒に設定されます;
- デフォルト値 の表も参照してください。

net.tcp.service[service,<ip>,<port>]

 サービスが実行中で、TCP 接続を受け付けているかをチェックします。
 戻り値: *0* - サービスは停止しています; *1* - サービスは実行中です。

パラメータ:

- **service** - 指定可能な値: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

コメント:

- *tcp* サービスでは、ポートの指定が必須であることに注意してください;
- これらのチェックにより、システムデーモンのログファイルに追加のメッセージが記録される場合があります (通常、SMTP および SSH セッションが記録されます);
- 暗号化プロトコル (ポート 993 の IMAP やポート 995 の POP など) のチェックは現在サポートされていません。回避策として、このようなチェックには `net.tcp.service[tcp,<ip>,port]` を使用してください。

例:

`net.tcp.service[ftp,,45]` #このアイテムは、TCPポート45上のFTPサーバーの可用性をテストするために使用できます

Attention:
SELinux が強制モードで動作している場合、カスタム TCP/UDP 簡易チェックはポリシーによってブロックされることがあります。新しい外向き接続を確認して許可するには、監査拒否を確認してください: `grep denied /var/log/audit/audit.log` を使用します

net.tcp.service.perf[service,<ip>,<port>]

 TCP サービスのパフォーマンスをチェックします。
 戻り値: *Float*: *0.000000* - サービスが停止しています。*seconds* - サービスへの接続に要した秒数。

パラメータ:

- **service** - 指定可能な値: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名（デフォルトではホストの IP/DNS が使用されます）;
- **port** - ポート番号（デフォルトでは標準のサービスポート番号が使用されます）。

コメント:

- *tcp* サービスでは、ポートの指定が必須であることに注意してください。
- 暗号化プロトコル（例: ポート 993 の IMAP やポート 995 の POP）のチェックは現在サポートされていません。回避策として、このようなチェックには `net.tcp.service[tcp,<ip>,port]` を使用してください。

例:

`net.tcp.service.perf[ssh]` #このアイテムは、SSHサーバーからの初期応答速度をテストするために使用できます。

net.udp.service[service,<ip>,<port>]

 サービスが実行中で、UDP リクエストに応答しているかどうかを確認します。
 戻り値: *0* - サービス停止; *1* - サービス稼働中。

パラメータ:

- **service** - 指定可能な値: *ntp* (詳細を参照);
- **ip** - IP アドレスまたは DNS 名 (デフォルトではホストの IP/DNS が使用されます);
- **port** - ポート番号 (デフォルトでは標準のサービスポート番号が使用されます)。

例:

`net.udp.service[ntp,,45]` #このアイテムは、UDPポート45でのNTPサービスの可用性をテストするために使用できます

net.udp.service.perf[service,<ip>,<port>]

 UDP サービスのパフォーマンスをチェックします。
 戻り値: *Float*: *0.000000* - サービスが停止しています。*seconds* - サービスからの応答を待機した秒数。

パラメータ:

- **service** - 指定可能な値: *ntp*（詳細を参照）;
- **ip** - IP アドレスまたは DNS 名（デフォルトではホストの IP/DNS が使用されます）;
- **port** - ポート番号（デフォルトでは標準のサービスポート番号が使用されます）。

例:

```
net.udp.service.perf[ntp] #このアイテムは、NTPサービスからの応答時間をテストするために使用できます。
```

Attention:
LDAP のシンプルチェック（例: `net.tcp.service[ldap]`）で SourceIP をサポートするには、OpenLDAP バージョン 2.6.1 以降が必要です。

タイムアウト処理

Zabbix は、アイテム設定フォームで定義された *Timeout* 秒を超えて、シンプルチェックを処理しません。VMware アイテム の場合、Zabbix は、Zabbix サーバー またはプロキシ の設定ファイルで定義された `Timeout` 秒を超えて、シンプルチェックを処理しません。icmpping* アイテムについては、タイムアウト値とリトライ値はアイテムキー内で直接指定され、グローバルな `Timeout` パラメータの影響を受けません。これらの値がアイテムキー内で適切に設定されていることを確認してください。icmpping* アイテムには、ハードコードされた最大タイムアウト値 600 秒が適用されることに注意してください（Zabbix 7.4.6 以降）。

ICMP ping

Zabbix は、ICMP ping（**icmpping**、**icmppingloss**、**icmppingretry**、**icmppingsec**）を処理するために、外部ユーティリティ **fping** を使用します。

インストール

fping は Zabbix には含まれておらず、別途インストールする必要があります。

- さまざまな Unix ベースのプラットフォームでは、デフォルトのリポジトリに fping パッケージが含まれていますが、あらかじめインストールされているわけではありません。この場合は、パッケージマネージャーを使用して fping をインストールできます。
- Zabbix は、RHEL およびその派生ディストリビューション向けに fping パッケージ を提供しています。これらのパッケージは、公式サポートなしで提供されている点にご注意ください。
- fping は、ソースから コンパイルすることもできます。

設定

fping の場所は、Zabbix サーバー/プロキシ設定ファイルの *FpingLocation* パラメータで指定します
(IPv6 アドレスを使用する場合は *Fping6Location* パラメータ)。

fping は、Zabbix サーバー/プロキシを実行しているユーザーが実行可能である必要があり、このユーザーには十分な権限が必要です。

参照: fping バージョン 3.10 未満でシンプルチェックを処理する場合は、既知の問題 も参照してください。

デフォルト値

ICMP チェックパラメータのデフォルト値、制限、および値の説明:

パラメータ	単位	説明	Fping のフラグ	デフォルト設定元		Zabbix で許可される制限	
				fping	**Zabbix**	最小	最大
packets	number	ターゲットに送信されるリクエストパケット数	-C		3	1	10000
interval	milliseconds	個々のターゲットに対して連続するパケット間で待機する時間	-p	1000		20	unlimited
size	bytes	バイト単位のパケットサイズ x86 では 56 バイト、x86_64 では 68 バイト	-b	56 または 68		24	65507

パラメータ	単位	説明	Fping のフラグ	デフォルト設定元		Zabbix で許可される制限	
timeout	milliseconds	**fping v3.x** - 最後のパケット送信後に待機するタイムアウト。*-C* フラグの影響を受けます **fping v4.x** - 各パケットごとの個別タイムアウト	-t	**fping v3.x** - 500 **fping v4.x** 以降 - *-p* フラグから継承されますが、2000 を超えません		50	unlimited
retries	number	最初の試行を含まない、ターゲットへの ping 試行回数	-r	3	1	0	unlimited
backoff factor	number	連続する各リクエストで待機時間に乗算される値	-B	1.5	1.0	1.0	5.0

デフォルト値は、プラットフォームやバージョンによって若干異なる場合があります。

さらに、Zabbix は fping オプション *-i interval ms*（上の表に記載されているアイテムパラメータ *interval* と混同しないでください。これは fping オプション *-p* に対応します）および *-S source IP address*（古い fping バージョンでは *-I*）を使用します。これらのオプションは、異なるオプションの組み合わせでチェックを実行することで自動検出されます。Zabbix は、*-i* とともに fping が使用を許可する最小のミリ秒値を、0、1、10 の 3 つの値を試すことで検出しようとします。最初に成功した値が、その後の ICMP チェックに使用されます。この処理は、各ICMP pingerプロセスごとに個別に実行されます。

自動検出された fping オプションは 1 時間ごとに無効化され、次回 ICMP チェックを実行しようとした際に再度検出されます。この処理の詳細をサーバーまたはプロキシのログファイルで確認するには、DebugLevel>=4 を設定してください。

Zabbix は、*icmpping** キーのいずれかでチェックされる IP アドレスを一時ファイルに書き込み、そのファイルを fping に渡します。アイテムのキーパラメータが異なる場合、同一のキーパラメータを持つものだけが 1 つのファイルに書き込まれます。1 つのファイルに書き込まれたすべての IP アドレスは fping によって並列にチェックされるため、Zabbix の ICMP pinger プロセスは、ファイル内の IP アドレス数に関係なく一定の時間で処理を行います。

1 VMware 監視アイテムキー

VMware 監視のアイテムキーの一覧は、VMware 監視セクションに移動しました。

6 ログファイル監視

概要

Zabbix は、ログローテーションサポートの有無にかかわらず、ログファイルの集中監視と分析に利用することができます。

ログファイルに特定の文字列や文字列パターンが含まれる場合、通知機能を使ってユーザに警告することができます。

ログファイルを監視するには、以下の環境が必要です。

- ホスト上で Zabbix agent が動作していること
- ログ監視 item の設定

Attention:
監視するログファイルのサイズ制限は、large file support に依存します。

設定

agent パラメーターの確認

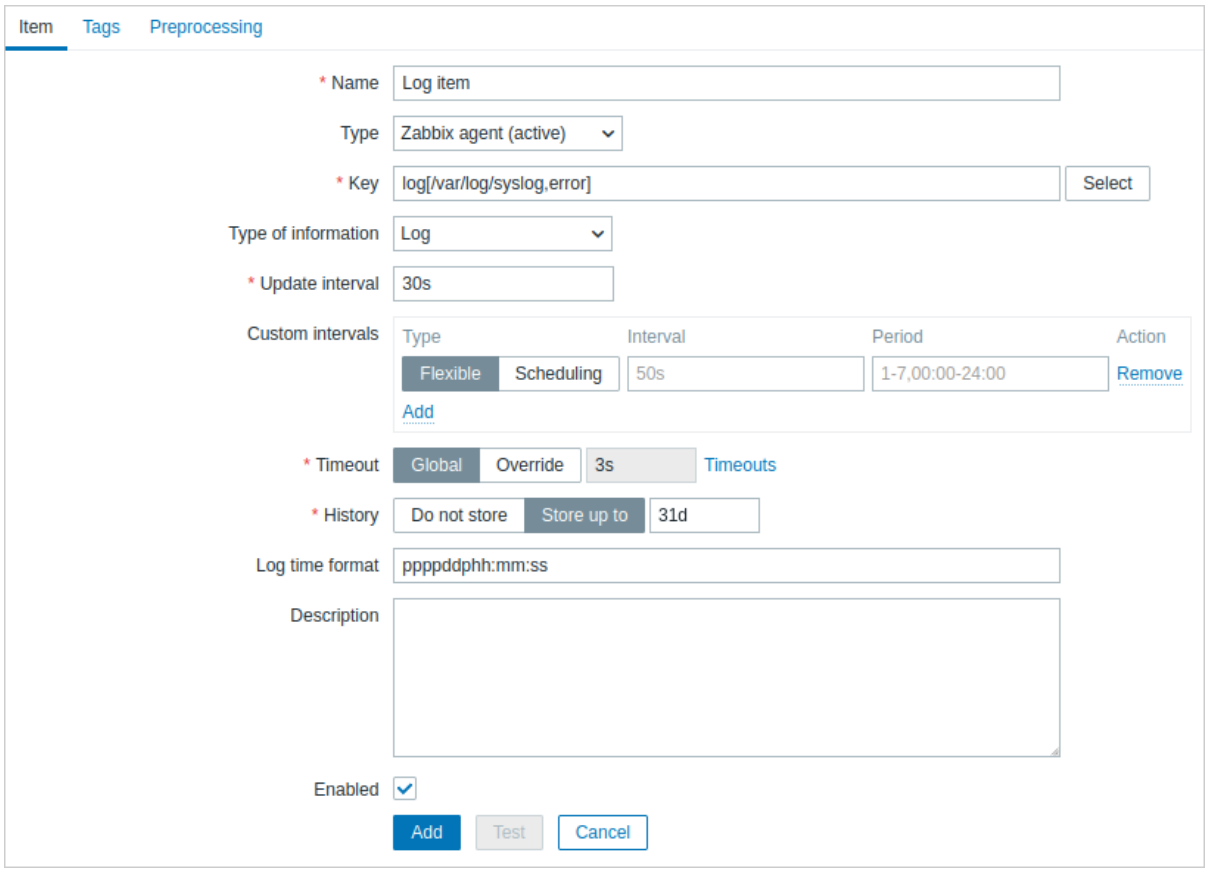
agent configuration fileで以下を確認します。

- 'Hostname' パラメータがフロントエンドのホスト名と一致していること。

- 'ServerActive' パラメータで、アクティブチェックの処理に使用する server が指定されていること。

アイテムの設定

ログ監視アイテムを設定します。



必須の入力フィールドには、赤いアスタリスクが付いています。

特にログ監視アイテムでは、以下を入力します。

Type	ここでは **Zabbix agent (active)** を選択します。
Key	次のいずれかのアイテムキーを使用します。 **log[]** または **logrt[]**: これら 2 つのアイテムキーでは、ログを監視し、regexp の内容によってログエントリをフィルタリングできます（指定されている場合）。 例: `log[/var/log/syslog,error]`。ファイルに'zabbix' ユーザーの読み取り権限があることを確認してください。そうでない場合、アイテムのステータスは'unsupported' に設定されます。 **log.count[]** または **logrt.count[]**: これら 2 つのアイテムキーでは、一致した行数のみを返すことができます。 これらのアイテムキーとそのパラメータの使用方法の詳細については、サポートされているZabbix agent item キーのセクションを参照してください。
Type of information	自動的に事前入力されます。 log[] または logrt[] アイテムの場合 - Log; log.count[] または logrt.count[] アイテムの場合 - `Numeric (unsigned)`。 必要に応じて `output` パラメータを使用する場合は、Log 以外の適切な情報の型を手動で選択できます。 なお、Log 以外の情報の型を選択すると、ローカルタイムスタンプは失われます。
Update interval (in sec)	このパラメータは、Zabbix agent がログファイルの変更をどのくらいの頻度で確認するかを定義します。1 秒に設定すると、新しいレコードをできるだけ早く取得できます。

Log time format	このフィールドでは、ログ行のタイムスタンプを解析するためのパターンを任意で指定できます。 サポートされるプレースホルダー: * **y**: 年 *(1970-2038)* * **M**: 月 *(01-12)* * **d**: 日 *(01-31)* * **h**: 時 *(00-23)* * **m**: 分 *(00-59)* * **s**: 秒 *(00-59)* 空欄のままにすると、タイムスタンプは Unix time で 0 に設定され、1970 年 1 月 1 日を表します。 例えば、Zabbix agent のログファイルに次の行があるとします。 ”23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” この行は、先頭に PID 用の 6 文字分の位置があり、その後に日付、時刻、メッセージの残りが続きます。 この行のログ時刻形式は”ppppppp:yyyyMMdd:hhmmss” になります。 なお、”p” および”:” 文字はプレースホルダーであり、”yMdhms” 以外の任意の文字を使用できます。

重要な注意事項

- サーバーとエージェントは、監視対象ログのサイズと最終更新時刻（logrt の場合）の追跡情報を、2 つのカウンターに保持します。さらに:
  - エージェントは、ログファイルが切り詰められたりローテーションされたりした際の判定精度を高めるために、inode 番号（UNIX/GNU/Linux）、ファイルインデックス（Microsoft Windows）、およびログファイル先頭 512 バイトの MD5 サムも内部的に使用します。
  - UNIX/GNU/Linux システムでは、ログファイルが保存されているファイルシステムが inode 番号を報告し、それをファイル追跡に利用できることを前提としています。
  - Microsoft Windows では、Zabbix エージェントはログファイルが存在するファイルシステムの種類を判定し、以下を使用します:
    * NTFS ファイルシステムでは 64 ビットのファイルインデックス。
    * ReFS ファイルシステムでは（Microsoft Windows Server 2012 以降のみ）128 ビットのファイル ID。
    * ファイルインデックスが変化するファイルシステム（例: FAT32、exFAT）では、ログファイルのローテーションにより最終更新時刻が同じ複数のログファイルが生じた場合、不確実な条件下でも妥当な処理を行うためのフォールバックアルゴリズムが使用されます。
  - inode 番号、ファイルインデックス、および MD5 サムは、Zabbix エージェントによって内部的に収集されます。これらは Zabbix サーバーには送信されず、Zabbix エージェントが停止すると失われます。
  - ログファイルの最終更新時刻を変更しないでください（たとえば `touch` を使用するなど）。また、監視対象のログファイルを、ファイルを元の名前にコピーして置き換えないでください（この操作では新しい inode が作成されます）。いずれの場合も、Zabbix はそのファイルを別のファイルとして扱い、先頭から再読み込みする可能性があり、その結果、重複アラートが発生することがあります。
  - `logrt[]` アイテムに一致するログファイルが複数あり、Zabbix エージェントがその中で最新のものを追跡しているときに、その最新のログファイルが削除されると、警告メッセージ `"there are no files matching "<regexp mask>" in "<directory>"` が記録されます。Zabbix エージェントは、チェック対象の `logrt[]` アイテムについて、エージェントが確認済みの最新の最終更新時刻より古い更新時刻を持つログファイルを無視します。
- エージェントは、前回停止した位置からログファイルの読み取りを開始します。
- すでに解析済みのバイト数（サイズカウンター）と最終更新時刻（時刻カウンター）は Zabbix データベースに保存され、エージェントに送信されます。これにより、エージェントが起動した直後や、以前は無効または未サポートだったアイテムを受信した場合でも、その位置からログファイルの読み取りを開始できます。ただし、エージェントがサーバーから 0 以外のサイズカウンターを受信していても、logrt[] または logrt.count[] アイテムが一致するファイルを見つけられない場合、後でファイルが現れた際に先頭から解析できるよう、サイズカウンターは 0 にリセットされます。
- ログファイルのサイズが、エージェントが認識しているログサイズカウンターより小さくなった場合は常に、そのカウンターは 0 にリセットされ、エージェントは時刻カウンターを考慮しつつログファイルの先頭から読み取りを開始します。
- ディレクトリ内に最終更新時刻が同じ一致ファイルが複数ある場合、エージェントは、データの取りこぼしや同じデータの二重解析を避けつつ、同じ更新時刻を持つすべてのログファイルを正しく解析しようとしますが、すべての状況で保証されるわけではありません。エージェントは、特定のログファイルローテーション方式を前提とせず、それを判定もしません。最終更新時刻が同じ複数のログファイルがある場合、エージェントはそれらを辞書順の降順で処理します。したがって、ローテーション方式によっては、ログファイルは元の順序どおりに解析および報告されます。一方、別のローテーション方式では元のログファイル順序が維持されず、一致したログファイルレコードが順序変更された状態で報告される可能性があります（この問題は、ログファイルの最終更新時刻が異なる場合には発生しません）。
- Zabbix エージェントは、ログファイルの新しいレコードを *Update interval* 秒ごとに処理します。
- Zabbix エージェントは、1 秒あたりログファイルから **maxlines** を超えて送信しません。この制限は、ネットワークおよび CPU リソースの過負荷を防ぐためのもので、agent configuration file の **MaxLinesPerSecond** パラメータで指定されたデフォルト値よりも **maxlines** が優先されます。

- 必要な文字列を見つけるために、Zabbix は MaxLinesPerSecond に設定された値の 10 倍の新規行を処理します。したがって、たとえば log[] または logrt[] アイテムの *Update interval* が 1 秒の場合、デフォルトではエージェントは 1 回のチェックで最大 200 件のログファイルレコードを解析し、そのうち最大 20 件の一致レコードを Zabbix サーバーに送信します。エージェント設定ファイルで **MaxLinesPerSecond** を増やすか、アイテムキーで **maxlines** パラメータを設定することで、この上限は 1 回のチェックあたり最大 10000 件のログファイルレコードの解析、および 1000 件の一致レコードの Zabbix サーバーへの送信まで引き上げることができます。*Update interval* が 2 秒に設定されている場合、1 回のチェックあたりの上限は *Update interval* が 1 秒の場合の 2 倍になります。
- さらに、log および log.count の値は、エージェント送信バッファサイズの 50% に常に制限されます。たとえそこに非ログ値が存在しない場合でも同様です。したがって、**maxlines** の値を 1 回の接続で送信するには（複数回の接続ではなく）、エージェントのBufferSize パラメータは少なくとも maxlines x 2 である必要があります。Zabbix エージェントはログ収集中にデータをアップロードしてバッファを解放できますが、Zabbix agent 2 はデータがアップロードされてバッファが解放されるまでログ収集を停止します。この処理は非同期で実行されます。
- ログアイテムが存在しない場合、エージェントのバッファサイズ全体が非ログ値に使用されます。ログ値が入ってくると、それらは必要に応じて古い非ログ値を置き換え、指定された 50% まで使用します。
- 256kB を超えるログファイルレコードについては、先頭の 256kB のみが正規表現との照合対象となり、残りのレコードは無視されます。ただし、Zabbix エージェントが長いレコードを処理中に停止すると、エージェント内部状態が失われるため、再起動後にその長いレコードが再度、しかも異なる形で解析される可能性があります。
- "\" パス区切り文字に関する特記事項: file_format が"file\.log" の場合、"file" ディレクトリが存在してはいけません。これは、"." がエスケープされているのか、それともファイル名の先頭文字なのかを曖昧さなく定義できないためです。
- logrt の正規表現はファイル名に対してのみサポートされており、ディレクトリに対する正規表現マッチングはサポートされていません。
- UNIX プラットフォームでは、ログファイルが存在するはずのディレクトリが存在しない場合、logrt[] アイテムは NOTSUPPORTED になります。
- Microsoft Windows では、ディレクトリが存在しなくてもアイテムは NOTSUPPORTED になりません（たとえば、アイテムキー内のディレクトリ名のスペルミスがある場合など）。
- logrt[] アイテムでログファイルが存在しないこと自体は、NOTSUPPORTED の原因にはなりません。logrt[] アイテムのログファイル読み取りエラーは、Zabbix エージェントのログファイルに警告として記録されますが、アイテムは NOTSUPPORTED にはなりません。
- log[] または logrt[] アイテムが NOTSUPPORTED になった理由を調べるには、Zabbix エージェントのログファイルが役立つ場合があります。Zabbix は自身のエージェントログファイルを監視できますが、DebugLevel=4 または DebugLevel=5 の場合を除きます。
- 正規表現を使用して疑問符を検索する場合、たとえば \? では、テキストファイルに NUL 文字が含まれていると誤検知が発生する可能性があります。これは、Zabbix が改行文字まで行の処理を継続するために、それらを"?" に置き換えるためです。

正規表現の一致部分の抽出

正規表現の一致が見つかった際に、行全体を返すのではなく、対象ファイルから必要な値だけを抽出したい場合があります。

ログアイテムには、一致した行から必要な値を抽出する機能があります。
これは、log および logrt アイテムの追加の **output** パラメータによって実現されます。

output パラメータを使用すると、一致結果の中で必要な「キャプチャグループ」を指定できます。

たとえば、次のようにします。

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

これにより、次の内容からエントリ数を返すことができます。

```
Fr Feb 07 2014 11:07:36.6690 */ Thread Id 1400 (GLEWF) large result
buffer allocation - /Length: 437136/Entries: 5948/Client Ver: >=10/RPC
ID: 41726453/User: AUser/Form: CFG:ServiceLevelAgreement
```

返されるのは数値のみです。これは、**\1** が最初で唯一のキャプチャグループ **([0-9]+)** を参照しているためです。

また、数値を抽出して返せるため、その値を使用してトリガーを定義できます。

maxdelay パラメータの使用

ログ項目の'maxdelay' パラメーターを使用すると、ログファイルから古い行を無視して、'maxdelay' 秒以内に分析された最新の行を取得できます。

> **Warning:**
> maxdelay' > 0 を指定すると、重要なログファイルの記録が無視されたり、警告が見逃されたりすることがあります。必要なときだけ、自己責任で慎重に使用してください。

デフォルトでは、ログ監視のための項目は、ログファイルに現れるすべての新しい行に従います。しかし、ある状況下でログファイルに膨大な数のメッセージを書き始めるアプリケーションがあります。例えば、データベースや DNS サーバーが利用できない場合、

そのようなアプリケーションは、正常な動作が回復するまで、ほぼ同じエラーメッセージを何千もログファイルに書き出します。デフォルトでは、これらのメッセージはすべて忠実に解析され、log と logrt の項目で設定されたとおりにマッチする行がサーバに送信されます。

過負荷に対する組み込みの保護は、構成可能な'maxlines' パラメーター (一致するログ行が多すぎる受信からサーバーを保護する) と 10*'maxlines' 制限 (1 回のチェックでエージェントによる過負荷からホスト CPU と I/O を保護する) で構成されます。それでも、組み込みの保護機能には 2 つの問題があります。まず、あまり有益ではない可能性のある多数のメッセージがサーバーに報告され、データベース内のスペースが消費されます。第 2 に、1 秒あたりに分析される行数が限られているため、エージェントは最新のログレコードから数時間遅れる可能性があります。多くの場合、古いレコードを何時間もクロールするよりも、ログファイルの現在の状況についてすぐに知りたいと思うかもしれません。

両方の問題の解決策は、'maxdelay' パラメーターを使用することです。'maxdelay' > 0 が指定されている場合、各チェック中に、処理されたバイト数、残りのバイト数、および処理時間が測定されます。これらの数値から、エージェントは推定遅延 (ログファイル内の残りのすべてのレコードを分析するのにかかる秒数) を計算します。

遅延が'maxdelay' を超えない場合、エージェントは通常どおりログファイルの分析を続行します。

遅延が'maxdelay' よりも大きい場合、エージェントは'maxdelay' 秒以内に残りの行を分析できるように、ログファイルのチャンクを無視して新しい推定位置に**"** ジャンプ**"** します。

エージェントは無視された行をバッファに読み込むことさえせず、ファイル内のジャンプ先のおおよその位置を計算することに注意してください。

ログファイルの行をスキップしたという事実は、次のようにエージェントログファイルに記録されます。

```
14287:20160602:174344.206 item:"logrt["/home/zabbix32/test[0-9].log",ERROR,,1000,,,120.0]" logfile:"/home/
```

" ジャンプ" の後、エージェントはファイル内の位置をログ行の先頭に調整するため、" バイトまで" の数値は概算です。

成長速度とログファイルの分析速度を比較する方法によっては、すべてのチェックで" ジャンプ" が見られない場合、ほとんどまたは頻繁に" ジャンプ" が見られる場合、大きなまたは小さな" ジャンプ" が見られる場合、または小さな" ジャンプ" が見られる場合さえあります。システム負荷とネットワーク遅延の変動も遅延の計算に影響を与えるため、"maxdelay" パラメータに追いつくために先に" ジャンプ" します。

'maxdelay' < 'update interval' を設定することはお勧めしません (小さな" ジャンプ" が頻繁に発生する可能性があります)

'copytruncate' ログファイルローテーションの取り扱いに関する注意点

logrt の copytruncate オプションは、異なるログファイルには異なる記録がある (少なくともタイムスタンプは異なる) と
 仮定しています。したがって、初期ブロック (最初の 512 バイトまで) の MD5 チェックサムは異なることになります。
 初期ブロックの MD5 チェックサムが同じである 2 つのファイルは、一方がオリジナルで、もう一方がコピーであることを意味します。

logrt の copytruncate オプションは、ログファイルのコピーを重複して報告することなく、正しく処理するように努力します。
 しかし、同じタイムスタンプのログファイルを複数作成したり、ログファイルのローテーションが logrt[] item の update interval より
 頻繁になったり、agent を頻繁に再起動することは推奨されません。agent は、これらの状況を合理的に処理しようとしますが、
 すべての状況でよい結果を保証することはできません。

log*[] item の persistent files に関する注意事項

永続化ファイルの目的

Zabbix agent の起動時に、Zabbix server または proxy からアクティブチェックのリストを受け取ります。
 log*[] メトリクスでは、ログファイルの監視を開始する場所を見つけるために、処理されたログのサイズと変更時間を受け取ります。
 ファイルシステムから報告された実際のログファイルサイズと変更時間に応じて、agent は処理されたログサイズから
 ログファイル監視を継続するか、ログファイルを最初から再分析するかを決定します。

実行中の agent は、チェックの間に監視されたすべてのログファイルを追跡するために、
 より大きな属性のセットを維持します。このインメモリの状態は、エージェントが停止されると失われます。

新しいオプションパラメータ **persistent_dir** は、この log[]、log.count[]、logrt[] または logrt.count[] item の状態を
 ファイルに格納するディレクトリを指定するものです。Zabbix agent を再起動すると、ログアイテムの状態は永続化ファイルから復元されます。

主な用途は、ミラーリングされたファイルシステム上にあるログファイルの監視です。ある時点まで、ログファイルは
 両方のミラーに書き込まれます。その後、ミラーは分割されます。アクティブなコピーでは、ログファイルは新しいレコードを
 取得し、まだ成長しています。Zabbix agent はこれを解析し、処理されたログのサイズと更新時刻を server に送信します。
 パッシブコピーでは、ログファイルは同じまま、アクティブコピーよりかなり遅れています。その後、OS と Zabbix エージェントが
 パッシブコピーからリブートされます。Zabbix agent が server から受け取った処理済みのログサイズと更新時刻は、
 パッシブコピー上の状況に対して有効でない場合があります。ファイルシステムのミラースプリット時に agent が停止した場所から
 ログファイルの監視を継続するために、agent は永続ファイルから状態をリストアします。

永続ファイルを使用した agent 操作

起動時、Zabbix agent は永続化ファイルについて何も知りません。Zabbix サーバからアクティブチェックのリストを受け取った後、
 agent はいくつかのログアイテムが指定されたディレクトリの永続ファイルでバックアップされることを認識します。

agent 操作の間、永続ファイルは書き込みのために開かれ (fopen(filename, ”w”)) 、最新のデータで上書きされます。
 上書きとファイルシステムのミラーリングが同時に起こった場合、永続ファイルのデータが失われる可能性は非常に低いので、
 特別な処理は行いません。永続ファイルへの書き込みの後にメディアへの強制同期を行わない (fsync() is not called) は呼び出されない) ため、
 永続ファイルへの書き込みの後にメディアへの強制同期を行うことはできません。

最新データの上書きは、一致するログファイルのレコードまたはメタデータ (処理されたログのサイズと更新時間) が Zabbix server に正常に報告された後に行われます。ログファイルが変化し続ける場合、item チェックのたびに行われる可能性があります。

agent シャットダウン中の特別な動作はありません。

アクティブチェックのリストを受信した後、エージェントは不要になった永続ファイルを削除対象としてマークします。
永続ファイルが不要になるのは、次の場合です:

1. 対応するログアイテムが監視対象ではなくなった場合。
2. ログアイテムが、以前とは異なる **persistent_dir** の場所で再設定された場合。

削除は 24 時間遅延して実行されます。これは、NOTSUPPORTED 状態のログファイルはアクティブチェックのリストに含まれませんが、後で SUPPORTED になり、その永続ファイルが有用になる可能性があるためです。

24 時間が経過する前にエージェントが停止した場合、不要になったファイルは削除されません。これは、Zabbix エージェントがそれらの場所に関する情報を、もはや Zabbix サーバーから取得しないためです。

> **Warning:**
> ユーザーが古い永続ファイルを削除しないまま、エージェント停止中にログアイテムの **persistent_dir** を元の **persistent_dir** の場所に再設定すると、古い永続ファイルからエージェントの状態が復元され、メッセージの取りこぼしや誤ったアラートの原因になります。

永続ファイルの命名と場所

Zabbix agent は、アクティブなチェックをキーで区別します。例えば、logrt[/home/zabbix/test.log] と <br> logrt[/home/zabbix/test.log,] は異なるアイテムです。フロントエンドの item logrt[/home/zabbix/test.log„,10] を <br> logrt[/home/zabbix/test.log„,20] に変更すると、item logrt[/home/zabbix/test.log„,10] が削除されます。
 logrt[/home/zabbix/test.log„10] を agent のアクティブチェックリストから削除し、logrt[/home/zabbix/test.log„20] item を作成します。
 （いくつかの属性は、agent ではなくフロントエンド/サーバで変更されます）

ファイル名は、item キーの MD5 チェックサムと、衝突の可能性を低減するために item キーの長さを付加したものです。
 例えば、logrt[/home/zabbix50/test.log„„„/home/zabbix50/agent_private] item の状態は、
 永続ファイル c963ade4008054813bbc0a650bb8e09266 に保存されます。

persistent_dir は、複数のログ item で同じ値を使用できます。

persistent_dir は、特定のファイルシステムのレイアウト、マウントポイント、マウントオプション、ストレージのミラーリング設定を
 考慮して指定されます。

persistent_dir ディレクトリが作成できないか存在しないか、Zabbix agent のアクセス権でファイルの作成/書き込み/読み取り/削除が
 できない場合、ログ item は NOTSUPPORTED になります。

agent 操作中に永続的ストレージファイルへのアクセス権が削除されたり、他のエラー (例：ディスクフル) が発生した場合、エラーは agent ログファイルに記録されますが、ログ item が NOTSUPPORTED になることはありません。

I/O の負荷

Item の永続化ファイルは、item のデータを含むデータを server に送信するたびに更新されます。例えば、デフォルトの‘BufferSize’は 100 です。
 ログアイテムが 70 件のマッチングレコードを発見した場合、最初の 50 件のレコードを一括して送信し、永続化ファイルを更新した後、
 残りの 20 レコードが 2 回目のバッチで送信され、永続化ファイルが再び更新されます。

agent と server 間の通信に失敗した場合の対処法

log[] と logrt[]`item` からマッチする各行と、各 log.count[] と logrt.count[]‘ item チェックの結果は、
 agent 送信バッファ内の指定された 50% エリアに空きスロットを必要とします。バッファの要素は定期的に server （または proxy）に送信され、
 バッファのスロットは再び空きます。

agent 送信バッファの指定ログ領域に空き枠があり、agent と server （または proxy）間の通信に失敗した場合、
 ログ監視結果が送信バッファに蓄積されます。これにより、短時間の通信障害を緩和することができます。

長時間の通信障害時には、すべてのログスロットが占有され、次のような動作が行われます。

- log[] と logrt[]‘ の item チェックは停止しています。通信が復旧し、バッファの空きスロットができると、チェックは前の位置から再開されます。マッチする行は失われず、後で報告されるだけです。

- maxdelay = 0（デフォルト）の場合、log.count[] と logrt.count[] のチェックが停止します。動作は上記の log[] と logrt[] の項目と同様です。これは log.count[] と logrt.count[]' の結果に影響を与えることに注意してください。例えば、あるチェックでログファイルの 100 行にマッチしましたが、バッファに空きスロットがないため、チェックは停止されました。通信が回復したとき、agent は同じ 100 のマッチする行と、70 の新しいマッチする行を数えます。agent は、あたかも一度のチェックで見つかったかのように、count = 170 を送信します。
- `maxdelay > 0` の `log.count[]` と `logrt.count[]` のチェック: チェック中に"jump" がなければ、上記の動作と同じになります。もし、ログファイルの行を"jump" した場合は、"jump" 後の位置を保持し、カウントした結果は破棄されます。したがって、agent は、通信障害が発生しても、ログファイルの増加についていこうとします。

正規表現のコンパイルエラーおよび実行時エラーの処理

`log[]`、`logrt[]`、`log.count[]`、または `logrt.count[]` アイテムで使用される正規表現を PCRE または PCRE2 ライブラリでコンパイルできない場合、そのアイテムはエラーメッセージとともに NOTSUPPORTED 状態になります。ログアイテムの監視を継続するには、正規表現を修正する必要があります。

正規表現のコンパイルに成功しても、実行時に失敗した場合（ログレコードの一部またはすべて）、そのログアイテムはサポートされた状態のままとなり、監視は継続されます。実行時エラーは、Zabbix エージェントのログファイルに記録されます（ログファイルのレコード自体は記録されません）。

Zabbix エージェントが自身のログファイルを監視できるようにするため、ログ出力率はチェックごとに実行時エラー 1 件に制限されています。たとえば、10 件のレコードが解析され、そのうち 3 件が正規表現の実行時エラーで失敗した場合、エージェントログには 1 件のレコードが出力されます。

例外: `MaxLinesPerSecond=1` かつ update interval=1 の場合（チェックごとに解析できるレコードが 1 件のみ許可される場合）、正規表現の実行時エラーはログに記録されません。

zabbix_agentd は実行時エラー発生時にアイテムキーをログに記録し、zabbix_agent2 はどのログアイテムで実行時エラーが発生しているかを特定しやすくするためにアイテム ID を記録します。実行時エラーが発生する場合は、正規表現を見直すことを推奨します。

7 計算

概要

計算アイテムでは、既存のいくつかのアイテムの値に基づいて計算を作成できます。たとえば、あるアイテム値の 1 時間ごとの平均を計算したり、アイテムのグループの合計値を計算したりできます。計算アイテムは、そのために使用します。

計算では、次の両方を使用できます。

- 個々のアイテムの単一の値
- 集計のために複数のアイテムを選択する複雑なフィルター（詳細は集計計算を参照）

計算アイテムは、仮想データソースを作成する方法の 1 つです。すべての計算は、Zabbix サーバーでのみ実行されます。値は、使用される算術式に基づいて定期的に計算されます。

結果のデータは、他のアイテムと同様に Zabbix データベースに保存されます。履歴値とトレンド値の両方が保存され、グラフを生成できます。

> **Note:**
> 計算結果が浮動小数点値であり、計算アイテムの情報の型が数値（符号なし）である場合、その値は整数に切り詰められます。また、キャッシュ内に新しいデータがなく、関数に問い合わせ期間が定義されていない場合、Zabbix はデフォルトで最大 1 週間前までさかのぼってデータベースに履歴値を問い合わせます。

計算アイテムは、トリガーの式と構文を共有します。計算アイテムでは文字列との比較も可能です。計算アイテムは、他の任意のアイテムタイプと同様に、マクロやその他のエンティティから参照できます。

計算アイテムを使用するには、アイテムタイプ **Calculated** を選択します。

設定可能なフィールド

キーは、アイテムの一意な識別子です（ホストごと）。サポートされている記号を使用して、任意のキー名を作成できます。

計算の定義は **Formula** フィールドに入力する必要があります。Formula とキーの間に関連はありません。キーのパラメータは、Formula では一切使用されません。

単純な式の構文は次のとおりです。

```
function(/host/key,<parameter1>,<parameter2>,...)
```

各要素の意味は次のとおりです。

function	サポートされている関数 のいずれか: last、min、max、avg、count など

host	計算に使用されるアイテムのホスト。 現在のホストは省略できます（つまり、function(//key,parameter,...) のように指定できます）。
key	計算に使用されるアイテムのキー。 バイナリのデータ型 の値を返すアイテムはサポートされていません。
parameter(s)	必要に応じた関数のパラメータ。 時間サフィックス および メモリサイズサフィックス がサポートされています。

Attention:
式内のユーザーマクロ は、関数パラメータ、アイテムフィルターパラメータ、または定数を参照するために使用された場合に展開されます。ユーザーマクロは、関数、ホスト名、アイテムキー、アイテムキーパラメータ、または演算子を参照している場合は展開されません。

より複雑な式では、関数、演算子、括弧を組み合わせて使用できます。トリガー式でサポートされているすべての関数と演算子 を使用できます。ロジックおよび演算子の優先順位はまったく同じです。

トリガー式とは異なり、Zabbix は新しい値を受信したときではなく、アイテムの更新間隔に従って計算アイテムを処理します。

計算アイテムの式内で履歴関数によって参照されるすべてのアイテムは、存在していてデータ収集を行っている必要があります。また、参照されているアイテムのキーを変更した場合は、そのキーを使用しているすべての式を手動で更新する必要があります。

計算アイテムは、いくつかのケースで未サポートになる場合があります。

- 参照されているアイテム
  - 見つからない
  - 無効になっている
  - 無効なホストに属している
  - 未サポートである（nodata() 関数および unknown 値を持つ演算子 の場合を除く）
- 関数を計算するためのデータがない
- ゼロによる除算
- 構文が正しくない

使用例

例 1

'/' のディスク空き容量のパーセンテージを計算します。

Use of function **last**:

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix はフリーディスクスペースとトータルディスクスペースの最新値を取得し、指定された計算式に従ってパーセンテージを計算します。

例 2

Zabbix で処理された値の 10 分間の平均値を計算します。

Use of function **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

長い時間の計算 item を多用すると、Zabbix server のパフォーマンスに影響を与える可能性があることに注意してください。

例 3

eth0 の総帯域幅を計算する。

2 つの関数の合計:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

例 4

受信トラフィックのパーセンテージを計算する。

より複雑な表現:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Examples of aggregate calculationsを参照してください。

1 集計計算

概要

集計計算は、Zabbix サーバーによって複数のアイテムから情報を収集し、使用する集計関数に応じて集計を計算できる計算アイテムタイプです。

集計計算では、監視対象のホストでエージェントを実行する必要はありません。

構文

集計値を取得するには、サポートされている集計関数のいずれか（avg、max、min、sum など）を使用します。次に、必要なアイテムを選択するために、**foreach** 関数を唯一のパラメータとして追加し、そのアイテムフィルターを指定します。

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

foreach 関数（例: *avg_foreach*、*count_foreach* など）は、選択された各アイテムに対して 1 つの集計値を返します。アイテムは、アイテム履歴からアイテムフィルター（/host/key?[group="host group"]）を使用して選択されます。詳細は、foreach 関数を参照してください。

要求された期間に対してデータを持たないアイテムがある場合、それらは計算で無視されます。データを持つアイテムが 1 つもない場合、関数はエラーを返します。

または、集計のパラメータとして複数のアイテムを列挙することもできます。

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

ここでの function は、履歴/トレンド関数である必要があることに注意してください。

> **Note:**
> 集計の結果が浮動小数点値になる場合、集計対象アイテムの情報の型が *Numeric (unsigned)* であれば、整数に切り詰められます。

ユーザーマクロおよびローレベルディスカバリマクロは、以下でサポートされています。

- アイテムキーのパラメータ
- 関数パラメータ
- フィルター条件（ホストグループ名およびタグ名）
- 式の定数

以下の場合、集計計算は未サポートになることがあります。

- 参照されているアイテムが 1 つも見つからない場合（これは、アイテムキーが正しくない、該当するアイテムが存在しない、または含まれるすべてのグループが正しくない場合に発生する可能性があります）
- 関数を計算するためのデータがない場合

使用例

集計のためのキーの例です。

例 1

ホストグループ「MySQL Servers」のディスク総容量。

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

例 2

ホスト上で net.if.in[*] に一致するすべてのアイテムの最新値の合計。

```
sum(last_foreach(/host/net.if.in[*]))
```

例 3

ホストグループ「MySQL Servers」の平均プロセッサー負荷。

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

例 4

ホストグループ「MySQL Servers」の 1 秒あたりのクエリ数の 5 分間平均。

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

例 5

特定のタグを持つ複数のホストグループ内のすべてのホストにおける、CPU 負荷の平均値。

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

例 6

ホストグループ全体の最新のアイテム値の合計に対して使用される計算。

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

例 7

ホストグループ「Zabbix servers」における未サポートのアイテムの総数。

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

例 8

すべてのホストにわたる DNS チェックの最新の数値結果の合計。表示されているアイテムの形式 `net.dns[192.0.2.0,example.com,A]` は、可能なキーの一例です。

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

ワイルドカードは、キー内のパラメータ数と一致している必要があることに注意してください（この場合、net.dns には 3 つのパラメータがあります: ip、name、type）。

正しい/誤った構文の例

式（関数呼び出しを含む）は、履歴、トレンド、または foreach 関数 のパラメータとして使用できません。ただし、これらの関数自体は、他の（履歴系ではない）関数パラメータで使用できます。

式	例
有効	`avg(last(/host/key1),last(/host/key2)*10,last(/host/key1)*100)`
	`max(avg(avg_foreach(/*/system.cpu.load?[group="Servers A"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers B"],5m)),avg(avg_foreach(/*/system.cpu.load?[group="Servers C"],5m)))`
無効	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

次のような式では、次の点に注意してください。

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

式の両側で、常に同じ値の集合になるとは限りません。式の一方が評価されている間に、要求された期間に対する新しい値が到着する可能性があり、その結果、式のもう一方では異なる値の集合を持つことになります。

8 内部チェック

概要

内部チェックを使用すると、Zabbix の内部プロセスを監視できます。つまり、Zabbix サーバーまたは Zabbix プロキシで何が起こっているかを監視できます。

内部チェックは次の場所で計算されます。

- Zabbix サーバー上 - ホストがサーバーによって監視されている場合
- Zabbix プロキシ上 - ホストがプロキシによって監視されている場合

内部チェックは、ホストのメンテナンス状態に関係なく、サーバーまたはプロキシによって処理されます。

このアイテムを使用するには、アイテムタイプとして **Zabbix internal** を選択します。

> **Note:**
> 内部チェックは Zabbix poller によって処理されます。

パフォーマンス

一部の内部アイテムを使用すると、パフォーマンスに悪影響を及ぼす可能性があります。該当するアイテムは次のとおりです。

- zabbix[host,,items]
- zabbix[host,,items_unsupported]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[queue,,]
- zabbix[requiredperformance]
- zabbix[stats,,,queue,,]
- zabbix[triggers]

システム情報 およびキュー の Web インターフェースセクションも影響を受けます。

サポートされているチェック

アイテムキーは、カスタマイズ可能なパラメータおよび追加情報を除いて一覧表示されています。
完全な詳細を表示するには、アイテムキーをクリックしてください。

Item key	説明
zabbix[boottime]	Zabbix サーバーまたは Zabbix プロキシプロセスの起動時刻（秒）。
zabbix[cluster,discovery,nodes]	高可用性クラスターのノードを検出します。
zabbix[connector_queue]	コネクターキューに追加された値の数。
zabbix[discovery_queue]	ディスカバリキューに追加されたネットワークチェックの数。
zabbix[host,,items]	ホスト上の有効なアイテム（サポートされているものとサポートされていないもの）の数。
zabbix[host,,items_unsupported]	ホスト上の有効な未サポートアイテムの数。
zabbix[host,maintenance]	ホストの現在のメンテナンス状態。
zabbix[host,active_agent,available]	ホスト上のアクティブエージェントチェックの可用性。
zabbix[host,discovery,interfaces]	Zabbix Web インターフェースで設定されたホストのすべてのインターフェースの詳細。
zabbix[host,,available]	ホスト上の特定の種類のチェックにおけるメインインターフェースの可用性。
zabbix[hosts]	監視対象ホストの数。
zabbix[items]	有効なアイテム（サポートされているものとサポートされていないもの）の数。
zabbix[items_unsupported]	未サポートアイテムの数。
zabbix[java,,]	Zabbix Java ゲートウェイに関する情報。
zabbix[lld_queue]	ローレベルディスカバリ処理キューに追加された値の数。
zabbix[preprocessing]	前処理マネージャーが受信した値の統計。
zabbix[preprocessing_queue]	前処理キューに追加された値の数。
zabbix[process,,,]	特定の Zabbix プロセス、またはプロセスのグループ（<type> および <mode> で識別）が、<state> に費やした時間の割合。
zabbix[proxy,,]	Zabbix プロキシに関する情報。
zabbix[proxy,discovery]	Zabbix プロキシの一覧。
zabbix[proxy group,,available]	プロキシグループ内のオンラインプロキシ数。
zabbix[proxy group,,pavailable]	プロキシグループ内のオンラインプロキシの割合。
zabbix[proxy group,,proxies]	プロキシグループ内の Zabbix プロキシの一覧。
zabbix[proxy group,,state]	プロキシグループの状態。
zabbix[proxy group,discovery]	設定データおよびリアルタイムデータを含むプロキシグループの一覧を返します。
zabbix[proxy_buffer,buffer,]	プロキシメモリバッファの使用状況統計を返します。
zabbix[proxy_buffer,state,changes]	起動以降のディスク/メモリバッファモード間の状態変更回数を返します。
zabbix[proxy_buffer,state,current]	新しいデータが保存されている現在の動作状態を返します。
zabbix[proxy_history]	サーバーへの送信待ちとなっているプロキシ履歴テーブル内の値の数。
zabbix[queue,,]	キュー内の監視対象アイテムのうち、少なくとも <from> 秒遅延し、かつ <to> 秒未満のものの数。
zabbix[rcache,,]	Zabbix 設定キャッシュの可用性統計。
zabbix[requiredperformance]	Zabbix サーバーまたは Zabbix プロキシに必要な性能。予想される 1 秒あたりの新規値数で表されます。
zabbix[stats,,]	Zabbix サーバーまたはプロキシの内部メトリクスを返します。<ip> と <port> が指定されている場合はリモートインスタンスから、そうでない場合はローカルインスタンスからメトリクスを取得します。

Item key	説明
zabbix[stats„,queue„]	Zabbix サーバーまたはプロキシの内部キューメトリクスを返します。<ip> と <port> が指定されている場合はリモートインスタンスから、そうでない場合はローカルインスタンスからメトリクスを取得します。
zabbix[tcache„]	Zabbix トレンド関数キャッシュの有効性統計。
zabbix[triggers]	Zabbix データベース内の有効なトリガー数。対象は、有効なホスト上で、すべてのアイテムが有効なものです。
zabbix[uptime]	Zabbix サーバーまたはプロキシプロセスの稼働時間（秒）。
zabbix[vcache,buffer,]	Zabbix 値キャッシュの可用性統計。
zabbix[vcache,cache,]	Zabbix 値キャッシュの有効性統計。
zabbix[version]	Zabbix サーバーまたはプロキシのバージョン。
zabbix[vmware,buffer,]	Zabbix vmware キャッシュの可用性統計。
zabbix[vps,written]	データベースに書き込まれた履歴値の総数。
zabbix[wcache„]	Zabbix 書き込みキャッシュの統計および可用性。

アイテムキーの詳細

- 山かっこなしのパラメータは必須であり、そのまま使用する必要があります（例: `zabbix[host,<type>,available]` の”host” および”available”）。
- 山かっこ **< >** 付きのパラメータは、有効な値に置き換える必要があります。パラメータにデフォルト値がある場合は、省略できます。
- 「not supported on proxy」と記載されたアイテムおよびアイテムパラメータの値は、そのホストがサーバーによって監視されている場合にのみ取得できます。逆に、「not supported on server」の値は、そのホストがプロキシによって監視されている場合にのみ取得できます。

zabbix[boottime]

 Zabbix サーバーまたは Zabbix プロキシプロセスの起動時刻（秒単位）。
 戻り値: *Integer*。

zabbix[cluster,discovery,nodes]

 高可用性クラスターのノードを検出します。
 戻り値: *JSON* オブジェクト。

コメント:

- このアイテムはローレベルディスカバリで使用できます。
- 返されるフィールドは、主に高可用性ノードオブジェクトのプロパティに対応しています。
- 追加フィールド:
  - `db_timestamp` — レスポンスに含まれる現在のサーバー時刻（Unix タイムスタンプ）
  - `lastaccess_age` — 最後のノードハートビートからの経過秒数（db_timestamp - lastaccess）
  - `status` — ノードの状態:
    - 0 — スタンバイ
    - 1 — 手動停止
    - 2 — 利用不可
    - 3 — アクティブ

戻り値の例:


```
[
  {
    "id": "ckvupihk70001z8mkpw5cg0u3",
    "name": "zabbix-prod-01",
    "status": 3,
    "address": "10.0.4.12:10051",
    "port": 10051,
    "lastaccess": 1756115995,
    "db_timestamp": 1756116000,
    "lastaccess_age": 5
  },
  {
    "id": "ckvx2a9k70004b1nq2hz9d7f",
    "name": "zabbix-standby-02",
    "status": 0,
    "address": "10.0.4.13:10051",
    "port": 10051,
    "lastaccess": 1756115550,
```

```
    "db_timestamp": 1756116000,
    "lastaccess_age": 450
  },
  {
    "id": "ckw0bq3l70007y4r1a0m5kz8",
    "name": "zabbix-backup-eu1",
    "status": 1,
    "address": "backup.example.com:10051",
    "port": 10051,
    "lastaccess": 1756105080,
    "db_timestamp": 1756116000,
    "lastaccess_age": 10920
  }
]
```


zabbix[connector_queue]

 コネクタキューにキューイングされている値の数。
 戻り値: 整数。

zabbix[discovery_queue]

 ディスカバリキューにキューイングされているネットワークチェックの数。
 戻り値: *Integer*。

zabbix[host,,items]

 ホスト上の有効なアイテム（サポートされているものとサポートされていないものの両方）の数。
 戻り値: 整数。

zabbix[host,,items_unsupported]

 ホスト上で有効になっている未サポートのアイテム数。
 戻り値: *Integer*。

zabbix[host,,maintenance]

 ホストの現在のメンテナンス状態。
 戻り値: *0* - 通常状態; *1* - データ収集ありのメンテナンス; *2* - データ収集なしのメンテナンス。

コメント:

- このアイテムは、ホストの場所（サーバー上またはプロキシ上）に関係なく、常に Zabbix サーバーによって処理されます。このアイテムは設定データとともにプロキシには送信されません。
- 2 番目のパラメータは空でなければならず、将来の使用のために予約されています。

zabbix[host,active_agent,available]

 ホスト上のアクティブエージェントチェックの可用性。
 戻り値: *0* - 不明; *1* - 利用可能; *2* - 利用不可。

zabbix[host,discovery,interfaces]

 Zabbix Web インターフェースにおけるホストの設定済みインターフェースすべての詳細です。
 戻り値: *JSON object*。

コメント:

- このアイテムはローレベルディスカバリで使用できます。
- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[host,<type>,available]

 ホスト上の特定のチェック種別におけるメインインターフェースの可用性。
 戻り値: *0* - 不明; *1* - 利用可能; *2* - 利用不可。

パラメータ:

- **type** - *agent*、*snmp*、*ipmi*、または *jmx*。

コメント:

- このアイテムの値は、ホストの到達不能/利用不可に関する設定パラメータに従って計算されます。

zabbix[hosts]

 監視対象のホスト数。
 戻り値: *Integer*。

zabbix[items]

 有効なアイテムの数（サポートされているものとサポートされていないもの）。
 戻り値: *Integer*。

zabbix[items_unsupported]

 未サポートのアイテム数。
 戻り値: *Integer*。

zabbix[java,,<param>]

 Zabbix Java ゲートウェイに関する情報です。
 戻り値: <param> が *ping* の場合は *1*、<param> が *version* の場合は *Java* ゲートウェイのバージョン（例: "7.4.0"）。

パラメータ:

- **param** - *ping* または *version*。

コメント:

- このアイテムは、nodata() トリガー関数を使用して Java ゲートウェイの可用性を確認するために使用できます。
- 2 番目のパラメータは空である必要があり、将来の使用のために予約されています。

zabbix[lld_queue]

 低レベルディスカバリ処理キューに追加された値の数。
 戻り値: 整数。

コメント:

- このアイテムは、低レベルディスカバリ処理キューの長さを監視するために使用できます。

zabbix[preprocessing]

 前処理マネージャーが受信した値の統計情報:

- *queued* - 前処理が必要なキュー済みの値の数とサイズ（カウンター）
- *direct* - 前処理が不要なキュー済みの値の数とサイズ（カウンター）
- *queue* - 前処理キューに追加された値の数（zabbix[preprocessing_queue] と同じ）

戻り値: *JSON*。

戻り値の例:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 前処理キューに投入された値の数。
 戻り値: 整数。

コメント:

- このアイテムは、前処理キューの長さを監視するために使用できます。

zabbix[process,<type>,<mode>,<state>]

 特定の Zabbix プロセス、または <type> と <mode> で識別されるプロセスグループが、<state> の状態で費やした時間の割合です。
直近 1 分間についてのみ計算されます。
 戻り値: *Float*。

パラメータ:

- **type** - server processes の場合: *agent poller*, *alert manager*, *alert syncer*, *alerter*, *availability manager*, *browser poller*, *configuration syncer*, *configuration syncer worker*, *connector manager*, *connector worker*, *discovery manager*, *discovery worker*, *escalator*, *ha manager*, *history poller*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *lld manager*, *lld worker*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *proxy group manager*, *proxy poller*, *self-monitoring*, *service manager*, *snmp poller*, *snmp trapper*, *task manager*, *timer*, *trapper*, *trigger housekeeper*, *unreachable poller*, *vmware collector*;
proxy processes の場合: *agent poller*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc*

poller, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;
- **mode** - *avg* - 指定したタイプのすべてのプロセスの平均値（デフォルト）;
count - 指定したプロセスタイプの fork 数を返します。この場合、<state> は指定しないでください;
max - 最大値;
min - 最小値;
<*process number*> - プロセス番号（1 から事前 fork されたインスタンス数まで。たとえば、4 つの trapper が動作している場合、値は 1 から 4 です）;
- **state** - *busy* - プロセスがビジー状態です。たとえば、リクエストを処理中の状態です（デフォルト）;
idle - プロセスが何もしていないアイドル状態です。

コメント:

- <mode> が動作していない Zabbix プロセス番号である場合（たとえば、5 つの poller が動作しているのに <mode> に 6 を指定した場合）、そのアイテムは未サポートになります。
- 最小値および最大値は、単一プロセスの使用率を指します。
したがって、3 つの poller グループ内で各プロセスの使用率が 2、18、66 であった場合、min は 2 を返し、max は 66 を返します。
- プロセスは共有メモリ内に自分が何をしているかを報告し、self-monitoring プロセスがそのデータを毎秒集計します。
状態の変化（busy/idle）は変化時に登録されます。つまり、プロセスが busy になるとその状態が登録され、idle になるまで状態は変更も更新もされません。
これにより、完全にハングしたプロセスであっても、100% busy として正しく記録されます。
- 現在、”busy” は「スリープしていない」ことを意味しますが、将来的には追加の状態が導入される可能性があります。たとえば、ロック待ち、データベースクエリの実行中などです。
非同期 poller は、MaxConcurrentChecksPerPoller server/proxy 設定パラメータで設定された上限に達している場合、busy と見なされることに注意してください。
- Linux およびその他ほとんどのシステムでは、分解能は 1/100 秒です。

例:

```
zabbix[process,poller,avg,busy] #the average time of poller processes spent doing something during the las
zabbix[process,"icmp pinger",max,busy] #the maximum time spent doing something by any ICMP pinger process
zabbix[process,"history syncer",2,busy] #the time spent doing something by history syncer number 2 during
zabbix[process,trapper,count] #the amount of currently running trapper processes
```

zabbix[proxy,<name>,<param>]

 Zabbix プロキシに関する情報です。
 戻り値: 整数。

パラメータ:

- **name** - プロキシ名。
- **param** - *lastaccess* - プロキシから最後に受信したハートビートメッセージのタイムスタンプ。
delay - 収集された値が未送信のままになっている時間。”proxy delay” + (”current server time” - ”proxy lastaccess”) として計算されます。ここで”proxy delay” は、現在のプロキシ時刻と、プロキシ上で最も古い未送信の値のタイムスタンプとの差です。

コメント:

- このアイテムは、ホストの場所（サーバー上またはプロキシ上）に関係なく、常に Zabbix サーバーによって処理されます。
- プロキシの可用性を確認するために、`fuzzytime()` 関数を使用できます。

例:

```
zabbix[proxy,"Germany",lastaccess] #the timestamp of the last heartbeat message received from "Germany" pr
```

zabbix[proxy,discovery]

 名前、モード、暗号化、圧縮、バージョン、最終確認時刻、ホスト数、アイテム数、1 秒あたりの必要値数（vps）、バージョンステータス（current/outdated/unsupported）、アイテムタイプごとのタイムアウト、プロキシグループ名（プロキシがグループに属している場合）、状態（unknown/offline/online）を含む Zabbix プロキシの一覧です。
 戻り値: *JSON object*。

zabbix[proxy group,<name>,available]

 プロキシグループ内でオンラインのプロキシ数。
 戻り値: *Integer*。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,<name>,pavailable]

 プロキシグループ内でオンラインのプロキシの割合。
 戻り値: *Float*。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,<name>,proxies]

 名前、モード、暗号化、圧縮、バージョン、最終アクセス時刻、ホスト数、アイテム数、1 秒あたりの必要値数（vps）、バージョン状態（current/outdated/unsupported）、タイムアウト、プロキシグループ名、状態（unknown/offline/online）を含む、指定したプロキシグループ内の Zabbix プロキシの一覧です。
 戻り値: *JSON*。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,<name>,state]

 プロキシグループの状態。
 戻り値: *0* - 不明; *1* - オフライン; *2* - 復旧中; *3* - オンライン; *4* - 劣化中。

パラメータ:

- **name** - プロキシグループ名。

zabbix[proxy group,discovery]

 設定データおよびリアルタイムデータを含むプロキシグループの一覧を返します。設定データには、プロキシグループ名、フェイルオーバー遅延、および必要なオンラインのプロキシの最小数が含まれます。リアルタイムデータには、プロキシグループの状態(詳細はコメントを参照)、オンラインのプロキシ数、およびオンラインのプロキシの割合が含まれます。
 戻り値: *JSON*。

コメント:

- このアイテムは、グループに属していないプロキシを返しません。
- ”failover_delay” または”min_online” に無効な値がある場合、それを示すために特別な値 *-1* が報告されます。無効な値は、設定にマクロが使用されていて、そのマクロを有効な値に展開できない場合に発生することがあります。
- プロキシグループの状態は整数として報告されます: *0* - 不明; *1* - オフライン; *2* - 復旧中; *3* - オンライン; *4* - 劣化中。

戻り値の例:

```
{
    "groups": [
       { "name": "Riga", "failover_delay": 60, "min_online": 1 },
       { "name": "Tokyo", "failover_delay": 60, "min_online": 2 },
       { "name": "Porto Alegre", "failover_delay": 60, "min_online": 3 }
    ],
    "details": {
        "Riga": { "state": 3, "available": 10, "pavailable": 20 },
        "Tokyo": { "state": 3, "available": 10, "pavailable": 20 },
        "Porto Alegre": { "state": 1, "available": 0, "pavailable": 0 }
    }
}
```

zabbix[proxy_buffer,buffer,<mode>]

 プロキシのメモリバッファ使用統計です。
 戻り値: *Integer*（サイズの場合）; *Float*（パーセンテージの場合）。

パラメータ:

- **mode**: *total* - バッファの合計サイズ（メモリバッファが有効かどうかの確認に使用できます）;
free - 空きバッファのサイズ;
pfree - 空きバッファの割合;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファの割合。

コメント:

- メモリバッファが無効な場合は、’Proxy memory buffer is disabled’ エラーを返します;
- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[proxy_buffer,state,changes]

 起動以降の、ディスク/メモリバッファモード間の状態変更回数を返します。
 戻り値: *Integer*; *0* - メモリバッファは無効です。

コメント:

- 状態変更が頻繁に発生する場合は、メモリバッファのサイズまたは保持期間を増やす必要があることを示しています。
- メモリバッファの状態の監視頻度が低い場合（たとえば 1 分に 1 回）、バッファが状態を切り替えても記録されない可能性があります。

zabbix[proxy_buffer,state,current]

 新しいデータが保存されている現在の動作状態を返します。
 戻り値: *0* - ディスク、*1* - メモリ。

コメント:

- メモリバッファが無効になっている場合も”0” が返されます。

zabbix[proxy_history]

 サーバーへの送信待ちとなっている、プロキシの history テーブル内の値の数。
 戻り値: 整数。

コメント:

- このアイテムは Zabbix サーバーではサポートされていません。

zabbix[queue,<from>,<to>]

 キュー内にある監視対象アイテムのうち、少なくとも <from> 秒以上遅延し、かつ <to> 秒未満のものの数。
 戻り値: 整数。

パラメータ:

- **from** - 少なくともこの秒数以上遅延（デフォルトは 6 秒）;
- **to** - 最大でこの秒数未満の遅延（デフォルトは無限大）。

コメント:

- パラメータでは時間のサフィックス (s,m,h,d,w) をサポートしています。

> **Attention:**
> Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算 を参照してください。

zabbix[rcache,<cache>,<mode>]

 Zabbix 設定キャッシュの可用性統計。
 戻り値: *Integer*（サイズの場合）; *Float*（パーセンテージの場合）。

パラメータ:

- **cache** - *buffer*;
- **mode** - *total* - バッファの合計サイズ;
free - 空きバッファのサイズ;
pfree - 空きバッファの割合;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファの割合。

zabbix[requiredperformance]

 想定される 1 秒あたりの新規値数で表した、Zabbix サーバーまたは Zabbix プロキシに必要なパフォーマンスです。
 戻り値: *Float*。

コメント:

- *Reports* > *System information* の「Required server performance, new values per second」とおおよそ相関します。

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

 Zabbix サーバーまたはプロキシの内部メトリクスを返します。<ip> および <port> が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON object*。

パラメータ:

- **ip** - リモート照会されるサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）;
- **port** - リモート照会されるサーバー/プロキシのポート（デフォルトは 10051）。

コメント:

- stats リクエストは、対象インスタンスの’StatsAllowedIP’ server/proxy パラメータに一覧表示されているアドレスからのみ受け付けられます。
- このアイテムは、選択された一連の内部メトリクスを返します。詳細については、Zabbix stats のリモート監視 を参照してください。

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

 Zabbix サーバーまたはプロキシの内部キューメトリクス（`zabbix[queue,<from>,<to>]` を参照）を返します。<ip> と <port> が指定されている場合、メトリクスはリモートインスタンスから取得されます。指定されていない場合は、ローカルインスタンスから取得されます。
 戻り値: *JSON object*。

パラメータ:

- **ip** - リモート照会するサーバー/プロキシの IP/DNS/ネットワークマスクリスト（デフォルトは 127.0.0.1）;
- **port** - リモート照会するサーバー/プロキシのポート（デフォルトは 10051）;
- **from** - 少なくともこの秒数だけ遅延（デフォルトは 6 秒）;
- **to** - 最大でこの秒数だけ遅延（デフォルトは無限大）。

コメント:

- stats リクエストは、対象インスタンスの'StatsAllowedIP' server/proxy パラメータに一覧表示されているアドレスからのみ受け付けられます。
- このアイテムは、選択された内部メトリクスのセットを返します。詳細については、Remote monitoring of Zabbix stats を参照してください。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細については、Calculation of queues during maintenance を参照してください。

zabbix[tcache,cache,<parameter>]

 Zabbix トレンド関数キャッシュの有効性に関する統計です。
 戻り値: *Integer*（サイズの場合）; *Float*（パーセンテージの場合）。

パラメータ:

- **parameter** - *all* - キャッシュ要求の合計（デフォルト）;
hits - キャッシュヒット数;
phits - キャッシュヒット率;
misses - キャッシュミス数;
pmisses - キャッシュミス率;
items - キャッシュされたアイテム数;
requests - キャッシュされた要求数;
pitems - キャッシュされたアイテム数 + 要求数に対する、キャッシュされたアイテム数の割合。割合が低い場合、キャッシュサイズを縮小できる可能性が高いことを意味します。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[triggers]

 有効なホスト上で、すべてのアイテムが有効になっているトリガーのうち、Zabbix データベース内で有効なトリガーの数です。
 戻り値: *Integer*。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[uptime]

 Zabbix サーバーまたはプロキシプロセスの稼働時間（秒単位）。
 戻り値: *Integer*。

zabbix[vcache,buffer,<mode>]

 Zabbix 値キャッシュの可用性統計。
 戻り値: *Integer*（サイズの場合）; *Float*（割合の場合）。

パラメータ:

- **mode** - *total* - バッファの合計サイズ;
free - 空きバッファのサイズ;
pfree - 空きバッファの割合;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファの割合。

コメント:

- このアイテムは Zabbix プロキシではサポートされていません。

zabbix[vcache,cache,<parameter>]

 Zabbix 値キャッシュの有効性に関する統計情報です。
 戻り値: *Integer*。<parameter> が *mode* に設定されている場合、戻り値は次のとおりです: *0* - 通常モード、*1* - 低メモリモード。

パラメータ:

- **parameter** - *requests* - リクエストの総数;
hits - キャッシュヒット数（キャッシュから取得された履歴値）;
misses - キャッシュミス数（データベースから取得された履歴値）;
mode - 値キャッシュの動作モード。

コメント:

- 低メモリモードが一度有効になると、このモードを引き起こした問題がそれより早く解決された場合でも、値キャッシュは 24 時間この状態のままになります。
- 値/秒の統計を取得するために、このキーを *Change per second* 前処理ステップとともに使用できます。
- このアイテムは Zabbix プロキシではサポートされていません。

zabbixversion

 Zabbix サーバーまたはプロキシのバージョン。
 戻り値: *String*。例: `7.4.0`。

zabbix[vmware,buffer,<mode>]

 Zabbix vmware キャッシュの可用性統計。
 戻り値: *Integer*（サイズの場合）; *Float*（パーセンテージの場合）。

パラメータ:

- **mode** - *total* - バッファの合計サイズ;
free - 空きバッファのサイズ;
pfree - 空きバッファの割合;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファの割合。

zabbix[vps,written]

 データベースに書き込まれた履歴値の総数。
 戻り値: 整数。

zabbix[wcache,<cache>,<mode>]

 Zabbix 書き込みキャッシュの統計情報と可用性。
 戻り値: *Integer*（数/サイズの場合）; *Float*（パーセンテージの場合）。

パラメータ:

- **cache** - *values*、*history*、*index*、または *trend*;
- **mode** - （*values* の場合）*all*（デフォルト）- サポートされていないアイテムを除き、Zabbix サーバー/プロキシによって処理された値の総数（カウンター）;
float - 処理された浮動小数点値の数（カウンター）;
uint - 処理された符号なし整数値の数（カウンター）;
str - 処理された文字/文字列値の数（カウンター）;
log - 処理されたログ値の数（カウンター）;
text - 処理されたテキスト値の数（カウンター）;
bin - 処理されたバイナリ値の数（カウンター）;
not supported - アイテムの処理によりアイテムがサポートされていない状態になった、またはその状態が維持された回数（カウンター）;
（*history*、*index*、*trend* キャッシュの場合）*pfree*（デフォルト）- 空きバッファの割合;
total - バッファの合計サイズ;
free - 空きバッファのサイズ;
used - 使用済みバッファのサイズ;
pused - 使用済みバッファの割合。

コメント:

- `trend` キャッシュパラメータは Zabbix プロキシではサポートされていません。
- 履歴キャッシュはアイテムの値を保存するために使用されます。数値が低い場合は、データベース側のパフォーマンス上の問題を示しています。
- 履歴インデックスキャッシュは、履歴キャッシュに保存された値のインデックス作成に使用されます。
- 履歴キャッシュがいっぱいになってからクリアされた後も、履歴インデックスキャッシュには一部のデータが保持されます。この動作は想定されたものであり、メモリサイズを絶えず変更するために必要な追加処理を回避することで、システムをより効率的に動作させるのに役立ちます。
- トレンドキャッシュには、データを受信するすべてのアイテムについて、現在の時間の集計が保存されます。
- 1 秒あたりの値の統計を取得するには、前処理ステップ *Change per second* とともに zabbix[wcache,values] キーを使用できます。

9 SSH チェック

概要

SSH チェックは、エージェントレス監視として実行されます。SSH チェックには Zabbix エージェントは不要です。

SSH チェックを実行するには、事前に Zabbix サーバーを SSH2 サポート（libssh または libssh2）付きで設定しておく必要があります。あわせて要件も参照してください。

> **Attention:**
> RHEL 8 以降では、libssh のみがサポートされています。その他のディストリビューションでは、libssh2 よりも libssh を推奨します。

設定

パスフレーズ認証

SSH チェックでは、2 つの認証方法を利用できます。ユーザー名/パスワードの組み合わせと、キーファイルベースの認証です。

キーを使用しない場合、ソースからビルドしているときに libssh または libssh2 を Zabbix にリンクすることを除き、追加の設定は不要です。

キーファイル認証

SSH アイテムで鍵ベース認証を使用するには、サーバー設定にいくつかの変更が必要です。

`root` で Zabbix サーバー設定ファイル (*zabbix_server.conf*) を開き、次の行を探します:

```
##### SSHKeyLocation=
```

コメントを外し、公開鍵と秘密鍵を配置するディレクトリへのフルパスを設定します:

```
SSHKeyLocation=/home/zabbix/.ssh
```

ファイルを保存し、その後 Zabbix サーバーを再起動します。

ここでのパス */home/zabbix* は *zabbix* ユーザーアカウントのホームディレクトリであり、*.ssh* はデフォルトで公開鍵と秘密鍵がホームディレクトリ内で ssh-keygen コマンドによって生成されるディレクトリです。

通常、各種 OS ディストリビューションの Zabbix サーバーインストールパッケージでは、*zabbix* ユーザーアカウントのホームディレクトリが別の場所に作成されます。たとえば、システムアカウントの場合は */var/lib/zabbix* です。

鍵を生成する前に、ホームディレクトリを */home/zabbix* に変更して、前述の Zabbix サーバー設定パラメータ SSHKeyLocation に対応させることができます。

> **Note:**
> 以下の手順は、インストールセクション に従って *zabbix* アカウントを手動で追加している場合は省略できます。その場合、*zabbix* アカウントのホームディレクトリはすでに */home/zabbix* である可能性が高いです。

zabbix ユーザーアカウントのホームディレクトリを変更するには、それを使用しているすべての稼働中プロセスを停止する必要があります:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

ホームディレクトリの場所を変更し、存在する場合は移動も試みるには、次のコマンドを実行します:

```
usermod -m -d /home/zabbix zabbix
```

古い場所にホームディレクトリが存在しなかった可能性もあるため、新しい場所に作成する必要があります。安全に試行するには次のようにします:

```
test -d /home/zabbix || mkdir /home/zabbix
```

すべてが安全であることを確実にするために、ホームディレクトリの権限を設定する追加コマンドを実行できます:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

先ほど停止したプロセスは、ここで再び起動できます:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

これで、公開鍵と秘密鍵を生成する手順を次のコマンドで実行できます（読みやすくするため、コマンドプロンプトはコメントアウトしています）:

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<空のままにする>
##### Enter same passphrase again:
<空のままにする>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
##### +--[ RSA 2048]----+
##### |                 |
##### |       .         |
##### |      o          |
##### | .     o         |
##### |+     . S        |
##### |.+   o =         |
##### |E .   * =        |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
公開鍵と秘密鍵 (*id_rsa.pub* および *id_rsa*) は、デフォルトで */home/zabbix/.ssh* ディレクトリに生成されます。これは、Zabbix サーバー設定パラメータ SSHKeyLocation に対応しています。

Attention:
”rsa” 以外の鍵タイプも ssh-keygen ツールおよび SSH サーバーでサポートされている場合がありますが、Zabbix が使用する libssh2 ではサポートされない可能性があります。

シェル設定フォーム

この手順は、SSH チェックで監視される各ホストに対して一度だけ実行する必要があります。

以下のコマンドを使用すると、公開鍵ファイルをリモートホスト *10.10.10.10* にインストールでき、*root* アカウントで SSH チェックを実行できるようになります（読みやすさのため、コマンドプロンプトはコメントアウトしています）。

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### ホスト '10.10.10.10 (10.10.10.10)' の真正性を確認できません。
##### RSA 鍵のフィンガープリントは 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46 です。
##### 接続を続行してもよろしいですか (yes/no)?
yes
##### 警告: '10.10.10.10' (RSA) を既知のホストの一覧に恒久的に追加しました。
##### root@10.10.10.10 のパスワード:
<root パスワードを入力>
##### 次に "ssh 'root@10.10.10.10'" でこのマシンにログインしてみて、
##### 追加されたのが意図した鍵のみであることを確認してください。
```

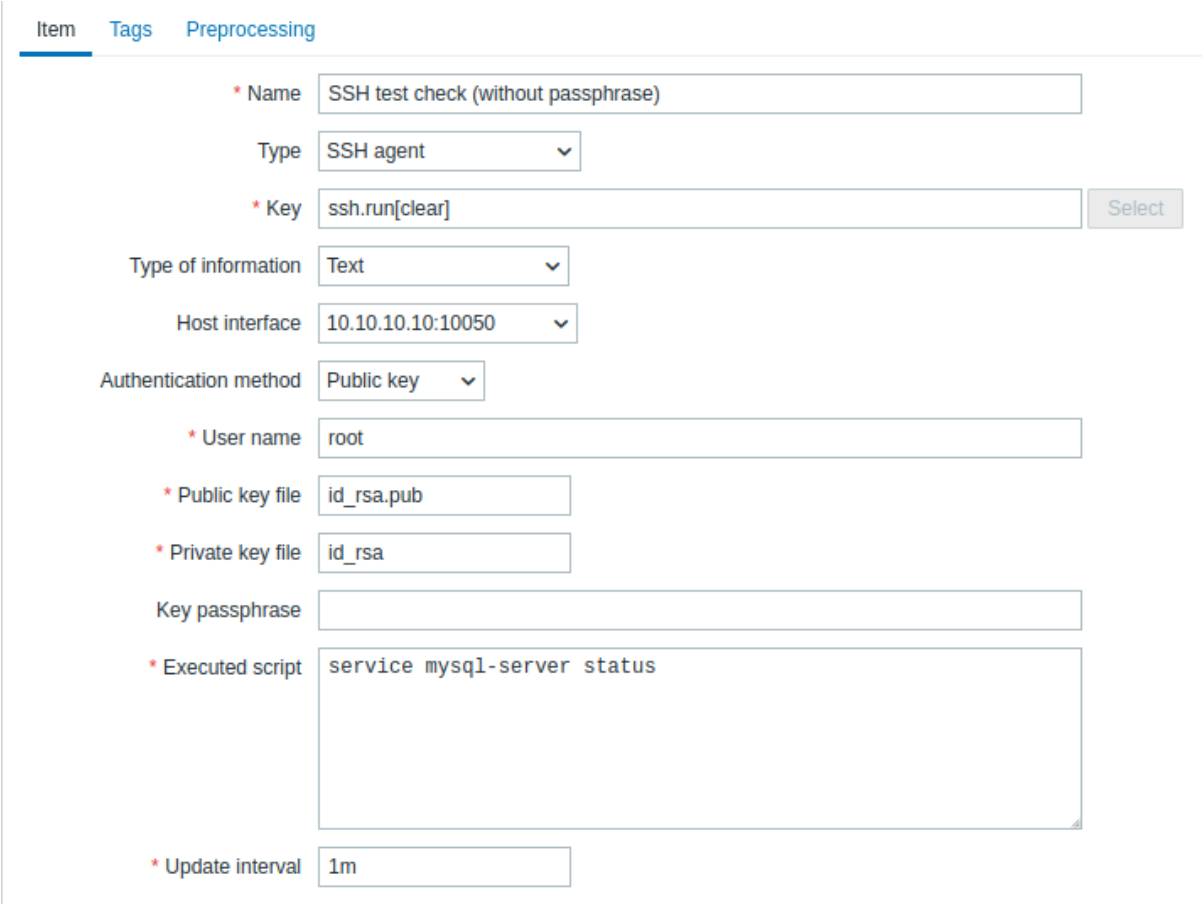
これで、*zabbix* ユーザーアカウントのデフォルトの秘密鍵 (*/home/zabbix/.ssh/id_rsa*) を使用して SSH ログインを確認できるようになります。

```
sudo -u zabbix ssh root@10.10.10.10
```

ログインに成功した場合、シェルでの設定作業は完了しており、リモート SSH セッションを閉じることができます。

アイテムの設定

実際に実行するコマンドは、アイテム設定の *Executed script* フィールドに記述する必要があります。複数のコマンドは、新しい行に記述することで順番に実行できます。この場合、返される値も複数行形式で整形されます。



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

SSH アイテムで固有の情報が必要なフィールドは次のとおりです。

パラメータ	説明	コメント
Type	ここで **SSH agent** を選択します。	

パラメータ	説明	コメント

Key	**ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]** 形式の、一意な（ホストごとの）アイテムキー。	**unique short description** は必須で、ホストごとに各 SSH アイテムで一意である必要があります。
		
		デフォルトのポートは 22 であり、このアイテムに割り当てられたインターフェースで指定されたポートではありません。
		
		ssh options では、*key1=value1;key2=value2,value3* 形式で追加の SSH オプションを渡せます。1 つのキーに対して複数の値を渡す場合はカンマで区切ります（この場合、パラメータはquotedで囲む必要があります）。複数のオプションキーはセミコロンで区切って渡せます。
		
		次のオプションキーがサポートされています: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`。オプションキーと値のサポートは SSH ライブラリに依存します（たとえば、`PubkeyAcceptedKeyTypes` は libssh でのみサポートされます）。オプションがサポートされていない場合はエラーが返され、アイテムは未サポートになります。
		
		暗号設定を追加するための”+” 記号や、特定の暗号設定を無効化するための”!”（GnuTLS や OpenSSL と同様）はサポートされていないことに注意してください。
		
		例:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-he`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey`
		
		subsystem では SSH サブシステムを渡すことができ、SSH 接続をそのサブシステムで許可された特定の操作に限定できます（例: SFTP を使用したファイル転送や、NETCONF を使用したネットワークデバイス管理）。サブシステムを使用する場合、*Executed script* パラメータで特定のスクリプト構文が必要になることがある点に注意してください。
		
		例:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentication method	”Password” または”Public key” のいずれか。	
User name	リモートホストで認証するためのユーザー名（最大 255 文字）。必須。	
Public key file	*Authentication method* が”Public key” の場合の公開鍵ファイル名。必須。	例: *id_rsa.pub* - コマンド ssh-keygen によって生成されるデフォルトの公開鍵ファイル名。
Private key file	*Authentication method* が”Public key” の場合の秘密鍵ファイル名。必須。	例: *id_rsa* - デフォルトの秘密鍵ファイル名。

パラメータ	説明	コメント
Password または *Key passphrase*	認証用のパスワード（最大 255 文字）または秘密鍵に使用されている場合のパスフレーズ。	パスフレーズを使用していない場合は、*Key passphrase* フィールドを空のままにしてください。 known issues のパスフレーズ使用に関する記述も参照してください。

Executed script	SSH リモートセッションを使用して実行されるシェルコマンド。	実行されたシェルコマンドの戻り値は 16MB に制限されます（切り詰められる末尾の空白を含む）。また、database limits も適用されます。
		
		libssh2 ライブラリでは、実行可能なスクリプトが ~32kB に切り詰められる場合があることに注意してください。
		
		例:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		例（NETCONF サブシステムの場合）:
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

10 Telnet チェック

概要

Telnet チェックはエージェントレス監視として実行されます。Zabbix エージェントは必要ありません。

設定可能なフィールド

実際に実行するコマンドは、アイテム設定の **Executed script** フィールドに配置する必要があります。
複数のコマンドは、新しい行に記述することで順番に実行できます。
この場合、返される値も複数行形式になります。

シェルプロンプトの末尾としてサポートされる文字:

- $
- #
- >
- %

Note:
これらの文字のいずれかで終わる telnet プロンプト行は返される値から削除されますが、これはコマンド一覧の最初のコマンドに対してのみ、つまり telnet セッションの開始時にのみ行われます。

キー	説明
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	telnet 接続を使用してリモートデバイス上でコマンドを実行します

Attention:
telnet チェックが非 ASCII 文字を含み、かつ非 UTF8 エンコーディングの値を返す場合は、キーの *<encoding>* パラメータを適切に指定する必要があります。詳細は返される値のエンコーディング ページを参照してください。

11 外部チェック

概要

外部チェックは、Zabbix サーバーがシェルスクリプトを実行するか、またはバイナリを実行して行うチェックです。ただし、ホストが Zabbix プロキシによって監視されている場合、外部チェックはプロキシによって実行されます。

外部チェックでは、監視対象のホスト上でエージェントを実行する必要はありません。

アイテムキーの構文は次のとおりです。

```
script[<parameter1>,<parameter2>,...]
```

各要素の意味は次のとおりです。

ARGUMENT	DEFINITION
script	シェルスクリプトまたはバイナリの名前。
parameter(s)	省略可能なコマンドラインパラメータ。

スクリプトにパラメータを渡したくない場合は、次のように指定できます。

```
script[] or
script
```

Zabbix サーバーまたはプロキシは、外部スクリプト用に指定されたディレクトリを検索し、コマンドを実行します（Zabbixサーバー/プロキシ設定ファイルの ExternalScripts パラメータを参照してください）。コマンドは Zabbix サーバー/プロキシと同じユーザーで実行されるため、必要に応じて、アクセス権限や環境変数はラッパースクリプト内で処理する必要があります。また、そのユーザーが実行できるように、コマンド自体の権限も設定されている必要があります。実行可能なのは、指定されたディレクトリ内のコマンドのみです。

Warning:
外部チェックを過度に使用しないでください。各スクリプトの実行には Zabbix サーバー/プロキシによる fork プロセスの開始が必要であり、多数のスクリプトを実行すると Zabbix のパフォーマンスが大幅に低下する可能性があります。

使用例

最初のパラメータ '-h' を指定してスクリプト **check_oracle.sh** を実行します。2 番目のパラメータは、ホストのプロパティでの選択に応じて、IP アドレスまたは DNS 名に置き換えられます。

```
check_oracle.sh["-h","{HOST.CONN}"]
```

ホストが IP アドレスを使用するように設定されている場合、Zabbix サーバー/プロキシは次を実行します。

```
check_oracle.sh '-h' '192.168.1.4'
```

外部チェックの結果

外部チェックの戻り値は、チェックによって生成される標準出力と標準エラー出力です。

Attention:
テキスト（character、log、または text 型の情報）を返すアイテムは、標準エラー出力があっても未サポートにはなりません。

戻り値は 16MB に制限されています（切り捨てられる末尾の空白文字を含む）。データベースの制限も適用されます。

要求されたスクリプトが見つからない場合、または Zabbix サーバー/プロキシにその実行権限がない場合、アイテムは未サポートになり、対応するエラーメッセージが表示されます。

タイムアウトした場合、アイテムは未サポートになり、対応するエラーメッセージが表示され、スクリプト用に fork されたプロセスは終了されます。

12 トラッパーアイテム

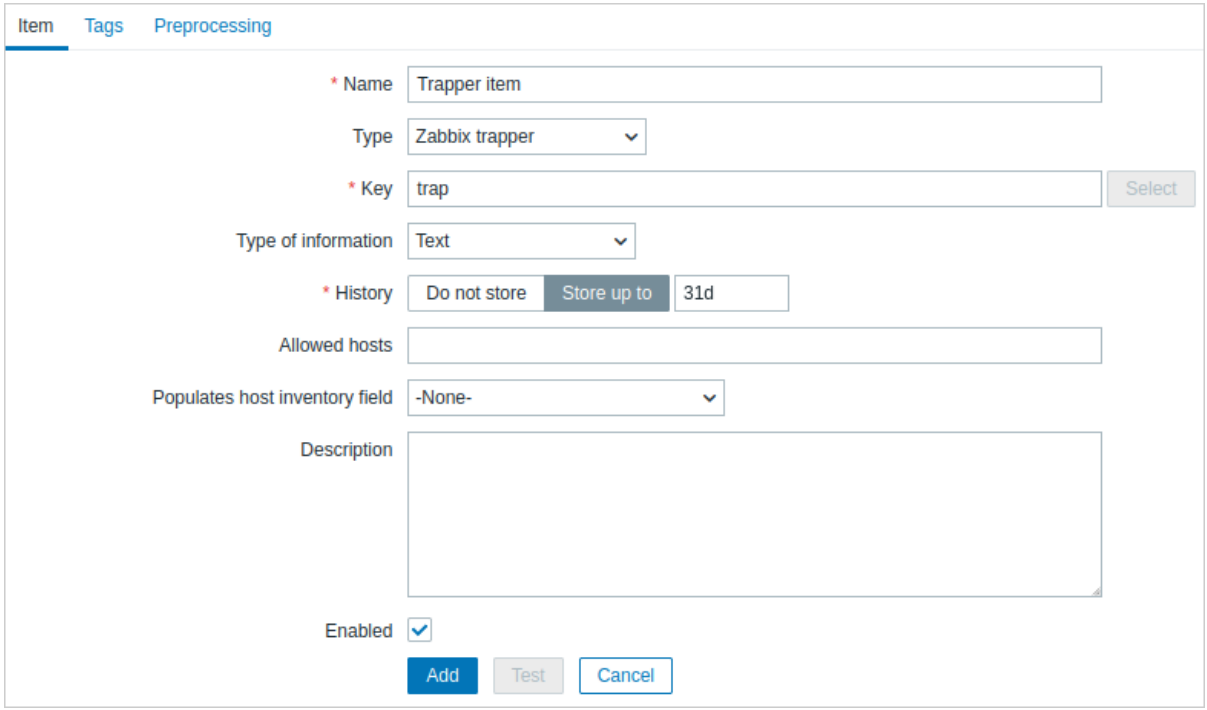
概要

トラッパーアイテムは、データを問い合わせて取得する代わりに、受信したデータを受け付けます。
これは、Zabbix に送信したいあらゆるデータに役立ちます。

設定

トラッパーアイテムを設定するには、次のようにします。

1. データ収集 → ホストに移動します。
2. ホストの行にある アイテムをクリックします。
3. アイテムの作成をクリックします。
4. 設定フォームにアイテムのパラメータを入力します。



必須入力フィールドには、赤いアスタリスクが付いています。

トラッパーアイテムで固有の情報が必要なフィールドは次のとおりです。

タイプ	「Zabbix トラッパー」を選択します。
キー	Zabbix サーバーにデータを送信する際にアイテムを識別するために使用されるキーを入力します。
データ型	送信されるデータの形式に対応する情報の種類を選択します。
許可するホスト	カンマ区切りの IP アドレス（CIDR 表記も可）または DNS 名の一覧です。 指定した場合、ここに一覧表示されたホストからの受信接続のみが許可されます。 IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。任意の IPv4 アドレスを許可するには'0.0.0.0/0' を使用できます。 なお、「IPv4 互換 IPv6 アドレス」（0000::/96 プレフィックス）はサポートされていますが、RFC4291 により非推奨とされています。 例: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} スペース、ユーザーマクロ、およびホストマクロ {HOST.HOST}、{HOST.NAME}、{HOST.IP}、{HOST.DNS}、{HOST.CONN} がサポートされています。

Note:
値を送信する前に、アイテム保存後、設定キャッシュの更新によって Zabbix サーバーが変更を反映するまで最大 60 秒待つ必要がある場合があります。

データの送信

Zabbix sender ユーティリティまたは Zabbix sender protocol を使用して、Zabbix サーバーまたはプロキシにデータを送信できます。`history.push` API メソッドを使用して、Zabbix サーバーにデータを送信することもできます。

Zabbix sender

Zabbix sender ユーティリティを使用して Zabbix サーバーまたはプロキシにデータを送信するには、次のコマンドを実行して「test value」を送信できます。

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

「test value」を送信するには、次のコマンドオプションを使用します。

- -z は Zabbix サーバーの IP アドレスを指定します
- -p は Zabbix サーバーのポート番号を指定します（デフォルトは 10051）
- -s はホストを指定します（表示名ではなく、技術的なホスト名を使用してください）
- -k はトラッパーアイテムで設定されたアイテムのキーを指定します
- -o は送信する値を指定します

Attention:
Zabbix トラッパープロセスは、対象ホストに対応するアイテムキーの存在を確認するために、アイテムキー内で使用されているマクロを展開しません。

Zabbix sender と Zabbix サーバーまたはプロキシ間の通信の詳細については、*Zabbix sender protocol*を参照してください。

history.push

history.push API メソッドを使用して Zabbix サーバーにデータを送信するには、いくつかのテスト値を含む次の HTTP POST リクエストを実行できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"history.push","params":[{"itemid":10600,"value":"test value 1"},{"ite
```

リクエストが正しければ、API から返されるレスポンスは次のようになります。

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

レスポンスデータ内のエラーは、特定のアイテムへのデータ送信が Zabbix サーバーによる検証に失敗したことを示します。これは、次の理由で発生する可能性があります。

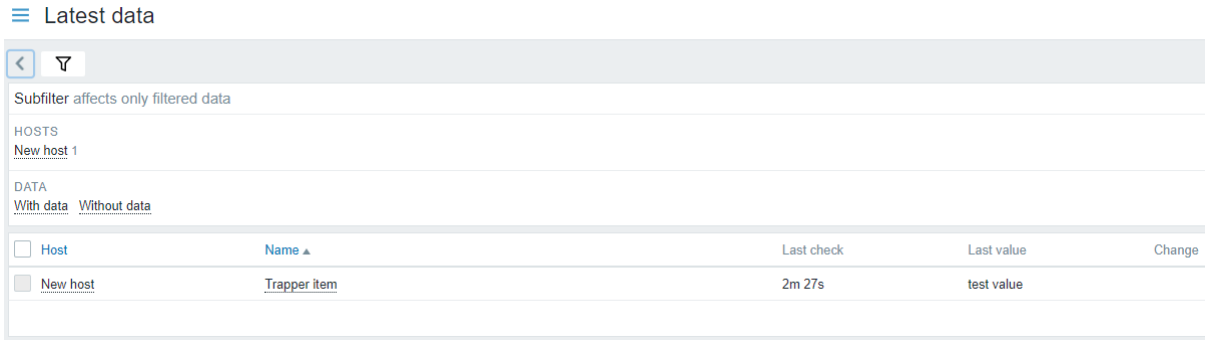
- データを送信するユーザーに、そのアイテムのホストに対する *read* 権限がない。
- ホストが無効になっている、またはデータ収集を伴わないメンテナンス中である。
- アイテムが存在しない、またはまだサーバー設定キャッシュに含まれていない。
- アイテムが無効になっている、またはそのタイプが Zabbix トラッパーもしくはHTTP agent（トラップ有効）以外である。
- ユーザーの IP または DNS が、そのアイテムの *Allowed hosts* リストに設定されていない。
- 別のアイテムに、ナノ秒レベルでタイムスタンプが重複する値がある。

エラーがない場合は、送信された値が処理対象として受け入れられたことを示します。これには、前処理（存在する場合）、トリガー処理、およびデータベースへの保存が含まれます。なお、受け入れられた値の処理も失敗する可能性があります（たとえば、preprocessing中）。その場合、その値は破棄されます。

Zabbix API の使用方法の詳細については、APIを参照してください。

データの表示

データが送信されると、監視 → 最新データ に移動して結果を確認できます。



Note:
単一の数値が送信された場合、データのグラフには、その値の時点を中心として左右に水平線が表示されます。

13 JMX 監視

概要

JMX 監視は、Java アプリケーションの JMX カウンターを監視するために使用できます。

JMX 監視は、"Zabbix Java ゲートウェイ" と呼ばれる Zabbix デーモンの形で、Zabbix にネイティブサポートされています。

ホスト上の特定の JMX カウンターの値を取得するために、Zabbix サーバーは Zabbix **Java** ゲートウェイに問い合わせを行い、Java ゲートウェイはさらに JMX 管理 API を使用して対象のアプリケーションにリモートで問い合わせます。

詳細および設定については、Zabbix Java ゲートウェイ セクションを参照してください。

Warning:
Java ゲートウェイと監視対象の JMX アプリケーション間の通信は、ファイアウォールで遮断されていてはいけません。

Java アプリケーションのリモート JMX 監視を有効にする

Java アプリケーションには追加のソフトウェアをインストールする必要はありませんが、リモート JMX 監視をサポートさせるには、以下で指定するコマンドラインオプションを付けて起動する必要があります。

最低限の構成として、セキュリティを有効にせず、ローカルホスト上のシンプルな Java アプリケーションの監視をすぐに開始したい場合は、次のオプションを付けて起動します。

```
java \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.registry.ssl=false \
-jar /path/to/your/application.jar
```

これにより、Java はローカルホストからの受信 JMX 接続をポート 12345 で待ち受けるようになり、認証や SSL を要求しないよう設定されます。

別のインターフェースでの接続を許可したい場合は、-Djava.rmi.server.hostname パラメータにそのインターフェースの IP を設定してください。

より厳格なセキュリティを適用したい場合は、利用可能な Java オプションが他にも多数あります。たとえば、次の例では、より柔軟なオプションセットを使用してアプリケーションを起動し、ローカルホストだけでなく、より広いネットワークに公開します。

```
java \
-Djava.rmi.server.hostname=192.168.3.14 \
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=12345 \
-Dcom.sun.management.jmxremote.authenticate=true \
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-openjdk/management/jmxremote.password \
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-openjdk/management/jmxremote.access \
-Dcom.sun.management.jmxremote.ssl=true \
-Dcom.sun.management.jmxremote.registry.ssl=true \
```

```
-Djavax.net.ssl.keyStore=$YOUR_KEY_STORE \
-Djavax.net.ssl.keyStorePassword=$YOUR_KEY_STORE_PASSWORD \
-Djavax.net.ssl.trustStore=$YOUR_TRUST_STORE \
-Djavax.net.ssl.trustStorePassword=$YOUR_TRUST_STORE_PASSWORD \
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \
-jar /path/to/your/application.jar
```

これらの設定の大半（すべてではないにしても）は、`$JRE/lib/management/management.properties`（またはシステム上でそのファイルが存在する場所）で指定できます。

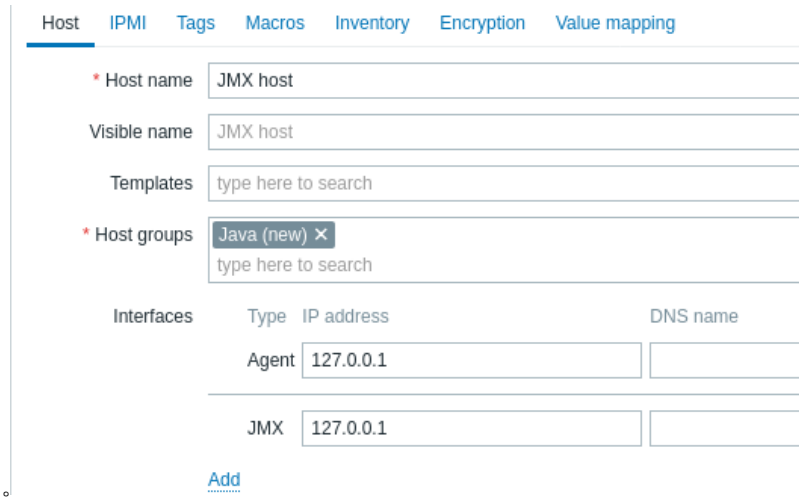
SSL を使用したい場合は、Java ゲートウェイがキーおよびトラストストアの場所を認識できるように、`-Djavax.net.ssl.*` オプションを Java ゲートウェイに追加して startup.sh スクリプトを変更する必要がある点に注意してください。

詳細については、Monitoring and Management Using JMXを参照してください。

Zabbix フロントエンドでの JMX インターフェースと item の設定

Java gateway が動作し、serveer が Java gateway の場所を認識し、リモート JMX 監視をサポートする Java アプリケーションが
 起動したら、Zabbix GUI でインタフェースと item を設定する必要があります。

JMX インターフェースの設定



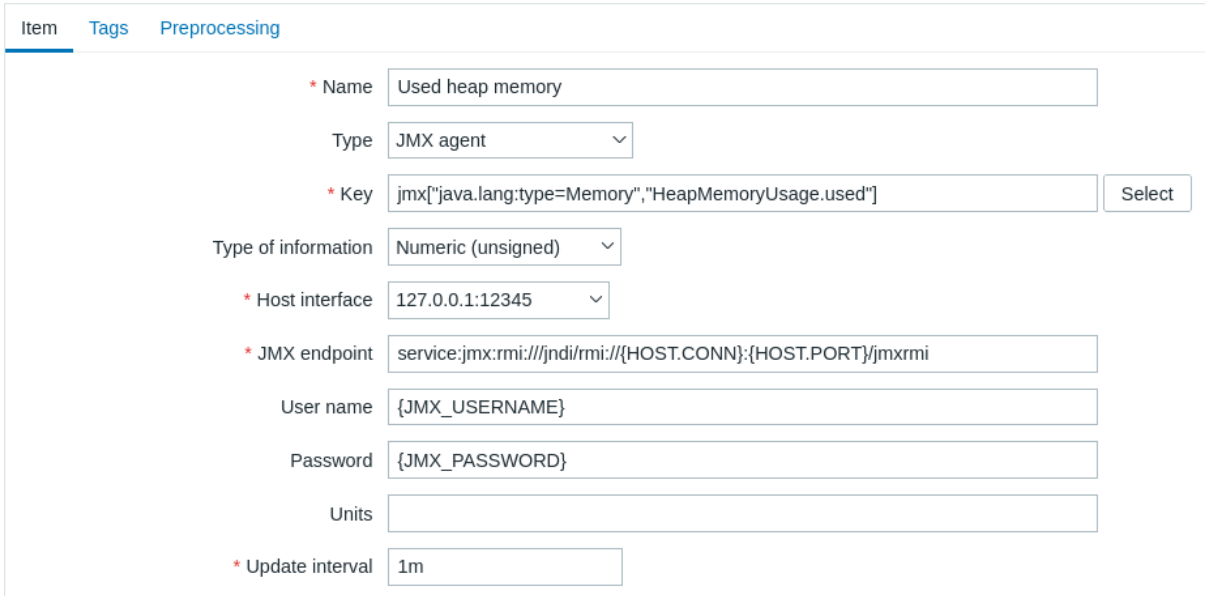
まず､対象のホストでJMX タイプのインターフェイスを作成します。

必須入力項目にはすべて赤いアスタリスクが付けられています。

JMX エージェントアイテムの追加

関心のある各 JMX カウンターについて、そのインターフェースに関連付けられた **JMX** エージェントアイテムを追加します。

以下のスクリーンショットのキーは `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]` です。



必須の入力フィールドには、赤いアスタリスクが付いています。

JMX アイテムで固有の情報が必要なフィールドは次のとおりです。

Type	ここで **JMX** エージェントを設定します。
Key	`jmx[]` アイテムキーには、3 つのパラメータが含まれます。 **object name** - MBean のオブジェクト名 **attribute name** - ドットで区切られたオプションの複合データフィールド名を含む MBean 属性名 **unique short description** - 同じオブジェクト名と属性名を持つ複数の JMX アイテムをホスト上で区別できるようにする一意の説明（任意） JMX アイテムキーの詳細については、以下を参照してください。 `jmx.discovery[]` ローレベルディスカバリ アイテムを使用して、MBean および MBean 属性を検出できます。
JMX endpoint	カスタム JMX エンドポイントを指定できます。JMX エンドポイントの接続パラメータが JMX インターフェースと一致していることを確認してください。これは、デフォルトの JMX エンドポイントで行われているように {HOST.*} マクロを使用することで実現できます。 {HOST.*} マクロ およびユーザーマクロがサポートされています。
User name	Java アプリケーションで認証を設定している場合は、ユーザー名（255 文字以内）を指定します。 ユーザーマクロがサポートされています。
Password	Java アプリケーションで認証を設定している場合は、パスワード（255 文字以内）を指定します。 ユーザーマクロがサポートされています。

”true” または”false” のいずれかとなる Boolean カウンターを監視したい場合は、情報の型として”Numeric (unsigned)” を指定し、前処理 タブで”Boolean to decimal” 前処理ステップを選択します。サーバーは、それぞれの Boolean 値を 1 または 0 として保存します。

JMX item キーの詳細

単純な属性

MBean のオブジェクト名は、Java アプリケーション内で定義する単なる文字列です。一方、属性名はより複雑になる場合があります。属性がプリミティブデータ型（整数、文字列など）を返す場合は、特に心配する必要はありません。キーは次のようになります。

```
jmx[com.example:Type=Hello,weight]
```

この例では、オブジェクト名は”com.example:Type=Hello”、属性名は”weight” で、返される値の型はおそらく” 数値（浮動小数）” になります。

複合データを返す属性

例えば、属性名が”apple” で、”weight” 、”color” などのパラメータを表すハッシュを返す場合、属性はより複雑になります。キーは次のようになります。

```
jmx[com.example:Type=Hello,apple.weight]
```

このように、属性名とハッシュ・キーは、ドット記号で区切られます。同じように、属性がネストした複合データを返す場合は、ドットで区切ります。

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

表形式データを返す属性

表形式データ属性は、1 つまたは複数の複合属性から構成されます。属性名パラメータにそのような
 属性が指定された場合、この item 値はその属性の完全な構造を JSON 形式で返します。
 表形式データ属性の中の個々の要素値は、プリプロセスによって取得することができます。

表形式データ属性の例:

Item 値:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

ドットに関する問題

ここまでは順調です。では、属性名またはハッシュキーにドット記号が含まれている場合はどうなるでしょうか。以下はその例です。

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

これは問題になります。属性名が単なる"all" ではなく "all.fruits" であることを、どのように Zabbix に伝えればよいのでしょうか。名前の一部であるドットと、属性名とハッシュキーを区切るドットを、どのように区別すればよいのでしょうか。

これは可能です。名前の一部であるドットをバックスラッシュでエスケープするだけです。

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

同様に、ハッシュキーにドットが含まれている場合も、それをエスケープします。

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

その他の問題

属性名内のバックスラッシュ文字は、エスケープする必要があります。

```
jmx[com.example:type=Hello,c:\\documents]
```

JMX アイテムキー内のその他の特殊文字の扱いについては、アイテムキー形式のセクション を参照してください。

以上で完了です。快適な JMX 監視を！

非プリミティブデータ型

toString() メソッドをオーバーライドする非プリミティブデータ型を返すカスタム MBean を扱うことができます。

JBoss EAP 6.4 でカスタムエンドポイントを使用する

カスタムエンドポイントを使用すると、デフォルトの RMI 以外のさまざまなトランスポートプロトコルで動作できます。

この可能性を示すために、例として JBoss EAP 6.4 の監視を設定してみましょう。まず、いくつか前提を置きます。

- すでに Zabbix Java ゲートウェイをインストールしていること。まだの場合は、documentation に従ってインストールできます。
- Zabbix サーバーと Java ゲートウェイは、プレフィックス /usr/local/ でインストールされていること
- JBoss はすでに /opt/jboss-eap-6.4/ にインストールされており、スタンドアロンモードで実行されていること
- これらすべてのコンポーネントは同じホスト上で動作しているものとします
- ファイアウォールおよび SELinux は無効化されていること（または適切に設定されていること）

zabbix_server.conf で簡単な設定を行います。

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

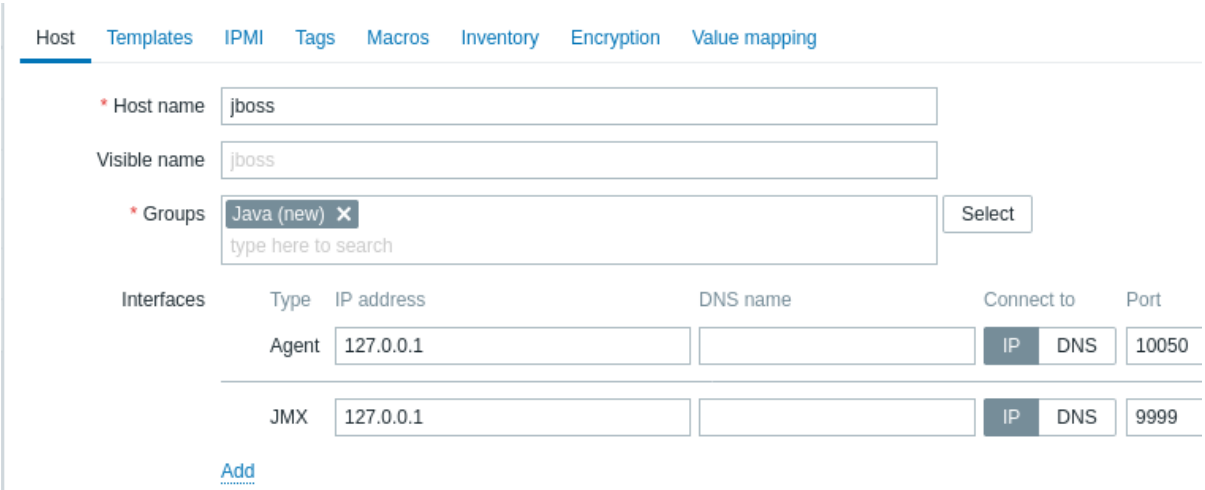
また、zabbix_java/settings.sh 設定ファイル（または zabbix_java_gateway.conf）では次のように設定します。

```
START_POLLERS=5
```

JBoss が標準の管理ポートで待ち受けていることを確認します。

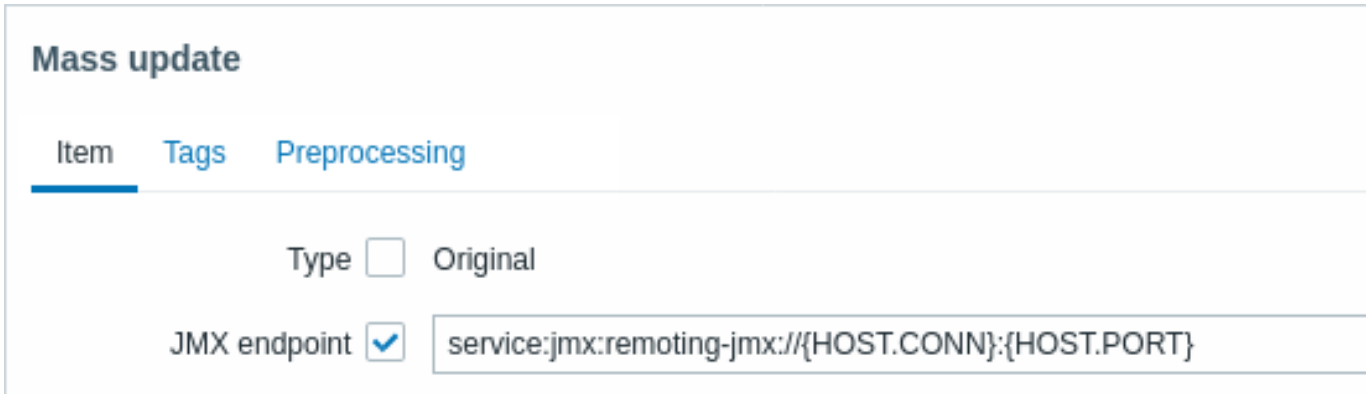
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

次に、Zabbix で JMX インターフェース 127.0.0.1:9999 を持つホストを作成します。



このバージョンの JBoss は RMI ではなく JBoss Remoting プロトコルを使用することが分かっているため、JMX テンプレート内のアイテムに対して JMX エンドポイントパラメータを一括更新し、次のように設定できます。

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



設定キャッシュを更新します。

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

最初はエラーが発生する場合があることに注意してください。

```
3. mc [root@centos7-dev]:/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN  com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885)
as stopped
2017-11-07 13:52:26.167 [main] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has
tarted
```

「Unsupported protocol: remoting-jmx」は、Java ゲートウェイが指定されたプロトコルの扱い方を認識していないことを意味します。これは、次の内容で ~/needed_modules.txt ファイルを作成することで修正できます。

```
jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

その後、次のコマンドを実行します。

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

これにより、Java ゲートウェイは jmx-remoting で動作するために必要なすべてのモジュールを利用できるようになります。あとは Java ゲートウェイを再起動し、しばらく待てば、すべて正しく実施されていれば、JMX 監視データが Zabbix に届き始めることを確認できます（参照: Latest data）。

14 ODBC 監視

概要

ODBC 監視は、Zabbix Web インターフェースのデータベースモニタアイテムタイプに対応します。

ODBC は、データベース管理システム（DBMS）にアクセスするための C プログラミング言語のミドルウェア API です。
ODBC の概念は Microsoft によって開発され、その後ほかのプラットフォームにも移植されました。

Zabbix は、ODBC でサポートされている任意のデータベースを照会できます。
そのために、Zabbix はデータベースへ直接接続するのではなく、ODBC で設定された ODBC インターフェースとドライバーを使用します。
これにより、さまざまな目的（たとえば、特定のデータベースキュー、使用統計などの確認）に応じて、異なるデータベースをより効率的に監視できます。

Zabbix は、最も一般的に使用されているオープンソースの ODBC API 実装の 1 つである unixODBC をサポートしています。

Attention:
ODBC チェックについては、既知の問題も参照してください。

unixODBC のインストール

unixODBC をインストールする推奨方法は、Linux オペレーティングシステムのデフォルトのパッケージリポジトリを使用することです。主要な Linux ディストリビューションの多くでは、unixODBC はデフォルトでパッケージリポジトリに含まれています。パッケージが利用できない場合は、ソースファイルを unixODBC のホームページから入手できます: http://www.unixodbc.org/download.html。

unixODBC をインストールするには、使用しているシステムに応じたパッケージマネージャーを使用してください:

```
##### Ubuntu/Debian システムの場合:
apt install unixodbc unixodbc-dev

##### RedHat/Fedora ベースのシステムの場合:
dnf install unixODBC unixODBC-devel

##### SUSE ベースのシステムの場合:
zypper in unixODBC-devel
```

Attention:
unixodbc-dev または unixODBC-devel パッケージは、unixODBC サポート付きで Zabbix をコンパイルするために必要です。ODBC サポートを有効にするには、Zabbix を次の設定オプションでコンパイルする必要があります:

--with-unixodbc[=ARG] # unixODBCパッケージに対してODBCドライバーを使用します。

unixODBC ドライバーのインストール

監視対象のデータベースに対して、unixODBC データベースドライバーをインストールする必要があります。
サポートされているデータベースとドライバーの一覧については、unixODBC のホームページを参照してください: http://www.unixodbc.org/drivers.html。

Note:
一部の Linux ディストリビューションでは、データベースドライバーがパッケージリポジトリに含まれています。

MySQL

MySQL unixODBC データベースドライバーをインストールするには、使用するシステムに応じたパッケージマネージャーを使用します。

```
##### Ubuntu/Debian システムの場合:
apt install odbc-mariadb

##### RedHat/Fedora ベースのシステムの場合:
dnf install mariadb-connector-odbc
```

```
##### SUSE ベースのシステムの場合:
zypper install mariadb-connector-odbc
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールする場合は、mysql-connector-odbc については MySQL documentation、mariadb-connector-odbc については MariaDB documentation を参照してください。

PostgreSQL

PostgreSQL unixODBC データベースドライバーをインストールするには、使用するシステムに応じたパッケージマネージャーを使用します。

```
##### Ubuntu/Debian システムの場合:
apt install odbc-postgresql

##### RedHat/Fedora ベースのシステムの場合:
dnf install postgresql-odbc

##### SUSE ベースのシステムの場合:
zypper install psqlODBC
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールする場合は、PostgreSQL documentationを参照してください。

Oracle

unixODBC データベースドライバーをインストールするには、Oracle documentationを参照してください。

MSSQL

MSSQL unixODBC データベースドライバーをインストールするには、使用しているシステムに応じたパッケージマネージャーを使用します。

```
##### Ubuntu/Debian システムの場合:
apt install tdsodbc

##### RedHat/Fedora ベースのシステムの場合 (EPEL パッケージ: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### SUSE ベースのシステムの場合:
zypper install libtdsodbc0
```

パッケージマネージャーを使用せずにデータベースドライバーをインストールする場合は、FreeTDS user guideを参照してください。

unixODBC の設定

unixODBC を設定するには、odbcinst.ini ファイルと odbc.ini ファイルを編集する必要があります。
これらのファイルの場所は、次のコマンドを実行して確認できます。

```
odbcinst -j
```

コマンドの実行結果には、次のような情報が含まれているはずです。

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

odbcinst.ini

odbcinst.ini ファイルには、インストールされている ODBC データベースドライバーが一覧表示されます。
odbcinst.ini が存在しない場合は、手動で作成する必要があります。

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parameter	Description
TEST_MYSQL	データベースドライバー名。

Parameter	Description
Description	データベースドライバーの説明。
Driver	データベースドライバーライブラリの場所。
FileUsage	データベースドライバーが、ローカルファイルへのアクセス機能のサポートなしでデータベースサーバーへの接続をサポートするか (0)、ファイルからのデータ読み取りをサポートするか (1)、ファイルへのデータ書き込みをサポートするか (2) を決定します。
Threading	スレッドのシリアル化レベル。PostgreSQL でサポートされています。 1.6 以降、ドライバーマネージャーがスレッドサポート付きでビルドされている場合は、別のドライバーエントリを追加できます。

odbc.ini

`odbc.ini` ファイルは、データソースの設定に使用されます。
サポートされるパラメータの一覧はデータベースドライバーによって異なることに注意してください（たとえば、Oracle データベースでは Server の代わりに ServerName を使用する場合があります）。

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Parameter	Description
TEST_MYSQL	データソース名（DSN）。
Description	データソースの説明。
Driver	データベースドライバー名（`odbcinst.ini` で指定）。
Server	データベースサーバーの IP/DNS。
User	接続に使用するデータベースユーザー。
Password	データベースユーザーのパスワード。
Port	データベース接続ポート。
Socket	データベース接続ソケット。
Database	データベース名。

その他の設定パラメータオプションについては、MySQL documentation を参照してください。

PostgreSQL 用の `odbc.ini` ファイルには、追加のパラメータを含めることができます。

```
[TEST_PSQL]
Description=PostgreSQL Test Database
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=7.4+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
SSLmode=Require
ConnSettings=
```

パラメータ	説明
ReadOnly	データベース接続で読み取り操作（SELECT クエリ）のみを許可し、変更操作（INSERT、UPDATE、DELETE 文）を制限するかどうかを指定します。データを変更しないままにしておく必要があるシナリオで役立ちます。
Protocol	PostgreSQL バックエンドプロトコルのバージョンです（SSL 接続を使用する場合は無視されます）。
ShowOidColumn	SQLColumns に Object ID（OID）を含めるかどうかを指定します。
FakeOidIndex	OID に対して疑似的な一意インデックスを作成するかどうかを指定します。
RowVersioning	行の更新を試みている間に、他のユーザーによってデータが変更されたかどうかをアプリケーションが検出できるようにするかどうかを指定します。このパラメータを使用すると、行を更新する際に WHERE 句ですべての列を個別に指定する必要がなくなるため、更新処理が高速化される場合があります。
ShowSystemTables	SQLTables において、データベースドライバーがシステムテーブルを通常のテーブルとして扱うかどうかを指定します。アクセシビリティの向上に役立ち、システムテーブルを参照できるようになります。
Fetch	ドライバーが SELECT 文を処理するために declare cursor/fetch を自動的に使用し、100 行のキャッシュを維持するかどうかを指定します。
BoolsAsChar	Boolean 型のマッピングを制御します。 ”Yes” に設定すると、Boolean 値は SQL_CHAR にマッピングされ、それ以外の場合は SQL_BIT にマッピングされます。
SSLmode	接続の SSL モードを指定します。
ConnSettings	接続時にバックエンドへ送信される追加設定です。

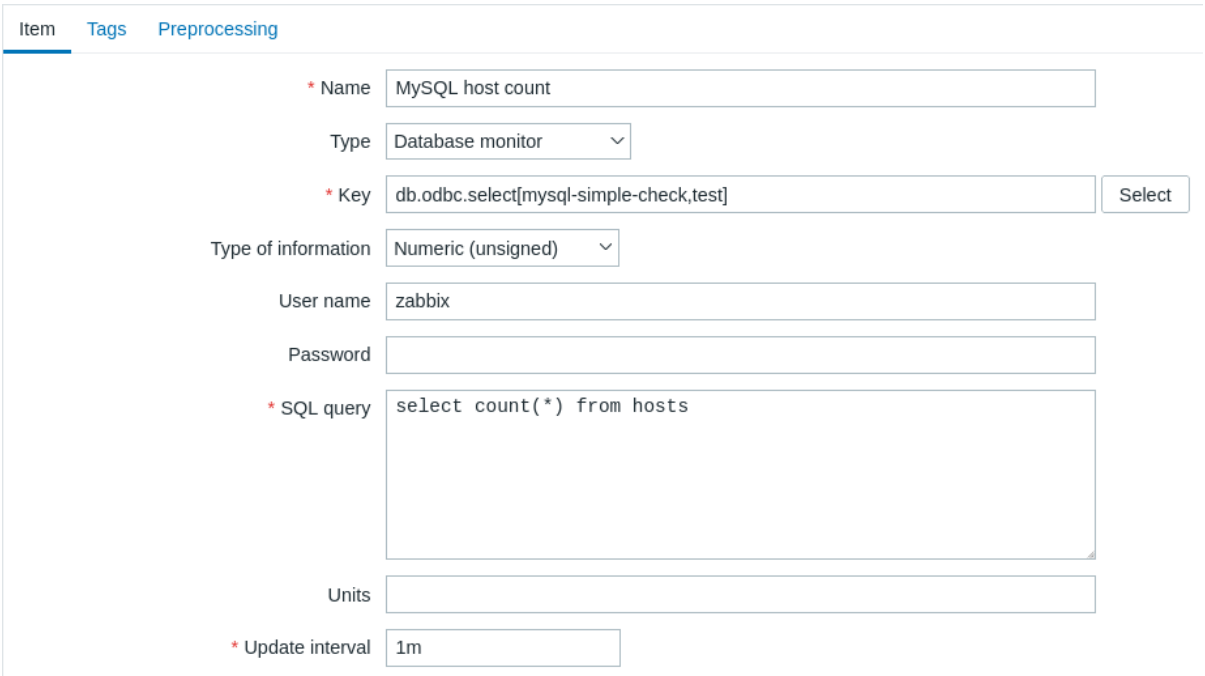
ODBC 接続のテスト

ODBC 接続が正常に動作しているかどうかをテストするには、isql ユーティリティ（unixODBC パッケージに含まれています）を使用できます。

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Zabbix Web インターフェースでのアイテム設定

データベース監視のアイテムを設定します。



必須の入力フィールドには、赤いアスタリスクが付いています。

データベース監視アイテムでは、以下を指定する必要があります。

Type	ここで「Database monitor」を選択します。
Key	サポートされているアイテムキーのいずれかを入力します。 **db.odbc.select[]** - このアイテムは 1 つの値を返します（SQL クエリ結果の最初の行の最初の列）。 **db.odbc.get[]** - このアイテムは複数の行/列を JSON 形式で返します。 **db.odbc.discovery[]** - このアイテムはローレベルディスカバリデータを返します。
User name	データベースのユーザー名を入力します（最大 255 文字）。 このパラメータは、データベースのユーザー名が `odbc.ini` ファイルで指定されている場合は省略可能です。 接続文字列を使用していて、*User name* フィールドが空でない場合は、`UID=<user>` として接続文字列に追加されます。
Password	データベースのユーザーパスワードを入力します（最大 255 文字）。 このパラメータは、パスワードが `odbc.ini` ファイルで指定されている場合は省略可能です。 接続文字列を使用していて、*Password* フィールドが空でない場合は、`PWD=<password>` として接続文字列に追加されます。 このフィールドでは特殊文字を使用できます。 パスワードは、たとえば `UID=<username>;PWD=P?;)*word` のように、ユーザー名の後に接続文字列へ追加されます。 生成される文字列をテストするには、次のコマンドを実行できます。 `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	SQL クエリを入力します。 `db.odbc.select[]` では、クエリは 1 つの値のみを返す必要があることに注意してください。
Type of information	ここでクエリによって返される情報の型を選択します。 情報の型が正しく選択されていない場合、アイテムは未サポートになります。

重要な注意事項

- サーバーまたはプロキシの設定で *odbc poller* プロセスが起動されていない場合、データベース監視アイテムは未サポートになります。ODBC poller を有効にするには、Zabbix server 設定ファイル、またはプロキシによって実行されるチェックの場合は Zabbix proxy 設定ファイルで、`StartODBCPollers` パラメータを設定してください。
- アイテム設定 フォームの *Timeout* パラメータ値は、ODBC のログインタイムアウトおよびクエリ実行タイムアウトとして使用されます。インストールされている ODBC ドライバーがこれらをサポートしていない場合、これらのタイムアウト設定は無視される可能性があることに注意してください。
- SQL コマンドは、`select` 文を使用するクエリと同様に、結果セットを返す必要があります。クエリ構文は、それを処理する RDBMS によって異なります。ストアドプロシージャへのリクエスト構文は、`call` キーワードで開始する必要があります。

アイテムキーの詳細

山かっこなしのパラメータは必須です。山かっこ **< >** で示されたパラメータは任意です。

db.odbc.select[< 一意の短い説明 >,<dsn>,< 接続文字列 >]

 1 つの値、つまり SQL クエリ結果の最初の行の最初の列を返します。
 戻り値: SQL クエリに応じます。

パラメータ:

- 一意の短い説明 - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（`odbc.ini` で指定）;
- 接続文字列 - 接続文字列（ドライバー固有の引数を含む場合があります）。

コメント:

- `dsn` と 接続文字列はオプションのパラメータですが、少なくともどちらか一方は必須です。両方が定義されている場合、`dsn` は無視されます。
- クエリが複数の列を返す場合は、最初の列のみが読み取られます。クエリが複数の行を返す場合は、最初の行のみが読み取られます。

db.odbc.get[< 一意の短い説明 >,<dsn>,< 接続文字列 >]

 SQL クエリの結果を JSON 配列に変換します。
 戻り値: *JSON object*。

パラメータ:

- 一意の短い説明 - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（`odbc.ini` で指定）;
- 接続文字列 - 接続文字列（ドライバー固有の引数を含む場合があります）。

コメント:

- dsn と 接続文字列はどちらもオプションのパラメータですが、少なくともどちらか一方は必須です。両方が定義されている場合、dsn は無視されます。
- JSON 形式で複数の行/列が返される場合があります。このアイテムは、1 回のシステムコールですべてのデータを収集するマスターアイテムとして使用でき、個々の値を抽出するために、依存アイテムで JSONPath 前処理を使用できます。詳細については、ローレベルディスカバリで使用される、返却形式の例を参照してください。

例:

```
# MySQL ODBC driver 5 の接続:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[< 一意の短い説明 >,<dsn>,< 接続文字列 >]

 SQL クエリの結果を JSON 配列に変換し、ローレベルディスカバリで使用します。クエリ結果の列名は、検出されたフィールド値と組み合わせたローレベルディスカバリマクロ名に変換されます。これらのマクロは、アイテム、トリガーなどのプロトタイプの作成に使用できます。
 戻り値: *JSON object*。

パラメータ:

- 一意の短い説明 - アイテムを識別するための一意の短い説明（トリガーなどで使用）;
- **dsn** - データソース名（odbc.ini で指定）;
- 接続文字列 - 接続文字列（ドライバー固有の引数を含む場合があります）。

コメント:

- dsn と 接続文字列はどちらもオプションのパラメータですが、少なくとも一方は必須です。両方が定義されている場合、dsn は無視されます。

エラーメッセージ

ODBC のエラーメッセージは、詳細情報を提供するためにフィールドごとに構成されています。例えば、エラーメッセージは次のようになります。

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "Cannot execute ODBC query" - Zabbix メッセージ
- "[SQL_ERROR]" - ODBC 戻りコード
- "[42601]" - SQLState
- "[7]" - ネイティブエラーコード
- "[ERROR: syntax error at or near ";"; Error while executing the query]" - ネイティブエラーメッセージ

エラーメッセージの長さは 2048 バイトに制限されているため、メッセージが切り詰められる場合があることに注意してください。ODBC 診断レコードが複数ある場合、Zabbix は長さの制限が許す範囲で、それらを（| で区切って）連結しようとします。

15 依存アイテム

概要

1 つのアイテムが一度に複数のメトリクスを収集する場合や、関連するメトリクスを同時に収集したほうが合理的な場合があります。例えば、次のようなケースです。

- 個々のコアの CPU 使用率
- 受信/送信/合計のネットワークトラフィック

このような一括メトリクス収集と、複数の関連アイテムでの同時利用を可能にするために、Zabbix は依存アイテムをサポートしています。依存アイテムは、1 回のクエリで同時にデータを収集するマスターアイテムに依存します。マスターアイテムの新しい値は、自動的に依存アイテムの値にも反映されます。依存アイテムには、マスターアイテムとは異なる更新間隔を設定できません。

Zabbix の前処理オプションを使用すると、マスターアイテムのデータから依存アイテムに必要な部分を抽出できます。

前処理は、前処理ステップを実行するワーカースレッドとともに、preprocessing manager プロセスによって管理されます。前処理が設定されたすべての値（Zabbix 7.4.1 より前では、すべての値）は、異なるデータ収集プロセスから受信された後、履歴キャッシュに追加される前に前処理マネージャーを通過します。データ収集プロセス（poller、trapper など）と前処理プロセスの間では、ソケットベースの IPC 通信が使用されます。

前処理ステップの実行と依存アイテムの処理は、Zabbix サーバー、または（ホストがプロキシによって監視されている場合は）Zabbix プロキシによって行われます。

任意のタイプのアイテムは、依存アイテムを含めて、マスターアイテムとして設定できます。既存の依存アイテムの値からさらに小さな部分を抽出するために、依存アイテムの追加レベルを使用することもできます。

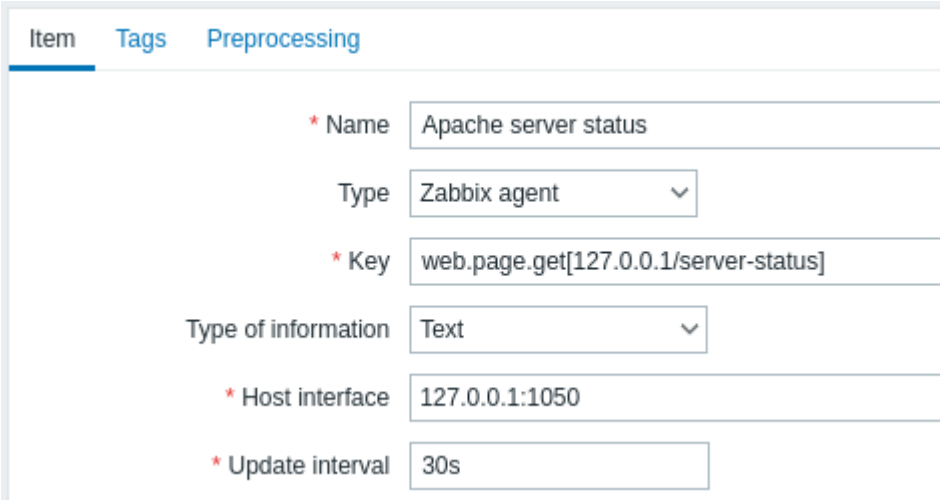
制限事項

- 同一ホスト（テンプレート）内の依存関係のみ許可されています
- アイテムプロトタイプは、同一ホスト上の別のアイテムプロトタイプまたは通常のアイテムに依存できます
- テンプレートのマスターアイテムを持つホスト上の依存アイテムは、XML にエクスポートされません

アイテムの設定

依存アイテムは、データについてマスターアイテムに依存します。そのため、まず マスターアイテムを設定しておく（または存在している）必要があります。

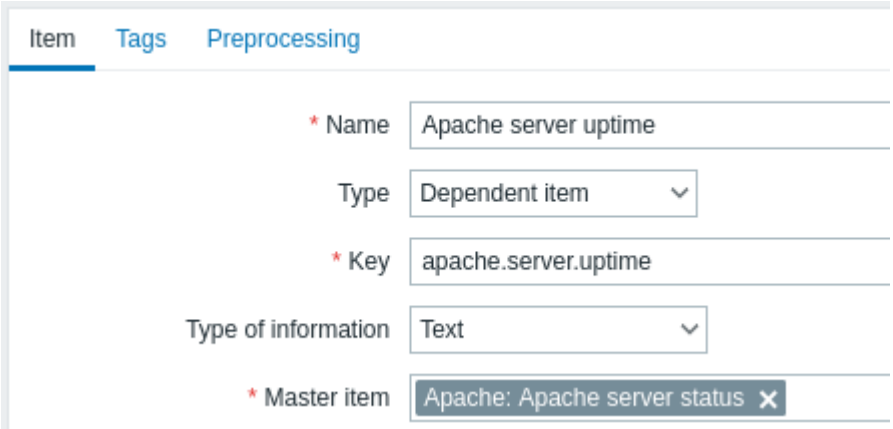
- データ収集 → ホストに移動します
- ホストの行にある アイテムをクリックします
- アイテムの作成をクリックします
- フォームにアイテムのパラメータを入力します



必須入力フィールドには、赤いアスタリスクが付いています。

追加をクリックして、マスターアイテムを保存します。

次に、依存アイテムを設定できます。

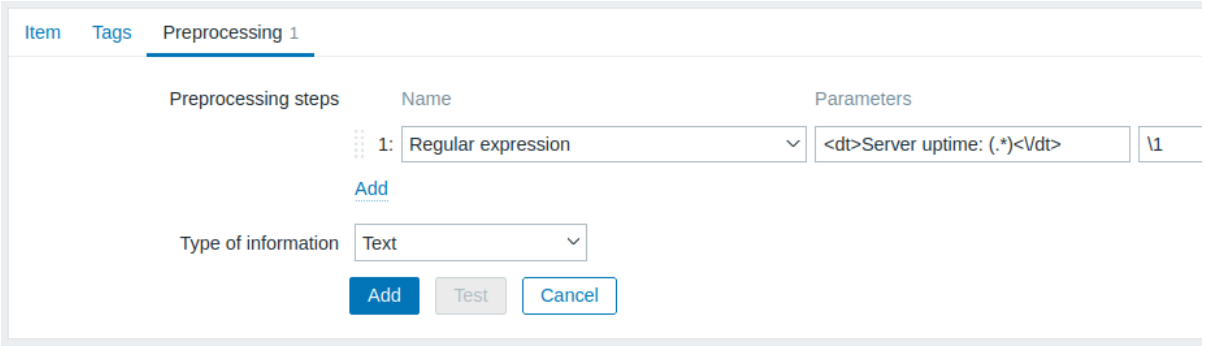


必須入力フィールドには、赤いアスタリスクが付いています。

依存アイテムで特定の情報の入力が必要なフィールドは次のとおりです。

タイプ	ここで 依存アイテムを選択します。
キー	アイテムを識別するために使用されるキーを入力します。
マスターアイテム	マスターアイテムを選択します。マスターアイテムの値は、依存アイテムの値を設定するために使用されます。
保存前処理	保存されるデータの形式に対応する情報のタイプを選択します。

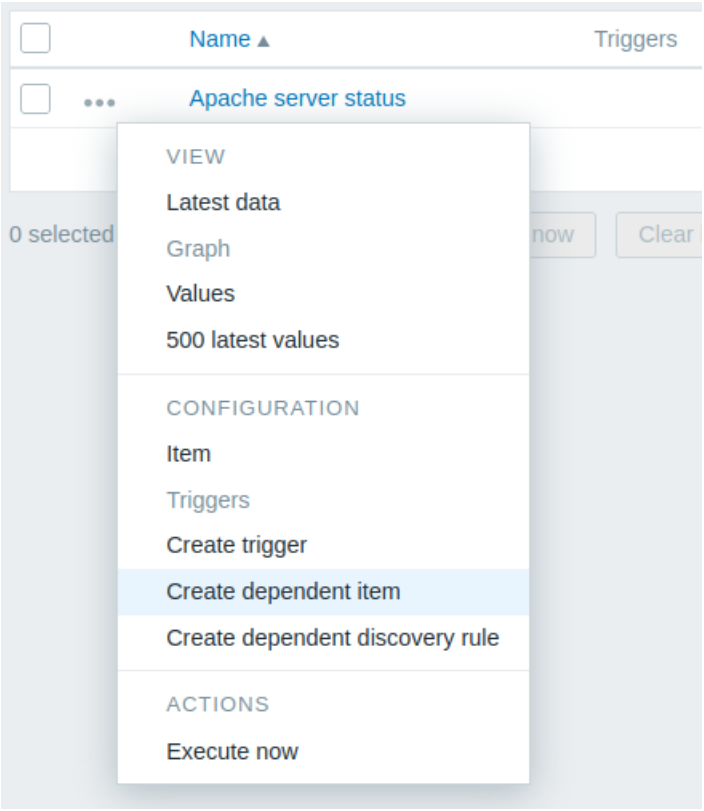
必要なマスターアイテム値の部分を抽出するために、アイテム値の保存前処理を使用できます。



保存前処理を行わない場合、依存アイテムの値はマスターアイテムの値とまったく同じになります。

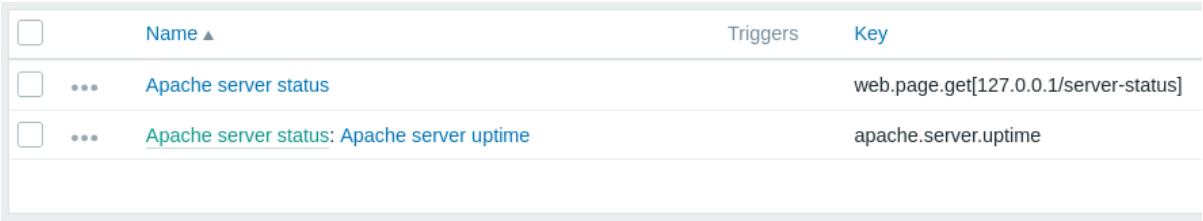
追加をクリックして、依存アイテムを保存します。

依存アイテムをよりすばやく作成するショートカットとして、アイテム一覧で ••• ボタンをクリックし、依存アイテムの作成を選択する方法もあります。



表示

アイテムリストでは、依存アイテムはマスターアイテム名をプレフィックスとして表示されます。



マスターアイテムが削除された場合、そのアイテムに依存する依存アイテムもすべて削除されます。

16 HTTP エージェント

概要

このアイテムタイプでは、HTTP/HTTPS プロトコルを使用したデータのポーリングが可能です。
また、Zabbix senderユーティリティまたは Zabbix sender protocol（Zabbix サーバーまたはプロキシへデータを送信するため）を使用したトラップ、あるいは`history.push` API メソッド（Zabbix サーバーへデータを送信するため）の使用も可能です。

HTTP アイテムのチェックは Zabbix サーバーによって実行されます。ただし、ホストが Zabbix プロキシによって監視されている場合、HTTP アイテムのチェックはプロキシによって実行されます。

HTTP アイテムのチェックでは、監視対象のホスト上でエージェントを実行する必要はありません。

HTTP エージェントは HTTP と HTTPS の両方をサポートしています。Zabbix は必要に応じてリダイレクトを追跡します (以下の *Follow redirects* オプションを参照)。リダイレクトの最大回数は 10 にハードコードされています（cURL オプション CURLOPT_MAXREDIRS を使用)。

Attention:
Zabbix サーバー/プロキシは、あらかじめ cURL（libcurl）サポート付きで設定されている必要があります。

HTTP チェックは非同期で実行されます。つまり、他のチェックを開始する前に、あるリクエストへの応答を受信している必要はありません。DNS の名前解決も同様に非同期です。

非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPoller で定義)。

非同期 HTTP エージェントポーラーの数は、StartHTTPAgentPollers パラメータで定義されます。

永続接続に関する cURL の機能は、Zabbix 7.0 以降の HTTP エージェントチェックで追加されました。

設定

HTTP アイテムを設定するには、以下のようにします。

- データ収集 → ホストに移動します
- ホストの行にある アイテムをクリックします
- アイテムの作成をクリックします
- フォームにアイテムのパラメータを入力します

Item | Tags | Preprocessing

* Name: HTTP agent item

Type: HTTP agent

* Key: http_value_search [Select]

Type of information: Numeric (unsigned)

* URL: http://localhost:9200/str/values/_search [Parse]

Query fields:

Name		Value	
scroll	⇒	10s	Remove

Add

Request type: POST

Request body type: Raw data | JSON data | XML data

Request body:

```
{
  "query":{
    "bool":{
     "must":{
      "match":{
        "itemid":28275
}
```

Headers:

Name		Value	
name	⇒	value	Remove

Add

Required status codes: 200

Follow redirects: ☑

Retrieve mode: Body | Headers | Body and headers

Convert to JSON: ☐

HTTP proxy: [protocol://][user[:password]@]proxy.example.com[:port]

HTTP authentication: None

SSL verify peer: ☐

SSL verify host: ☐

SSL certificate file:

SSL key file:

SSL key password:

Host interface: 127.0.0.1:10050

Units:

* Update interval: 1m

Custom intervals:

Type	Interval	Period	Action
Flexible | Scheduling	50s	1-7,00:00-24:00	Remove

Add

* Timeout: Global | Override 3 Timeouts

* History: Do not store | Store up to 31d

* Trends: Do not store | Store up to 365d

Value mapping: type here to search [Select]

Enable trapping: ☐

Populates host inventory field: -None-

Description:

Enabled: ☑

Add | Test | Cancel

必須入力フィールドには、すべて赤いアスタリスクが付いています。

HTTP アイテムで固有の情報が必要なフィールドは次のとおりです。

Parameter	Description
Type	ここでは **HTTP** エージェントを選択します。
Key	一意のアイテムキーを入力します。
URL	接続してデータを取得する URL です。例: https://www.example.com http://www.example.com/download ドメイン名は Unicode 文字で指定できます。HTTP チェックの実行時に自動的に punycode へ変換され、ASCII になります。 *Parse* ボタンを使用すると、オプションのクエリフィールド（例: ?name=Admin&password=mypassword）を URL から分離し、属性と値を *Query fields* に移動して自動的に URL エンコードできます。 2048 文字までに制限されています。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_URL cURL オプションを設定します。
Query fields	URL 用の変数です（上記参照）。 属性と値のペアとして指定します。 値は自動的に URL エンコードされます。マクロからの値は展開された後、自動的に URL エンコードされます。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_URL cURL オプションを設定します。
Request type	リクエストメソッドの種類を選択します: *GET*、*POST*、*PUT*、*HEAD*
Request body type	リクエストボディの種類を選択します: **Raw data** - カスタム HTTP リクエストボディ。マクロは置換されますが、エンコードは行われません **JSON data** - JSON 形式の HTTP リクエストボディ。マクロは文字列、数値、true、false として使用できます。文字列として使用するマクロは二重引用符で囲む必要があります。マクロからの値は展開された後、自動的にエスケープされます。”Content-Type” がヘッダーで指定されていない場合、デフォルトで”Content-Type: application/json” になります **XML data** - XML 形式の HTTP リクエストボディ。マクロはテキストノード、属性、または CDATA セクションで使用できます。マクロからの値は展開された後、テキストノードおよび属性内で自動的にエスケープされます。”Content-Type” がヘッダーで指定されていない場合、デフォルトで”Content-Type: application/xml” になります 注:*XML data* を選択するには libxml2 が必要です。
Request body	リクエストボディを入力します。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。
Headers	リクエスト実行時に送信されるカスタム HTTP ヘッダーです。 属性と値のペアとして指定します。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_HTTPHEADER cURL オプションを設定します。
Required status codes	期待される HTTP ステータスコードの一覧です。Zabbix が一覧にないコードを受信した場合、そのアイテムは未サポートになります。空の場合、チェックは実行されません。 例: 200,201,210-299 一覧内でサポートされるマクロ: ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLINFO_RESPONSE_CODE cURL オプションを使用します。
Follow redirects	HTTP リダイレクトに従うには、このチェックボックスをオンにします。 これは CURLOPT_FOLLOWLOCATION cURL オプションを設定します。
Retrieve mode	取得する必要があるレスポンスの部分を選択します: **Body** - ボディのみ **Headers** - ヘッダーのみ **Body and headers** - ボディとヘッダー

Parameter	Description

Convert to JSON	*Retrieve mode* が *Body* に設定されていない場合、ヘッダーは”header” キーの下に属性と値のペアとして保存されます。
	’Content-Type: application/json’ が検出された場合、ボディはオブジェクトとして保存され、それ以外の場合は文字列として保存されます。例:
	`{`
	`  "header": {`
	`  "<key>": "<value>",`
	`  "<key2>": "<value>"`
	`  },`
	`  "body": <body>`
	`}`

HTTP proxy	使用する HTTP プロキシを `[protocol://][username[:password]@]proxy.example.com[:port]` の形式で指定できます。 オプションの `protocol://` プレフィックスを使用して、代替のプロキシプロトコル（例: https、socks4、socks5）を指定できます（documentation を参照。プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました）。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。誤ったプロトコルを指定すると接続に失敗し、アイテムは未サポートになります。 デフォルトでは 1080 ポートが使用されます。 指定した場合、プロキシは http_proxy、HTTPS_PROXY などのプロキシ関連環境変数を上書きします。指定しない場合、プロキシはプロキシ関連環境変数を上書きしません。入力した値はそのまま渡され、妥当性チェックは行われません。 注 HTTP プロキシでは単純認証のみサポートされます。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_PROXY cURL オプションを設定します。
HTTP authentication	認証オプションを選択します: **None** - 認証を使用しません; **Basic** - Basic 認証を使用します; **NTLM** - NTLM (Windows NT LAN Manager) 認証を使用します; **Kerberos** - Kerberos 認証を使用します（あわせてConfiguring Kerberos with Zabbix も参照してください）; **Digest** - Digest 認証を使用します。 これは CURLOPT_HTTPAUTH cURL オプションを設定します。
User name	ユーザー名を入力します（255 文字まで）。 このフィールドは、*HTTP authentication* が Basic、NTLM、Kerberos、または Digest に設定されている場合に使用できます。ユーザーマクロおよびローレベルディスカバリマクロをサポートします。
Password	ユーザーパスワードを入力します（255 文字まで）。 このフィールドは、*HTTP authentication* が Basic、NTLM、Kerberos、または Digest に設定されている場合に使用できます。ユーザーマクロおよびローレベルディスカバリマクロをサポートします。
SSL verify peer	Web サーバーの SSL 証明書を検証するには、このチェックボックスをオンにします。サーバー証明書は、システム全体の認証局（CA）の場所から自動的に取得されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータ SSLCALocation を使用して上書きできます。 これは CURLOPT_SSL_VERIFYPEER cURL オプションを設定します。
SSL verify host	Web サーバー証明書の Common Name フィールドまたは Subject Alternate Name フィールドが一致することを検証するには、このチェックボックスをオンにします。 これは CURLOPT_SSL_VERIFYHOST cURL オプションを設定します。
SSL certificate file	クライアント認証に使用する SSL 証明書ファイルの名前です。証明書ファイルは PEM[1] 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は、SSL key file フィールドは空のままにしてください。鍵が暗号化されている場合は、SSL key password フィールドにパスワードを指定します。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ SSLCertLocation で指定します。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_SSLCERT cURL オプションを設定します。

Parameter	Description
SSL key file	クライアント認証に使用する SSL 秘密鍵ファイルの名前です。秘密鍵ファイルは PEM[1] 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ SSLKeyLocation で指定します。 サポートされるマクロ: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_SSLKEY cURL オプションを設定します。
SSL key password	SSL 秘密鍵ファイルのパスワードです。 サポートされるマクロ: ユーザーマクロ、ローレベルディスカバリマクロ。 これは CURLOPT_KEYPASSWD cURL オプションを設定します。
Timeout	Zabbix は、URL の処理に設定された時間（1～600 秒）を超えて費やしません。実際には、このパラメータは URL への接続を確立する最大時間と、HTTP リクエストを実行する最大時間を定義します。したがって、Zabbix は 1 回のチェックに対して 2 x *Timeout* 秒を超えて費やしません。 これは CURLOPT_TIMEOUT cURL オプションを設定します。 *Timeout* パラメータの詳細については、一般的なアイテム属性 を参照してください。
Enable trapping	このチェックボックスをオンにすると、このアイテムはトラッパーアイテムとしても機能し、Zabbix sender ユーティリティまたは Zabbix sender protocol を使用して Zabbix サーバーまたはプロキシに送信されたデータ、あるいは`history.push` API メソッドを使用して Zabbix サーバーに送信されたデータを受け付けます。データ送信の詳細については、トラッパーアイテム を参照してください。
Allowed hosts	*Enable trapping* チェックボックスがオンの場合にのみ表示されます。 カンマ区切りの IP アドレス一覧で、必要に応じて CIDR 表記または DNS 名も指定できます。 指定した場合、ここに一覧されたホストからの受信接続のみが許可されます。 IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。 '0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。 "IPv4-compatible IPv6 addresses"（0000::/96 プレフィックス）はサポートされていますが、RFC4291 により非推奨であることに注意してください。 例: 127.0.0.1, 192.168.1.0/24, 192.168.3.1-255, 192.168.1-10.1-255, ::1,2001:db8::/32, mysqlserver1, zabbix.example.com, {HOST.HOST} このフィールドではスペースおよびuser macros を使用できます。 このフィールドではホストマクロ {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} を使用できます。

Note:
HTTP proxy フィールドを空のままにした場合、HTTP プロキシを使用する別の方法として、プロキシ関連の環境変数を設定できます。
HTTP の場合 - Zabbix サーバーユーザーに対して `http_proxy` 環境変数を設定します。例: `http_proxy=http://proxy_ip:proxy_port`。
HTTPS の場合 - `HTTPS_PROXY` 環境変数を設定します。例: `HTTPS_PROXY=http://proxy_ip:proxy_port`。詳細は、シェルコマンド *# man curl* を実行して確認できます。

Attention:
[1] Zabbix は証明書ファイルおよび秘密鍵ファイルとして PEM 形式のみをサポートします。証明書および秘密鍵のデータが PKCS #12 形式のファイル（通常は拡張子 *.p12 または *.pfx）にある場合は、次のコマンドを使用してそこから PEM ファイルを生成できます。

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

例

例 1

Elasticsearch などのサービスからデータを取得するための、シンプルな GET リクエストを送信します。

- URL: `localhost:9200/?pretty` で GET アイテムを作成します
- レスポンスを確認します。

```
{
  "name" : "YQ2VAY-",
  "cluster_name" : "elasticsearch",
```

```
  "cluster_uuid" : "kH4CYqh5QfqgeTsjh2F9zg",
  "version" : {
    "number" : "6.1.3",
    "build_hash" : "af51318",
    "build_date" : "2018-01-26T18:22:55.523Z",
    "build_snapshot" : false,
    "lucene_version" : "7.1.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You know, for search"
}
```

- 次に、JSONPath の前処理ステップを使用してバージョン番号を抽出します。
  `$.version.number`

例 2

Elasticsearch などのサービスからデータを取得するために、シンプルな POST リクエストを送信します。

- 次の URL で POST アイテムを作成します: `http://localhost:9200/str/values/_search?scroll=10s`
- プロセッサー負荷（コアあたりの 1 分平均）を取得するには、次の POST 本文を設定します

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- 受信内容:

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
    "_shards": {
        "total": 5,
        "successful": 5,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": 1,
        "max_score": 1.0,
        "hits": [{
            "_index": "dbl",
            "_type": "values",
            "_id": "dqX9VWEBV6sEKSMyk6sw",
            "_score": 1.0,
            "_source": {
                "itemid": 28275,
                "value": "0.138750",
```

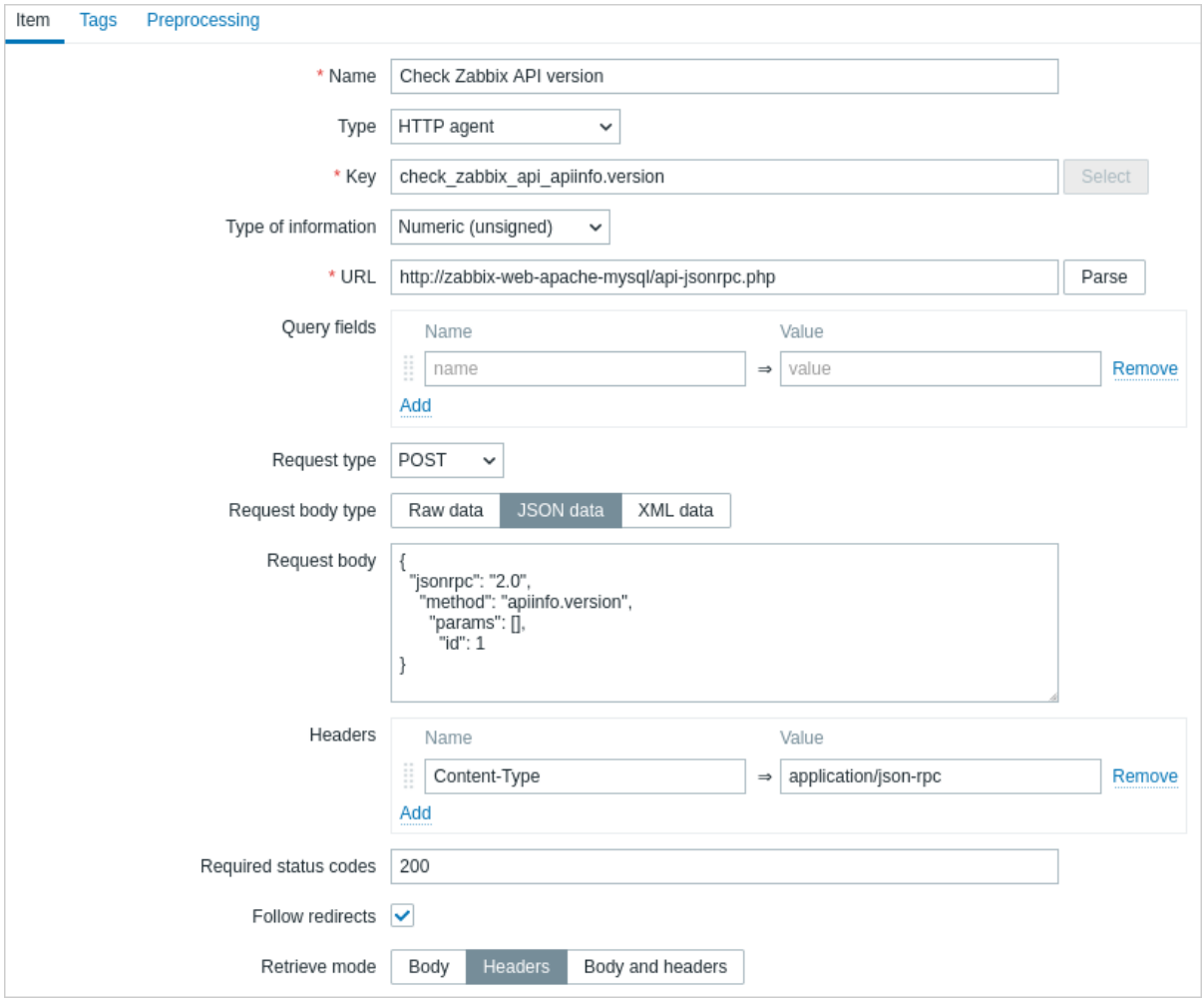
```
                "clock": 1517566136,
                "ns": 25388713,
                "ttl": 604800
            }
        }]
    }
}
```

- 次に、JSONPath の前処理ステップを使用してアイテムの値を取得します: `$.hits.hits[0]._source.value`

例 3

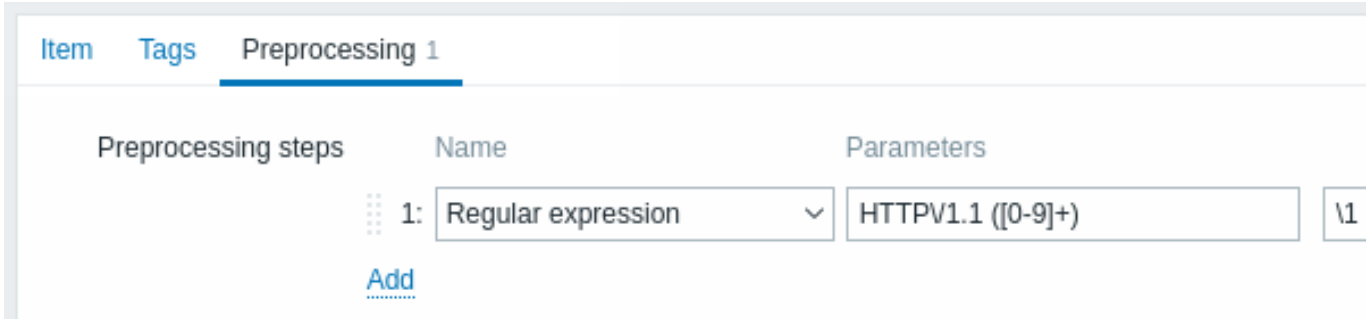
apiinfo.version を使用して、Zabbix API が稼働しているかどうかを確認します。

- アイテムの設定:

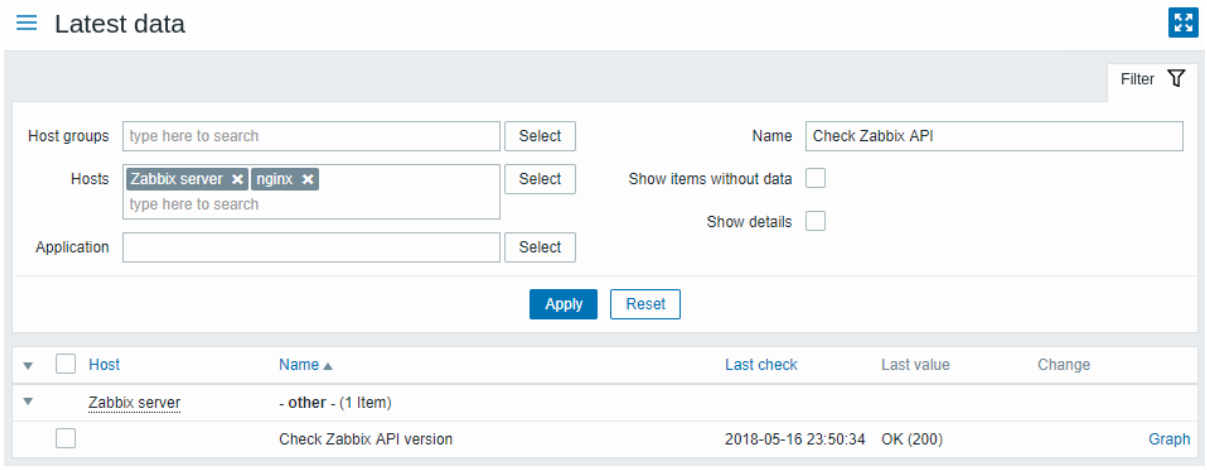


POST メソッドを JSON データとともに使用し、リクエストヘッダーを設定して、ヘッダーのみを返すよう指定している点に注意してください。

- HTTP コードを取得するための正規表現によるアイテム値の前処理:



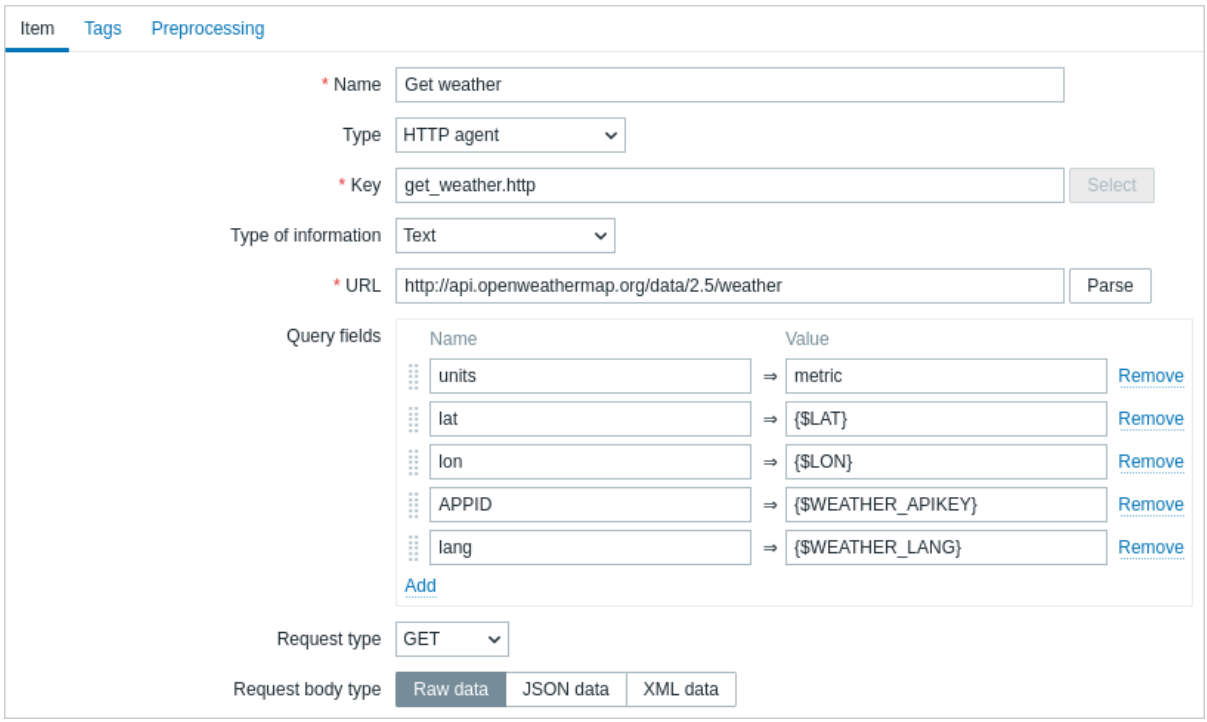
- 最新データで結果を確認:



例 4

Openweathermap の公開サービスに接続して天気情報を取得します。

- 単一の JSON で一括データ収集を行うためのマスターアイテムを設定します。



クエリフィールドでマクロが使用されている点に注意してください。これらの入力方法については、Openweathermap API を参照してください。

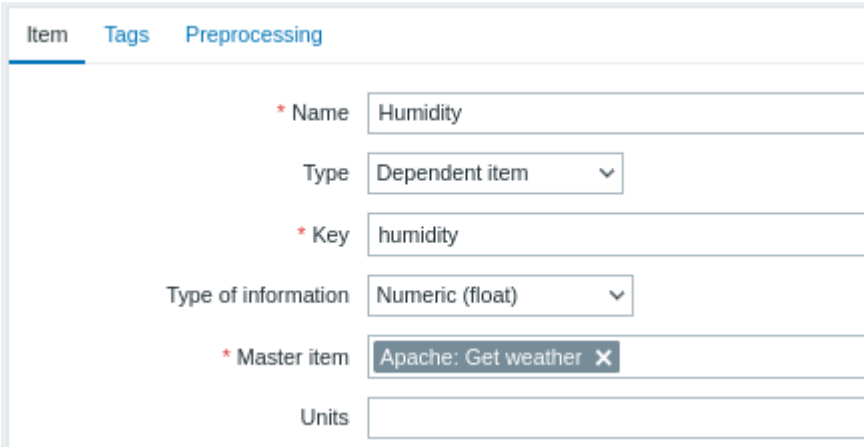
HTTP エージェントへの応答として返される JSON の例:

```
{
    "body": {
        "coord": {
            "lon": 40.01,
            "lat": 56.11
        },
        "weather": [{
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02n"
        }],
        "base": "stations",
        "main": {
            "temp": 15.14,
            "pressure": 1012.6,
```

```
            "humidity": 66,
            "temp_min": 15.14,
            "temp_max": 15.14,
            "sea_level": 1030.91,
            "grnd_level": 1012.6
        },
        "wind": {
            "speed": 1.86,
            "deg": 246.001
        },
        "clouds": {
            "all": 20
        },
        "dt": 1526509427,
        "sys": {
            "message": 0.0035,
            "country": "RU",
            "sunrise": 1526432608,
            "sunset": 1526491828
        },
        "id": 487837,
        "name": "Stavrovo",
        "cod": 200
    }
}
```

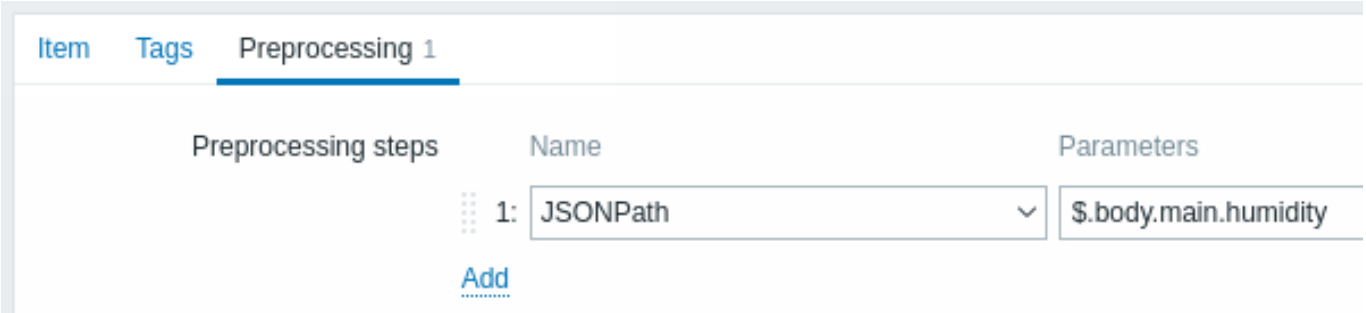

次の作業は、この JSON からデータを抽出する依存アイテムを設定することです。

- 湿度用の依存アイテムの設定例:

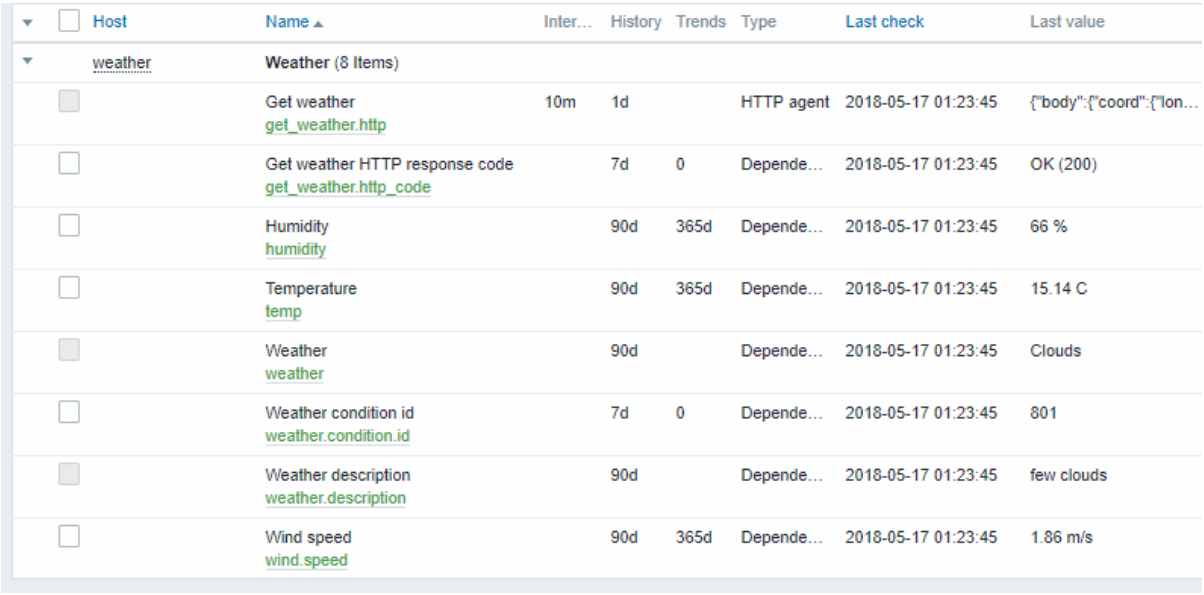


「Temperature」などの他の気象指標も、同様の方法で追加します。

- JSONPath を使用した依存アイテム値の前処理例:



- 最新データで気象データの結果を確認します。

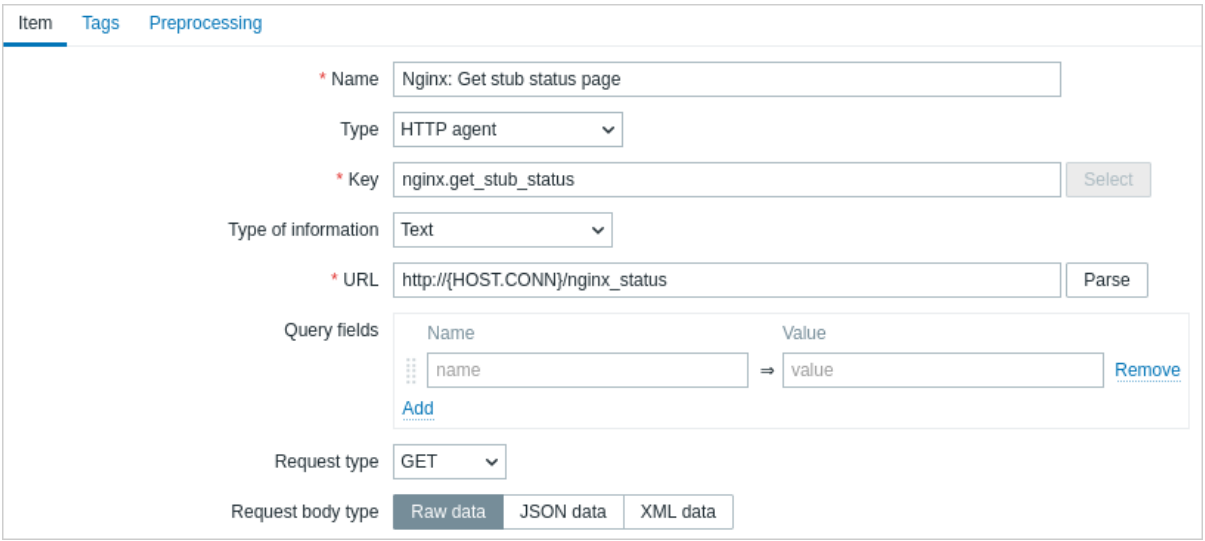


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity humidity		90d	365d	Depende...	2018-05-17 01:23:45	66 %
	Temperature temp		90d	365d	Depende...	2018-05-17 01:23:45	15.14 C
	Weather weather		90d		Depende...	2018-05-17 01:23:45	Clouds
	Weather condition id weather.condition.id		7d	0	Depende...	2018-05-17 01:23:45	801
	Weather description weather.description		90d		Depende...	2018-05-17 01:23:45	few clouds
	Wind speed wind.speed		90d	365d	Depende...	2018-05-17 01:23:45	1.86 m/s

例 5

Nginx のステータスページに接続し、そのメトリクスを一括で取得します。

- 公式ガイドに従って Nginx を設定します。
- 一括データ収集用のマスターアイテムを設定します。

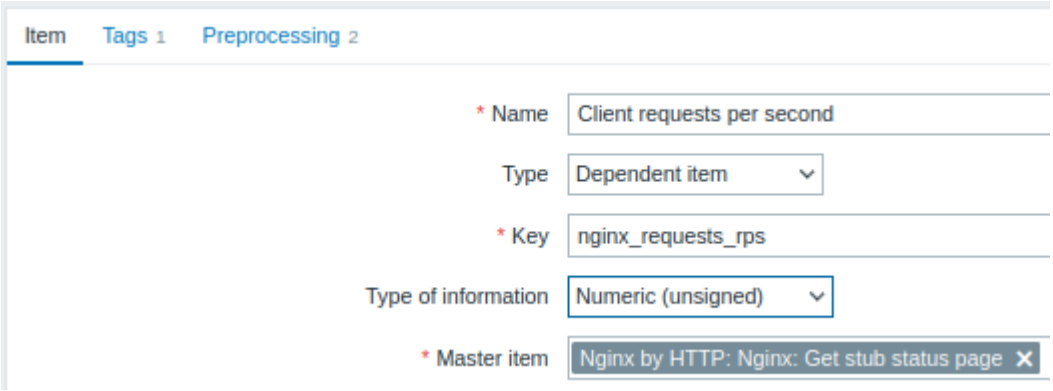


Nginx stub status の出力例:

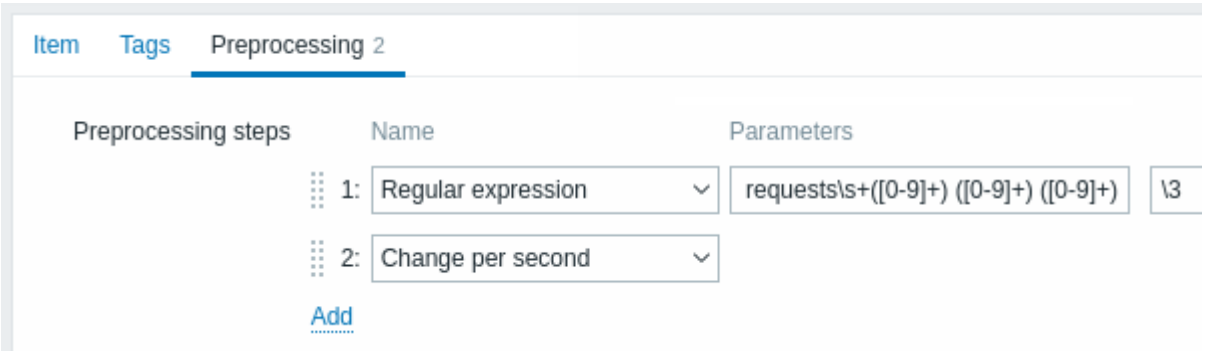
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

次のタスクは、データを抽出する依存アイテムを設定することです。

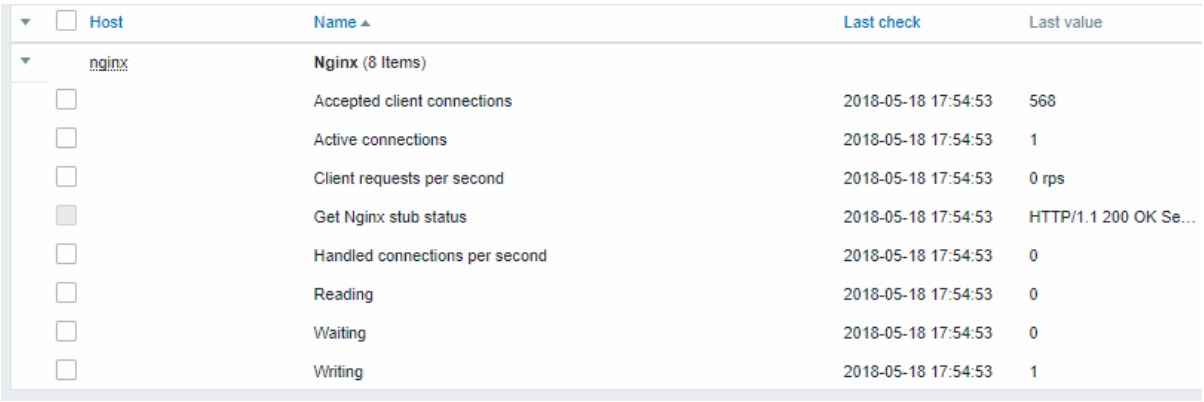
- 1 秒あたりのリクエスト数の依存アイテムの例を設定します。



- 正規表現 `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)` を使用した依存アイテムの値の前処理の例:



- 最新データで stub モジュールの完全な結果を確認します。



17 Prometheus チェック

概要

Zabbix は、Prometheus の行形式で公開されたメトリクスを問い合わせることができます。

Prometheus データの収集を開始するには、次の 2 つの手順が必要です。

- 適切なデータエンドポイントを指すHTTP マスターアイテム。例: `https://<prometheus host>/metrics`
- Prometheus の前処理オプションを使用する依存アイテム。マスターアイテムによって収集されたメトリクスから必要なデータを問い合わせます

Prometheus データの前処理オプションは 2 種類あります。

- *Prometheus* パターン - 通常のアイテムで Prometheus データを問い合わせるために使用します
- *Prometheus* から *JSON* - 通常のアイテムおよびローレベルディスカバリで使用します。この場合、問い合わせた Prometheus データは JSON 形式で返されます。

バルクプロセッシング

依存 item はバルクプロセッシングに対応しています。
 キャッシュとインデックスを有効にするには、*Prometheus pattern* のプリプロセスを 最初のプリプロセスステップにする必要があります。
 Prometheus pattern が最初のプリプロセスステップである場合、パースされた Prometheus データは、
 Prometheus pattern プリプロセスステップの最初の `<label>==<value>` 条

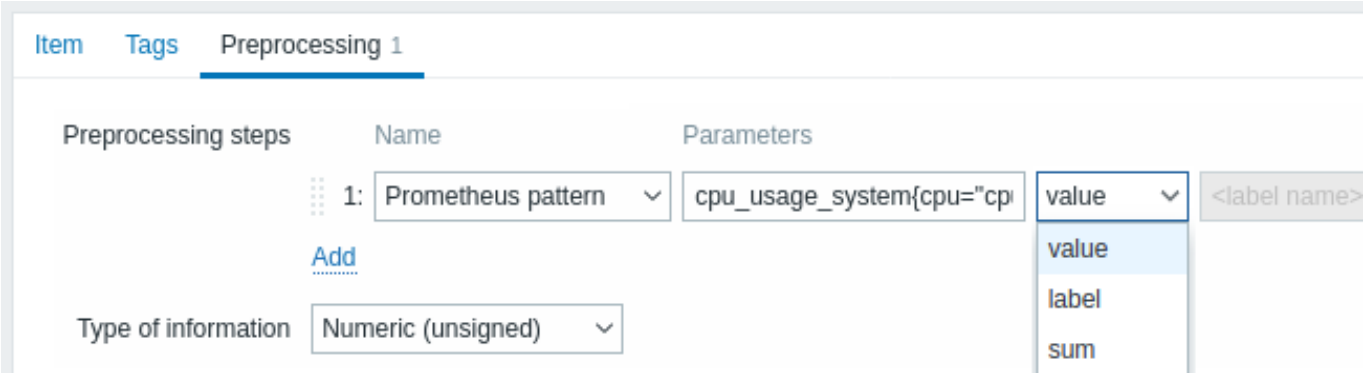
件によってキャッシュされ、インデックスが作成されます。
 このキャッシュは、このバッチ内の他の依存 item を処理する際に再利用されます。
 最適なパフォーマンスを得るために、最初のラベルは最も異なる値を持つものにすべきです。

最初のステップの前に他のプリプロセスがある場合は、マスター item か、従属 item のマスター item として使用される
 新しい従属 item のどちらかに移動させる必要があります。

設定

HTTP マスターアイテムが設定されている場合は、Prometheus の前処理ステップを使用する依存アイテムを作成する必要があります。

- 設定フォームで依存アイテムの一般的なパラメータを入力します
- 前処理タブを開きます
- Prometheus 前処理オプション（*Prometheus pattern* または *Prometheus to JSON*）を選択します



以下のパラメータは、*Prometheus pattern* 前処理オプション固有のものです。

パラメータ	説明	例
Pattern	必要なデータパターンを定義するには、Prometheus クエリ言語に似たクエリ言語を使用できます（比較表を参照）。例: <metric name> - メトリクス名で選択 {__name__=”<metric name>”} - メトリクス名で選択 {__name__=~”<regex>”} - 正規表現に一致するメトリクス名で選択 {<label name>=”<label value>”,...} - ラベル名で選択 {<label name>=~”<regex>”,...} - 正規表現に一致するラベル名で選択 {__name__=~”.*”}==<value> - メトリクス値で選択 または、上記の組み合わせ: <metric name>{<label1 name>=”<label1 value>”,<label2 name>=~”<regex>”,...}==<value> ラベル値には任意の UTF-8 文字列を使用できますが、バックスラッシュ、二重引用符、改行文字はそれぞれ \\、\"、\n としてエスケープする必要があります。その他の文字はエスケープしてはいけません。	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu=”cpu-total”}* *cpu_usage_system{cpu=~”.*”}* *cpu_usage_system{cpu=”cpu-total”,host=~”.*”}* *wmi_service_state{name=”dhcp”}==1* *wmi_os_timezone{timezone=~”.*”}==1*

パラメータ	説明	例
Result processing	値を返すか、ラベルを返すか、または適切な関数を適用するかを指定します（パターンが複数行に一致し、結果を集計する必要がある場合）： **value** - メトリクス値を返します（複数行に一致した場合はエラー） **label** - *Label* フィールドで指定したラベルの値を返します（複数のメトリクスに一致した場合はエラー） **sum** - 値の合計を返します **min** - 最小値を返します **max** - 最大値を返します **avg** - 平均値を返します **count** - 値の件数を返します このフィールドは *Prometheus pattern* オプションでのみ使用できます。	以下のパラメータ使用例も参照してください。
Output	ラベル名を定義します（任意）。この場合、そのラベル名に対応する値が返されます。 このフィールドは、*Result processing* フィールドで「Label」が選択されている場合に限り、*Prometheus pattern* オプションでのみ使用できます。	

パラメータの使用例

1. 最も一般的な使用例は、**value** を返すことです。次のデータから /var/db の値を返すには:

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096`

次のパラメータを使用します。

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - 「value」を選択

2. すべての `node_disk_usage_bytes` パラメータの平均値が必要な場合もあります。

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - 「avg」を選択

3. Prometheus は数値データのみをサポートしていますが、関連するテキスト説明も返せるようにする回避策が広く使われています。これは、フィルターを使用してラベルを指定することで実現できます。したがって、次のデータから'color' ラベルの値を返すには:

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

次のパラメータを使用します。

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - 「label」を選択
- *Label* - 「color」を指定

フィルター（数値'1' に基づく）は適切な行に一致し、ラベルはヘルスステータスの説明を返します（現在は'green' ですが、'red' または'yellow' になる可能性もあります）。

Prometheus から JSON へ

Prometheus からのデータローレベルディスカバリーに使用することができます。この場合、JSON 形式のデータが必要で、
 Prometheus to JSON プリプロセッシングオプションはまさにそのデータを返します。

詳しくは、Discovery using Prometheus dataを参照してください。

クエリ言語の比較

以下の表は、PromQL と Zabbix の Prometheus 前処理クエリ言語の相違点と類似点を示しています。

PromQL instant vector selector	Zabbix Prometheus 前処理
相違点	

PromQL instant vector selector		Zabbix Prometheus 前処理
クエリ対象	Prometheus サーバー	Prometheus exposition format のプレーンテキスト
返される値	Instant vector	メトリクスまたはラベル値（Prometheus パターン） JSON 内の単一値に対するメトリクスの配列（Prometheus to JSON）
ラベル一致演算子	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
ラベルまたはメトリクス名の一致に使用される正規表現	RE2	PCRE
比較演算子	一覧を参照	値のフィルタリングでは **==**（等しい）のみサポートされます
類似点		
文字列と等しいメトリクス名による選択	<metric name> または {__name__=”<metric name>”}	<metric name> または {__name__=”<metric name>”}
正規表現に一致するメトリクス名による選択	{__name__=~”<regex>”}	{__name__=~”<regex>”}
文字列と等しい <label name> 値による選択	{<label name>=”<label value>”,...}	{<label name>=”<label value>”,...}
正規表現に一致する <label name> 値による選択	{<label name>=~”<regex>”,...}	{<label name>=~”<regex>”,...}
文字列と等しい値による選択	{__name__=~”.*”} == <value>	{__name__=~”.*”} == <value>

18 スクリプト

概要

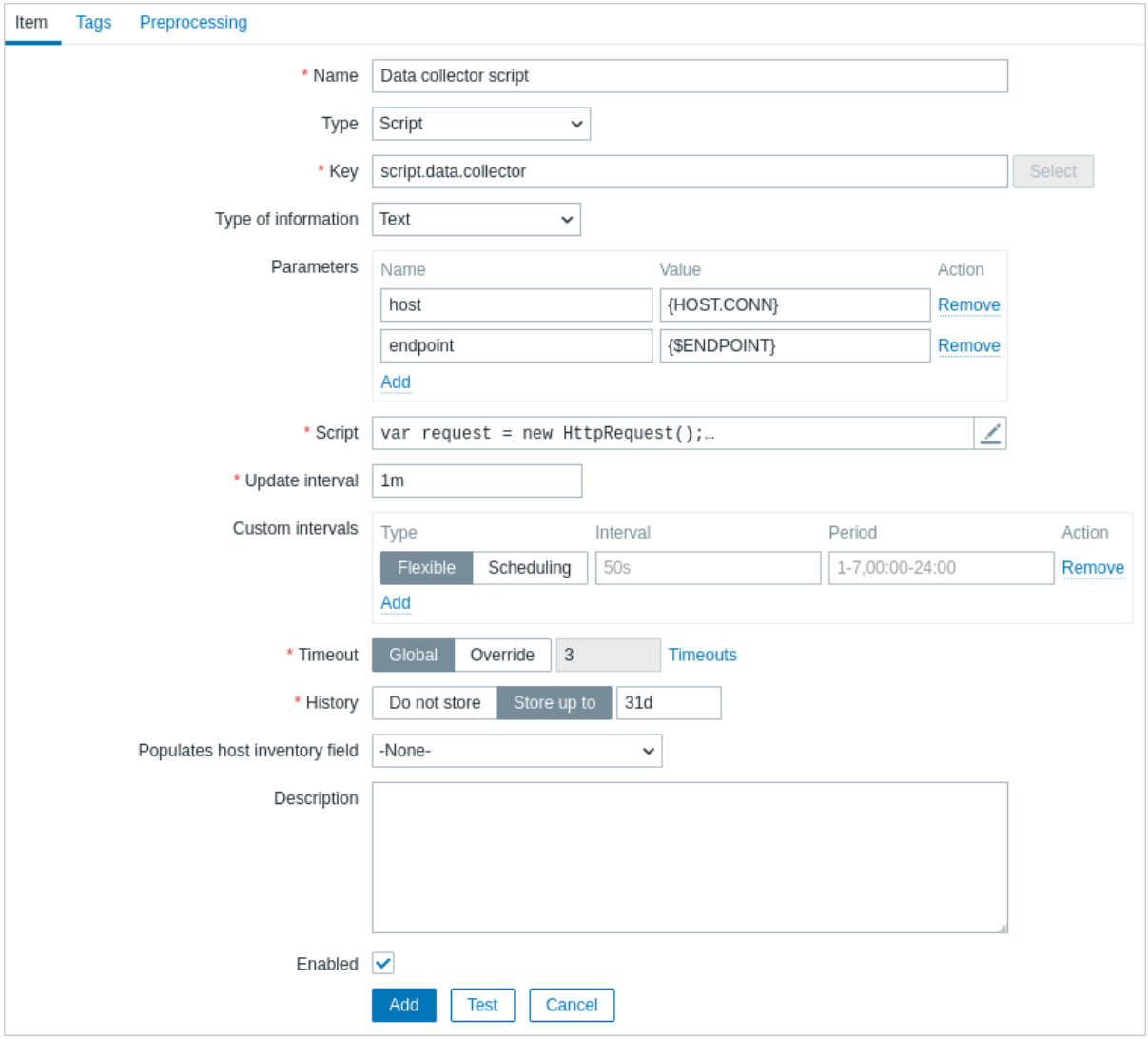
スクリプトアイテムは、HTTP/HTTPS 経由でデータを取得する機能を備えた、ユーザー定義の JavaScript コードを実行してデータを収集するために使用できます。スクリプトに加えて、オプションのパラメーター一覧（名前と値の組）およびタイムアウトを指定できます。

このアイテムタイプは、複数のステップや複雑なロジックを必要とするデータ収集シナリオで役立つ場合があります。例えば、スクリプトアイテムを設定して HTTP 呼び出しを行い、その最初のステップで受信したデータを何らかの方法で処理し、変換した値を 2 回目の HTTP 呼び出しに渡すことができます。

スクリプトアイテムは、Zabbix サーバーまたはプロキシのポーラーによって処理されます。

設定

アイテム設定フォームのタイプフィールドで「スクリプト」を選択し、必要なフィールドに入力します。



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

スクリプトアイテムで固有の情報が必要なフィールドは次のとおりです。

Field	Description
Key	アイテムの識別に使用する一意のキーを入力します。
Parameters	スクリプトに渡す変数を、属性と値のペアとして指定します。 ユーザーマクロを使用できます。どの組み込みマクロがサポートされているかを確認するには、サポートされるマクロの表で「Script-type item」を検索してください。
Script	パラメーターフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターで、JavaScript コードを入力します。このコードには、メトリクス値を返すためのロジックを実装する必要があります。 このコードでは、すべてのパラメーターと、Zabbix によって追加されたすべての追加の JavaScript オブジェクトにアクセスできます。 参照: JavaScript Guide。
Timeout	JavaScript の実行タイムアウトです（1～600 秒。超過するとエラーが返されます）。 スクリプトによっては、タイムアウトが発生するまでにさらに時間がかかる場合があることに注意してください。 *Timeout* パラメーターの詳細については、一般的なアイテム属性を参照してください。

例

シンプルなデータ収集

https://www.example.com/release_notes の内容を収集するには、次のようにします。

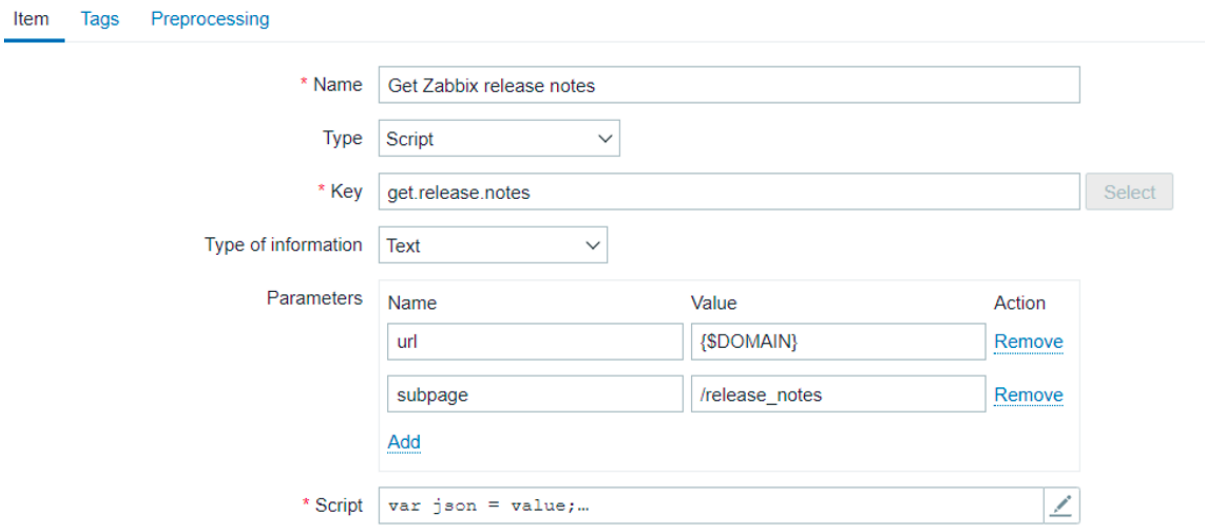
- タイプ「Script」のアイテムを作成します。
- *Script* フィールドに次を入力します。

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

パラメータを使用したデータ収集

特定のページの内容を収集し、パラメータを活用します。

- タイプ「Script」のアイテムを作成し、2 つのパラメータを設定します。
  - **url : {$DOMAIN}**（ユーザーマクロ {$DOMAIN} を定義する必要があります。定義場所はホストレベルが推奨されます）
  - **subpage : /release_notes**



- *Script* フィールドに次を入力します。

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

複数の HTTP リクエスト

https://www.example.com と *https://www.example.com/release_notes* の両方の内容を収集します。

- タイプ「スクリプト」のアイテムを作成します。
- スクリプトフィールドに、次を入力します。

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

ロギング

Zabbix server ログに”Log test” エントリを追加し、アイテム値”1” を返信します。

- タイプ”Script” の item を作成します。
- *Script* フィールドに、以下のコードを入力します。

```
Zabbix.log(3, 'Log test');
return 1;
```

19 ブラウザアイテム

概要

ブラウザアイテムでは、ブラウザを使用して複雑な Web サイトや Web アプリケーションを監視できます。

> **Attention:**
> ブラウザアイテムのサポートは現在実験的です。

ブラウザアイテムは、ユーザー定義の JavaScript コードを実行し、HTTP/HTTPS 経由でデータを取得してデータを収集します。このアイテムでは、クリック、テキスト入力、Web ページ間の移動、その他の Web サイトまたは Web アプリケーションに対するユーザー操作など、ブラウザ関連のアクションをシミュレートできます。

スクリプトに加えて、オプションのパラメーター一覧（名前と値のペア）およびタイムアウトを指定できます。

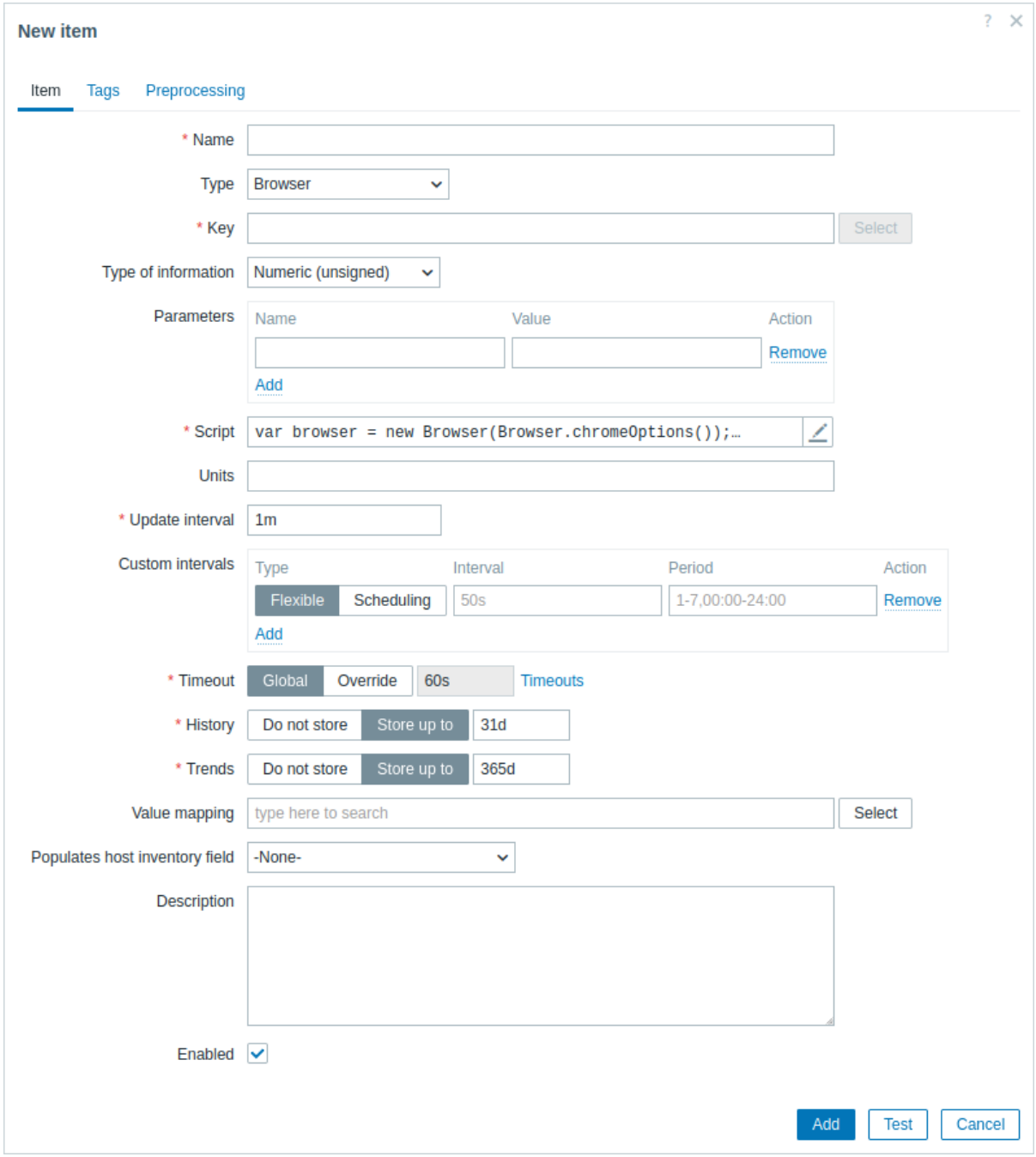
> **Attention:**
> このアイテムは、Selenium Server またはプレーンな WebDriver（たとえば ChromeDriver）を Web テストエンドポイントとして使用し、W3C WebDriver standard を部分的に実装しています。このアイテムを動作させるには、Zabbix server/proxy 設定ファイルの `WebDriverURL` パラメータでエンドポイントを設定してください（ChromeDriver を使用する場合は、Security Considerations を参照してください）。パフォーマンス向上のため、Web テスト環境専用のサーバーの使用を検討してください。

ブラウザアイテムのチェックは、Zabbix サーバーまたはプロキシの browser poller によって実行および処理されます。必要に応じて、Zabbix server/proxy 設定ファイルの `StartBrowserPollers` パラメータで、事前に fork された browser poller インスタンス数を調整できます。

複雑な Web サイトおよび Web アプリケーションの監視には、Website by Browser テンプレートを標準搭載テンプレートとして利用できます。

設定

アイテム設定フォームのタイプフィールドで、Browser を選択し、必須フィールドに入力します。



必須の入力フィールドには、赤いアスタリスクが付いています。

Browser アイテムで固有の情報が必要なフィールドは次のとおりです。

Field	Description
Key	アイテムの識別に使用される一意のキーを入力します。
Parameters	スクリプトに渡す変数を、属性と値のペアとして指定します。 ユーザーマクロをサポートしています。サポートされている組み込みマクロを確認するには、サポートされるマクロの表で「Browser-type item」を検索してください。
Script	パラメーターフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターで、JavaScript コードを入力します。このコードには、メトリクス値を返すためのロジックを実装する必要があります。 このコードからは、すべてのパラメーター、すべての追加の JavaScript オブジェクト、および Zabbix によって追加されたBrowser アイテムの JavaScript オブジェクトにアクセスできます。 参照: JavaScript Guide。
Timeout	JavaScript の実行タイムアウトです（1～600 秒。これを超えるとエラーが返されます）。 スクリプトによっては、タイムアウトが発生するまでにさらに時間がかかる場合があることに注意してください。 *Timeout* パラメーターの詳細については、一般的なアイテム属性を参照してください。

例

Website by Browser テンプレートを使用して Web サイトを監視するように Zabbix を設定する方法の例については、Browser アイテムで Web サイトを監視するを参照してください。

デフォルトスクリプト

次のスクリプトは以下を実行します。

1. ブラウザーセッションを初期化します。
2. 指定した URL に移動します。
3. パフォーマンスエントリとセッション統計を収集し、それらを JSON 文字列として返します。

Script フィールドに、次を入力します。

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

カスタム capabilities でブラウザを初期化する

次のスクリプトは以下を行います。

1. スクリプト内で指定された順序で最初に一致した利用可能なブラウザに基づいて、ブラウザセッションを初期化します。
2. ページ読み込み戦略や、Chrome、Firefox、Microsoft Edge ブラウザのヘッドレスモードなど、各ブラウザ固有のオプションを含むブラウザの capabilities を定義します。

Script フィールドに、次を入力します。

```
var browser = new Browser({
    "capabilities":{
        "firstMatch":[
            {
                "browserName":"chrome",
                "pageLoadStrategy":"normal",
                "goog:chromeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"firefox",
                "pageLoadStrategy":"normal",
                "moz:firefoxOptions":{
                    "args":[
                        "--headless"
                    ]
                }
            },
            {
                "browserName":"MicrosoftEdge",
                "pageLoadStrategy":"normal",
                "ms:edgeOptions":{
                    "args":[
                        "--headless=new"
                    ]
                }
            },
            {
                "browserName":"safari",
                "pageLoadStrategy":"normal"
```

```
            }
        ]
    }
});
```

GUI を有効にしてブラウザを初期化する

デフォルトでは、ブラウザセッション（Safari を除く）はヘッドレスモードで初期化されます。これは、ブラウザのグラフィカルユーザーインターフェース（GUI）が表示されないことを意味します。

次のスクリプトは、GUI を有効にしてブラウザセッションを初期化します。

WebDriver がブラウザのバイナリを見つけられない場合は、パスを手動で指定できることに注意してください。

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// Firefox セッションを GUI 付きで初期化するには、次の行のコメントを解除します:
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// Microsoft Edge セッションを GUI 付きで初期化するには、次の行のコメントを解除します:
// var opts = Browser.edgeOptions();
// opts.capabilities.alwaysMatch['ms:edgeOptions'].binary = 'usr/bin/microsoft-edge';
// opts.capabilities.alwaysMatch['ms:edgeOptions'].args = [];

var browser = new Browser(opts);
```

Note:
テストをリモートサーバー上またはコンテナ内で実行している場合は、Virtual Network Computing（VNC）クライアントを使用して、そのマシンの VNC サーバーに接続できます。これにより、ブラウザの GUI をリモートで表示して操作できます。

スクリーンショットの取得

次のスクリプトは以下を実行します。

1. ブラウザーセッションを初期化します。
2. スクリーンショットのサイズを決定するために、ブラウザーのビューポートサイズを設定します（パラメータで指定。以下を参照）。
3. URL に移動します（パラメータで指定。以下を参照）。
4. セッション統計を収集し、スクリーンショットを取得して、収集した統計に追加します。
5. エラーメッセージとスクリーンショットを取得して、エラーを処理します。
6. 収集した結果を JSON 文字列として返します。

このスクリプトでは、アイテム設定フォームのパラメータも使用します。

- webURL - http://example.com
- width - 1920
- height - 1080

Script フィールドに、次を入力します。

```
var browser, result;

var browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from Zabbix.

    browser.setScreenSize(Number(params.width), Number(params.height))

    browser.navigate(params.webURL);

    result = browser.getResult();
    result.screenshot = browser.getScreenshot();
}
```

```
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

Zabbix ログインの確認

次のスクリプトは以下を実行します。

1. ブラウザセッションを初期化します。
2. ページ（パラメータとして指定。詳細は以下を参照）に移動します。
3. ユーザー名とパスワード（パラメータとして指定。詳細は以下を参照）を入力します。
4. ログインボタンを見つけてクリックします。
5. ログアウトボタンを見つけてクリックします。
6. ログイン前後およびログアウト後のパフォーマンスデータを収集します。
7. エラーメッセージとスクリーンショットを取得してエラーを処理します。
8. 収集した結果を JSON 文字列として返します。

このスクリプトは、アイテム設定フォームのパラメータも使用します。

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

Script フィールドに次を入力します。

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from Zabbix.

    browser.navigate(params.webURL);
    browser.collectPerfEntries("open page");

    var el = browser.findElement("xpath", "//input[@id='name']");
    if (el === null) {
        throw Error("cannot find name input field");
    }
    el.sendKeys(params.username);

    el = browser.findElement("xpath", "//input[@id='password']");
    if (el === null) {
        throw Error("cannot find password input field");
    }
    el.sendKeys(params.password);

    el = browser.findElement("xpath", "//button[@id='enter']");
    if (el === null) {
        throw Error("cannot find login button");
    }
    el.click();

    browser.collectPerfEntries("login");

    el = browser.findElement("link text", "Sign out");
    if (el === null) {
        throw Error("cannot find logout button");
    }
```

```
    el.click();

    browser.collectPerfEntries("logout");

    result = browser.getResult();
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

リンクの検索

次のスクリプトは以下を実行します。

1. ブラウザセッションを初期化します。
2. 配列から重複要素を削除する関数を定義します（手順 5 を参照）。
3. ページに移動します（パラメータで指定。以下を参照）。
4. ページ上のリンクを検索します。
5. リンクが一意になるように重複したリンクを削除します。
6. "http" で始まるリンクのみを抽出します。
7. 抽出したリンクを特定の構造に整形します。
8. エラーメッセージとスクリーンショットを取得してエラーを処理します。
9. 収集した結果を JSON 文字列として返します。

このスクリプトは、アイテム設定フォームのパラメータも使用します。

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

Script フィールドに次を入力します。

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value);  // Zabbix から渡されたパラメータを含む JSON 文字列を解析します。

    function uniq(a) {
        return a.sort().filter(function (item, pos, ary) {
            return !pos || item != ary[pos - 1];
        });
    }

    browser.navigate(params.scheme + '://' + params.domain + params.path);

    var el = browser.findElements("link text", "");
    var links = [];
    for (var n = 0; n < el.length; n++) {
        links.push(el[n].getAttribute('href'));
    }

    links = uniq(links);

    result = [];
    for (i = 0; i < links.length; i++) {
        if (links[i].match(/^http.*/)) {
            var row = {};
```

```
            row["{#URL}"] = links[i];
            result.push(row);
        }
    }
}
catch (err) {
    if (!(err instanceof BrowserError)) {
        browser.setError(err.message);
    }
    result = browser.getResult();
    result.error.screenshot = browser.getScreenshot();
}
finally {
    return JSON.stringify(result);
}
```

4 ヒストリとトレンド

概要

履歴とトレンドは、Zabbix で収集したデータを保存する 2 つの方法です。

履歴は収集した各値をそのまま保持するのに対し、トレンドは 1 時間単位の平均情報を保持するため、必要なリソースが少なくなります。

履歴の保持

履歴を何日間保持するかは、以下で設定できます。

- アイテムのプロパティフォーム
- アイテムの一括更新時
- ハウスキーパータスクを設定する際

これより古いデータは、ハウスキーパーによって削除されます。

一般的には、履歴の保持日数は可能な限り最小限にし、そうすることで大量の履歴値によってデータベースに過度な負荷をかけないようにすることを強く推奨します。

長期間の履歴を保持する代わりに、トレンドデータをより長く保持できます。
例えば、履歴は 14 日間、トレンドは 5 年間保持できます。

履歴データとトレンドデータで必要な容量がどの程度異なるかは、データベースサイズのページを参照すると把握できます。

履歴の保持期間を短くしても、グラフでは古いデータを引き続き確認できます。これは、古いデータの表示にグラフがトレンド値を使用するためです。

Attention:
履歴が「0」に設定されている場合、アイテムは依存アイテムとインベントリのみを更新します。
トリガーの評価は履歴データのみに基づいて行われるため、トリガー関数は評価されません。

Note:
履歴を保持する別の方法として、ロード可能モジュールの履歴エクスポート機能の使用を検討してください。

トレンドの保持

トレンドは、数値データ型について 1 時間ごとの最小値、最大値、平均値、および値の総数を保存する、組み込みの履歴データ削減メカニズムです。

トレンドを何日間保持するかは、次の場所で設定できます。

- アイテムのプロパティフォーム
- アイテムの一括更新時
- Housekeeper タスクの設定時

通常、トレンドは履歴よりもはるかに長く保持できます。
それより古いデータは housekeeper によって削除されます。

Zabbix サーバーは、データが流入するにつれて、実行時にトレンドキャッシュへトレンドデータを蓄積します。
サーバーは、次の状況で各アイテムの前の時間のトレンドをデータベースに書き出します（Web インターフェースはそこから参照します）。

- サーバーがそのアイテムの現在の時間帯における最初の値を受信したとき
- 現在の時間帯の残り時間が 5 分以下で、なおそのアイテムの現在の時間帯の値がまだ存在しないとき
- サーバーが停止するとき

グラフ上でトレンドを確認するには、少なくとも次の時間帯の開始まで（アイテムが頻繁に更新される場合）、遅い場合は次の時間帯の終了まで（アイテムがまれにしか更新されない場合）待つ必要があります。つまり、最大で 2 時間です。

サーバーがトレンドキャッシュを書き出す際、その時間帯のトレンドがすでにデータベース内に存在する場合（たとえば、サーバーが時間帯の途中で再起動された場合）、サーバーは単純な insert ではなく update 文を使用する必要があります。
そのため、大規模な環境で再起動が必要な場合は、トレンドデータの重複を避けるために、ある時間帯の終わりにサーバーを停止し、次の時間帯の始まりに起動することが望まれます。

履歴テーブルは、トレンド生成には一切関与しません。

Attention:
トレンドが'0' に設定されている場合、Zabbix サーバーはトレンドをまったく計算せず、保存もしません。

Note:
トレンドは、元の値と同じデータ型で計算および保存されます。
その結果、unsigned データ型の値の平均値計算は丸められ、値の間隔が小さいほど結果の精度は低くなります。
たとえば、アイテムの値が 0 と 1 の場合、平均値は 0.5 ではなく 0 になります。
また、サーバーの再起動により、現在の時間帯における unsigned データ型の平均値計算の精度が失われる可能性があります。

5 ユーザーパラメーター

概要

Zabbix にあらかじめ定義されていないエージェントチェックを実行したい場合があります。そのような場合に役立つのがユーザーパラメータです。

必要なデータを取得するコマンドを作成し、それをエージェント設定ファイル の「UserParameter」設定パラメータとしてユーザーパラメータに含めることができます。

ユーザーパラメータの構文は次のとおりです。

```
UserParameter=<key>,<command>
```

ご覧のとおり、ユーザーパラメータにはキーも含まれます。このキーは、アイテムを設定する際に必要になります。参照しやすい任意のキーを指定してください（ホスト内で一意である必要があります）。

新しいパラメータを反映するには、エージェントを再起動するか、エージェントのランタイム制御オプションを使用します。例:

```
zabbix_agentd -R userparameter_reload
```

その後、アイテムの設定時に、実行したいユーザーパラメータのコマンドを参照するためのキーを入力します。

ユーザーパラメータは、Zabbix エージェントによって実行されるコマンドです。アイテム値の前処理ステップの前に、最大 16MB のデータを返すことができます。

UNIX オペレーティングシステムでは、コマンドラインインタープリターとして **/bin/sh** が使用されます。ユーザーパラメータはエージェントチェックのタイムアウトに従います。タイムアウトに達すると、fork されたユーザーパラメータのプロセスは終了されます。

関連情報:

- ユーザーパラメータの活用方法に関するステップバイステップのチュートリアル
- コマンド実行

単純なユーザーパラメータの例

単純なコマンド:

```
UserParameter=ping,echo 1
```

agent は、'ping' キーを持つ item に対して常に'1' を返します。

より複雑な例:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

agent は、MySQL サーバが生きている場合は'1' を、そうでない場合は'0' を返します。

柔軟なユーザーパラメータ

柔軟なユーザーパラメータでは、キーにパラメータを受け付けることができます。これにより、1 つの柔軟なユーザーパラメータを基に複数のアイテムを作成できます。

柔軟なユーザーパラメータの構文は次のとおりです。

```
UserParameter=key[*],command
```

Parameter	Description
Key	一意のアイテムキーです。[*] は、このキーが角括弧内のパラメータを受け付けることを示します。パラメータはアイテムの設定時に指定します。
Command	キーの値を評価するために実行されるコマンドです。 柔軟なユーザーパラメータのみ： コマンド内では、アイテムキー内の対応するパラメータを参照するために、位置参照 $1...$9 を使用できます。 Zabbix は、アイテムキーの [ ] 内に含まれるパラメータを解析し、それに応じてコマンド内の $1,...,$9 を置き換えます。 $0 は、実行される元のコマンド（$0,...,$9 の展開前）に置き換えられます。 位置参照は、ダブルクォート (") またはシングルクォート (') で囲まれているかどうかに関係なく解釈されます。 位置参照を変更せずに使用するには、ドル記号を 2 つ指定します。例えば、awk '{print $$2}' のようにします。この場合、コマンド実行時に $$2 は実際には $2 になります。

Attention:
$ 記号を含む位置参照の検索と置換は、柔軟なユーザーパラメータに対してのみ Zabbix エージェントによって行われます。単純なユーザーパラメータでは、このような参照処理は行われないため、$ 記号のクォートは不要です。

Attention:
デフォルトでは、ユーザーパラメータでは使用できない文字があります。文字の完全な一覧については、UnsafeUserParameters を参照してください。

例 1

とてもシンプルなもの:

```
UserParameter=ping[*],echo $1
```

ping[something] というフォーマットを持つ監視項目は無制限に定義することができる。

- ping[0] - 常に'0' を戻す
- ping[aaa] - 常に'aaa' を戻す

例 2

もっとセンスを磨こう!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

このパラメータは、MySQL データベースの可用性を監視するために使用することができます。
 ユーザー名とパスワードを渡すことができます:

```
mysql.ping[zabbix,our_password]
```

例 3

正規表現にマッチする行はファイル中に何行あるか？

```
UserParameter=wc[*],grep -c "$2" $1
```

このパラメータは、ファイル内の行数を計算するために使用することができます。

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

コマンドの結果

コマンドの戻り値は、コマンドによって生成された標準出力と標準エラー出力です。

Attention:
テキスト（character、log、または text 型の情報）を返すアイテムは、標準エラー出力がある場合でも未サポートにはなりません。

戻り値は 16MB に制限されます（切り捨てられる末尾の空白を含む）。また、データベースの制限 も適用されます。

テキスト（character、log、または text 型の情報）を返すユーザーパラメータは、空白文字のみを返すこともできます。結果が無効な場合、アイテムは未サポートになります。

1 Zabbix エージェントの機能拡張

このチュートリアルでは、ユーザーパラメータを使用して Zabbix エージェントの機能を拡張する方法を段階的に説明します。

ステップ 1

必要なパラメータを取得するためのスクリプトまたはコマンドラインを記述します。

例えば、MySQL サーバで実行されたクエリの総数を取得するために、以下のようなコマンドを記述することができます。

```
mysqladmin -uroot status | cut -f4 -d":". | cut -f1 -d "S"
```

このコマンドを実行すると、SQL クエリの総数が返されます。

ステップ 2

zabbix_agentd.conf にコマンドを追加します。

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":". | cut -f1 -d "S"
```

mysql.questions は一意な識別子です。それは任意の有効なキーであることができます。例えば、*queries* などです。

このパラメータをテストするには、Zabbix agent を「-t」フラグで使用します (root 権限で実行する場合)。ただし、デーモンとして起動した場合、agent のパーミッションが異なる可能性があります。

```
zabbix_agentd -t mysql.questions
```

ステップ 3

以下のコマンドを実行して、設定ファイルからユーザパラメータを再ロードします。

```
zabbix_agentd -R userparameter_reload
```

実行時制御コマンドの代わりに、agent を再起動することもできます。

zabbix_get ユーティリティを使用して、パラメータをテストします。

ステップ 4

監視するホストに Key=mysql.questions の新しい item を追加します。
 item のタイプは、Zabbix agent または Zabbix agent (active) のいずれかにする必要があります。

Zabbix server で返される値のタイプが正しく設定されている必要があることに注意してください。
 そうでない場合、Zabbix は値を受け付けません。

6 Windows パフォーマンスカウンタ

概要

Windows のパフォーマンスカウンタは、*perf_counter[]* キーを使用して効率的に監視することができます。

例えば:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

あるいは

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

このキーまたは英語版のみの同等物 `perf_counter_en` の使用に関する詳細については、[Windows-specific item keys]((/manual/config/items/it
を参照してください。

監視可能なパフォーマンスカウンタの全リストを取得するには、次のコマンドを実行します。

```
typeperf -qx
```

また、ローレベルディスカバリ機能を使用して、複数のobject instancesを検出し、複数のインスタンスオブジェクトに対する <br> perf_counter item の作成を自動化することも可能です。

数値表現

Windows は、オブジェクト名およびパフォーマンスカウンター名の数値表現（インデックス）を保持しています。Zabbix は、これらの数値表現を perf_counter、perf_counter_en アイテムキーのパラメータ、および PerfCounter、PerfCounterEn 設定パラメータでサポートしています。

ただし、数値インデックスが特定のホスト上で正しい文字列に対応していることを保証できない限り、これらを使用することは推奨されません。さまざまなローカライズ版 Windows を使用する異なるホスト間で動作するポータブルなアイテムを作成する必要がある場合は、システムロケールに関係なく英語名を使用できる perf_counter_en キーまたは PerfCounterEn 設定パラメータを使用できます。

数値の対応を確認するには、**regedit** を実行し、*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009* 内の *Counter* を探します。

レジストリエントリには、次のような情報が含まれています。

```
1
1847
2
System
4
Memory
6
% Processor Time
10
File Read Operations/sec
12
File Write Operations/sec
14
File Control Operations/sec
16
File Read Bytes/sec
18
File Write Bytes/sec
....
```

ここでは、'\System\% Processor Time' のようなパフォーマンスカウンターの各文字列部分に対応する番号を確認できます。

```
System → 2
% Processor Time → 6
```

その後、これらの番号を使用してパスを数値で表現できます。

```
\2\6
```

パフォーマンスカウンタパラメーター

Windows のパフォーマンスカウンタを監視するために、いくつかの PerfCounter パラメータを導入することができます。

例えば、Zabbix エージェント設定ファイルに以下を追加することができます。

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
ま た は
PerfCounter=UserPerfCounter2,"\4\24",30
```

このようなパラメータを設定すると、それぞれの item を作成するためのキーとして *UserPerfCounter1* または *UserPerfCounter2* を
 使用するだけでよくなります。

設定ファイルを変更した後は、Zabbix エージェントを再起動することを忘れないでください。

7 一括アップデート

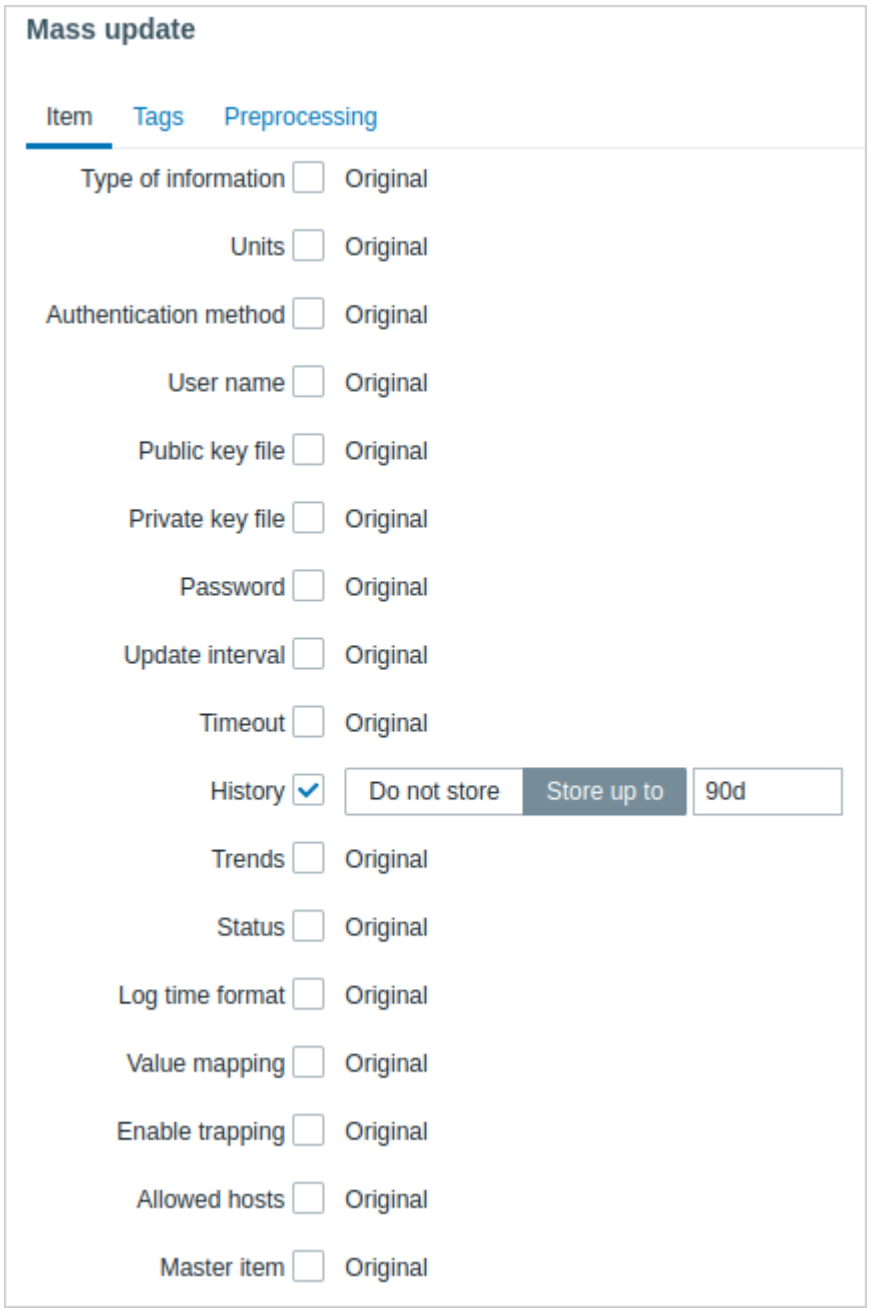
概要

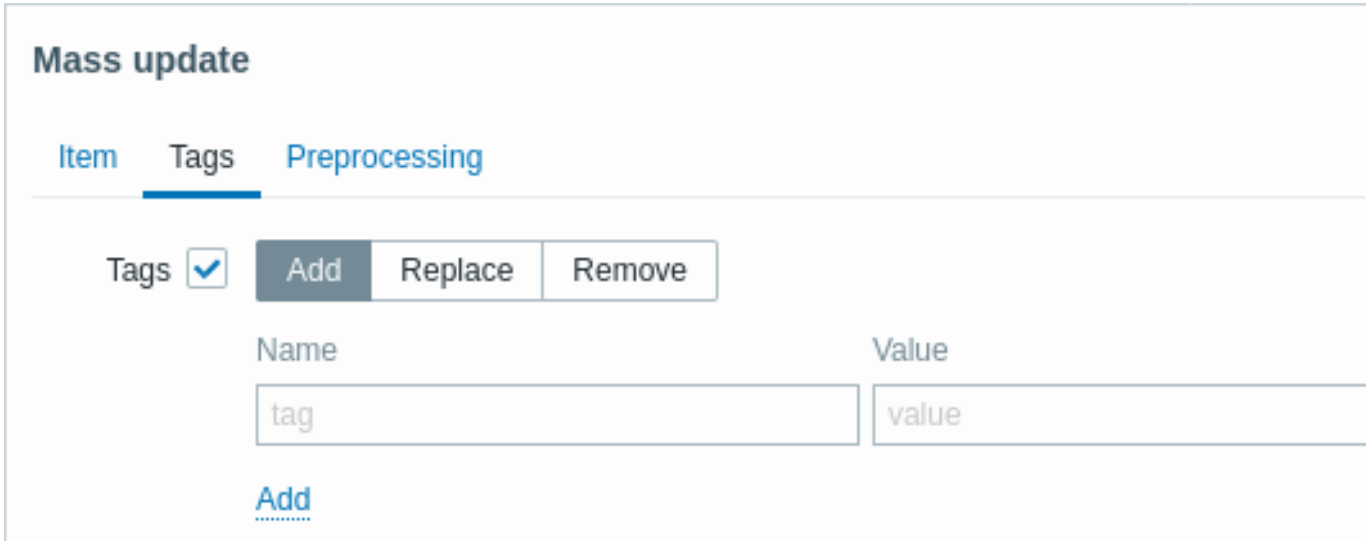
一度に複数のアイテムの属性を変更したい場合があります。個々のアイテムを開いて編集する代わりに一括更新機能を使います。

一括更新の使用

いくつかのアイテムを一括で更新するには、次のようにします。

- リスト内の更新するアイテムのチェックボックスにチェックを付ける
- リストの下にある一括更新をクリックする
- 必要な属性 (アイテム、タグ、保存前処理) のあるタブに移動する
- 更新する属性のチェックボックスにチェックを付ける
- 属性の新しい値を入力する

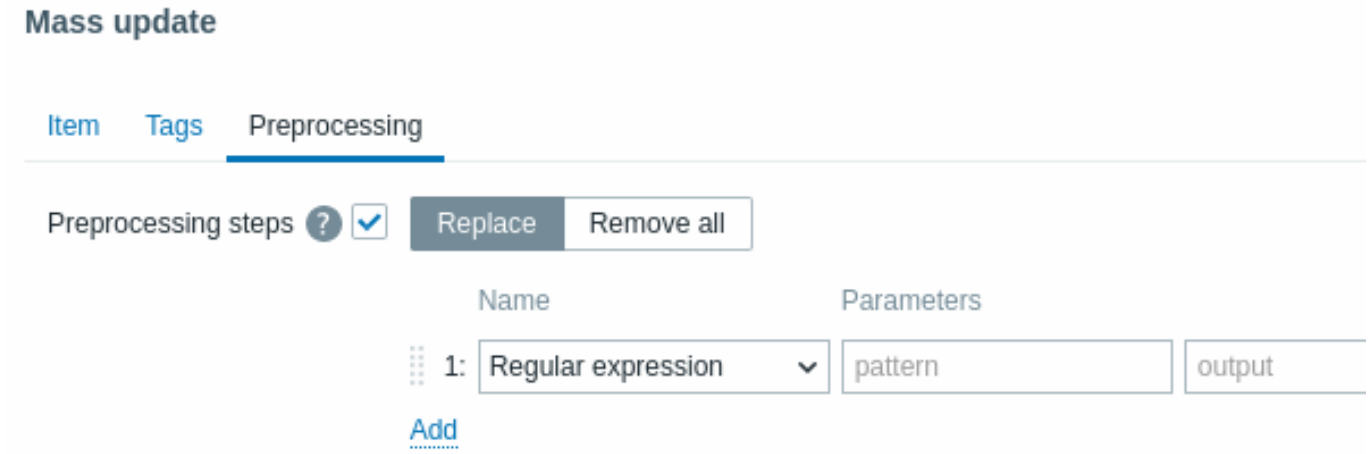




タグオプションでは次の操作が可能です。

- 追加 - 指定したタグをアイテムに追加します (同じ名前でも異なる値のタグは' 重複' とみなされず、同じホストに追加できます)。
- 置換 - 指定されたタグを削除し、新しい値のタグを追加します。
- 削除 - 指定されたタグを項目から削除します。

タグではユーザーマクロ、{INVENTORY. *} マクロ、{HOST.HOST}、{HOST.NAME}、{HOST.CONN}、{HOST.DNS}、{HOST.IP}、{HOST.PORT}、そして {HOST.ID} マクロがサポートされています。



保存前処理のステップの横にあるチェックボックスをチェックして、保存前処理のステップの一括更新を有効にします。

保存前処理の一括更新では次の操作が可能です。

- *Replace* - アイテムの既存の保存前処理のステップを、以下に指定する保存前処理のステップに置き換えます
- *Remove all* - アイテムからすべての保存前処理ステップを削除します

完了したら、更新をクリックします。

8 値のマッピング

概要

値のマッピングを使用すると、数値/文字列の値と文字列表現の対応付けにより、受信した値をより分かりやすく表示するよう設定できます。

例えば、アイテムの値が"0" または"1" の場合、値のマッピングを使用して、これらの値をより分かりやすく次のように表現できます。

- 0 → 利用不可
- 1 → 利用可能

データバックアップの種類に対する値のマッピングは、例えば次のように設定できます。

- F → フル

- D → 差分
- I → 増分

電圧の値の範囲は、例えば次のように設定できます。

- <=209 → 低
- 210-230 → 正常
- >=231 → 高

値のマッピングは、Zabbix Web インターフェースおよびメディアタイプによって送信される通知で使用されます。

Attention:
受信した値を設定済みの表現に置き換える処理は、Zabbix Web インターフェースとサーバーの両方で実行されます。ただし、サーバーが置き換えを処理するのは、次の場合のみです。

- ホストインベントリ への値の設定時
- サポートされているマクロ {ITEM.VALUE}、{ITEM.LASTVALUE}、{EVENT.OPDATA}、{EVENT.CAUSE.OPDATA} の展開時

値のマッピングは、テンプレートまたはホストに設定します。一度設定すると、それぞれのテンプレートまたはホスト内のすべてのアイテムで利用できます。アイテムの設定 時に、*Value mapping* パラメータで、あらかじめ設定した値のマッピング名を指定してください。

Note:
値マップの継承はありません。ホストおよびテンプレートは、リンクされたテンプレートから値のマッピングを継承しません。ホスト上のテンプレートアイテムは、引き続きテンプレート上で設定された値のマッピングを使用します。

Note:
値のマッピングは、情報の型が *Numeric (unsigned)*、*Numeric (float)*、*Character* のアイテムで使用できます。

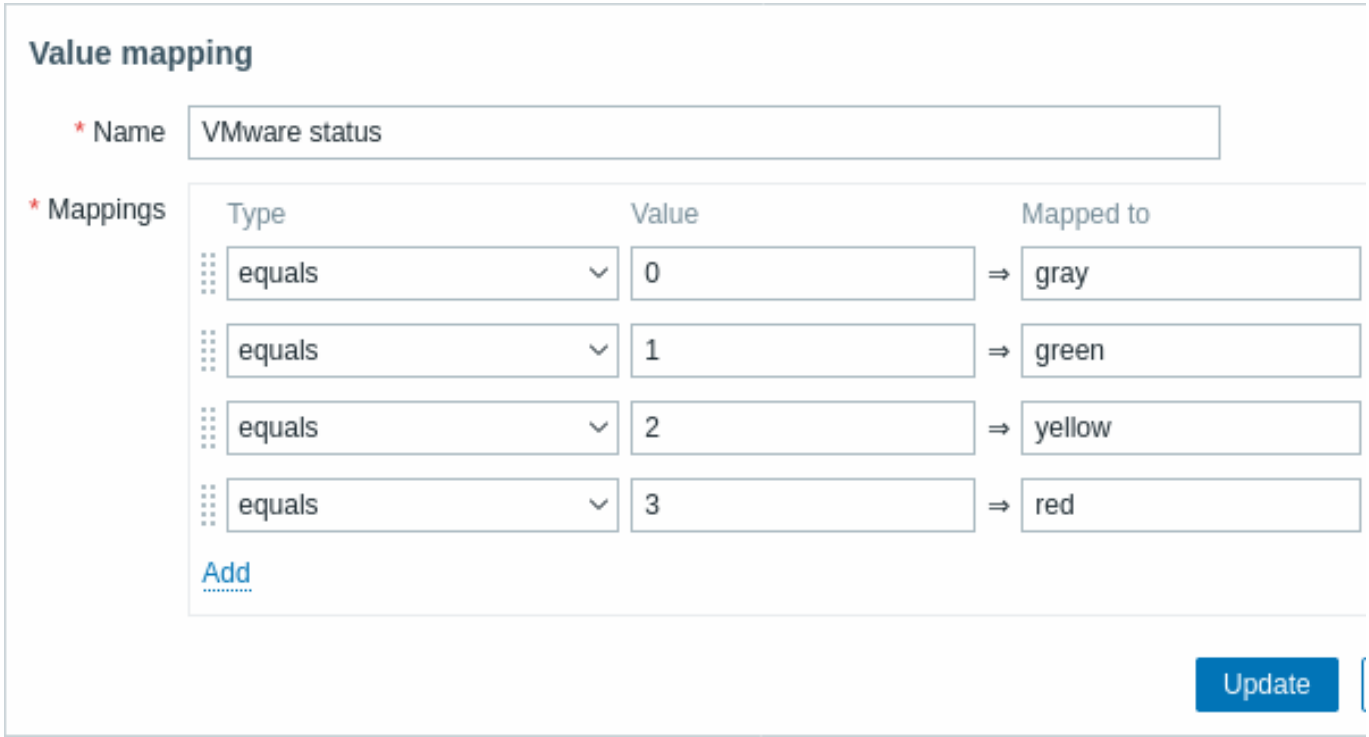
値のマッピングは、対応するテンプレートまたはホストとともにエクスポート/インポートされます。また、ホスト およびテンプレート の一括更新フォームを使用して一括更新することもできます。

設定

値のマッピングを設定するには、次の手順に従います。

1. ホストまたはテンプレートの設定フォームを開きます。

2. *Value mapping* タブで、*Add* をクリックして新しい値のマッピングを追加するか、既存のマッピング名をクリックして編集します。

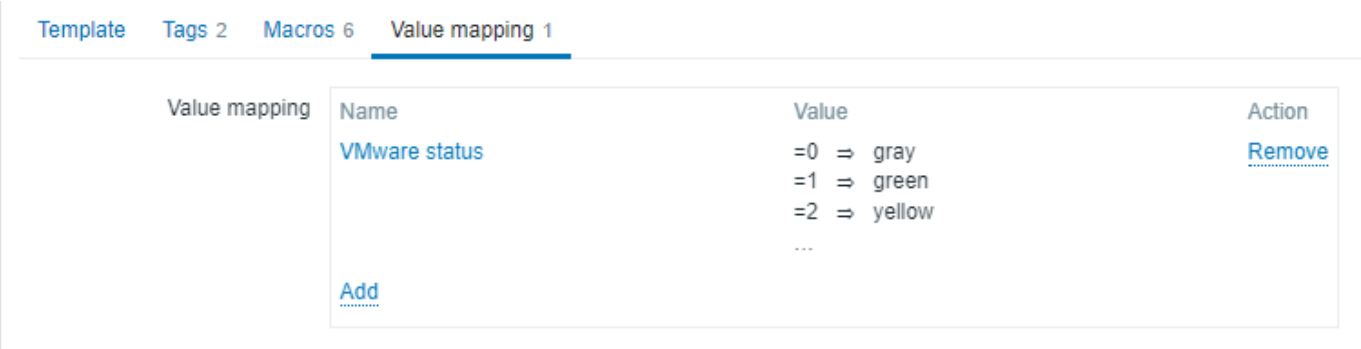


値のマッピングのパラメータ:

Parameter	Description
Name	値のマッピングのセットの一意な名前。
Mappings	数値/文字列の値を文字列表現にマッピングする個別のルール。 マッピングは、ドラッグして並べ替え可能なルールの順序で適用されます。
Type	マッピングの種類: **equals** - 等しい値がマッピングされます。 **is greater than or equals** - 等しい値またはそれより大きい値がマッピングされます。 **is less than or equals** - 等しい値またはそれより小さい値がマッピングされます。 **in range** - 範囲内の値がマッピングされます。範囲は <number1>-<number2> または <number> として表されます。複数の範囲を指定できます（例: 1-10,101-110,201）。 **regexp** - 正規表現 に一致する値がマッピングされます（グローバル正規表現はサポートされていません）。 **default** - 特定のマッピングが設定されていない残りのすべての値がマッピングされます。 範囲のマッピングでは、数値の値タイプ（*is greater than or equals*、*is less than or equals*、*in range*）のみがサポートされます。
Value	受信した値（マッピングの種類に応じて、範囲または正規表現を含めることができます）。
Mapped to	受信した値に対応する文字列表現（最大 64 文字）。

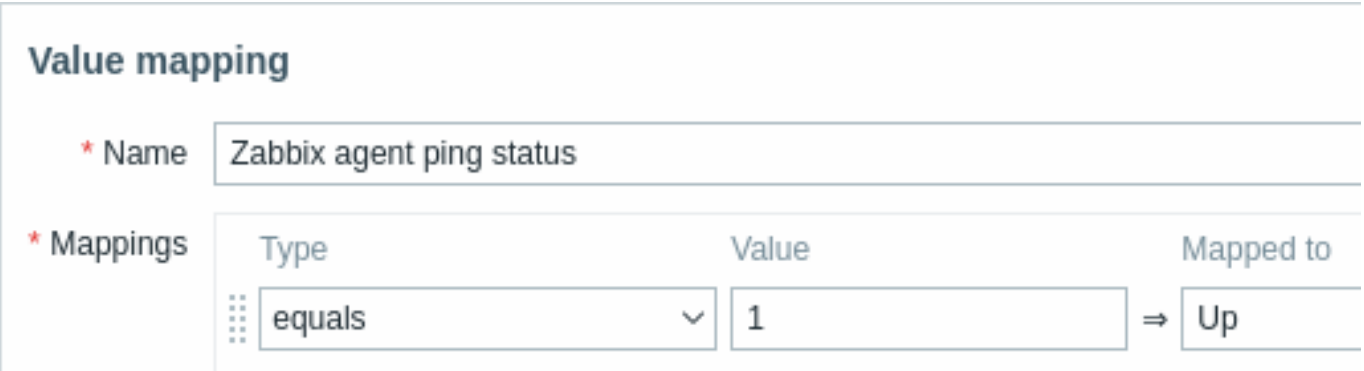
必須入力フィールドには、赤いアスタリスクが付いています。

一覧で値のマッピングを表示する際は、最初の 3 つのマッピングのみが表示され、それ以上のマッピングが存在することは 3 つのドットで示されます。



値マッピングの例

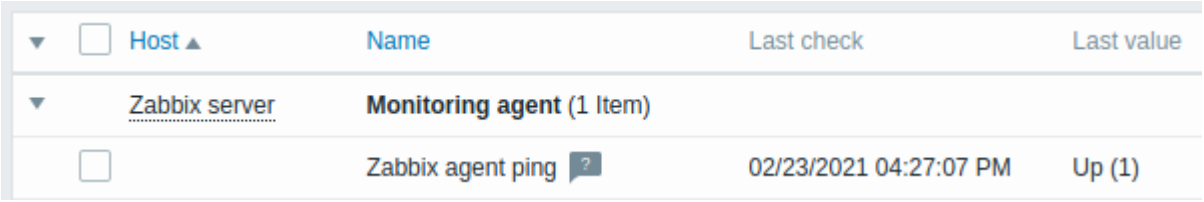
定義済みのエージェントアイテムの 1 つである *Zabbix agent ping* は、テンプレートレベルの値マッピング「Zabbix agent ping status」を使用して、その値を表示します。



アイテムの設定フォームでは、*Value mapping* フィールドにこの値マッピングへの参照があります。

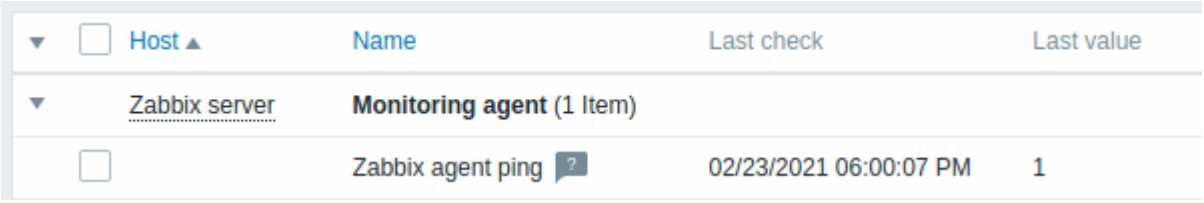


このマッピングは、監視 → 最新データセクションで”Up”（括弧内に生の値を表示）を表示するために使用されます。



Note:
最新データセクションでは、表示される値は 20 文字に短縮されます。
値マッピングが使用されている場合、この短縮はマッピング後の値には適用されず、生の値（括弧内に表示）にのみ適用されます。

定義済みの値マッピングがない場合、表示されるのは”1” のみであり、理解が難しい可能性があります。



9 キュー

概要

キューには、更新待ちのアイテムが表示されます。キューはデータの論理的な表現にすぎません。Zabbix には IPC キューやその他のキュー機構は存在しません。

プロキシによって監視されているアイテムもキューに含まれます。これらは、プロキシの履歴データ更新期間の間、キューに入っているものとしてカウントされます。

キューには、更新時刻がスケジュールされているアイテムのみが表示されます。つまり、次のアイテムタイプはキューから除外されます。

- log、logrt、およびイベントログのアクティブ Zabbix エージェントアイテム
- SNMP トラップアイテム
- トラッパーアイテム
- Web 監視アイテム
- 依存アイテム

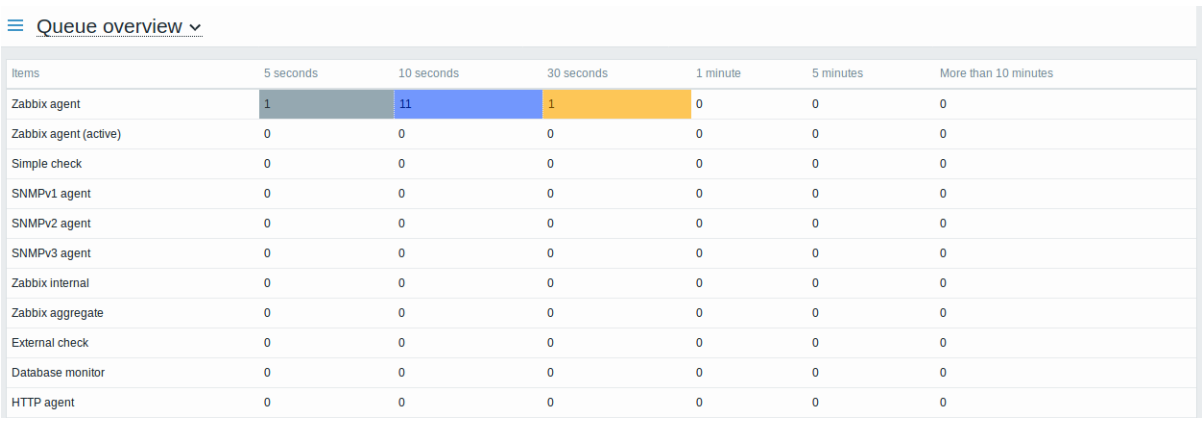
キューに表示される統計は、Zabbix サーバーのパフォーマンスを示す良い指標です。

キューは JSON プロトコルを使用して Zabbix サーバーから直接取得されます。この情報は、Zabbix サーバーが稼働している場合にのみ利用できます。

Attention:
アイテムのインターフェースが接続の問題により利用できなくなった場合、またはエージェントが正常に動作していない場合、そのアイテムはキューにカウントされません。

キューの確認

キューを確認するには、*Administration* > *Queue* > *Queue overview* に移動します。



ここに示されている状態は一般的に「問題なし」なので、サーバーは正常に動作していると考えられます。

キューには、最大 30 秒待機しているアイテムがいくつか表示されています。これらがどのアイテムなのかを把握できると便利です。

そのためには、タイトルのドロップダウンで *Queue details* を選択します。すると、遅延しているアイテムの一覧を確認できます。

Queue details

Scheduled check	Delayed by	Host	Name	Proxy
2019-09-02 11:46:40	58s	My host	CPU idle time	Remote proxy
2019-09-02 11:46:41	57s	My host	CPU interrupt time	Remote proxy
2019-09-02 11:46:42	56s	My host	CPU iowait time	Remote proxy
2019-09-02 11:46:43	55s	My host	CPU nice time	Remote proxy
2019-09-02 11:46:44	54s	My host	CPU softirq time	Remote proxy
2019-09-02 11:46:45	53s	My host	CPU steal time	Remote proxy
2019-09-02 11:46:46	52s	My host	CPU system time	Remote proxy

これらの詳細情報があれば、なぜこれらのアイテムが遅延しているのかを特定できる可能性があります。

1 つまたは 2 つの遅延アイテムであれば、おそらく心配する必要はありません。1 秒後には更新されるかもしれません。しかし、多数のアイテムが長時間遅延している場合は、より深刻な問題がある可能性があります。

関連情報**:** スケジューリング間隔 を使用する際のタイムゾーンの調整。

Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	0	1	1	26	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0
IPMI agent	0	0	0	0	0	0
SSH agent	0	0	0	0	0	0
TELNET agent	0	0	0	0	0	0
JMX agent	0	0	0	0	0	0
Calculated	0	0	0	0	0	0

キュー item

特別な内部 item **zabbix[queue,<from>,<to>]** を使用すると、Zabbix のキューの状態を監視することができます。
 設定された時間だけ遅延したアイテムの数が返されます。詳細については、Internal itemsを参照してください。

10 バリューキャッシュ

概要

トリガー式、計算アイテム、一部のマクロの計算を高速化するために、Zabbix server ではバリューキャッシュオプションがサポートされています。

このインメモリキャッシュを利用すると、直接データベースへ SQL を発行せずに過去のデータへアクセスすることができます。過去の値がキャッシュに存在しない場合、不足分をデータベースから取得し、キャッシュの更新を行います。

アイテムの値は、次のいずれかの状態になるまでバリューキャッシュに残ります:

- アイテムが削除された場合 (キャッシュされた値は、次の構成同期後に削除されます)
- アイテムの値が、トリガーもしくは計算アイテム式で指定された時間またはカウントの範囲外になった場合 (キャッシュされた値は、新しい値を受信した時に削除されます)
- トリガー/計算アイテム式で指定された時間またはカウント範囲が変更され、計算に必要なデータが少なくなった場合 (不要なキャッシュされた値は 24 時間後に削除されます)

Note:
バリューキャッシュの状態は、サーバーのランタイムコントロール でオプション diaginfo (または diaginfo=valuecache) を実行し、診断情報のバリューキャッシュセクションを調べることで確認できます。これは、誤って構成されたトリガーまたは計算アイテムを特定する場合に役立ちます。

バリューキャッシュ機能を有効にするために、オプションの **ValueCacheSize** パラメータが Zabbix server の設定ファイル でサポートされています。

バリューキャッシュを監視するには、以下の 2 つの内部チェックアイテムが利用できます。**zabbix[vcache,buffer,<mode>]** **zabbix[vcache,cache,<parameter>]** 詳細は、内部チェックを参照してください。

11 即時実行

概要

今すぐ実行オプションを使用すると、パッシブチェックを即座に実行できます。

Zabbix でのアイテム値の収集は、設定された更新間隔に従って行われます。
一部のアイテム（例：ローレベルディスカバリルール）は間隔が長く、新しい値をすぐに取得する必要がある場合があります。たとえば、検出可能なリソースの変更を検出するためです。

今すぐ実行オプションは、次のアイテムタイプでサポートされています。

- Zabbix エージェント（パッシブ）
- シンプルチェック
- SNMP エージェント（v1/v2/v3）
- Zabbix 内部
- 外部チェック
- データベースモニタ
- HTTP エージェント
- IPMI エージェント
- SSH エージェント
- TELNET エージェント
- JMX エージェント
- 計算
- 依存アイテム
- スクリプト
- ブラウザ

アイテムのタイプが「依存アイテム」の場合、そのマスターアイテムも上記のいずれかのタイプである必要があります。

設定

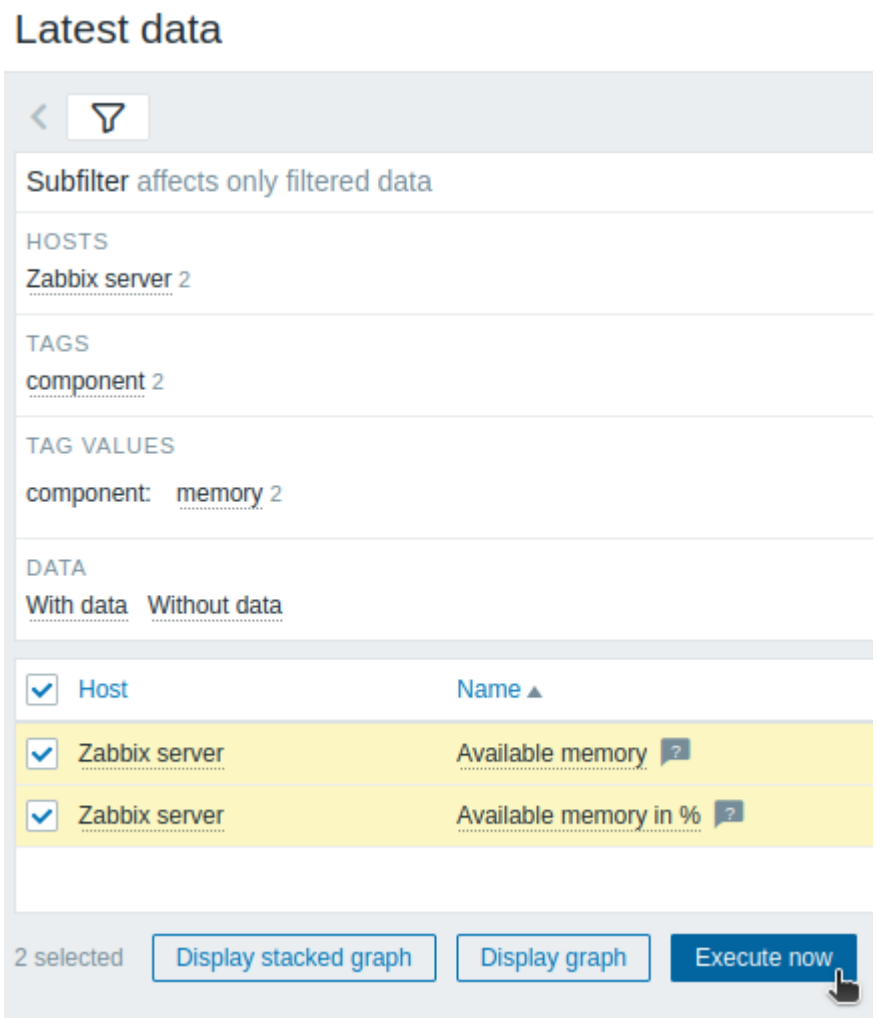
パッシブチェックは、次の場所から実行できます。

- 監視 > 最新データ
- データ収集 > ホスト > アイテム またはディスカバリ
- アイテムメニュー
- アイテムまたはローレベルディスカバリルールの編集中（各設定フォーム内）

Attention:
チェックを実行するには、そのチェックが設定キャッシュ内に存在している必要があります。あわせてCacheUpdateFrequency も参照してください。
チェックの実行前にキャッシュは更新されないため、アイテムまたはローレベルディスカバリルールの設定に対する最近の変更は反映されません。
作成中または新規作成直後のアイテムや LLD ルールをテストする場合は、代わりにTest オプションを使用できます。

たとえば、監視 > 最新データ でパッシブチェックを実行するには、次のようにします。

1. アイテムを選択します。一度に複数のアイテムに対してチェックを実行できます。
2. 今すぐ実行をクリックします。



Note:
最新データでは、ユーザーは *read-write* permission を持つホスト上のアイテム、または *read* permission を持ち、かつそのロールで *Invoke "Execute now" on read-only hosts* action が有効になっている場合に、今すぐ実行を使用できます。

12 エージェントチェックの制限

概要

アイテムチェック、リモートコマンド、またはスクリプトの実行時に、Zabbix エージェントまたは agent 2 がどのアイテムキーの使用を許可または拒否するかを制御できます。

そのためには、これらのエージェント設定パラメータを使用して、許可/拒否ルールを定義します。

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`

`<pattern>` には単一のアイテムキーを含める必要があり、ワイルドカード（`*`）をサポートしています。ワイルドカードは、その位置にある任意の文字を任意の数だけ一致させることができ、アイテムキーまたはパラメータの一致に使用できます（例: `vfs.file.*[*]`）。

Attention:
セキュリティを向上させるため、ワイルドカードの代わりに正確なアイテムキーを使用することを推奨します。詳細は、許可/拒否ルールの保護を参照してください。

他のエージェント設定パラメータとは異なり、`AllowKey` または `DenyKey` パラメータは無制限に指定できます。

重要な注意事項

- すべての`system.run` アイテムは、デフォルトで無効になっています（`DenyKey` が空の場合でも）。これは、最後のルールとして `DenyKey=system.run[*]` が設定されている場合と同じです。このため、他の `system.run` アイテムを明示的に拒否しなくても、特定の `system.run` アイテムを許可できます。

- AllowKey で指定したアイテムは、DenyKey にも指定する必要があります（system.run アイテムを除く）。そうしないと、Zabbix エージェントは起動しません。
- 可能であれば、AllowKey を使用して必要なアイテムのみを許可し、それ以外はすべて拒否してください。一部のキーは、パストラバーサルを介して意図しないファイルを読み取るために悪用される可能性があります（例: vfs.file.contents["../../../../etc/passwd"]）。また、新しいバージョンの Zabbix エージェントでは、DenyKey ルールでカバーされていないキーが導入される可能性があります。
- AllowKey および DenyKey の設定は、HostnameItem、HostMetadataItem、HostInterfaceItem のエージェントパラメータには影響しません。
- 拒否されたアイテムは、ヒントやエラーメッセージなしで未サポートになります。例えば:
  - Zabbix エージェントの--print (-p) コマンドラインパラメータでは、拒否されたアイテムキーは表示されません。
  - Zabbix エージェントの--test (-t) コマンドラインパラメータでは、拒否されたアイテムキーに対して"Unsupported item key." が返されます。
  - ログ記録が有効な場合（LogRemoteCommands=1）、Zabbix エージェントのログファイルには、拒否されたリモートコマンドは記録されません。

Allow/deny ルールの順序

AllowKey または DenyKey ルールは無制限に指定できますが、その順序は重要です。

- ルールは上から下へ、1 つずつ評価されます。
- アイテムキーがルールに一致すると、許可または拒否され、ルールの評価は停止します。

例えば、vfs.file.contents[/etc/passwd] を評価する場合、ルールは次のように処理されます。

```
AllowKey=vfs.file.contents[/tmp/app.log]    # アイテムキーパターンが一致しないため、エージェントは次のルールに進みます。
AllowKey=vfs.file.contents[/etc/passwd]     # アイテムキーパターンが一致するため、エージェントはアイテムチェックを許可
DenyKey=vfs.file.*[*]                       # 評価が停止しているため、エージェントはこのルールを無視します。
```

次のルール順序では、アイテムチェックは拒否されます。

```
DenyKey=vfs.file.*[*]                       # アイテムキーパターンが一致するため、エージェントはアイテムチェックを拒否
AllowKey=vfs.file.contents[/etc/passwd]     # 評価が停止しているため、エージェントはこのルールを無視します。
AllowKey=vfs.file.contents[/tmp/app.log]    # 評価が停止しているため、エージェントはこのルールを無視します。
```

例

以下の例は、AllowKey と DenyKey の一般的な設定パターンを示しています。

特定のチェックとコマンドを許可する

2 つのvfs.fileアイテムチェックと、2 つのsystem.runコマンドのみを許可します。

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

Note:
DenyKey=system.run[*] を設定する必要はありません。これは、他のすべての system.run コマンドはデフォルトで拒否されるためです。

スクリプトの許可

利用可能なすべての方法で、Zabbix エージェントがホスト上でスクリプトを実行できるようにします。

- Web インターフェースまたは API 経由で実行できるグローバルスクリプト（この方法では常に system.run[myscript.sh] キーを使用します）
- アクションの実行内容からのリモートコマンド（この方法では常に system.run[myscript.sh,nowait] キーを使用します）
- スクリプトを含むsystem.run Zabbix エージェントのアイテム。例:
  - system.run[myscript.sh]
  - system.run[myscript.sh,wait]
  - system.run[myscript.sh,nowait]

```
AllowKey=system.run[myscript.sh,*]
```

wait/nowait パラメータを制御するには、別のルールを設定する必要があります。
たとえば、system.run[myscript.sh,wait] アイテムのみを許可し、他の方法を除外できます。

```
AllowKey=system.run[myscript.sh,wait]
```

Allow/Deny ルールの保護

この例では、過度に許容的な AllowKey または DenyKey ルールを安全にする方法を示します。

次のルールを考えてみましょう。

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

Note:
Windows では、パス内のスペースをキャレット (^) を使用してエスケープする必要があります。

これらのルールにはワイルドカード (*) が含まれており、悪用される可能性があります。

- test.bat スクリプトは、意図しない引数を含む任意の引数で実行できます。
- vfs.file.* パターンは、パラメータのないアイテムキーにのみ一致します。しかし、すべてのvfs.file アイテムにはパラメータが必要です。
- system.cpu.load[*] パターンは、パラメータ付きのアイテムキーにのみ一致します。しかし、system.cpu.load アイテムではパラメータは必須ではありません。

これらのルールを安全にするには、test.bat の実行を特定の引数の場合にのみ明示的に許可し、正しいアイテムキーパターンを拒否します。例えば、次のようにします。

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

次のコマンドを実行してルールをテストできます。これらは ZBX_NOTSUPPORTED を返します。

```
cd "C:\Program Files\Zabbix Agent 2"
zabbix_agent2.exe -t system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat debug"]
zabbix_agent2.exe -t vfs.file.size["C:\ProgramData\MyApp\config.ini"]
zabbix_agent2.exe -t vfs.file.contents["C:\Windows\System32\drivers\etc\hosts"]
zabbix_agent2.exe -t system.cpu.load
zabbix_agent2.exe -t system.cpu.load[all,avg1]
```

パターンの例

次の表は、アイテムキーのパターンがどのように一致するかを示しています。

- キーは、一致条件列のすべての条件を満たす場合にのみ、パターンに一致します。
- パラメータは必ず角括弧で完全に囲む必要があります（例: vfs.file.contents[* および vfs.file.contents*utf8] は無効なパターンです）。

Pattern	Matches	Examples
*	パラメータの有無にかかわらず任意のキー	
vfs.file.*	キーが vfs.file. で始まる パラメータなし	一致: vfs.file.size vfs.file.contents 不一致: vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	キーが vfs. で始まる キーが .contents で終わる パラメータなし	一致: vfs..contents vfs.mount.point.file.contents 不一致: vfs.contents vfs.file.contents[]

Pattern	Matches	Examples
vfs.file.*[*]	キーが vfs.file. で始まる 任意または空のパラメータ	一致 vfs.file.get.custom[] vfs.file.size[/var/log/app.log, utf8] 不一致: vfs.file.get.custom
vfs.file.contents	キーが vfs.file.contents である パラメータなし	一致: vfs.file.contents 不一致: vfs.file.contents[/etc/passwd]
vfs.file.contents[]	キーが vfs.file.contents[] である 空のパラメータ	一致: vfs.file.contents[] 不一致: vfs.file.contents
vfs.file.contents[*]	キーが vfs.file.contents である 任意または空のパラメータ	一致: vfs.file.contents[/path/to/file] 不一致: vfs.file.contents
vfs.file.contents[/etc/passwd,*]	キーが vfs.file.contents である 1 番目のパラメータが /etc/passwd である 2 番目のパラメータが任意または空である	一致: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] 不一致: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	キーが vfs.file.contents である 1 番目のパラメータに passwd が含まれる 2 番目のパラメータなし	一致: vfs.file.contents[/etc/passwd] 不一致: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	キーが vfs.file.contents である 1 番目のパラメータに passwd が含まれる 2 番目のパラメータが任意または空である	一致: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] 不一致: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	キーが vfs.file.contents である 1 番目のパラメータが /etc/passwd である 2 番目のパラメータが utf8 である	一致: vfs.file.contents[/etc/passwd,utf8] 不一致: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

3 トリガー

概要

トリガーは、アイテムによって収集されたデータを「評価」する論理式であり、現在のシステム状態を表します。

アイテムはシステムデータの収集に使用されますが、警告が必要な状態や注意を要する条件を待ちながら、常にそれらのデータを追い続けるのは非常に非現実的です。データを「評価」する役割は、トリガー式に任せることができます。

トリガー式では、どのようなデータ状態を「許容可能」とするかのしきい値を定義できます。そのため、受信したデータが許容可能な状態を超えた場合、トリガーは「発火」し、ステータスが PROBLEM に変わります。

トリガーは次のステータスを持つ場合があります。

Status	Description
OK	通常のトリガーのステータスです。
Problem	何らかの問題が発生しています。たとえば、プロセッサー負荷が高すぎる場合です。
Unknown	トリガー値を計算できません。Unknown statusを参照してください。

単純なトリガーでは、たとえば CPU 負荷のようなデータの 5 分間平均に対してしきい値を設定したい場合があります。これは、次の条件を持つトリガー式を定義することで実現できます。

- 'avg' 関数がアイテムキーで受信した値に適用される
- 評価には 5 分間の期間が使用される
- しきい値として「2」が設定される

```
avg(/host/key,5m)>2
```

このトリガーは、5 分間平均が 2 を超えた場合に「発火」し（PROBLEM になります）。

より複雑なトリガーでは、式に複数の関数と複数のしきい値の組み合わせを含めることができます。あわせてTrigger expression も参照してください。

バイナリのdata type値を返すアイテムに対しては、トリガーを作成できません。

Note:
トリガーを有効化した後（設定ステータスを *Disabled* から *Enabled* に変更した後）、トリガー式は、その中のアイテムが値を受信した時点、またはdate and timeおよび/またはnodata()関数を処理する時刻になると、直ちに評価されます。

ほとんどのトリガー関数はアイテムの値のヒストリデータに基づいて評価されますが、一部のトリガー関数は長期的な分析に使用されます。**trendavg()**、**trendcount()** などはトレンドデータを使用します。

計算時間

トリガーは、Zabbix サーバーが式の一部である新しい値を受信するたびに再計算されます。新しい値を受信すると、式に含まれる各関数が再計算されます（新しい値を受信した関数だけではありません）。

さらに、式でdate and timeおよび/またはnodata()関数が使用されている場合、トリガーは新しい値を受信するたびに再計算され、さらに 30 秒ごとにも再計算されます。

Date and timeおよび/またはnodata()関数は、Zabbix history syncer プロセスによって 30 秒ごとに再計算されます。

トレンド関数のみを参照するトリガーは、式内の最小の時間期間ごとに 1 回評価されます。あわせてtrend functionsも参照してください。

評価期間

評価期間は、アイテムの履歴を参照する関数で使用されます。これにより、対象とする間隔を指定できます。指定方法は、時間期間（30s、10m、1h）または値の範囲（#5 - 直近 5 個の値）です。

評価期間は「現在」までを基準に測定されます。ここでの「現在」はトリガーの最新の再計算時刻（上記の計算時刻を参照）であり、サーバーの「現在」時刻ではありません。

評価期間では、次のいずれかを指定します。

- 「現在-時間期間」から「現在」までのすべての値を対象とする（または、タイムシフトを使用する場合は、「現在-タイムシフト-時間期間」から「現在-time_shift」まで）
- 過去から「現在」までの値のうち、最大で num 個までを対象とする
  - 指定した時間期間または num 個の値に対して利用可能な値が 0 個の場合、この関数を使用するトリガーまたは計算アイテムは未サポートになります

注意:

- トリガーで使用される関数が 1 つだけ（データ履歴を参照する関数）である場合、「現在」は常に最後に受信した値となります。例えば、最後の値を 1 時間前に受信していた場合、評価期間は 1 時間前の最新値までとして扱われます。
- 新しいトリガーは、最初の値を受信するとすぐに計算されます（履歴関数の場合）。日付と時刻 およびnodata() 関数については、30 秒以内に計算されます。したがって、設定された評価期間（例えば 1 時間）がトリガー作成後まだ経過していなくても、トリガーは計算されます。評価範囲が例えば直近 10 個の値に設定されている場合でも、最初の値の受信後にトリガーは計算されます。

不明なステータス

次の場合、トリガー式に不明なオペランドが現れることがあります。

- 未サポートのアイテムが使用されている
- サポートされているアイテムの関数評価でエラーが発生する

この場合、通常トリガーは「不明」と評価されます（ただし、いくつかの例外があります）。詳細は、不明なオペランドを含む式を参照してください。

不明なトリガーについて通知を受け取ることも可能です。

1 トリガーの設定

概要

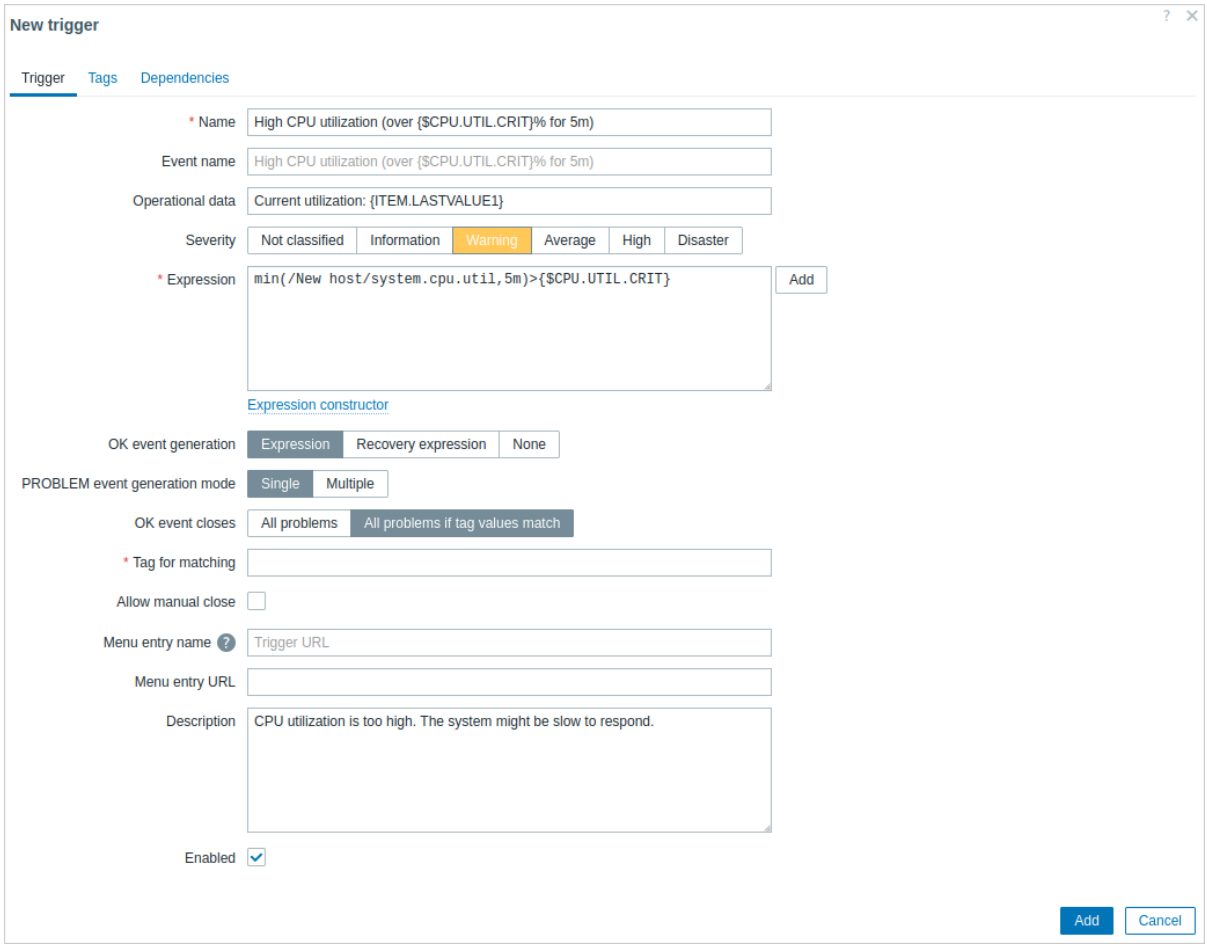
トリガーを設定するには、次の手順を実行します。

- データ収集 > ホストに移動します
- ホストの行にある トリガーをクリックします
- 右側の トリガーの作成をクリックします（既存のトリガーを編集する場合は、トリガー名をクリックします）
- フォームにトリガーのパラメータを入力します

トリガーおよびその計算時間の詳細については、一般情報も参照してください。

設定

トリガータブには、トリガーの主要な属性がすべて含まれています。

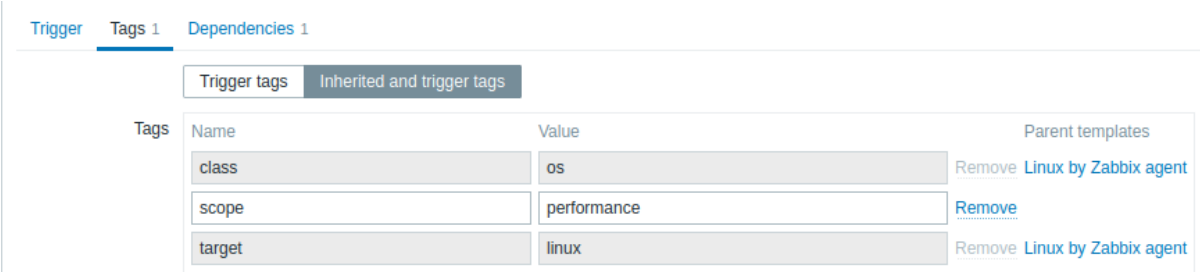


必須の入力フィールドには、赤いアスタリスクが付いています。

Parameter	Description
Name	トリガー名。 サポートされているマクロは次のとおりです: {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {ITEM.LOG.*}, および {$MACRO} ユーザーマクロ。 **$1, $2...$9** マクロは、式内の 1 番目、2 番目...9 番目の定数を参照するために使用できます。 注: $1-$9 マクロは、比較的単純で明快な式内の定数を参照している場合に正しく展開されます。例えば、名前が”Processor load above $1 on {HOST.NAME}” の場合、式が last(/New host/system.cpu.load[percpu,avg1])>5 であれば、自動的に”Processor load above 5 on New host” に変わります。
Event name	定義されている場合、この名前が障害イベント名の作成に使用されます。デフォルトでは、イベント名はトリガー名と同じです。 イベント名は、障害データを含む意味のあるアラートを作成するために使用できます（例を参照）。 トリガー名と同じマクロセットに加えて、{TIME}、{TIMESTAMP}、および {?EXPRESSION} 式マクロがサポートされています。
Operational data	運用データでは、マクロとともに任意の文字列を定義できます。マクロは、*Monitoring* > *Problems* でリアルタイムデータに動的に展開されます。一方、トリガー名内のマクロ（上記参照）は障害発生時点の値に展開され、静的な障害名の基礎となりますが、運用データ内のマクロは最新情報を動的に表示する機能を維持します。トリガーレベルで運用データが設定されていない場合、式に含まれるすべてのアイテムの最新値が表示されます。 トリガー名と同じマクロセットがサポートされています。
Severity	ボタンをクリックして、必要なトリガーの深刻度を設定します。
Expression	障害の条件を定義するために使用される論理式。 時間サフィックスおよびメモリサイズサフィックスがサポートされています。 式に含まれるすべての条件が満たされると、つまり式が TRUE と評価されると障害が作成されます。 *Recovery expression* で追加の復旧条件が指定されていない限り、式が FALSE と評価されるとすぐに障害は解決されます。
OK event generation	OK イベント生成オプション: **Expression** - 障害イベントと同じ式に基づいて OK イベントが生成されます。 **Recovery expression** - 障害式が FALSE と評価され、かつ復旧式が TRUE と評価された場合に OK イベントが生成されます。 **None** - この場合、トリガーが自動的に OK 状態に戻ることはありません。
Recovery expression	元の障害式がすでに FALSE と評価された後に、障害が解決される前に満たす必要がある追加条件を定義する論理式（任意）。 復旧式は、トリガーのヒステリシスに役立ちます。障害式がまだ TRUE の場合、復旧式だけで障害を解決することはできません。 このフィールドは、*OK event generation* で’Recovery expression’ が選択されている場合にのみ使用できます。
PROBLEM event generation mode	障害イベントを生成するモード: **Single** - トリガーが初めて’Problem’ 状態になったときに単一のイベントが生成されます。 **Multiple** - トリガーが’Problem’ と評価されるたびにイベントが生成されます。
OK event closes	OK イベントがクローズする対象を選択します: **All problems** - このトリガーのすべての障害。 **All problems if tag values match** - イベントタグ値が一致するトリガー障害のみ。
Tag for matching	イベント相関に使用するイベントタグ名を入力します。 このフィールドは、*OK event closes* プロパティで’All problems if tag values match’ が選択されている場合に表示され、この場合は必須です。
Allow manual close	このトリガーによって生成された障害イベントの手動クローズを許可するにはチェックします。手動クローズは、障害イベントを確認応答する際に可能です。
Menu entry name	空でない場合、ここで入力した名前（最大 64 文字）が、*Menu entry URL* パラメータで指定されたトリガー URL のラベルとして、複数の Web インターフェース上の場所で使用されます。空の場合、デフォルト名 *Trigger URL* が使用されます。 トリガー名と同じマクロセットに加えて、{EVENT.ID}、{HOST.ID}、および {TRIGGER.ID} がサポートされています。
Menu entry URL	空でない場合、ここで入力した URL（最大 2048 文字）は、複数の Web インターフェース上の場所にあるevent menuでリンクとして利用できます。例えば、*Monitoring* > *Problems* または *Problems* ダッシュボードウィジェットで障害名をクリックしたときです。 トリガー名と同じマクロセットに加えて、{EVENT.ID}、{HOST.ID}、および {TRIGGER.ID} がサポートされています。注: secret 値を持つユーザーマクロは URL 内では展開されません。

Parameter	Description
Description	このトリガーに関する追加情報を提供するためのテキストフィールドです。特定の障害を修正するための手順、担当スタッフの連絡先詳細などを含めることができます。 トリガー名と同じマクロセットがサポートされています。
Enabled	必要に応じて、このボックスのチェックを外すことでトリガーを無効にできます。 無効化されたトリガーの障害は Web インターフェースに表示されなくなりますが、削除はされません。

Tags タブでは、トリガーレベルのタグを定義できます。このトリガーのすべての障害には、ここで入力した値のタグが付けられます。



さらに、トリガーがそのテンプレートに由来する場合、*Inherited and trigger tags* オプションを使用してテンプレートレベルで定義されたタグを表示できます。同じタグを持つテンプレートが複数ある場合、これらのタグは 1 回だけ表示され、テンプレート名はカンマで区切られます。トリガーはホストレベルのタグを「継承」して表示することはありません。

Parameter	Description
Name/Value	トリガーイベントを識別するためのカスタムタグを設定します。 タグは、タグ名と値のペアです。名前のみを使用することも、値と組み合わせることもできます。 トリガーは同じ名前で異なる値を持つ複数のタグを持つことができます。 ユーザーマクロ、コンテキスト付きユーザーマクロ、ローレベルディスカバリマクロ、および {{ITEM.VALUE}}、{{ITEM.LASTVALUE}} を使用したマクロ関数がイベントタグでサポートされています。ローレベルディスカバリマクロはマクロコンテキスト内で使用できます。 {TRIGGER.ID} マクロはトリガータグ値でサポートされています。これは、トリガープロトタイプから作成されたトリガーを識別したり、例えばメンテナンス中にこれらのトリガーからの障害を抑制したりするのに役立つ場合があります。 展開後の値の合計長が 255 を超える場合、255 文字に切り詰められます。 イベントタグでサポートされるすべてのマクロを参照してください。 イベントタグは、イベント相関やアクション条件で使用でき、*Monitoring* > *Problems* または *Problems* ウィジェットでも確認できます。

Dependencies タブには、トリガーのすべての依存関係 が含まれています。

新しい依存関係を追加するには、*Add* をクリックします。

Note:
既存のトリガーを開き、*Clone* ボタンを押してから別の名前で保存することでも、トリガーを設定できます。

評価式のテスト

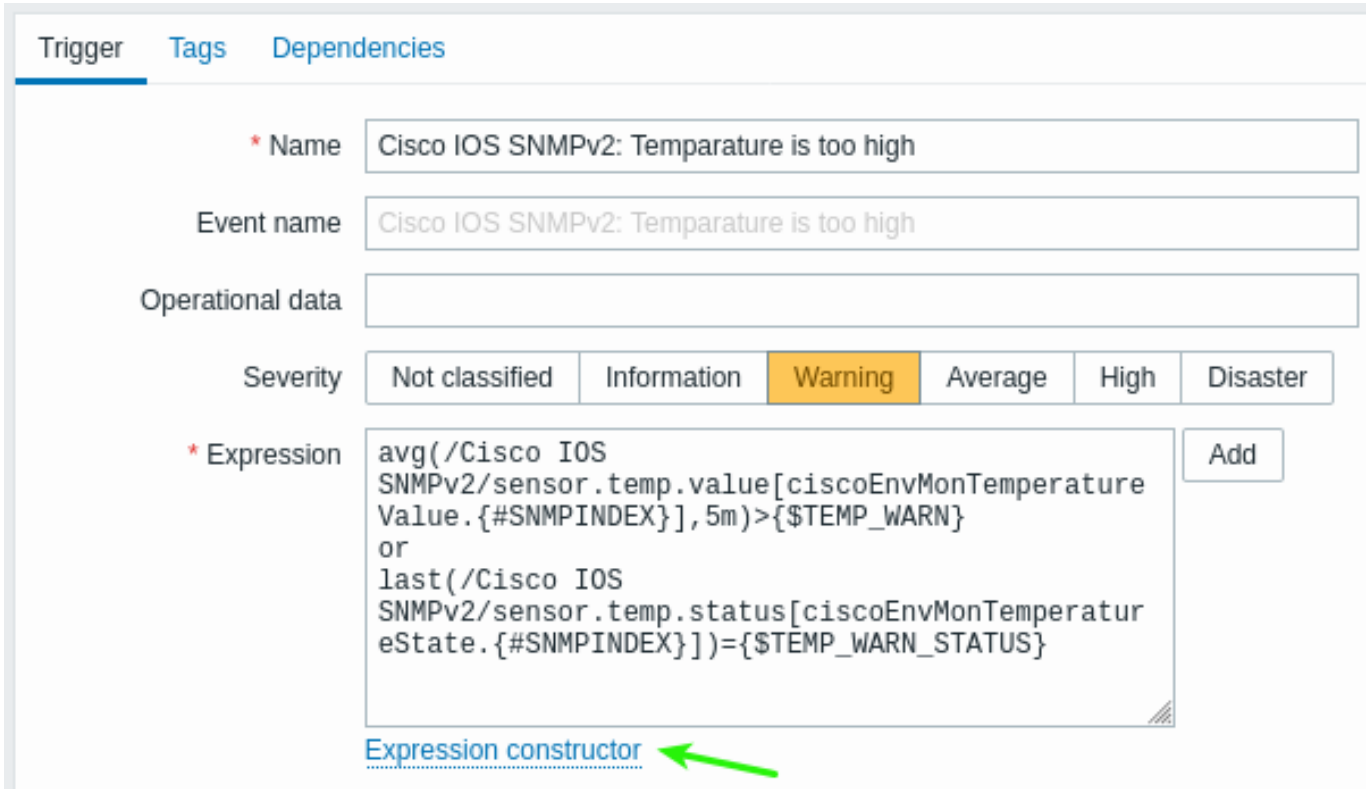
設定した trigger 式が、受け取った値によってどのような式結果になるかをテストすることが可能である。

例として、公式テンプレートにある以下の式を挙げる。

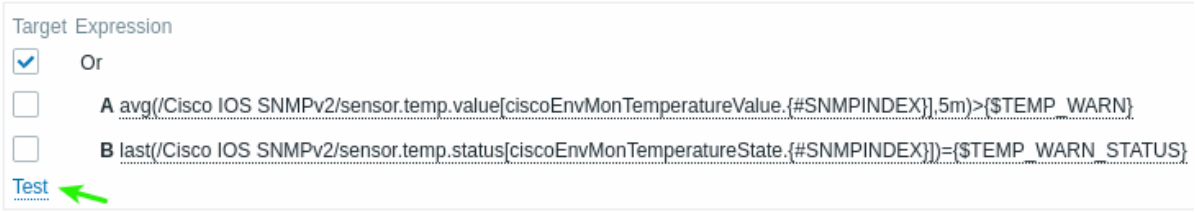
```
avg(/Cisco IOS SNMPv2/sensor.temp.value[ciscoEnvMonTemperatureValue.{#SNMPINDEX}],5m)>{$TEMP_WARN}
```

```
last(/Cisco IOS SNMPv2/sensor.temp.status[ciscoEnvMonTemperatureState.{#SNMPINDEX}])={$TEMP_WARN_STATUS}
```

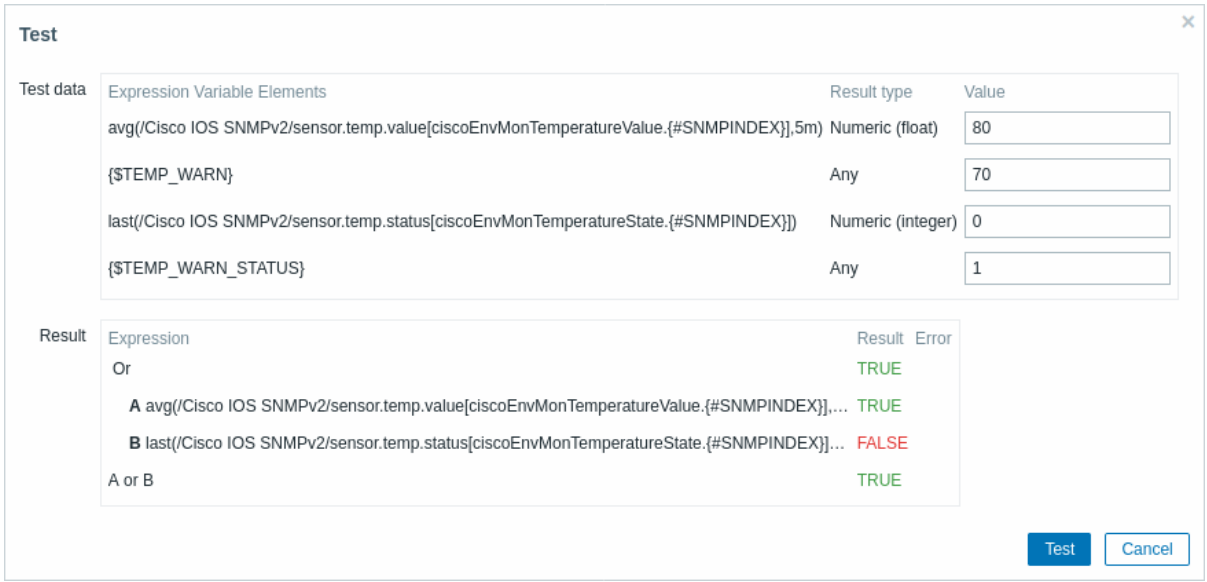
式をテストするには、式フィールドの下にある *Expression constructor* をクリックします。



Expression コンストラクタでは、個々の式がすべてリストアップされます。テスト・ウィンドウを開くには、
 式リストの下にある *Test* をクリックします。



テスト画面では、サンプル値（この例では'80'，'70'，'0'，'1'）を入力し、*Test* ボタンをクリックすると、式の結果を確認することができます。



個々の表現だけでなく、全体の表現の結果も見ることができます。

"TRUE " は、指定された式が正しいことを意味する。このケース A では、"80" は {$TEMP_WARN} の指定値 (この例では"70" です) より大きい。
 予想通り、"TRUE " の結果が表示される。

"FALSE " は、指定された式が正しくないことを意味します。この場合この特定のケース B では、
 {$TEMP_WARN_STATUS} の"1 " が"1 " と等しい必要がある。"1" は、指定された値、この例では"0 " と等しくなければならない。
 予想通り、"FALSE" の結果

が表示される。

式のタイプは"OR " である。指定された条件のうち少なくとも 1 つ（この例では A または B）が
 指定された条件（この例では A または B）の少なくとも 1 つが"TRUE" であれば、全体の結果も"TRUE" となります。
 つまり、現在値が警告値を超えており、問題が発生していることを意味します。

2 トリガー条件式

概要

トリガーで使用される式は非常に柔軟です。これらを使用して、監視対象の統計に関する複雑な論理テストを作成できます。

単純な式では、いくつかのパラメータを指定してアイテムに適用される関数を使用します。関数は結果を返し、その結果は演算子と定数を使用してしきい値と比較されます。

単純で有用な式の構文は `function(/host/key,parameter)<operator><constant>` です。

例えば:

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

これは、直近 5 分間に受信したバイト数が常に 100 キロバイトを超えていた場合にトリガーします。

構文はまったく同じですが、機能の観点からはトリガー式には 2 種類あります:

- 障害時の式 - 障害の条件を定義します
- 復旧時の式（任意）- 障害の解決に関する追加条件を定義します

障害時の式のみを定義した場合、この式は障害のしきい値と障害復旧のしきい値の両方として使用されます。障害時の式が TRUE と評価されると、障害が発生します。障害時の式が FALSE と評価されると、障害は解決されます。

障害時の式と補足の復旧時の式の両方を定義した場合、障害の解決はより複雑になります。障害時の式が FALSE であるだけでなく、復旧時の式も TRUE でなければなりません。これは、ヒステリシスを作成し、トリガーのフラッピングを回避するのに役立ちます。

> **Note:**
> 復旧時の式で {TRIGGER.VALUE} マクロを使用しても意味がありません。これは、この式がトリガーが「障害」状態のときにのみ評価されるためです。したがって、式の評価中、{TRIGGER.VALUE} は常に"1"（「障害」状態を示す）に展開されます。

関数

関数を使用すると、収集した値の計算（平均、最小、最大、合計）、文字列の検索、現在時刻やその他の要素の参照ができます。

完全な一覧は、サポートされている関数で確認できます。

通常、関数は比較のために数値を返します。
文字列を返す場合は、**=** および **<>** 演算子で比較できます（例を参照）。

関数パラメータ

関数パラメータでは、以下を指定できます。

- ホストとアイテムキー（ホストのアイテム履歴のみを参照する関数）
- 関数固有のパラメータ
- その他の式（ホストのアイテム履歴を参照する関数では使用不可。例についてはその他の式を参照してください）

ホストとアイテムキーは `/host/key` として指定できます。
最初のパラメータでホスト名を省略すること（つまり `function(//key,parameter,...)` の形式）は、特定のコンテキストでのみサポートされています。

- 計算アイテムの数式内
- 以下で使用できる式マクロ内:
  - イベント名 フィールド
  - グラフ名
  - 「ホスト」および「トリガー」のマップ要素のラベル

これらのコンテキストでは、`{HOST.HOST}` マクロも使用できます。
`{HOST.HOST<1-9>}` は、イベント名フィールドおよび「トリガー」マップ要素で、トリガー式内の特定のアイテムを参照する場合に使用できます。
これらのコンテキストでホスト名が省略されている場合、または {HOST.HOST} に置き換えられている場合、参照先はトリガー式内の最初のアイテム、またはグラフ内の最初のアイテムになります。
これらのサポート対象コンテキスト以外でトリガー式内のホスト名を省略すると、エラーになります。
イベント名マクロでのダブルスラッシュの使用例については、例 18を参照してください。

参照されるアイテムはサポート状態である必要があります（ただし、**nodata()** 関数は例外で、未サポートのアイテムについても計算されます）。

関数パラメータとしての他のトリガー式は、トリガーでは非履歴関数に限定されますが、この制限は計算アイテムには適用されません。

関数固有のパラメータ

関数固有のパラメータはアイテムキーの後に配置され、カンマでアイテムキーと区切られます。これらのパラメータの完全な一覧については、サポートされている関数を参照してください。

数値関数の多くは、パラメータとして時間を受け付けます。時間を示すには、秒または時間のサフィックスを使用できます。ハッシュ記号を前に付けると、パラメータは異なる意味を持ちます。

Expression	Description
sum(/host/key,**10m**)	直近 10 分間の値の合計。
sum(/host/key,**#10**)	直近 10 個の値の合計。

ハッシュ記号付きのパラメータは、関数 **last** では異なる意味を持ちます。これは N 番目の直前の値を表します。したがって、値が 30、70、20、60、50（新しいものから古いものの順）である場合:

- last(/host/key,#2) は'70' を返します
- last(/host/key,#5) は'50' を返します

タイムシフト

関数パラメータとして時間または値の件数を指定する場合、オプションのタイムシフトをサポートしています。このパラメータを使用すると、過去の一定期間のデータを参照できます。

タイムシフトは、現在時刻を指定する now で始まり、その後に +N<time unit> または -N<time unit> が続き、N 個の時間単位を加算または減算します。

例えば、avg(/host/key,1h:now-1d) は、1 日前の 1 時間の平均値を返します。

Attention:
月 (M) および年 (y) で指定するタイムシフトは、trend functions でのみサポートされています。その他の関数では、秒 (s)、分 (m)、時間 (h)、日 (d)、週 (w) をサポートしています。

絶対時間期間を使用したタイムシフト

絶対時間期間はタイムシフトパラメータでサポートされています。例えば、1 日では午前 0 時から午前 0 時まで、1 週間では月曜日から日曜日まで、1 か月では月初から月末までを指定できます。

絶対時間期間のタイムシフトは、現在時刻を指定する now で始まり、その後に任意の数の時間操作が続きます。/<time unit> は時間単位の開始と終了を定義します。例えば、1 日では午前 0 時から午前 0 時までです。+N<time unit> または -N<time unit> は、N 個の時間単位を加算または減算します。

タイムシフトの値は 0 以上を指定できますが、時間期間の最小値は 1 である点に注意してください。

Parameter	Description
1d:now/d	昨日
1d:now/d+1d	今日
2d:now/d+1d	直近 2 日間
1w:now/w	先週
1w:now/w+1w	今週

その他の式

関数パラメータには、次の構文のように他の式を含めることができます。

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

関数がアイテムの履歴を参照する場合、他の式は使用できないことに注意してください。例えば、次の構文は使用できません。

~~min(/host/key,#5*10)~~

演算子

トリガーでサポートされている演算子は以下のとおりです （実行優先順位の高い順）:

優先順位	演算子	定義	不明な値 に関する注記	オペランドを強制的に float にキャスト[1]
1	**-**	単項マイナス	**-**Unknown → Unknown	はい
2	**not**	論理 NOT	**not** Unknown → Unknown	はい
3	*****	乗算	0 * Unknown → Unknown (算術演算で Unknown を失わないため、 0 ではなく Unknown になります) 1.2 * Unknown → Unknown	はい
	/	除算	Unknown / 0 → error Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	はい
4	**+**	算術加算	1.2 + Unknown → Unknown	はい
	-	算術減算	1.2 - Unknown → Unknown	はい
5	**<**	より小さい。演算子は次のように定義されます： A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	はい
	<=	以下。演算子は次のように定義されます： A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	はい
	>	より大きい。演算子は次のように定義されます： A>B ⇔ (A>B+0.000001)		はい
	>=	以上。演算子は次のように定義されます： A>=B ⇔ (A≥B-0.000001)		はい
6	**=**	等しい。演算子は次のように定義されます： A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		いいえ[1]
	<>	等しくない。演算子は次のように定義されます： A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		いいえ[1]
7	**and**	論理 AND	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	はい

優先順位	演算子	定義	不明な値 に関する注記	オペランドを強制的に float にキャスト[1]
8	**or**	論理 OR	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	はい

[1] 文字列オペランドも、次の場合は数値にキャストされます：

- もう一方のオペランドが数値である
- **=** または **<>** 以外の演算子がオペランドに使用されている

関係演算子（<、<=、>、>=）では、キャストに失敗した場合（たとえば "" の場合）、式の結果は Unknown になります（文字列比較へのフォールバックはありません）。

等価演算子（=、<>）では、キャストに失敗した場合、オペランドは生の文字列として比較されます。

not、**and**、**or** 演算子は大文字と小文字を区別し、小文字でなければなりません。また、これらはスペースまたは括弧で囲まれている必要があります。

単項 **-** と **not** を除くすべての演算子は、左から右への結合性を持ちます。単項 **-** と **not** は非結合です（つまり、**--1** や **not not 1** の代わりに **-(-1)** および **not (not 1)** を使用する必要があります）。

評価結果：

- **<**、**<=**、**>**、**>=**、**=**、**<>** 演算子は、指定された関係が真であればトリガー式で'1' を返し、偽であれば'0' を返します。少なくとも 1 つのオペランドが Unknown の場合、結果は Unknown になります。
- **and** は、既知のオペランドに対しては、両方のオペランドが'0' と等しくない場合に'1' を返し、それ以外の場合は'0' を返します。未知のオペランドに対しては、**and** は一方のオペランドが'0' と等しい場合にのみ'0' を返し、それ以外の場合は'Unknown' を返します。
- **or** は、既知のオペランドに対しては、いずれかのオペランドが'0' と等しくない場合に'1' を返し、それ以外の場合は'0' を返します。未知のオペランドに対しては、**or** は一方のオペランドが'0' と等しくない場合にのみ'1' を返し、それ以外の場合は'Unknown' を返します。
- 論理否定演算子 **not** の結果は、既知のオペランドに対しては、そのオペランドの値が'0' と等しくない場合は'0'、'0' と等しい場合は'1' になります。未知のオペランドに対しては、**not** は'Unknown' を返します。

値キャッシュ

トリガーの評価に必要な値は、Zabbix サーバーによってキャッシュされます。
このため、サーバーの再起動後しばらくの間は、トリガーの評価によってデータベース負荷が高くなります。
アイテムの履歴値が削除されても（手動または housekeeper による場合のいずれでも）、値キャッシュはクリアされません。そのため、トリガー関数で定義された期間より古くなるか、サーバーが再起動されるまで、サーバーはキャッシュされた値を使用します。

> **Note:**
> キャッシュ内に最近のデータがなく、かつ関数内で問い合わせ期間が定義されていない場合、Zabbix はデフォルトで最大 1 週間前までさかのぼって、履歴値をデータベースに問い合わせます。

trigger の例

例 1

Zabbix server のプロセッサの負荷が高すぎます。

```
last(/Zabbix server/system.cpu.load[all,avg1])>5
```

関数'last()' を使用することで、最新の値を参照することができます。
 /Zabbix server/system.cpu.load[all,avg1] には監視するパラメータのショートネームが入ります。
 ホストが Zabbix server で、監視するキーが system.cpu.load[all,avg1] であることを指定します。最後に、>5 は、Zabbix server からの
 最新のプロセッサ負荷測定値が 5 より大きい場合、トリガーが PROBLEM 状態になることを意味します。

例 2

www.example.com は、過負荷になっています。

```
last(/www.example.com/system.cpu.load[all,avg1])>5 or min(/www.example.com/system.cpu.load[all,avg1],10m)>
```

この式は、現在のプロセッサ負荷が 5 以上であるか、過去 10 分間のプロセッサ負荷が 2 以上であった場合に”TRUE” となります。

例 3

/etc/passwd が変更されました。

```
last(/www.example.com/vfs.file.cksum[/etc/passwd],#1)<>last(/www.example.com/vfs.file.cksum[/etc/passwd],#
```

この式は、/etc/passwd のチェックサムの直前の値が最新の値と異なる場合に真となります。

同様の式は、/etc/passwd、/etc/inetd.conf、/kernel などの重要なファイルの変更を監視するのに役立ちます。

例 4

誰かがインターネットから大きなファイルをダウンロードしている。

min 関数を使って:

```
min(/www.example.com/net.if.in[eth0,bytes],5m)>100K
```

eth0 の受信バイト数が過去 5 分以内に 100KB 以上であれば，この式は”TRUE” となります。

例 5

クラスタ化された SMTP サーバの両ノードがダウン

1 つの式に 2 つの異なるホストが使用されていることに注意してください。

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

smtp1.example.com と smtp2.example.com の両方で SMTP サーバがダウンしている場合、この式は成立します。

例 6

Zabbix agent のアップグレードが必要

関数 find() を使用して:

```
find(/example.example.com/agent.version,,"like","beta8")=1
```

この式は、Zabbix agent のバージョンが beta8 である場合に”TRUE” となります。

例 7

サーバにアクセスできない

```
count(/example.example.com/icmpping,30m,,"0")>5
```

この式は、ホスト”example.example.com ” が過去 30 分間に 5 回以上到達不能になった場合に”TRUE” となる。

例 8

過去 3 分以内に heartbeats がない。

関数 nodata() を使用して:

```
nodata(/example.example.com/tick,3m)=1
```

この trigger を使用するには、Zabbix trapper item として’tick’ を定義する必要があります。
 ホストはこの item のデータを zabbix_sender を使用して定期的に送信する必要があります。180 秒以内にデータが受信されないと、
 trigger の値が PROBLEM になります。

注意’nodata’ は任意の item タイプに使用できることに注意してください。

例 9

夜間の CPU アクティビティ。

time() 関数の使用:

```
min(/Zabbix server/system.cpu.load[all,avg1],5m)>2 and time()<060000
```

このトリガーは、夜間（00:00 ～06:00）にのみ状態が障害へ変化する可能性があります。

例 10

例外を除く任意の時点での CPU アクティビティ。

関数 time() と **not** 演算子の使用:

```
min(/zabbix/system.cpu.load[all,avg1],5m)>2
and not (dayofweek()=7 and time()>230000)
and not (dayofweek()=1 and time()<010000)
```

このトリガーは、週の切り替わり時の 2 時間（日曜日 23:00 ～月曜日 01:00）を除き、任意の時点で障害状態に変化する可能性があります。

例 11

クライアントのローカル時刻が Zabbix サーバー時刻と同期しているかを確認します。

関数 fuzzytime() の使用:

```
fuzzytime(/MySQL_DB/system.localtime,10s)=0
```

サーバー MySQL_DB 上のローカル時刻と Zabbix サーバーの時刻との差が 10 秒を超えると、このトリガーは障害状態に変わります。なお、'system.localtime' は Zabbix エージェントではパッシブチェックとして設定する必要があります。Zabbix エージェント 2 では、アクティブチェックとして設定することもできます。

例 12

本日の平均負荷を昨日の同時刻の平均負荷と比較します（タイムシフトとして now-1d を使用）。

```
avg(/server/system.cpu.load,1h)/avg(/server/system.cpu.load,1h:now-1d)>2
```

直近 1 時間の平均負荷が、昨日の同じ 1 時間帯の平均負荷の 2 倍を超えた場合、このトリガーが発生します。

例 13

別のアイテムの値を使用して、トリガーのしきい値を取得します。

```
last(/Template PfSense/hrStorageFree[{#SNMPVALUE}])<last(/Template PfSense/hrStorageSize[{#SNMPVALUE}])*0.
```

空き容量が 10% を下回ると、トリガーが起動します。

例 14

しきい値を超えるトリガーの数を取得するためにevaluation resultを使用します:

```
(last(/server1/system.cpu.load[all,avg1])>5) + (last(/server2/system.cpu.load[all,avg1])>5) + (last(/serve
```

このトリガーは、式内のトリガーのうち少なくとも 2 つが障害状態にある場合に発生します。

例 15

2 つのアイテムの文字列値の比較 - ここでのオペランドは、文字列を返す関数です。

問題: ホストによって Ubuntu のバージョンが異なる場合にアラートを作成する

```
last(/NY Zabbix server/vfs.file.contents[/etc/os-release])<>last(/LA Zabbix server/vfs.file.contents[/etc/
```

例 16

2 つの文字列値を比較します。オペランドは次のとおりです。

- 文字列を返す関数
- マクロと文字列の組み合わせ

問題: DNS クエリの変更を検出する

アイテムキーは次のとおりです。

```
net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1]
```

マクロは次のように定義されています。

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

通常は次を返します。

```
example.com           MX       0 mail.example.com
```

したがって、DNS クエリ結果が期待される結果から逸脱したかどうかを検出するためのトリガー式は次のとおりです。

```
last(/Zabbix server/net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

2 番目のオペランドが引用符で囲まれていることに注意してください。

例 17

2 つの文字列値の比較 - オペランドは次のとおりです。

- 文字列を返す関数
- 特殊文字 \ および" を含む文字列定数

問題: /tmp/hello ファイルの内容が次と等しいかどうかを検出します。

```
\" //hello ?\"
```

オプション 1. 文字列を直接記述する:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

文字列を直接比較する場合、\ および" 文字がエスケープされることに注意してください。

オプション 2. マクロを使用する

```
{$HELLO_MACRO} = \" //hello ?\"
```

これを式内で使用します。

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

例 18

長期間の比較

問題：先月、Exchange server の負荷が 10% 以上増加した。

```
trendavg(/Exchange/system.cpu.load,1M:now/M)>1.1*trendavg(/Exchange/system.cpu.load,1M:now/M-1M)
```

trigger 設定のEvent nameフィールドを使って、例えば次のようなものを受け取るために、意味のある警告メッセージを構築することも可能です。

```
"Load of Exchange server increased by 24% in July (0.69) comparing to June (0.56)"
```

以下のようにイベント名を定義する必要があります:

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

また、このような問題のために、trigger 設定において手動で閉じることができるようにすることも有効です。

ヒステリシス (Hysteresis)

問題状態と回復状態の間に単純な閾値ではなく、インターバルが必要な場合があります。
 例えば、サーバルームの温度が 20°C 以上になったときに問題を報告する trigger を定義し、温度が 15°C 以下に下がるまで
 問題状態を維持したい場合、20°C の単純な trigger 閾値では十分ではありません。

その代わりに、最初に問題が発生した問題事象（温度が 20°C 以上）に対する trigger 式を定義する必要があります。
 次に、回復条件（温度 15°C 以下）を追加で定義する必要があります。
 これは、次のように定義することで行われます。*Recovery expression* パラメータを定義し、defining trigger
 を定義します。

この場合、問題の回復は 2 つのステップで行われます。

- まず、問題式 (20°C 以上の温度) は、"FALSE" に評価されなければなりません。
- 次に、リカバリー式 (温度 15°C 以下) を"TRUE" に評価します。

回復式は、問題事象が先に解決された場合にのみ評価されます。

> **Warning:**
> リカバリー式が"TRUE" であることだけでは、問題式がまだ"TRUE" である場合に問題を解決することができません！

例 1

サーバールームの温度が高すぎる

問題式:

```
last(/server/temp)>20
```

回復式:

```
last(/server/temp)<=15
```

例 2

空きディスク容量が少なすぎる

問題式: 直近 5 分間の空きディスク容量が 10GB 以下

```
max(/server/vfs.fs.size[/,free],5m)<10G
```

回復式: 問題式: 直近 10 分間の空きディスク容量が 40GB 以上

```
min(/server/vfs.fs.size[/,free],10m)>40G
```

未知のオペランドを持つ式

通常、未知のオペランド (サポートされていないアイテムなど) は、トリガー値をすぐに' 不明' にレンダリングします。

ただし、未知のオペランド (サポートされていないアイテム、関数エラー等) が式の評価に認められる場合があります。

- nodata() 関数は、参照された項目がサポートされているかどうかに関係なく評価されます。
- OR および AND を使用した論理式は、未知のオペランドに関係なく、次の 2 つのケースで既知の値に評価できます。
  - ケース **1**: "1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ..." は既知の結果 ('1' または" 障害") に評価できます。
  - ケース **2**: "0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ..." は、既知の結果 ('0' または"OK") に評価できます。
    Zabbix は、サポートされていないアイテムを未知のオペランドとして取得することで、論理式を評価しようとします。上記の 2 つのケースでは、既知の値が生成されます (それぞれ" 障害" または"OK") が、その他すべてのケースでは、トリガーは' 不明' と評価されます。
- サポートされているアイテムの関数評価でエラーが発生した場合、関数の値は不明になり、以降の式の評価で不明なオペランドとして使用されます。

上記のように、未知のオペランドは論理式でのみ「消える」場合があることに注意してください。算術式では、未知のオペランドの結果は常に 不明になります (0 による除算を除く)。

Attention:
不明になる式は、トリガーの状態 (" 障害/OK") を変更しません。したがって、状態が" 障害" (ケース 1 を参照) であった場合、既知の部分が解決されても ('1' が'0' になる)、トリガー状態を変更せず同じ障害状態のままになります。

サポートされていないアイテムがいくつかあるトリガー式が不明と評価される場合、フロントエンドのエラーメッセージは、最後に評価されたサポートされていないアイテムを参照します。

3 トリガーの依存関係

概要

ホストの可用性が別のホストに依存する場合があります。ルーターの背後にあるサーバーは、ルーターがダウンすると到達不能になります。両方に対してトリガーが設定されていると、実際に問題の原因がルーターだけであっても、2 台のホストがダウンしたという通知を受け取る可能性があります。

このような場合、ホスト間の依存関係が役立ちます。依存関係を設定すると、依存先の問題に起因する通知を抑止し、根本原因の問題に対する通知のみを送信できます。

Zabbix はホスト間の依存関係を直接サポートしていませんが、より柔軟な別の方法であるトリガーの依存関係によって定義できます。1 つのトリガーは、依存先として 1 つ以上のトリガーを持つことができます。

この単純な例では、サーバーのトリガー設定フォームを開き、対応するルーターのトリガーに依存するよう設定します。このような依存関係がある場合、依存先のトリガーが「PROBLEM」状態である限り、サーバーのトリガーは状態を変更しません。そのため、依存するアクションは実行されず、通知も送信されません。

サーバーとルーターの両方がダウンしており、依存関係が設定されている場合、Zabbix は依存するトリガーに対してアクションを実行しません。

親トリガーが PROBLEM 状態にある間、その依存先は信頼できない値を報告する可能性があります。そのため、依存するトリガーは、親トリガー（上記の例ではルーター）が次のいずれかになるまで再評価されません。

- 「PROBLEM」状態から「OK」状態に戻る
- 状態が「PROBLEM」から「UNKNOWN」に変わる
- 手動、相関関係、またはdate and time および/またはnodata() 関数を使用してクローズされる
- 依存するトリガーに関与していないアイテムの値によって解決される
- 無効化される、無効なアイテムを持つ、または無効なアイテムを持つホストになる

上記のいずれの場合でも、依存するトリガー（サーバー）は、そのトリガーに対する新しいメトリクスを受信したときにのみ再評価されます。これは、依存するトリガーがすぐには更新されない可能性があることを意味します。

また:

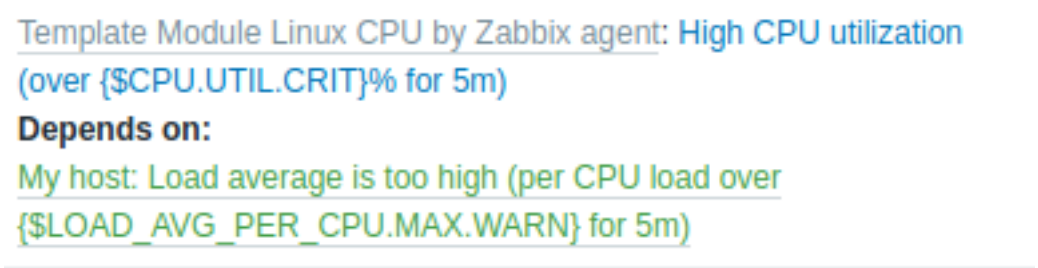
- トリガーの依存関係は、循環依存にならない限り、任意のホストのトリガーから別の任意のホストのトリガーへ追加できます。
- トリガーの依存関係は、あるテンプレートから別のテンプレートへ追加できます。テンプレート A のあるトリガーがテンプレート B のあるトリガーに依存している場合、テンプレート A はテンプレート B と一緒にのみホスト（または別のテンプレート）へリンクできます。一方、テンプレート B は単独でホスト（または別のテンプレート）へリンクできます。
- トリガーの依存関係は、テンプレートのトリガーからホストのトリガーへ追加できます。この場合、そのようなテンプレートをホストにリンクすると、同じ依存先トリガーテンプレートに依存するホストのトリガーが作成されます。これにより、たとえば、いくつかのトリガーがルーター（ホスト）のトリガーに依存するテンプレートを作成できます。このテンプレートにリンクされたすべてのホストは、その特定のルーターに依存することになります。
- トリガーの依存関係は、ホストのトリガーからテンプレートのトリガーへは追加できません。
- トリガーの依存関係は、あるトリガープロトタイプから別のトリガープロトタイプ（同じローレベルディスカバリルール内）または実際のトリガーへ追加できます。トリガープロトタイプは、別の LLD ルールのトリガープロトタイプや、トリガープロトタイプから作成されたトリガーに依存することはできません。ホストのトリガープロトタイプは、テンプレートのトリガーに依存できません。

設定

依存関係を定義するには、トリガーの設定フォームで「Dependencies」タブを開きます。「Dependencies」ブロックの *Add* をクリックし、そのトリガーが依存する 1 つ以上のトリガーを選択します。



Update をクリックします。これで、一覧内のトリガーに依存関係の表示が追加されます。



複数の依存関係の例

たとえば、ホストが Router2 の背後にあり、Router2 が Router1 の背後にあるとします。

```
Zabbix - Router1 - Router2 - Host
```

Router1 がダウンすると、当然ながらホストと Router2 にも到達できなくなりますが、ホスト、Router1、Router2 のすべてがダウンしているという通知を 3 件受け取るのは過剰です。

そのため、この場合は次の 2 つの依存関係を定義します。

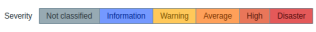
```
「Host is down」トリガーは「Router2 is down」トリガーに依存する
「Router2 is down」トリガーは「Router1 is down」トリガーに依存する
```

「Host is down」トリガーのステータスを変更する前に、Zabbix は対応するトリガーの依存関係を確認します。そのような依存関係が見つかり、それらのトリガーのいずれかが「Problem」状態にある場合、トリガーのステータスは変更されず、アクションは実行されず、通知も送信されません。

Zabbix はこの確認を再帰的に実行します。Router1 または Router2 に到達できない場合、ホストのトリガーは更新されません。

4 トリガーの深刻度

トリガーの深刻度は、トリガーの重要度レベルを表します。



Zabbix は、次のデフォルトのトリガー深刻度をサポートしています。

深刻度	色	説明
未分類 (Not classified)	灰色	イベントの深刻度レベルが不明、決定されていない、通常の監視範囲に含まれていない場合などに使用できます。たとえば、初期構成中、将来の評価のためのプレースホルダーとして、または統合プロセスの一部として使用できます。
情報 (Information)	水色	すぐに対処する必要はないが、貴重な洞察を提供できる情報イベントに使用できます。
警告 (Warning)	黄色	調査や対応が必要になる可能性があるが、重大ではない潜在的な問題を示すために使用できます。
軽度の障害 (Average)	オレンジ	さらなる問題を防ぐために比較的早く対処する必要がある重大な問題を示すために使用できます。
重度の障害 (High)	淡い赤	重大な中断を避けるためにすぐに対処する必要がある重大な問題を示すために使用できます。
致命的な障害 (Disaster)	赤	システム停止やデータ損失などを防ぐためにすぐに対処する必要がある重大なインシデントを示すために使用できます。

Note:
トリガーの深刻度名と色はカスタマイズできます。

深刻度は、以下の用途で使用されます:

- トリガーの視覚的表現 - 深刻度に応じて異なる色
- グローバルアラームの音声 - 深刻度に応じて異なる音声
- ユーザーメディア - 深刻度に応じて異なるメディア (通知チャネル) (たとえば、深刻度が重度の障害および致命的な障害のトリガーの場合は SMS、その他の深刻度のトリガーの場合は電子メール)
- トリガーの深刻度に対する条件によるアクションの制限

5 トリガーの深刻度のカスタマイズ

GUI に関連づけられているトリガーの深刻度の名称および色については、管理 → 一般 → トリガー表示オプションで設定することができます。色はすべての GUI テーマで共有されます。

カスタマイズされた深刻度名の翻訳

Attention:
Zabbix フロントエンドの翻訳を使用している場合、カスタマイズした深刻度名は、デフォルトで翻訳された名前より優先されます。

デフォルトのトリガーの深刻度名は、すべてのロケールにおいて翻訳可能です。深刻度名を変更した場合、すべてのロケールにおいてカスタム名が使用されるため、手動での追加翻訳が必要です。

カスタム深刻度名の翻訳手順

- 必要なカスタム深刻度名、例えば'Important' を設定します。
- <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po を編集します。
- 次の 2 行を追加します。

```
msgid "Important"
msgstr "<翻訳する文>"
```

そしてファイルを保存してください。

- <frontend_dir>/locale/README で説明されているように.mo ファイルを作成します。

ここで **msgid** は新しいカスタム深刻度名と一致し、**msgstr** は特定の言語での翻訳でなければなりません。

この手順は深刻度名を変更するたびに実行する必要があります。

6 一括アップデート

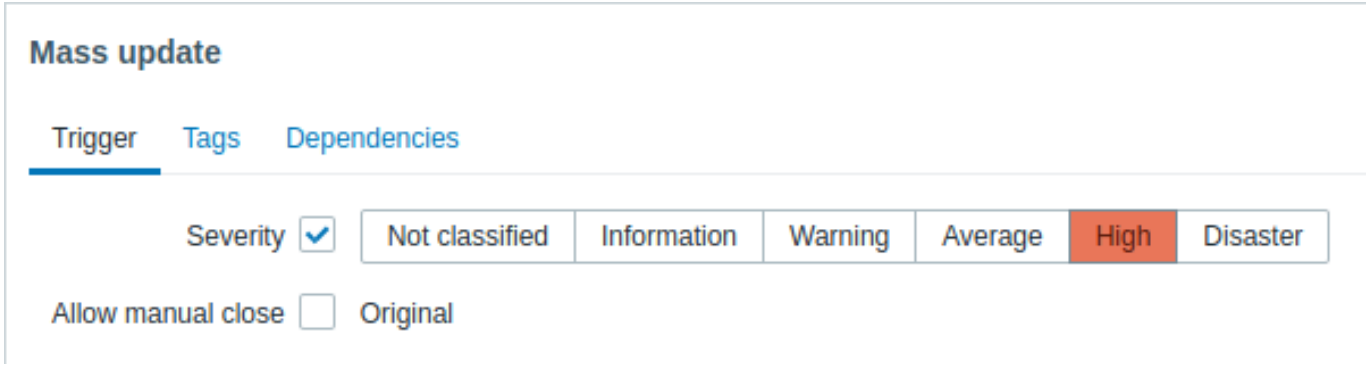
概要

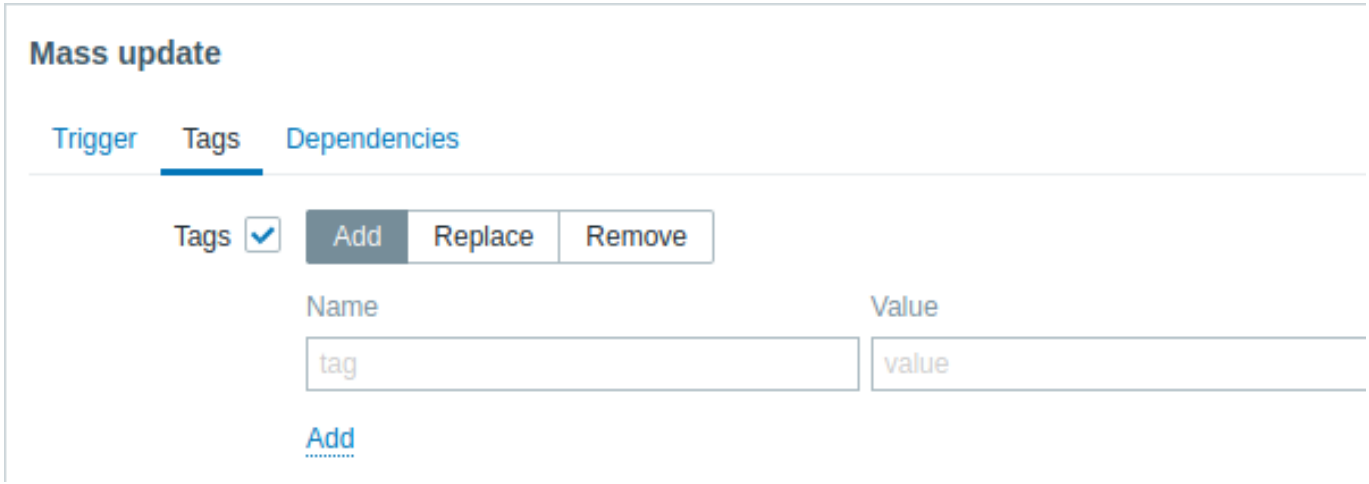
一括アップデートでは、一度に多くの trigger の属性を変更することができ、編集のために個々の trigger を開く手間を省くことができます。

一括アップデートの利用

いくつかの trigger を一括でアップデートするには、以下のようにします。

- リスト内の更新したい trigger のチェックボックスにマークを付けます。
- リストの下にある *Mass update* をクリックします。
- 必要な属性（*Trigger*、*Tags*、*Dependencies*）のあるタブに移動します。
- 更新する属性のチェックボックスにマークを付けます。

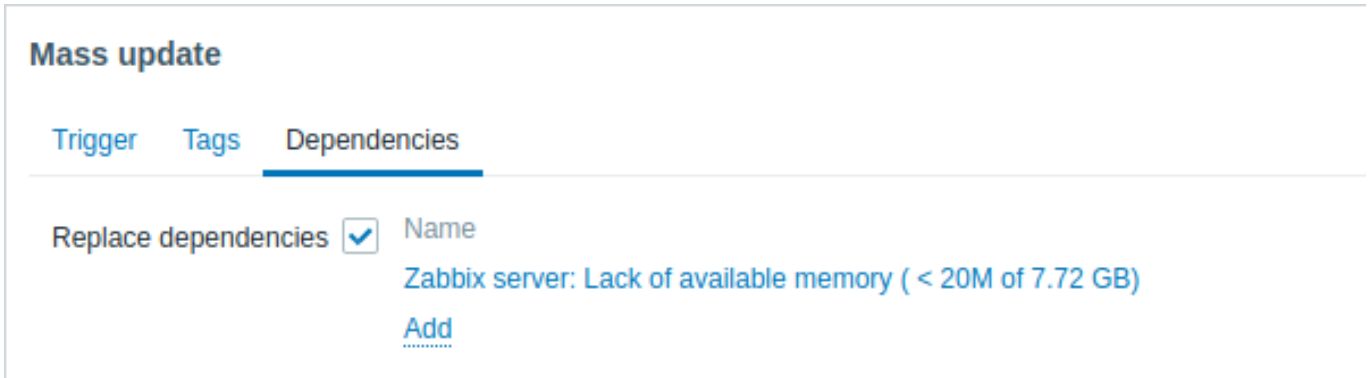




タグ更新の各ボタンを選択すると、以下のオプションが利用できます。

- *Add* - トリガーに新しいタグを追加することができます。
- *Replace* - トリガーから既存のタグを削除し、以下で指定するタグに置き換えます。
- *Remove* - 指定したタグをトリガーから削除します。

同じ名前で異なる値を持つタグは、' 重複' とはみなされず、同じトリガーに追加することができることに注意してください。



Replace dependencies - トリガーから既存の依存関係を削除し、指定されたものに置き換えます。

変更を適用するには、*Update* をクリックしてください。

7 予測トリガー関数

概要

時には、問題が発生する予兆があることがあります。これらの兆候を発見することで、事前に対策を講じ、問題の影響を防ぐ、
または少なくとも最小限に抑えることができます。

Zabbix は history データに基づいて監視システムの将来の挙動を予測するツールを備えています。
 これらのツールは予兆トリガー機能で実現されます。

関数

トリガーを設定する前に、問題状態とは何か、また対応を取るためにどれだけの時間が必要かを定義する必要があります。次に、望ましくない状況が発生する可能性を通知するトリガーを設定する方法は 2 つあります。1 つ目は、「対応までの時間」が経過した時点でシステムが問題状態にあると予想される場合に、トリガーが発生するようにする方法です。2 つ目は、システムが「対応までの時間」未満で問題状態に到達すると予想される場合に、トリガーが発生するようにする方法です。対応するトリガー関数は **forecast** と **timeleft** です。なお、両方の関数で使用される基礎となる統計分析は基本的に同一です。ほぼ同様の結果が得られるため、好みに応じてどちらの方法でもトリガーを設定できます。

パラメータ

両方の関数は、ほぼ同じパラメータセットを使用します。参照用として、サポートされている関数 の一覧を使用してください。

時間間隔

まず最初に、予測を行うために Zabbix が分析すべき履歴期間を指定する必要があります。これは、`time period` パラメータと任意のタイムシフトを使用して、**avg**、**count**、**delta**、**max**、**min**、**sum** 関数と同様の使い慣れた方法で行います。

予測期間

(**forecast** のみ)
パラメータ `time` は、履歴データ内で見つかった依存関係を Zabbix が将来のどこまで外挿するかを指定します。`time_shift` を使用するかどうかにかかわらず、`time` は常に現在の時点から数えられます。

到達すべきしきい値

(**timeleft** のみ)
パラメータ `threshold` は、解析対象のアイテムが到達しなければならない値を指定します。
上側から到達するか下側から到達するかは問いません。f(t) （以下を参照）が求められたら、方程式 f(t) = `threshold` を解き、現在に最も近く、かつ現在より右側にある根を返す必要があります。そのような根が存在しない場合は、1.7976931348623158E+308 を返します。

Note:
アイテムの値がしきい値に近づいた後にそれを横切る場合、**timeleft** はその交点がすでに過去にあるとみなし、存在する場合は `threshold` レベルとの次の交点に切り替えます。ベストプラクティスとしては、予測は通常の障害診断の代替ではなく、補完として使用することが推奨されます。[1]

フィット関数

デフォルトの `fit` は *linear* 関数です。ただし、監視対象のシステムがより複雑な場合は、選択できるオプションがさらにあります。

`fit`	x = f(t)
linear	$x = a + b*t$
polynomialN[2]	$x = a_0 + a_1*t + a_2*t^2 + ... + a_n*t^n$
exponential	$x = a*\exp(b*t)$
logarithmic	$x = a + b*\log(t)$
power	$x = a*t^b$

モード

(**forecast** のみ)
トリガー関数が評価されるたびに、指定された履歴期間のデータを取得し、そのデータに指定された関数を当てはめます。したがって、データがわずかに異なれば、当てはめられる関数もわずかに異なります。単純に、将来の指定時点における当てはめ関数の値を計算するだけでは、現在からその将来時点までの間に、分析対象のアイテムがどのように振る舞うと予想されるかは分かりません。`fit` の一部のオプション（*polynomial* など）では、将来の単純な値は誤解を招く可能性があります。

`mode`	**forecast** の結果
value	f(now + `time`)
max	$\max_{now <= t <= now + time} f(t)$
min	$\min_{now <= t <= now + time} f(t)$

mode	**forecast** の結果
delta	*max - min*
avg	定義に従った f(t) の平均値（now <= t <= now + time）

詳細

非常に大きな数での計算を避けるため、指定期間内の最初の値のタイムスタンプに 1 ns を加えたものを、新しいゼロ時刻として扱います（現在のエポック時刻は 10^9 のオーダー、エポック時刻の二乗は 10^{18}、倍精度は約 10^{-16} です）。1 ns を加えるのは、log(t) の計算を含む 対数近似および べき乗近似において、すべての時刻の値を正にするためです。時刻のシフトは、線形、多項式、指数近似には影響しません（計算がより容易かつ高精度になる点を除く）が、対数関数および べき乗関数の形状は変化します。

起こり得るエラー

このような場合、関数は -1 を返します。

- 指定された評価期間にデータが存在しない場合。
- 数学演算の結果が定義されていない場合[3]。
- 数値計算上の問題が発生した場合（残念ながら、入力データの一部の組み合わせでは、倍精度浮動小数点形式の範囲と精度が不十分になることがあります）[4]。

Note:
選択した近似が提供されたデータを十分に表していない場合や、正確な予測を行うにはデータが少なすぎる場合でも、警告やエラーは示されません。

例とエラーへの対処

ホストの空きディスク容量が不足しそうなときに警告を受け取るには、次のようなトリガー式を使用できます:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

ただし、エラーコード -1 が影響し、トリガーが障害状態になる場合があります。一般的には、これは予測が正しく機能しておらず、その原因を調べるためにより詳しく確認すべきであるという警告を受け取れるため、望ましい動作です。しかし、場合によっては望ましくありません。というのも、-1 は単に直近 1 時間にホストの空きディスク容量に関するデータが取得されなかったことを意味するだけの場合があるためです。誤検知のアラートが多すぎる場合は、より複雑なトリガー式の使用を検討してください[5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

forecastの場合は、状況が少し複雑になります。まず、`forecast(/host/item,(...))<...` のような式か、`forecast(/host/item,(...))` のような式かによって、-1 がトリガーを障害状態にする場合としない場合があります。

さらに、アイテムの値が負になることが通常である場合、-1 は有効な予測値である可能性があります。しかし、実際の環境でそのような状況が発生する可能性はごくわずかです（演算子 = の動作についてはこちらを参照してください）。したがって、-1 を障害として扱いたい場合は `... or forecast(/host/item,(...))=-1` を、障害として扱いたくない場合は `... and forecast(/host/item,(...))<>-1` を追加してください。

脚注

[1] たとえば、単純なトリガー `timeleft(/host/item,1h,X) < 1h` は、アイテムの値が X に近づいたときに障害状態になり、その後 X に達すると突然復旧する場合があります。問題がアイテムの値が X 未満であることであれば、次を使用してください: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` 問題がアイテムの値が X を超えることであれば、次を使用してください: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] 多項式の次数は 1 から 6 まで指定でき、*polynomial1* は *linear* と同等です。ただし、高次の多項式は注意して使用してください。評価期間に多項式係数を決定するのに必要な点数より少ないデータ点しか含まれていない場合、多項式の次数は引き下げられます（たとえば *polynomial5* が要求されても、点が 4 つしかない場合は *polynomial3* が当てはめられます）。

[3] たとえば、*exponential* または *power* 関数の当てはめでは、アイテム値の log() を計算します。データに 0 または負の数が含まれている場合、log() は正の値に対してのみ定義されているため、エラーになります。

[4] *linear*、*exponential*、*logarithmic*、*power* の当てはめでは、必要な計算はすべて明示的に記述できます。*polynomial* では、追加の手順なしで計算できるのは *value* のみです。*avg* の計算では、多項式の原始関数を（解析的に）計算する必要があります。*max*、*min*、*delta* の計算では、多項式の導関数を（解析的に）計算し、その根を（数値的に）求める必要があります。f(t) = 0 を解くには、多項式の根を（数値的に）求める必要があります。

[5] ただし、この場合は -1 によってトリガーが障害状態から復旧してしまう可能性があります。完全に防ぐには、次を使用してください: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1 or {TRIGGER.VALUE}=1)`

4 イベント

概要

Zabbix で生成されるイベントにはいくつかの種類があります。

- トリガーイベント - トリガーのステータスが変わるたびに (*OK*→ 障害 →*OK*)
- サービスイベント - サービスのステータスが変わるたびに (*OK*→ 障害 →*OK*)
- ディスカバリイベント - ホストまたはサービスが検出されたとき
- 自動登録イベント - アクティブなエージェントがサーバーによって自動登録されたとき
- 内部イベント - アイテム/ローレベルディスカバリルールがサポートされなくなったとき、またはトリガーが不明な状態になったとき

イベントにはタイムスタンプが付けられ、通知メールの送信などのアクションの元となることができます。

フロントエンドでイベントの詳細を表示するには、監視 → 障害へ移動してください。そこでイベントの日付と時刻をクリックすると、イベントの詳細が表示されます。

詳細については、以下を参照してください。

- トリガーイベント
- その他のイベントソース

1 トリガーイベントの生成

概要

トリガーステータスの変化は、最も頻繁かつ重要なイベントの発生源です。トリガーステータスが変化するたびに、イベントが生成されます。イベントには、トリガーステータスの変化の詳細 (いつ発生したか、新しいステータスは何か) が含まれます。

トリガーによって、障害と正常の 2 種類のイベントが作成されます。

障害イベント

次のとき、障害イベントが作成されます。

- トリガーが正常ステータスの場合、トリガー式が TRUE と評価されたときに。
- トリガーに対して複数の障害イベントの生成が有効になっている場合、トリガー式が TRUE と評価されるたびに。

正常イベント

正常イベントは関連する障害イベントを閉じます。次の 3 つのコンポーネントによって作成される場合があります。

- トリガー - ’ 正常イベントの生成’ および’ 正常イベントの終了’ 設定に基づく
- イベント相関関係
- タスクマネージャー – イベントが手動でクローズされたとき

トリガー

トリガーには、OK イベントの生成方法を制御する’OK イベント生成’ 設定があります。

- *Expression* - 問題のある状態のトリガーに対して、その式が FALSE と評価されると OK イベントが生成されます。これは、最も単純な設定で、デフォルトで有効になっています。
- *Recovery expression* - OK イベントは、その式が FALSE と評価され、回復式が TRUE と評価されたとき、問題状態のトリガーに対して生成されます。これは、トリガーの回復基準が問題基準と異なる場合に使用できます。
- *None* - OK イベントは、決して生成されません。これは、複数の問題イベント生成と組み合わせて、何かが起こったときに単に通知を送るために使用することができます。

さらに、トリガーには’OK イベントクローズ’ の設定があり、どの問題イベントがクローズされるかを制御します。

- *All problems* - OK イベントは、トリガーによって作成されたすべての未解決問題を閉じます。
- *All problems if tag values match* - OK イベントは、トリガーによって作成され、少なくとも 1 つの一致するタグ値を持つ未解決の問題を閉じます。タグは、’Tag for matching’ トリガ設定によって定義されます。閉じるべき問題イベントがない場合、OK イベントは生成されません。これは、トリガレベルイベント相関と呼ばれます。

イベント相関

イベント相関（グローバルイベント相関とも呼ばれる）は、カスタムイベント終了（結果的に OK イベントを生成する）ルールを
 設定する方法です。

このルールでは、新しい問題イベントと既存の問題イベントの組み合わせ方法を定義し、対応する OK イベントを生成して、
 新しいイベントまたは一致したイベントを閉じることができます。

しかし、イベント相関は、イベント処理のパフォーマンスに悪影響を及ぼしたり、設定を誤ると意図したよりも多くのイベントを
 クローズしてしまう（最悪の場合、すべての問題イベントをクローズしてしまう）ため、非常に慎重に設定する必要があります。
 設定のヒントをいくつか紹介します。

1. コントロールイベント（古いイベントと対になるイベント）に一意のタグを設定し、' 新しいイベントタグ' 相関条件を使用することで、常に相関範囲を縮小します。
2. 'close old event' オペレーションを使用する際に、古いイベントに基づいた条件を追加することを忘れないでください。
3. 異なる相関設定によって使用される共通のタグ名を使用しないようにします。

タスクマネージャー

トリガーに対して' 手動クローズを許可' 設定が有効になっている場合、トリガーによって生成された障害イベントを手動で閉じることができます。これは、障害を更新する ときに Web インターフェースで行われます。イベントは直接閉じられません。代わりに'close event' タスクが作成され、タスクマネージャーによってすぐに処理されます。タスクマネージャーは対応する正常イベントを生成し、障害イベントは閉じられます。

2 その他のイベントソース

サービスイベント

サービスイベントは、これらのイベントに対するサービスアクションが有効な場合にのみ生成されます。
 この場合、サービスステータスの変更ごとに新しいイベントを生成します。

- 問題イベント - サービスステータスが OK から PROBLEM に変更されたとき
- OK イベント - サービスステータスが PROBLEM から OK に変更されたとき

このイベントには、サービス状態の変化の詳細（いつ起こったか、新しい状態は何か）が含まれています。

ディスカバリイベント

Zabbix は、ネットワークディスカバリルールで定義された IP 範囲を定期的にスキャンします。チェックの頻度は、ルールごとに個別に設定できます。ホストまたはサービスが検出されると、ディスカバリイベント (または複数のイベント) が生成されます。

Zabbix は次のイベントを生成します。

イベント	発生条件
Service Up	Zabbix がアクティブなサービスを検出するたびに
Service Down	Zabbix がサービスを検出できないたびに
Host Up	少なくとも 1 つのサービスが IP に対して稼働している場合
Host Down	すべてのサービスが応答していない場合
Service Discovered	サービスがダウンタイム後に戻った場合、または初めて検出された場合
Service Lost	起動後にサービスが失われた場合
Host Discovered	ホストがダウンタイム後に復帰した場合、または初めて検出された場合
Host Lost	起動後にホストが失われた場合

アクティブエージェントの自動登録イベント

アクティブエージェントの自動登録により、Zabbix でイベントが作成されます。

設定されている場合、アクティブエージェントの自動登録イベントは、これまで認識されていないアクティブエージェントがチェックを要求したとき、またはホストメタデータが変更されたときに作成されます。サーバーは、受信したエージェントの IP アドレスとポートを使用して、新しい自動登録済みホストを追加します。

詳細については、アクティブエージェントの自動登録ページを参照してください。

内部イベント

内部イベントは、次の場合に発生します。

- アイテムの状態が「normal」から「unsupported」に変わった場合
- アイテムの状態が「unsupported」から「normal」に変わった場合
- ローレベルディスカバリルールの状態が「normal」から「unsupported」に変わった場合
- ローレベルディスカバリルールの状態が「unsupported」から「normal」に変わった場合
- トリガーの状態が「normal」から「unknown」に変わった場合
- トリガーの状態が「unknown」から「normal」に変わった場合

内部イベントを導入する目的は、たとえばアイテムが unsupported になってデータ収集を停止した場合など、何らかの内部イベントが発生した際にユーザーへ通知できるようにすることです。

内部イベントは、これらのイベントに対する内部アクションが有効になっている場合にのみ作成されます。内部イベントの生成を停止するには（たとえば、アイテムが unsupported になる場合）、Alerts → Actions → Internal actions で内部イベントに対するすべてのアクションを無効にしてください。

> **Note:**
> 内部アクションが無効になっている場合、オブジェクトが「unsupported」状態にある間でも、そのオブジェクトのリカバリーイベントは引き続き作成されます。
>
> 内部アクションが有効になっている場合、オブジェクトが「unsupported」状態にある間、そのオブジェクトに対して「problem event」が作成されていなくても、そのオブジェクトのリカバリーイベントは作成されます。

参照: unsupported なアイテムの通知を受け取る

3 問題の手動クローズ

概要

通常、問題イベントはトリガーのステータスが「Problem」から「OK」に変わると自動的に解決されますが、トリガー式によって問題が解決されたかどうかを判断するのが難しい場合があります。そのような場合は、問題を手動で解決する必要があります。

例えば、*syslog* が最適なパフォーマンスのために一部のカーネルパラメータを調整する必要があると報告することがあります。この場合、問題は Linux 管理者に報告され、管理者が修正した後で問題を手動でクローズします。

問題を手動でクローズできるのは、*Allow manual close* オプションが有効になっているトリガーのみです。

問題が「手動でクローズ」されると、Zabbix は Zabbix サーバー向けに新しい内部タスクを生成します。次に、*task manager* プロセスがこのタスクを実行して OK イベントを生成し、その結果、問題イベントがクローズされます。

手動でクローズされた問題であっても、基になるトリガーが今後二度と「Problem」状態にならないことを意味するわけではありません。トリガー式は再評価され、その結果として再び問題になる可能性があります。

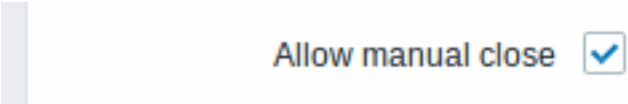
- トリガー式に含まれるいずれかのアイテムに新しいデータが到着したとき（スロットリング前処理ステップで破棄された値は受信されたものとは見なされず、トリガー式の再評価は行われないことに注意してください）。
- 式でdate and time および/またはnodata() 関数が使用されているとき。

設定

手動で問題を閉じるには、2 つのステップが必要です。

トリガー設定

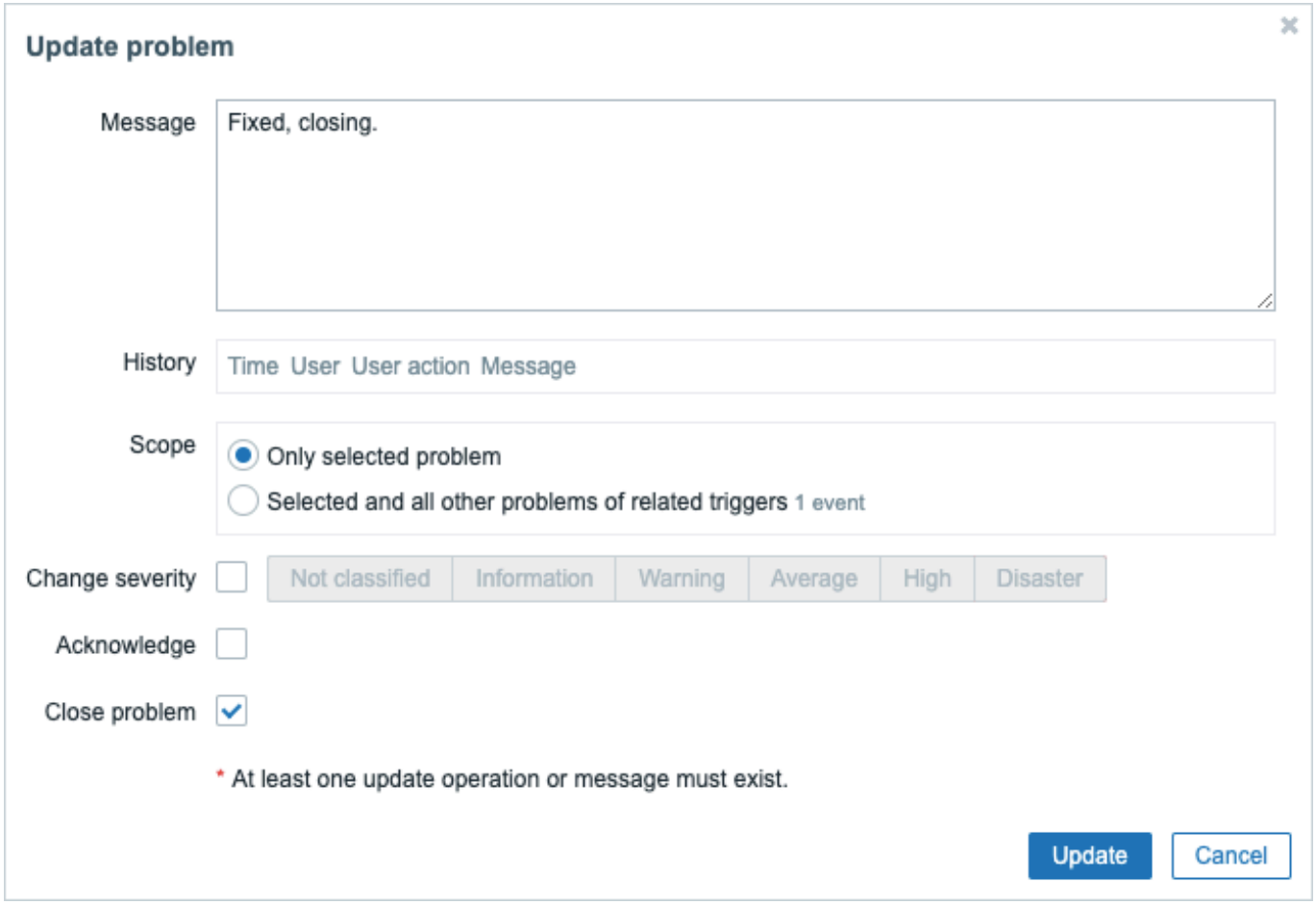
トリガー設定で、手動クローズを許可オプションを有効にします。



障害更新ウィンドウ

手動クローズフラグが設定されたトリガーで障害が発生した場合、その障害の障害更新ポップアップウィンドウを開き、障害を手動でクローズできます。

障害をクローズするには、フォームで *Close problem* オプションをチェックし、*Update* をクリックします。



必須入力フィールドには、赤いアスタリスクが付いています。

このリクエストは Zabbix サーバーによって処理されます。通常、障害がクローズされるまでに数秒かかります。その処理中は、*Monitoring → Problems* に障害のステータスとして *CLOSING* が表示されます。

確認

問題が手動で解決されたことを確認することができます。:

- 監視データ → 障害からイベントの詳細で確認できます。
- 通知メッセージの {EVENT.UPDATE.HISTORY} マクロを使用して、この情報を提供します。

5 イベント相関

概要

イベント相関により、非常に正確かつ柔軟な方法で、障害のイベントをその解決策に関連付けることができます。

イベントの相関関係は次のように定義できます。

- トリガーレベル I - 1 つのトリガーを使用して、個別の障害をその解決策に関連付けることができます
- グローバル - グローバル相関ルールを使用して、別のトリガー/ポーリングメソッドから障害をその解決策に関連付けることができます

1 トリガーによるイベント相関

概要

トリガーベースのイベント相関では、1 つのトリガーによって報告された個別の障害を相関付けることができます。

通常、OK イベントは 1 つのトリガーによって作成されたすべての障害イベントをクローズできますが、より詳細なアプローチが必要になる場合があります。たとえば、ログファイルを監視する際に、ログファイル内の特定の障害を検出し、それらをまとめてではなく個別にクローズしたいことがあります。

これは、*PROBLEM event generation mode* パラメータが *Multiple* に設定されているトリガーの場合に該当します。このようなトリガーは通常、ログ監視、トラップ処理などで使用されます。

Zabbix では、タグ付け に基づいて障害イベントを関連付けることができます。タグは値を抽出し、障害イベントの識別情報を作成するために使用されます。これを活用することで、一致するタグに基づいて障害を個別にクローズすることもできます。

言い換えると、同じトリガーでも、イベントタグによって識別される個別のイベントを作成できます。したがって、障害イベントは 1 件ずつ識別でき、イベントタグによる識別に基づいて個別にクローズできます。

仕組み

ログ監視では、次のような行に遭遇することがあります。

```
Line1: Service 1 stopped
Line2: Service 2 stopped
Line3: Service 1 was restarted
Line4: Service 2 was restarted
```

イベント相関の考え方は、Line1 の障害イベントを Line3 の復旧に対応付け、Line2 の障害イベントを Line4 の復旧に対応付けて、これらの障害を 1 つずつクローズできるようにすることです。

```
Line1: Service 1 stopped
Line3: Service 1 was restarted #Line 1 の障害をクローズ

Line2: Service 2 stopped
Line4: Service 2 was restarted #Line 2 の障害をクローズ
```

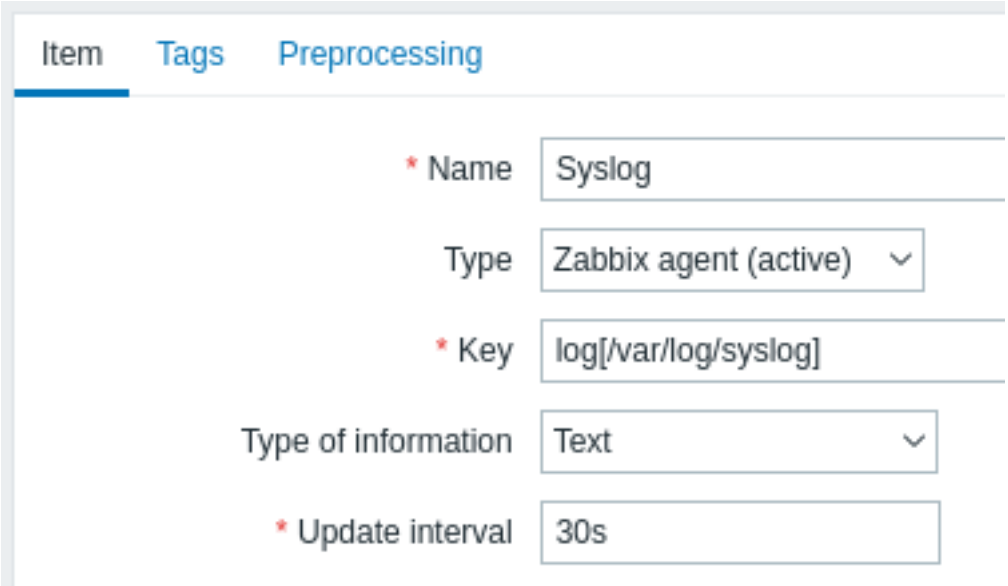
これを行うには、これらの関連するイベントに、たとえば”Service 1” および”Service 2” というタグを付ける必要があります。これは、ログ行に正規表現を適用してタグ値を抽出することで実現できます。すると、イベントの作成時にそれぞれ”Service 1” および”Service 2” のタグが付けられ、障害を復旧に対応付けることができます。

設定

item

始めに、例えばログファイルを監視するような項目を設定するのがよいでしょう:

```
log[/var/log/syslog]
```



item が設定された状態で、設定変更が反映されるまで 1 分ほど待ってから、Latest dataにアクセスして、
 item がデータ収集を開始したことを確認します。

トリガー

アイテムが動作したら、トリガーを設定する必要があります。重要なのは、ログファイル内のどのエントリに注意を払うべきかを判断することです。例えば、次のトリガー式は、潜在的な問題を示すために 'Stopping' のような文字列を検索します。

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
文字列 "Stopping" を含む各行が問題として扱われるようにするには、トリガー設定の *Problem event generation mode* も 'Multiple' に設定してください。

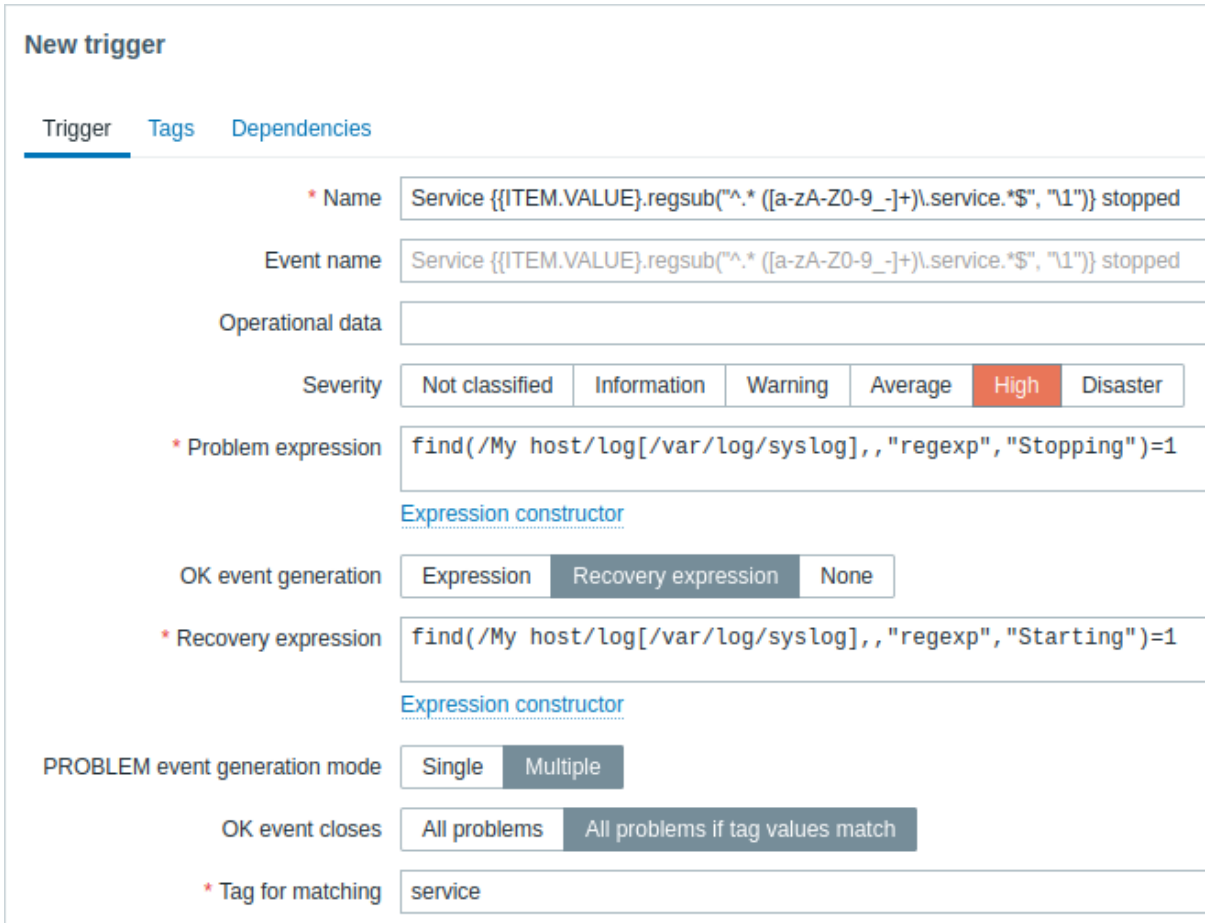
次に、復旧条件式を定義します。次の復旧条件式は、文字列 "Starting" を含むログ行が見つかった場合に、すべての問題を復旧します。

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

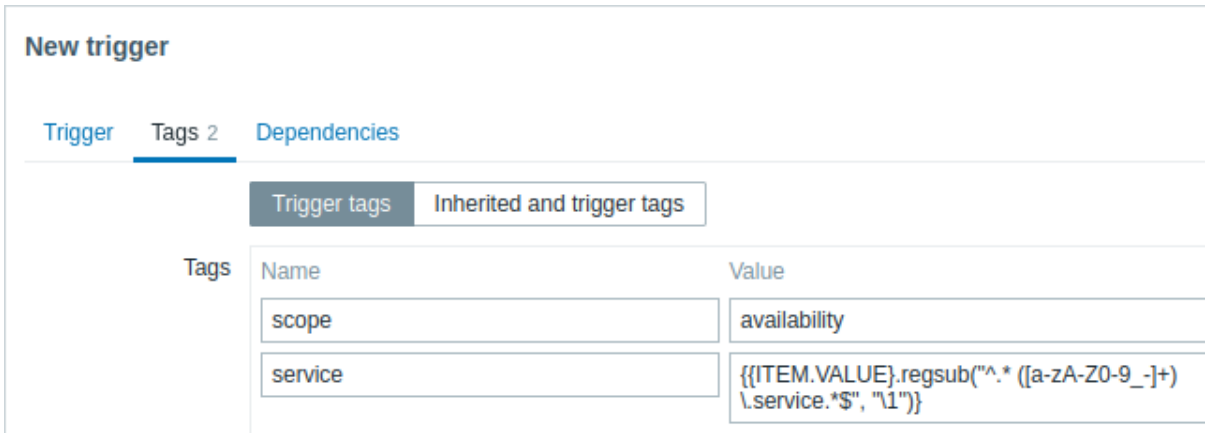
しかし、これは望ましくないため、対応する根本の問題だけがクローズされ、単にすべての問題がクローズされるわけではないことを、何らかの方法で確実にすることが重要です。ここでタグ付けが役立ちます。

問題と復旧は、トリガー設定でタグを指定することで対応付けできます。次の設定を行う必要があります。

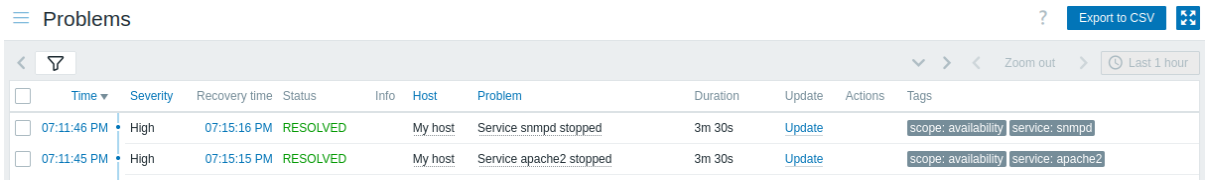
- *Problem event generation mode*: Multiple
- *OK event closes*: All problems if tag values match
- イベントの照合に使用するタグ名を入力する



- ログ行からタグ値を抽出するようにタグを設定する



正しく設定されると、*Monitoring* → *Problems* で、アプリケーションごとにタグ付けされ、対応する復旧と照合された障害イベントを確認できるようになります。



Warning:
設定ミスにより、無関係な問題に対して類似したイベントタグが作成される可能性があるため、以下で説明するケースを確認してください。

- インデックス付きマクロは常にトリガー設定の *Expression* フィールドを参照し、*Recovery expression* は参照しません。
例えば、復旧イベントでは、**{ITEM.VALUE1}** は復旧時点における障害条件式内の最初のアイテムの最新値に展開されます。
復旧条件式が別のアイテムに基づいており、復旧時までに障害条件式のアイテム値が変化している場合、イベントには異なるタグが付与され、相関付けされません。

- 2 つのアプリケーションが同じログファイルにエラーおよび復旧メッセージを書き込む場合、ユーザーは同じトリガー内で 2 つの *service* タグを使用し、タグ値内で個別の正規表現を使って {ITEM.VALUE} マクロから、たとえばサービス A とサービス B の名前を抽出し、それぞれ異なるタグ値を設定しようとすることがあります（たとえば、メッセージ形式が異なる場合）。しかし、正規表現に一致しない場合、この方法は想定どおりに動作しないことがあります。
一致しない正規表現は空のタグ値を生成し、problem イベントと OK イベントの両方に単一の空タグ値があるだけで、それらは相関付けされます。
そのため、サービス A の復旧メッセージが誤ってサービス B のエラーメッセージをクローズしてしまう可能性があります。
- 実際のタグとタグ値は、トリガーが発生したときにのみ表示されます。使用した正規表現が無効な場合、それは暗黙的に *UNKNOWN* 文字列に置き換えられます。初回の障害イベントで *UNKNOWN* タグ値が見落とされると、後続の OK イベントに同じ *UNKNOWN* タグ値が現れ、本来クローズすべきでない障害イベントをクローズしてしまう可能性があります。
- ユーザーがマクロ関数を使わずに {ITEM.VALUE} マクロをタグ値として使用する場合、255 文字の制限が適用されます。ログメッセージが長く、先頭 255 文字に特定性がない場合、これによっても無関係な問題に対して類似したイベントタグが生成される可能性があります。

2 グローバルイベント相関

概要

グローバルイベント相関を使用すると、Zabbix が監視するすべてのメトリクスを対象にして相関関係を作成できます。

まったく異なるトリガーによって作成されたイベントを相関付けし、それらすべてに同じ操作を適用することが可能です。インテリジェントな相関ルールを作成することで、何千もの繰り返し通知を回避し、問題の根本原因に集中できるようになります。

グローバルイベント相関は、1 つのトリガーに基づく問題および復旧のロジックから解放してくれる強力な仕組みです。これまでは、1 つの問題イベントは 1 つのトリガーによって作成され、その問題の復旧も同じトリガーに依存していました。つまり、あるトリガーによって作成された問題を別のトリガーで復旧することはできませんでした。しかし、イベントタグに基づくイベント相関を使用すれば、それが可能になります。

たとえば、ログトリガーはアプリケーションの問題を報告し、ポーリングトリガーはアプリケーションが正常に稼働していることを報告する場合があります。イベントタグを活用することで、ログトリガーには *status:down*、ポーリングトリガーには *status:up* というタグを付けることができます。その後、グローバル相関ルールでこれらのトリガーを関連付け、この相関に対して古いイベントをクローズするといった適切な操作を割り当てることができます。

別の用途として、グローバル相関では類似したトリガーを識別し、それらに同じ操作を適用できます。ネットワークポートの問題ごとに問題レポートを 1 件だけ受け取れたらどうでしょうか。すべてを報告する必要はありません。これもグローバルイベント相関で実現できます。

グローバルイベント相関は、相関ルールで設定します。相関ルールでは、新しい問題イベントを既存の問題イベントとどのように組み合わせるか、また一致した場合に何を行うか（新しいイベントをクローズする、対応する OK イベントを生成して一致した古いイベントをクローズする）を定義します。問題がグローバル相関によってクローズされた場合、そのことは 監視 > 障害の 情報列に表示されます。

グローバル相関ルールを設定できるのは、Super Admin レベルのユーザーのみです。

Attention:
イベント相関は非常に慎重に設定する必要があります。設定によってはイベント処理性能に悪影響を与える可能性があり、また設定を誤ると意図した以上のイベントをクローズしてしまうことがあります（最悪の場合、すべての問題イベントがクローズされる可能性もあります）。

グローバル相関を安全に設定するには、次の重要なポイントに従ってください。

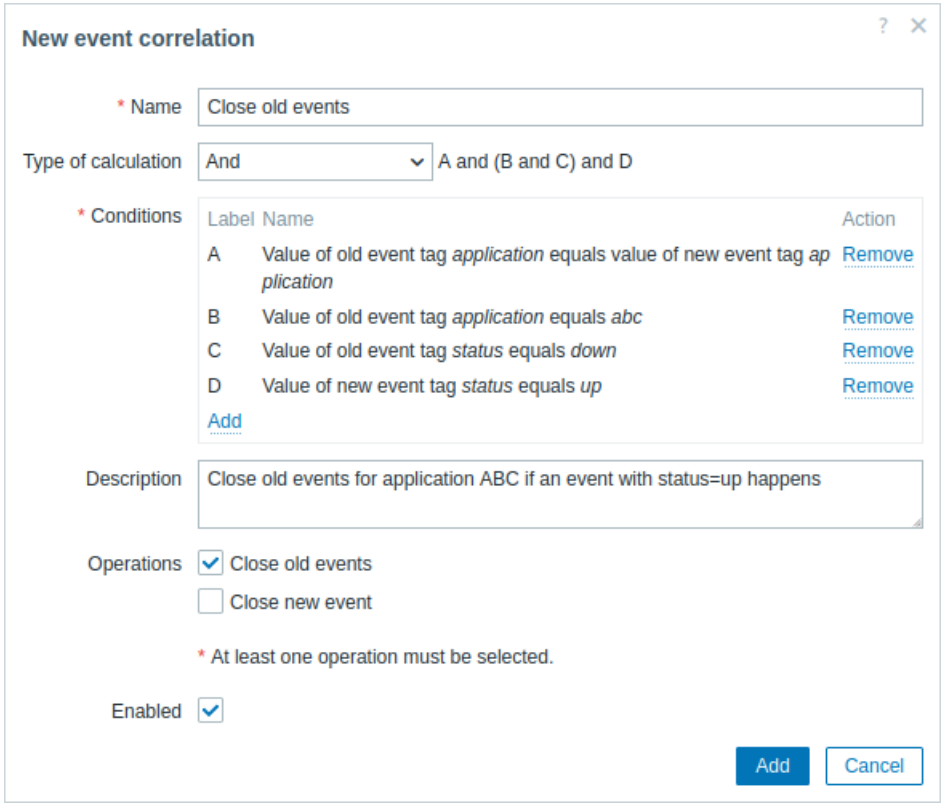
- 相関の対象範囲を絞ってください。
  古いイベントと組み合わせる新しいイベントには、必ず一意のタグを設定し、相関条件として 新しいイベントのタグ名を使用してください。
- 古いイベントをクローズを使用する場合は、古いイベントに対する明示的な条件を追加してください。
  古いイベントをクローズを選択する場合は、必ず少なくとも 1 つの 古いイベント条件（たとえば 古いイベントのタグ名、古いイベントのタグ値、または イベントタグのペア）を追加してください。そうしないと、ルールが無関係な既存の問題イベントにも一致してクローズしてしまう可能性があります（最悪の場合、すべての問題が対象になります）。
  実行時の値（host:port、session id など）の一致には、イベントタグのペアを使用することを推奨します。また、可能であればホストまたはホストグループでさらに対象を絞り込んでください。
- 異なる相関設定で使用される可能性のある一般的なタグ名の使用は避けてください。
- 相関ルールの数は、本当に必要なものだけに限定してください。

参照: 既知の問題。

設定

イベント相関ルールをグローバルに設定するには、次の手順を実行します。

- データ収集 > イベント相関に移動します
- 右側の イベント相関の作成をクリックします（既存のルールを編集する場合は相関名をクリックします）
- フォームに相関ルールのパラメータを入力します

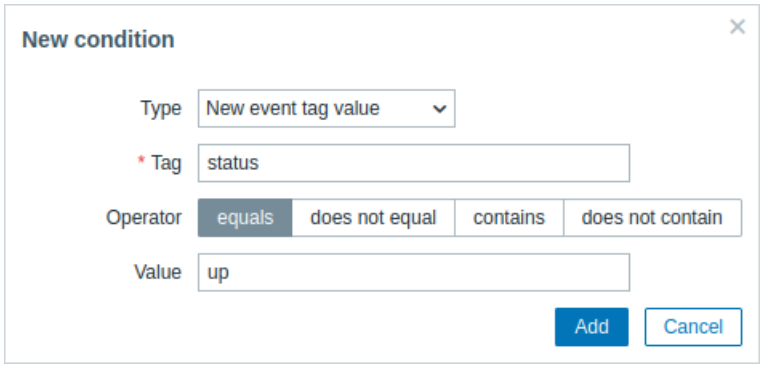


必須入力フィールドには、赤いアスタリスクが付いています。

Parameter	Description
Name	一意の相関ルール名。
Type of calculation	条件の計算方法として、次のオプションを使用できます。 **And** - すべての条件を満たす必要があります **Or** - いずれか 1 つの条件を満たせば十分です **And/Or** - 異なる条件タイプ間では AND、同じ条件タイプ間では OR **Custom expression** - アクション条件を評価するためのユーザー定義の計算式。すべての条件（大文字の A、B、C、... で表される）を含める必要があり、スペース、タブ、括弧 ( )、**and**（大文字小文字を区別）、**or**（大文字小文字を区別）、**not**（大文字小文字を区別）を含めることができます。
Conditions	条件の一覧。条件の設定方法の詳細は以下を参照してください。
Description	相関ルールの説明。

Parameter	Description
Operations	イベントが相関付けられたときに実行する操作のチェックボックスをオンにします。使用可能な操作は次のとおりです。 **Close old events** - 新しいイベントが発生したときに古いイベントをクローズします。*Close old events* 操作を使用する場合は、既存のすべての障害がクローズされる可能性があるため、必ず古いイベントに基づく条件を追加してください。 **Close new event** - 新しいイベントが発生したときにそのイベントをクローズします。 警告**!** *Close old events*/*Close new event* を使用する場合、古いイベント/新しいイベントの条件を空のままにしないでください。古いイベントに一致する条件を追加せずに *Close old events* 操作を選択すると、Zabbix は既存のすべての古いイベントに一致すると判断して、それらをクローズする可能性があります。*Close old events* を使用する場合は、必ず明示的な古いイベント条件（たとえば *Old event tag name* または *Event tag pair*）を追加してください。たとえば、*New event condition* と *Close old events operation* のみを使用するルールは、（欠落している）古いイベント条件を満たすすべての古いイベントに一致し、結果として古い障害をクローズしてしまいます。
Enabled	このチェックボックスをオンにすると、相関ルールが有効になります。

新しい条件の詳細を設定するには、Conditions ブロック内の Add をクリックします。条件の詳細を編集できるポップアップウィンドウが開きます。



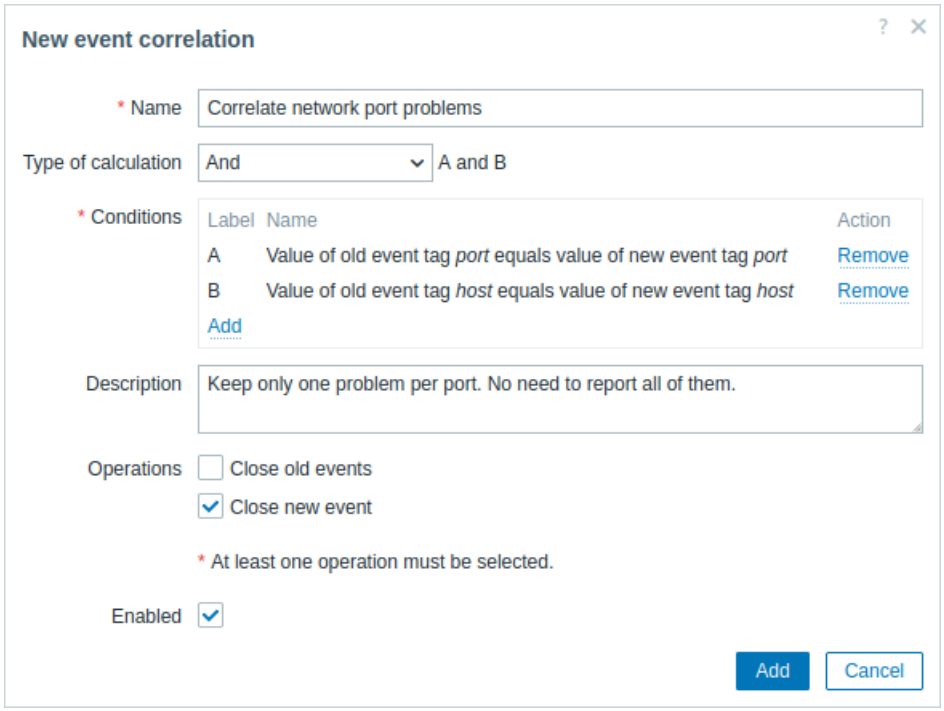
Parameter	Description
New condition	イベントを相関付けるための条件を選択します。 古いイベント条件が指定されていない場合、すべての古いイベントが一致してクローズされる可能性があることに注意してください。同様に、新しいイベント条件が指定されていない場合、すべての新しいイベントが一致してクローズされる可能性があります。 使用可能な条件は次のとおりです。 **Old event tag name** - 一致対象となる古いイベントのタグ名を指定します。 **New event tag name** - 一致対象となる新しいイベントのタグ名を指定します。 **New event host group** - 一致対象となる新しいイベントのホストグループを指定します。 **Event tag pair** - 一致対象となる新しいイベントのタグ名と古いイベントのタグ名を指定します。この場合、両方のイベント内のタグの値が一致すると一致とみなされます。タグの名前は一致している必要はありません。 このオプションは、設定時点では不明な可能性がある実行時の値を一致させるのに便利です（例 も参照）。 **Old event tag value** - 次の演算子を使用して、一致対象となる古いイベントのタグ名と値を指定します。 *equals* - 古いイベントのタグ値と等しい *does not equal* - 古いイベントのタグ値と等しくない *contains* - 古いイベントのタグ値に文字列を含む *does not contain* - 古いイベントのタグ値に文字列を含まない **New event tag value** - 次の演算子を使用して、一致対象となる新しいイベントのタグ名と値を指定します。 *equals* - 新しいイベントのタグ値と等しい *does not equal* - 新しいイベントのタグ値と等しくない *contains* - 新しいイベントのタグ値に文字列を含む *does not contain* - 新しいイベントのタグ値に文字列を含まない

Warning:
設定ミスが起こり得るため、無関係な障害に対して類似したイベントタグが作成される可能性がある場合は、以下のケースを確認してください。

- 実際のタグとタグ値は、トリガーが発生したときにのみ表示されます。正規表現が無効な場合、それは暗黙的に *UNKNOWN* 文字列に置き換えられます。*UNKNOWN* タグ値を持つ最初の障害イベントを見逃すと、同じ *UNKNOWN* タグ値を持つ後続の OK イベントが現れ、本来クローズすべきでない障害イベントをクローズしてしまう可能性があります。
- ユーザーがタグ値としてマクロ関数なしで {ITEM.VALUE} マクロを使用する場合、255 文字の制限が適用されます。ログメッセージが長く、先頭 255 文字に特定性がない場合、これによっても無関係な障害に対して類似したイベントタグが生成される可能性があります。

例

同じネットワークポートからの問題イベントの繰り返し発生を停止します。



このグローバル相関ルールは、トリガー上に ホストおよび *port* タグの値が存在し、それらが元のイベントと新しいイベントで同一である場合に、問題を相関付けします。

この操作により、同じネットワークポート上の新しい問題イベントがクローズされ、元の問題のみがオープンのまま維持されます。

6 タグ付け

概要

タグは、タグ名とタグ値で構成されます。エンティティにタグを付ける際は、名前だけを使用することも、値と組み合わせて使用することもできます（例: `mysql`、`jira`、`target:mysql`、`service:jira` など）。

タグは、さまざまなエンティティに定義できます。

- テンプレート
- ホスト
- アイテム
- Web シナリオ
- トリガー
- サービス
- テンプレートのアイテムおよびトリガー
- ホスト、アイテム、トリガーのプロトタイプ

Note:
タグの定義に関する一般的な推奨事項、および テンプレート、アイテム、トリガー、ローレベルディスカバリルールに関する具体的なガイダンスについては、Zabbix 公式ガイドラインを参照してください。

タグには複数の用途がありますが、特に重要なのはイベントの識別です。エンティティにタグが付けられている場合、そのタグ付きエンティティに関連する新しいイベントは、そのタグを継承します。例えば、次のとおりです。

- タグ付きテンプレートの場合 - ホストのすべての障害（テンプレートのトリガーによって作成されたもの）は、テンプレートのタグを継承します。
- タグ付きホストの場合 - ホストのすべての障害は、ホストのタグを継承します。
- タグ付きアイテム/Web シナリオの場合 - アイテム/Web シナリオのすべての障害は、アイテム/Web シナリオのタグを継承します。
- タグ付きトリガーの場合 - トリガーによって作成されたすべての障害は、トリガーのタグを継承します。

障害イベントは、テンプレート、ホスト、アイテム/Web シナリオ、トリガーというエンティティの連鎖全体から、すべてのタグを継承します。同一の `tag:value` の組み合わせ（マクロ解決後）は 1 つにまとめられるため、重複は回避されます。

カスタムイベントタグにより、より柔軟な運用が可能になります。例えば、次のようなことができます。

- イベント相関は、イベントタグに基づいて設定できます。
- アクション条件は、イベントタグに基づいて設定できます。
- アイテムの障害は、イベントタグに基づいてグループ化できます。
- 障害タグは、障害をサービスにマッピングするために使用できます。

エンティティには、同じタグ名で異なるタグ値を持つタグを付けることができます（例: `component:memory` と `component:storage`）。同様に、エンティティには値のないタグと、同じ名前で値のあるタグを付けることもできます（例: `database` と `database:postgresql`）。このようなタグは重複とは見なされません。

ユースケース

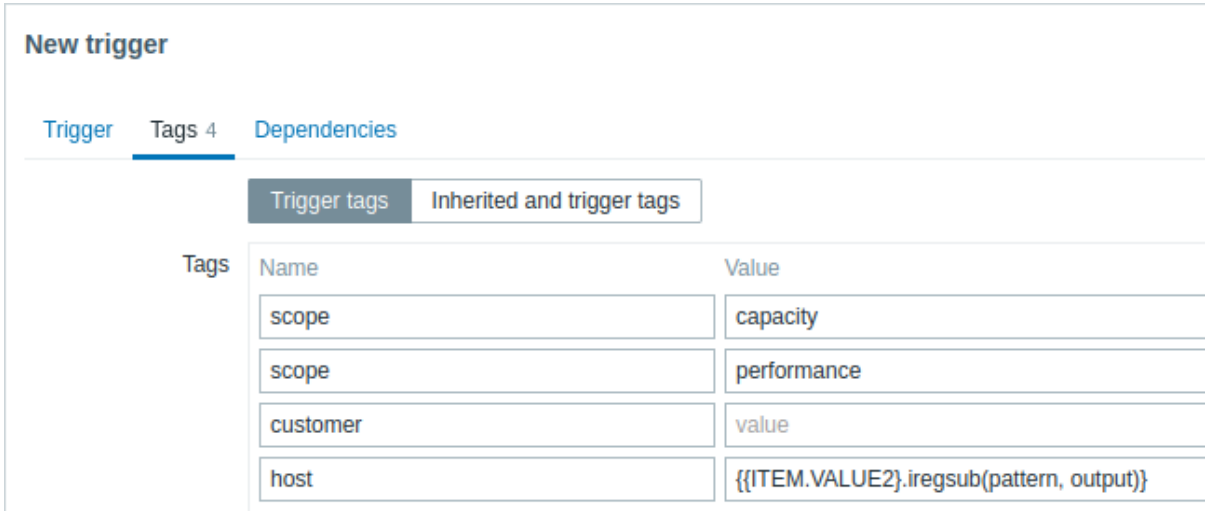
タグ付けの一般的なユースケースは次のとおりです。

1. トリガーイベントを識別する:
   - トリガーのタグを定義します（例: `scope:performance`）。
   - このトリガーによって作成された障害には、そのトリガーのタグが付きます。
2. テンプレート継承された障害を識別する:
   - テンプレートのタグを定義します（例: `target:mysql`）。
   - このテンプレートのトリガーによって作成された障害には、そのテンプレートのタグが付きます。
3. ホストの障害を識別する:
   - ホストのタグを定義します（例: `service:jira`）。
   - このホストのトリガーによって作成された障害には、そのホストのタグが付きます。
4. 関連するアイテムをフィルタリングする:
   - アイテムのタグを定義します（例: `component:cpu`）。
   - 監視 → 最新データ では、アイテムを `component:cpu` タグでフィルタリングできます。
5. アイテム値から抽出した情報をタグ値として使用する:
   - タグ値としてマクロを含むタグを定義します（例: `tag-name:{{ITEM.VALUE<N>}.regsub()}` ）。
   - 監視 → 障害 では、障害のタグ値はアイテム値から抽出されたデータに解決されます。
6. ログファイル内の障害を識別し、それぞれ個別にクローズする:
   - ログ監視アイテム のトリガーに対して、マクロを使用してアイテム値から値を抽出するトリガーのタグを定義します（例: `service:{{ITEM.VALUE<N>}.regsub()}` ）。
   - トリガー設定 で、イベント相関関係 を設定します:
     - 障害イベント生成モードを「複数」に設定します。
     - *OK* イベントでクローズを「タグ値が一致する場合はすべての障害」に設定します。
     - 一致に使用するタグを設定します。
   - ログアイテムのトリガーによって作成された障害にはそのトリガーのタグが付き、個別にクローズされます。
7. 通知をフィルタリングする:
   - トリガーのタグを定義します（例: trigger1 には `scope:security`、trigger2 には `scope:availability`）。
   - アクション条件 でタグフィルタリングを使用し、タグデータに一致するイベントの通知のみを受信します。
8. 通知内で障害を識別する:
   - トリガーのタグを定義します。
   - 障害通知で{EVENT.TAGS} マクロを使用します。
   - 障害通知にはトリガーのタグが含まれるため、その通知がどのアプリケーション/サービスに属するかを容易に識別できます。
9. テンプレートタグを使用して設定作業を簡素化する:
   - テンプレートのトリガーのタグを定義します。
   - このテンプレートトリガーから作成されたトリガーには、そのタグが付きます。

10. ローレベルディスカバリ（LLD）からタグ付きトリガーを作成する:
    - タグ名またはタグ値に LLD マクロを含むトリガープロトタイプのタグを定義します（例: `scope:{#FSNAME}`）。
    - このトリガープロトタイプから作成されたトリガーには、そのタグが付きます。
11. サービスタグを使用してサービスを照合する:
    - サービスタグ を定義します。
    - 一致するタグを持つサービスに対してサービスアクション を設定します。
    - さらに、SLA 計算のために、サービスタグを使用してサービスをSLA に関連付けます。
12. サービス障害タグを使用してサービスを障害に関連付ける:
    - サービス設定 で障害タグ を定義します（例: `target:mysql`）。
    - 一致するタグを持つ障害は自動的にそのサービスに相関付けられ、設定されたサービスステータス計算ルールに基づいてサービスステータスが変化します。
13. ホストがメンテナンスモードのときに障害を抑制する:
    - メンテナンス期間の設定 でタグを定義します。
    - 定義されたタグを持つ障害は抑制されます。
14. ユーザーグループにアクセス権を付与する:
    - ユーザーグループ設定 でタグを定義します。
    - そのユーザーグループのユーザーは、定義されたタグを持つ障害のみを表示できます。

設定

タグは専用のタブで定義できます。たとえば、トリガーの設定です。



マクロのサポート

タグ内の組み込みマクロおよびユーザーマクロは、イベント発生時に展開されます。イベントが発生するまでは、これらのマクロは Zabbix Web インターフェースで未展開のまま表示されます。

ローレベルディスカバリマクロは、ディスカバリ処理中に展開されます。

以下のマクロは、トリガーのタグ名および値で使用できます。

- {ITEM.VALUE}, {ITEM.VALUE.AGE}, {ITEM.VALUE.DATE}, {ITEM.VALUE.TIME}, {ITEM.VALUE.TIMESTAMP}, {ITEM.LASTVALUE}, {ITEM.LASTVALUE.AGE}, {ITEM.LASTVALUE.DATE}, {ITEM.LASTVALUE.TIME}, {ITEM.LASTVALUE.TIMESTAMP}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}, および {HOST.ID} の組み込みマクロ
- {INVENTORY.*} の組み込みマクロ（トリガー式内の 1 つまたは複数のホストからホストインベントリ値を参照するため）
- ユーザーマクロおよびコンテキスト付きユーザーマクロ（コンテキストにはローレベルディスカバリマクロを含めることができます）
- ローレベルディスカバリマクロ（トリガープロトタイプのタグ内のみ）

以下のマクロは、テンプレート、ホスト、アイテム/Web シナリオのタグ名および値で使用できます。

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}, および {HOST.ID} の組み込みマクロ
- {INVENTORY.*} の組み込みマクロ
- ユーザーマクロ
- ローレベルディスカバリマクロ（ホストプロトタイプおよびアイテムプロトタイプのタグ内のみ）

以下のマクロは、トリガーベースの通知で使用できます。

- {EVENT.TAGS} および {EVENT.RECOVERY.TAGS} の組み込みマクロ（これらのマクロは、イベントタグまたは復旧イベントタグのカンマ区切りリストに展開されます）
- {EVENT.TAGSJSON} および {EVENT.RECOVERY.TAGSJSON} の組み込みマクロ（これらのマクロは、イベントタグオブジェクトまたは復旧イベントタグオブジェクトを含む JSON 配列に展開されます）

トリガータグでの部分文字列抽出

タグ名またはタグ値の設定では、マクロ関数を使用した部分文字列抽出がサポートされています。この関数は、サポートされているマクロによって取得された値に正規表現を適用します。例えば、次のようになります。

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

マクロ解決後のタグ名またはタグ値が 255 文字を超える場合、255 文字に切り詰められます。

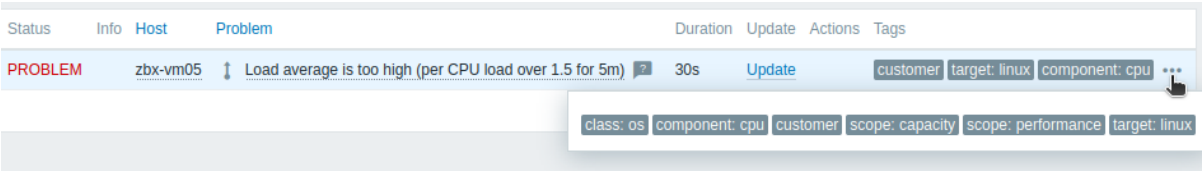
関連情報: イベントタグ付けのためのローレベルディスカバリマクロでのマクロ関数の使用も参照してください。

イベントタグの表示

タグが定義されている場合、新しいイベントで次の場所に表示できます。

- 監視 → 障害
- 監視 → 障害 → イベントの詳細
- ダッシュボード → 障害ウィジェット

表示されるタグの順序と数は、監視 → 障害または 障害ダッシュボードウィジェットの タグ表示優先度および タグの表示フィルタリングオプションによって決まります。
表示できるタグは最大 3 つまでであることに注意してください。タグがそれより多い場合は、3 つのドットにマウスオーバーすると、ポップアップウィンドウですべてのタグが表示されます。



7 視覚化

サイドバーから、参照したい項目を選択してください。

1 グラフ

概要

Zabbix には大量のデータが流入するため、数字だけでなく何が起こっているかを視覚的に表現できれば、ユーザーにとってははるかに簡単になります。

ここでグラフが登場します。グラフを使用すると、データフローを一目で把握したり、問題を関連付けたり、何かがいつ始まったかを発見したり、いつ問題が発生する可能性があるかを提示したりすることができます。

Zabbix はユーザーに以下を提供します。

- 1 アイテムのデータの組み込みシンプルグラフ
- より複雑なものが作れるカスタムグラフ
- 複数のアイテムの比較にアドホックグラフ
- モダンなカスタマイズが可能なベクトルグラフと円グラフ

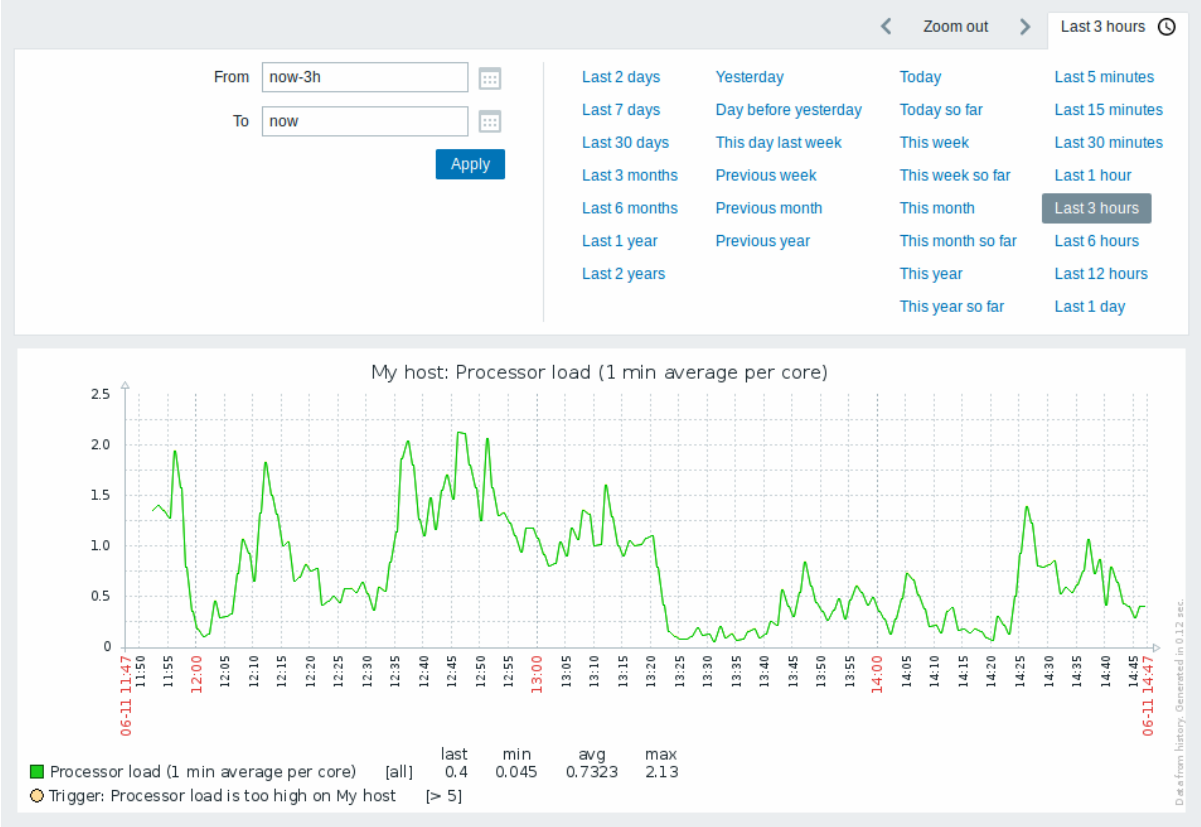
1 シンプルグラフ

概要

item で収集したデータを視覚化するためのシンプルなグラフを提供します。

シンプルグラフを表示するために、ユーザ側で設定する必要はありません。Zabbix はシンプルグラフを自由に利用する
 ことができます。

Monitoring → *Latest data* から各 item のグラフリンクをクリックするだけで、グラフが表示されます。



Note:
数値項目については、シンプルなグラフが用意されています。文字項目については、*Monitoring → Latest data* に履歴へのリンクが用意されています。

期間セレクター

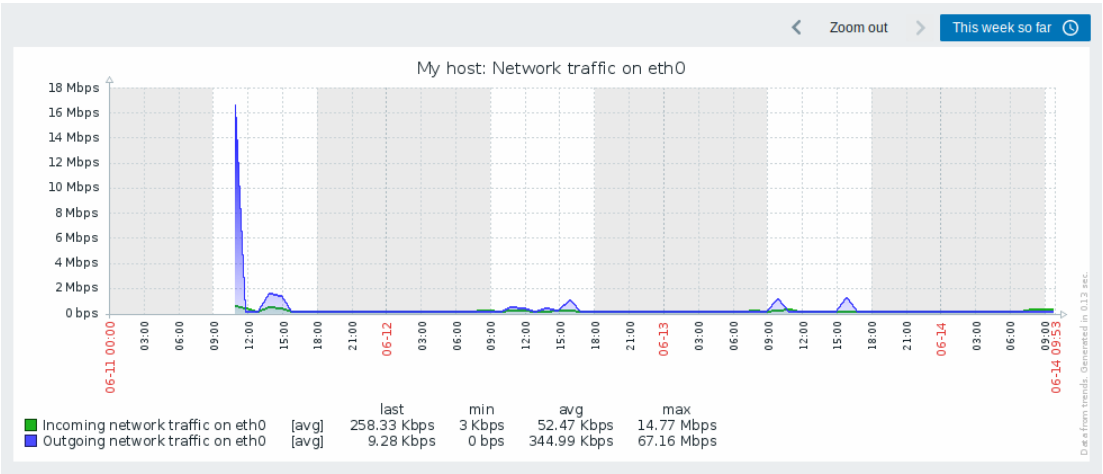
グラフの上にある期間セレクターを使用すると、よく使用する期間をマウスのクリック 1 回で選択できます。詳細については、期間セレクターを参照してください。

最近のデータと長い期間のデータ

ごく最近のデータでは、受信した各値を結ぶ **1** 本の線が描かれます。1 本の線は、1 つの値に対して少なくとも 1 つの
 水平ピクセルが利用可能である限り、描かれます。

濃い緑色の線は平均値を、薄いピンク色と薄い緑色の線はその時点の最大値と最小値を示しています。最大値と最小値の間は、
 黄色の背景で埋められます。

稼働時間（稼働日）はグラフでは白い背景で表示され、非稼働時間はグレーで表示されます。
 （*Original blue* デフォルトのフロントエンドのテーマで表示されます）



シンプルなグラフでは常に稼働時間が表示されますが、custom graphs では、ユーザーの
 好みで表示することができます。

3 ヶ月以上のグラフの場合、稼働時間は表示されません。

トリガー線

単純なトリガーは、トリガーの重要度の色に黒いダッシュが付いた線で表示されます。グラフ上の青い線と凡例に表示された
 トリガー情報に注意してください。グラフ上に最大 3 つのトリガーラインを表示することができます。もし、より多くの
 トリガーがある場合、重要度の低いトリガーが優先されます。トリガーは常にシンプルなグラフで表示されますが、
 custom graphsで表示することはユーザーの好みによります。



履歴/トレンドからの生成

グラフは、アイテムの履歴またはトレンドに基づいて描画できます。

Web インターフェースのデバッグモードを有効にしているユーザーには、グラフの右下に灰色の縦書きキャプションが表示され、データの取得元が示されます。

履歴とトレンドのどちらが使用されるかには、いくつかの要因が影響します。

- アイテム履歴の保持期間。たとえば、アイテム履歴を 14 日間保持することができます。その場合、14 日より古いデータはすべてトレンドから取得されます。
- グラフ内のデータの密度。横方向のグラフの 1 ピクセルあたりに表示する秒数が 3600/16 を超える場合、トレンドデータが表示されます（同じ期間のアイテム履歴がまだ利用可能であっても）。
- トレンドが無効になっている場合、グラフの作成にはアイテム履歴が使用されます（その期間のデータが利用可能な場合）。

データが存在しない場合

通常の更新間隔を持つアイテムでは、アイテムのデータが収集されていない場合、グラフには何も表示されません。

ただし、trapper アイテムおよびスケジュールされた更新間隔を持つアイテム（通常の更新間隔が 0 に設定されている場合）では、最初に収集された値までの区間と、最後に収集された値からグラフの終端までの区間に直線が描画されます。この線は、それぞれ最初/最後の値のレベルに表示されます。

生の値への切り替え

右上のドロップダウンで、単純なグラフから *Values/500 latest values* のリストへ切り替えることができます。これは、グラフを構成する数値を表示するのに便利です。

ここで表示される数値は生のものであり、単位や数値の後処理は行われていません。しかし、値のマッピングは適用されます。

既知の問題

グラフについては、known issuesを参照してください。

2 カスタムグラフ

概要

カスタムグラフは、その名の通り、カスタマイズ機能を提供するものです。

単純なグラフは、1 つの項目のデータを見るには適していますが、設定機能はありません。

そのため、グラフのスタイルや線の表示方法を変更したり、複数の項目、例えばトラフィックの入出力を 1 つのグラフで
 比較したい場合は、カスタムグラフが必要になります。

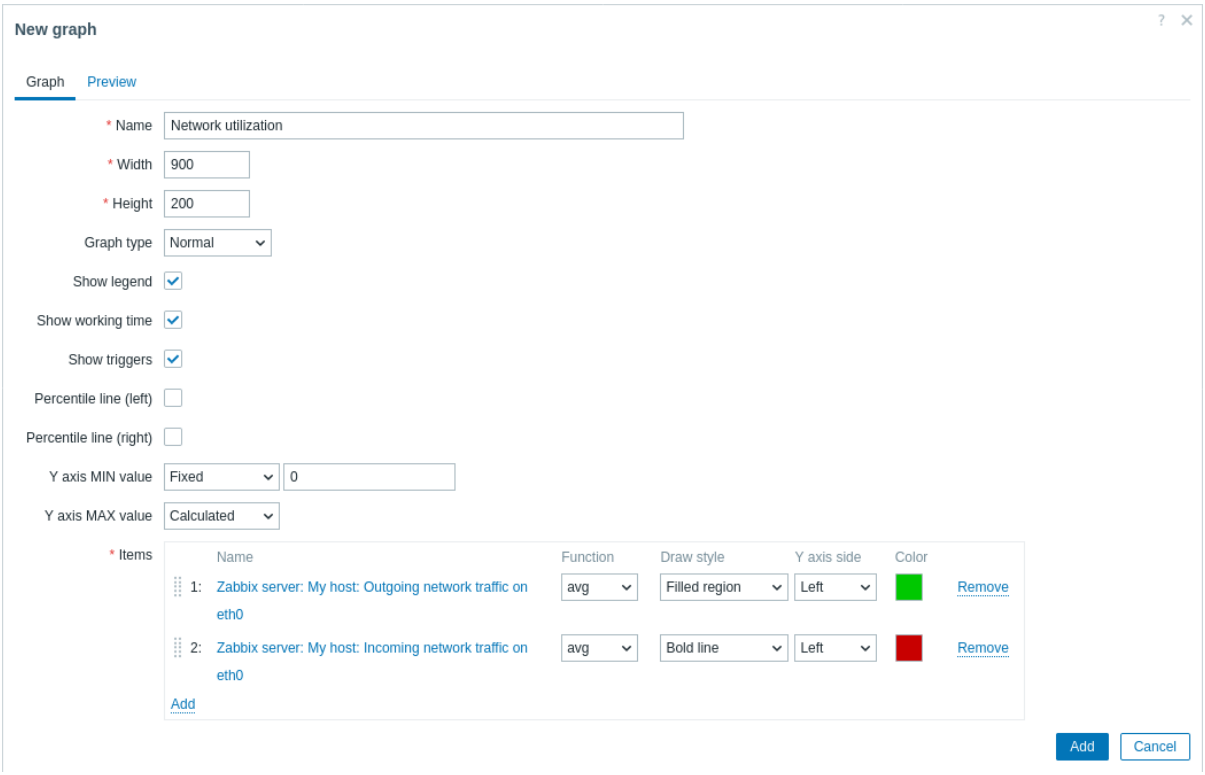
カスタムグラフは、手動で設定します。

カスタムグラフは、1 つのホスト、複数のホスト、または 1 つのテンプレートに対して作成することができます。

カスタムグラフの設定

カスタムグラフを作成するには、以下のようにします。

- データ収集 → ホスト (もしくはテンプレート) へ移動します。
- 目的のホストまたはテンプレートの行のグラフをクリックします。
- グラフ画面でグラフの作成をクリックします。
- グラフの属性を編集します。



必須入力項目には、赤いアスタリスクが表示されています。

グラフの属性:

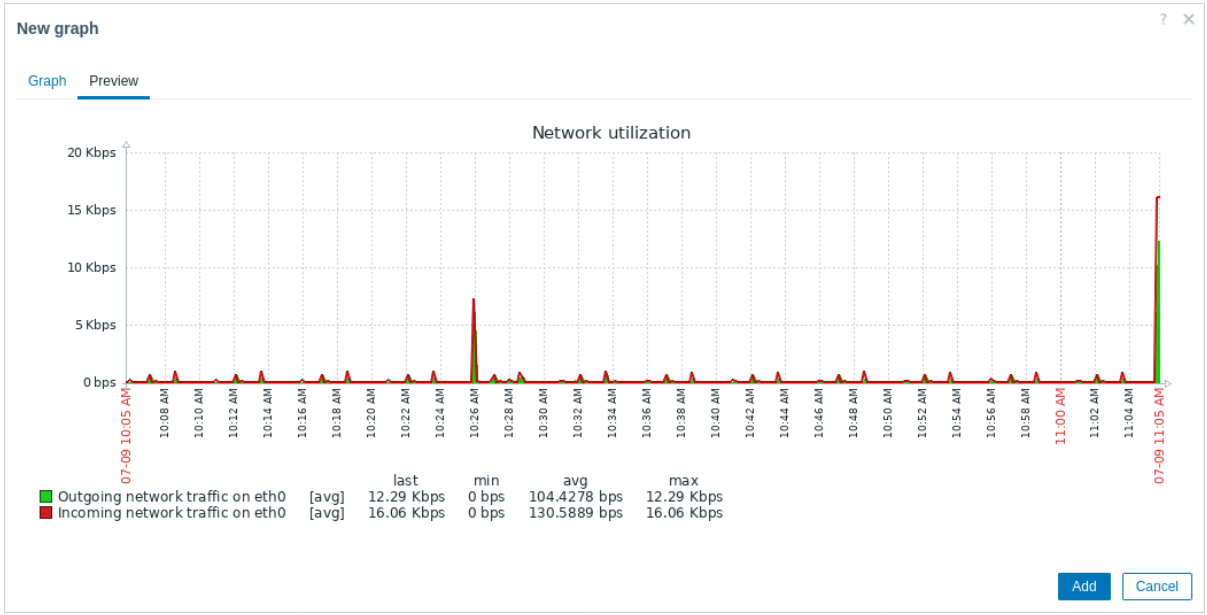
パラメーター	説明
名前	Unique graph name. Expression macros are supported in this field, but only with `avg`, `last`, `min` and `max` functions, with time as parameter (for example, `{?avg(/host/key,1h)}`). {HOST.HOST<1-9>} macros are supported for the use within this macro, referencing the first, second, third, etc. host in the graph, for example `{?avg(/{HOST.HOST2}/key,1h)}`. Note that referencing the first host with this macro is redundant, as the first host can be referenced implicitly, for example `{?avg(//key,1h)}`.
幅	Graph width in pixels (for preview and pie/exploded graphs only).
高さ	Graph height in pixels.

パラメーター	説明
グラフのタイプ	Graph type: **Normal** - normal graph, values displayed as lines **Stacked** - stacked graph, filled areas displayed **Pie** - pie graph **Exploded** - ”exploded” pie graph, portions displayed as ”cut out” of the pie
凡例を表示	Checking this box will set to display the graph legend.
ワーキングタイムの表示	If selected, non-working hours will be shown with a gray background. This parameter is not available for pie and exploded pie graphs.
トリガーを表示	If selected, simple triggers will be displayed as lines with black dashes over trigger severity color. This parameter is not available for pie and exploded pie graphs.
パーセンタイルライン *(左)*	Display percentile for left Y-axis. If, for example, 95% percentile is set, then the percentile line will be at the level where 95 percent of the values fall under. Displayed as a bright green line. Only available for normal graphs.
パーセンタイルライン *(右)*	Display percentile for right Y-axis. If, for example, 95% percentile is set, then the percentile line will be at the level where 95 percent of the values fall under. Displayed as a bright red line. Only available for normal graphs.

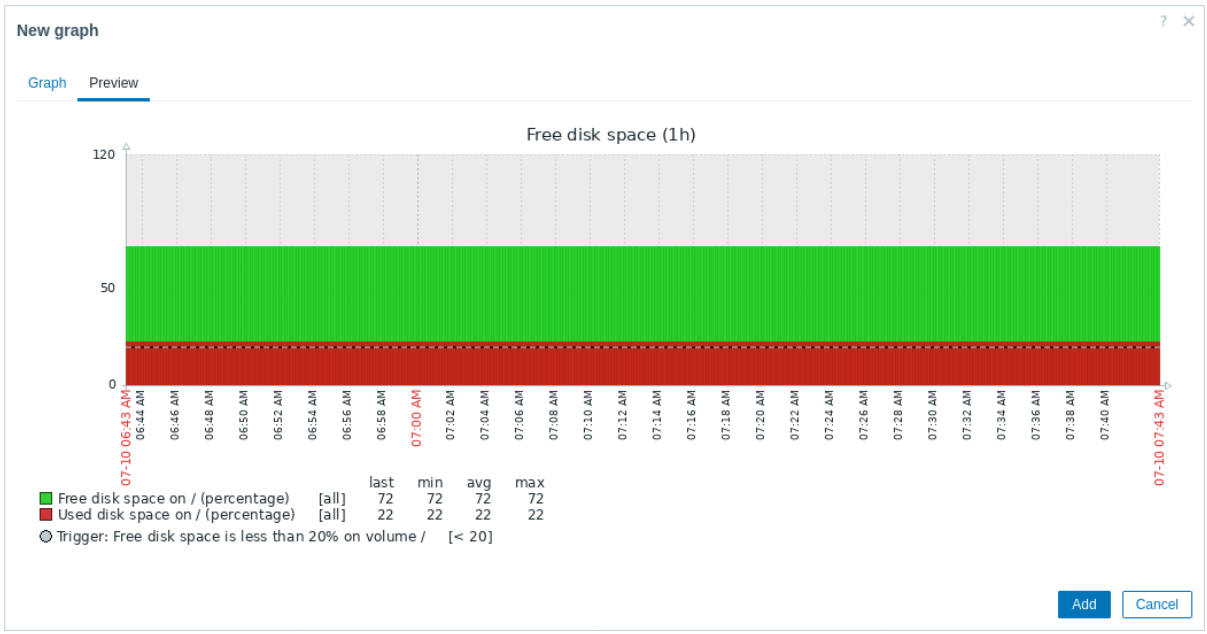
パラメーター		説明
Y 軸の最小値		Minimum value of Y-axis: **Calculated** - Y axis minimum value will be automatically calculated. **Fixed** - fixed minimum value for Y-axis. **Item** - last value of the selected item will be the minimum value. This parameter is not available for pie and exploded pie graphs.
Y 軸の最大値		Maximum value of Y-axis: **Calculated** - Y axis maximum value will be automatically calculated. **Fixed** - fixed maximum value for Y-axis. **Item** - last value of the selected item will be the maximum value This parameter is not available for pie and exploded pie graphs.
3D 表示		Enable 3D style. For pie and exploded pie graphs only.
アイテム		Items, data of which are to be displayed in this graph. Click on *Add* to select items. You can also select various displaying options (function, draw style, left/right axis display, color).
	Sort order (0→100)	Draw order. 0 will be processed first. Can be used to draw lines or regions behind (or in front of) another. You can drag and drop items using the icon at the beginning of a line to set the sort order or which item is displayed in front of the other.
	Name	Name of the selected item is displayed as a link. Clicking on the link opens the list of other available items.
	Type	Type (only available for pie and exploded pie graphs): **Simple** - the value of the item is represented proportionally on the pie **Graph sum** - the value of the item represents the whole pie Note that coloring of the ”graph sum” item will only be visible to the extent that it is not taken up by ”proportional” items.
	Function	Select what values will be displayed when more than one value exists per vertical graph pixel for an item: **all** - display all possible values (minimum, maximum, average) in the graph. Note that for shorter periods this setting has no effect; only for longer periods, when data congestion in a vertical graph pixel increases, ’all’ starts displaying minimum, maximum, and average values. This function is only available for *Normal* graph type. See also: Generating graphs from history/trends. **avg** - display the average values **last** - display the latest values. This function is only available if either *Pie/Exploded pie* is selected as graph type. **max** - display the maximum values **min** - display the minimum values
	Draw style	Select the draw style (only available for normal graphs; for stacked graphs filled region is always used) to apply to the item data - *Line*, *Bold line*, *Filled region*, *Dot*, *Dashed line*, *Gradient line*.
	Y axis side	Select the Y axis side to show the item data - *Left*, *Right*.
	Color	Select the color to apply to the item data.

グラフプレビュー

Preview タブでは、グラフのプレビューが表示され、作成中のグラフをすぐに確認することができます。



なお、プレビューでは、テンプレートアイテムのデータは表示されません。



この例では、トリガレベルを表示する破線の太線と、凡例に表示されるトリガ情報に注目します。

Note:
表示できるトリガーは 3 行までです。それ以上のトリガーがある場合、重要度の低いトリガーが優先的に表示されます。グラフの高さが 120 ピクセル以下に設定されている場合、凡例にはトリガーが表示されません。

3 アドホックグラフ

概要

simple graph は 1 つのアイテムのデータにアクセスするのに優れ、custom graphs はカスタマイズオプションを提供しますが、この 2 つのどちらでも、少ない手間かつメンテナンス不要で、複数のアイテムを比較するグラフをすばやく作成することはできません。

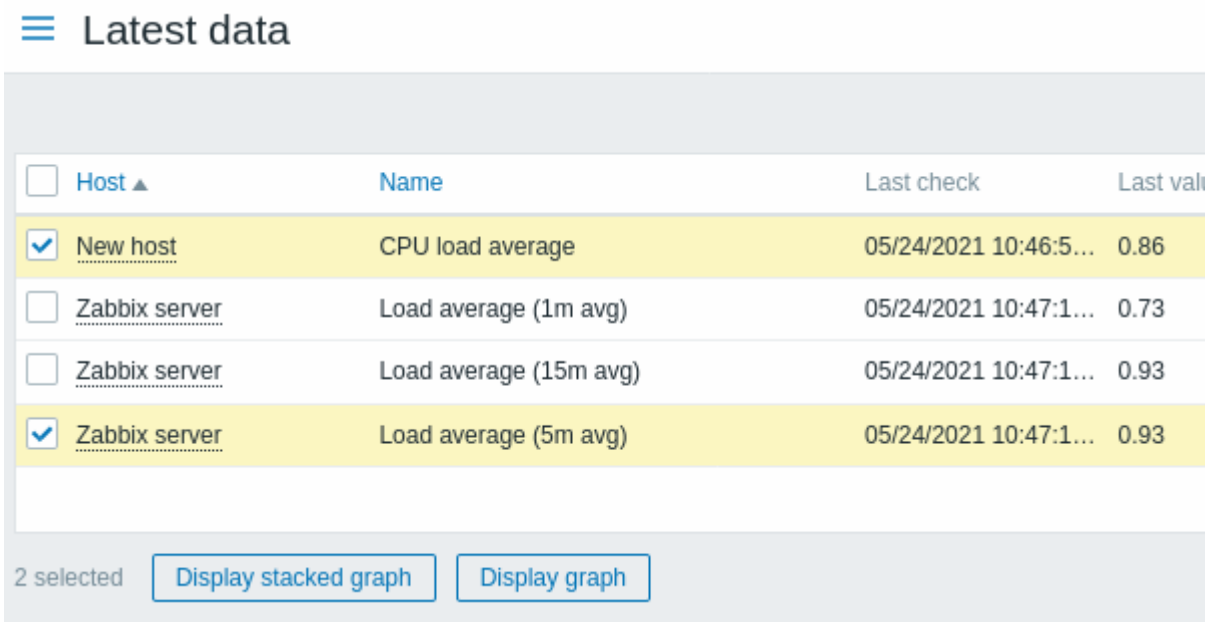
この問題に対処するため、複数のアイテムに対するアドホックなグラフを非常にすばやく作成することができます。

設定

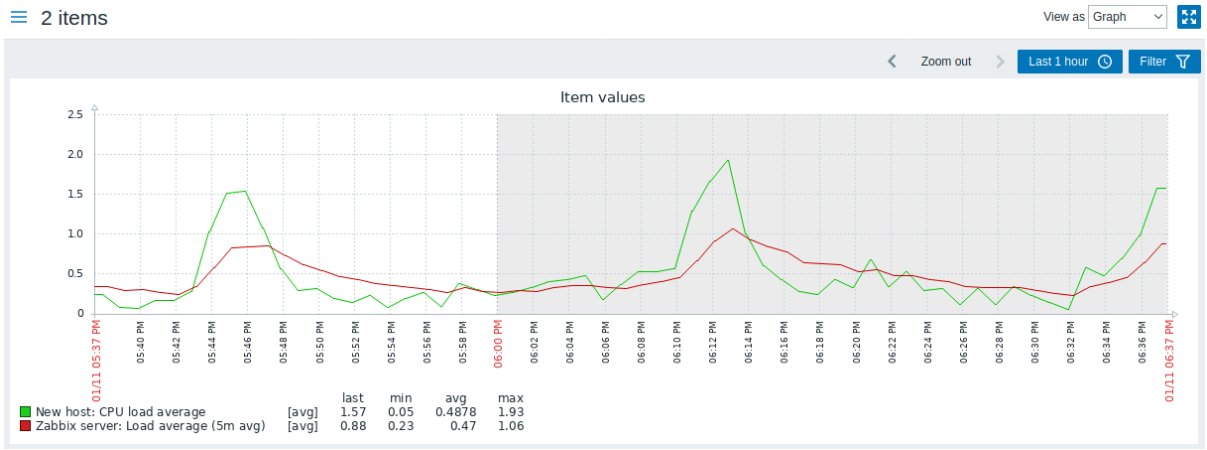
アドホックグラフを作成するには、次の手順を実行します。

- 監視 → 最新データに移動します
- フィルターを使用して、表示したいアイテムを表示します
- グラフ化したいアイテムのチェックボックスをオンにします

- 積み上げグラフを表示または グラフを表示ボタンをクリックします

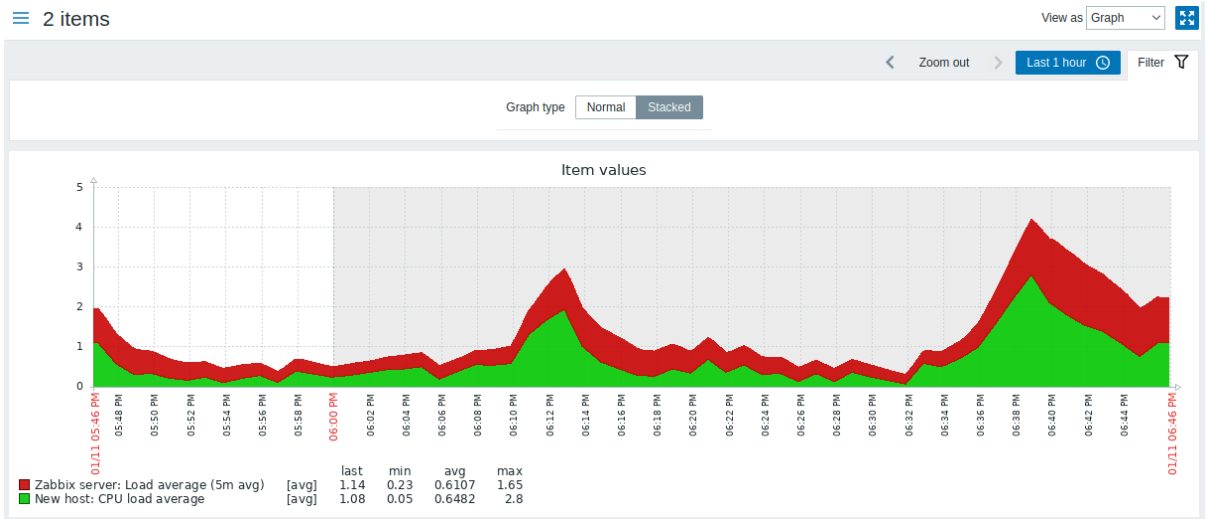


グラフは即座に作成されます。



グラフに表示される線が多くなりすぎるのを避けるため、各アイテムについては平均値のみが表示されることに注意してください(最小値/最大値の線は表示されません)。トリガーおよびトリガー情報はグラフに表示されません。

作成されたグラフウィンドウでは、期間セレクターを利用でき、通常の折れ線グラフから積み上げグラフへ（またはその逆へ）切り替えることもできます。



4 グラフの集計

概要

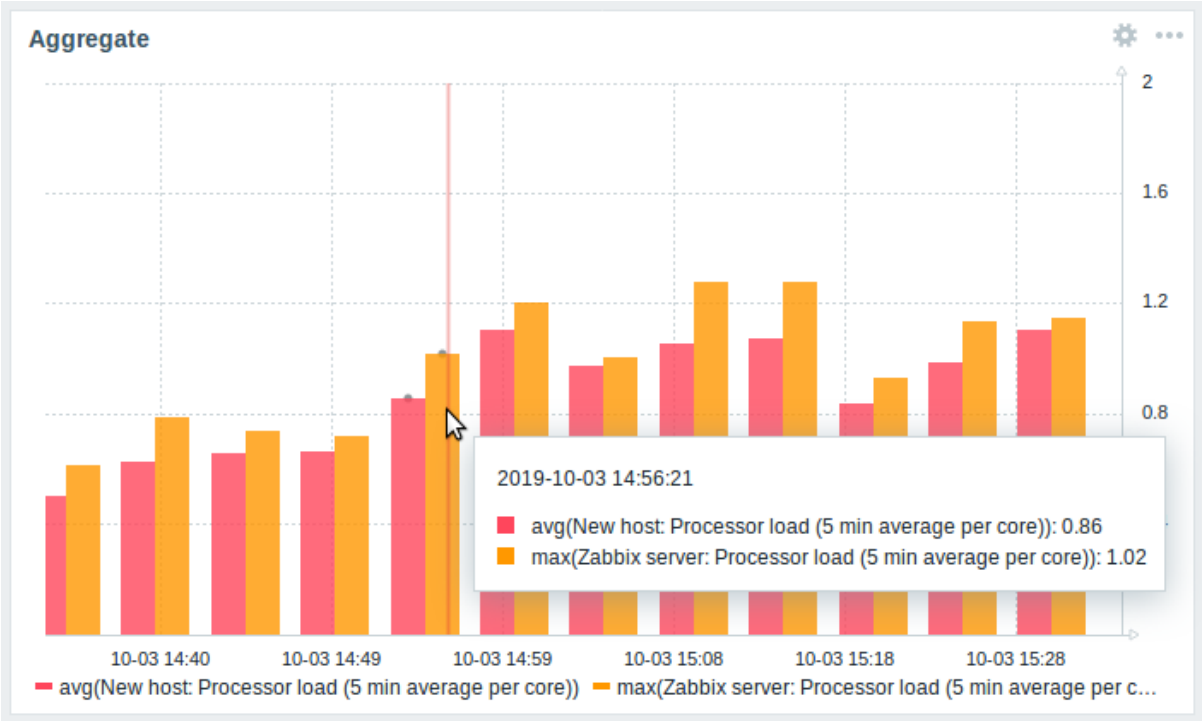
ダッシュボードのグラフウィジェットおよび円グラフウィジェットで利用できる集計関数を使用すると、すべての値の代わりに、選択した期間（5 分、1 時間、1 日）の集計値を表示できます。

このセクションでは、グラフウィジェットの集計オプションについて詳しく説明します。

集計オプションは次のとおりです。

- min
- max
- avg
- count
- sum
- first（最初に表示される値）
- last（最後に表示される値）

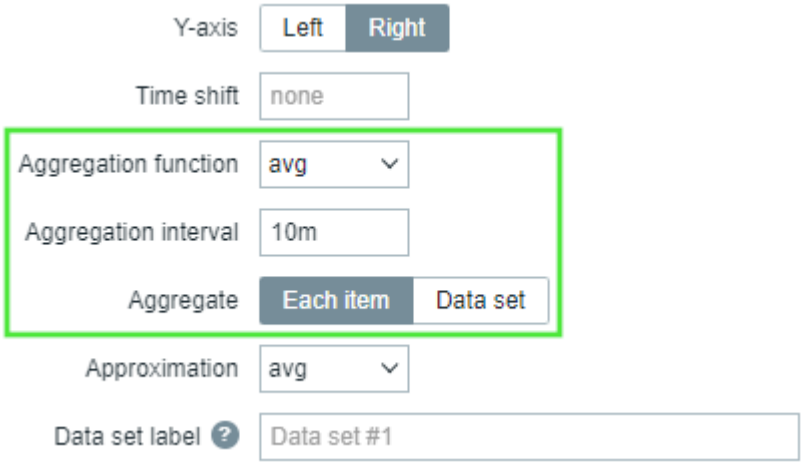
データ集計の最も魅力的な用途は、一定期間のデータを見やすく並べて比較できることです。



グラフ内のある時点にマウスカーソルを合わせると、アイテムとその集計値に加えて、日付と時刻が表示されます。アイテムは括弧内に表示され、使用されている集計関数が先頭に付きます。グラフウィジェットでデータセットラベルが設定されている場合、ラベルは括弧内に表示され、使用されている集計関数が先頭に付きます。これはグラフ上の時点の日付と時刻であり、実際の値の日付と時刻ではないことに注意してください。

設定

集計のオプションは、グラフウィジェットを設定する際のデータセット設定で利用できます。



集計関数と時間間隔を選択できます。データセットには複数のアイテムが含まれる場合があるため、各アイテムごとに個別の集計データを表示するか、データセット内のすべてのアイテムを 1 つの集計値として表示するかを選択できる別のオプションもあります。

使用例

Nginx サーバーへの平均リクエスト数

Nginx サーバーへの 1 日あたりの平均リクエスト数 (秒) を表示します。

- データセットに秒間リクエスト数の item を追加します。
- 集約関数 avg を選択し、インターバル 1d を指定します。
- 棒グラフが表示され、各棒は一日あたりの平均秒間リクエスト数を表します。

クラスタ間の最小週次ディスク容量

1 週間におけるクラスタ間の最小ディスク容量を表示します。

- データセットに追加します: ホスト `cluster*`, キー”Free disk space on /data”
- 集約関数 `min` を選択し、間隔 `1w` を指定します。
- 棒グラフが表示され、各棒はクラスタの各 /data ボリュームの 1 週間あたりの最小ディスク容量を表します。

2 ネットワークマップ

概要

ネットワークの管理をしている場合、インフラの概要をどこかで把握したいと思うかもしれません。
 そのために、Zabbix ではネットワークや好きなもののマップを作成することができます。

すべてのユーザがネットワークマップを作成することができます。
 マップはパブリック (すべてのユーザが利用可能) またはプライベート (選択されたユーザが利用可能) にすることができます。

configuring a network mapに進みます。

1 ネットワークマップの設定

概要

Zabbix でマップを設定するには、まず一般的なパラメータを定義してマップを作成し、その後、実際のマップに要素とそれらのリンクを追加していきます。

マップには、ホスト、ホストグループ、トリガー、画像、または別のマップである要素を配置できます。

マップ要素を表すためにアイコンが使用されます。アイコンとともに表示する情報を定義できるほか、最近の障害が特別な方法で表示されるように設定することもできます。アイコン同士をリンクし、リンク上に表示する情報を定義することもできます。

アイコンをクリックしてアクセスできるカスタム URL を追加できます。これにより、ホストアイコンをホストのプロパティに、またはマップアイコンを別のマップにリンクできます。

マップ内の障害数は、原因障害についてのみ表示されます。

マップは *Monitoring* > *Maps* で管理され、ここで設定、管理、表示を行うことができます。監視ビューでは、アイコンをクリックして、一部のスクリプトや URL へのリンクを利用できます。

ネットワークマップはベクターグラフィックス（SVG）に基づいています。

パブリックマップとプライベートマップ

Zabbix のすべてのユーザー（非管理者ユーザーを含む）は、ネットワークマップを作成できます。マップには所有者がいます。所有者とは、そのマップを作成したユーザーです。マップはパブリックまたはプライベートにできます。

- パブリックマップはすべてのユーザーに表示されますが、表示するには、ユーザーが少なくとも 1 つのマップ要素に対する読み取り権限を持っている必要があります。パブリックマップは、ユーザー/ユーザーグループがこのマップに対する読み書き権限を持ち、対応するマップのすべての要素（リンク内のトリガーを含む）に対して少なくとも読み取り権限を持っている場合に編集できます。
- プライベートマップは、その所有者と、所有者によってマップが共有されたユーザー/ユーザーグループにのみ表示されます。通常の（Super admin ではない）ユーザーは、自分が所属するグループ、およびそれらのグループに所属するユーザーとのみ共有できます。Admin レベルのユーザーは、所有者であるか共有ユーザー一覧に含まれているかどうかに関係なく、プライベートマップを表示できます。プライベートマップは、マップの所有者が編集でき、また、ユーザー/ユーザーグループがこのマップに対する読み書き権限を持ち、対応するマップのすべての要素（リンク内のトリガーを含む）に対して少なくとも読み取り権限を持っている場合にも編集できます。

ユーザーが読み取り権限を持たないマップ要素は、グレーアウトされたアイコンで表示され、その要素上のすべてのテキスト情報は非表示になります。ただし、ユーザーがそのトリガーに対する権限を持っていない場合でも、トリガーラベルは表示されます。

マップに要素を追加するには、ユーザーはその要素に対して少なくとも読み取り権限も持っている必要があります。

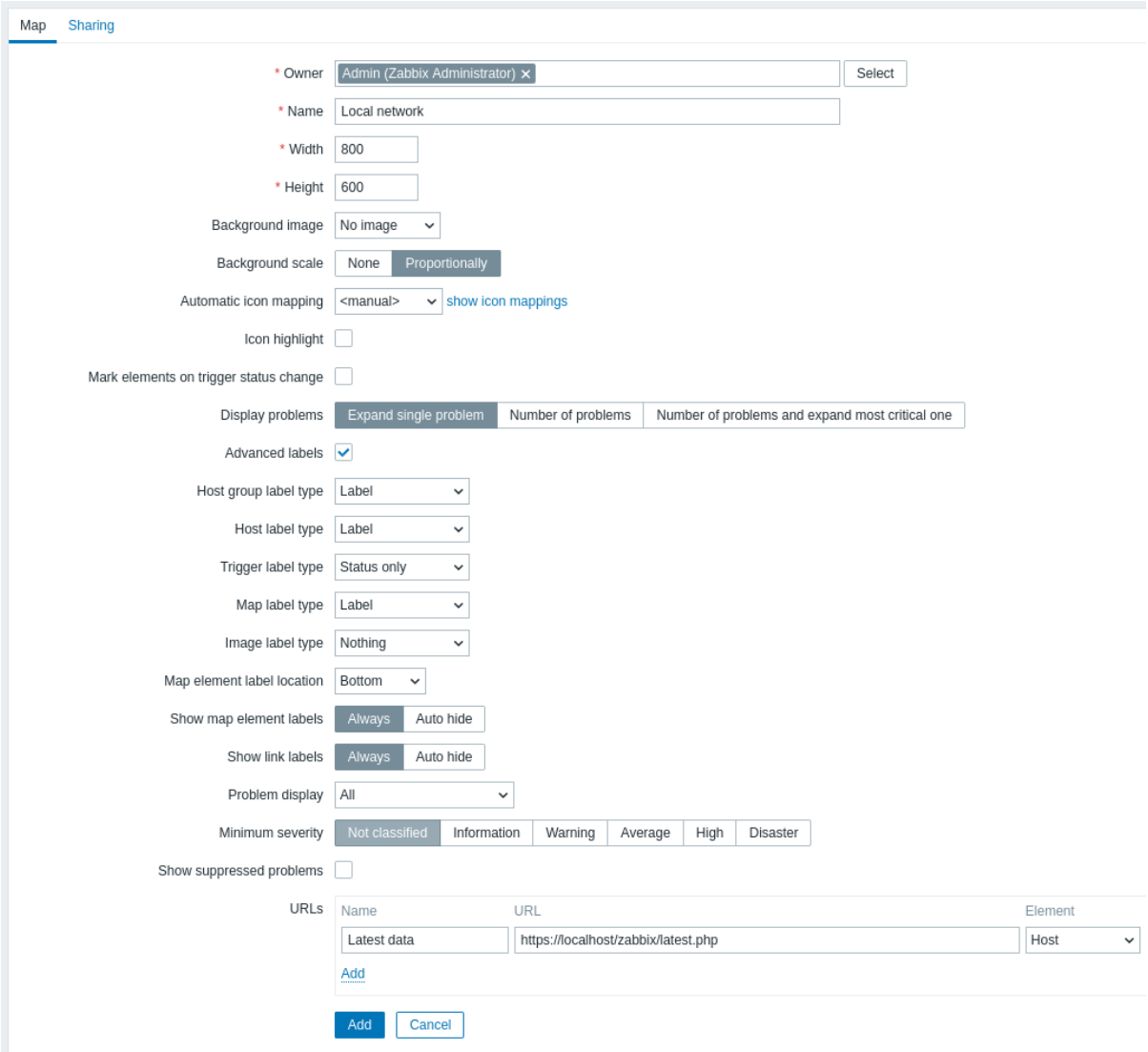
マップの作成

マップを作成するには、次の手順を実行します。

1. 監視 > マップに移動します。
2. すべてのマップが表示されるビューに移動します。
3. マップの作成をクリックします。

既存のマップの設定フォームにある 複製ボタンを使用して、新しいマップを作成することもできます。
このマップには、一般的なレイアウト属性に加え、既存のマップの要素を含む、既存のマップのすべてのプロパティが引き継がれます。

マップタブには、マップの一般属性が含まれています。



必須入力フィールドには、すべて赤いアスタリスクが付いています。

マップの一般属性:

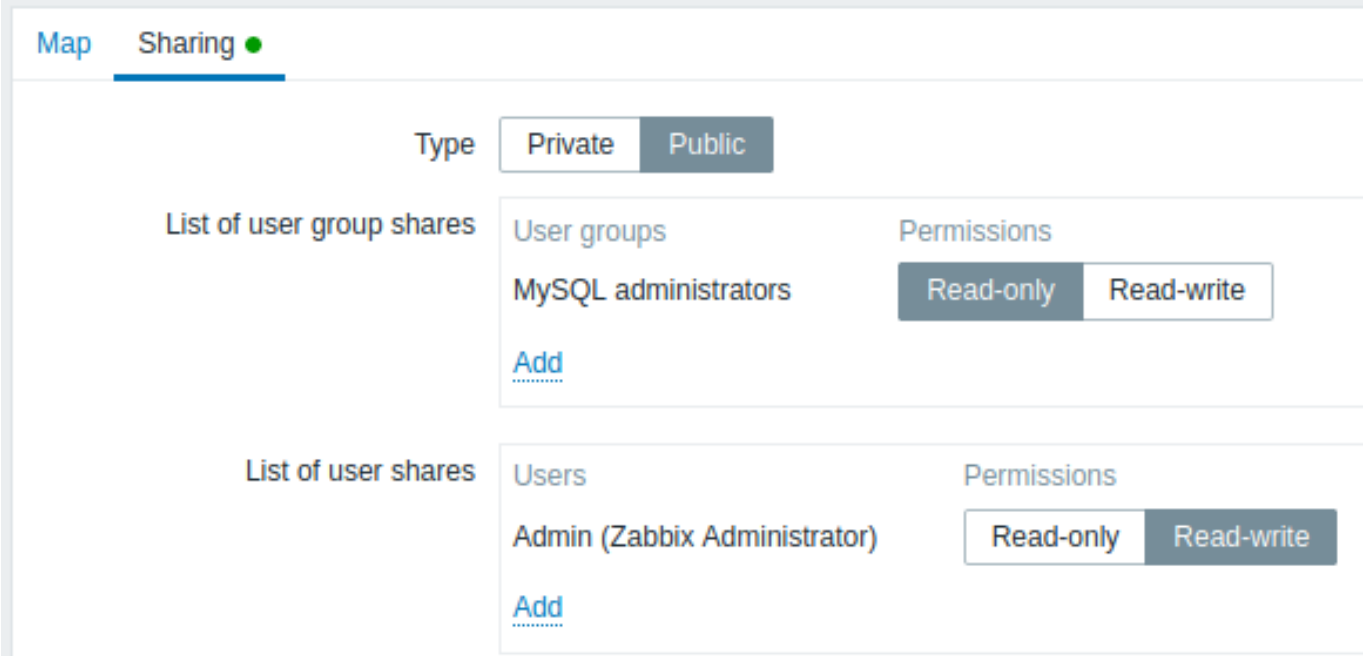
パラメータ	説明
Owner	マップ所有者の名前。
Name	一意のマップ名。
Width	マップの幅（ピクセル単位）。
Height	マップの高さ（ピクセル単位）。

パラメータ	説明
Background image	背景画像の使用: **No image** - 背景画像なし（白背景） **Image** - 選択した画像を背景画像として使用します。地図画像やその他の画像を使用して、マップをより分かりやすくできます。
Background scale	背景のスケーリングの使用: **None** - スケーリングなし; **Proportionally** - 画像の縦横比を維持したまま、マップ背景全体を覆います（デフォルト）。
Automatic icon mapping	管理 > 一般設定 > アイコンのマッピングで設定された自動アイコンマッピングを使用するように設定できます。アイコンマッピングでは、特定のアイコンを特定のホストインベントリフィールドに対応付けることができます。
Icon highlight	このチェックボックスをオンにすると、マップ要素が強調表示されます。 アクティブなトリガーを持つ要素には、最も高い深刻度のトリガーと同じ色の円形背景が付きます。さらに、すべての障害が確認済みである場合は、円の周囲に太い緑色の線が表示されます。 「無効」または「メンテナンス中」状態の要素には、それぞれ灰色およびオレンジ色の四角形背景が付きます。 参照: マップの表示
Mark elements on trigger status change	トリガーの状態が最近変更された場合（新しい障害または復旧）、ラベルがない要素アイコンの 3 辺にマーカー（内向きの赤い三角形）で強調表示されます。マーカーは 30 分間表示されます。
Display problems	マップ要素に障害をどのように表示するかを選択します: **Expand single problem** - 障害が 1 件のみの場合は障害名を表示し、それ以外の場合は障害の総数を表示します。 **Number of problems** - 障害の総数を表示します **Number of problems and expand most critical one** - 最も重大な障害の名前と障害の総数を表示します。 「最も重大」は、障害の深刻度に基づいて判定され、同じ場合は障害イベント ID（ID が大きい、つまり後の障害が先に表示される）で判定されます。トリガーマップ要素の場合は、障害の深刻度に基づき、同じ場合はトリガーリスト内のトリガーの位置で判定されます。同じトリガーに複数の障害がある場合は、最も新しいものが表示されます。
Advanced labels	このチェックボックスをオンにすると、要素タイプごとに個別のラベルタイプを定義できます。
Host group label type	マップ要素に使用するラベルタイプ: **Label** - マップ要素ラベル **IP address** - IP アドレス **Element name** - 要素名（例: ホスト名） **Status only** - ステータスのみ（OK または PROBLEM） **Nothing** - ラベルを表示しません
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	マップ要素に対するラベルの位置: **Bottom** - マップ要素の下 **Left** - 左側 **Right** - 右側 **Top** - 上側
Show map element labels	マップ要素ラベルの表示方法を選択します: **Always** - 常に表示（デフォルト）; **Auto hide** - ホバーまたは選択されていない場合はラベルを非表示にします。
Show link labels	リンクラベルの表示方法を選択します: **Always** - 常に表示（デフォルト）; **Auto hide** - ホバーまたは選択されていない場合はラベルを非表示にします。
Problem display	障害数の表示方法: **All** - 障害の総数を表示します **Separated** - 未確認の障害数を、障害総数とは分けて表示します **Unacknowledged only** - 未確認の障害数のみを表示します
Minimum severity	選択した最小深刻度レベル未満の障害は、マップに表示されません。 たとえば *Warning* を選択すると、*Information* および *Not classified* レベルのトリガーの変化はマップに反映されません。
Show suppressed problems	このチェックボックスをオンにすると、ホストのメンテナンスによって抑制（非表示）される障害も表示します。

パラメータ	説明
URLs	ここでは各要素タイプごとに URL を定義できます。各項目の意味は次のとおりです: **Name** - URL のラベル; **URL** - URL（最大 2048 文字）; **Element** - 要素タイプ（例: ホスト、ホストグループなど）。 これらは、マップ表示モードでユーザーが要素をクリックしたときにリンクとして表示されます。 マップ URL の名前と値にはマクロを使用できます。完全な一覧については、サポートされているマクロ を参照し、「map URL names and values」を検索してください。

共有

共有タブには、マップのタイプと、プライベートマップの共有オプション（ユーザーグループ、ユーザー）が含まれます。



パラメータ	説明
Type	マップのタイプを選択します。 **Private** - マップは選択したユーザーグループとユーザーにのみ表示されます **Public** - マップはすべてのユーザーに表示されます
List of user group shares	マップへのアクセスを許可するユーザーグループを選択します。 読み取り専用または読み書き可能のアクセスを許可できます。
List of user shares	マップへのアクセスを許可するユーザーを選択します。 読み取り専用または読み書き可能のアクセスを許可できます。

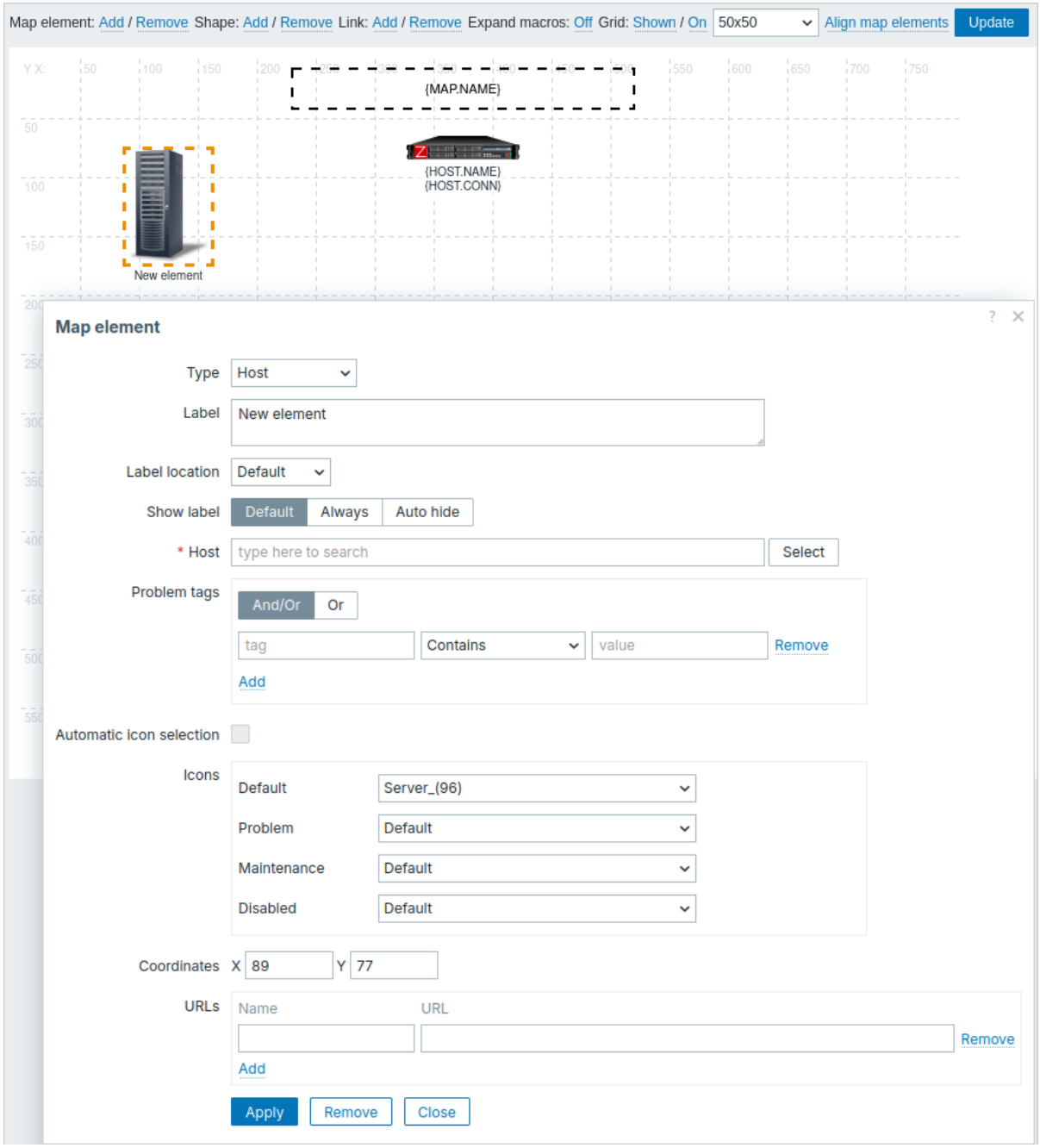
このマップを保存するために *Add* をクリックすると、名前、サイズ、およびいくつかの設定を持つ空のマップが作成されます。次に、いくつかの要素を追加する必要があります。そのためには、マップ一覧で *Edit* をクリックして編集可能な領域を開きます。

要素の追加

要素を追加するには、`Map element` の横にある *Add* をクリックします。新しい要素はマップの左上隅に表示されます。好きな場所にドラッグ＆ドロップしてください。

Grid オプションが”On” の場合、要素は常にグリッドに揃えられます（ドロップダウンからさまざまなグリッドサイズを選択でき、グリッドの表示/非表示も切り替えられます）。要素を揃えずに任意の場所へ配置したい場合は、このオプションを”Off” にしてください。（後で *Align map elements* をクリックして、ばらばらに配置された要素をグリッドに揃えることもできます。）

いくつかの要素を配置したら、名前を付けるなどして区別したくなるでしょう。要素をクリックするとフォームが表示され、要素タイプの設定、名前の指定、別のアイコンの選択などを行えます。



マップ要素の属性:

Parameter	Description
Type	要素のタイプ: **Host** - 選択したホストのすべてのトリガーのステータスを表すアイコン **Map** - マップ内のすべての要素のステータスを表すアイコン **Trigger** - 1 つ以上のトリガーのステータスを表すアイコン **Host group** - 選択したグループに属するすべてのホストのすべてのトリガーのステータスを表すアイコン **Image** - リソースにリンクされていないアイコン
Label	アイコンのラベル。任意の文字列を指定できます。 マクロおよび複数行の文字列を使用できます。 このフィールドでは式マクロがサポートされていますが、使用できるのは時間をパラメータとする `avg`、`last`、`min`、`max` 関数のみです（例: `{?avg(/host/key,1h)}`）。 サポートされているマクロの完全な一覧は、supported macros を参照し、'map element labels' を検索してください。

Parameter	Description
Label location	アイコンに対するラベルの位置: **Default** - マップのデフォルトのラベル位置; **Bottom** - アイコンの下; **Left** - 左側; **Right** - 右側; **Top** - 上側。
Show label	要素ラベルの表示方法を選択します: **Default** - マップ設定の *Show map element labels* を使用（デフォルト）; **Always** - 常に表示（デフォルト）; **Auto hide** - ホバーまたは選択されていない場合はラベルを非表示にします。
Host	ホストを選択します。あるいは、ホスト名を入力します。このフィールドはオートコンプリートに対応しているため、ホスト名を入力し始めると一致するホストのドロップダウンが表示されます。下にスクロールして 1 つ選択してください。選択を解除するには、削除 (x) アイコンをクリックします。 このフィールドは、*Host* タイプの要素でのみ使用できます。
Map	マップを選択します。あるいは、マップ名を入力します。このフィールドはオートコンプリートに対応しているため、マップ名を入力し始めると一致するマップのドロップダウンが表示されます。下にスクロールして 1 つ選択してください。選択を解除するには、削除 (x) アイコンをクリックします。 このフィールドは、*Map* タイプの要素でのみ使用できます。
Triggers	下の *New triggers* フィールドで 1 つ以上のトリガーを選択し、*Add* をクリックします。あるいは、トリガー名を入力し始めて、一致するトリガーのドロップダウンリストから 1 つ選択します。選択を解除するには、削除 (x) アイコンをクリックします。 このフィールドは、*Trigger* タイプの要素でのみ使用できます。 選択したトリガーの順序は変更できますが、同じ深刻度のトリガー内でのみ変更可能です。複数トリガーの選択は、編集モードと表示モードの両方で {HOST.*} マクロの解決にも影響します。 編集モードでは、最初に表示される {HOST.*} マクロは、リスト内の最初のトリガー（トリガーの深刻度に基づく）に応じて解決されます。 表示モードは、一般的なマップ属性のDisplay problems パラメータに依存します: - *Expand single problem* モードが選択されている場合、最初に表示される {HOST.*} マクロは、最後に検出された障害トリガー（深刻度は関係なし）に応じて、または障害が検出されていない場合はリスト内の最初のトリガーに応じて解決されます。 - *Number of problems and expand most critical one* モードが選択されている場合、最初に表示される {HOST.*} マクロは、トリガーの深刻度に応じて解決されます。
Host group	ホストグループを選択します。あるいは、ホストグループ名を入力します。このフィールドはオートコンプリートに対応しているため、ホストグループ名を入力し始めると一致するホストグループのドロップダウンが表示されます。下にスクロールして 1 つ選択してください。選択を解除するには、削除 (x) アイコンをクリックします。 このフィールドは、*Host group* タイプの要素でのみ使用できます。
Problem tags	ウィジェットに表示される障害の数を制限するためのタグを指定します。特定のタグやタグ値を含めることも除外することも可能です。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます: **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別） **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含める（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外する（部分一致、大文字と小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があり、同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です このフィールドは、*Host* および *Host group* タイプの要素でのみ使用できます。
Automatic icon selection	このチェックボックスをオンにすると、表示するアイコンの決定にアイコンマッピングが使用されます。
Icons	次の場合に、要素に対して異なるアイコンを表示するよう選択できます: *Default*、*Problem*、*Maintenance*、*Disabled*。 このフィールドは、*Host* タイプの要素でのみ使用できます。

Parameter	Description
Coordinates	マップ要素の **X** 座標および **Y** 座標。
URLs	ここでは要素固有の URL を設定できます。各項目は次のとおりです: **Name** - URL のラベル; **URL** - URL（最大 2048 文字）。 これらは、ユーザーがマップ表示モードで要素をクリックしたときにリンクとして表示されます。 要素に独自の URL があり、かつそのタイプに対してマップレベルの URL が定義されている場合、それらは同じメニュー内にまとめて表示されます。 マクロはマップ要素の名前と値で使用できます。完全な一覧については、supported macros を参照し、'map URL names and values' を検索してください。

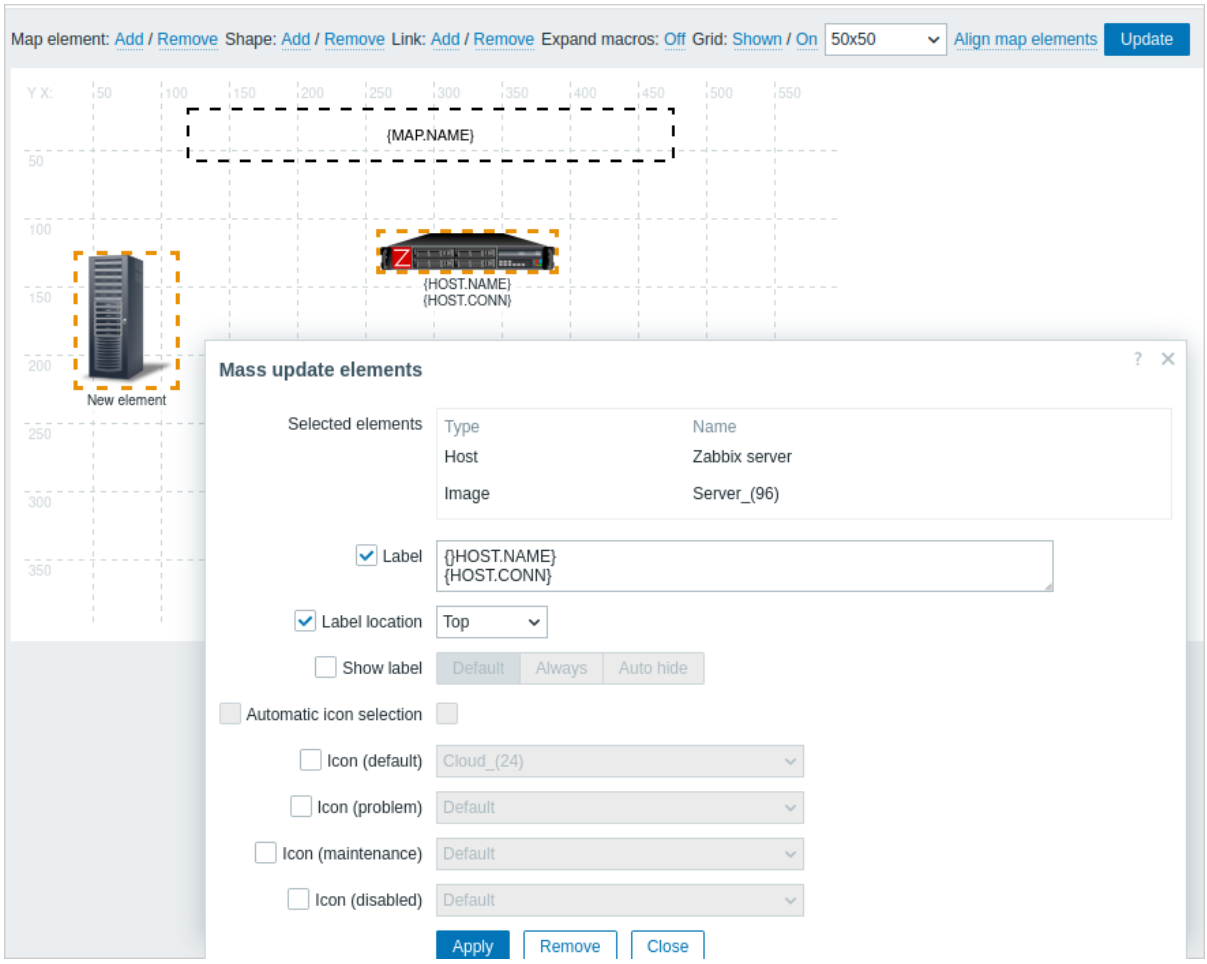
Attention:
追加した要素は自動的には保存されません。ページを離れると、すべての変更が失われる可能性があります。
そのため、右上隅にある **Update** ボタンをクリックすることをお勧めします。クリックすると、その後のポップアップで何を選択しても変更は保存されます。
選択したグリッドオプションも、各マップごとに保存されます。

要素の選択

要素を選択するには、1 つを選択してから *Ctrl* を押しながら他の要素を選択します。

また、編集可能な領域で矩形をドラッグし、その中のすべての要素を選択することで、複数の要素を選択することができます。

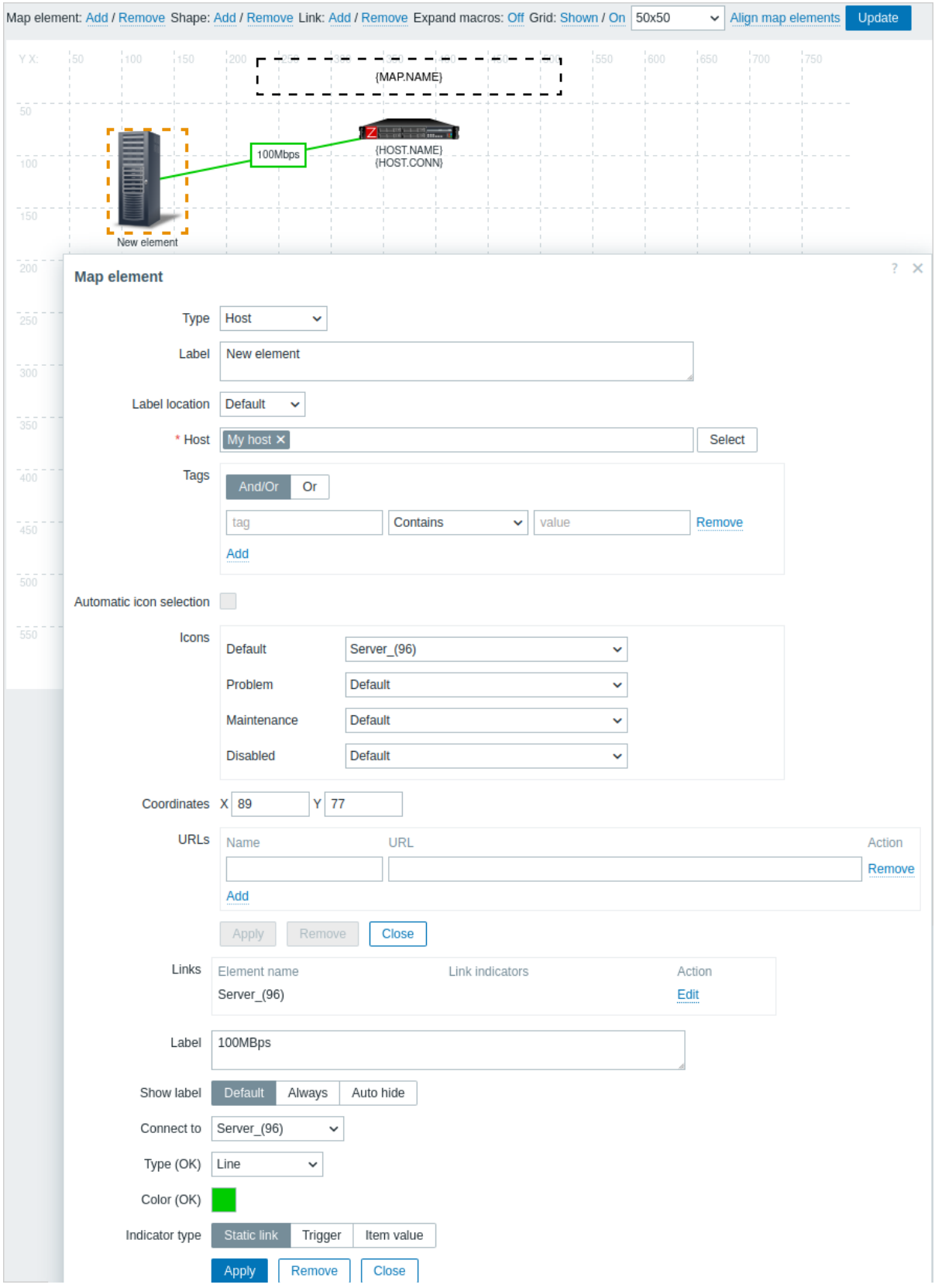
複数の要素を選択すると、要素のプロパティフォームが一括更新モードに移行し、選択した要素の属性を一括して変更する
 ことができます。そのためには、チェックボックスを使って属性をマークし、その属性に新しい値を入力します。
 ここでマクロを使用することもできます (例えば、要素のラベルに {HOST.NAME} を使用)。



要素のリンク

マップ上にいくつかの要素を配置したら、次はそれらをリンクします。2 つの要素をリンクするには、まずそれらを選択する必要があります。要素を選択した状態で、リンク の横にある 追加をクリックします。

リンクを作成すると、単一要素のフォームに リンクセクションが追加されます。リンク属性を編集するには、編集をクリックします。



リンク属性:

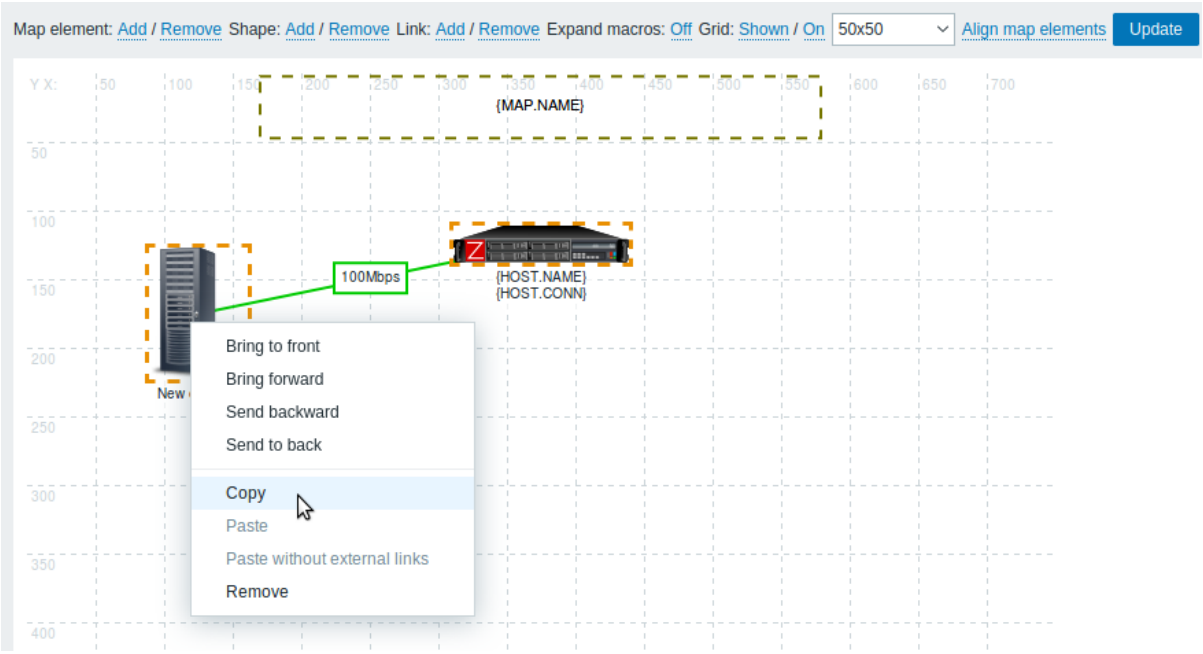
Parameter	Description
Label	リンクの上に表示されるラベル。 このフィールドでは式マクロがサポートされていますが、使用できるのは時間をパラメータとする `avg`、`last`、`min`、`max` 関数のみです（例: `{?avg(/host/key,1h)}`）。
Show label	リンクラベルの表示方法を選択します: **Default** - マップ設定の *Show map element labels* 設定を使用します（デフォルト）。 **Always** - 常に表示します（デフォルト）。 **Auto hide** - ホバーまたは選択されていない場合はラベルを非表示にします。

Parameter	Description
Connect to	リンクの接続先となる要素。
Type (OK)	デフォルトのリンクスタイル。スタイルとして *Line*、*Bold line*、*Dot*、または *Dashed line* を選択します。
Color (OK)	デフォルトのリンク色。色見本をクリックして別の色を選択します。
Indicator type	リンクインジケーターのタイプを選択します: **Static link** - インジケーターなし。 **Trigger** - トリガーベースのリンクインジケーターを使用できます。 **Item value** - アイテム値ベースのリンクインジケーターを使用できます。
Item	アイテムを選択します。このアイテムの値がリンクスタイルに影響します。 このフィールドは、*Indicator type* で *Item value* を選択した場合に使用できます。
Indicators	リンクインジケーターの一覧。 このフィールドは、*Indicator type* が *Trigger* または *Item value* に設定されている場合に使用できます。 *Trigger* に設定すると、トリガーをリンクインジケーターとして割り当てることができます。トリガーが *Problem* 状態になると、そのスタイル（選択した色と線種）がリンクに適用されます。 *Item value* に設定し、*Item* フィールドでアイテムを選択すると、アイテム値をリンクインジケーターとして割り当てることができます。アイテム値が指定したしきい値（数値アイテムの場合）に達するか、パターンに一致した場合（テキストアイテムの場合）、そのスタイル（選択した色と線種）がリンクに適用されます。

要素の移動とコピーペースト

選択した要素の 1 つをクリックし、マウスボタンを押したままカーソルを目的の場所に移動すると、選択した複数の要素を
 マップ内の別の場所に 移動することができます。

要素を選択し、マウスの右ボタンで選択した要素をクリックし、メニューから **Copy** を選択することで、1 つまたは
 複数の要素を コピーすることができます。

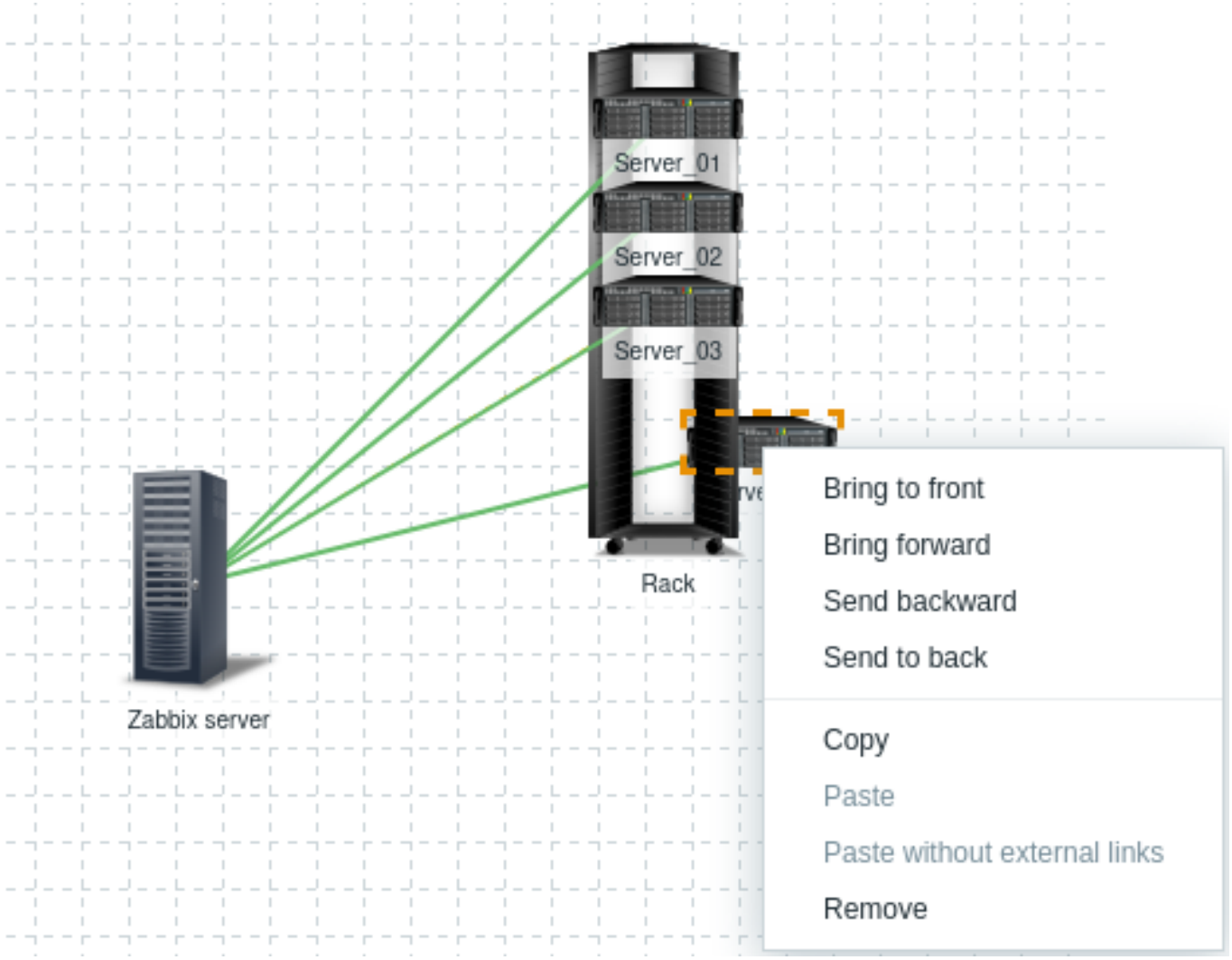


要素を貼り付けるには、マウスの右ボタンで地図上の領域をクリックし、メニューから *Paste* を選択します。*Paste without external links* を選択すると、選択した要素間のリンクのみを残して貼り付けられます。

コピー＆ペーストは、同じブラウザウィンドウ内で動作します。キーボードショートカットはサポートされていません。

要素の順序

ある要素を別の要素の前面に移動する（またはその逆を行う）には、要素をマウスの右ボタンでクリックし、*Bring forward*/*Bring to front* または *Send backward*/*Send to back* を選択します。

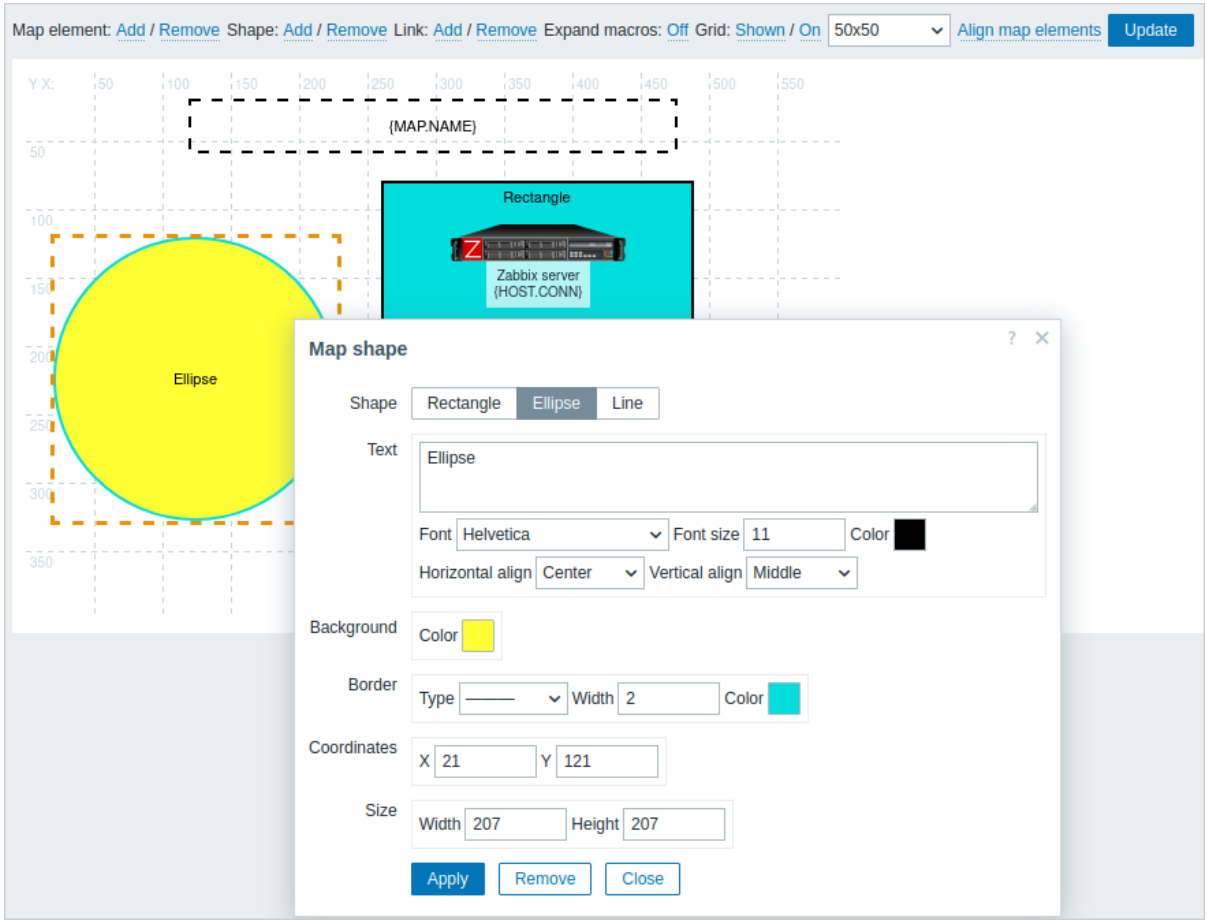


図形の追加

マップ要素に加えて、いくつかの図形を追加することもできます。図形はマップ要素ではなく、単なる視覚的な表現です。たとえば、長方形の図形は複数のホストをグループ化するための背景として使用できます。追加できる図形は、長方形と楕円です。

図形を追加するには、Shape の横にある *Add* をクリックします。新しい図形はマップの左上隅に表示されます。ドラッグ＆ドロップして任意の場所に配置してください。

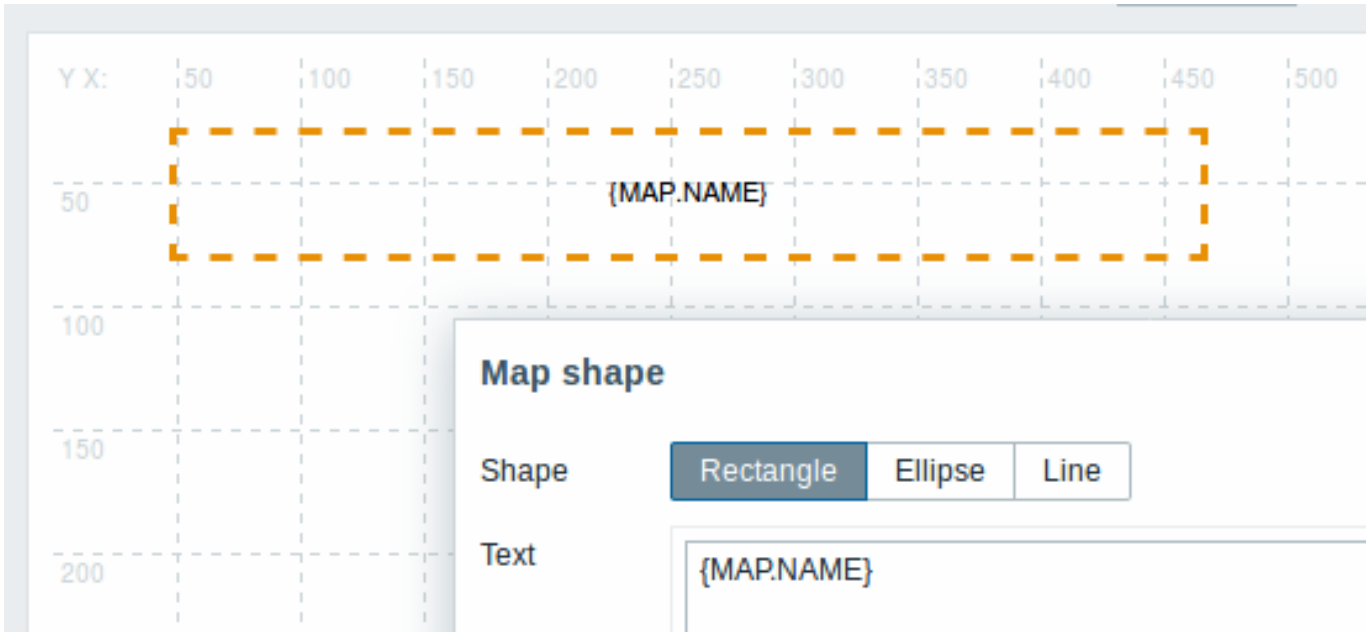
新しい図形はデフォルトの色で追加されます。図形をクリックするとフォームが表示され、図形の見た目のカスタマイズやテキストの追加などを行えます。



図形を選択するには、まず 1 つ選択し、その後 *Ctrl* キーを押したまま他の図形を選択します。複数の図形を選択すると、要素の場合と同様に、共通のプロパティを一括更新できます。

図形内にはテキストを追加できます。テキストでは式マクロがサポートされていますが、使用できるのは `avg`、`last`、`min`、`max` 関数のみで、パラメータには時間を指定する必要があります（例: `{?avg(/host/key,1h)}`）。

テキストのみを表示するには、図形の枠線を削除して図形を見えなくできます（*Border* フィールドで「None」を選択します）。たとえば、上のスクリーンショットに表示されている {MAP.NAME} マクロは、実際にはテキスト付きの長方形の図形であり、マクロをクリックするとそれが確認できます。



{MAP.NAME} は、マップの表示時に設定済みのマップ名に展開されます。

テキスト内でハイパーリンクを使用すると、マップ表示時にクリック可能になります。

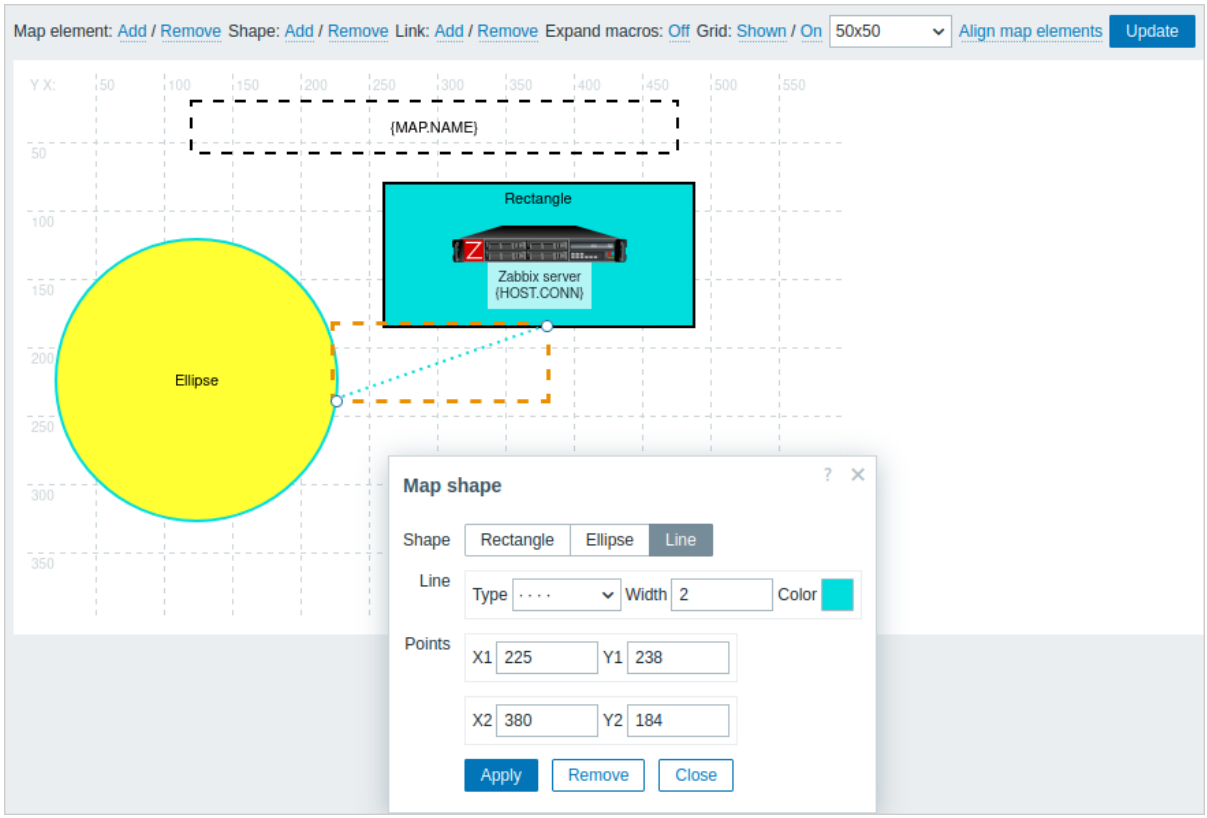
図形内のテキストの行折り返しは常に有効です。ただし、楕円内では、楕円を長方形とみなした場合と同じように行が折り返されます。単語の折り返しは実装されていないため、長い単語（図形内に収まらない単語）は折り返されず、マップ編集ページではマスクさ

れ、マップを含むその他のページでは切り詰められます。

線の追加

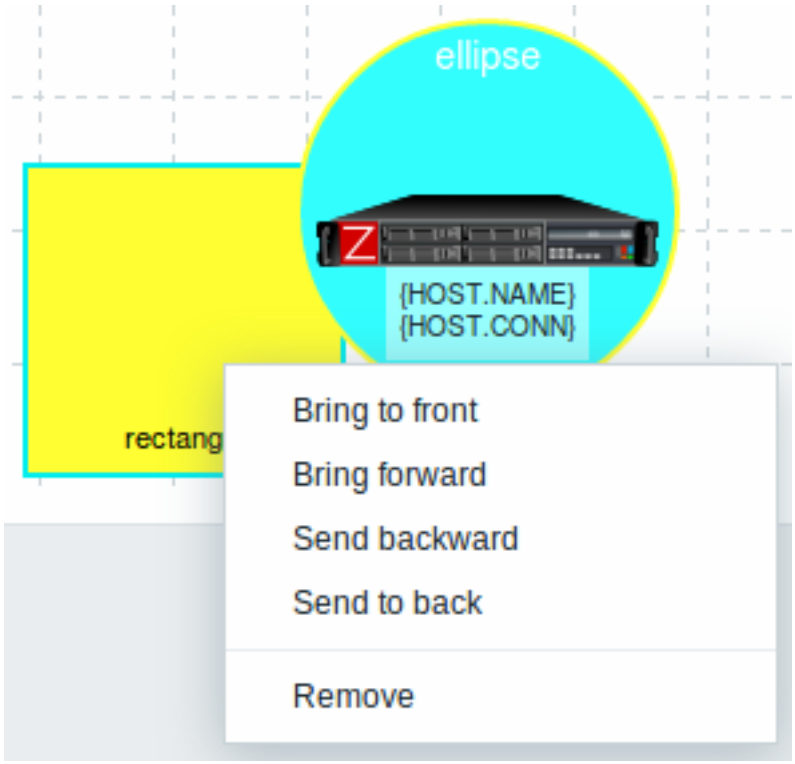
図形に加えて、いくつかの線を追加することもできます。線は、マップ内の要素や図形をリンクするために使用できます。

線を追加するには、Shape の横にある *Add* をクリックします。新しい図形がマップの左上隅に表示されます。それを選択し、編集フォームで *Line* をクリックして、図形を線に変更します。次に、線種、幅、色などの線のプロパティを調整します。



図形と線の順序

ある図形を別の図形の前面に表示する（またはその逆にする）には、図形をマウスの右ボタンでクリックして、マップ図形メニューを表示します。

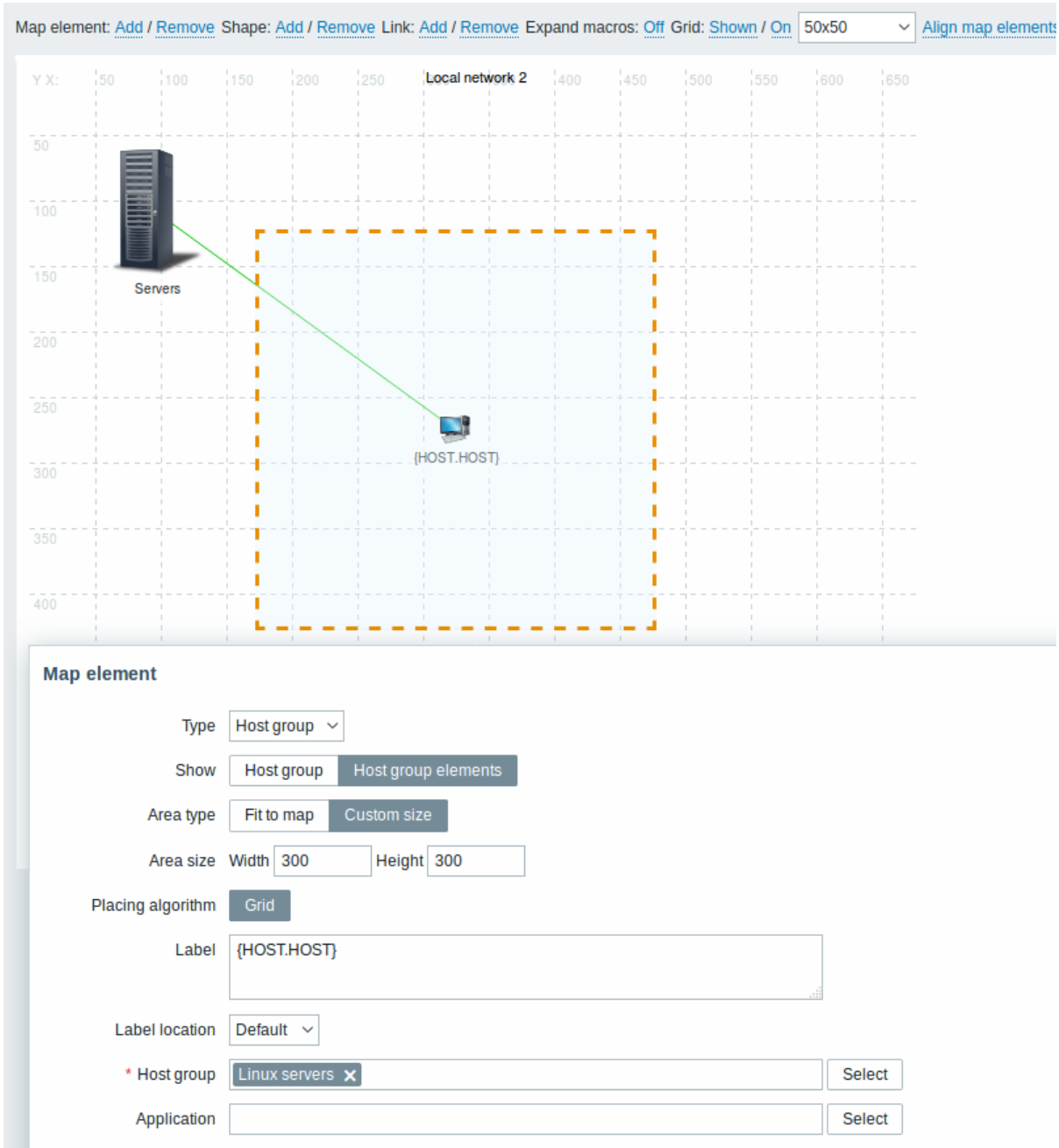


2 ホストグループの構成要素

概要

このセクションでは、ネットワークマップの設定時に、ホストグループタイプの要素を追加する方法について説明します。

設定



必須の入力フィールドには、赤いアスタリスクが付いています。

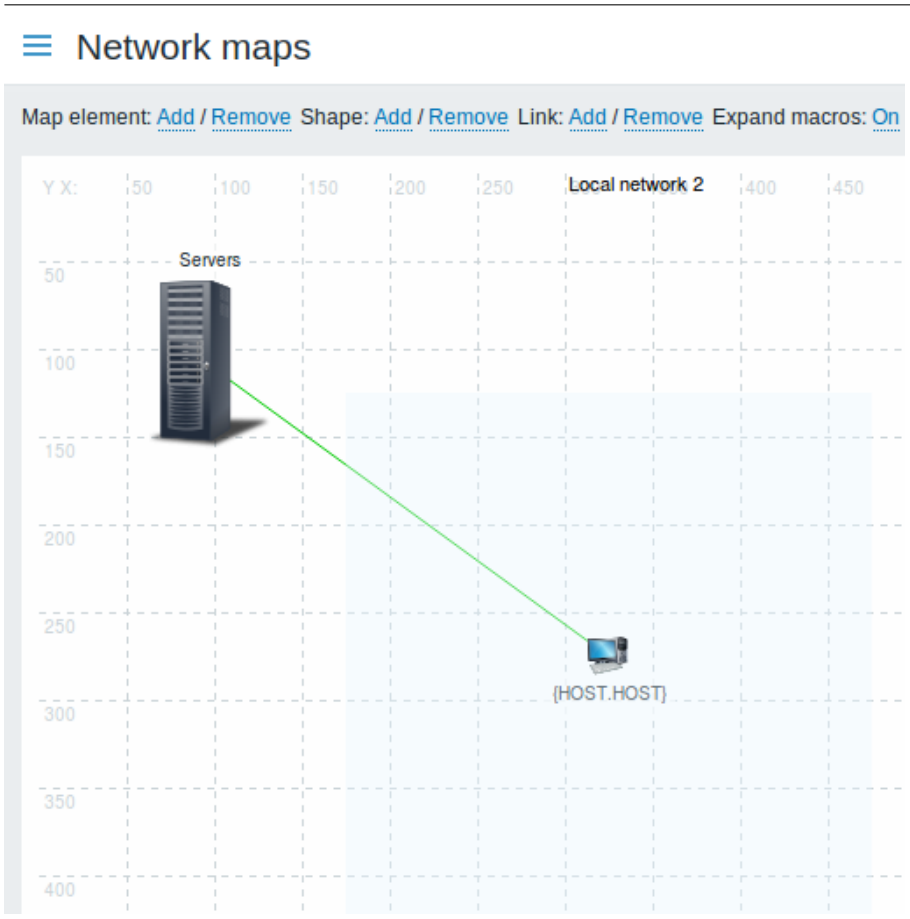
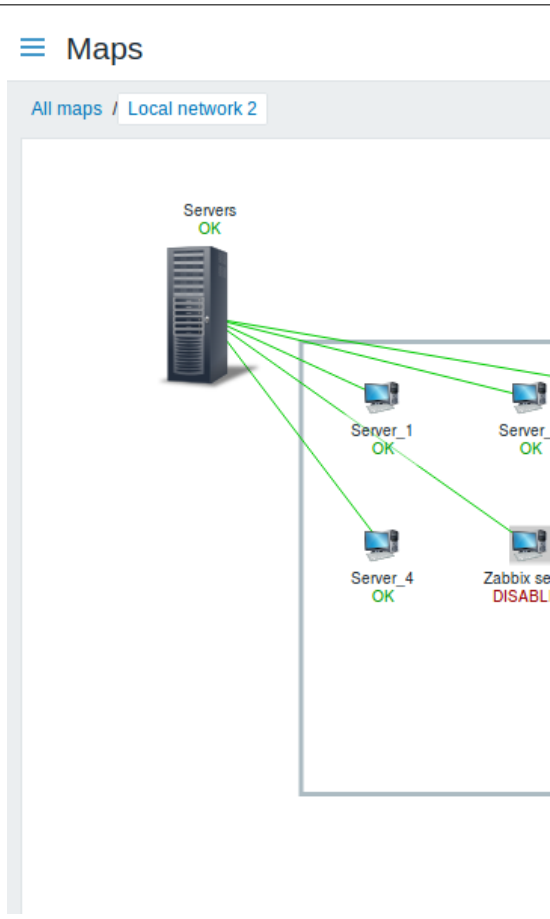
この表は、ホストグループ要素タイプに特有のパラメータで構成されています。

Parameter	Description
Type	要素の Type を選択します: **Host group** - 選択したグループに属するすべてのホストの、すべてのトリガーのステータスを表すアイコン。
Show	表示オプション: **Host group** - このオプションを選択すると、特定のホストグループに関する対応情報を表示する単一のアイコンとして表示されます。 **Host group elements** - このオプションを選択すると、特定のホストグループの各要素（ホスト）に関する対応情報を表示する複数のアイコンとして表示されます。

Parameter	Description
Area type	この設定は、*Show* が *Host group elements* に設定されている場合に使用できます: **Fit to map** - すべてのホストグループ要素がマップ内に均等に配置されます。 **Custom size** - 表示されるすべてのホストグループ要素に対するマップ領域を手動で設定します。
Area size	この設定は、*Area type* が *Custom size* に設定されている場合に使用できます: **Width** - マップ領域の幅を指定するピクセル単位の値。 **Height** - マップ領域の高さを指定するピクセル単位の値。
Placing algorithm	**Grid** - すべてのホストグループ要素を表示するための唯一の利用可能なオプションです。
Label	アイコンラベル。任意の文字列を指定できます。 ラベルには、マクロ および複数行の文字列を使用できます。 マップ要素のタイプが *Host group* の場合、特定のマクロを指定すると、各ホストに関する対応情報を表示するマップビューに影響します。たとえば、{HOST.IP} マクロを使用すると、マップ編集ビューにはマクロ {HOST.IP} 自体のみが表示されますが、マップビューには各ホスト固有の IP アドレスが含まれて表示されます。

ホストグループ要素の表示

このオプションは、ホストグループ要素の表示オプションが選択されている場合に利用できます。
表示オプションとして ホストグループ要素を選択すると、最初はホストグループに対して 1 つのアイコンだけが表示されます。
ただし、マップを保存してからマップ表示に移動すると、そのホストグループに属するすべての要素（ホスト）がマップに含まれていることがわかります。

マップ編集画面	マップ表示
	

{HOST.HOST} マクロの使用方法に注目してください。
マップ編集ではマクロ名は未解決のまま表示されますが、マップ表示ではホストの一意な名前がすべて表示されます。

3 リンクインジケーター

概要

ネットワークマップ内の要素間のリンクにインジケーターを割り当てることができます。

インジケーターは、トリガーまたはアイテムの値に基づいて設定できます。異なるリンクのスタイルと色を表示することが可能です。

- トリガーが障害状態になったとき。
- アイテムの値が次の場合:
  - しきい値に達した場合（数値アイテムの場合）。
  - 正規表現に一致した場合（テキストアイテムの場合）。

リンクを設定する際には、デフォルトのリンクタイプと色を設定します。リンクにインジケーターを割り当てることで、リンクのスタイルと色をトリガーのステータスまたはアイテムの値に応じて変化させることができます。

たとえば、割り当てられたトリガーのいずれかが障害状態になると、それを反映してリンクのスタイルと色が変化します。たとえば、デフォルトのリンクが緑色の線だったとします。この場合、トリガーが障害状態になると、リンクは太い赤線になることがあります(そのように定義している場合)。

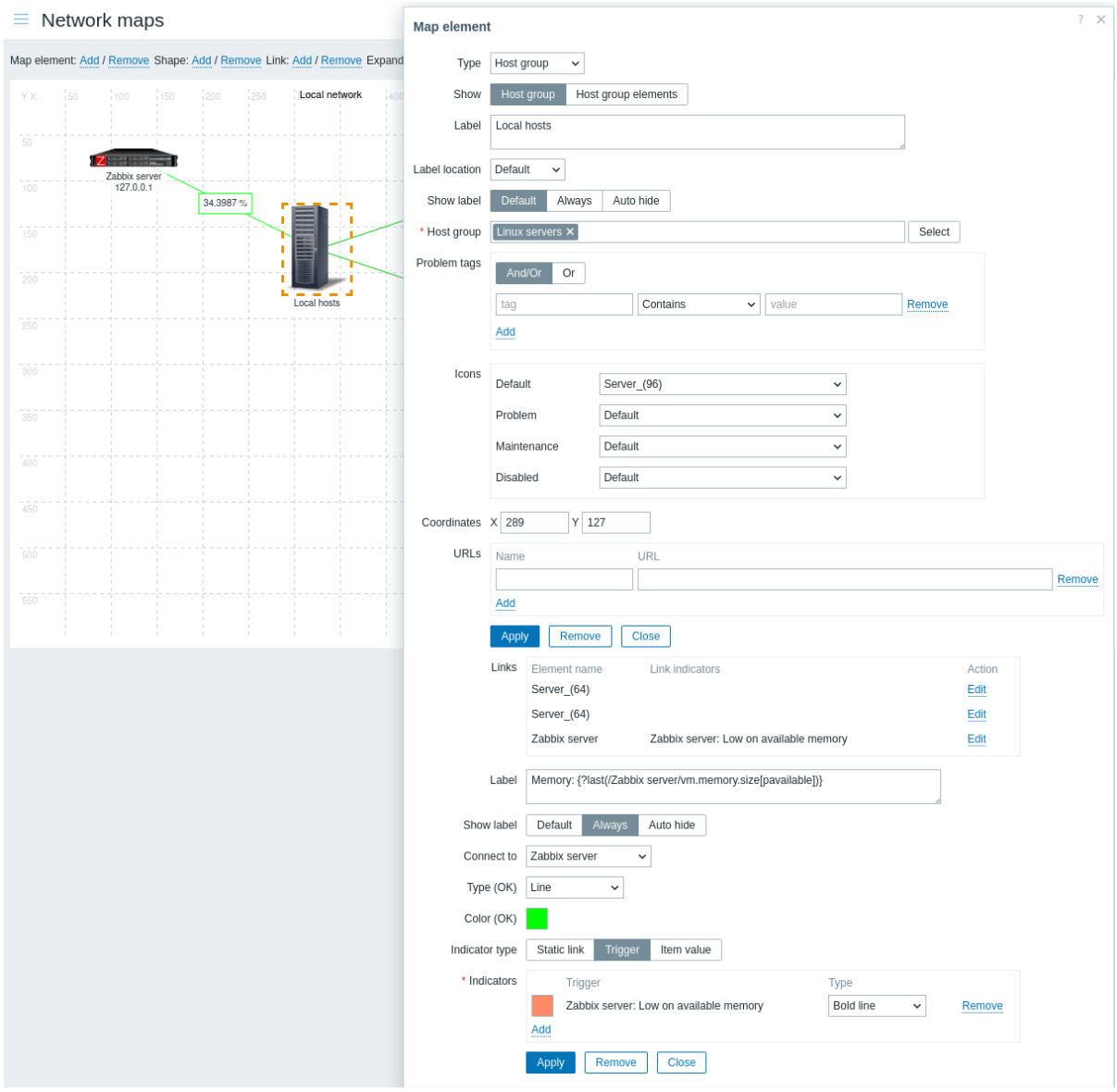
同様に、アイテムの値が指定したしきい値に達した場合、または指定した正規表現に一致した場合にも、リンクのスタイルにそれが反映されます。

設定

トリガー

トリガーをリンクインジケーターとして割り当てるには、次の手順を実行します。

- マップ要素を選択します
- 該当するリンクの *Links* セクションで *Edit* をクリックします
- インジケータータイプとして *Trigger* を選択します
- *Indicators* ブロックで *Add* をクリックし、1 つ以上のトリガーを選択します

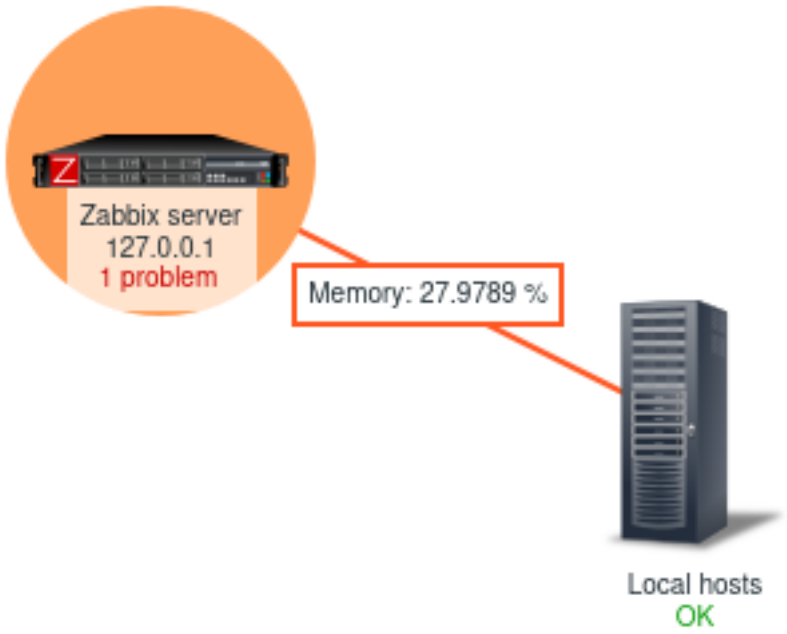


必須入力フィールドには赤いアスタリスクが付いています。

追加したトリガーは *Indicators* リストで確認できます。

各インジケーターのリンクタイプと色は、リストから直接設定できます。設定が完了したら、*Apply* をクリックし、フォームを閉じてから *Update* をクリックして、マップの変更を保存します。

Monitoring → *Maps* では、トリガーが障害状態になると、対応するリンクタイプと色が表示されます。



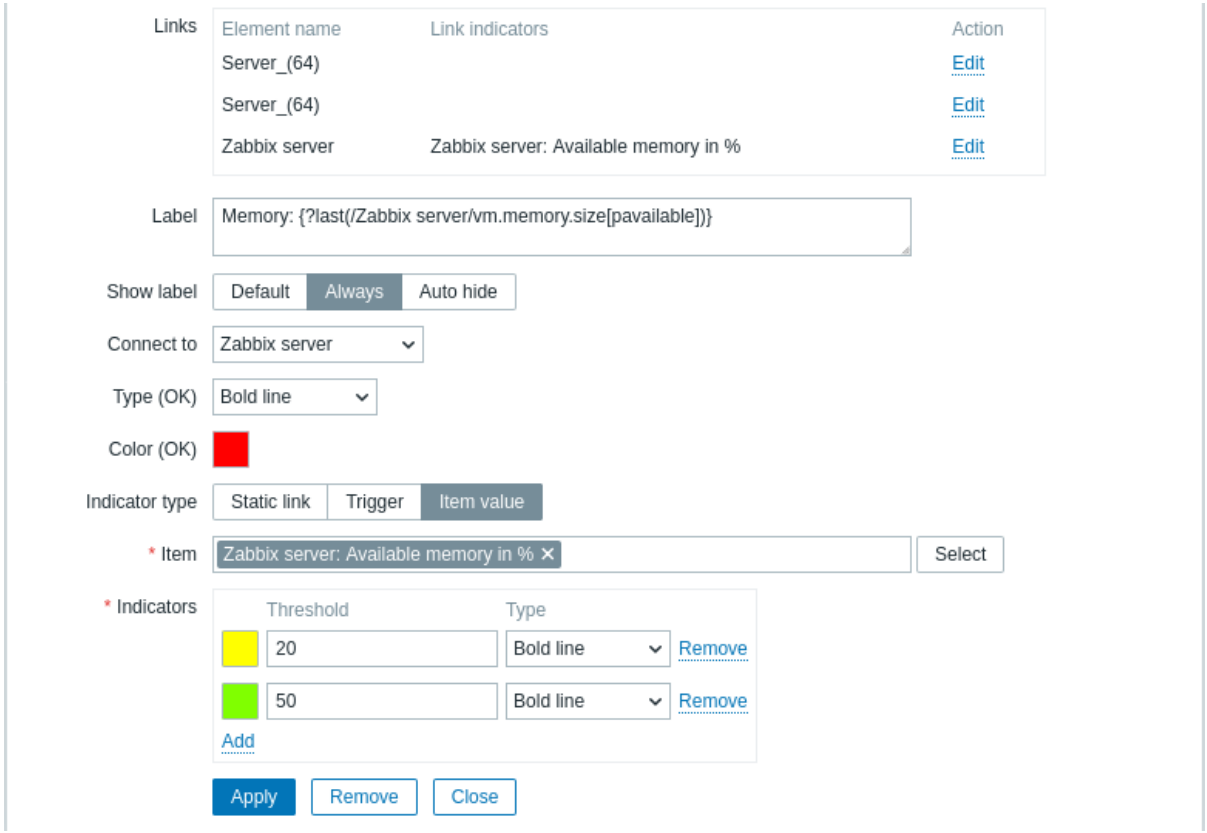
Note:
複数のトリガーが障害状態になった場合、最も深刻度の高い障害によってリンクのスタイルと色が決定されます。同じ深刻度の複数のトリガーが同じマップリンクに割り当てられている場合は、ID が最も小さいものが優先されます。さらに、次の点にも注意してください。

1. マップ設定の *Minimum trigger severity* および *Show suppressed problem* の設定は、どの障害が考慮されるかに影響します。
2. 複数の障害を持つトリガー（複数障害生成）の場合、各障害はトリガーの深刻度とは異なる深刻度を持つことがあり（手動で変更された場合）、異なるタグを持つことがあり（マクロによる）、抑制されることがあります。

アイテム値

アイテム値をリンクインジケーターとして割り当てるには、次の手順を実行します。

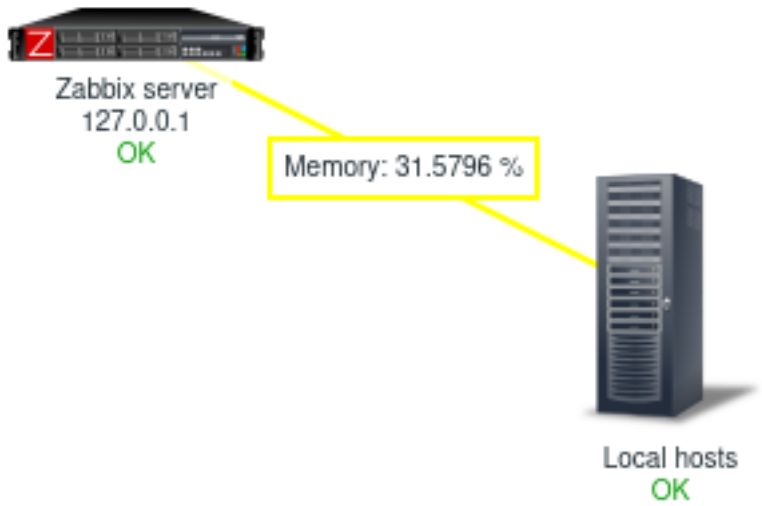
- マップ要素を選択します
- 該当するリンクの *Links* セクションで *Edit* をクリックします
- インジケータータイプとして *Item value* を選択します
- アイテムを選択します
- *Indicators* ブロックで、1 つ以上のアイテム値のしきい値またはパターンを追加します



追加したアイテムのしきい値/パターンは *Indicators* リストで確認できます。

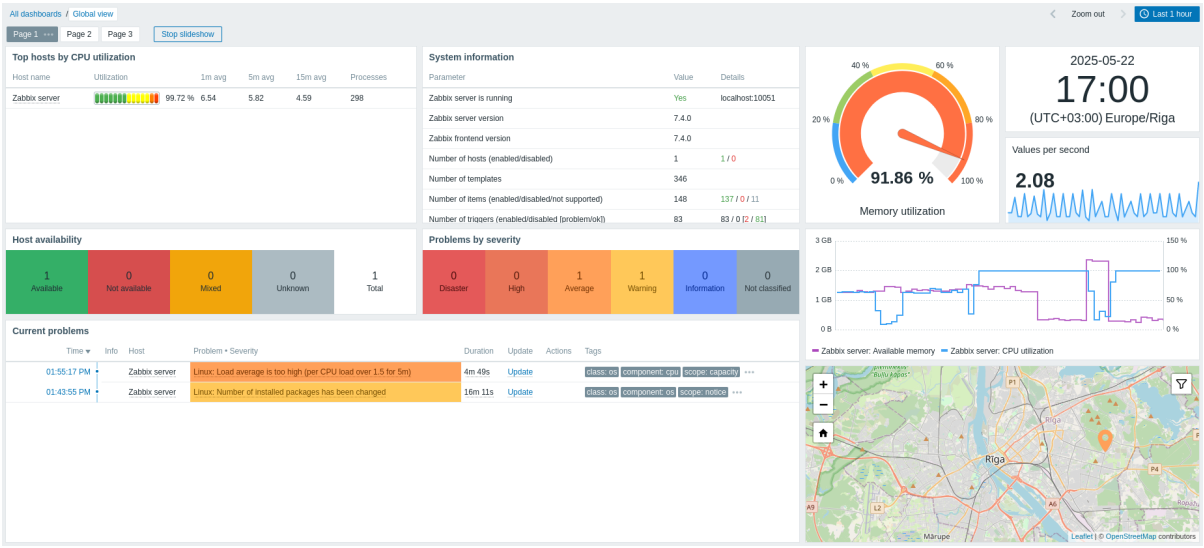
各インジケーターのリンクタイプと色は、リストから直接設定できます。設定が完了したら、*Apply* をクリックし、フォームを閉じてから *Update* をクリックして、マップの変更を保存します。

Monitoring → *Maps* では、アイテム値が設定したしきい値に達した場合（数値データ型の場合）、または正規表現パターンに一致した場合（テキストデータ型の場合）に、対応するリンクタイプと色が表示されます。



3 ダッシュボード

ダッシュボード - グローバルダッシュボードとホストダッシュボードの両方とも - 最新のグラフ、地図、スライドショーなどのウィジェットやツールを備えた強力な視覚化プラットフォームを提供します。



8 テンプレートとテンプレートグループ

概要

テンプレートの使用は、作業負荷を軽減し、Zabbix の設定を効率化するための非常に優れた方法です。テンプレートとは、複数のホストに簡単に適用できるエンティティのセットです。

エンティティには、次のものがあります。

- アイテム
- トリガー
- グラフ
- ダッシュボード
- ローレベルディスカバリルール
- Web シナリオ

実際の環境では、多くのホストが同一またはかなり似通っているため、1 つのホスト用に作成したエンティティのセット（アイテム、トリガー、グラフなど）は、他の多くのホストでも役立つことが自然に考えられます。もちろん、それらを新しいホストごとにコピーすることもできますが、それでは手作業が非常に多くなってしまいます。代わりに、テンプレートを使えば、それらを 1 つのテンプレートにまとめてコピーし、そのテンプレートを必要な数のホストに適用できます。

テンプレートがホストにリンクされると、そのテンプレートのすべてのエンティティ（アイテム、トリガー、グラフなど）がホストに追加されます。テンプレートは、各ホストに対して個別に直接割り当てられます（ホストグループには割り当てられません）。

テンプレートは、特定のサービスやアプリケーション（Apache、MySQL、PostgreSQL、Postfix など）のエンティティをグループ化するためによく使用され、それらのサービスを実行しているホストに適用されます。

テンプレートを使用するもう 1 つの利点は、すべてのホストに対して何らかの変更が必要になった場合です。テンプレートレベルで一度変更を行えば、その変更はリンクされているすべてのホストに反映されます。

テンプレートは、テンプレートグループで整理されます。

テンプレートの作成と設定に進んでください。

9 すぐに使えるテンプレート

概要

Zabbix は、監視対象の設定を簡素化し、迅速化するために、事前設定済みのテンプレートを拡充し続けています。

すべての標準搭載テンプレートは、データ収集 > テンプレート で利用できます。

Zabbix のアップグレード時には、カスタム変更が上書きされるのを防ぐため、既存のテンプレートは自動的には更新されません。既存のテンプレートをアップグレードする場合、または新しいテンプレートを追加する場合は、テンプレートのアップグレードを参照してください。

特定のテンプレートタイプや運用要件に関する情報にアクセスするには、サイドバーを使用してください。

関連情報:

- テンプレートのリンク
- テンプレートに関する既知の問題

テンプレートのアップグレード

Zabbix のアップグレード後に、単一のテンプレートをアップグレードする、または新しいテンプレートを追加するには、次のようにします。

1. Zabbix Git リポジトリに移動します。
2. ブランチセレクター（*Source* ラベルの下）で、使用している Zabbix のバージョンに一致するブランチを選択します（例: Zabbix 7.4 の場合は *release/7.4*）。
3. 追加またはアップグレードしたいテンプレートを開きます（例: MySQL by Zabbix agent）。
4. テンプレートファイルの内容（例: `template_db_mysql_agent.yaml`）をコピーし、ローカルファイルとして保存します。
5. Zabbix Web インターフェースで、*Data collection* > *Templates* に移動し、ローカルファイルをインポートします。

すべてのテンプレートをアップグレードし、新しいテンプレートをまとめて追加することもできます。

1. 使用している Web サーバーに応じてテンプレートバンドルをダウンロードします: `templates_1M_7.4.7z`（Nginx）または`templates_2M_7.4.7z`（Apache）。
2. バンドルを展開します。これには YAML ファイル（それぞれに複数のテンプレートを含む）と、各 YAML ファイルに含まれるテンプレートを一覧表示するインデックス（`index-1M_74.md`）が含まれています。
3. Zabbix Web インターフェースで、*Data collection* > *Templates* に移動し、YAML ファイルを 1 つずつインポートします。

または、`import_templates.sh`スクリプトを使用して、すべての YAML ファイルを一度にインポートすることもできます。
このスクリプトには`curl`および`jq`ユーティリティが必要で、`configuration.import` Zabbix API メソッドを使用します。
スクリプトの実行時には、YAML ファイルを含むディレクトリと、Zabbix API のURLおよび認証トークンを指定してください。

> **Warning:**
> `import_templates.sh` スクリプトは、既存のすべてのテンプレートを上書きし、テンプレートの変更を確認したり元に戻したりするオプションはありません。

例:

```
./import_templates.sh /tmp/templates_2M_7.4 https://example.com/zabbix/api_jsonrpc.php d8d6b5c78ee2a8333db
```

Host Wizard とのテンプレート互換性

現在Host Wizardと互換性のある、すぐに利用可能なテンプレートの完全な一覧については、`hw_compatible_templates.txt`を参照してください。

カスタムテンプレートを Host Wizard と互換性を持たせることもできます。

1. Zabbix の Web インターフェースで、*Data collection* > *Templates* に移動し、カスタムテンプレートをエクスポートします。
2. エクスポートしたファイルを編集します。
   - Host Wizard との互換性を示すために、`wizard_ready` テンプレート要素を追加します。
   - 必要に応じて、`readme` または `config` テンプレート要素を追加します。
3. Zabbix の Web インターフェースで、*Data collection* > *Templates* に移動し、更新したテンプレートファイルをインポートします。

1 Zabbix エージェントテンプレートの操作

Zabbix エージェントでメトリクスを収集するテンプレートが正しく動作するようにするための手順:

1. Zabbix エージェントがホストにインストールされていることを確認します。アクティブチェックの場合は、Zabbix サーバー/プロキシのアドレスがエージェントの設定ファイルの'ServerActive' パラメータにも追加されていることを確認してください。
2. テンプレートを対象のホストにリンクします（テンプレートが Zabbix インストール環境で利用できない場合は、まずテンプレートをインポートする必要がある場合があります）。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスを設定し、Zabbix とデータを共有できるようにします。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックするとアクセス可能）で確認できます。

以下のテンプレートを利用できます:

- Apache by Zabbix agent

- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

2 Zabbix agent 2 テンプレートの操作

Zabbix agent 2 でメトリクスを収集するテンプレートが正しく動作することを確認する手順:

1. エージェント 2 がホストにインストールされており、インストールされているバージョンに必要なプラグインが含まれていることを確認してください。場合によっては、先にエージェント 2 をアップグレードする必要があります。
2. テンプレートを対象のホストにリンクしてください（テンプレートがお使いの Zabbix インストール環境で利用できない場合は、先にテンプレートをインポートする必要があります）。
3. 必要に応じて、テンプレートマクロの値を調整してください。ユーザーマクロを使用して設定パラメータを上書きできることに注意してください。
4. 監視対象のインスタンスを設定し、Zabbix とデータを共有できるようにしてください。

> **Attention:**
> Zabbix agent 2 テンプレートはプラグインと連携して動作します。基本的な設定はユーザーマクロを調整するだけで行えますが、より詳細なカスタマイズはプラグイン自体を設定することで実現できます。
> たとえば、プラグインが名前付きセッションをサポートしている場合、設定ファイル内で各エンティティに対してそれぞれ独自の URI、ユーザー名、パスワードを持つ名前付きセッションを指定することで、同種の複数のエンティティ（例: MySQL1 および MySQL2）を監視できます。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックするとアクセス可能）で確認できます。

以下のテンプレートが利用可能です:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

3 HTTP テンプレートの操作

HTTP エージェントでメトリクスを収集するテンプレートが正しく動作するようにするための手順:

1. Zabbix でホストを作成し、監視対象の IP アドレスまたは DNS 名をメインインターフェースとして指定します。これは、テンプレートのアイテム内で {HOST.CONN} マクロが正しく解決されるために必要です。
2. 手順 1 で作成したホストにテンプレートをリンクします（テンプレートがお使いの Zabbix インストール環境で利用できない場合

は、先にテンプレートをインポートする必要がある場合があります)。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスを設定し、Zabbix とデータを共有できるようにします。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックするとアクセスできます）で確認できます。

以下のテンプレートを利用できます:

- Acronis Cyber Protect Cloud by HTTP
- Apache by HTTP
- Asterisk by HTTP
- AWS by HTTP
- AWS Cost Explorer by HTTP
- AWS EC2 by HTTP
- AWS ECS Cluster by HTTP
- AWS ECS Serverless Cluster by HTTP
- AWS ELB Application Load Balancer by HTTP
- AWS ELB Network Load Balancer by HTTP
- AWS Lambda by HTTP
- AWS RDS instance by HTTP
- AWS S3 bucket by HTTP
- Azure by HTTP
- Cisco Meraki organization by HTTP
- Cisco SD-WAN by HTTP
- Cisco Secure Firewall Threat Defense by HTTP
- ClickHouse by HTTP
- Cloudflare by HTTP
- CockroachDB by HTTP
- Control-M enterprise manager by HTTP
- Control-M server by HTTP
- DELL PowerEdge R720 by HTTP
- DELL PowerEdge R740 by HTTP
- DELL PowerEdge R820 by HTTP
- DELL PowerEdge R840 by HTTP
- Elasticsearch Cluster by HTTP
- Envoy Proxy by HTTP
- Etcd by HTTP
- FortiGate by HTTP
- GitHub repository by HTTP
- GitLab by HTTP
- Google Cloud Platform (GCP) by HTTP
- Hadoop by HTTP
- HAProxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP
- HashiCorp Nomad by HTTP
- HashiCorp Vault by HTTP
- Hikvision camera by HTTP
- HPE iLO by HTTP
- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP
- HPE Synergy by HTTP
- InfluxDB by HTTP
- Jenkins by HTTP
- Kubernetes API server by HTTP
- Kubernetes cluster state by HTTP
- Kubernetes Controller manager by HTTP
- Kubernetes kubelet by HTTP
- Kubernetes nodes by HTTP

- Kubernetes Scheduler by HTTP
- MantisBT by HTTP
- Microsoft 365 reports by HTTP
- Microsoft SharePoint by HTTP
- NetApp AFF A700 by HTTP
- Nextcloud by HTTP
- NGINX by HTTP
- NGINX Plus by HTTP
- Nutanix Prism Element by HTTP
- OpenStack by HTTP
- OpenWeatherMap by HTTP
- Oracle Cloud by HTTP
- Palo Alto PA-440 by HTTP
- PHP-FPM by HTTP
- Proxmox VE by HTTP
- Pure Storage FlashArray v1 and v2 by HTTP
- RabbitMQ cluster by HTTP
- Ribbon SBC Edge by HTTP
- Ribbon SBC SWe CE by HTTP
- Ribbon SBC SWe Core by HTTP
- TiDB by HTTP
- TiDB PD by HTTP
- TiDB TiKV by HTTP
- Travis CI by HTTP
- Veeam Backup Enterprise Manager by HTTP
- Veeam Backup and Replication by HTTP
- VeloCloud SD-WAN by HTTP
- VeloCloud SD-WAN Edge by HTTP
- YugabyteDB by HTTP
- ZooKeeper by HTTP

4 IPMI テンプレートの操作

IPMI テンプレートには、特別な設定は必要ありません。監視を開始するには、対象のホストにテンプレートをリンクしてください(テンプレートがお使いの Zabbix インストール環境で利用できない場合は、先にテンプレートをインポートする必要がある場合があります)。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックするとアクセスできます）で確認できます。

利用可能なテンプレート:

- IPMI によるシャーシ

5 JMX テンプレートの操作

JMX によってメトリクスを収集するテンプレートが正しく動作することを確認する手順:

1. Zabbix Java ゲートウェイ がインストールされ、適切に設定されていることを確認します。
2. テンプレートを対象のホストにリンクします。ホストには JMX インターフェースが設定されている必要があります。
テンプレートがお使いの Zabbix インストール環境で利用できない場合は、まずテンプレートをインポートする必要がある場合があります。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスを設定し、Zabbix とデータを共有できるようにします。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックするとアクセス可能）で確認できます。

以下のテンプレートを利用できます:

- Apache ActiveMQ by JMX
- Apache Cassandra by JMX
- Apache Kafka by JMX
- Apache Tomcat by JMX

- GridGain by JMX
- Ignite by JMX
- Jira Data Center by JMX
- WildFly Domain by JMX
- WildFly Server by JMX

6 ODBC テンプレートの操作

ODBC 監視 を介してメトリクスを収集するテンプレートが正しく動作するようにする手順:

1. 必要な ODBC ドライバーが Zabbix サーバーまたはプロキシにインストールされていることを確認します。
2. テンプレートを対象のホストにリンクします（テンプレートがお使いの Zabbix インストール環境で利用できない場合は、まずテンプレートをインポートする必要がある場合があります）。
3. 必要に応じて、テンプレートマクロの値を調整します。
4. 監視対象のインスタンスを設定し、Zabbix とデータを共有できるようにします。

マクロ、アイテム、トリガーの完全な一覧を含むテンプレートの詳細な説明は、テンプレートの README ファイル（テンプレート名をクリックするとアクセスできます）で確認できます。

以下のテンプレートを利用できます:

- ODBC による MSSQL
- ODBC による MySQL
- ODBC による Oracle
- ODBC による PostgreSQL

7 ネットワークデバイス用の標準テンプレート

概要

スイッチやルーターなどのネットワークデバイスの監視を提供するために、いわゆる 2 つのモデルを作成しました。1 つはネットワークデバイス自体（基本的にはそのシャーシ）用、もう 1 つはネットワークインターフェース用です。

多くのネットワークデバイスファミリー向けのテンプレートが提供されています。すべてのテンプレートは、（デバイスからこれらのアイテムを取得できる場合に）以下を対象としています。

- シャーシ障害監視（電源、ファン、温度、全体ステータス）
- シャーシ性能監視（CPU およびメモリのアイテム）
- シャーシインベントリ収集（シリアル番号、モデル名、ファームウェアバージョン）
- IF-MIB および EtherLike-MIB を使用したネットワークインターフェース監視（インターフェースステータス、インターフェーストラフィック負荷、Ethernet の duplex ステータス）

テンプレートがご使用の Zabbix インストール環境で利用できない場合は、まずテンプレートをインポートする必要があるかもしれません。

新しい標準搭載テンプレートをインポートする場合は、`@Network interfaces for discovery` グローバル正規表現も次のように更新することをお勧めします。

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

これにより、ほとんどのシステムでループバックおよび null インターフェースを除外できます。

デバイス

テンプレートが利用可能なデバイスファミリーの一覧:

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Aruba CX スイッチシリーズ	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Certified
Brocade FC SNMP	Brocade	Brocade FC スイッチ	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Performance Fault
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certified
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Performance Fault
Check Point Next Generation Firewall by SNMP	Check Point	次世代ファイアウォール	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certified
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certified
Cisco Catalyst 3750<device model> SNMP	Cisco	Cisco Catalyst 3750	Cisco Catalyst 3750V2-24FS, Cisco Catalyst 3750V2-24PS, Cisco Catalyst 3750V2-24TS, Cisco Catalyst SNMP, Cisco Catalyst SNMP		CISCO-MEMORY-POOL-MIB, IF-MIB, EtherLike-MIB, SNMPv2-MIB, CISCO-PROCESS-MIB, CISCO-ENVMON-MIB, ENTITY-MIB	Certified
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T and 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certified
Cisco IOS prior to 12.0_3_T SNMP	Cisco	Cisco IOS 12.0 3 T	-	IOS	OLD-CISCO-CPU-MIB,CISCO-MEMORY-POOL-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX スイッチ	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certified
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Performance Fault Interfaces
Dell Force S-Series SNMP	Dell	Dell Force S シリーズ	S4810		F10-S-SERIES-CHASSIS-MIB	Certified
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certified
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Performance Inventory
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certified
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Certified
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado with V5 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V5/5500V5/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certified
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infini-band デバイス	Infiniband 12300		ICS-CHASSIS-MIB	Fault Inventory
Juniper SNMP	Juniper	MX,SRX,EX モデル	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certified
Juniper MX NETCONF	Juniper	MX モデル	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	該当なし *(SSH 上の NETCONF を使用)*	Certified
Juniper MX SNMP	Juniper	MX モデル	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Certified
Mellanox SNMP	Mellanox	Mellanox Infini-band デバイス	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certified
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Router (CCR シリーズ)	MikroTik CCR1009-7G-1C-1S+, MikroTik CCR1009-7G-1C-1S+PC, MikroTik CCR1009-7G-1C-PC, MikroTik CCR1016-12G, MikroTik CCR1016-12S-1S+, MikroTik CCR1036-12G-4S-EM, MikroTik CCR1036-12G-4S, MikroTik CCR1036-8G-2S+, MikroTik CCR1036-8G-2S+EM, MikroTik CCR1072-1G-8S+, MikroTik CCR2004-16G-2S+, MikroTik CCR2004-1G-12S+2XS 向けの個別専用テンプレートが利用可能	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switch (CRS シリーズ)	MikroTik CRS106-1C-5S, MikroTik CRS109-8G-1S-2HnD-IN, MikroTik CRS112-8G-4S-IN, MikroTik CRS112-8P-4S-IN, MikroTik CRS125-24G-1S-2HnD-IN, MikroTik CRS212-1G-10S-1S+IN, MikroTik CRS305-1G-4S+IN, MikroTik CRS309-1G-8S+IN, MikroTik CRS312-4C+8XG-RM, MikroTik CRS317-1G-16S+RM, MikroTik CRS326-24G-2S+IN, MikroTik CRS326-24G-2S+RM, MikroTik CRS326-24S+2Q+RM, MikroTik CRS328-24P-4S+RM, MikroTik CRS328-4C-20S-4S+RM, MikroTik CRS354-48G-4S+2Q+RM, MikroTik CRS354-48P-4S+2Q+RM 向けの個別専用テンプレートが利用可能	RouterOS/SwitchOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switch (CSS シリーズ)	MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN 向けの個別専用テンプレートが利用可能	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S 向けの個別専用テンプレートが利用可能	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R 向けの個別専用テンプレートが利用可能	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	MikroTik PowerBox, MikroTik PowerBox Pro 向けの個別専用テンプレートが利用可能	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik RB<device model> SNMP	MikroTik	MikroTik RB シリーズルーター	MikroTik RB1100AHx4, MikroTik RB1100AHx4 Dude Edition, MikroTik RB2011iL-IN, MikroTik RB2011iL-RM, MikroTik RB2011iLS-IN, MikroTik RB2011UiAS-IN, MikroTik RB2011UiAS-RM, MikroTik RB260GS, MikroTik RB3011UiAS-RM, MikroTik RB4011iGS+RM, MikroTik RB5009UG+S+IN 向けの個別専用テンプレートが利用可能	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
MikroTik SNMP	MikroTik	MikroTik RouterOS デバイス	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certified
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Fault Inventory
QTech QSW SNMP	QTech	Qtech デバイス	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Performance Inventory

テンプレート名	ベンダー	デバイスファミリー	既知のモデル	OS	使用される MIB	タグ
Stormshield SNS by SNMP	Stormshield	Stormshield Network Security (SNS) ファイアウォール	SN3100	Stormshield SNS	HOST-RESOURCES-MIB, UCD-SNMP-MIB, STORMSHIELD-ASQ-STATS-MIB, STORMSHIELD-AUTOUPDATE-MIB, STORMSHIELD-HA-MIB, STORMSHIELD-PROPERTY-MIB, STORMSHIELD-HEALTH-MONITOR-MIB, STORMSHIELD-IF-MIB, STORMSHIELD-SYSTEM-MONITOR-MIB, STORMSHIELD-IPSEC-STATS-MIB	Certified
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Performance Inventory
Ubiquiti AirOS SNMP	Ubiquiti	Ubiquiti AirOS ワイヤレスデバイス	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Performance
Vyatta Virtual Router by SNMP	Ciena	Vyatta	Vyatta Virtual Router 1908e	Vyatta 1908e	SNMPv2-MIB, HOST-RESOURCES-MIB, UCD-SNMP-MIB, IF-MIB, DISMAN-EVENT-MIB	Performance Inventory

テンプレート設計

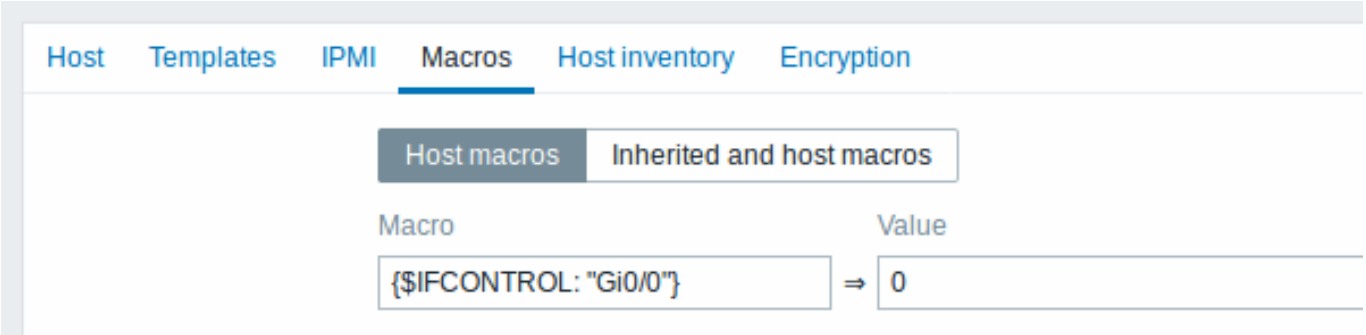
テンプレートは、以下の点を考慮して設計されています。

- ユーザーがトリガーを調整できるように、ユーザーマクロを可能な限り使用しています。
- サポートされないアイテムの数を最小限に抑えるために、ローレベルディスカバリを可能な限り使用しています。
- すべてのテンプレートは Template ICMP Ping に依存しているため、すべてのデバイスは ICMP によってもチェックされます。
- アイテムでは MIB を使用していません。SNMP OID はアイテムおよびローレベルディスカバリで使用されています。そのため、テンプレートを動作させるために Zabbix に MIB を読み込む必要はありません。
- ディスカバリ時には、ループバックネットワークインターフェース、および ifAdminStatus = down(2) のインターフェースが除外されます。
- 可能な場合は、IF-MIB::ifXTable の 64bit カウンターを使用しています。サポートされていない場合は、代わりにデフォルトの 32bit カウンターを使用します。

ディスカバリされたすべてのネットワークインターフェースには、その動作状態（リンク）を監視するトリガーがあります。例:

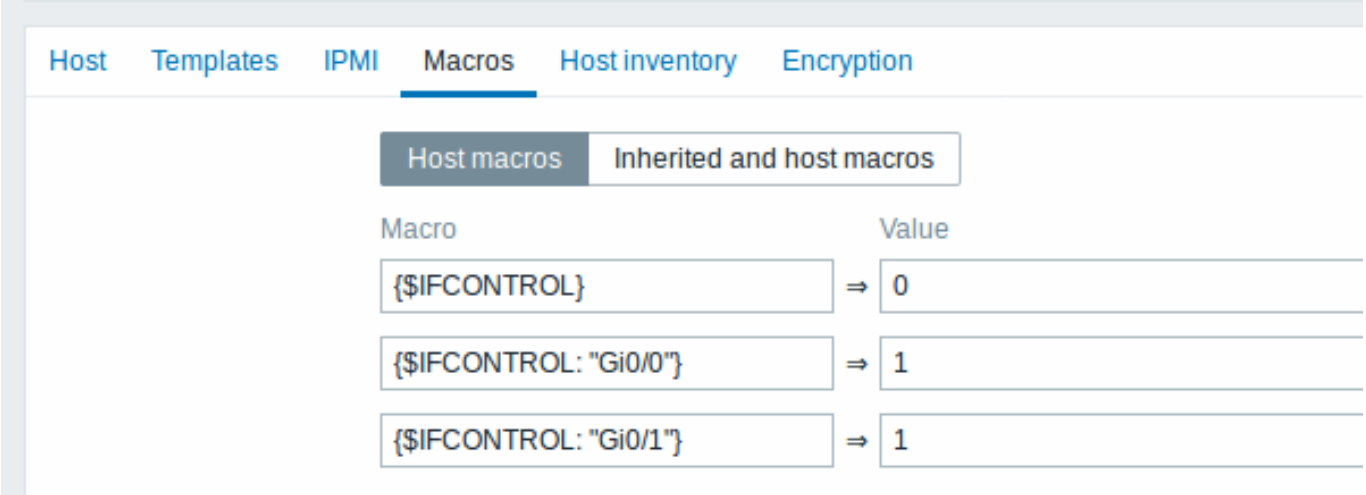
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 特定のインターフェースについてこの状態を監視したくない場合は、値 0 のコンテキスト付きユーザーマクロを作成してください。例:



ここで Gi0/0 は {#IFNAME} です。これにより、この特定のインターフェースではトリガーは使用されなくなります。

- また、すべてのトリガーが発生しないようにデフォルトの動作を変更し、このトリガーをアップリンクのような限られた数のインターフェースに対してのみ有効化することもできます。



タグ

- Performance - デバイスファミリーの MIB は、CPU とメモリ item を監視する方法を提供します。
- Fault - デバイスファミリーの MIB は、少なくとも 1 つの温度センサを監視する方法を提供します。
- Inventory - デバイスファミリーの MIB は、少なくともデバイスのシリアル番号とモデル名を収集する方法を提供します。
- Certified - 上記 3 つの主要カテゴリを全てカバーします。

8 VMware テンプレートの操作

概要

Zabbix には、VMware vCenter および ESXi ハイパーバイザー向けに設計された、仮想マシン監視 用のすぐに使える各種テンプレートが含まれています。利用可能なテンプレートは、2 つの別個のテンプレートセットに含まれています。

- VMware - 対応するマクロに UUID データを使用します
- VMware FQDN - 対応するマクロに FQDN データを使用します

VMware vCenter または ESXi ハイパーバイザーの監視用に設計されたこれらのテンプレートには、仮想環境を監視するためのさまざまな組み込みチェックに加えて、事前設定済みのローレベルディスカバリルールが含まれています。

Note:
VMware FQDN テンプレートを正しく機能させるには、監視対象の各仮想マシンが、FQDN のルールに準拠した一意の OS 名を持っている必要があります。さらに、VMware Tools/Open Virtual Machine Tools をすべてのマシンにインストールする必要があります。これらの前提条件が満たされている場合は、*VMware FQDN* テンプレートの使用を推奨します。*VMware FQDN* テンプレートは、カスタムインターフェースを持つホストを作成できる機能の導入に伴い、Zabbix 5.2 以降で利用可能です。

 従来の *VMware* テンプレートも利用可能であり、FQDN の要件を満たせない場合に使用できます。ただし、*VMware* テンプレートには既知の問題があります。検出された仮想マシン用のホストは、vCenter に保存されている名前（たとえば”VM1”、”VM2” など）で作成されます。これらのホストに Zabbix エージェントがインストールされており、アクティブ Zabbix エージェントの自動登録が有効になっている場合、自動登録プロセスは、起動時に登録されたホスト名（たとえば”vm1.example.com”、”vm2.example.com” など）を読み取ります。その結果、既存の仮想マシンに対して新しいホストが作成される可能性があり（名前の一致が見つからないため）、異なる名前を持つ重複ホストが発生します。

以下の点に注意してください。

- テンプレート *VMware Hypervisor* は、ホストに手動でリンクすることも、ディスカバリで使用することもできます。
- テンプレート *VMware Guest* は、ホストに手動でリンクしてはいけません。使用できるのはディスカバリのみです。

ホストマクロの設定

VMware シンプルチェックを使用するには、ホストで以下のユーザーマクロを定義する必要があります。

- {$VMWARE.URL} - VMware サービス（vCenter または ESXi ハイパーバイザー）の SDK URL（https://servername/sdk）
- {$VMWARE.USERNAME} - VMware サービスのユーザー名
- {$VMWARE.PASSWORD} - VMware サービスの {$VMWARE.USERNAME} ユーザーのパスワード

10 イベント時の通知

概要

いくつかのアイテムとトリガーを構成し、トリガーの状態変更の結果としていくつかのイベントが発生していると仮定して、いくつかのアクションを検討します。

そもそも、トリガーやイベントのリストを常に眺めたくはありません。何か重大な、障害などが発生した場合に通知を受け取る方がはるかに良いでしょう。また、障害が発生した場合には、関係者全員に周知したいと考えております。

そのため、通知の送信は Zabbix が提供する主要なアクションの 1 つです。特定のイベントが発生したときに、誰に、いつ、通知するかを定義できます。

Zabbix から通知を送受信できるようにするには、次のことを行う必要があります。

- メディアの定義
- 定義されたメディアにメッセージを送信するアクションの設定

アクションは条件と操作で構成されます。基本的に条件が満たされた場合に操作が実行されます。2 つの主要な操作は、メッセージの送信 (通知) とリモートコマンドの実行です。

ディスカバリおよび自動登録によって作成されたイベントについては、いくつかの追加操作を使用できます。追加操作はホストの追加または削除、テンプレートのリンクなどです。

1 メディアタイプ

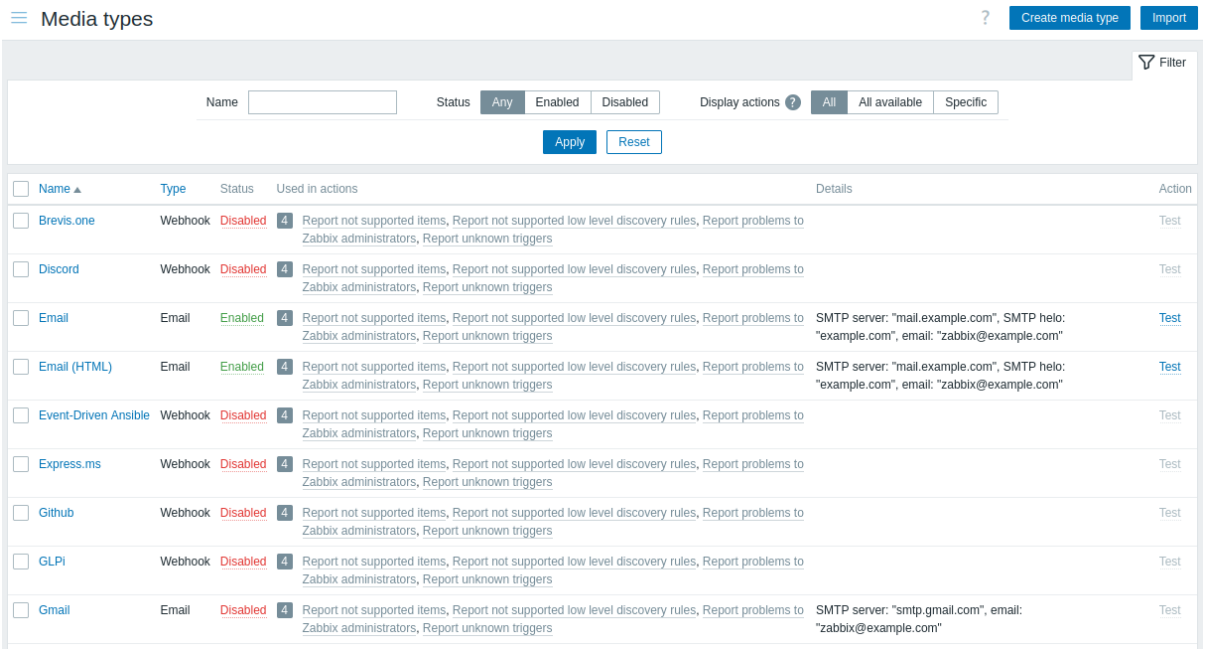
概要

メディアタイプは、Zabbix から通知やアラートを送信するために使用される配信チャネルです。

メディアタイプは、次の配信方法をサポートしています。

- Email
- SMS
- カスタムスクリプト
- Webhook

メディアタイプは、アラート > メディアタイプで管理されます。一部のメディアタイプは、デフォルトのデータセットにあらかじめ定義されています。
動作させるには、それらのパラメータを微調整するだけで済みます。



メディアタイプがアラート処理の中でどのように組み込まれているかを確認するために、Zabbix からエンドユーザーへ通知を配信するための 3 つの要件を見てみましょう。

1. 通知を送信するアクション操作を定義する必要があります
2. 動作するメディアタイプを定義する必要があります（SMTP を使用してアラートを送信する *Email* など）
3. ユーザーメディアで、ユーザーレベルの配信先情報（メールアドレス、電話番号など）を定義する必要があります

メディアタイプのテスト

設定済みのメディアタイプが機能するかどうかをテストするには、メディアタイプ一覧で テストをクリックします。

テスト要求は Zabbix サーバーに送信されます。Zabbix サーバーは、指定されたメディアタイプを使用してアラートの送信を試み、その結果を Web インターフェースに返します。Web インターフェースは、サーバーから結果が返されるのを待機します。メディアタイプのテストのタイムアウトはデフォルトで 65 秒です（管理 > 一般 > タイムアウト で設定可能）。

以下のテストの詳細も参照してください。

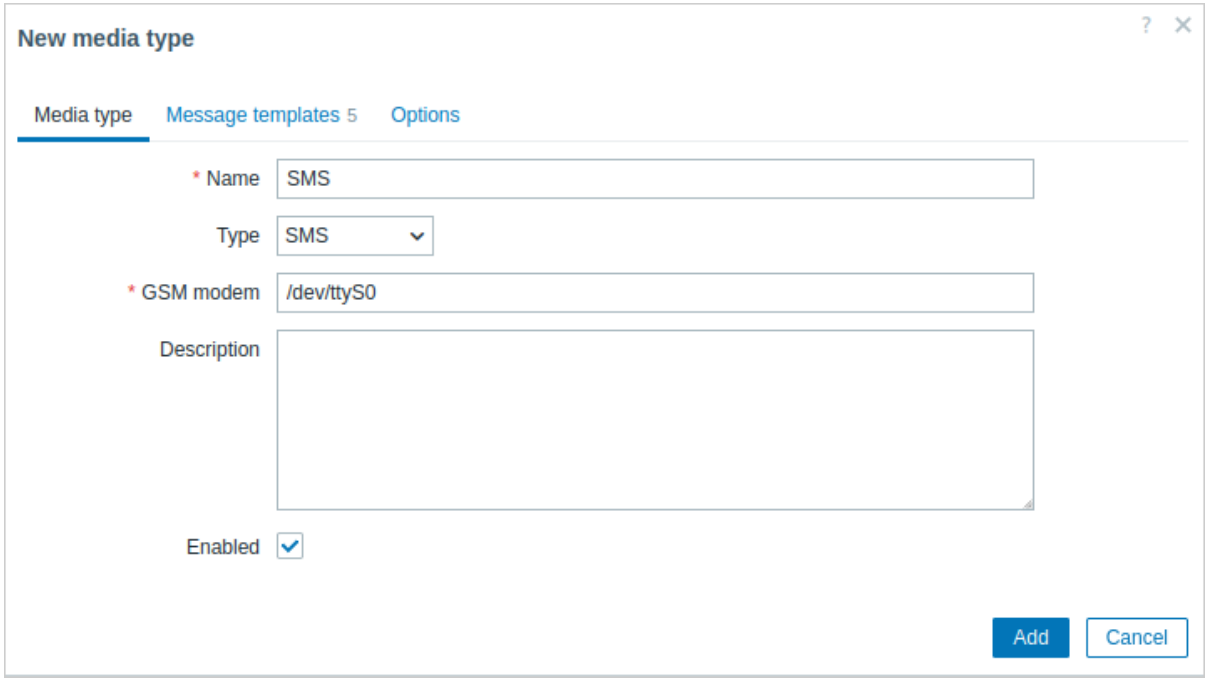
- Email
- Webhook
- Script

設定

Zabbix Web インターフェースでメディアタイプを作成するには、次の手順を実行します。

- アラート > メディアタイプに移動します
- メディアタイプの作成をクリックします
- フォームにメディアタイプのパラメータを入力します

一部のパラメータは、すべての配信方法で共通です。

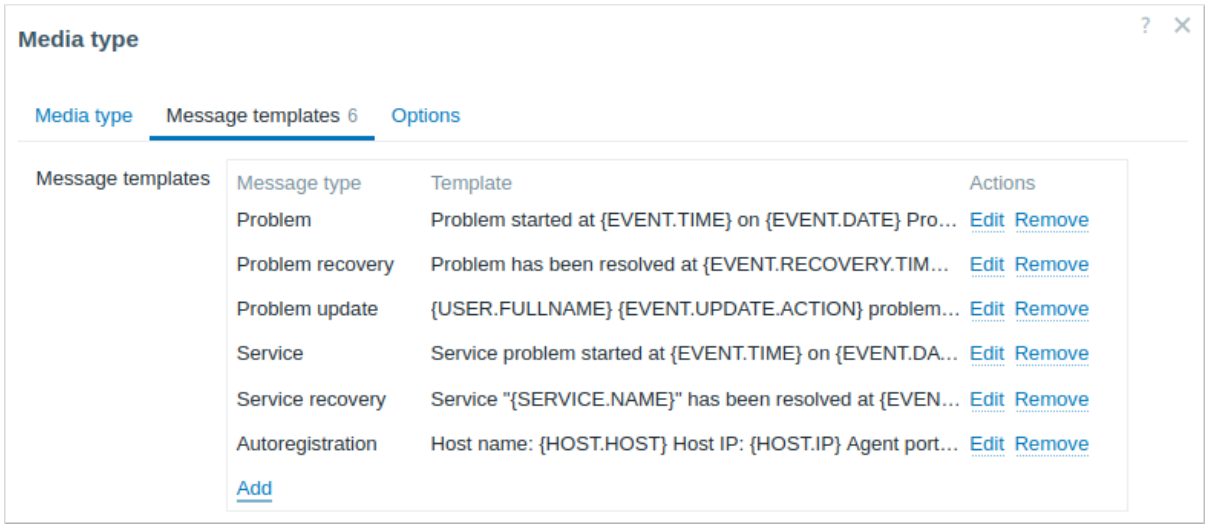


パラメータ	説明
Name	メディアタイプの名前。
Type	メディアタイプの配信方法を選択します。
Description	メディアタイプの説明を入力します。
Enabled	チェックボックスをオンにしてメディアタイプを有効にします。

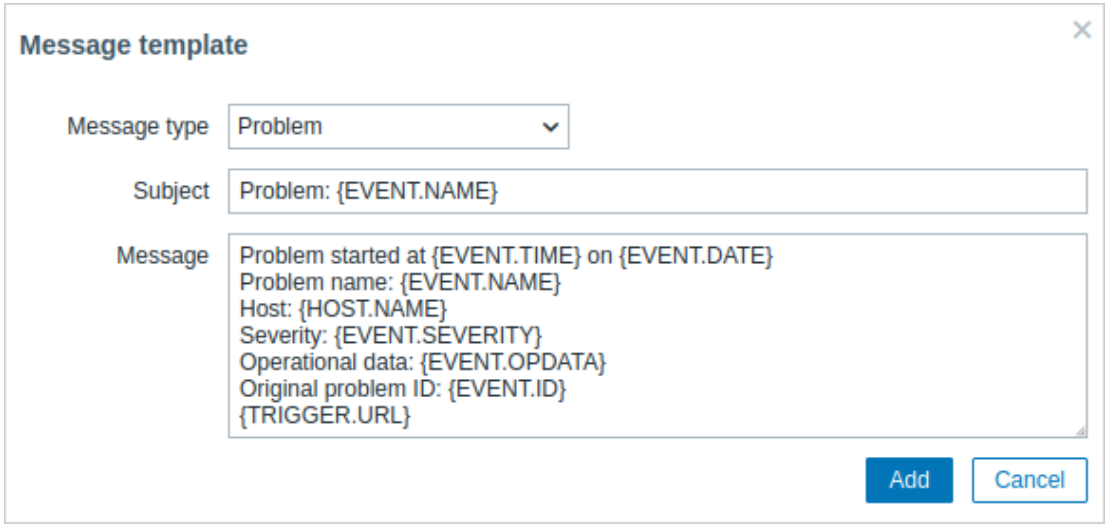
方法固有のパラメータについては、email、SMS、カスタムアラートスクリプト、またはwebhook のページを参照してください。

メッセージテンプレート

メッセージテンプレートタブには、イベントタイプごと（障害、障害復旧、ディスカバリなど）のデフォルトメッセージが含まれています。



追加をクリックしてデフォルトメッセージを定義します（既存のメッセージを更新する場合は編集）：



Parameter	Description
Message type	デフォルトメッセージを使用するイベントのタイプ。 各イベントタイプにつき定義できるデフォルトメッセージは 1 つだけです。
Subject	デフォルトメッセージの件名。件名にはマクロを含めることができます。255 文字までに制限されています。 SMS メディアタイプでは件名は使用できません。
Message	デフォルトメッセージ。データベースの種類に応じて文字数の上限があります（詳細はメッセージの送信を参照）。 メッセージには、サポートされているマクロを含めることができます。 障害メッセージおよび障害更新メッセージでは、式マクロがサポートされています（例: `{?avg(/host/key,1h)}`）。

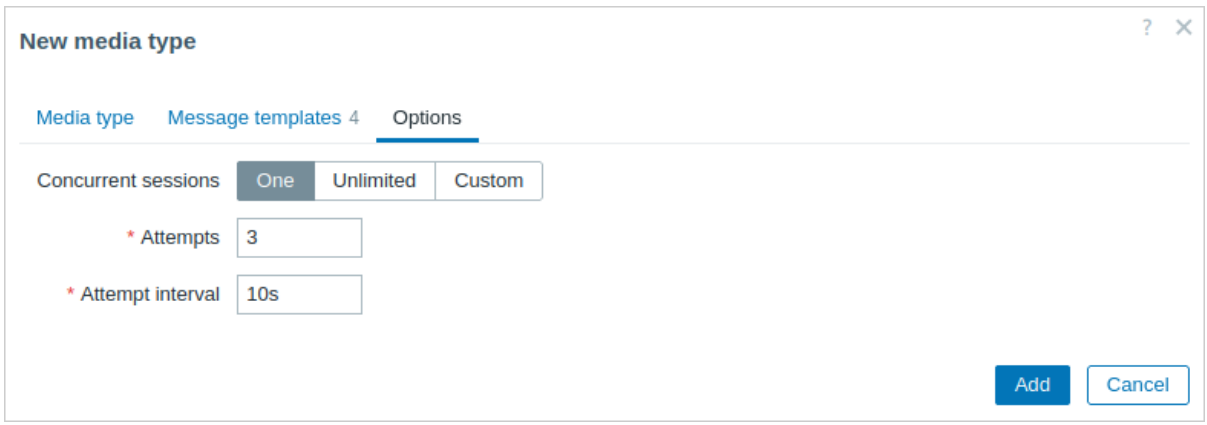
action operations で定義されている場合、デフォルトメッセージはカスタムメッセージによって上書きされることに注意してください。

Warning:
メッセージテンプレートの定義は、通知にデフォルトメッセージを使用しない webhook やカスタムアラートスクリプトを含む、すべての配信方法で必須です。
たとえば、Pushover webhook へのメッセージ送信アクションでは、Pushover webhook 用の障害メッセージが定義されていない場合、障害通知の送信に失敗します。

オプション

オプションタブには、アラート処理の設定が含まれています。
同じ一連のオプションを各メディアタイプごとに設定できます。

すべてのメディアタイプは並列に処理されます。
同時セッションの最大数はメディアタイプごとに設定できますが、
サーバー上のアラート送信プロセスの総数は、StartAlerters parameter によってのみ制限できます。
1 つのトリガーによって生成されたアラートは順次処理されます。
そのため、複数の通知が同時に処理されるのは、それらが複数のトリガーによって生成された場合に限られます。



Parameter	Description
Concurrent sessions	メディアタイプの並列アラート送信セッション数を選択します: **One** - 1 セッション **Unlimited** - セッション数無制限 **Custom** - カスタムのセッション数を選択 Unlimited/高い値は、より多くの並列セッションと、通知送信容量の増加を意味します。 Unlimited/高い値は、多数の通知を同時に送信する必要がある大規模環境で使用してください。 同時セッション数を超える通知を送信する必要がある場合、残りの通知はキューに入れられます。 通知が失われることはありません。
Attempts	通知送信を試行する回数です。最大 100 回まで指定でき、デフォルト値は「3」です。「1」を指定した場合、Zabbix は通知を 1 回だけ送信し、送信に失敗しても再試行しません。
Attempt interval	送信に失敗した場合に通知の再送信を試みる間隔を秒単位で指定します (0-3600)。「0」を指定した場合、Zabbix は即座に再試行します。 時間のサフィックスを使用できます。例: 5s、3m、1h。

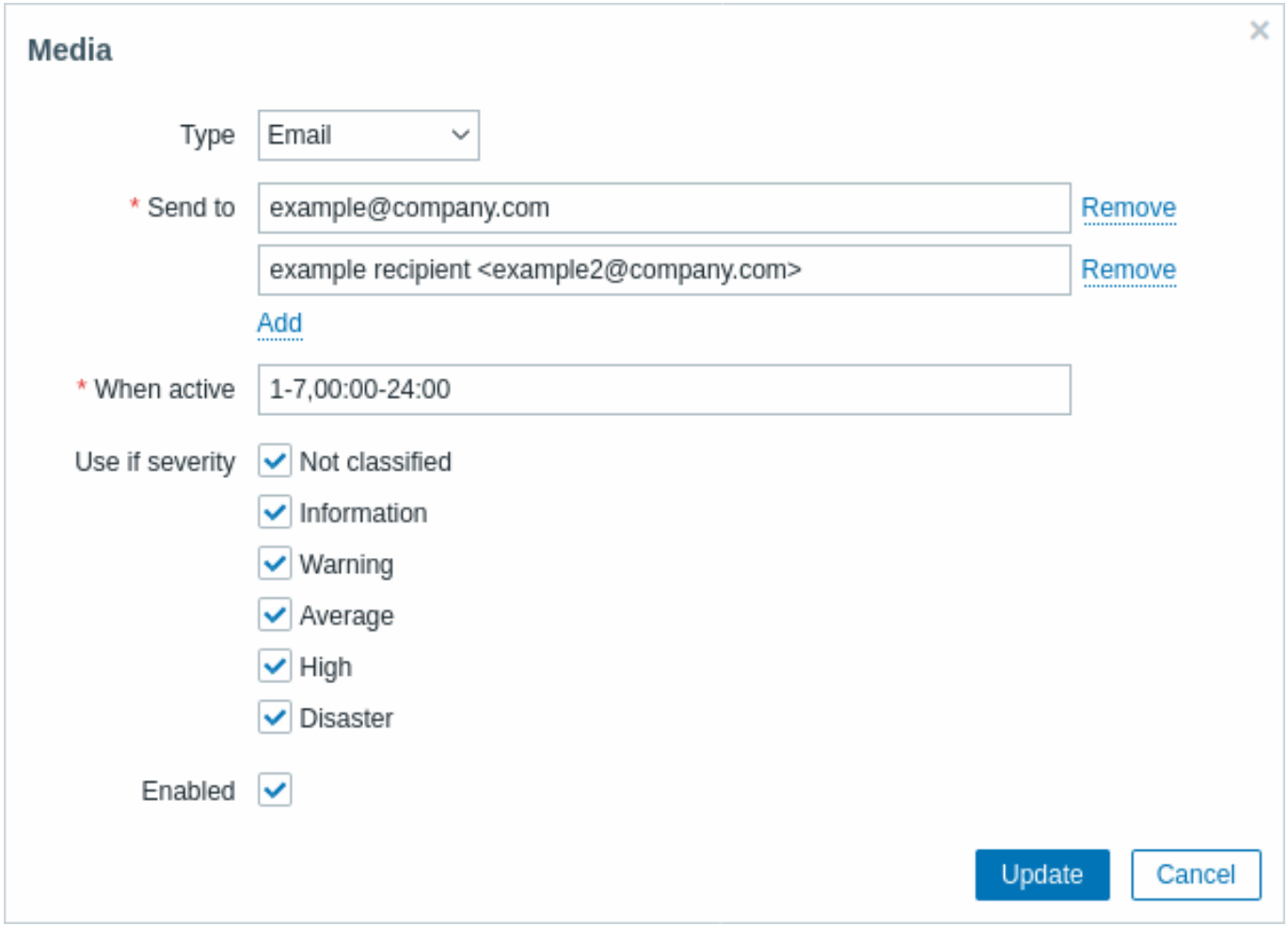
ユーザーメディア

メディアタイプは通知をどのように送信するかを定義するのに対し、ユーザーメディアは通知をどこに送信するかを定義します。

ユーザーメディア（例: メールアドレス、webhook ユーザー ID など）は、配信方法に関係なく、ユーザープロファイルで定義する必要があります。webhook *X* を使用して *Admin* ユーザーにメッセージを送信するアクションは、Admin ユーザープロファイルに webhook *X* の配信先詳細が定義されていない場合、配信に失敗します。

ユーザーメディアを定義するには:

- *Users* > *Users* に移動してユーザーのプロパティフォームを開きます（または、自分のユーザープロファイルの *User settings* > *Notifications* に移動します）
- メディアタブで *Add* をクリックします



Parameter	Description
Type	ドロップダウンリストには、有効なメディアタイプの名前が表示されます。 無効なメディアタイプのメディアを編集中の場合、そのタイプは赤色で表示されることに注意してください。
Send to	メッセージの送信先となる連絡先情報を入力します。 メールメディアタイプでは、アドレス欄の下にある Add ボタンをクリックすることで、複数のアドレスを追加できます。この場合、通知は一覧にあるすべてのアドレスに送信されます。アドレスの例については、メディアタイプemail の *Email* パラメータの説明を参照してください。
When active	メッセージを送信する時間を制限できます。たとえば、稼働日のみ（1-5,09:00-18:00）に設定できます。この制限はユーザーのタイムゾーンに基づくことに注意してください。ユーザーのタイムゾーンが変更され、システムのタイムゾーンと異なる場合は、重要なメッセージを見逃さないように、この制限をそれに応じて調整する必要がある場合があります。 形式の説明については、期間指定ページを参照してください。 ユーザーマクロを使用できます。
Use if severity	通知を受け取りたいトリガーの深刻度のチェックボックスをオンにします。 トリガー以外のイベントの通知を受け取りたい場合は、デフォルトの深刻度（'Not classified'）を必ずチェックする必要があることに注意してください。 保存後、選択したトリガーの深刻度は対応する深刻度の色で表示され、未選択のものはグレー表示になります。
Status	ユーザーメディアのステータスです。 **Enabled** - 使用中です。 **Disabled** - 使用されていません。

1 E-mail

概要

メッセージの配信チャネルとしてメールを設定するには、メディアタイプとしてメールを設定し、ユーザーに特定のメールアドレスを割り当てる必要があります。

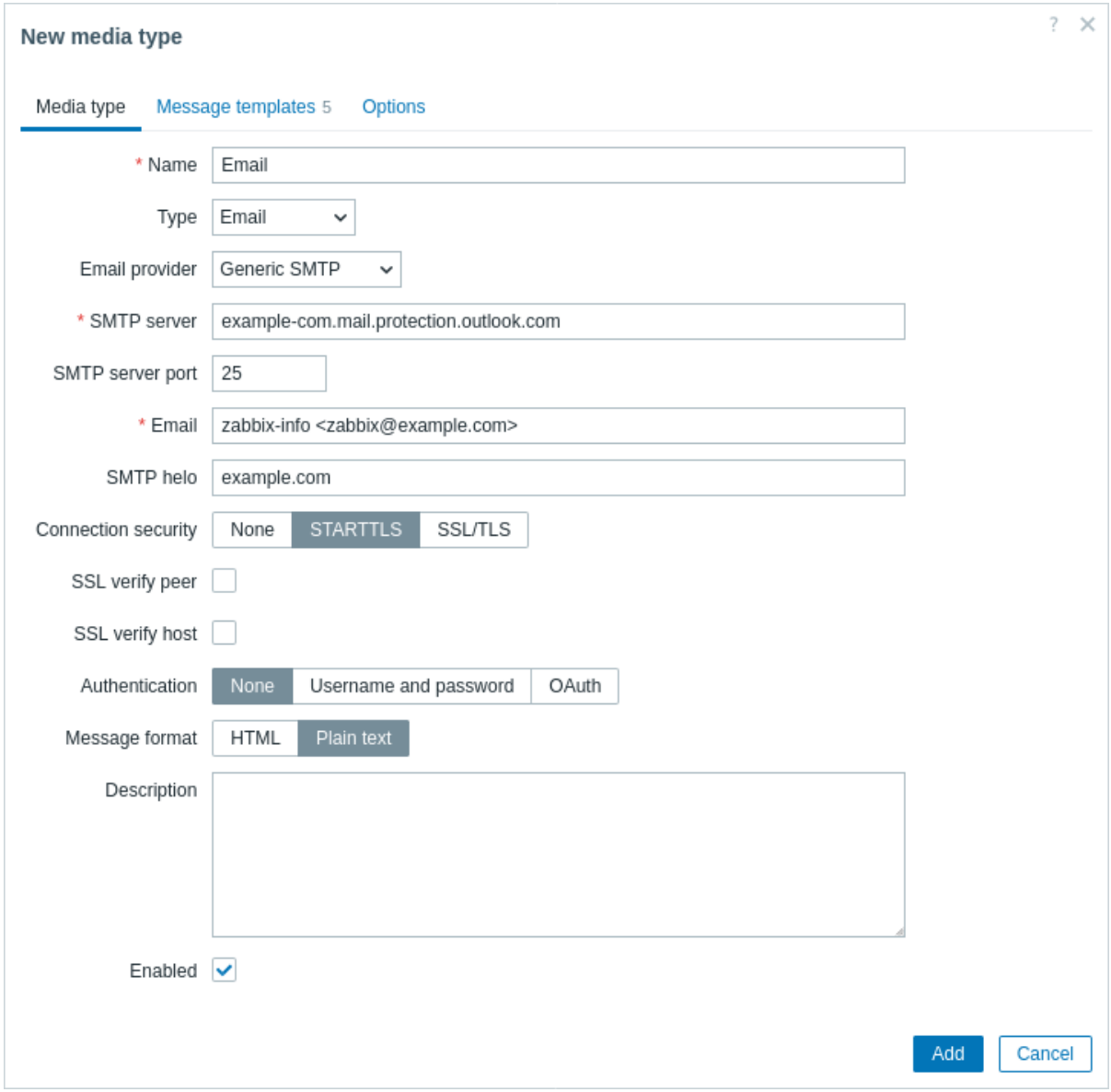
> **Note:**
> 単一のイベントに対する複数の通知は、同じメールスレッドにまとめられます。

設定

メディアタイプとしてメールを設定するには、次のようにします。

1. *Alerts* > *Media types* に移動します。
2. *Create media type* をクリックします（または、事前定義されたメディアタイプの一覧で *Email* をクリックします）。

Media type タブには、メディアタイプの一般的な属性が含まれています。



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

Warning:
メールメディアタイプ設定をエクスポートすると、パスワードは平文でエクスポートされます。

以下のパラメータは、メールメディアタイプ固有のものです。

Parameter	Description
Email provider	メールプロバイダーを選択します: *Generic SMTP*、*Gmail*、*Gmail relay*、*Office365*、または *Office365 relay*。 Gmail/Office365 関連のオプションを選択した場合、送信元メールアドレスとパスワードを指定するだけで済みます。*SMTP server*、*SMTP server port*、*SMTP helo*、*Connection security* などのオプションは、Zabbix によって自動的に入力されます。あわせて参照: Automated Gmail/Office365 media types。
SMTP server	送信メッセージを処理する SMTP サーバーを設定します。 このフィールドは、メールプロバイダーとして *Generic SMTP* が選択されている場合に使用できます。
SMTP server port	送信メッセージを処理する SMTP サーバーポートを設定します。 このフィールドは、メールプロバイダーとして *Generic SMTP* が選択されている場合に使用できます。

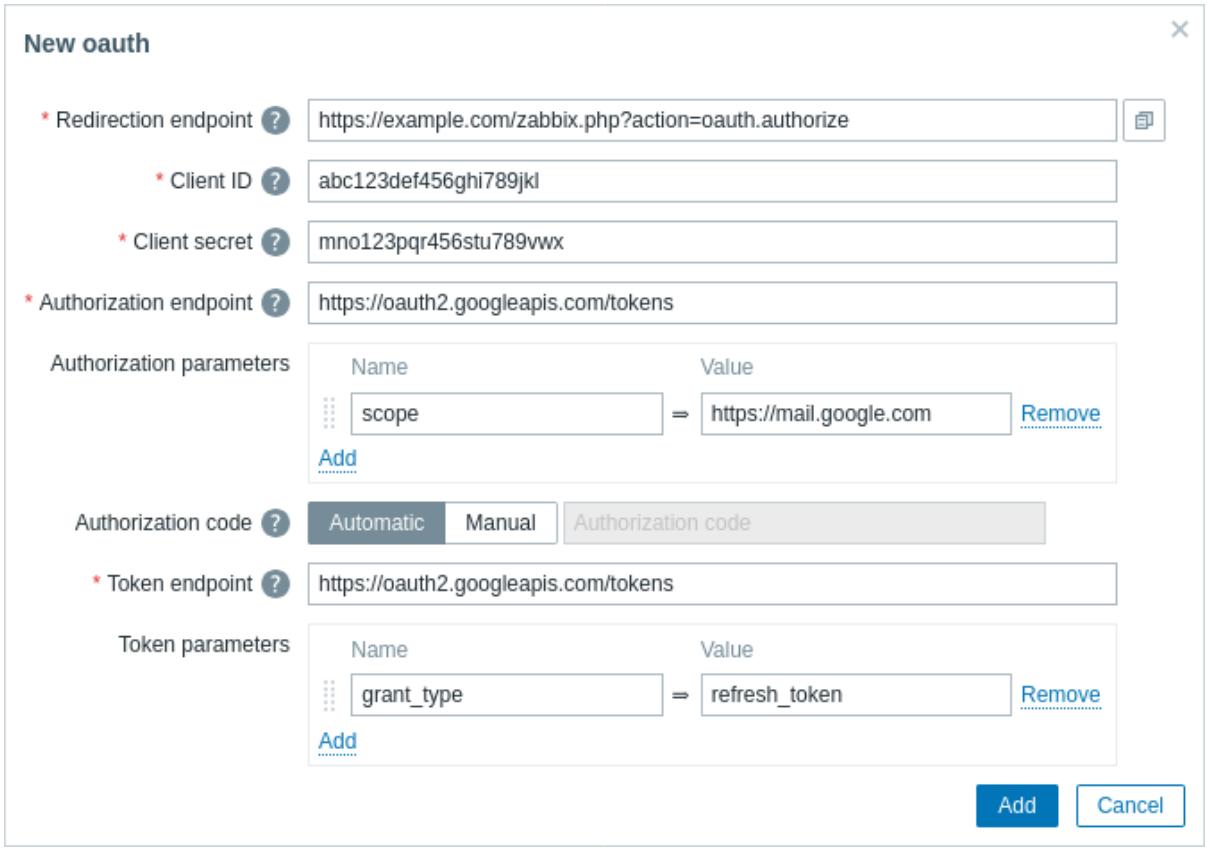
Parameter	Description
Email	ここに入力したアドレスは、送信されるメッセージの **From** アドレスとして使用されます。 実際のメールアドレスに送信者表示名を追加することもサポートされています（上のスクリーンショットの *Zabbix_info <zabbix@company.com>* における”Zabbix_info” など）。 Zabbix のメールにおける表示名には、RFC 5322 で許可されている内容と比べていくつか制限があります。以下の例を参照してください。 有効な例: *zabbix@company.com*（メールアドレスのみ。山括弧は不要） *Zabbix_info <zabbix@company.com>*（表示名とメールアドレスを山括弧で囲む） *∑Ω-monitoring <zabbix@company.com>*（表示名に UTF-8 文字を使用） 無効な例: *Zabbix HQ zabbix@company.com*（表示名はあるが、メールアドレスが山括弧で囲まれていない） *”Zabbix\@\<H(comment)Q\>” <zabbix@company.com>*（RFC 5322 では有効ですが、引用ペアおよびコメントは Zabbix のメールではサポートされていません）
SMTP helo	通常はドメイン名である、正しい SMTP helo 値を設定します。 空の場合は、メールのドメイン名（つまり、*Email* フィールドの @ の後ろの部分）が送信されます。 ドメイン名を取得できない場合は、デバッグレベルの警告がログに記録され、HELO コマンドのドメインとしてサーバーのホスト名が送信されます。 このフィールドは、メールプロバイダーとして *Generic SMTP* が選択されている場合に使用できます。
Connection security	接続セキュリティのレベルを選択します。 **None** - CURLOPT_USE_SSL オプションを使用しません **STARTTLS** - CURLOPT_USE_SSL オプションを CURLUSESSL_ALL 値で使用します **SSL/TLS** - CURLOPT_USE_SSL の使用は任意です
SSL verify peer	SMTP サーバーの SSL 証明書を検証するには、チェックボックスをオンにします。 証明書の検証には、サーバー設定ディレクティブ”SSLCALocation” の値を CURLOPT_CAPATH に設定する必要があります。 これにより、cURL オプション CURLOPT_SSL_VERIFYPEER が設定されます。
SSL verify host	SMTP サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスをオンにします。 これにより、cURL オプション CURLOPT_SSL_VERIFYHOST が設定されます。
Authentication	認証レベルを選択します。 **None** - cURL オプションは設定されません **Username and password** - ”AUTH=*” を意味し、認証メカニズムの選択は cURL に委ねられます **OAuth** - OAuth 認証 *Office365 relay* メールプロバイダーでは OAuth 認証はサポートされていません。
Username	認証に使用するユーザー名です。 これにより、CURLOPT_USERNAME の値が設定されます。 ユーザーマクロ をサポートしています。
Password	認証に使用するパスワードです。 これにより、CURLOPT_PASSWORD の値が設定されます。 ユーザーマクロ をサポートしています。 ここに入力した値は、メディアタイプ設定をエクスポートするとそのままエクスポートされます。 エクスポートファイル内で機密性の高い認証情報が露出するのを避けるには、リテラルなパスワードの代わりにユーザーマクロ（たとえば `{$EMAIL_SMTP_PASSWORD}`）を使用してください。この場合、エクスポートされるのはマクロ参照であり、秘密の値自体は移行先システムで設定（または再設定）する必要があります。
OAuth tokens	*Configure* をクリックすると、新しいウィンドウでOAuth tokens を取得するためのパラメータを設定できます。 このフィールドは、*Authentication* フィールドで”OAuth” が選択されている場合にのみ使用できます。
Message format	メッセージ形式を選択します。 **HTML** - HTML として送信 **Plain text** - プレーンテキストとして送信

Attention:
SMTP 認証を有効にするには、Zabbix サーバーを `--with-libcurl` compilation オプション付きでコンパイルする必要があります。これには libcurl が必要です（バージョンの詳細についてはserver 要件を参照してください）。

デフォルトメッセージおよびアラート処理オプションの設定方法の詳細については、common media type parameters も参照してください。

OAuth トークン

OAuth トークンを取得するには、以下のパラメータが必要です。



Parameter	Description
Redirection endpoint	OAuth サービスが OAuth 認可後にリダイレクトする先の Zabbix Web インターフェース URL を入力します（形式: `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`）。 frontend URL が定義されている場合、新しいメールメディアタイプでは自動的に設定されます。
Client ID	OAuth 認可サーバーに登録されたクライアントアプリの一意な識別子を入力します。
Client secret	OAuth 認可サーバーに登録されたクライアントアプリの秘密鍵を入力します。
Authorization endpoint	ユーザー認可を要求するための OAuth 認可サーバー URL を入力します。
Authorization parameters	認可エンドポイントのパラメータを入力します。
Authorization code	認可コードを入力します。 **Automatic** - リダイレクトページを通じてコードが自動的に取得されます **Manual** - 自動取得に失敗した場合は、コードを手動で入力します
Token endpoint	認可コードをアクセストークンおよびリフレッシュトークンに交換するための OAuth 認可サーバー URL を入力します。
Token parameters	アクセストークンのパラメータを入力します。

Note:
OAuth トークンの取得には、**Gmail**、**Gmail relay**、**Office365** のメールプロバイダー向けに自動化機能があります。必要なのは、*Redirection endpoint*、*Client ID*、*Client secret* の各パラメータ値を指定することだけです。Zabbix がその他の必要な値を自動的に入力します（プロバイダーごとの OAuth URL デフォルト値を参照）。

OAuth アクセストークンおよびリフレッシュトークンの取得

OAuth パラメータを送信すると、次のように処理されます。

1. ブラウザのポップアップが開き、ユーザーは認可エンドポイントに移動します。
2. ユーザーが OAuth サービスで Zabbix を認可します。
3. OAuth サービスは、認可コードとスコープ値を付加して、ユーザーを Zabbix の Web インターフェースアクション `oauth.authorize` にリダイレクトします。

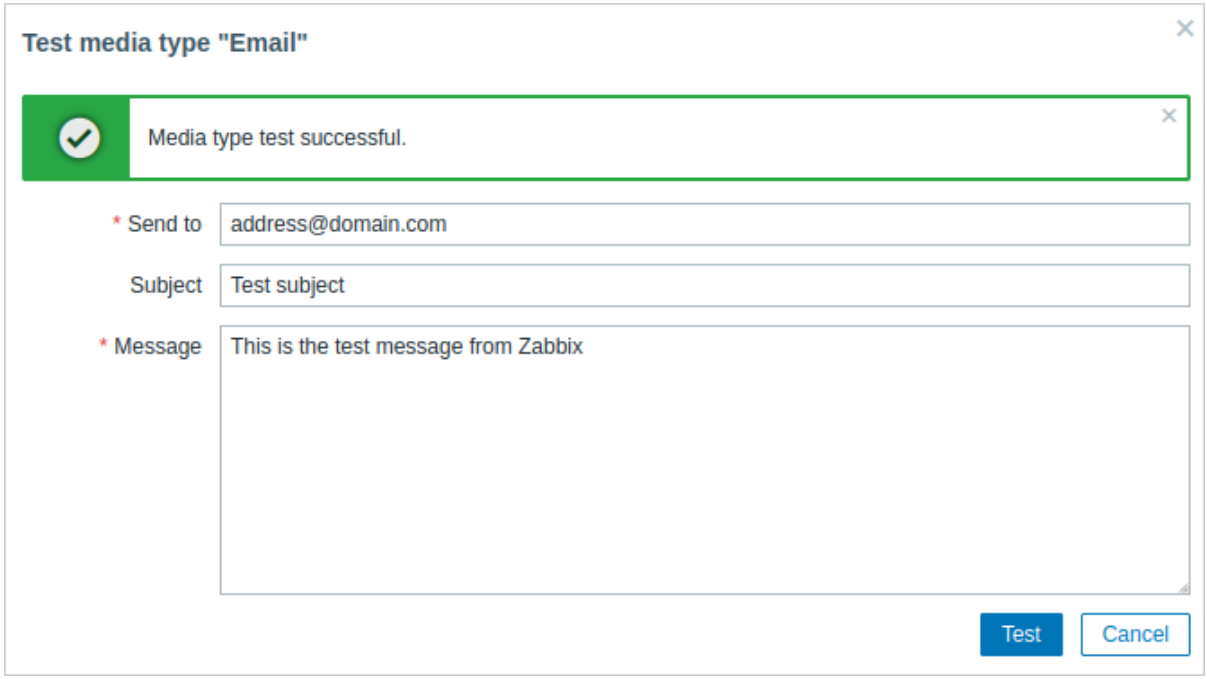
4. これに応答して、Zabbix は認可コードをアクセストークンおよびリフレッシュトークンに交換するため、トークンエンドポイントにリクエストを送信します。

テスト

設定済みのメールメディアタイプが正しく動作するかどうかをテストするには、次の手順を実行します。

1. メディアタイプのlistで該当するメールを見つけます。
2. 一覧の最後の列にある *Test* をクリックします（テスト用のウィンドウが開きます）。
3. *Send to* の受信者アドレス、メッセージ本文、および必要に応じて件名を入力します。
4. *Test* をクリックしてテストメッセージを送信します。

テストの成功または失敗のメッセージは、同じウィンドウに表示されます。



ユーザーメディア

メールメディアタイプを設定したら、ユーザー > ユーザーセクションに移動し、ユーザープロファイルを編集してユーザーにメールメディアを割り当てます。
すべてのメディアタイプに共通するユーザーメディアの設定手順は、メディアタイプ ページで説明されています。

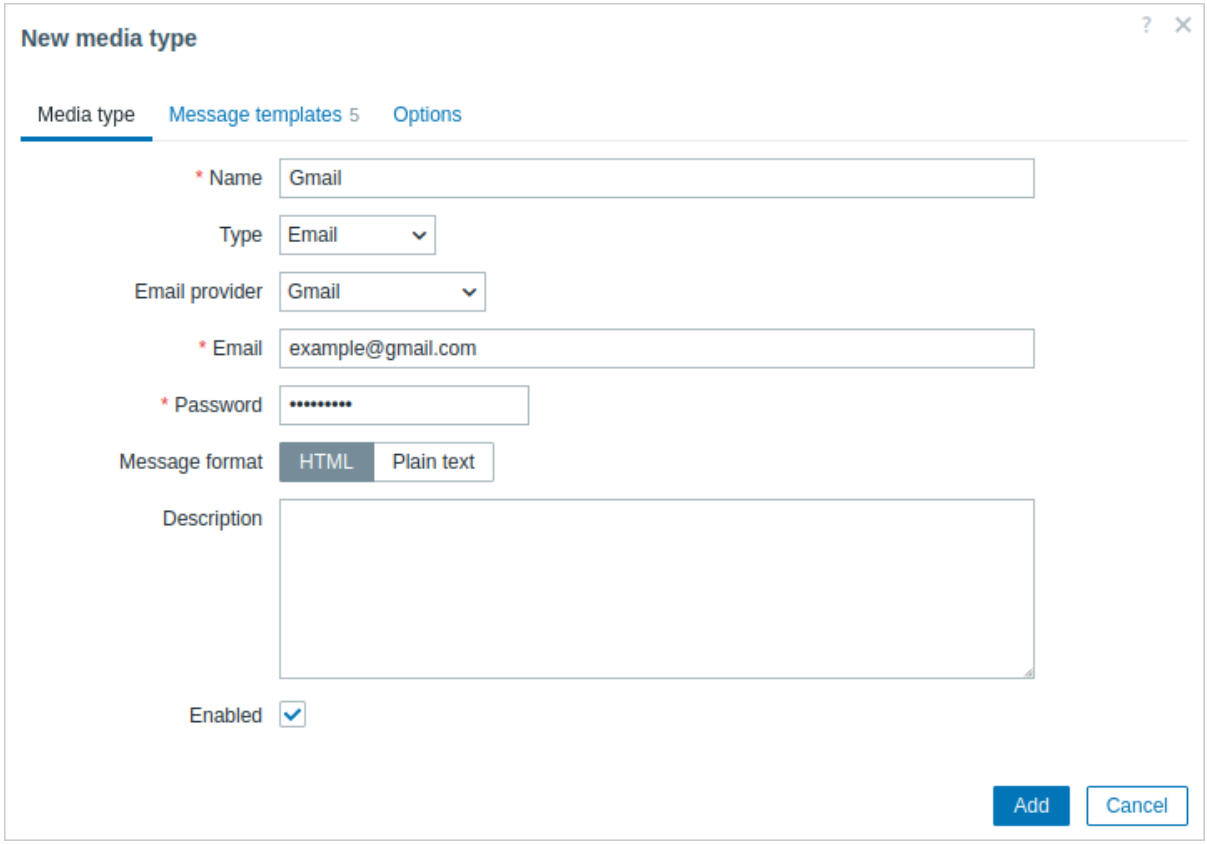
1 自動化された Gmail/Office365 メディアタイプ

概要

Gmail または Office365 ユーザーは、メディアタイプの設定における自動化機能の恩恵を受ける可能性があります。

設定

メールメディアタイプの設定にある *Email provider* フィールドでは、Gmail および Office 365 用の事前設定済みオプションを選択できます。Gmail/Office 365 関連のオプションを選択する場合、動作するメディアタイプを作成するには送信者メールアドレスとパスワードの入力のみが必要です。



メールアドレス/パスワードを入力すると、Zabbix は Gmail/Office365 メディアタイプに必要な設定項目 (*SMTP* サーバー、*SMTP* サーバーポート、*SMTP helo*、接続セキュリティ) を実際の値または推奨値で自動的に入力します。この自動化により、これらのフィールドは表示されませんが、メディアタイプ一覧の詳細列で SMTP サーバーとメールの詳細を確認できます。

なお以下の点に注意してください：- リレーオプションではパスワードは不要です - Office365 リレーの場合、指定されたメールアドレスのドメイン名が SMTP サーバーに動的に設定されます (例：`example-com.mail.protection.outlook.com` 内の「example.com」が実際のドメイン名に置換されます)

OAuth トークン

OAuth トークンの取得には、*Gmail*、*Gmail relay*、*Office365* メールプロバイダー向けの自動化機能が備わっています。



OAuth アクセスおよびリフレッシュトークン取得には、以下の限定されたパラメータセットのみを提供する必要があります：リダイレクトエンドポイント、クライアント *ID*、クライアントシークレット。Zabbix は他の必須値を自動的に入力します (プロバイダー別 OAuth URL デフォルトを参照)。なお、これらのプロバイダー向けに OAuth 認証を設定する際には、Generic SMTP形式の使用も可能です。

Note:
SMTP クライアント認証は、Office 365 においてユーザー/共有メールボックス単位またはテナント全体単位で有効化する必要があります（デフォルト設定では有効化されていません）。

プロバイダーごとのデフォルト OAuth URL

以下の表は、Zabbix が使用するプロバイダーごとのデフォルト OAuth URL とパラメータを一覧表示したものです。

| パラメータ |<|Gmail|Office365| 汎用 SMTP| |-|---------|-------------------|---------- ----------|-------------------| |*Authorization endpoint*|<|`https://accounts.google.com/o/oauth2/v2/auth`|`https://login.microsoftonline.com/common/oauth2/v2.0/au`

デフォルトなし | | |*scope*|https://mail.google.com|https://outlook.office.com/SMTP.Send, offline_access| デフォルトなし | |^|*access_type*|offline| 未使用 | デフォルトなし | |^|*prompt*|consent| 未使用 | デフォルトなし | |^|*redirect_uri*|*Redirection endpoint* フィールドの値はコードによって自動的に追加されます。*Authorization endpoint* パラメータには記載されません。|<|<| |^|*state*|OAuth トークンで更新されるメディアを識別するための一意のハッシュは、コードによって自動的に追加されます。認証エンドポイントパラメータには記載されません。|<|<| |*Token endpoint*|<|https://oauth2.googleapis.com/token|https://login.microsoftonline.com/common/oauth2/v2.0/token| デフォルトなし | | |*grant_type*|authorization_code|authorization_code| デフォルトなし | |^|*redirect_uri*|*Redirection endpoint* フィールドの値がコードによって自動的に追加されます。*Token endpoint* パラメータには記載されません。|<|<| |^|*state*|OAuth トークンで更新されるメディアを識別するための一意のハッシュがコードによって自動的に追加されます。*Token endpoint* パラメータには記載されません。|<|<|

個人アカウント向け Office365 ワークアラウンド

Office365 個人アカウントでは、リダイレクト URL 内のクエリ文字列がサポートされていません。この問題を回避するには、Apache ウェブサーバー管理者が.htaccess ファイルに以下の書き換えルールを追加します：

```
RewriteEngine On RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

以下のような簡略化されたリダイレクト URL の使用が可能になります：

```
http://server-name/zabbix/oauth.authorize
```

これにより、クエリパラメータを含む URL の代わりに、Office365 個人アカウントとの互換性が確保されます。

2 SMS

概要

Zabbix は Zabbix server のシリアルポートに接続されたシリアル GSM モデムを使用した SMS メッセージの送信に対応しています。

以下の点を確認してください:

- シリアルデバイス (Linux では通常/dev/ttyS0) の速度は GSM モデムの速度に合わせます。Zabbix はシリアルリンクの速度を設定しません。デフォルトの設定を使用します。
- zabbix ユーザは、シリアルデバイスへのリード/ライトアクセス権を持ちます。ls -l /dev/ttyS0 コマンドを実行し、シリアルデバイスの現在のパーミッションを確認します。
- GSM モデムには PIN が入力されており、パワーリセット後も PIN が保持されます。また、SIM カードの PIN を無効化することもできます。PIN は Unix の minicom や Windows の HyperTerminal などのターミナルソフトで AT+CPIN=”NNNN”（NNNN は PIN 番号、引用符は必ず入れてください）コマンドを発行することで入力することができます。

Zabbix は以下の GSM モデムでテストされています:

- Siemens MC35
- Teltonika ModemCOM/G10

メッセージの配信チャネルとして SMS を設定するには、メディアタイプとして SMS を設定し、ユーザーのそれぞれの電話番号を入力する
 必要があります。

設定

SMS をメディアタイプとして設定するには、次の手順を実行します。

- アラート → メディアタイプに移動します
- メディアタイプの作成をクリックします（または、事前定義されたメディアタイプの一覧で *SMS* をクリックします）。

以下のパラメータは、SMS メディアタイプ固有のものです。

Parameter	Description
GSM modem	GSM モデムのシリアルデバイス名を設定します。 ここで入力したパスは、SMSDevices サーバーパラメータ（指定されている場合）に対して検証されます。

デフォルトメッセージおよびアラート処理オプションの設定方法の詳細については、共通のメディアタイプパラメータ を参照してください。SMS 通知の送信を並列処理することはできない点に注意してください。

ユーザーメディア

SMS メディアタイプの設定が完了したら、ユーザー → ユーザーセクションに移動し、ユーザープロファイルを編集して SMS メディアをユーザーに割り当てます。すべてのメディアタイプに共通するユーザーメディア設定の手順は、メディアタイプ ページで説明されています。

3 カスタムアラートスクリプト

概要

アラート送信のための既存のメディアタイプに満足できない場合は、別の方法があります。
独自の方法で通知を処理するスクリプトを作成できます。

カスタムアラートスクリプトは Zabbix サーバー上で実行されます。
これらのスクリプトは、サーバー設定ファイルのAlertScriptsPath パラメータで指定されたディレクトリに配置する必要があります。

以下は、カスタムアラートスクリプトの例です。

```
#####!/bin/bash

to=$1
subject=$2
body=$3
host=$4
value=$5

cat <<EOF | mail -s "$subject" "$to"
$body

Host: $host
Value: $value
EOF
```

Attention:
Zabbix は、実行されたコマンドおよびスクリプトの終了コードを確認します。
0 以外の終了コードはすべて、コマンド実行 エラーと見なされます。
その場合、Zabbix は失敗した実行を再試行します。

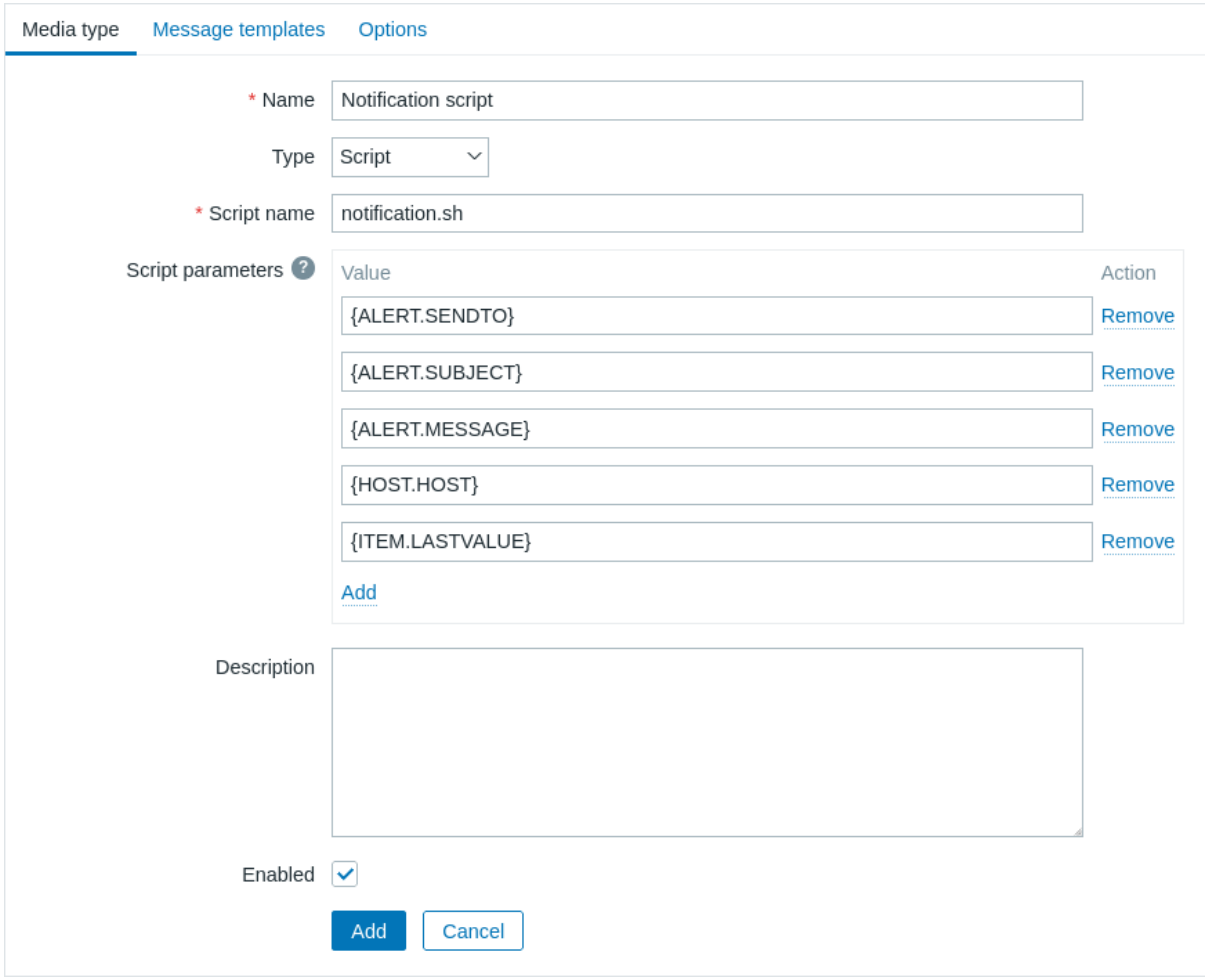
環境変数はスクリプトのために保持または作成されないため、明示的に処理する必要があります。

設定

カスタムアラートスクリプトをメディアタイプとして設定するには、次の手順を実行します。

1. *Alerts* → *Media types* に移動します。
2. *Create media type* をクリックします。

Media type タブには、メディアタイプの一般的な属性が含まれています。



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

以下のパラメータは、スクリプトメディアタイプ固有のものです。

Parameter	Description
Script name	AlertScriptsPath サーバー設定パラメータで指定されたディレクトリに配置されているスクリプトファイル名（例: notification.sh）を入力します。
Script parameters	オプションのスクリプトパラメータを追加します。これらは、定義された順序でコマンドライン引数としてスクリプトに渡されます。 スクリプトパラメータでは、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} マクロ、および通知でサポートされているすべてのマクロと、ユーザーマクロを使用できます。

デフォルトメッセージおよびアラート処理オプションの設定方法の詳細については、共通のメディアタイプパラメータを参照してください。

Warning:
アラートスクリプトがデフォルトメッセージを使用しない場合でも、このメディアタイプで使用される操作タイプのメッセージテンプレートは定義しておく必要があります。そうしないと、通知は送信されません。

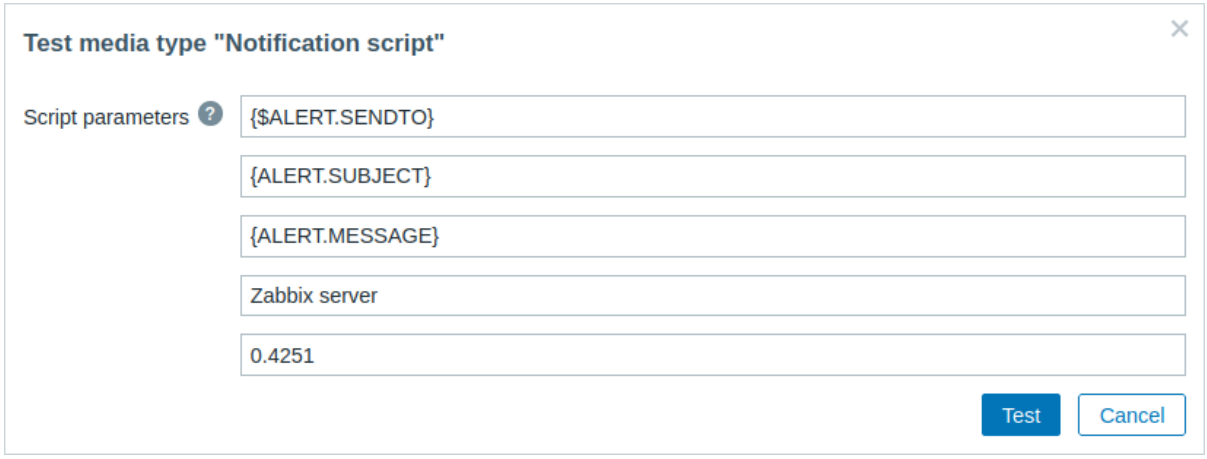
Attention:
複数のスクリプトメディアタイプが設定されている場合、これらのスクリプトは alerter プロセスによって並列に処理されることがあります。
alerter プロセスの総数は、サーバー設定ファイルの`StartAlerters` パラメータによって制限されます。

テスト

設定済みのスクリプトメディアタイプをテストするには、次の手順に従います。

1. メディアタイプの一覧で、該当するスクリプトを見つけます。

2. 一覧の最後の列にある *Test* をクリックします。ポップアップウィンドウでテストフォームが開きます。
テストフォームには、そのスクリプトメディアタイプに設定されているものと同じ数のパラメータが含まれます。
3. 必要に応じてスクリプトパラメータの値を編集します。編集はテスト手順にのみ影響し、実際の値は変更されません。
4. *Test* をクリックします。



Note:
設定済みのスクリプトメディアタイプをテストする際、{ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} およびユーザーマクロはそれぞれの値に展開されますが、イベントに関連するマクロ（例: {HOST.HOST}、{ITEM.LASTVALUE} など）は展開されません。これは、テスト時には詳細を取得するための関連イベントが存在しないためです。
また、{ALERT.SUBJECT} および {ALERT.MESSAGE} マクロ内のマクロも展開されないことに注意してください。
たとえば、{ALERT.SUBJECT} の値が"Problem: {EVENT.NAME}" で構成されている場合、{EVENT.NAME} マクロは展開されません。

ユーザーメディア

メディアタイプを設定したら、ユーザー → ユーザーセクションに移動し、このメディアタイプをユーザーに割り当ててユーザープロファイルを編集します。すべてのメディアタイプに共通するユーザーメディアの設定手順は、メディアタイプ ページで説明されています。

ユーザーメディアを定義する際、送信先フィールドは空にできないことに注意してください。このフィールドがアラートスクリプトで使用されない場合は、検証要件を回避するため、サポートされている文字の任意の組み合わせを入力してください。

4 Webhook

概要

webhook メディアタイプは、ヘルプデスクシステム、チャット、メッセンジャーなどの外部ソフトウェアと簡単に統合するために、
 カスタム JavaScript コードを使用して HTTP コールを行うのに便利です。Zabbix が提供する統合機能をインポートするか、
 カスタム統合機能を一から作成することができます。

連携

以下の連携が利用可能で、定義済みの webhook メディアタイプを使用して、Zabbix 通知を次の宛先に送信できます。

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams

- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
ここに記載されているサービスに加えて、Zabbix は **Spiceworks** とも連携できます（webhook は不要です）。Zabbix 通知を Spiceworks のチケットに変換するには、メールメディアタイプ を作成し、指定した Zabbix ユーザーのプロファイル設定に Spiceworks のヘルプデスクメールアドレス（例: help@zabbix.on.spiceworks.com）を入力してください。

設定

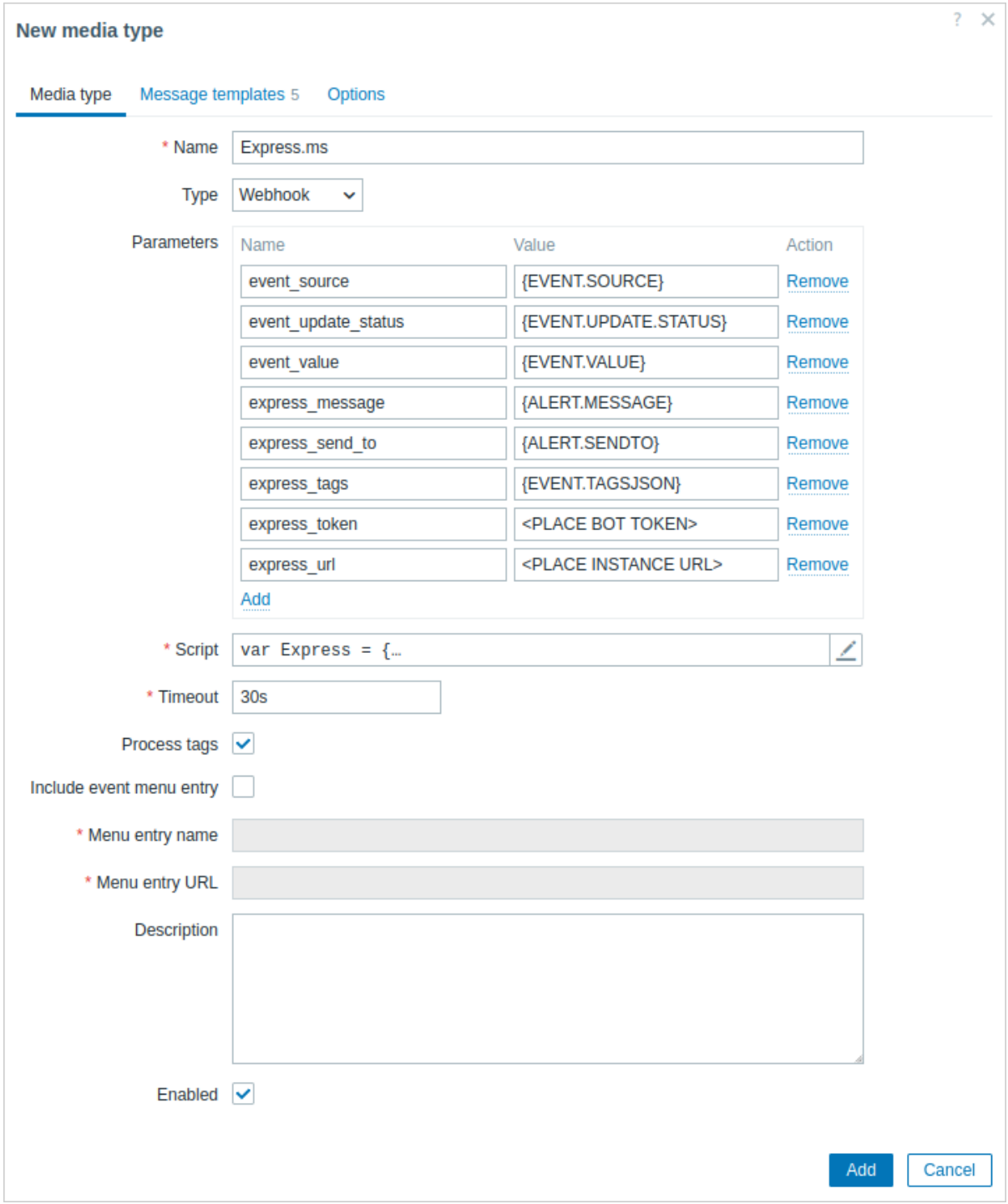
webhook 連携の使用を開始するには、次の手順を実行します。

1. ダウンロードした Zabbix バージョンの `templates/media` ディレクトリで必要な.yaml ファイルを探すか、Zabbix のgit repositoryからダウンロードします。
2. ファイルを Zabbix インストール環境にインポートします。
   webhook はメディアタイプの一覧に表示されます。
3. *Readme.md* ファイルの手順に従って webhook を設定します（上の webhook 名をクリックすると、*Readme.md* にすばやくアクセスできます）。

カスタム webhook を一から作成するには、次の手順を実行します。

1. *Alerts* > *Media types* に移動します。
2. *Create media type* をクリックします。

Media type タブには、このメディアタイプ固有のさまざまな属性が含まれています。



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

以下のパラメータは、webhook メディアタイプ固有のものです。

Parameter	Description
Parameters	属性と値のペアとして webhook 変数を指定します。 事前設定済みの webhook では、パラメータの一覧はサービスによって異なります。パラメータの説明は webhook の *Readme.md* ファイルを確認してください。 新しい webhook では、いくつかの共通変数がデフォルトで含まれています（URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}）。必要に応じてそのまま使用することも、削除することもできます。 webhook パラメータでは、ユーザーマクロ、障害通知でサポートされるすべてのマクロ、および追加で {ALERT.SENDTO}、{ALERT.SUBJECT}、{ALERT.MESSAGE} マクロをサポートします。 HTTP プロキシを指定する場合、このフィールドはアイテム設定のHTTP proxyフィールドと同じ機能をサポートします。プロキシ文字列には、使用するプロキシの種類を指定するために `[scheme]://` を接頭辞として付けることができます（例: https、socks4、socks5。詳細はdocumentationを参照してください）。
Script	パラメータフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターに JavaScript コードを入力します。このコードが webhook 処理を実行します。 スクリプトは、パラメータと値のペアを受け取る関数コードです。値は JSON.parse() メソッドを使用して JSON オブジェクトに変換する必要があります。例: `var params = JSON.parse(value);`。 このコードはすべてのパラメータにアクセスでき、HTTP GET、POST、PUT、DELETE リクエストを実行できるほか、CONNECT、PATCH、HEAD、OPTIONS、TRACE などの追加メソッドもサポートし、HTTP ヘッダーおよびリクエストボディを制御できます。 スクリプトには return 演算子を含める必要があります。含まれていない場合は無効になります。OK ステータスと、任意でタグおよびタグ値の一覧（*Process tags* オプションを参照）を返すことも、エラー文字列を返すこともできます。 スクリプトはアラートが作成された後にのみ実行される点に注意してください。スクリプトがタグを返して処理するよう設定されている場合、スクリプトがまだ実行されていないため、これらのタグは初回の障害メッセージおよび復旧メッセージ内の {EVENT.TAGS} および {EVENT.RECOVERY.TAGS} マクロでは解決されません。 注: 各スクリプトが自身のデータのみを扱い、同時呼び出し間の競合を避けるため、グローバル変数（例: `global = 1`）ではなくローカル変数（例: `var local = 1`）を使用することを推奨します（known issuesを参照）。 以下も参照してください: Webhook development guidelines、Webhook script examples、Additional JavaScript objects。
Timeout	JavaScript の実行タイムアウト（1～60 秒、デフォルトは 30 秒）。 時間の接尾辞をサポートします。例: 30s、1m。
Process tags	返された JSON プロパティ値をタグとして処理するには、このチェックボックスをオンにします。これらのタグは既存の障害タグに追加されます。 webhook tagsを使用する場合、webhook は少なくとも空の tags オブジェクトを含む JSON オブジェクトを返す必要がある点に注意してください: `var result = {tags: {}};` 返すことができるタグの例: *jira-id:prod-1234*、*responsible:John Smith*、*processed:<no value>*
Include event menu entry	作成された外部チケットへのリンクをevent menuに含めるには、このチェックボックスをオンにします。 有効化されていて、このチェックボックスがオンになっている各 webhook について、1 つのエントリが追加されます。*Menu entry name* および *Menu entry URL* パラメータに{EVENT.TAGS.<tag name>} マクロが含まれている場合、これらのマクロが解決可能なときのみ（つまり、そのイベントにこれらのタグが定義されている場合のみ）エントリが追加される点に注意してください。 このオプションを有効にした場合、その webhook は異なるユーザーへの通知送信には使用しないでください（代わりに専用ユーザーの作成を検討してください）。また、単一の障害イベントに対して複数のアラートアクションで使用しないでください。
Menu entry name	メニューエントリ名を指定します。 {EVENT.TAGS.<tag name>} マクロをサポートします。 このフィールドは、*Include event menu entry* がオンの場合にのみ必須です。
Menu entry URL	メニューエントリのリンク先 URL を指定します。 {EVENT.TAGS.<tag name>} マクロをサポートします。 このフィールドは、*Include event menu entry* がオンの場合にのみ必須です。

デフォルトメッセージおよびアラート処理オプションの設定方法の詳細については、共通メディアタイプパラメータを参照してください。

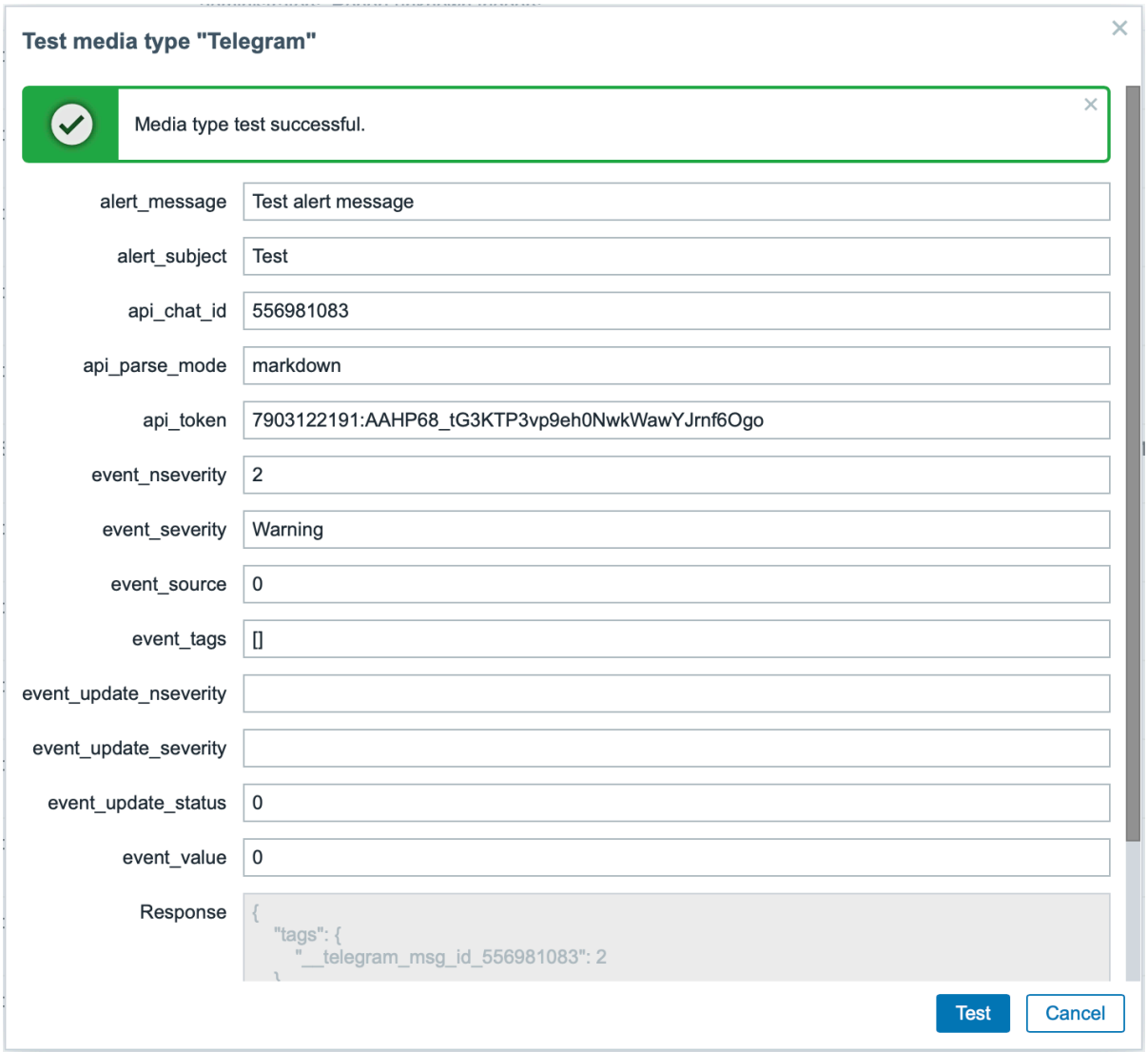
Warning:
webhook がデフォルトメッセージを使用しない場合でも、この webhook で使用される操作タイプのメッセージテンプレートは定義しておく必要があります。

テスト

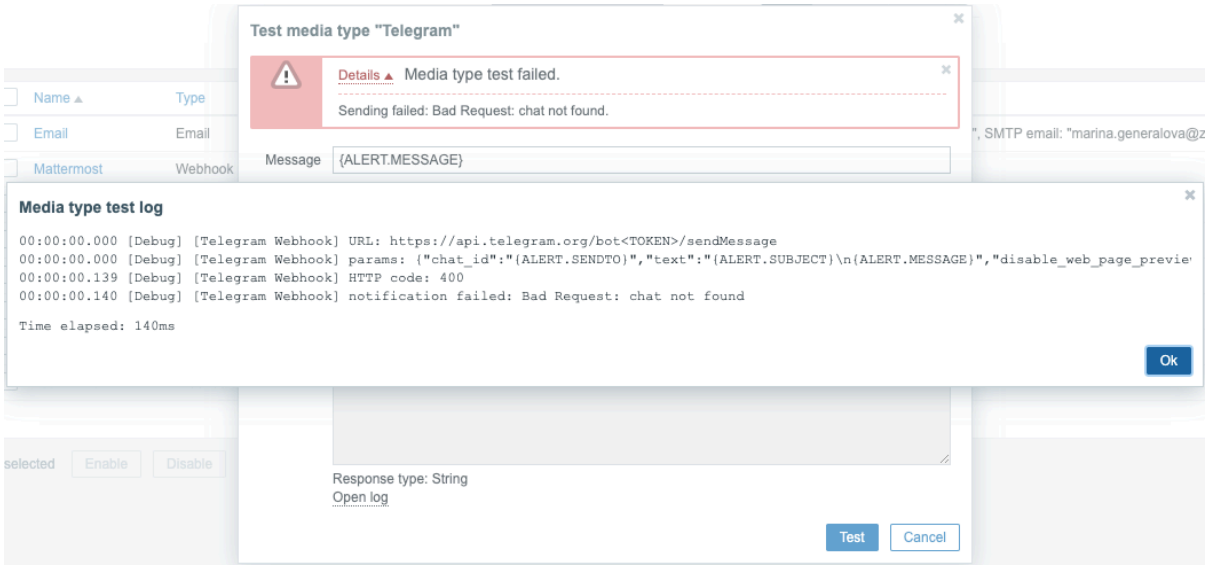
設定済みの webhook メディアタイプをテストするには、次のようにします。

1. メディアタイプの一覧で、該当する webhook を見つけます。
2. 一覧の最後の列にある *Test* をクリックします（テスト用ウィンドウが開きます）。
3. 必要に応じて webhook パラメータの値を編集します。
   マクロは例の値に置き換えてください。そうしないと、マクロが展開されず、テストは失敗します。
4. *Test* をクリックします。

テスト用ウィンドウで値を置き換えたり削除したりしても、影響するのはテスト手順のみであり、実際の webhook 属性値は変更されません。



テスト用ウィンドウを閉じずにメディアタイプのテストログエントリを表示するには、*Open log* をクリックします（新しいポップアップウィンドウが開きます）。



webhook テストが成功した場合**:**

- 「メディアタイプのテストに成功しました。」というメッセージが表示されます。
- サーバーのレスポンスが灰色の *Response* フィールドに表示されます。
- レスポンスタイプ（JSON または String）が *Response* フィールドの下に表示されます。

webhook テストが失敗した場合**:**

- 「メディアタイプのテストに失敗しました。」というメッセージが表示され、その後に追加の失敗の詳細が続きます。

ユーザーメディア

メディアタイプの設定が完了したら、*Users* > *Users* セクションに移動し、既存のユーザーに webhook メディアを割り当てるか、webhook を表す新しいユーザーを作成します。既存ユーザーに対するユーザーメディアの設定手順は、すべてのメディアタイプで共通であり、メディアタイプ ページで説明されています。

webhook がタグを使用してチケット\メッセージ ID を保存する場合、同じ webhook を異なるユーザーにメディアとして割り当てることは避けてください。そうしないと、webhook エラーが発生する可能性があります（これは *Include event menu entry* オプションを利用する大半の webhook に当てはまります）。この場合のベストプラクティスは、webhook を表す専用ユーザーを作成することです。

1. webhook メディアタイプを設定した後、*Users* > *Users* セクションに移動し、webhook を表す専用の Zabbix ユーザーを作成します。たとえば、Slack webhook 用であればユーザー名を *Slack* にします。
   メディア以外のすべての設定はデフォルトのままで構いません。このユーザーは Zabbix にログインしないためです。
2. ユーザープロファイルで *Media* タブに移動し、必要な連絡先情報を指定してwebhook を追加 します。
   webhook が *Send to* フィールドを使用しない場合は、バリデーション要件を回避するため、サポートされている文字の任意の組み合わせを入力してください。
3. このユーザーに対して、アラートを送信する必要があるすべてのホストに少なくとも読み取り権限を付与します。

アラートアクションを設定する際は、Operation details の *Send to users* フィールドにこのユーザーを追加してください。これにより、Zabbix はこのアクションからの通知に webhook を使用するようになります。

アラートアクションの設定

アクションは、webhook 経由でどの通知を送信するかを決定します。webhook を含むアクションの設定手順は、以下の例外を除き、他のすべてのメディアタイプの場合と同じです。

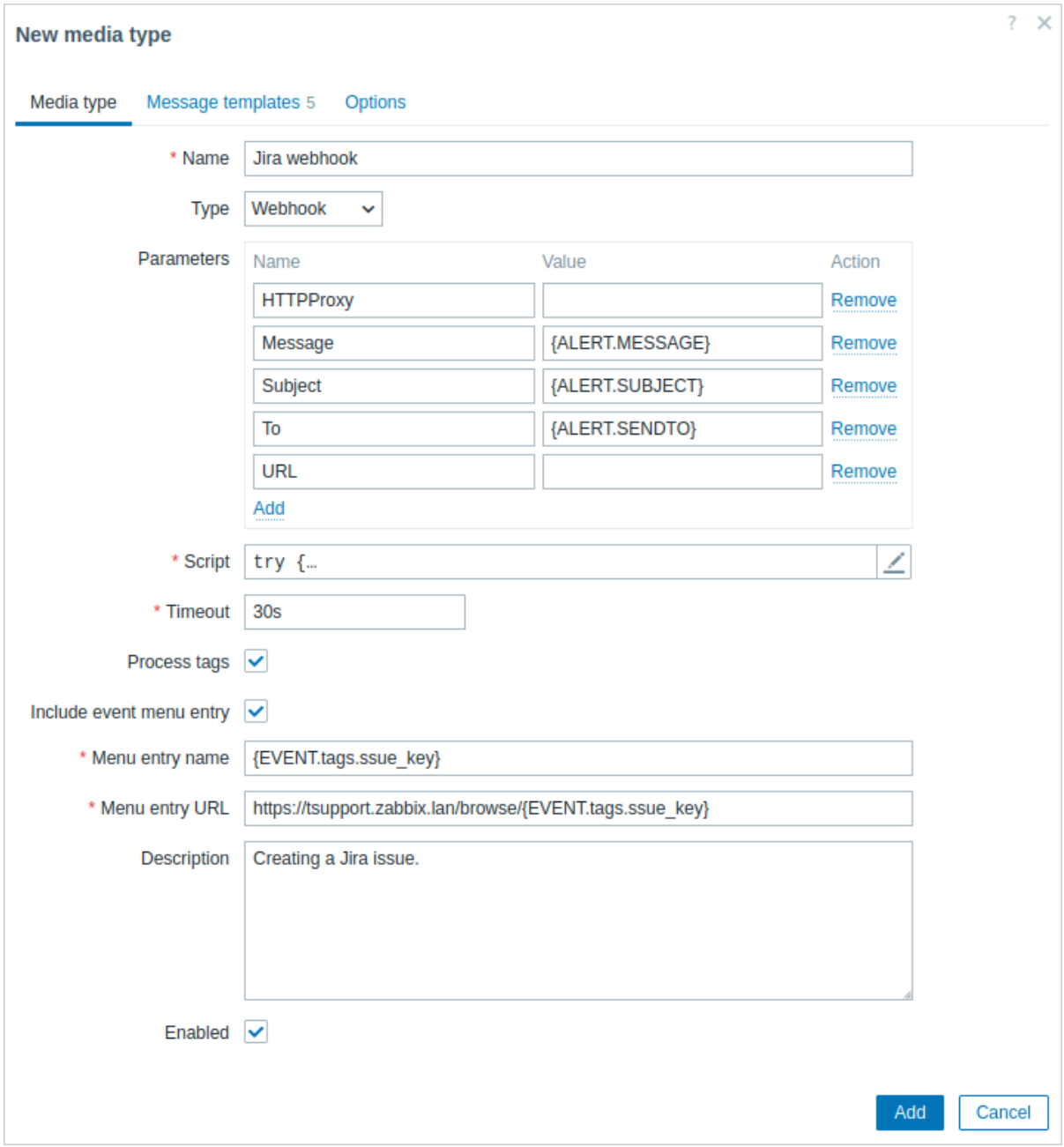
- webhook がチケット\メッセージ ID の保存や更新\解決操作の処理のために webhook tags を使用する場合、単一の障害イベントに対して同じ webhook を複数のアラートアクションで使用しないでください。{EVENT.TAGS.<tag name>} が存在し、webhook 内で更新されると、その結果の値は未定義になります。これを回避するには、更新された値を保存するために、webhook で新しいタグ名を使用してください。これは、Zabbix が提供する Jira、Jira Service Desk、Mattermost、Opsgenie、OTRS、Redmine、ServiceNow、Slack、Zammad、Zendesk の webhook と、*Include event menu entry* オプションを利用するほとんどの webhook に適用されます。ただし、同じ webhook は、同一アクション内の複数の操作またはエスカレーションステップで使用できるほか、異なる条件により同じ障害イベントではトリガーされない別のアクションでも使用できます。
- 内部イベント用のアクションで webhook を使用する場合は、必ず *Custom message* チェックボックスをオンにし、アクション操作の設定でカスタムメッセージを定義してください。そうしないと、通知は送信されません。

1 Webhook スクリプトの例

概要

Zabbix ではすぐに使用できる多数の Webhook 統合が提供されていますが、独自の Webhook を作成することもできます。このセクションでは、カスタム Webhook スクリプト (*Script* パラメーターで使用) の例を示します。他の Webhook パラメーターの説明については、webhookセクションを参照してください。

Jira webhook (カスタム)



このスクリプトは JIRA 課題を作成し、作成された課題に関する情報を返します。

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
                'endpoint': 'jira'
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
```

```
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

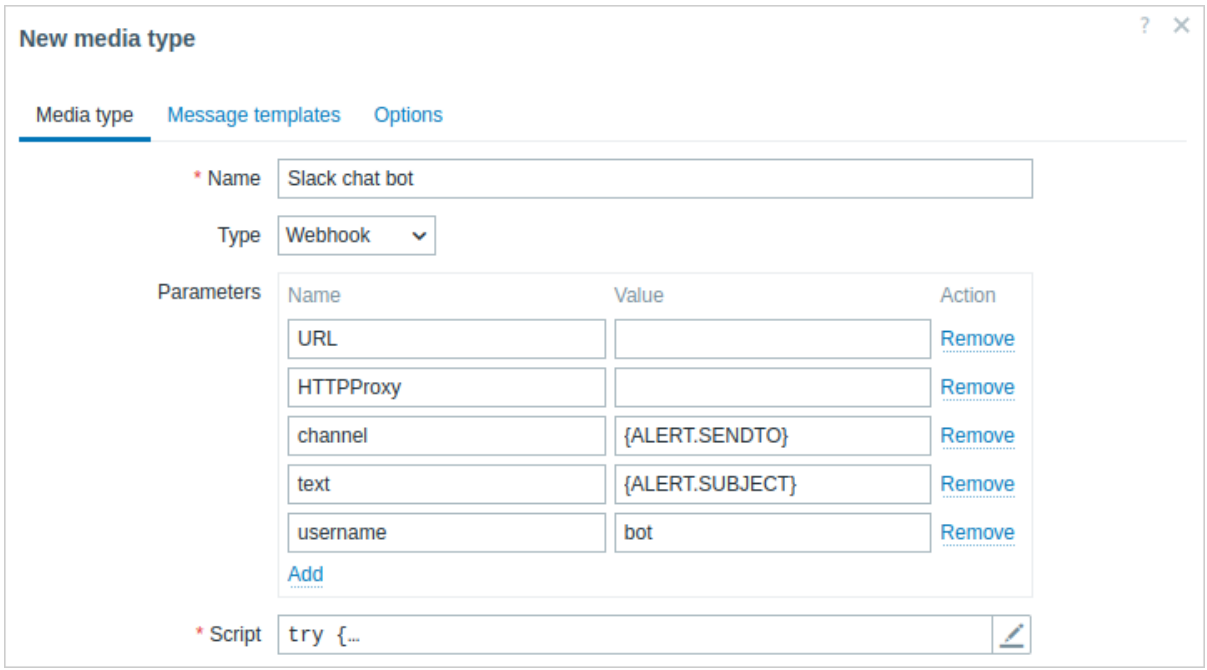
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Slack webhook (カスタム)

この Webhook は、Zabbix から Slack チャネルに通知を転送します。




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
```

```
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);

    response = req.post(params.hook_url, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

2 アクション

概要

イベントの結果として何らかの操作（たとえば通知の送信）を実行したい場合は、アクションを設定する必要があります。

アクションは、サポートされているすべてのタイプのイベントに応じて定義できます。

- トリガーアクション - トリガーのステータスが *OK* から *PROBLEM* に変わり、再び戻るときのイベント用
- サービスアクション - サービスのステータスが *OK* から *PROBLEM* に変わり、再び戻るときのイベント用
- ディスカバリアクション - ネットワークディスカバリが実行されるときのイベント用
- 自動登録アクション - 新しいアクティブエージェントが自動登録されるとき（または登録済みのもののホストメタデータが変更されるとき）のイベント用
- 内部アクション - アイテムが未サポートになったり、トリガーが不明な状態になったりしたときのイベント用

サービスアクションの主な違いは次のとおりです。

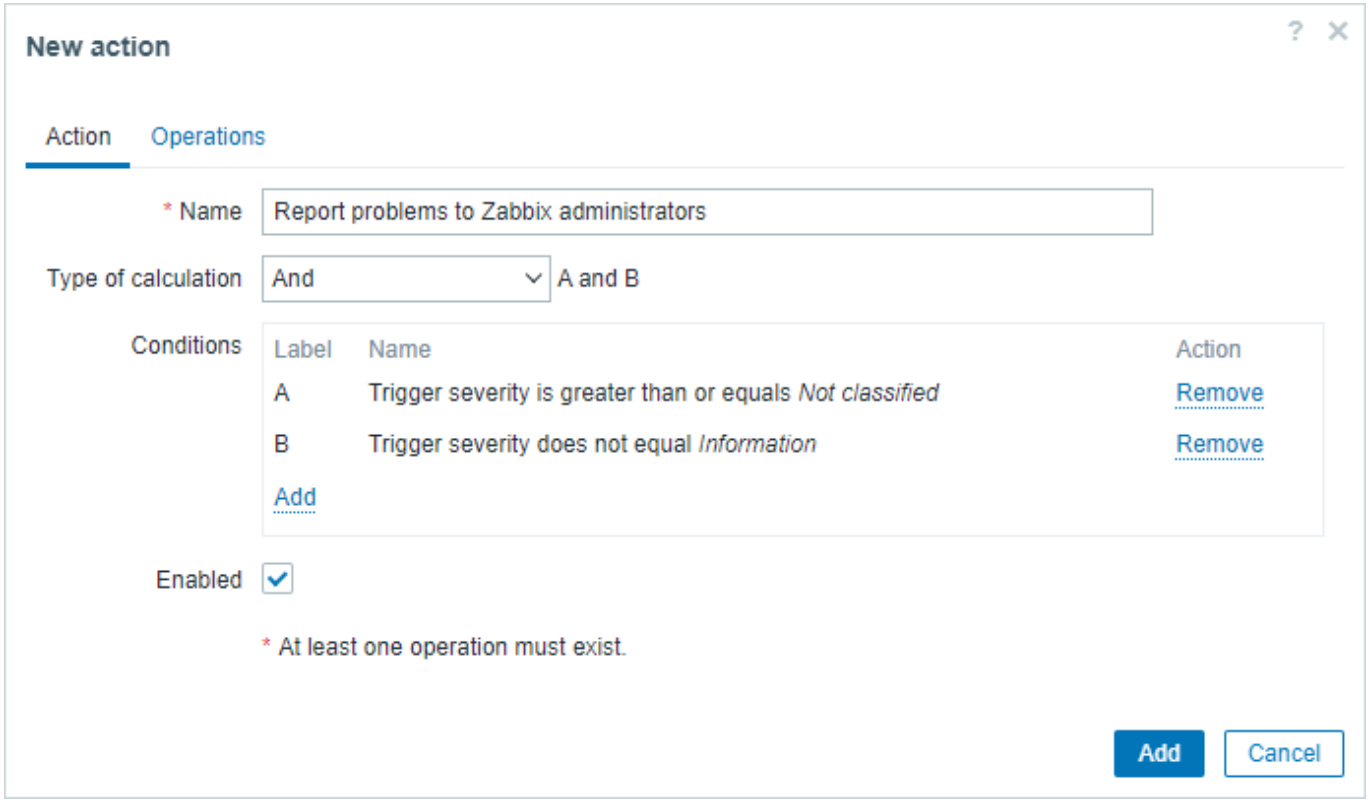
- サービスアクションへのユーザーアクセスは、ユーザーの役割によって付与されたサービスへのアクセス権に依存します
- サービスアクションでは、異なる条件のセットがサポートされます

アクションの設定

アクションを設定するには、次の手順を実行します。

- *Alerts* > *Actions* に移動し、サブメニューから必要なアクションタイプを選択します（タイトルのドロップダウンを使用して、後で別のタイプに切り替えることもできます）。
- *Create action* をクリックします。
- アクションに名前を付けます。
- オペレーションが実行される条件を選択します。
- 実行するオペレーションを選択します。

アクションの一般属性:



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

パラメータ	説明
Name	一意のアクション名。
Type of calculation	アクション条件（複数の条件がある場合）の評価オプションを選択します: **And** - すべての条件を満たす必要があります。 **Or** - いずれか 1 つの条件を満たせば十分です。 **And/Or** - 2 つを組み合わせたものです。異なる条件タイプでは AND、同じ条件タイプでは OR になります。 **Custom expression** - アクション条件を評価するためのユーザー定義の計算式。
Conditions	アクション条件の一覧。 新しい条件を追加するには、*Add* をクリックします。 条件が設定されていない場合、設定中のアクションタイプに対応するすべてのイベントに対してアクションが実行されます。
Enabled	アクションを有効にするには、チェックボックスをオンにします。そうしない場合は無効になります。

1 アクション

概要

イベントが定義済みの一連の条件に一致する場合にのみアクションが実行されるように定義できます。アクションの設定時に条件を設定します。

条件一致では大文字と小文字が区別されます。

トリガーアクション

トリガーベースのアクションでは、以下の条件を使用できます。

条件タイプ	サポートされる演算子	説明
ホストグループ	equals does not equal	ホストグループ、または除外するホストグループを指定します。 **equals** - イベントがこのホストグループに属します。 **does not equal** - イベントがこのホストグループに属しません。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。親グループのみを指定するには、ネストされたすべてのグループを **does not equal** 演算子で追加指定する必要があります。
テンプレート	equals does not equal	テンプレート、または除外するテンプレートを指定します。 **equals** - イベントがこのテンプレートから継承されたトリガーに属します。 **does not equal** - イベントがこのテンプレートから継承されたトリガーに属しません。
ホスト	equals does not equal	ホスト、または除外するホストを指定します。 **equals** - イベントがこのホストに属します。 **does not equal** - イベントがこのホストに属しません。
タグ名	equals does not equal contains does not contain	イベントタグ、または除外するイベントタグを指定します。 **equals** - イベントがこのタグを持ちます。 **does not equal** - イベントがこのタグを持ちません。 **contains** - イベントがこの文字列を含むタグを持ちます。 **does not contain** - イベントがこの文字列を含むタグを持ちません。
タグ値	equals does not equal contains does not contain	イベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。 **equals** - イベントがこのタグと値を持ちます。 **does not equal** - イベントがこのタグと値を持ちません。 **contains** - イベントがこれらの文字列を含むタグと値を持ちます。 **does not contain** - イベントがこれらの文字列を含むタグと値を持ちません。
トリガー	equals does not equal	トリガー、または除外するトリガーを指定します。 **equals** - イベントがこのトリガーによって生成されます。 **does not equal** - イベントが、このトリガーを除く他の任意のトリガーによって生成されます。
イベント名	contains does not contain	トリガーによって生成されたイベント名に含まれる文字列、または除外する文字列を指定します。デフォルトでは、トリガー設定 でカスタムイベント名が指定されていない限り、イベント名はトリガー名と一致します。 **contains** - イベント名にこの文字列が含まれます。 **does not contain** - この文字列はイベント名から除外されます。 注: 入力した値は、すべてのマクロを展開したイベント名と比較されます。
トリガー深刻度	equals does not equal is greater than or equals is less than or equals	トリガー深刻度を指定します。 **equals** - トリガー深刻度と等しい。 **does not equal** - トリガー深刻度と等しくない。 **is greater than or equals** - トリガー深刻度以上。 **is less than or equals** - トリガー深刻度以下。
期間	in not in	期間、または除外する期間を指定します。 **in** - イベント時刻が期間内にあります。 **not in** - イベント時刻が期間内にありません。 形式の説明については、期間指定 ページを参照してください。 ユーザーマクロ をサポートしています。
障害が抑制されている	no yes	ホストメンテナンスにより障害が抑制されている（表示されない）かどうかを指定します。 **no** - 障害は抑制されていません。 **yes** - 障害は抑制されています。

サービスアクション

サービスアクションでは、以下の条件を使用できます。

条件タイプ	サポートされる演算子	説明
サービス	等しい 等しくない	サービス、または除外するサービスを指定します。 等しい - イベントがこのサービスに属します。 等しくない - イベントがこのサービスに属しません。 親サービスを指定すると、すべての子サービスも暗黙的に選択されます。親サービスのみを指定するには、ネストされたすべてのサービスを 等しくない演算子で追加指定する必要があります。
サービス名	含む 含まない	サービス名に含まれる文字列、または除外する文字列を指定します。 含む - 名前にこの文字列を含むサービスによってイベントが生成されます。 含まない - サービス名にこの文字列が見つかりません。
サービスタグ名	等しい 等しくない 含む 含まない	イベントタグ、または除外するイベントタグを指定します。サービスイベントタグは、サービス設定の タグセクションで定義できます。 等しい - イベントがこのタグを持ちます。 等しくない - イベントがこのタグを持ちません。 含む - イベントがこの文字列を含むタグを持ちます。 含まない - イベントがこの文字列を含むタグを持ちません。
サービスタグ値	等しい 等しくない 含む 含まない	イベントタグと値の組み合わせ、または除外するタグと値の組み合わせを指定します。サービスイベントタグは、サービス設定の タグセクションで定義できます。 等しい - イベントがこのタグと値を持ちます。 等しくない - イベントがこのタグと値を持ちません。 含む - イベントがこれらの文字列を含むタグと値を持ちます。 含まない - イベントがこれらの文字列を含むタグと値を持ちません。

Attention:
アラート → メディアタイプメニューで、サービスアクション用のメッセージテンプレートを必ず定義してください。そうしないと、通知は送信されません。

ディスカバリアクション

ディスカバリーベースのイベントでは、以下の条件を使用できます。

条件タイプ	サポートされる演算子	説明
ホスト *IP*	等しい 等しくない	検出されたホストに対して、IP アドレス範囲、または除外する範囲を指定します。 **equals** - ホスト IP がその範囲内にあります。 **does not equal** - ホスト IP がその範囲内にありません。 指定できる形式は以下のとおりです。 単一 IP: 192.168.1.33 IP アドレス範囲: 192.168.1-10.1-254 IP マスク: 192.168.4.0/24 リスト: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 リスト形式ではスペースもサポートされます。
サービス種別	等しい 等しくない	検出されたサービスのサービス種別、または除外するサービス種別を指定します。 **equals** - 検出されたサービスに一致します。 **does not equal** - 検出されたサービスに一致しません。 利用可能なサービス種別: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix エージェント, SNMPv1 エージェント, SNMPv2 エージェント, SNMPv3 エージェント, ICMP ping, telnet。
サービスポート	等しい 等しくない	検出されたサービスの TCP ポート範囲、または除外する範囲を指定します。 **equals** - サービスポートがその範囲内にあります。 **does not equal** - サービスポートがその範囲内にありません。
ディスカバリールール	等しい 等しくない	ディスカバリールール、または除外するディスカバリールールを指定します。 **equals** - このディスカバリールールを使用しています。 **does not equal** - これ以外の任意のディスカバリールールを使用しています。

条件タイプ	サポートされる演算子	説明
ディスカバリーチェック	等しい 等しくない	ディスカバリーチェック、または除外するディスカバリーチェックを指定します。 **equals** - このディスカバリーチェックを使用しています。 **does not equal** - これ以外の任意のディスカバリーチェックを使用しています。
ディスカバリーオブジェクト	等しい	検出されたオブジェクトを指定します。 **equals** - 検出されたオブジェクト（デバイスまたはサービス）と等しい。
ディスカバリーステータス	等しい	**Up** - 「Host Up」および「Service Up」イベントに一致します。 **Down** - 「Host Down」および「Service Down」イベントに一致します。 **Discovered** - 「Host Discovered」および「Service Discovered」イベントに一致します。 **Lost** - 「Host Lost」および「Service Lost」イベントに一致します。
稼働時間/停止時間	以上 以下	「Host Up」および「Service Up」イベントでは稼働時間です。「Host Down」および「Service Down」イベントでは停止時間です。 **is greater than or equals** - 指定値以上です。パラメータは秒単位で指定します。 **is less than or equals** - 指定値以下です。パラメータは秒単位で指定します。
受信値	等しい 等しくない 以上 以下 含む 含まない	ディスカバリールール内のエージェント（Zabbix、SNMP）チェックから受信した値を指定します。文字列比較です。1 つのルールに複数の Zabbix エージェントまたは SNMP チェックが設定されている場合、それぞれの受信値が確認されます（各チェックは新しいイベントを生成し、そのイベントがすべての条件に対して照合されます）。 **equals** - 値と等しい。 **does not equal** - 値と等しくない。 **is greater than or equals** - 値以上。 **is less than or equals** - 値以下。 **contains** - 部分文字列を含みます。パラメータは文字列として指定します。 **does not contain** - 部分文字列を含みません。パラメータは文字列として指定します。
プロキシ	等しい 等しくない	プロキシ、または除外するプロキシを指定します。 **equals** - このプロキシを使用しています。 **does not equal** - これ以外の任意のプロキシを使用しています。

Note:
ディスカバリールール内のサービスチェックで、ディスカバリーイベントを生成するものは、同時には実行されません。したがって、アクション内の `Service type`、`Service port`、または `Received value` 条件に 複数の値が設定されている場合、それらは一度に 1 つのディスカバリーイベントと比較され、複数のイベントと同時に比較されることは ありません。その結果、同じチェックタイプに対して複数の値を持つアクションは、正しく実行されない可能性があります。

自動登録アクション

次の条件は、アクティブなエージェントの自動登録に基づくアクションで使用できます。

タイプ	オペレータ	説明
ホストメタデータ	含む 含まない 一致する 一致しない	ホストのメタデータまたは除外するホストのメタデータを指定します。 含む - ホストメタデータには指定文字列が含まれる。 含まない - ホストメタデータに指定文字列が含まれない。 ホストのメタデータはエージェント設定ファイルで指定できます。 一致する - ホストメタデータは正規表現と一致する。 一致しない - ホストメタデータは正規表現と一致しない。
ホスト名	含む 含まない 一致する 一致しない	ホスト名または除外するホスト名を指定します。 含む - ホスト名に文字列が含まれる。 含まない - ホスト名に文字列が含まれない。 ** 一致する ** - ホスト名が正規表現に一致する。 一致しない - ホスト名が正規表現に一致しない。

タイプ	オペレータ	説明
プロキシ	等しい 等しくない	プロキシまたは除外するプロキシを指定します。 等しい - このプロキシを使用します。 等しくない - このプロキシ以外の他のプロキシを使用します。

内部イベントアクション

内部イベントに基づくアクションには、次の条件を設定できます。

タイプ	オペレータ	説明
イベントタイプ	等しい	アイテムが”取得不可”の状態 - アイテムが’通常’から’サポートされていない’に移行するイベントに一致します ローレベルディスカバリが”取得不可”の状態 - ローレベルディスカバリルールが’通常’から’サポートされていない’に移行するイベントに一致します トリガーが”不明”の状態 - トリガーが’通常’から’不明’に移行するイベントに一致します
ホストグループ	等しい 等しくない	ホストグループまたは除外するホストグループを指定。 等しい - イベントはこのホストグループに属します。 等しくない - イベントはこのホストグループに属しません。
タグ名	等しい 等しくない 含む 含まない	イベントタグまたは除外するイベントタグを指定。 等しい - イベントにこのタグがある 等しくない - イベントにこのタグが無い 含む - イベントにこの文字列を含むタグがある * * 含まない ** - イベントにこの文字列を含むタグがない
タグの値	等しい 等しくない 含む 含まない	タグと値の組み合わせ、または除外するイベントタグと値の組み合わせを指定します。 等しい - イベントにこのタグと値がある 等しくない - イベントにこのタグと値がない 含む - イベントにこれらの文字列を含むタグと値がある 含まない - イベントにこれらの文字列を含むタグと値がない
テンプレート	等しい 等しくない	テンプレートまたは除外するテンプレートを指定します。 等しい - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスカバリルールに属する 等しくない - イベントはこのテンプレートから継承されたアイテム/トリガー/ローレベルディスカバリルールに属さない
ホスト	等しい 等しくない	ホストまたは除外するホストを指定します。 等しい - イベントはこのホストに属する 等しくない - イベントはこのホストに属さない

計算の種類

条件の計算方法には、以下のオプションがあります。

- **And** - すべての条件を満たす必要があります

複数のトリガーが `Trigger=` 条件として選択されている場合、それらの間で”And” 計算を使用することはできません。アクションは、1 つのトリガーのイベントに基づいてのみ実行できます。

- **Or** - 1 つの条件を満たせば十分です
- **And/Or** - 2 つの組み合わせです。異なる条件タイプの間では AND、同じ条件タイプの間では OR になります。例:

ホストグループが Oracle servers と等しい
ホストグループが MySQL servers と等しい
イベント名に’Database is down’ が含まれる
イベント名に’Database is unavailable’ が含まれる

これは次のように評価されます。

(ホストグループが Oracle servers と等しい **or** ホストグループが MySQL servers と等しい**) and (**イベント名に’Database is down’ が含まれる **or** イベント名に’Database is unavailable’ が含まれる**)**

- **Custom expression** - アクション条件を評価するためのユーザー定義の計算式です。すべての条件（大文字の A、B、C、... で表されます）を含める必要があり、スペース、タブ、括弧 ( )、**and**（大文字小文字を区別）、**or**（大文字小文字を区別）、**not**（大文字小文字を区別）を含めることができます。

前の And/Or の例は (A or B) and (C or D) と表されますが、カスタム式では他にも複数の計算方法を使用できます。

(A and B) and (C or D)
(A and B) or (C and D)
((A or B) and C) or D
(not (A or B) and C) or not D
など。

アクションで使用されているオブジェクトの削除

アクションの条件/実行内容で使用されているオブジェクトを削除しようとすると、ユーザーには対応するエラーメッセージが表示されます。

アクションの条件または実行内容が以下に該当する場合でも、アクションは無効になりません。

- 削除されたオブジェクトがある場合
- ローレベルディスカバリによって検出されなくなり、削除されているオブジェクトがある場合（*Delete lost resources* 期間後）

オブジェクトが存在しない場合でも、アクションの条件/実行内容は削除されません。代わりに、削除されたオブジェクトは *Deleted host group*、*Deleted host*、*Deleted trigger* などと表示されます。アクション一覧の「Info」列には、「This action has conditions or operations referencing deleted object(s).」というメッセージ付きの警告アイコンが表示されます。

2 実行内容

概要

すべてのイベントに対して、以下の実行内容を定義できます。

- メッセージを送信
- リモートコマンドを実行

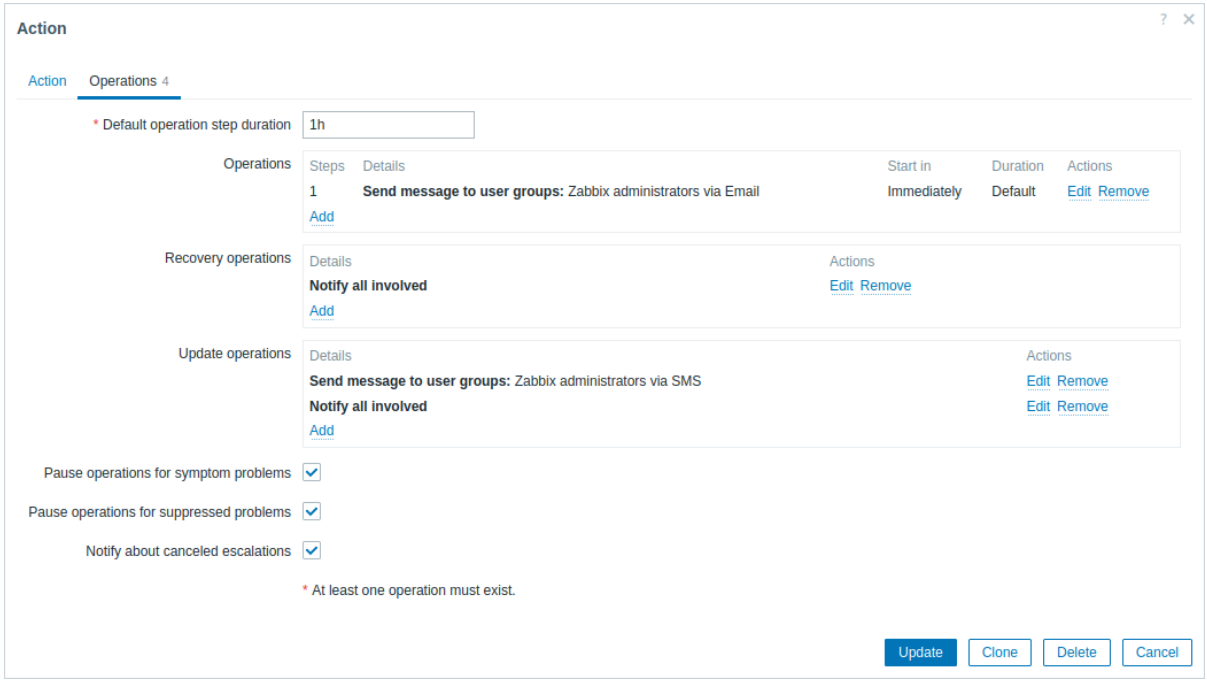
Attention:
アクションの実行内容の受信者として定義されたユーザーに対して、ホストへのアクセスが明示的に「拒否」されている場合、またはそのユーザーにそのホストに対する権限がまったく定義されていない場合、Zabbix サーバーはアラートを作成しません。

ディスカバリイベントおよび自動登録イベントでは、追加の実行内容を利用できます。

- ホストを追加
- ホストを削除
- ホストを有効化
- ホストを無効化
- ホストグループに追加
- ホストグループから削除
- ホストタグを追加
- ホストタグを削除
- テンプレートをリンク
- テンプレートのリンクを解除
- ホストインベントリモードを設定

実行内容の設定

実行内容を設定するには、アクション設定の 実行内容タブに移動します。



実行内容の一般的な属性:

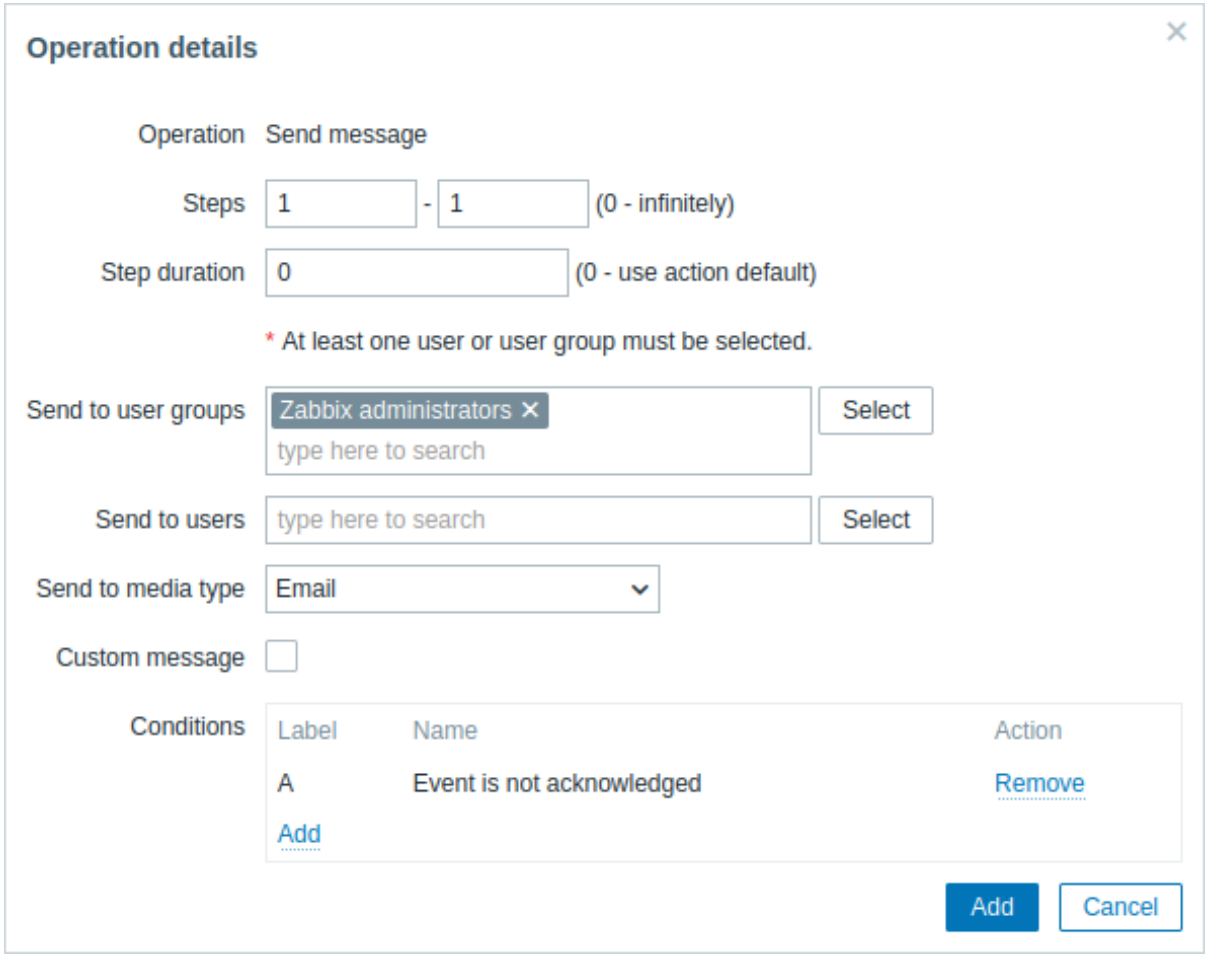
Parameter	Description
Default operation step duration	1 つの実行ステップのデフォルトの期間です（60 秒～1 週間）。 例えば、ステップ期間が 1 時間の場合、ある実行内容が実行されると、次のステップまで 1 時間経過します。 例えば 60s、1m、2h、1d のように、時間のサフィックスを使用できます。 ユーザーマクロを使用できます。
Operations	アクションの実行内容（存在する場合）が、次の詳細とともに表示されます。 **Steps** - 実行内容が割り当てられているエスカレーションステップ。 **Details** - 実行内容の種類とその受信者/対象。 実行内容の一覧には、使用されるメディアタイプ（メール、SMS、またはスクリプト）に加え、通知の受信者の名前と姓（ユーザー名の後ろに括弧付き）も表示されます。 **Start in** - イベント発生後、どれくらいで実行内容が実行されるか。 **Duration (sec)** - ステップ期間が表示されます。ステップがデフォルト期間を使用する場合は *Default*、カスタム期間を使用する場合は時間が表示されます。 **Actions** - 実行内容を編集および削除するためのリンクが表示されます。
Recovery operations	アクションの実行内容（存在する場合）が、次の詳細とともに表示されます。 **Details** - 実行内容の種類とその受信者/対象。 実行内容の一覧には、使用されるメディアタイプ（メール、SMS、またはスクリプト）に加え、通知の受信者の名前と姓（ユーザー名の後ろに括弧付き）も表示されます。 **Actions** - 実行内容を編集および削除するためのリンクが表示されます。
Update operations	アクションの実行内容（存在する場合）が、次の詳細とともに表示されます。 **Details** - 実行内容の種類とその受信者/対象。 実行内容の一覧には、使用されるメディアタイプ（メール、SMS、またはスクリプト）に加え、通知の受信者の名前と姓（ユーザー名の後ろに括弧付き）も表示されます。 **Actions** - 実行内容を編集および削除するためのリンクが表示されます。
Pause operations for symptom problems	このチェックボックスをオンにすると、症状問題に対する実行内容（最初の実行内容の後）が一時停止されます。 この設定は問題のエスカレーションにのみ影響し、復旧時および更新時の実行内容には影響しないことに注意してください。 このオプションは トリガーアクションでのみ使用できます。
Pause operations for suppressed problems	このチェックボックスをオンにすると、メンテナンス期間中は実行内容の開始が遅延されます。メンテナンス終了後に実行内容が開始されると、メンテナンス中のイベントに対するものを含め、すべての実行内容が実行されます。 この設定は問題のエスカレーションにのみ影響し、復旧時および更新時の実行内容には影響しないことに注意してください。 このチェックボックスをオフにすると、メンテナンス期間中であっても実行内容は遅延なく実行されます。 このオプションは サービスアクションでは使用できません。

Parameter	Description
Notify about canceled escalations	このチェックボックスをオフにすると、キャンセルされたエスカレーションに関する通知（ホスト、アイテム、トリガー、またはアクションが無効化された場合）が無効になります。

必須入力フィールドには、赤いアスタリスクが付いています。

新しい実行内容の詳細を設定するには、実行内容ブロック内の Add をクリックします。既存の実行内容を編集するには、実行内容の横にある Edit をクリックします。実行ステップの詳細を編集できるポップアップウィンドウが開きます。

操作の詳細



パラメータ	説明
操作	操作を選択します: メッセージの送信 - ユーザーにメッセージを送信します。 < リモートコマンド名 > - リモートコマンドを実行します。コマンドは、事前にグローバルスクリプトでスコープとしてアクションの操作を選択して定義されている場合に実行できます。 ディスカバリおよび自動登録ベースのイベントでは、さらに多くの操作を利用できます(上記を参照)。
ステップ	エスカレーションスケジュールで操作を割り当てるステップを選択します: 開始 - このステップから実行します。 終了 - このステップまで実行します（0= 無制限、実行は制限されません）。

パラメータ		説明
ステップの期間操作タイプ: メッセージの送信		これらのステップのカスタム期間です（0= デフォルトのステップ期間を使用）。 時間のサフィックスをサポートしています。例: 60s, 1m, 2h, 1d。 ユーザーマクロをサポートしています。 複数の操作を同じステップに割り当てることができます。これらの操作に異なるステップ期間が定義されている場合は、最も短いものが考慮され、そのステップに適用されます。
	送信先ユーザーグループ	メッセージの送信先となるユーザーグループを選択します。 通知を受けるには、そのユーザーグループがホストに対して少なくとも「read」の権限を持っている必要があります。
	送信先ユーザー	メッセージの送信先となるユーザーを選択します。 通知を受けるには、そのユーザーがホストに対して少なくとも「read」の権限を持っている必要があります。
	送信先メディアタイプ	利用可能なすべてのメディアタイプ（設定済みかつ有効）または特定の 1 つのみにメッセージを送信します。
	カスタムメッセージ	選択すると、カスタムメッセージを設定できます。 webhook経由の内部イベントに関する通知では、カスタムメッセージは必須です。

パラメータ		説明
	件名	カスタムメッセージの件名です。件名にはマクロを含めることができます。255 文字までに制限されます。
	メッセージ	カスタムメッセージです。メッセージにはマクロを含めることができます。データベースの種類に応じて文字数の上限が異なります（詳細はメッセージの送信を参照してください）。
操作タイプ: リモートコマンド		
	ターゲットリスト	コマンドを実行する対象を選択します: 現在のホスト - 問題イベントの原因となったトリガーのホストでコマンドを実行します。トリガーに複数のホストが含まれている場合、このオプションは機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。そのため、リモートコマンドはネストされたグループ内のホストでも実行されます。 ホスト上のコマンドは、そのホストが複数回一致した場合でも 1 回だけ実行されます (例: 複数のホストグループから一致した場合、または個別指定とホストグループ指定の両方で一致した場合)。 Zabbix サーバー上でカスタムスクリプトを実行する場合、ターゲットリストには意味がありません。この場合により多くのターゲットを選択しても、スクリプトがサーバー上でその回数だけ多く実行されるだけです。 グローバルスクリプトについては、ターゲットの選択はグローバルスクリプトの設定にあるホストグループ設定にも依存することに注意してください。 ターゲットリストオプションは、サービスアクションでは利用できません。この場合、リモートコマンドは常に Zabbix サーバー上で実行されるためです。
条件		操作を実行する条件です: イベントが未確認 - イベントが未確認の場合のみ。 イベントが確認済み - イベントが確認済みの場合のみ。 条件オプションは、トリガーアクションでのみ利用できます。

完了したら、追加をクリックして操作を操作のリストに追加します。

1 メッセージの送信

概要

メッセージの送信は、障害について人々に通知する最良の方法の 1 つです。これが、Zabbix が提供する主要なアクションの 1 つである理由です。

設定

Zabbix から通知を送受信できるようにするには、次の設定が必要です。

- メッセージの送信先となるメディアを定義する

ユーザー設定で選択したメディアに対して定義された有効な期間外で実行される操作の場合、メッセージは送信されません。

ディスカバリ、アクティブエージェントの自動登録、内部イベントなど、トリガー以外のイベントの通知を受け取りたい場合は、ユーザーメディアの設定でデフォルトのトリガー深刻度（「未分類」）を必ずチェックする必要があります。

- 定義済みのいずれかのメディアにメッセージを送信するアクションの実行内容を設定する

Attention:
Zabbix は、イベントを生成したホストに対して少なくとも「読み取り」権限を持つユーザーにのみ通知を送信します。トリガー式に含まれる少なくとも 1 つのホストにアクセスできる必要があります。

エスカレーションを使用して、メッセージ送信のカスタムシナリオを設定できます。

Zabbix からのメールを正常に受信して読めるようにするには、Zabbix が UTF-8 データを送信するため、メールサーバー/クライアントは標準の「SMTP/MIME email」形式をサポートしている必要があります（件名に ASCII 文字のみが含まれる場合は、UTF-8 エンコードされません）。件名とメッセージ本文は、「SMTP/MIME email」形式標準に従うため、base64 エンコードされます。

すべてのマクロ展開後のメッセージ制限は、リモートコマンドのメッセージ制限と同じです。

メッセージの追跡

送信されたメッセージのステータスは、監視 → 障害で確認できます。

アクション列では、実行されたアクションの要約情報を確認できます。ここでは、緑色の数字は送信されたメッセージを表し、赤色の数字は送信に失敗したメッセージを表します。進行中は、アクションが開始されていることを示します。失敗は、どのアクションも正常に実行されなかったことを示します。

イベントの詳細を表示するためにイベント時刻をクリックすると、アクションブロックで、そのイベントにより送信された（または送信されなかった）メッセージの詳細を確認できます。

レポート → アクションログでは、アクションが設定されているイベントについて、実行されたすべてのアクションの詳細を確認できます。

2 リモートコマンド

概要

リモートコマンドを使用すると、特定の定義済みコマンドが監視対象ホストで何らかの条件で自動的に実行されるように定義できます。

リモートコマンドはスマートプロアクティブモニタリングの強力なメカニズムです。

この機能の最も明白な使用法で、次のことを試みることができます。

- 一部のアプリケーション (Web サーバー、ミドルウェア、CRM) が応答しない場合、自動的に再起動する
- 要求に応答しない場合、IPMI の'reboot' コマンドを使用してリモートサーバーを再起動する
- ディスク容量が不足した場合に自動でディスク容量を解放する (古いファイルの削除、/tmp のクリーニング)
- CPU 負荷に応じて、ある物理ボックスから別の物理ボックスに VM を移行する
- 不十分な CPU (またはディスク、メモリなど) リソースに新しいノードをクラウド環境に追加する

リモートコマンドのアクションの設定は、メッセージを送信する場合と似ていますが、Zabbix がメッセージを送信する代わりにコマンドを実行する点だけが異なります。

リモートコマンドは、Zabbix サーバー、プロキシ、またはエージェントによって実行できます。Zabbix エージェントのリモートコマンドは、Zabbix サーバーまたは Zabbix プロキシ経由で直接実行できます。Zabbix エージェントと Zabbix プロキシの両方で、リモートコマンドはデフォルトで無効になっています。それらは次の方法で有効にできます。

- エージェント設定に `AllowKey=system.run[*]` パラメーターを追加します。
- プロキシ設定で EnableRemoteCommands パラメータを'1' に設定します。

Zabbix サーバーによって実行されるリモートコマンドは、コマンドの実行 で説明されているように、終了コードのチェックを含めて実行されます。

対象ホストがメンテナンス中であってもリモートコマンドを実行します。

リモートコマンドの制限

すべてのマクロを解決した後のリモートコマンドの制限は、データベースの種類と文字セットに依存します（非 ASCII 文字は保存に 1 バイトを超える領域を必要とします）：

データベース	文字数の制限	バイト数の制限
MySQL	65535	65535

データベース	文字数の制限	バイト数の制限
PostgreSQL	65535	制限なし
SQLite（Zabbix プロキシのみ）	65535	制限なし

リモートコマンド実行の出力（戻り値）は 16MB に制限されています（切り捨てられる末尾の空白文字を含む）。IPMI リモートコマンドの制限は、インストールされている IPMI ライブラリに基づきます。なお、データベースの制限はすべてのリモートコマンドに適用されます。

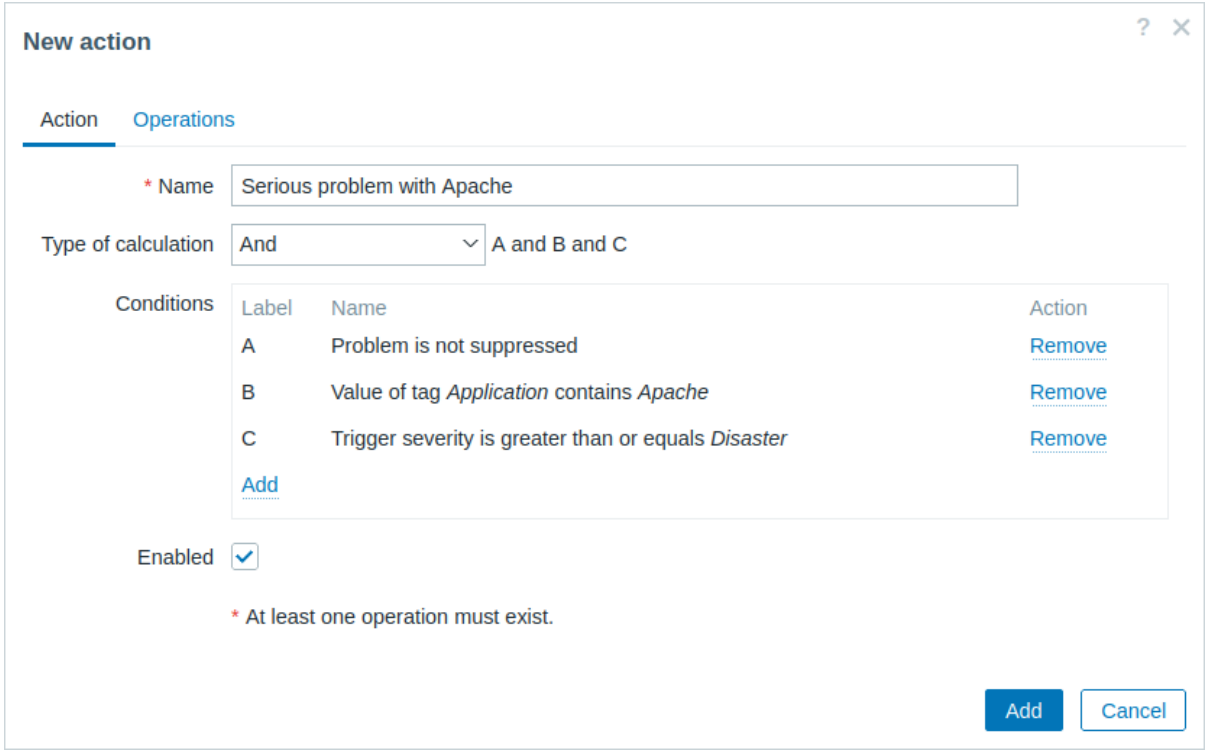
設定

Zabbix エージェント上で実行されるこれらのリモートコマンド（カスタムスクリプト）は、まずエージェントの設定で有効にする必要があります。

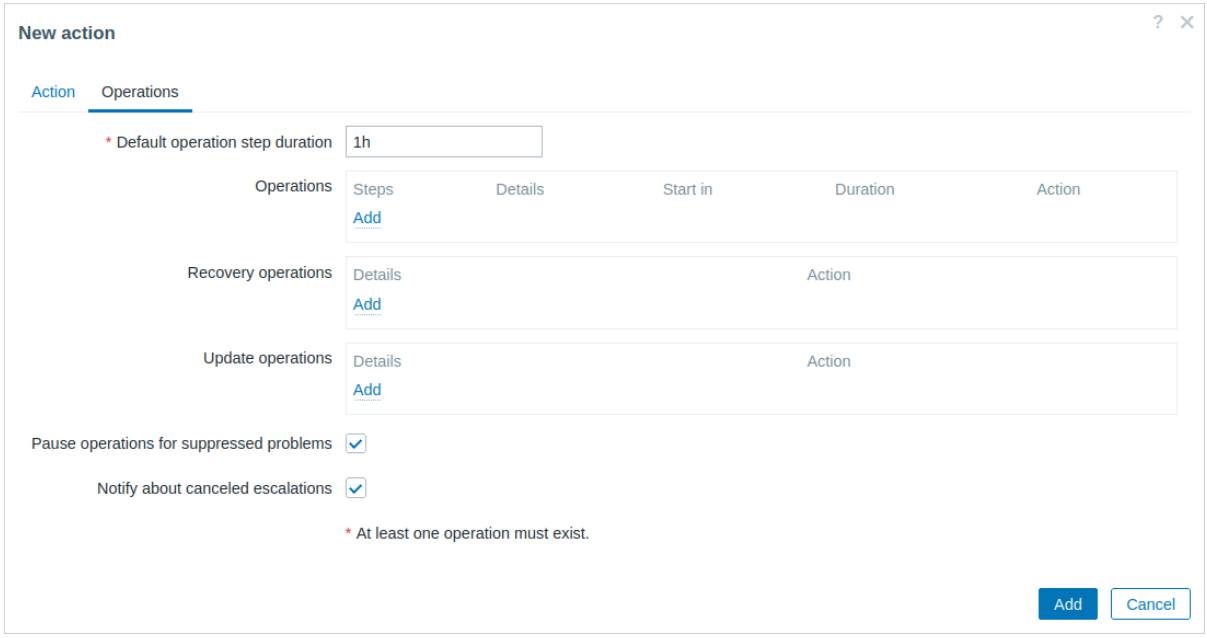
nowait モードで特定のコマンドを許可するため､許可する各コマンドについて､エージェント設定に AllowKey=system.run[<command>,*] パラメータが追加されていることを確認してください。このパラメータを変更した場合は、エージェントデーモンを再起動してください。

次に、アラート → アクション → トリガーアクションで新しいアクションを設定する際は、以下を行います。

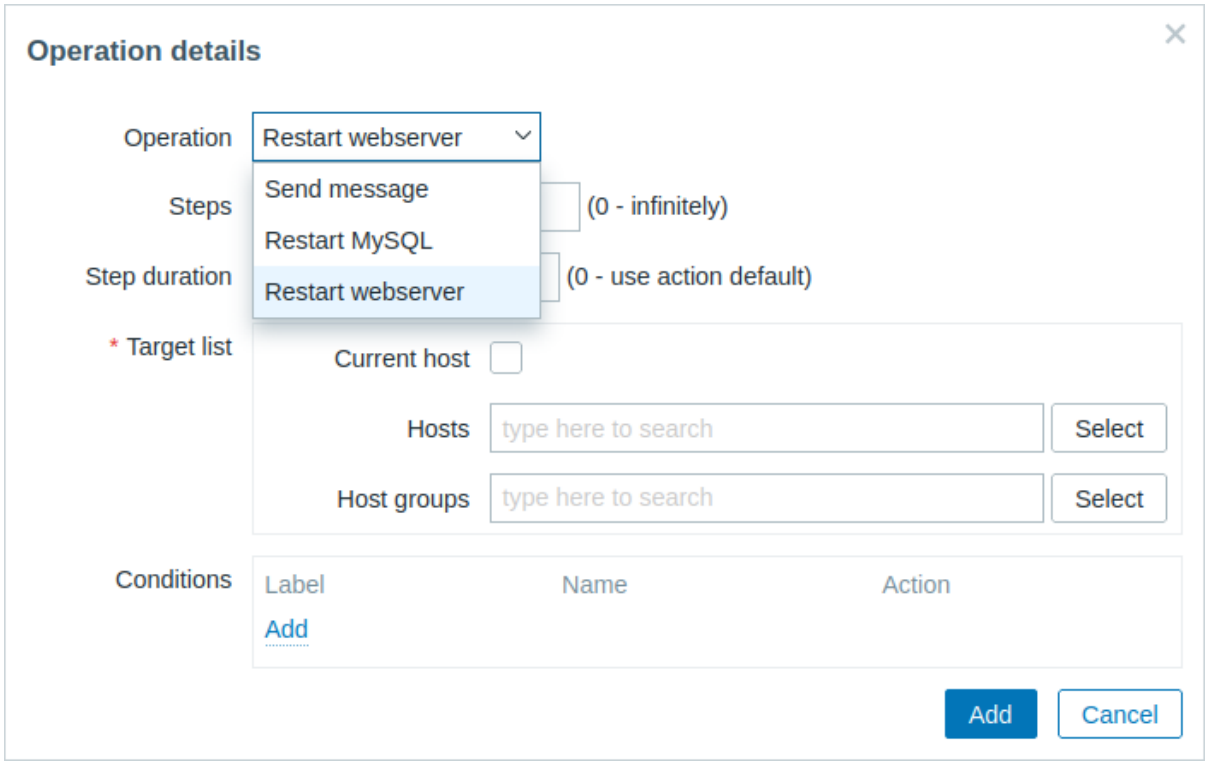
1. 適切な条件を定義します。例えば、Apache アプリケーションのいずれかで発生したすべての障害レベルの問題に対してアクションが実行されるよう設定します。



2. 実行内容タブで、実行内容、復旧時の実行内容、または更新時の実行内容ブロックの 追加をクリックします。



3. 実行内容ドロップダウンリストから定義済みスクリプトのいずれかを選択し、そのスクリプトの 対象リストを設定します。



定義済みスクリプト

アクションの実行操作（webhook、スクリプト、SSH、Telnet、IPMI）で利用可能なスクリプトは、グローバルスクリプトで定義されます。

例:

```
sudo /etc/init.d/apache restart
```

この場合、Zabbix は Apache プロセスの再起動を試みます。このコマンドでは、コマンドが Zabbix エージェント上で実行されるようにしてください（実行先に対して *Zabbix agent* ボタンをクリックします）。

Attention:
sudo の使用に注意してください。Zabbix ユーザーには、デフォルトではシステムサービスを再起動する権限がありません。
sudo の設定方法のヒントについては、以下を参照してください。

Note:
Zabbix agent 7.0 以降では、アクティブモードで動作しているエージェント上でもリモートコマンドを実行できます。アクティブまたはパッシブを問わず、Zabbix エージェントはリモートホスト上で実行されている必要があり、コマンドはバックグラウンドで実行されます。

Zabbix エージェント上のリモートコマンドは、system.run[,nowait] キーによってタイムアウトなしで実行され、実行結果は確認されません。
Zabbix サーバーおよび Zabbix プロキシ上のリモートコマンドは、zabbix_server.conf または zabbix_proxy.conf ファイルのTrapperTimeout パラメータで設定されたタイムアウト付きで実行され、実行結果が確認されます。
追加情報については、スクリプトのタイムアウトを参照してください。

アクセス権限

'zabbix' ユーザーが、設定されたコマンドに対する実行権限を持っていることを確認してください。特権コマンドへのアクセスを許可するために、**sudo** の使用を検討するとよいでしょう。アクセスを設定するには、root として次を実行します:

```
visudo
```

sudoers ファイルで使用できる設定例:

```
# 'zabbix' user にパスワードなしですべてのコマンドの実行を許可します。
zabbix ALL=NOPASSWD: ALL

# 'zabbix' user にパスワードなしで apache の再起動を許可します。
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
一部のシステムでは、*sudoers* ファイルによりローカル以外のユーザーがコマンドを実行できないようになっています。これを変更するには、*/etc/sudoers* の **requiretty** オプションをコメントアウトしてください。

複数のインターフェースを持つ場合のリモートコマンド

対象システムに選択したタイプ（Zabbix エージェントまたは IPMI）の複数のインターフェースがある場合、リモートコマンドはデフォルトのインターフェースで実行されます。

SSH および Telnet 経由のリモートコマンドは、Zabbix エージェントのものとは別のインターフェースを使用して実行できます。使用可能なインターフェースは、次の順序で選択されます。

- Zabbix エージェントのデフォルトインターフェース
- SNMP のデフォルトインターフェース
- JMX のデフォルトインターフェース
- IPMI のデフォルトインターフェース

IPMI リモートコマンド

IPMI リモートコマンドの場合、次の構文を使用する必要があります。

```
<command> [<value>]
```

内容

- <command> - スペースなしの 1 つの IPMI コマンド
- <value> - 'on', 'off' または任意の符号なし整数。<value> はオプションパラメーターです。

例

アクションの実行内容でリモートコマンドとして使用できるグローバルスクリプトの例。

例 **1**

特定の条件での Windows の再起動。

Zabbix によって検出された問題発生時に Windows を自動的に再起動するには、次のスクリプトを定義します。

スクリプトパラメータ	値
スコープ	'アクションの実行'
タイプ	'スクリプト'
コマンド	c:\windows\system32\shutdown.exe -r -f

例 **2**

IPMI 制御を使用してホストを再起動します。

スクリプトパラメータ	値
スコープ	' アクションの実行'
タイプ	'IPMI'
コマンド	reset

例 **3**

IPMI 制御を使用してホストの電源をオフにします。

スクリプトパラメータ	値
スコープ	' アクションの実行'
タイプ	'IPMI'
コマンド	power off

3 追加操作

概要

このセクションでは、ディスカバリ/自動登録イベントの追加操作の詳細を確認できます。

ホストの追加

ホストが検出されるとディスカバリプロセスの最後まで待たず、ディスカバリプロセス中すぐに追加されます。

Note:
多くの利用できないホスト/サービスが原因で、ネットワークの検出に時間がかかることがあるので、忍耐力と妥当な IP 範囲を使用することをお勧めします。

ホストを追加するとき、その名前は標準の **gethostbyname** 関数によって決定されます。ホストが解決できる場合は、解決された名前が使用されます。そうでない場合は、IP アドレスが使用されます。また、ホスト名に IPv6 アドレスが使用されている場合は、ホスト名にコロンが使用できないため、すべての":" (コロン) は"_"(アンダースコア) に置き換えられます。

Attention:
プロキシによるディスカバリを実行する場合でも、ホスト名の検索は Zabbix サーバー上で行われます。

Attention:
新しく検出されたホストと同じ名前のホストがすでに Zabbix 設定に存在する場合、Zabbix はホスト名に **_N** を追加します。**N** は 2 から始まる数字で、増えていきます。

4 メッセージでのマクロ使用

概要

メッセージの件名とメッセージテキストでは、マクロを使用して障害をより効率的に報告できます。

多くの組み込みマクロに加えて、ユーザーマクロ と式マクロ もサポートされています。Zabbix でサポートされているマクロの完全なリスト が利用可能です。

例

以下の例は、メッセージでマクロを使用する方法を示しています。

例 1

メッセージの件名:

```
Problem: {TRIGGER.NAME}
```

メッセージを受信すると、メッセージの件名は次のように置き換えられます。

```
Problem: Processor load is too high on Zabbix server
```

例 2

メッセージ本文:

```
Processor load is: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

メッセージは受信時に次のように置き換えられます。

```
Processor load is: 1.45
```

例 3

メッセージ本文:

```
Latest value: {?last(/{HOST.HOST}/{ITEM.KEY})}
MAX for 15 minutes: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MIN for 15 minutes: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

メッセージは受信時に次のように置き換えられます。

```
Latest value: 1.45
MAX for 15 minutes: 2.33
MIN for 15 minutes: 1.01
```

例 4

メッセージ本文:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

メッセージを受信すると、本文にイベントの詳細ページへのリンクが含まれます。このページには、イベント、そのトリガー、および同じトリガーによって生成された最新のイベントのリストに関する情報が表示されます。

例 5

トリガー式で複数のホストからの値を通知します。

メッセージ本文:

```
Problem name: {TRIGGER.NAME}
Trigger expression: {TRIGGER.EXPRESSION}

1. Item value on {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Item value on {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

メッセージを受信すると、メッセージは次のように置き換えられます:

```
Problem name: Processor load is too high on a local host
Trigger expression: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[perc

1. Item value on Myhost: 0.83 (Processor load (1 min average per core))
2. Item value on Myotherhost: 5.125 (Processor load (1 min average per core))
```

例 6

recoveryメッセージで障害イベントと復旧イベントの両方の詳細を受信する:

```
Message:

Problem:

Event ID: {EVENT.ID}
Event value: {EVENT.VALUE}
Event status: {EVENT.STATUS}
Event time: {EVENT.TIME}
Event date: {EVENT.DATE}
Event age: {EVENT.AGE}
Event acknowledgment: {EVENT.ACK.STATUS}
Event update history: {EVENT.UPDATE.HISTORY}

Recovery:

Event ID: {EVENT.RECOVERY.ID}
```

```
Event value: {EVENT.RECOVERY.VALUE}
Event status: {EVENT.RECOVERY.STATUS}
Event time: {EVENT.RECOVERY.TIME}
Event date: {EVENT.RECOVERY.DATE}
Operational data: {EVENT.OPDATA}
```

メッセージを受信すると、次のように置き換えられます:

```
Problem:

Event ID: 21874
Event value: 1
Event status: PROBLEM
Event time: 13:04:30
Event date: 2018.01.02
Event age: 5m
Event acknowledgment: Yes
Event update history: 2018.01.02 13:05:51 "John Smith (Admin)"
Actions: acknowledged.

Recovery:

Event ID: 21896
Event value: 0
Event status: OK
Event time: 13:10:07
Event date: 2018.01.02
Operational data: Current value is 0.83
```

3 復旧時実行

概要

復旧操作により、障害が解決されたときに通知を受けることができます。

復旧操作では、メッセージとリモートコマンドの両方がサポートされています。複数の操作を追加できますが、エスカレーションはサポートされていません。すべての操作が 1 つのステップに割り当てられるため、同時に実行されます。

ユースケース

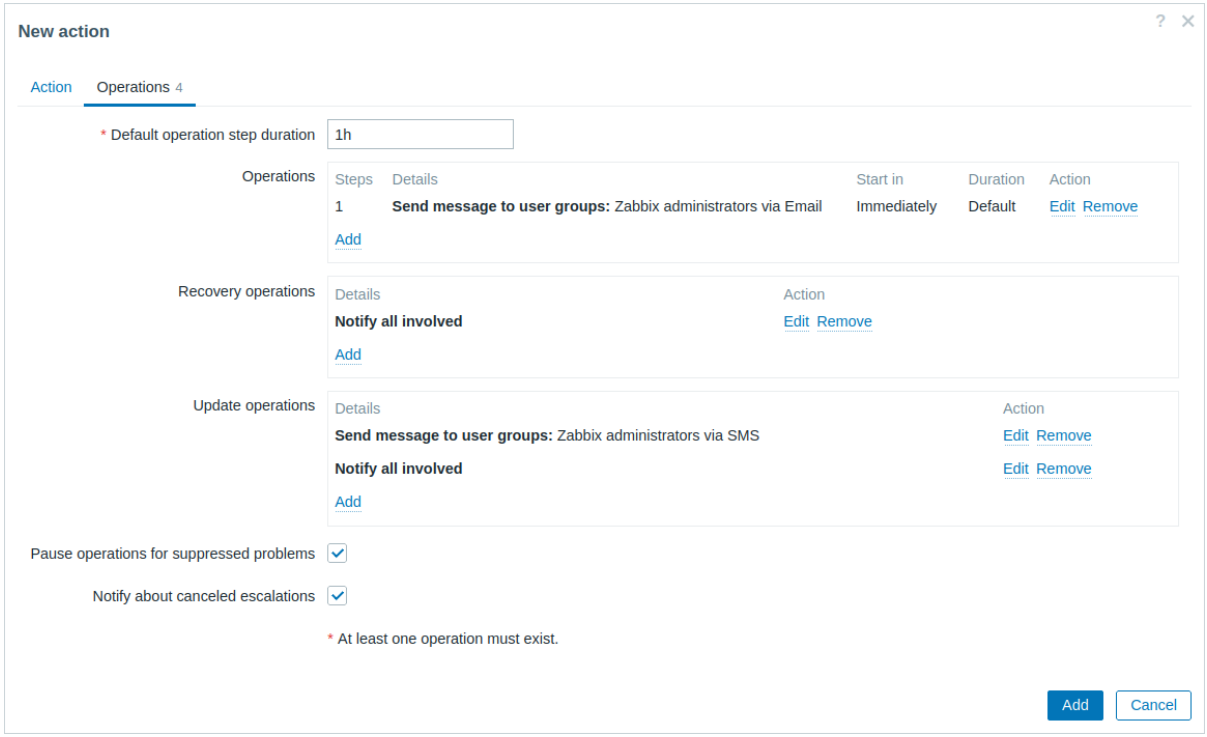
障害復旧オペレーションのユースケースの例を以下に示します。

1. 障害時に通知されたすべてのユーザーに、復旧時にも通知する:
    - オペレーションタイプとして *Notify all involved* を選択します。
2. 復旧時に複数のオペレーションを実行する: 通知を送信し、リモートコマンドを実行する:
    - メッセージ送信およびコマンド実行のオペレーションタイプを追加します。
3. 外部のヘルプデスク/チケットシステムでチケットを起票し、障害が解決されたらクローズする:
    - ヘルプデスクシステムと通信する外部スクリプトを作成します。
    - このスクリプトを実行してチケットを起票するオペレーションを持つアクションを作成します。
    - 別のパラメータでこのスクリプトを実行し、チケットをクローズする障害復旧オペレーションを設定します。
    - 元の障害を参照するために、{EVENT.ID} マクロを使用します。

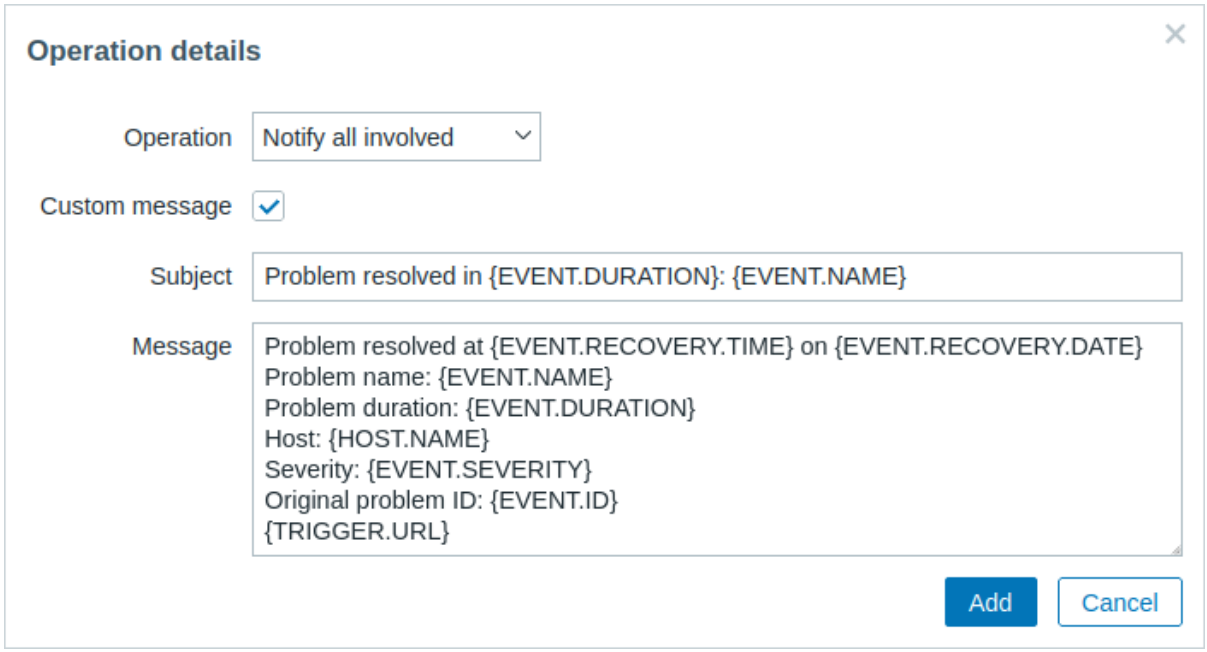
復旧操作の設定

復旧操作を設定するには、次の手順に従います。

1. アクション設定フォームの操作タブに移動します。
2. 新しい復旧操作を設定するには、復旧操作セクションで追加をクリックします。既存の操作を編集するには、その操作の横にある編集をクリックします。
3. 操作の詳細を設定します。



復旧時の実行内容の詳細



復旧イベントでは、次の 3 種類の実行内容を利用できます。

- メッセージの送信 - 指定したユーザーに復旧メッセージを送信します。
- 関係者全員に通知 - 障害イベントで通知されたすべてのユーザーに復旧メッセージを送信します。
- **<** リモートコマンド名 **>** - リモートコマンドを実行します。コマンドは、事前にグローバルスクリプトでスコープとして *Action operation* を選択して定義されている場合に実行できます。

各実行内容タイプのパラメータについては、以下で説明します。必須の入力フィールドには、すべて赤いアスタリスクが付いています。
完了したら、追加をクリックして、実行内容を 復旧時の実行内容の一覧に追加します。

Note:
同じ受信者が複数の実行内容タイプで定義されており、カスタムメッセージが指定されていない場合、重複した通知は送信されません。

実行内容のタイプ: メッセージの送信

パラメータ		説明
送信先ユーザーグループ		リカバリメッセージの送信先となるユーザーグループを選択します。 通知を受け取るには、ユーザーグループがそのホストに対して少なくとも「read」の権限を持っている必要があります。
送信先ユーザー		リカバリメッセージの送信先となるユーザーを選択します。 通知を受け取るには、ユーザーがそのホストに対して少なくとも「read」の権限を持っている必要があります。
送信先メディアタイプ		デフォルトのリカバリメッセージを、利用可能なすべての（設定済みかつ有効な）メディアタイプ、または特定の 1 つのみに送信します。
カスタムメッセージ		選択すると、カスタムメッセージを定義できます。
	件名	カスタムメッセージの件名です。件名にはマクロを含めることができます。
	メッセージ	カスタムメッセージです。メッセージにはマクロを含めることができます。

操作タイプ: リモートコマンド

パラメータ	説明
対象リスト	コマンドを実行する対象を選択します: 現在のホスト - 問題イベントの原因となったトリガーのホスト上でコマンドが実行されます。このオプションは、トリガーに複数のホストが含まれている場合は機能しません。 ホスト - コマンドを実行するホストを選択します。 ホストグループ - コマンドを実行するホストグループを選択します。親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。そのため、リモートコマンドはネストされたグループ内のホストでも実行されます。 ホスト上のコマンドは、そのホストが複数回一致した場合でも 1 回だけ実行されます（例: 複数のホストグループに属している場合、または個別指定とホストグループ指定の両方に一致する場合）。 コマンドが Zabbix サーバー上で実行される場合、対象リストには意味がありません。この場合に対象を多く選択しても、サーバー上でコマンドがその回数だけ多く実行されるだけです。 グローバルスクリプトの場合、対象の選択はグローバルスクリプトの設定にある *Host group* 設定にも依存することに注意してください。

実行タイプ: 関係者全員に通知

パラメータ		説明
カスタムメッセージ		選択すると、カスタムメッセージを定義できます。
	件名	カスタムメッセージの件名です。件名にはマクロを含めることができます。
	メッセージ	カスタムメッセージです。メッセージにはマクロを含めることができます。

4 更新時実行

概要

更新オペレーションは、以下のイベントソースを持つアクションで利用できます。

- トリガー - 問題が他のユーザーによって更新された場合、つまりコメントの追加、確認、深刻度の変更、（手動での）クローズが行われた場合。
- サービス - サービスの深刻度が変更されたが、サービスがまだ復旧していない場合。

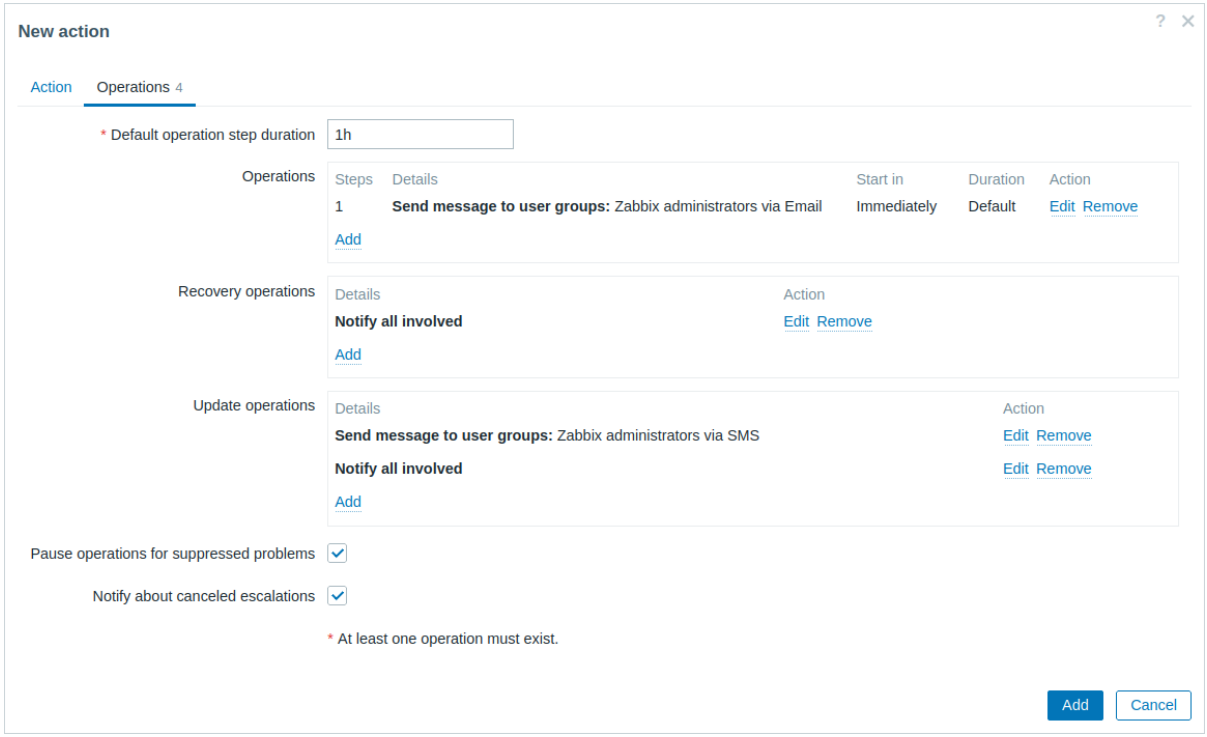
ユーザーは自分自身の更新に関する通知を受け取らないことに注意してください。

更新オペレーションでは、メッセージとリモートコマンドの両方がサポートされています。複数のオペレーションを追加できますが、エスカレーションはサポートされていません。すべてのオペレーションは単一のステップに割り当てられるため、同時に実行されます。

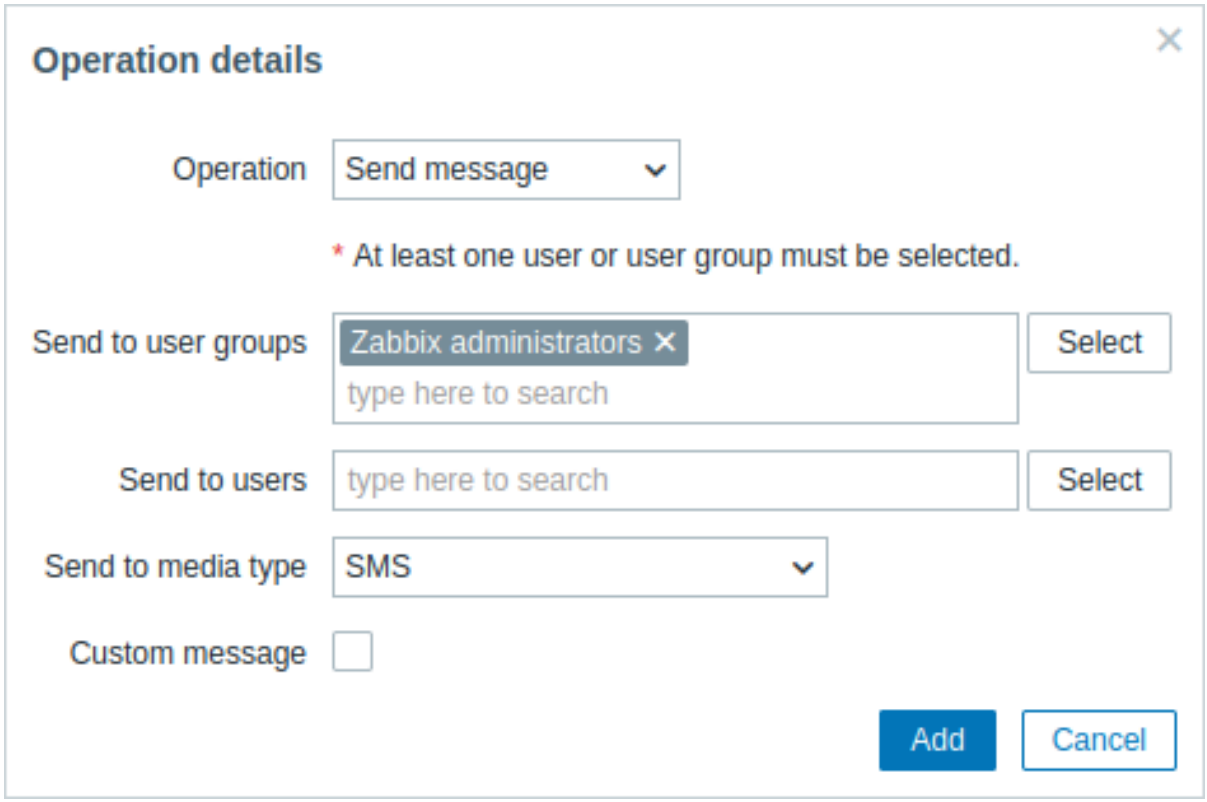
更新オペレーションの設定

更新オペレーションを設定するには、次の手順に従います。

1. アクション設定フォームのオペレーションタブを開きます。
2. 新しい更新オペレーションを設定するには、更新オペレーションセクションで追加をクリックします。既存のオペレーションを編集するには、そのオペレーションの横にある編集をクリックします。
3. オペレーションの詳細を設定します。



更新時の実行内容の詳細



更新時の実行内容では、復旧時の実行内容と同じ一連のパラメータを使用できます。

5 エスカレーション

概要

エスカレーションを使用すると、通知を送信したり、リモートコマンドを実行したりするためのカスタムシナリオを作成できます。

具体的には、次のようなことです。

- ユーザーは新しい問題をすぐに通知できます
- 問題が解決するまで通知を繰り返すことができます
- 通知の送信を遅らせることができます

- 通知は、別の” より権限が上の” ユーザーグループにエスカレートできます
- リモートコマンドは、障害発生時すぐに、または障害が長期間解決されない場合に実行できます

アクションは、エスカレーションステップに基づいてエスカレートされます。各ステップには期間があります。

個々のステップのデフォルト期間とカスタム期間の両方を定義できます。1 つのエスカレーションステップの最小所要時間は 60 秒です。

通知の送信やコマンドの実行などのアクションは、どのステップからでも開始できます。ステップ 1 は即時のアクションです。アクションを遅らせたい場合は、後のステップに割り当てることができます。ステップごとに、いくつかのアクションを定義できます。

エスカレーションのステップ数に制限はありません。

エスカレーションは、オペレーションの設定時に定義されます。エスカレーションは障害のある操作に対してのみサポートされており、復旧はサポートされていません。

エスカレーション動作その他の例

アクションに複数のエスカレーション手順が含まれている場合、さまざまな状況で何が起こるかを考えてみましょう。

状況	動作
障害が発生したホストは、最初の障害通知が送信された後にメンテナンスに入る場合	アクション設定 の メンテナンス中の場合に実行を保留の設定に応じて、残りのエスカレーション手順はすべて、メンテナンス期間中遅延するか、遅延なしで即時実行されます。メンテナンス期間によって操作がキャンセルされることはありません。
* 期間アクション条件で定義された期間は、最初の通知が送信された後、即時終了する場合 *	残りのすべてのエスカレーション手順が実行されます。期間条件では操作を停止できません。操作ではなく、アクションが開始されるかされないかに関して効果があります。
メンテナンス中に不具合が発生し、メンテナンス終了後も継続する（解消されない）場合	アクション設定 の メンテナンス中の場合に実行を保留の設定に応じて、すべてのエスカレーション手順は、メンテナンスが終了した瞬間、または即時実行されます。
データなしのメンテナンス中に障害が発生し、メンテナンス終了後も継続する (解決されない) 場合	すべてのエスカレーション手順が実行される前に、トリガーが起動するのを待つ必要があります。
さまざまなエスカレーションが連続して続き、重なる場合	新しい各エスカレーションの実行は前のエスカレーションに取って代わりますが、前のエスカレーションで常に実行されるエスカレーション手順が少なくとも 1 つ必要です。この動作は、トリガーのすべての障害評価で作成されたイベントに対するアクションに関連しています。
進行中のエスカレーション中 (メッセージの送信など)、あらゆるタイプのイベントに基づいて: - アクションは無効 トリガーイベントに基づいて: - トリガーは無効 - ホストまたはアイテムは無効 トリガーに関する内部イベントに基づいて: - トリガーは無効 アイテム/ローレベルディスカバリルールに関する内部イベントに基づいて: - アイテムは無効 - ホストは無効	進行中のメッセージが送信され、エスカレーションに関するメッセージがもう 1 つ送信されます。フォローアップメッセージには、メッセージ本文の先頭にキャンセルテキスト (注: エスカレーションがキャンセルされました) があり、理由が示されます (例: 注: エスカレーションがキャンセルされました: アクション’< アクション名 >’ が無効化されました)。このように、エスカレーションがキャンセルされ、それ以上の手順が実行されないことが受信者に通知されます。このメッセージは、以前に通知を受け取ったすべての人に送信されます。キャンセルの理由は、サーバーのログファイルにも記録されます (デバッグレベル 3=Warning 以上)。 エスカレーションがキャンセルされましたメッセージもオペレーションが終了したが、リカバリオペレーションが設定されていて、まだ実行されていない場合に送信されます。
エスカレーションの進行中 (メッセージの送信など) にアクションが削除される	これ以上メッセージは送信されません。情報はサーバーログファイルに記録されます (デバッグレベル 3= 警告以上)

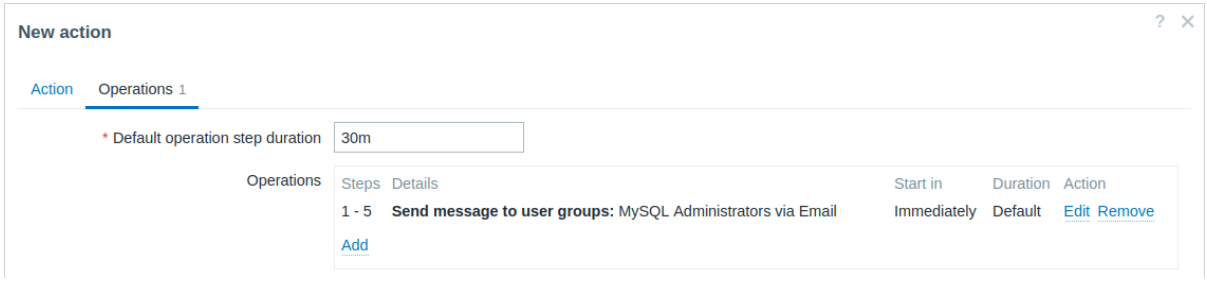
エスカレーション例

例 1

「MySQL Administrators」グループに、30 分ごとに繰り返し通知を送信する（合計 5 回）ようにするには、次のように設定します。

- *Operations* タブで、*Default operation step duration* を「30m」(30 分）に設定します。
- エスカレーションの *Steps* を「1」から「5」に設定します。

- メッセージの送信先として「MySQL Administrators」グループを選択します。



通知は、障害の発生後 0:00、0:30、1:00、1:30、2:00 に送信されます（もちろん、障害がそれより前に復旧した場合を除きます）。

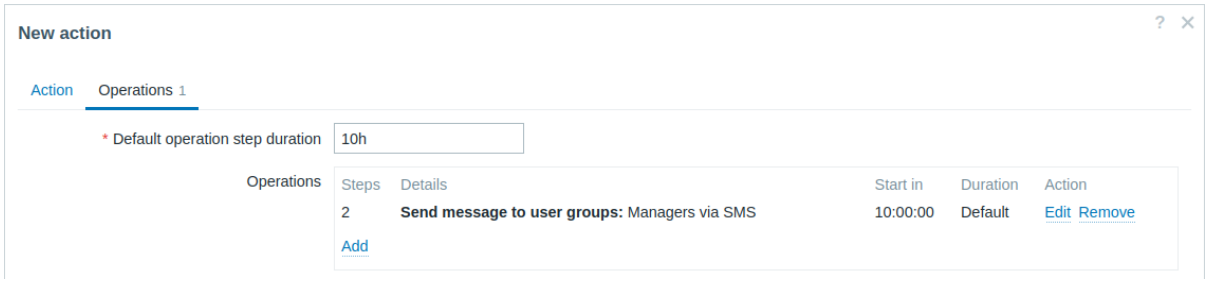
障害が復旧し、復旧メッセージが設定されている場合は、このエスカレーションシナリオ内で少なくとも 1 回障害メッセージを受信したユーザーに送信されます。

Note:
アクティブなエスカレーションを生成したトリガーが無効化された場合、Zabbix はすでに通知を受信しているすべてのユーザーにその旨を知らせる情報メッセージを送信します。

例 2

長時間継続している障害について、遅延通知を送信します。設定するには、次のようにします。

- 実行内容タブで、デフォルトの実行ステップの間隔を「10h」（10 時間）に設定します。
- エスカレーションの ステップを「2」から「2」に設定します。



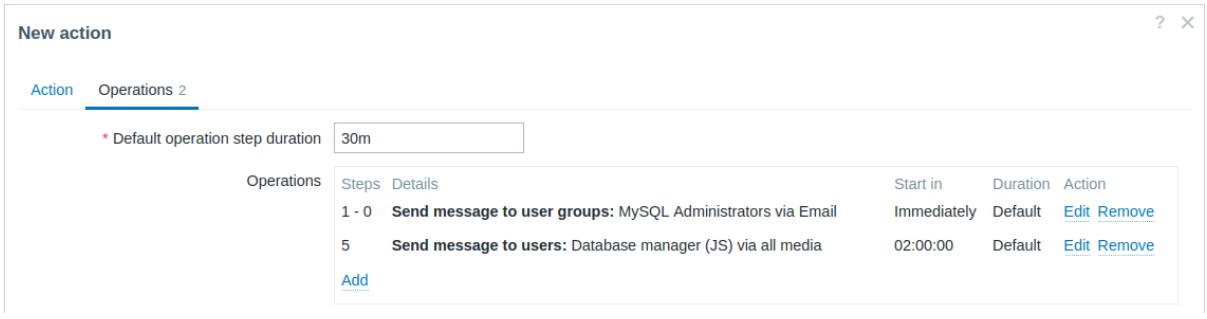
通知は、エスカレーションシナリオのステップ 2 でのみ、つまり障害の発生から 10 時間後に送信されます。

メッセージのテキストは、「この障害は 10 時間以上継続しています」のようにカスタマイズできます。

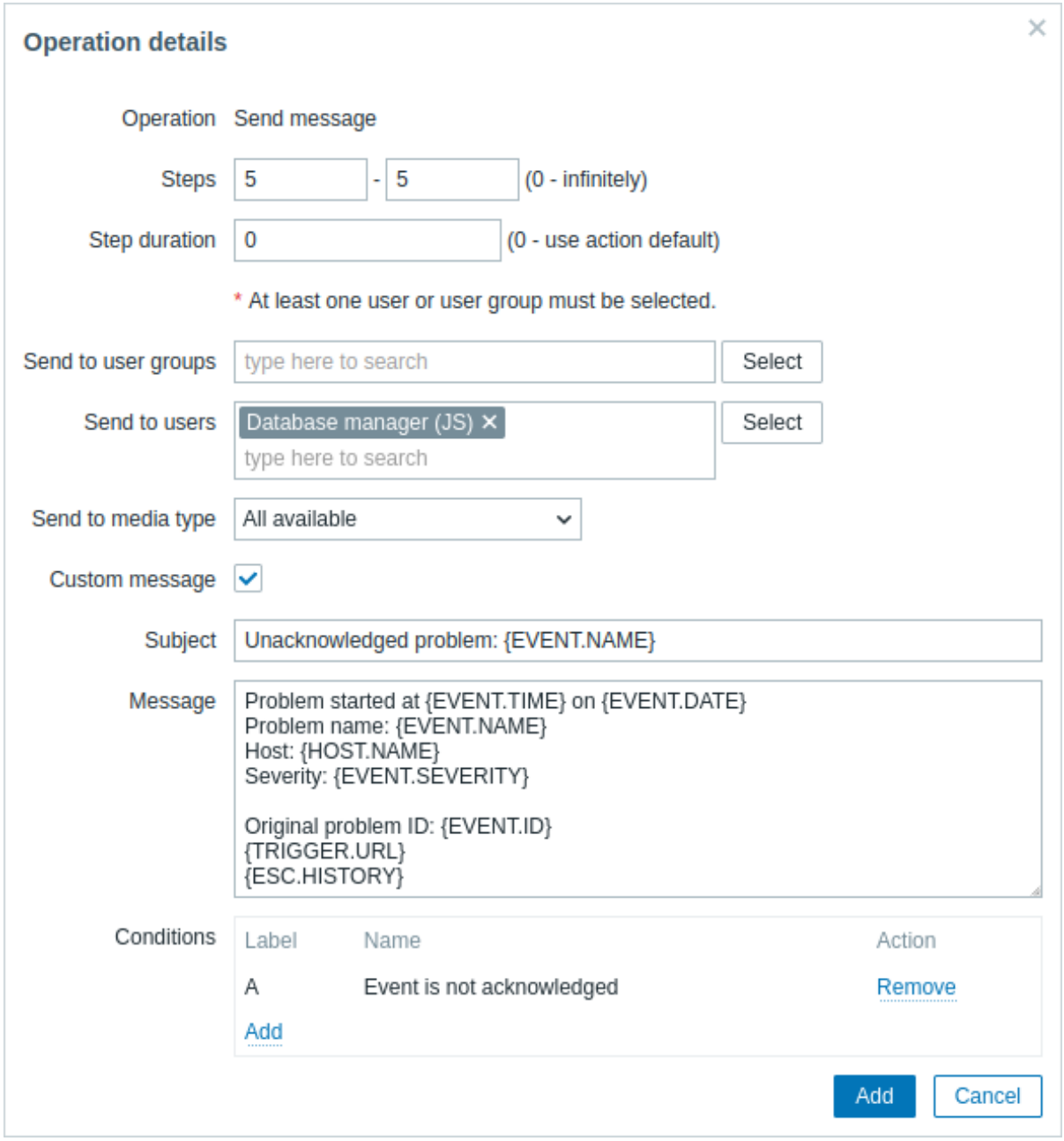
例 3

問題を上司にエスカレーションします。

上記の最初の例では、MySQL 管理者に対してメッセージを定期的に送信するよう設定しました。この場合、問題がデータベースマネージャーにエスカレーションされるまでに、管理者は 4 通のメッセージを受け取ります。マネージャーがメッセージを受け取るのは、問題がまだ確認応答されておらず、つまり誰も対応していないと考えられる場合のみであることに注意してください。



操作 2 の詳細:



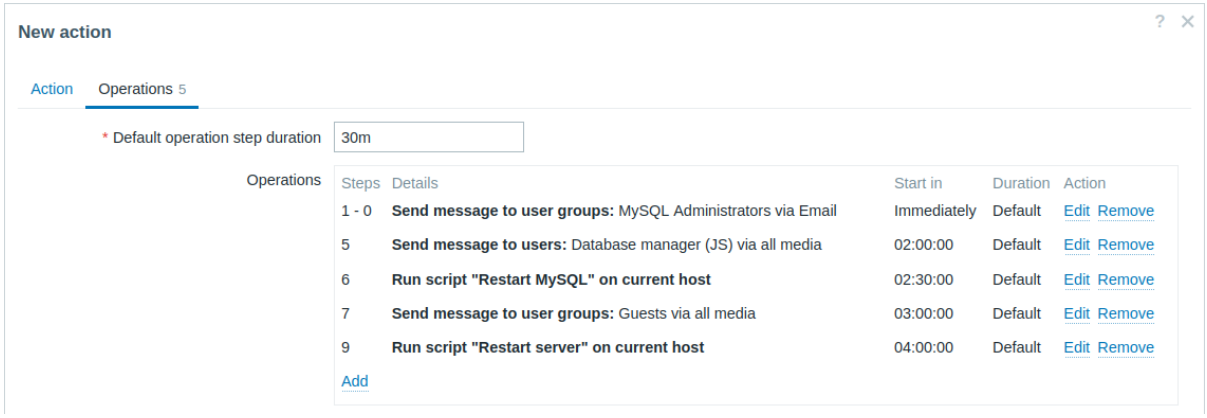
カスタマイズされたメッセージで {ESC.HISTORY} マクロが使用されている点に注意してください。このマクロには、このエスカレーションで以前に実行されたすべてのステップに関する情報、たとえば送信された通知や実行されたコマンドが含まれます。

例 4

より複雑なシナリオ。MySQL 管理者への複数のメッセージとマネージャーへのエスカレーションの後、Zabbix は MySQL データベースの再起動を試みます。障害が 2:30 時間存在し、確認されていない場合に発生します。

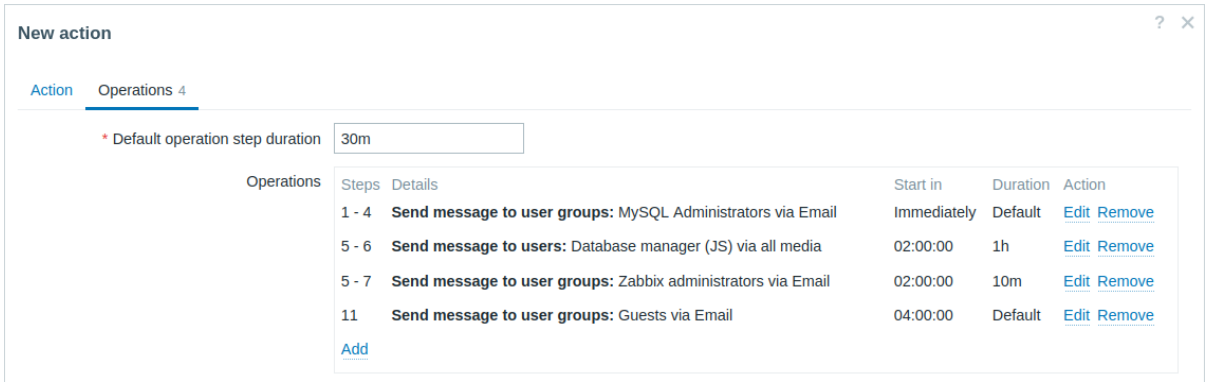
それでも障害が解決しない場合は、さらに 30 分後に Zabbix がすべてのゲストユーザーにメッセージを送信します。

これで障害が解決しない場合は、さらに 1 時間後に、Zabbix は IPMI コマンドを使用して MySQL データベース (2 番目のリモートコマンド) でサーバーを再起動します。



例 5

複数のオペレーションが重複するステップ範囲とカスタム間隔を持つエスカレーションの例です。デフォルトのオペレーションステップの期間は 30 分です。



通知は次のように送信されます。

- 問題の発生開始から 0:00、0:30、1:00、1:30 後に、MySQL 管理者へ送信されます。
- 問題の発生開始から 2:00 と 2:10 に、Database manager へ送信されます（後続のオペレーションで定義された、より短い 10 分のカスタムステップ期間が、ここで設定された 1 時間のより長いステップ期間を上書きします。これは、ステップが重複する場合の *Step duration* についてオペレーションの詳細 で説明されているとおりです）。
- 問題の発生開始から 2:00、2:10、2:20 後に、Zabbix 管理者へ送信されます（10 分のカスタムステップ期間が適用されます）。
- 問題の発生開始から 4:00 後に、guest users へ送信されます（ステップ 8 から 11 の間では、デフォルトの 30 分のステップ期間が有効になります）。

####3 サポート対象外通知の受信 {#manual-config-notifications-unsupported_item}

概要

Zabbix では、未サポートのアイテムに関する通知を受け取ることができます。

これは Zabbix の内部イベントの概念の一部であり、このような場合にユーザーへ通知できるようにするものです。内部イベント は、状態の変化を反映します。

- アイテムが「正常」から「未サポート」に変わったとき（およびその逆）；
- トリガーが「正常」から「不明」に変わったとき（およびその逆）；
- ローレベルディスカバリルールが「正常」から「未サポート」に変わったとき（およびその逆）。

このセクションでは、アイテムが未サポートになったときに通知を受信する方法を説明します。

設定

おおまかには通知を設定するプロセスは以前に Zabbix でアラートを設定したことがある人には馴染みのあるものです。

ステップ 1

通知に使用するメディア（メール、SMS、スクリプトなど）を設定します。この作業を行うには、マニュアルの対応するセクションを参照してください。

Attention:
内部イベントの通知にはデフォルトの深刻度（「未分類」）が使用されるため、内部イベントの通知を受け取りたい場合は、ユーザーメディアの設定時にこのチェックを外さないでください。

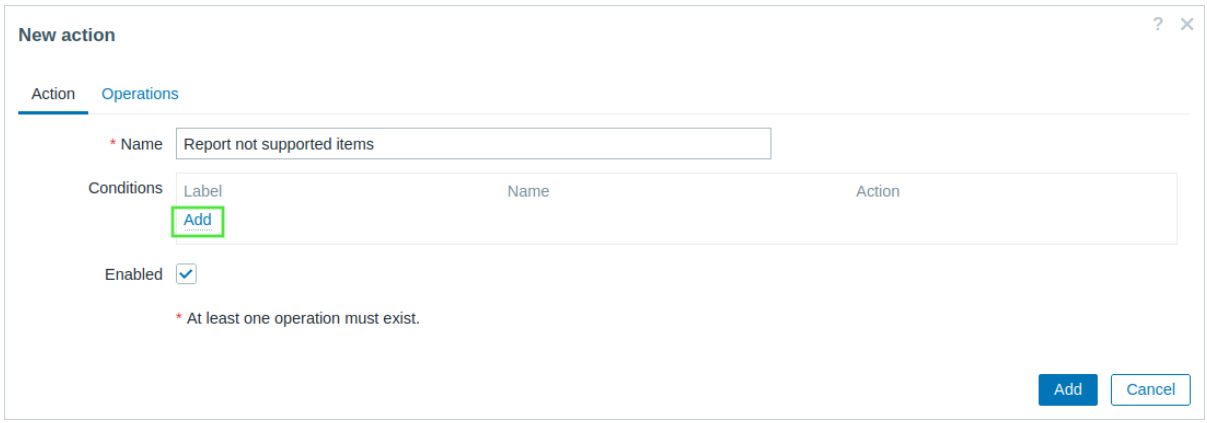
手順 2

アラート → アクション → 内部アクションに移動します。

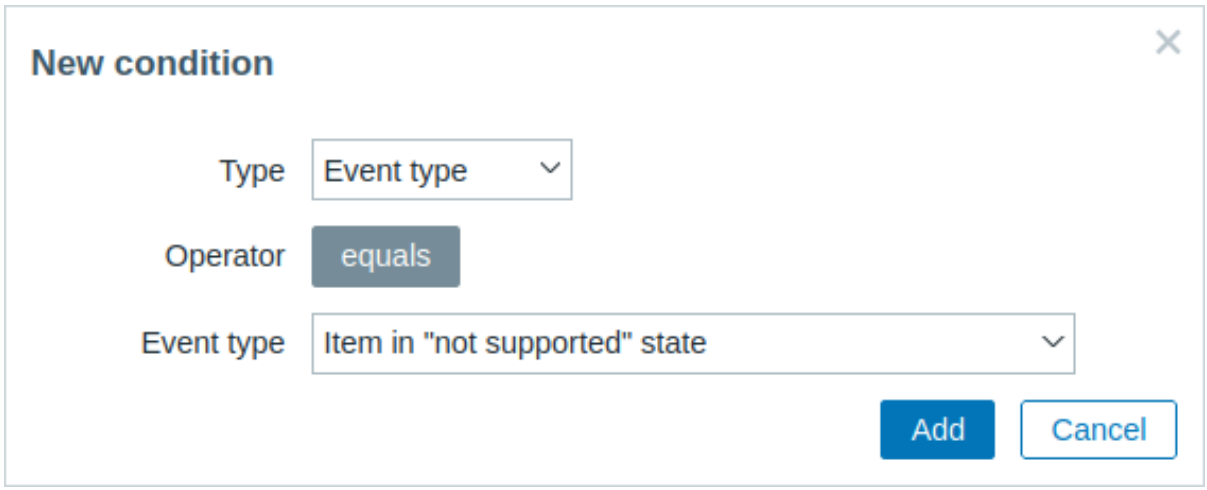
ページ右上の アクションの作成をクリックして、アクション設定フォームを開きます。

ステップ 3

アクションタブで、アクションの名前を入力します。
次に、条件ブロックで追加をクリックして、新しい条件を追加します。



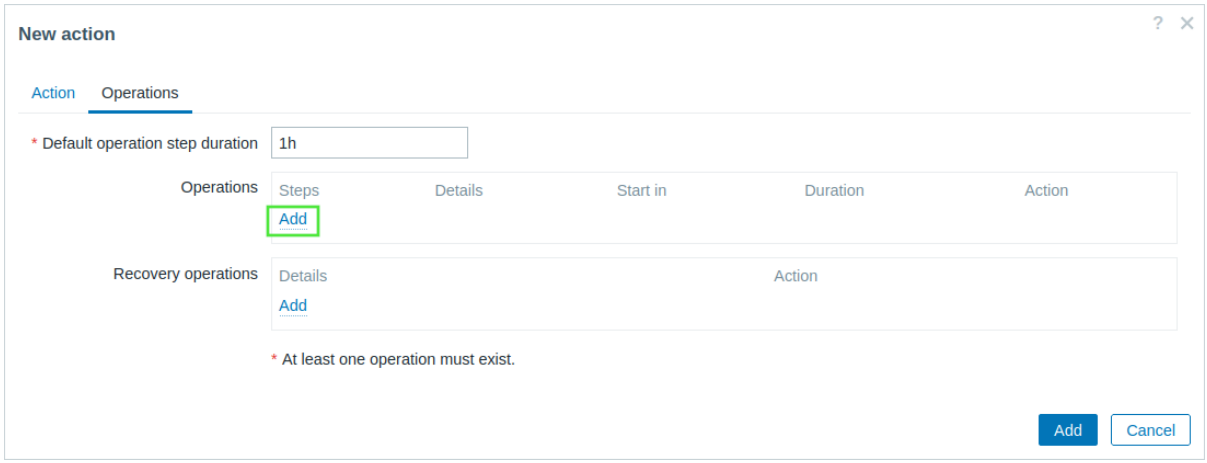
新しい条件ポップアップウィンドウで、条件タイプとして「イベントタイプ」を選択し、次にイベントタイプとして「' サポートされていない' 状態のアイテム」を選択します。



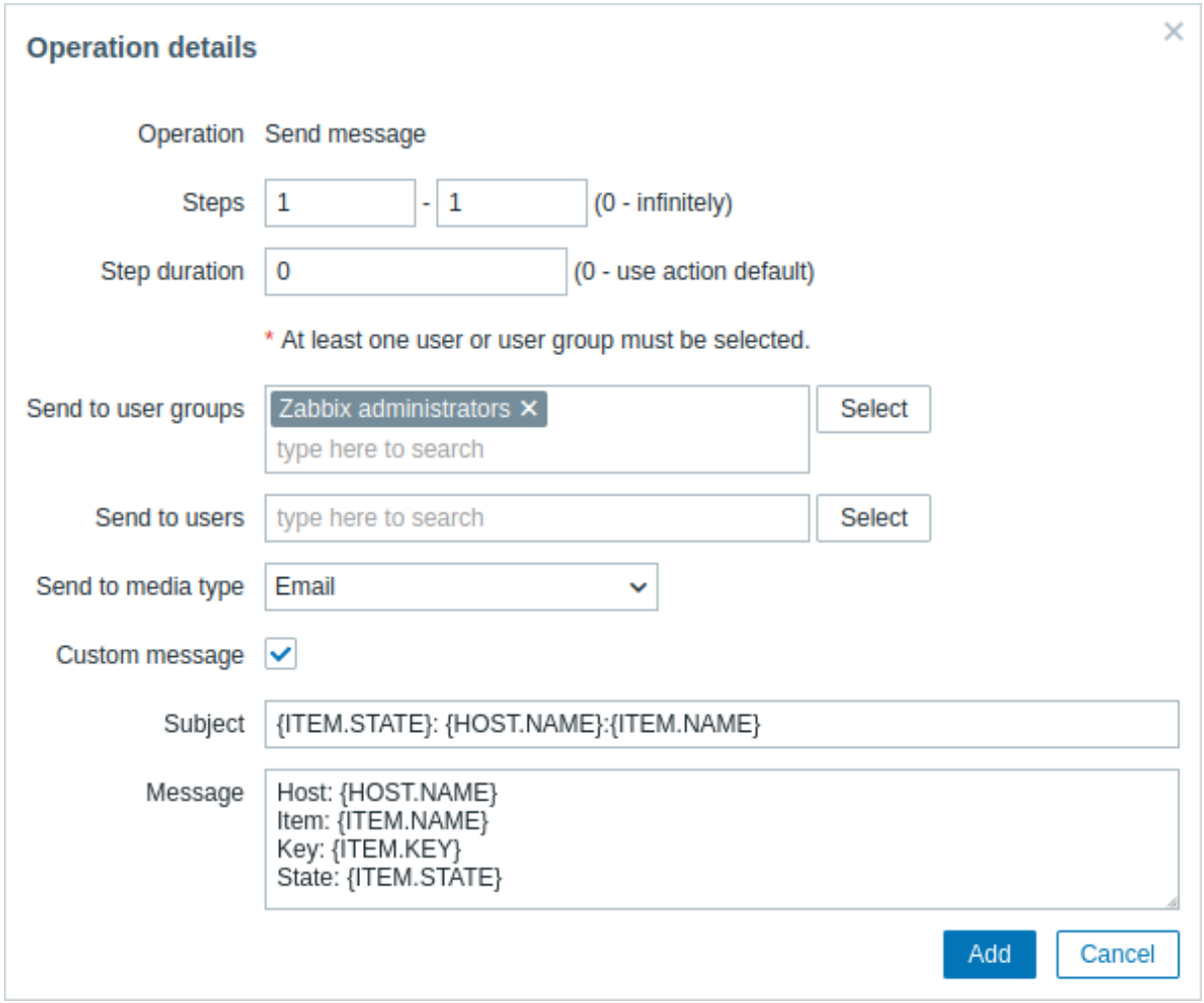
条件を実際に条件ブロックに一覧表示するには、追加をクリックすることを忘れないでください。

ステップ 4

Operations タブで、*Operations* ブロック内の *Add* をクリックして、新しいオペレーションを追加します。



メッセージの受信者（ユーザーグループ/ユーザー）と、配信に使用するメディアタイプ（または「All」）を選択します。障害メッセージの件名/内容をカスタマイズして入力する場合は、*Custom message* チェックボックスをオンにします。

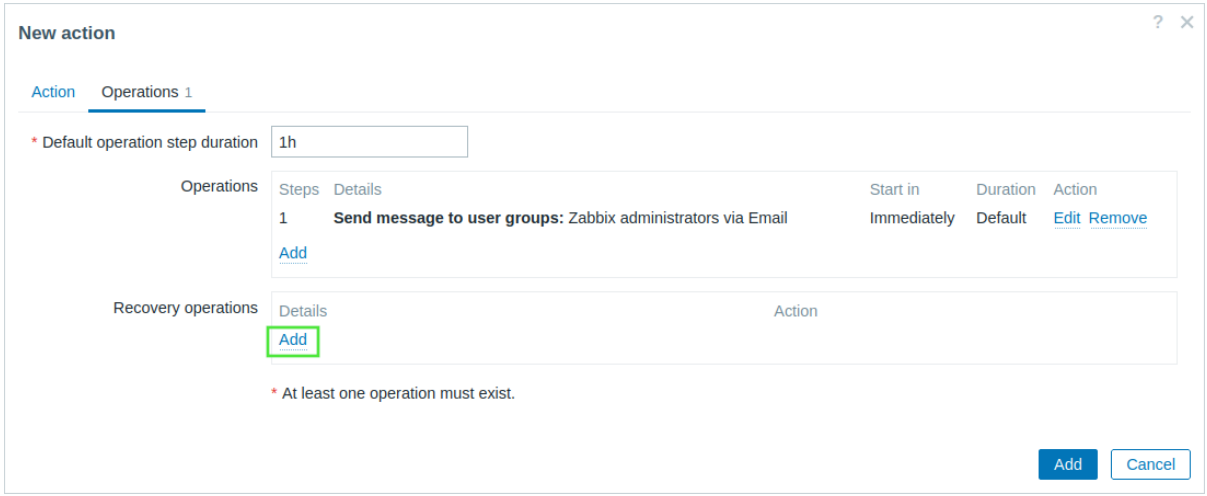


Add をクリックして、オペレーションを実際に *Operations* ブロック内に一覧表示します。

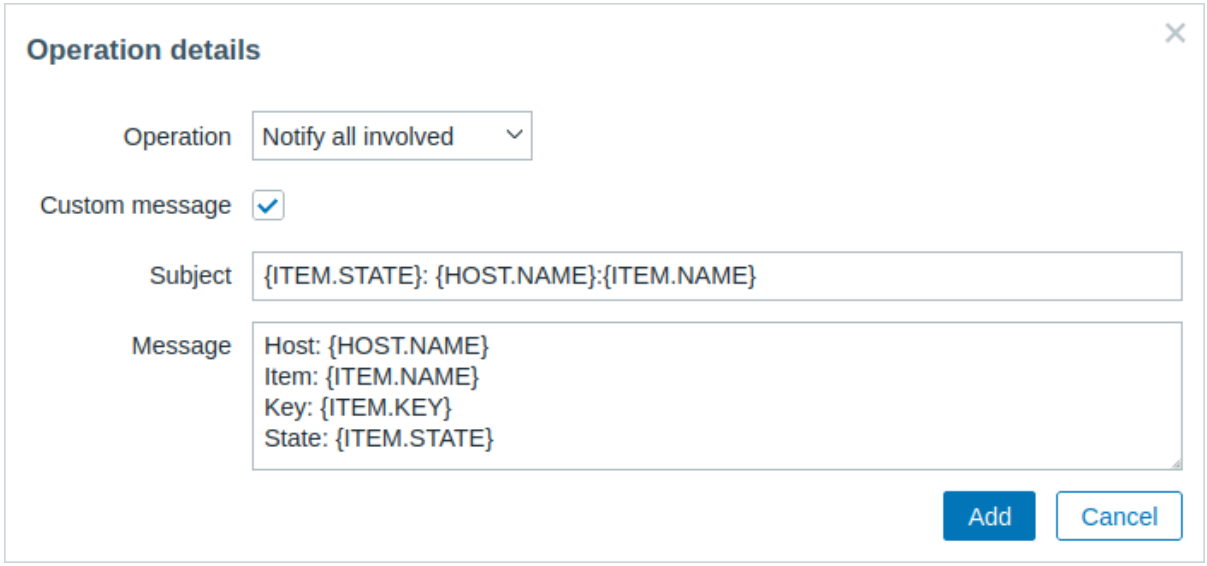
複数の通知を受け取りたい場合は、オペレーションのステップ期間（送信されるメッセージ間の間隔）を設定し、別のステップを追加します。

ステップ 5

Recovery operations ブロックでは、アイテムが正常な状態に戻ったときの復旧通知を設定できます。
新しい復旧オペレーションを追加するには、*Recovery operations* ブロックの *Add* をクリックします。



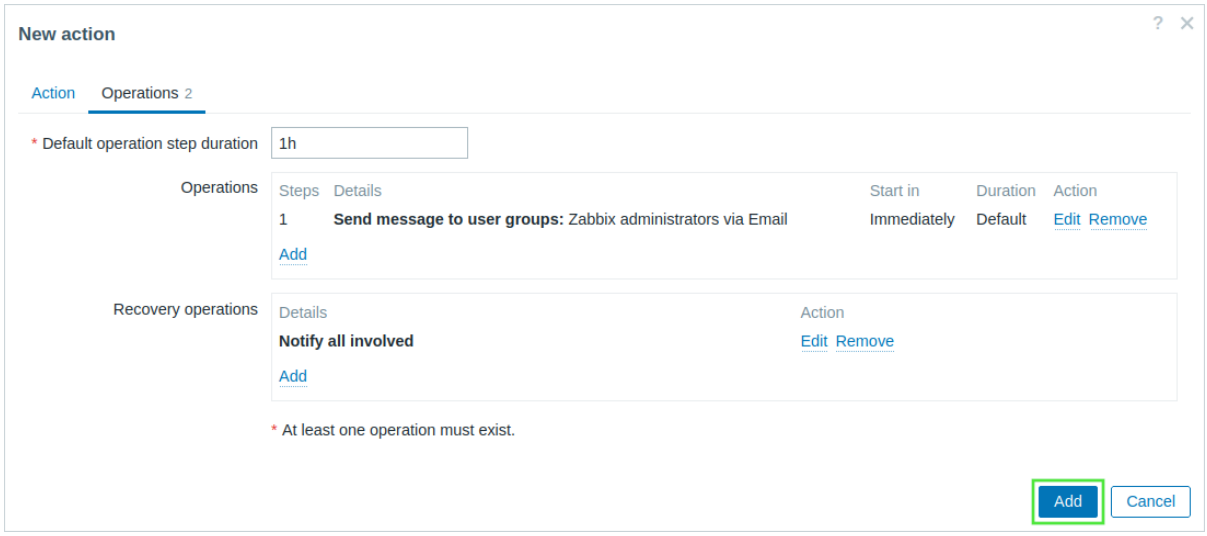
オペレーションタイプ”Notify all involved” を選択します。
復旧メッセージの件名/内容をカスタマイズして入力する場合は、*Custom message* チェックボックスを選択します。



Operation details ポップアップウィンドウの *Add* をクリックすると、*Recovery operations* ブロックにオペレーションが実際に一覧表示されます。

ステップ 6

完了したら、フォームの下部にある 追加ボタンをクリックします。



以上で完了です。これで、いずれかのアイテムが未サポート状態になった場合に、Zabbix から最初の通知を受け取れるようになります。

11 マクロ

概要

Zabbix は、さまざまな状況で使用できる多数の組み込みマクロをサポートしています。これらのマクロは変数であり、特定の構文で識別されます:

`{MACRO}`

マクロは、コンテキストに応じて特定の値に解決されます。

マクロを効果的に使用することで、時間を節約し、Zabbix の設定をより透過的にすることができます。

典型的な用途の 1 つとして、テンプレートでマクロを使用することがあげられます。例えばテンプレートのトリガーの名前を”{HOST.NAME} のプロセッサ負荷が高すぎます” とします。テンプレートが Zabbix サーバーなどのホストに適用され、トリガーが監視セクションに表示されると、この名前は”Zabbix サーバーのプロセッサ負荷が高すぎます” に解決されます。

マクロは、アイテムキーパラメーターで使用できます。`item.key[server_{HOST.HOST}_local]` のようにパラメーターの一部としても可能です。解決されたマクロにあいまいな特殊記号存在する場合、Zabbix が処理するのでパラメーターをダブルクオートで囲む必要はありません。

Zabbix には他の種類のマクロもあります。

Zabbix は次のマクロをサポートしています。

- {MACRO} - 組み込みマクロ (全リストを参照)
- {<macro>.<func>(<params>)} - マクロ関数
- {$MACRO} - ユーザー定義のマクロ、オプションでコンテキスト付き
- {#MACRO} - ローレベルディスカバリのマクロ
- {?EXPRESSION} - 式マクロ

1 マクロ関数

概要

マクロ関数を使用すると、マクロの値をカスタマイズし（例えば、特定の部分文字列を短縮または抽出するなど）、より扱いやすくできます。

マクロ関数の構文は次のとおりです。

```
{macro.func(params)}
```

ここで

- **macro** - カスタマイズするマクロ。
- **func** - 適用する関数（サポートされている関数を参照）。
- **params** - 関数パラメータのカンマ区切りリスト。次の場合は 二重引用符で囲む必要があります。
  - スペースまたは二重引用符で始まる場合。
  - 閉じ括弧またはカンマを含む場合。

例えば:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

マクロ関数は以下でサポートされています。

- 組み込みマクロ
- ユーザーマクロ
- ローレベルディスカバリマクロ
- 式マクロ

マクロ関数は、上記のマクロをサポートするすべての場所で使用できます。
ただし、マクロのみが想定されていることが明示されている場合は除きます（例えば、ホストマクロやローレベルディスカバリルールのフィルターを設定する場合）。

1 つのマクロにつきサポートされる関数は 1 つのみであり、複数のマクロ関数を連結して使用することはできません。

> **Note:**
> マクロ関数が他のコンテキスト（関数、アイテムキー、別のマクロなど）の中で使用される場合については、エスケープの例を参照してください。

サポートされている関数

関数は追加情報なしで一覧表示されています。
関数をクリックすると、詳細を確認できます。

Function	Description
btoa	マクロ値を Base64 エンコーディングにエンコードします。
fmtnum	小数点以下に表示する桁数を制御するための数値書式設定。
fmttime	時刻の書式設定。
htmldecode	HTML エンコーディングされたマクロ値をデコードします。
htmlencode	マクロ値を HTML エンコーディングにエンコードします。
iregsub	正規表現の一致による部分文字列の抽出（大文字・小文字を区別しない）。
lowercase	マクロ値の文字を小文字に変換します。
regrepl	マクロ値内の文字/部分文字列を置換します。
regsub	正規表現の一致による部分文字列の抽出（大文字・小文字を区別する）。

Function	Description
tr	マクロ値の文字を変換します。
uppercase	マクロ値の文字を大文字に変換します。
urldecode	URL エンコーディングされたマクロ値をデコードします。
urlencode	マクロ値を URL エンコーディングにエンコードします。

関数の詳細

オプションの関数パラメータは < > で示されます。

btoa

マクロ値を Base64 エンコードします。
Base64 エンコードは、バイナリデータをテキストとして表現する方法であり、テキストベースのプロトコル上でバイナリコンテンツを保存したり、安全に送信したりする際に役立ちます。

例:

```
{{ITEM.VALUE}.btoa()} - "zabbix" のような値を "emFiYml4" に Base64 エンコードします
```

fmtnum(digits)

小数点以下に表示する桁数を制御するための数値書式設定です。

パラメータ:

- **digits** - 小数点以下の桁数。有効範囲: 0～20。末尾のゼロは出力されません。

例:

```
{{ITEM.VALUE}.fmtnum(2)} - 受信した値 "24.3483523" から "24.35" を返します
{{ITEM.VALUE}.fmtnum(0)} - 受信した値 "24.3483523" から "24" を返します
```

fmttime(format,<time_shift>)

時刻の書式設定。
この関数は、以下のいずれかの時刻形式の値に展開されるマクロで使用できることに注意してください。

- hh:mm:ss
- yyyy-mm-ddThh:mm:ss[tz]（ISO8601 標準）
- UNIX タイムスタンプ

パラメータ:

- **format** - 必須の書式文字列。strftime 関数の書式設定と互換性があります。
- **time_shift**（任意）- 書式設定の前に時刻へ適用される時間シフト。-<N><time_unit> または +<N><time_unit> で始まる必要があります。ここで:
  - N - 加算または減算する時間単位の数。
  - time_unit - h（時間）、d（日）、w（週）、M（月）、または y（年）。

コメント:

- time_shift パラメータは複数段階の時間操作をサポートしており、時間単位の先頭へ移動するための /<time_unit> を含めることができます（/d - 深夜 0 時、/w - 週の 1 日目（月曜日）、/M - 月の 1 日目、など）。例: -1w - ちょうど 7 日前、-1w/w - 前週の月曜日、-1w/w+1d - 前週の火曜日。
- 時間操作は優先順位なしで左から右に計算されます。例えば、-1M/d+1h/w は ((-1M/d)+1h)/w として解析されます。

例:

```
{{TIME}.fmttime(%B)} - 受信した値 "1633098961" から "October" を返します
{{TIME}.fmttime(%d %B,-1M/M)} - 受信した値 "1633098961" から "1 September" を返します
```

htmldecode

マクロ値を HTML エンコードからデコードします。

以下の文字がサポートされています。

Value	デコード後の値
&	&
<	<
>	>

Value	デコード後の値
"	"
'	'
'	'

例:

```
{{ITEM.VALUE}.htmldecode()} - "&lt;" のような値は "<" に HTML デコードされます
```

htmlencode

マクロ値を HTML エンコードに変換します。

以下の文字がサポートされています。

値	エンコード後の値
&	&
<	<
>	>
"	"
'	'

例:

```
{{ITEM.VALUE}.htmlencode()} - "<" のような文字を "&lt;" に HTML エンコードします
```

iregsub(pattern,output)

正規表現の一致による部分文字列の抽出（大文字・小文字を区別しない）。

パラメータ:

- **pattern** - 一致させる正規表現;
- **output** - 出力オプション。キャプチャグループには **\1 - \9** プレースホルダーを使用できます。**\0** は一致したテキストを返します。

コメント:

- 正規表現に一致しない場合、この関数は空文字列を返します。
- 関数の pattern が不正な正規表現である場合、マクロは'UNKNOWN' と評価されます（ただし、ローレベルディスカバリマクロは例外で、この場合は関数は無視され、マクロは未解決のままになります）。
- 置換文字列内で存在しないキャプチャグループを参照した場合は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - "fail"、"error"、"fault"、または "problem" の部分
```

小文字

マクロ値のすべての文字を小文字に変換します。
シングルバイト文字セット（ASCII など）で動作し、UTF-8 はサポートしていません。

例:

```
{{ITEM.VALUE}.lowercase()} - "Zabbix SERVER" のような値を "zabbix server"（小文字）に変換します
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

マクロ値内の文字/部分文字列を置換します。

パラメータ:

- **pattern** - 一致させる正規表現。
- **replacement** - 置換文字列。置換文字列では、キャプチャグループを参照するためのプレースホルダー **\1 - \9** を使用できます。

コメント:

- パターンと置換は順番に処理され、後続の各ペアは前の置換結果に基づいて適用されます。
- 置換文字列内で存在しないキャプチャグループへの参照は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - "oldParam" を "newParam" に置き換えます
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - 英字以外のすべての文字はバックスラッシュでエスケープされます
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - 末尾のバックスラッシュを削除します（たとえば "C:\" を "C:" に置
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - アイテム値内の複数の部分を置き換えます
```

regsub(pattern,output)

正規表現の一致による部分文字列の抽出（大文字・小文字を区別）。

パラメータ:

- **pattern** - 一致させる正規表現。
- **output** - 出力オプション。**\1 - \9** プレースホルダーはキャプチャグループに対応しています。**\0** は一致したテキストを返します。

コメント:

- 正規表現に一致しない場合、この関数は空文字列を返します。
- 関数の pattern が不正な正規表現である場合、マクロは'UNKNOWN' と評価されます（ただし、ローレベルディスカバリマクロの場合はこの関数は無視され、マクロは未解決のままとなります）。
- 置換文字列内で存在しないキャプチャグループを参照した場合は、空文字列に置き換えられます。

例:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - "123 Log line" のような値を受信した場合は "Problem ID: 123
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - "fail"、"error"、"fault"、または "problem" の部分
```

さらに例を見る。

tr(characters,replacement)

マクロ値の文字の変換を行います。

- **characters** - 置換する文字のセット。
- **replacement** - 位置に対応する置換文字のセット。

例:

```
{{ITEM.VALUE}.tr(abc, xyz)} - "a" のすべての出現を "x" に、"b" を "y" に、"c" を "z" に置換します
{{ITEM.VALUE}.tr(abc, xyzq)} - "a" のすべての出現を "x" に、"b" を "y" に、"c" を "z" に置換します（"q" は無視されま
{{ITEM.VALUE}.tr(abcde, xyz)} - "a" のすべての出現を "x" に、"b" を "y" に、"c" を "z" に、"d" を "z" に、"e" を
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - バックスラッシュのすべての出現をスラッシュに、シングルクォートをダブルクォートに置換
{{ITEM.VALUE}.tr(A-Z,a-z)} - すべての文字を小文字に変換します
{{ITEM.VALUE}.tr(0-9a-z,*)} - すべての数字と小文字を "*" に置換します
{{ITEM.VALUE}.tr(0-9,ab)} - 0 のすべての出現を "a" に置換し、1、2、3、4、5、6、7、8、9 のすべての出現を "b" に置換しま
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - すべての数字、"abc" の文字、および A-L の範囲を "*" に置換します
{{ITEM.VALUE}.tr("\n","*")} - 行末の出現を * に置換します
{{ITEM.VALUE}.tr("e", "\n")} - すべての "e" を改行に置換します
```

リテラル文字を含めるには:

```
backslash - \\ としてエスケープする必要があります
single quote - \' としてエスケープする必要があります
double quote - \" としてエスケープする必要があります
```

バックスラッシュで使用できるエスケープシーケンス:

```
\\\\ => \\ - 二重バックスラッシュを単一のバックスラッシュに
\\a  => \a - アラート
\\b  => \b - バックスペース
\\f  => \f - 改ページ
\\n  => \n - 改行
\\r  => \r - 復帰
\\t  => \t - 水平タブ
\\v  => \v - 垂直タブ
```

大文字

マクロ値のすべての文字を大文字に変換します。
シングルバイト文字セット（ASCII など）で動作し、UTF-8 はサポートしていません。

例:

```
{{ITEM.VALUE}.uppercase()} - "Zabbix Server" のような値は "ZABBIX SERVER"（大文字）に変換されます
```

urldecode

マクロ値を URL エンコードからデコードします。

例:

```
{{ITEM.VALUE}.urldecode()} - "%2F" のような値を URL デコードして "/" にします
```

urlencode

マクロ値を URL エンコード形式にエンコードします。

例:

```
{{ITEM.VALUE}.urlencode()} - "/" のような文字を "%2F" に URL エンコードします
```

追加の例

以下の表は、マクロ関数の使用例をさらに示しています。

{#IFALIAS} はLLD macros であり、低レベルディスカバリのコンテキスト（ディスカバリルール、プロトタイプ、およびそれらから作成されたアイテム/トリガー）でのみ定義されます。
LLD の外部でこれを使用すると、トークンは展開されません。

マクロ関数	受信した値	出力
{{ITEM.VALUE}.regsub(^[0-9]+, Problem)}	123Log line	Problem
{{ITEM.VALUE}.regsub("^([0-9]+)", "Problem")}	123 Log line	Problem
{{ITEM.VALUE}.regsub(".*", "Problem ID: \1")}	Log line	Problem ID:
{{ITEM.VALUE}.regsub("^(\w+).*?([0-9]+)", " Problem ID: \1_\2 ")}	MySQL crashed errno 123	Problem ID: MySQL_123
{{ITEM.VALUE}.regsub("([1-9]+", "Problem ID: \1")}	123 Log line	*UNKNOWN*（無効な正規表現）
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}	customername_1	customername
{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}	customername_11	
{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}	customername_1	{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}（無効な正規表現）
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"customername"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \2)}"}	customername_1	{$MACRO:"1"}
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}（無効な正規表現）
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"	customername_1	"{$MACRO:"\customername\"}"
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \2)}\"}"	customername_1	"{$MACRO:"\1\"}"
"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+", \1)}\"}"	customername_1	"{$MACRO:\"{{#IFALIAS}.regsub(\\"(.*)_([0-9]+\\", \1)}\"}"（無効な正規表現）

完全なアイテム値を表示する

テキスト/ログアイテムに対して解決された {ITEM.VALUE} および {ITEM.LASTVALUE} マクロの長い値は、一部の Web インターフェース上の場所では 20 文字に切り詰められます。
これらのマクロの完全な値を表示するには、たとえば次のようにマクロ関数を使用できます。

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

参照: {ITEM.VALUE} および {ITEM.LASTVALUE} のmacro details。

2 ユーザーマクロ

概要

ユーザーマクロは、標準でサポートされているマクロに加えて、より高い柔軟性を実現するために Zabbix でサポートされています。

ユーザーマクロは、グローバル、テンプレート、ホストの各レベルで定義できます。これらのマクロは、次の特別な構文を持ちます。

`{$MACRO}`

Zabbix は、次の優先順位に従ってマクロを解決します。

1. ホストレベルのマクロ（最初に確認）
2. ホストの第 1 レベルのテンプレート（つまり、ホストに直接リンクされたテンプレート）で定義されたマクロ。テンプレート ID 順にソート
3. ホストの第 2 レベルのテンプレートで定義されたマクロ。テンプレート ID 順にソート
4. ホストの第 3 レベルのテンプレートで定義されたマクロ。テンプレート ID 順にソート、以下同様
5. グローバルマクロ（最後に確認）

言い換えると、あるマクロがホストに存在しない場合、Zabbix は深さが浅い順にホストのテンプレート内でそのマクロを探します。それでも見つからない場合は、存在すればグローバルマクロが使用されます。

Warning:
同じ名前のマクロが同じレベルの複数のリンク済みテンプレートに存在する場合、最も ID の小さいテンプレートのマクロが使用されます。したがって、複数のテンプレートに同じ名前のマクロを持たせることは、設定上のリスクとなります。

Zabbix がマクロを見つけられない場合、そのマクロは解決されません。

Attention:
マクロ（ユーザーマクロを含む）は、複雑な設定をより分かりやすくするため、仕様により設定セクション（たとえばトリガー一覧）では未解決のまま表示されます。

ユーザーマクロは次の場所で使用できます。

- アイテム名
- アイテムキーのパラメータ
- アイテムの更新間隔および柔軟な間隔
- トリガー名および説明
- トリガー条件式のパラメータおよび定数（例を参照）
- その他多数の場所 - 完全な一覧を参照

グローバルマクロとホストマクロの一般的な使用例

- グローバルマクロを複数の場所で使用し、マクロの値を変更すると、ワンクリックですべての場所に設定変更が適用されます。
- ホスト固有の属性（パスワード、ポート番号、ファイル名、正規表現など）を持つテンプレートを活用します。

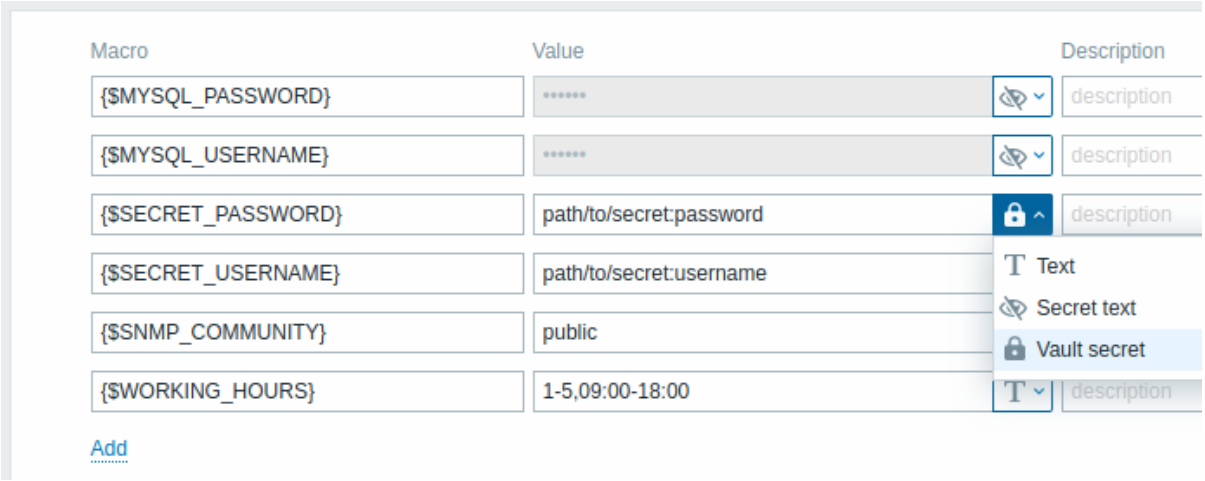
Note:
グローバルマクロの追加、更新、削除を行うと、すべてのホストに対して増分設定更新が強制されるため、グローバルマクロではなくホストマクロを使用することを推奨します。詳細については、パッシブおよびアクティブなエージェントチェックを参照してください。

設定

ユーザーマクロを定義するには、Web インターフェース内の該当する場所に移動します。

- グローバルマクロの場合は、*Administration* → *Macros* を開きます
- ホストおよびテンプレートレベルのマクロの場合は、ホストまたはテンプレートのプロパティを開き、*Macros* タブを探します

ユーザーマクロには、次の属性があります。



Parameter	Description
Macro	マクロ名。名前は中括弧で囲み、ドル記号で始める必要があります。 例: {$FRONTEND_URL}。マクロ名で使用できる文字は次のとおりです: **A-Z**（大文字のみ）、**0-9**、_、.
Value	マクロ値。3 種類の値タイプがサポートされています。 **Text**（デフォルト）- プレーンテキスト値 **Secret text** - 値はアスタリスクでマスクされます **Vault secret** - 値には、vault secret へのパス/クエリが含まれます。 値タイプを変更するには、値入力フィールドの末尾にあるボタンをクリックします。 ユーザーマクロ値の最大長は 2048 文字です。
Description	このマクロに関する詳細情報を提供するためのテキストフィールドです。

ユーザーマクロを設定する際は、以下のコンテキスト固有の動作に注意してください。

- ユーザーマクロがテンプレートのアイテムまたはトリガーで使用される場合は、それらのマクロをテンプレートにも追加することを検討してください（グローバルに定義されている場合でも）。そうすることで、テンプレートを XML にエクスポートして別のシステムにインポートした後も、*Text* タイプのマクロが期待どおりに動作します（シークレットマクロの値はエクスポートされません）
- ユーザーマクロがトリガー式で使用される場合、それらのマクロはパラメータまたは定数を参照している場合にのみ展開されます。ホスト、アイテムキー、関数、演算子、または別のトリガー式を参照している場合は展開されません（シークレットマクロはトリガー式では使用できません）
- ユーザーマクロが、ホストプロトタイプを持つローレベルディスカバリルールが設定されたホストで使用される場合、discovered hosts はそのホストで定義されたすべてのユーザーマクロを継承します

例

例 1

”Status of SSH daemon” item キーにホストレベルマクロを使用する。

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

この項目は、複数のホストで **{$SSH_PORT}** の値が定義されている場合に、複数のホストに割り当てることができます。

例 2

”CPU load is too high” トリガーでのホストレベルマクロの使用:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

このようなトリガーは、個々のホストで編集するのではなく、テンプレート上で作成されることになります。

Note:
値の量を関数のパラメーターとして使用する場合（例えば, **max(/host/key,#3)**),
 SOME_PERIOD => #3
 というようにマクロ定義にハッシュ記号を入れてください。

例 3

”CPU load is too high” トリガーで 2 つのマクロを使用する:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

マクロはトリガー関数（この例では関数 **min()**）のパラメータとして使用することができることに注意。

例 4

agent の利用不可条件と item の更新間隔を同期させる。:

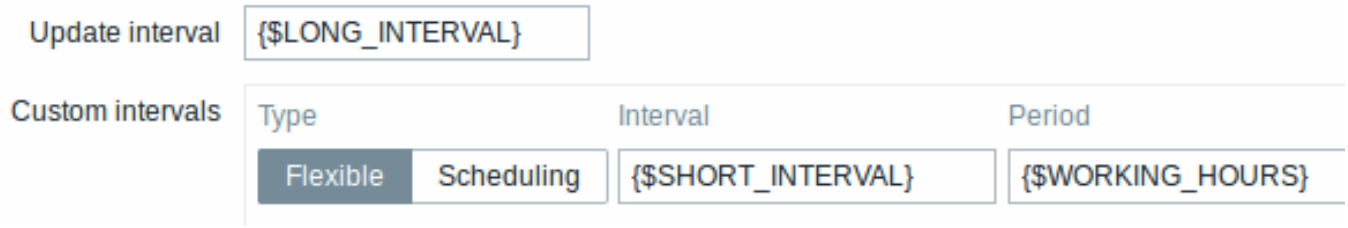
- {INTERVAL} マクロを定義し、item の更新間隔に使用する。
- {INTERVAL} を agent 使用不可トリガーのパラメータとして使用する。

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

例 5

稼働時間の設定を一元化します。

- 1-5,09:00-18:00 に等しいグローバルマクロ {$WORKING_HOURS} を作成します。
- *Administration* → *General* → *GUI* の *Working time* フィールドでこれを使用します。
- *Users* → *Users* のユーザーの *Media* タブにある *When active* フィールドでこれを使用します。
- 稼働時間中にアイテムをより高頻度でポーリングする設定にこれを使用します。

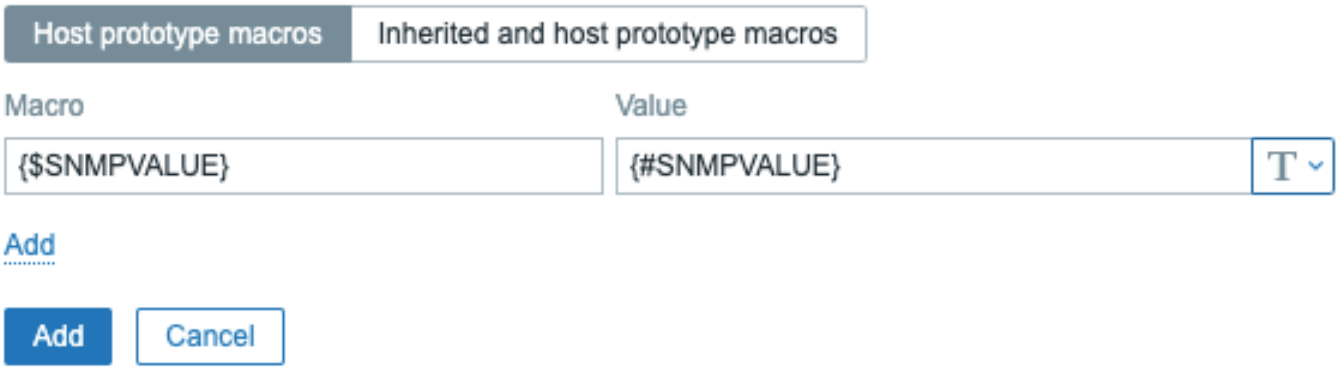


- *Time period* アクション条件でこれを使用します。
- 必要に応じて、*Administration* → *Macros* で稼働時間を調整します。

例 6

ホストプロトタイプマクロを使用して、検出されたホストの item を設定します:

ホストプロトタイプで、{$SNMPVALUE} を {#SNMPVALUE}low-level discovery マクロ値として定義します:



- ホストプロトタイプに *Generic SNMPv2* テンプレートを割り当てます。
- {SNMPVALUE} を使用します。汎用 SNMPv2 テンプレート項目の SNMP OID* フィールドに {$SNMPVALUE} を使用します。

ユーザーマクロのコンテキスト

user macros with context を参照してください。

3 コンテキストユーザーマクロ

概要

ユーザーマクロ でオプションのコンテキストを使用して、デフォルト値をコンテキスト固有の値で上書きすることができます。

マクロ名にコンテキストが追加されます。構文は、コンテキストが静的テキスト値であるかどうかによって異なります。

```
{$MACRO:"static text"}
```

または正規表現:

```
{$MACRO:regex:"regular expression"}
```

正規表現コンテキストを持つマクロは、ユーザーマクロ構成でのみ定義できることに注意してください。`regex:` プレフィックスがトリガー式のようにユーザーマクロコンテキストとして他の場所で使用されている場合、静的コンテキストとして扱われます。

コンテキストの引用はオプションです (重要な注意事項 も参照してください)

マクロコンテキストの例:

例	説明
{$LOW_SPACE_LIMIT}	コンテキストのないユーザーマクロ
{$LOW_SPACE_LIMIT:/tmp}	コンテキストを持つユーザーマクロ (静的文字列)
{$LOW_SPACE_LIMIT:regex:"^/tmp$"}	コンテキストを持つユーザーマクロ (正規表現)。{$LOW_SPACE_LIMIT:/tmp} と同じ。
{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}	コンテキストを持つユーザーマクロ (正規表現)。/var/log/ で始まるすべての文字列に一致。

ユースケース

コンテキストを持つユーザーマクロを定義して、トリガー式で (ローレベルディスカバリによって取得された値に基づく) より柔軟なしきい値を実現できます。たとえば、次のマクロを定義できます。

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:"^/[a-z]+$"} = 30

次に、ローレベルディスカバリマクロを、マウントされたファイルシステムディスカバリのトリガープロトタイプでマクロコンテキストとして使用できます:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

検出後、検出されたマウントポイントまたはファイルシステムの種類に応じて、さまざまな低容量しきい値がトリガーに適用されます。障害イベントは、次の場合に生成されます。

- /home フォルダーの空きディスク容量が 20% 未満
- 正規表現パターン (/etc、/tmp、/var など) に一致するフォルダーの空きディスク容量が 30% 未満
- 正規表現パターンに一致せず、/home ではないフォルダーの空きディスク容量が 10% 未満

重要事項

- コンテキストを持つ複数のユーザーマクロが存在する場合、Zabbix は最初に単純なコンテキストマクロを照合し、次にコンテキストマクロを未定義の順序で正規表現と照合しようとします。

Warning:
未定義の動作を避けるために、同じ文字列に一致する別のコンテキストマクロを作成しないでください。

- コンテキストを持つマクロがホスト、リンクされたテンプレート、またはグローバルで見つからない場合、コンテキストのないマクロが検索されます。
- コンテキストでは、ローレベルディスカバリマクロのみがサポートされています。その他のマクロはすべて無視され、プレーンテキストとして扱われます。

技術的には、マクロコンテキストはitem key パラメータと同様のルールを使用して指定されますが、, 文字がある場合、マクロコンテキストは複数のパラメータとして解析されません。

- コンテキストに } 文字が含まれている場合、または " 文字で始まる場合、マクロコンテキストは " で引用する必要があります。引用されたコンテキスト内の引用符は、\ 文字でエスケープする必要があります。
- \ 文字自体はエスケープされません。つまり、\ 文字で終わる引用符付きコンテキストを持つことは不可能です - マクロ {$MACRO:"a:\b\c\"} は無効です。
- コンテキスト内の先頭のスペースは無視されますが、末尾のスペースは無視されます:
  - たとえば、{$MACRO:A} は {$MACRO: A} と同じですが、{$MACRO:A } とは異なります。
- 先頭の引用符の前と末尾の引用符の後のすべてのスペースは無視されますが、引用符内のすべてのスペースは無視されません。
  - マクロ {$MACRO:"A"}、{$MACRO: "A"}、{$MACRO:"A" }、および {$MACRO: "A" } は同じですが、マクロ {$MACRO:"A"} および {$MACRO:" A "} ではありません。

{$MACRO:A}、{$MACRO: A}、および {$MACRO:"A"} のマクロは、同じコンテキストを持っているため、すべて同等です。これは、'key[a]'、'key[a]'、および'key["a"]' が意味的に同じである項目キーとは対照的ですが、一意性のために異なります。

4 Secret ユーザーマクロ

概要

Zabbix では、ユーザーマクロの値に含まれる機密情報を保護するために、2 つのオプションを提供しています。

- シークレットテキスト
- Vault シークレット

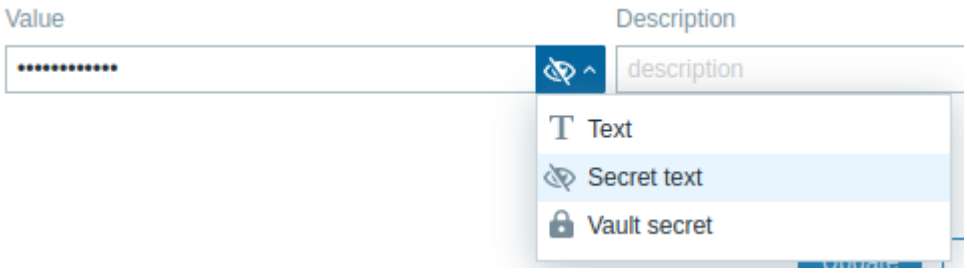
Note:
シークレットマクロの値は非表示になりますが、アイテムで使用することで表示される可能性があります。たとえば、外部スクリプトでシークレットマクロを参照する echo 文を使用すると、Zabbix サーバーは実際のマクロ値にアクセスできるため、そのマクロ値が Web インターフェースに表示される可能性があります。シークレットマクロの値がマスクされずに表示される場所については、locations を参照してください。

シークレットマクロはトリガー式では使用できません。

シークレットテキスト

シークレットテキストマクロでは、マクロ値はアスタリスクでマスクされます。

マクロ値をシークレットにするには、*Value* フィールドの末尾にあるボタンをクリックし、*Secret text* オプションを選択します。



設定を保存すると、その値を表示できなくなります。

マクロ値を変更するには、*Value* フィールドにカーソルを合わせ、*Set new value* ボタンをクリックします（ホバー時に表示されます）。



Set new value ボタンをクリックすると（またはマクロ値のタイプを変更すると）、現在の値は消去されます。

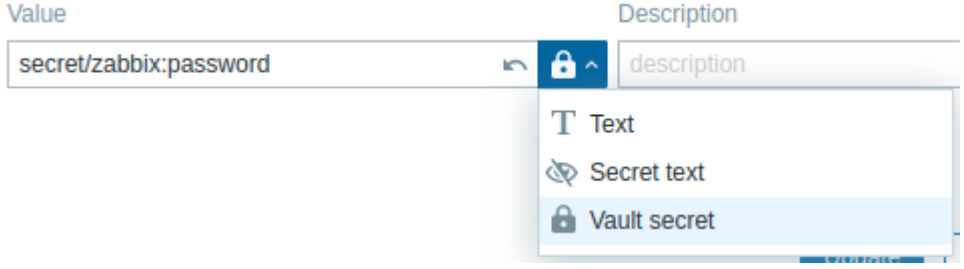
元の値は、*Value* フィールドの末尾にある 矢印をクリックすることで復元できます（新しい設定を保存する前のみ利用可能です）。
元の値を復元しても、その値が表示されることはありません。

Note:
シークレットマクロを含む URL は機能しません。これは、その中のマクロが”******” として展開されるためです。

Vault シークレット

Vault シークレットマクロでは、マクロ値は外部のシークレット管理ソフトウェア（vault）に保存されます。

Vault シークレットマクロを設定するには、*Value* フィールドの末尾にあるボタンをクリックし、*Vault secret* オプションを選択します。



マクロ値は vault シークレットを指している必要があります。
入力形式は vault プロバイダーによって異なります。プロバイダー固有の設定例については、以下を参照してください。

- HashiCorp
- CyberArk

Vault シークレットマクロの値は、設定データが更新されるたびに、Zabbix サーバー（および、*Resolve secret vault macros by* が設定 で *Zabbix server and proxy* に設定されている場合は Zabbix プロキシ）によって vault から取得され、その後設定キャッシュに保存されます。
Zabbix サーバーと Zabbix プロキシは、異なる vault を使用する場合があります。

Resolve secret vault macros by が設定 で *Zabbix server* に設定されている場合、vault シークレットはサーバーによってのみ取得され、Zabbix プロキシは各設定同期時に Zabbix サーバーから Vault シークレットマクロの値を受け取り、それを自身の設定キャッシュに保存します。
これは、Zabbix プロキシが再起動後、Zabbix サーバーから設定更新を受信するまでデータ収集を開始できないことを意味します。

vault からシークレット値を手動で更新するには、`secrets_reload` のruntime controlオプション（サーバーのみ）を使用します。

Zabbix サーバーとプロキシ間では暗号化を有効にする必要があります。そうでない場合、サーバーの警告メッセージがログに記録されます。

Warning:
マクロ値を正常に取得できない場合、その値を使用している対応するアイテムは未サポートになります。

マスク解除される場所

この一覧は、シークレットマクロの値がマスク解除されるパラメータの場所を示しています。

Note:
シークレットマクロの値が間接的に参照される場合、以下の場所でもマスクされたままになります。
たとえば、メディアタイプ（スクリプトまたは Webhook パラメータ）で使用される {ITEM.KEY}、{ITEM.KEY<1-9>}、{LLDRULE.KEY} 組み込みマクロ は、`net.tcp.port[192.0.2.0,80]` ではなく `net.tcp.port[******,******]` のように、マスクされたシークレットマクロを含むアイテムキーに展開されます。

コンテキスト		パラメータ
アイテム、アイテムのプロトタイプ、**LLD** ルール		
	アイテム	アイテムキーのパラメータ
	アイテムのプロトタイプ	アイテムプロトタイプキーのパラメータ
	ローレベルディスカバリルール	ディスカバリアイテムキーのパラメータ
	SNMP エージェント	*SNMP* コミュニティ
		コンテキスト名 (SNMPv3)
		セキュリティ名 (SNMPv3)
		認証パスフレーズ (SNMPv3)
		プライバシーパスフレーズ (SNMPv3)
	HTTP エージェント	*URL*
		クエリフィールド
		リクエストボディ
		ヘッダー
		ユーザー名
		パスワード
		SSL キーパスワード
	スクリプト	パラメータ
		スクリプト
	ブラウザ	パラメータ
		スクリプト
	データベースモニタ	*SQL* クエリ
	TELNET エージェント	スクリプト
		ユーザー名
		パスワード
	SSH エージェント	スクリプト
		ユーザー名
		パスワード
	シンプルチェック	ユーザー名
		パスワード
	JMX エージェント	ユーザー名
		パスワード
アイテム値の前処理		

コンテキスト		パラメータ
	JavaScript 前処理ステップ	スクリプト
Web シナリオ		
	Web シナリオ	変数値 ヘッダー値 *URL* クエリフィールド値 *POST* フィールド値 *Raw post*
	Web シナリオ認証	ユーザー パスワード *SSL* キーパスワード
コネクタ		
	コネクタ	*URL* ユーザー名 パスワード トークン *HTTP* プロキシ *SSL* 証明書ファイル *SSL* キーファイル *SSL* キーパスワード
ネットワークディスカバリ		
	SNMP	*SNMP* コミュニティ コンテキスト名 (SNMPv3) セキュリティ名 (SNMPv3) 認証パスフレーズ (SNMPv3) プライバシーパスフレーズ (SNMPv3)
グローバルスクリプト		
	Webhook	*JavaScript* スクリプト *JavaScript* スクリプトパラメータ値
	Telnet	ユーザー名 パスワード
	SSH	ユーザー名 パスワード
	スクリプト	スクリプト
メディアタイプ		
	スクリプト	スクリプトパラメータ
	Webhook	パラメータ
IPMI 管理		
	ホスト	ユーザー名 パスワード

5 ローレベルディスカバリマクロ

概要

ローレベルディスカバリ（LLD）機能内で使用されるマクロの種類があります。

```
{#MACRO}
```

これは LLD ルールで使用されるマクロで、ファイルシステム名、ネットワークインターフェース、SNMP OID などの実際の値を返します。

これらのマクロは、エンティティのプロトタイプ（アイテム、トリガー、グラフ、LLD ルール、ホスト、ホストグループのプロトタイプ）の作成に使用できます。
実際のファイルシステム、ネットワークインターフェース、仮想マシンなどのディスカバリ時に、これらのマクロは実際の値に置き換えられ、実際のエンティティを作成するための基礎となります。

一部のローレベルディスカバリマクロは、Zabbix の LLD 機能にあらかじめ含まれています - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}。
ただし、カスタムのローレベルディスカバリルールを作成する際に、これらの名前に従う必要はありません。
その場合は、他の任意の LLD マクロ名を使用し、その名前を参照できます。

サポートされるデータ型

カスタムディスカバリルールを定義する際、LLD マクロ用に JSON オブジェクトで返されるプロパティ値は、次のいずれかのプリミティブ型である必要があります。

- string
- number
- boolean

配列、オブジェクト、および null 値はサポートされていません。
そのような値を参照する LLD マクロは展開されないままとなり、アイテムの前処理および作成時にそのままの文字列（例: `'{#MY_MACRO}'`）として表示されます。

サポートされている場所

LLD マクロは、以下で使用できます。

- ローレベルディスカバリルールのフィルター内
- アイテムプロトタイプおよびディスカバリプロトタイプの以下の項目
  - 名前
  - キーパラメータ
  - 単位
  - 更新間隔[1]
  - タイムアウト[1]
  - 履歴保存期間[1]
  - トレンド保存期間[1]
  - アイテム値の前処理ステップ
  - SNMP OID
  - IPMI センサーフィールド
  - 計算/集計アイテムの式。具体的には以下
    - 式の定数および関数パラメータ
    - アイテムキーのパラメータ
  - 集計アイテムのフィルター条件（ホストグループ名およびタグ名）
  - SSH スクリプトおよび Telnet スクリプト
  - データベース監視の SQL クエリ
  - JMX アイテムのエンドポイントフィールド
  - 説明
  - HTTP エージェントの URL フィールド
  - HTTP エージェントの HTTP クエリフィールド
  - HTTP エージェントのリクエストボディフィールド
  - HTTP エージェントの必須ステータスコードフィールド
  - HTTP エージェントのヘッダーフィールドのキーおよび値
  - HTTP エージェントの HTTP 認証ユーザー名フィールド
  - HTTP エージェントの HTTP 認証パスワードフィールド
  - HTTP エージェントの HTTP プロキシフィールド
  - HTTP エージェントの HTTP SSL 証明書ファイルフィールド
  - HTTP エージェントの HTTP SSL キーファイルフィールド
  - HTTP エージェントの HTTP SSL キーパスワードフィールド
  - タグ
- トリガープロトタイプの以下の項目
  - 名前
  - 運用データ
  - 式（定数および関数パラメータ内のみ）
  - URL
  - 説明
  - タグ
- グラフプロトタイプの以下の項目
  - 名前
- ホストプロトタイプの以下の項目
  - 名前
  - 表示名
  - カスタムインターフェースフィールド: IP、DNS、ポート、SNMP v1/v2 コミュニティ、SNMP v3 コンテキスト名、SNMP v3 セキュリティ名、SNMP v3 認証パスフレーズ、SNMP v3 プライバシーパスフレーズ
  - ホストグループプロトタイプ名
  - ホストタグ値
  - ホストマクロ値
  - (完全な一覧を参照)

これらすべての場所では、ローレベルディスカバリルールのフィルターを除き、LLD マクロを静的ユーザーマクロコンテキスト内で使用できます。

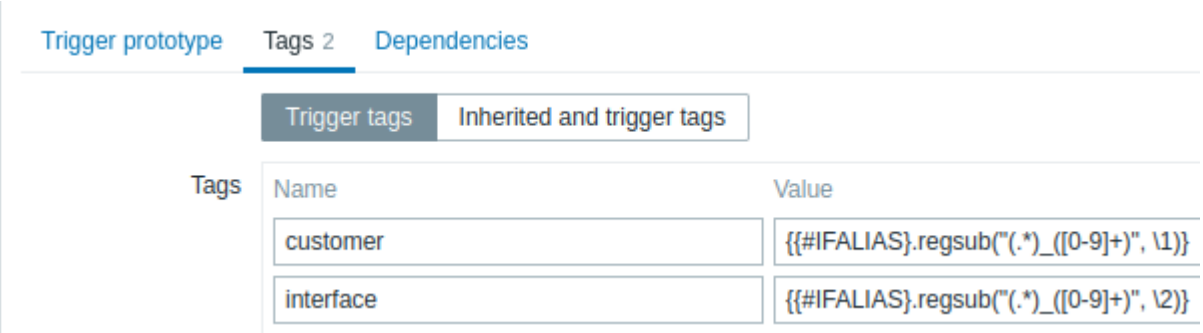
マクロ関数の使用

マクロ関数はローレベルディスカバリマクロでサポートされています（ローレベルディスカバリルールのフィルターを除く）。これにより、正規表現を使用してマクロ値の特定の部分を抽出できます。

例えば、イベントタグ付けのために、次の LLD マクロから顧客名とインターフェース番号を抽出したい場合があります。

```
{#IFALIAS}=customername_1
```

これを行うには、トリガープロトタイプのイベントタグ値フィールドで、マクロに対して regsub マクロ関数を使用できます。



引用符で囲まれていないアイテムのkey パラメータではカンマを使用できないため、マクロ関数を含むパラメータは引用符で囲む必要があります。パラメータ内の二重引用符をエスケープするには、バックスラッシュ（\）文字を使用してください。例:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

マクロ関数の構文の詳細については、マクロ関数を参照してください。

マクロ関数は、Zabbix 4.0 以降、ローレベルディスカバリマクロでサポートされています。

キーパラメータのないアイテムのプロトタイプ

アイテムキーのパラメータを使用しない場合は、LLD マクロをitem keyのパラメータ括弧 [...] 内に配置して、パラメータとして扱われ、ディスカバリ中に置換されるようにします。

```
v_[{#MACRO}]
```

脚注

[1] で示されたフィールドでは、単一のマクロがフィールド全体を満たしている必要があります。1 つのフィールド内で複数のマクロを使用すること、またはマクロをテキストと混在させることはサポートされていません。

6 式マクロ

概要

式マクロを使用すると、フィールド内で計算を実行できます。

その値は、まず内部のマクロを解決し、その後で生成された式を評価することで計算されます。

構文:

```
{?EXPRESSION}
```

EXPRESSION は、トリガー式 と同じ構文を使用し、同じ関数をサポートします。

例:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

式マクロをサポートする場所については、サポートされているマクロ の表で"{?EXPRESSION}" を検索してください。

使用上の注意:

- トリガー式/リカバリー式の関数値を参照するには、{FUNCTION.*} マクロを使用します。
- ホストおよびアイテムを参照するには、{HOST.HOST<1-9>} および{ITEM.KEY<1-9>} マクロを使用します。
- テンプレートでは、テンプレート名の代わりに{HOST.HOST<1-9>} マクロを使用するか、最初のホストについてはホストを完全に省略してください。例えば {?avg(//item1,1h)} のようにします。これは、テンプレートのリンク 中にテンプレート名がホスト名に置き換えられないためです。

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

イベント名で式マクロを使用する例については、トリガー式の例 も参照してください。

12 ユーザーとユーザーグループ

概要

Zabbix のすべてのユーザは、Web ベースのフロントエンドから Zabbix アプリケーションにアクセスします。各ユーザーには
 固有のログイン名とパスワードが割り当てられます。

全てのユーザパスワードは暗号化され、Zabbix のデータベースに保存されます。ユーザーは、UNIX に応じた設定をしていない限り、
 ユーザー ID とパスワードを使用して UNIX サーバに直接ログインすることはできません。Web サーバとユーザブラウザ間の通信は SSL を使用して
 保護することができます。

柔軟なuser permission schemaにより、以下の権限を制限し差別化することができます:

- Zabbix フロントエンドの管理用機能へのアクセス
- フロントエンドでの特定のアクションの実行
- ホストグループ内の監視対象ホストへのアクセス
- 特定の API メソッドの使用

1 ユーザの設定

概要

初期状態の Zabbix インストールには、あらかじめ 2 つのユーザーが定義されています。

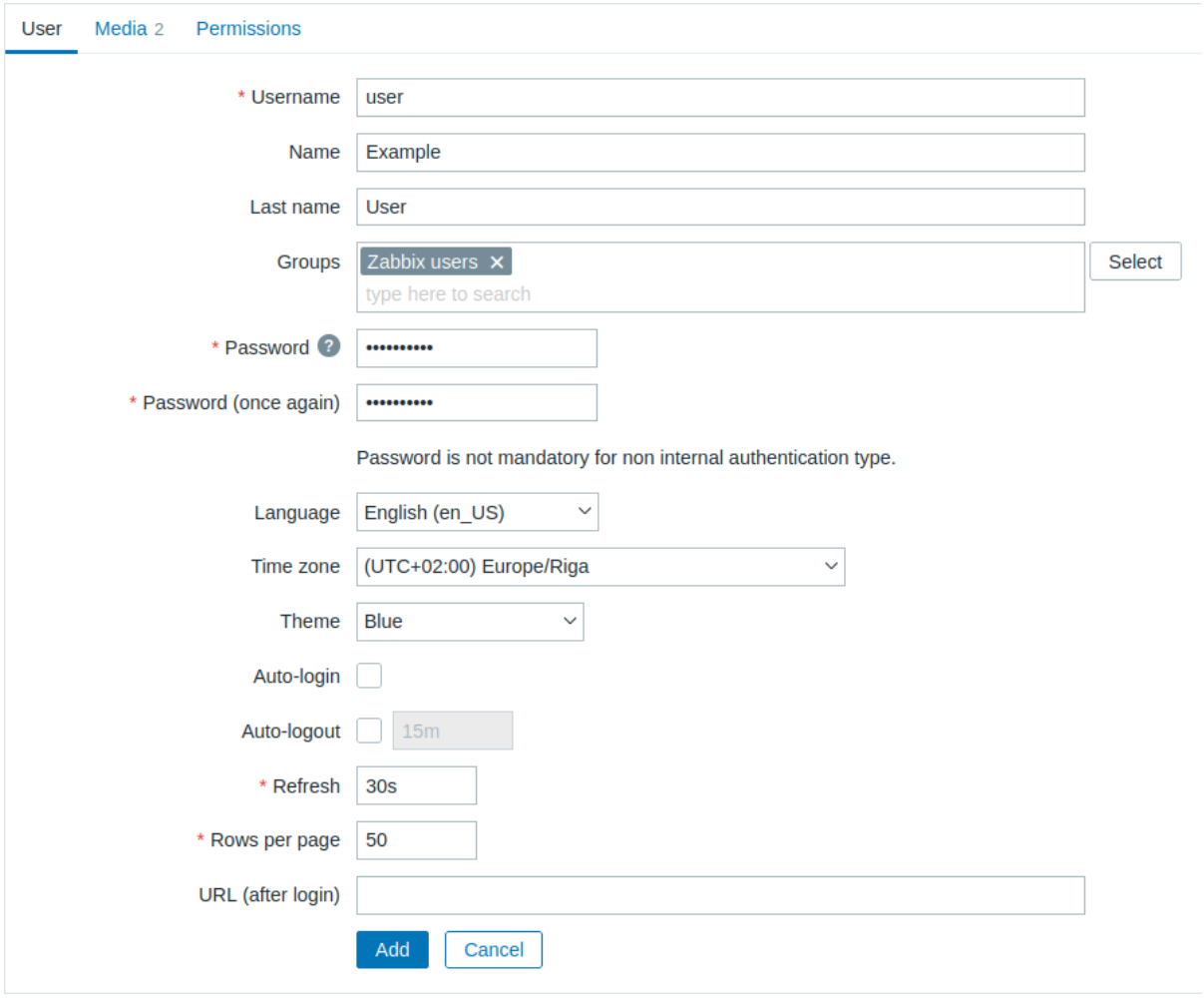
- *Admin* - すべての権限を持つ Zabbix スーパーユーザー です。
- *guest* - 特別な Zabbix ユーザー です。'guest' ユーザーはデフォルトで無効になっています。これを Guests ユーザーグループに追加すると、このユーザーでログインして、Zabbix の監視ページにアクセスできます。なお、デフォルトでは'guest' には Zabbix オブジェクトに対する権限はありません。

ユーザーを設定するには、次の手順を実行します。

- *Users* → *Users* に移動します。
- *Create user* をクリックします（既存のユーザーを編集する場合は、ユーザー名をクリックします）。
- フォームでユーザー属性を編集します。

一般属性

ユーザータブには、ユーザーの一般属性が含まれています。



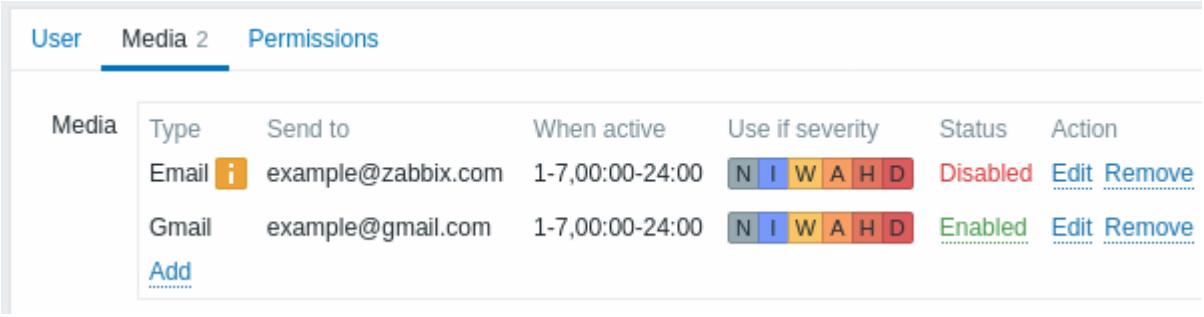
必須入力フィールドには、すべて赤いアスタリスクが付いています。

パラメータ	説明
Username	ログイン名として使用される、一意のユーザー名です。
Name	ユーザーの名（任意）。 空でない場合、障害対応メッセージ情報および通知の受信者情報に表示されます。
Last name	ユーザーの姓（任意）。 空でない場合、障害対応メッセージ情報および通知の受信者情報に表示されます。
Groups	ユーザーが所属するユーザーグループを選択します。このフィールドはオートコンプリート対応のため、ユーザーグループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。下にスクロールして選択してください。あるいは、*Select* をクリックしてグループを追加することもできます。選択済みの項目を削除するには'x' をクリックします。 ユーザーグループへの所属によって、ユーザーがアクセス可能なホストグループおよびホストが決まります。
Password	ユーザーパスワードを入力するための 2 つのフィールド、またはユーザーがすでに存在する場合は *Change password* ボタンが表示されます。 *Change password* ボタンをクリックすると、新しいパスワードを入力するための 2 つのフィールドが開きます。 *Super admin role* を持つユーザーが自分自身のパスワードを変更する場合、*Change password* ボタンをクリックすると、現在の（古い）パスワードを入力するための追加フィールドも開きます。 パスワードの変更に成功すると、パスワードが変更されたユーザーはすべてのアクティブなセッションからログアウトされます。 なお、パスワードを変更できるのは、Zabbix の内部認証を使用しているユーザーのみです。
Language	Zabbix Web インターフェースの言語です。 翻訳を機能させるには、php gettext 拡張が必要です。
Time zone	ユーザーレベルでグローバルなタイムゾーンを上書きするタイムゾーンを選択するか、グローバルなタイムゾーン設定を使用するには **System default** を選択します。

パラメータ	説明
Theme	Web インターフェースの外観を定義します。 **System default** - デフォルトのシステム設定を使用 **Blue** - 標準の青色テーマ **Dark** - 代替のダークテーマ **High-contrast light** - 高コントラストのライトテーマ **High-contrast dark** - 高コントラストのダークテーマ
Auto-login	このチェックボックスをオンにすると、Zabbix がユーザーを記憶し、30 日間自動的にログインします。 *Remember for 30 days* を使用してログインした場合： - 自動ログアウトはリセットされます（セッションは 30 日間維持されます）。 - シームレスな再認証のため、自動ログインが有効になります。 *Remember for 30 days* を使用せずにログインした場合： - シームレスな再認証のため、自動ログインが有効になります。 - 自動ログアウトは標準のタイムアウト設定に従います。 この機能にはブラウザの Cookie が使用されます。
Auto-logout	このチェックボックスをオンにすると、設定した秒数後にユーザーは自動的にログアウトされます (最小 90 秒、最大 1 日)。 なお、*Remember for 30 days* が有効な場合、セッションが全期間にわたって延長されるため、この設定は上書きされます。 時間サフィックスに対応しています。例：90s、5m、2h、1d。 なお、次の場合、このオプションは機能しません。 * グローバル設定オプション”Show warning if Zabbix server is down” が有効で、Zabbix Web インターフェースを開いたままにしている場合。 * 監視メニューのページがバックグラウンドで情報を更新している場合。 * *Remember me for 30 days* オプションをチェックしてログインした場合。
Refresh	グラフ、プレーンテキストデータなどに使用する更新間隔を設定します。0 に設定すると無効にできます。 時間サフィックスに対応しています。例：90s、5m、1h。
Rows per page	リストで 1 ページあたりに表示する行数を指定できます。
URL (after login)	ログイン成功後に、ユーザーを特定の URL へ移動させることができます。たとえば、*Problems* ページなどです。

ユーザーメディア

メディアタブには、そのユーザーに定義されているすべてのメディアの一覧が表示されます。メディアは通知の送信に使用されます。



追加をクリックして、ユーザーにメディアを割り当てます。

メディアタイプが無効になっている場合:

- 名前の後に黄色の情報アイコンが表示されます。
- ステータス列に無効と表示されます。

Note:
自分自身のメディア詳細を変更するためのユーザー権限は、ユーザーロールに基づいて付与/取り消しできます（自分のメディアの作成と編集オプションを参照）。他のユーザーのメディア詳細を変更するための特権管理者ユーザー権限も、ユーザーロールに基づいて付与/取り消しできます（ユーザーメディアの作成と編集オプションを参照）。

ユーザーメディアの設定の詳細については、メディアタイプ セクションを参照してください。

権限

権限タブには、以下の要素に関する情報が含まれています。

- ユーザーロール（新規作成されたすべてのユーザーに必須）。これは *Super admin* タイプのユーザーのみが変更できます。

Attention:
ユーザーは、user role なしでは作成できません（Zabbix のUser API を使用する場合を除く）。以前に作成された、ロールを持たないユーザーは、ロールを割り当てなくても引き続き編集できます。ただし、いったんロールが割り当てられると、削除はできず、変更のみ可能です。

 ロールを持たないユーザーは、LDAP/SAML 情報が Zabbix で設定されたユーザーグループのマッピングと一致している場合に限り、LDAP またはSAML 認証を使用してのみ Zabbix にログインできる点に注意してください。

- ユーザータイプ（*User*、*Admin*、*Super admin*）。これはユーザーロール設定で定義されます。
- ユーザーがアクセスできるホストグループおよびテンプレートグループ。
  - *User* および *Admin* タイプのユーザーは、デフォルトではどのグループ、テンプレート、ホストにもアクセスできません。このようなアクセスを許可するには、対象エンティティへの権限が設定されたユーザーグループにユーザーを含める必要があります。
- Zabbix Web インターフェースのセクションおよび要素、モジュール、API メソッドへのアクセス権。
  - アクセスが許可されている要素は緑色で表示され、拒否されている要素は薄い灰色で表示されます。
- 特定のアクションを実行する権限。
  - ユーザーに許可されているアクションは緑色で表示され、拒否されているアクションは薄い灰色で表示されます。

詳細はPermissions ページを参照してください。

2 ユーザー権限

概要

Zabbix の権限は、ユーザータイプ、カスタマイズされたユーザーロール、およびユーザーグループに基づいて指定されるホストへのアクセスに依存します。

ユーザータイプ

Zabbix の権限は、主にユーザータイプによって決まります。

- ユーザー - メニューセクションへのアクセス権が制限されており（以下を参照）、デフォルトではどのリソースにもアクセスできません。ホストグループまたはテンプレートグループへの権限は、明示的に割り当てる必要があります。
- 管理者 - メニューセクションへのアクセス権が一部付与されています（以下を参照）。デフォルトではどのホストグループにもアクセスできません。ホストグループまたはテンプレートグループへの権限は、明示的に付与する必要があります。
- 特権管理者 - すべてのメニューセクションにアクセスできます。すべてのホストグループおよびテンプレートグループに対して読み書きアクセス権を持ちます。特定のグループへのアクセスを拒否しても、この権限を取り消すことはできません。

メニューアクセス

Note:
一部の UI 要素へのアクセス制限は、そのページを開けなくするだけであり、インターフェースの他の部分から基盤となるデータへアクセスする機能までは取り除きません。

次の表は、ユーザータイプごとの Zabbix メニューセクションへのアクセスを示しています。

メニューセクション		ユーザー	管理者	特権管理者
ダッシュボード		+	+	+
監視		+	+	+
	障害	+	+	+
	ホスト	+	+	+
	最新データ	+	+	+
	マップ	+	+	+
	ディスカバリ		+	+
サービス		+	+	+
	サービス	+	+	+
	SLA		+	+
	SLA レポート	+	+	+
インベントリ		+	+	+
	概要	+	+	+
	ホスト	+	+	+
レポート		+	+	+
	システム情報			+

メニューセクション		ユーザー	管理者	特権管理者
	定期レポート		+	+
	可用性レポート	+	+	+
	トリガー上位 *100* 件	+	+	+
	監査ログ			+
	アクションログ			+
	通知		+	+
データ収集			+	+
	テンプレートグループ		+	+
	ホストグループ		+	+
	テンプレート		+	+
	ホスト		+	+
	メンテナンス		+	+
	イベント相関			+
	ディスカバリ		+	+
アラート			+	+
	トリガーアクション		+	+
	サービスアクション		+	+
	ディスカバリアクション		+	+
	自動登録アクション		+	+
	内部アクション		+	+
	メディアタイプ			+
	スクリプト			+
ユーザー				+
	ユーザーグループ			+
	ユーザーロール			+
	ユーザー			+
	API トークン			+
	認証			+
管理				+
	一般			+
	監査ログ			+
	ハウスキーピング			+
	プロキシグループ			+
	プロキシ			+
	マクロ			+
	キュー			+

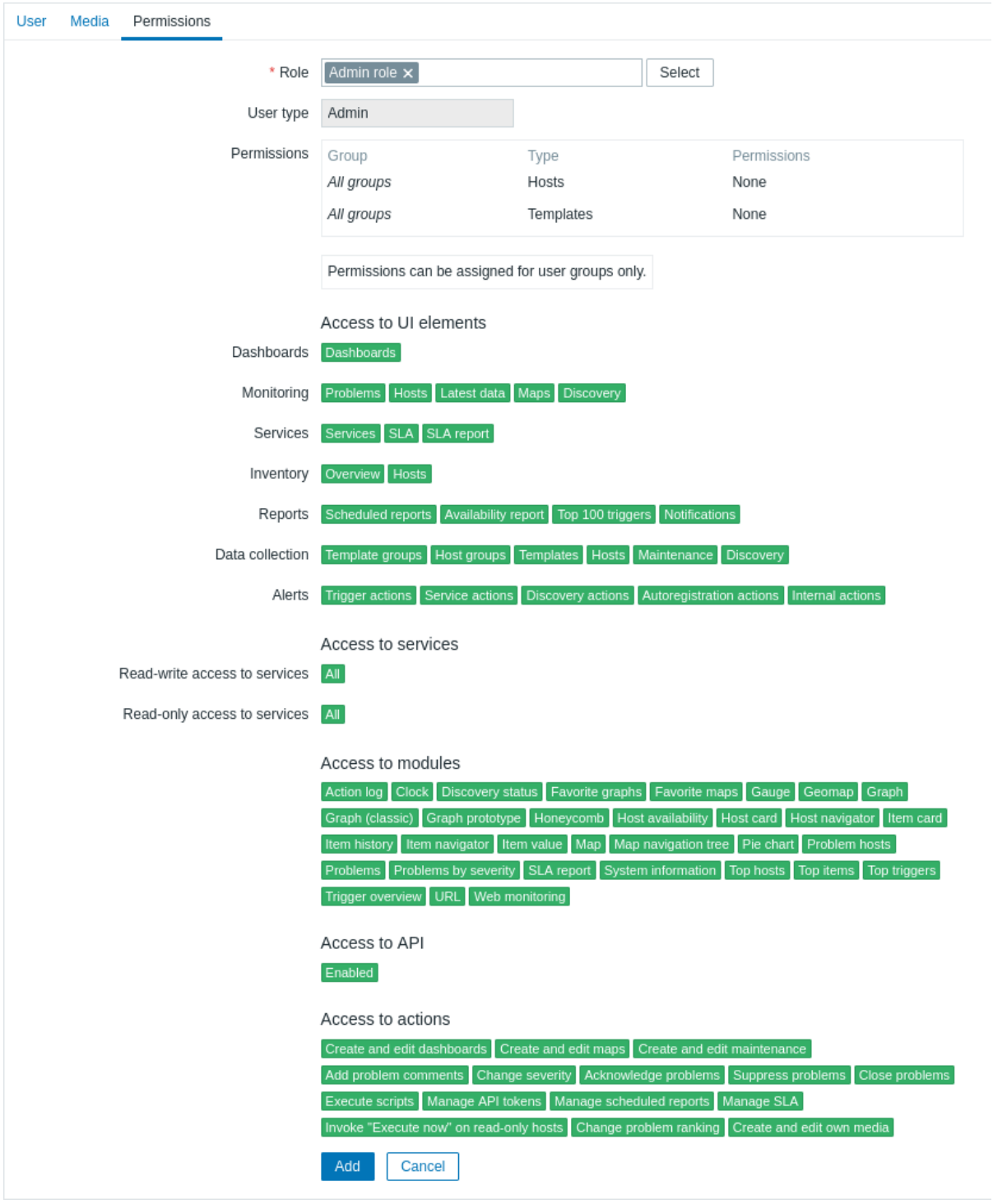
ユーザーロール

ユーザーロールを使用すると、ユーザータイプで定義された権限に対してカスタム調整を行うことができます。
権限を追加することはできませんが（それはユーザータイプの権限を超えるため）、一部の権限を取り消すことはできます。

さらに、ユーザーロールはメニューセクションへのアクセスだけでなく、サービス、モジュール、API メソッド、および Web インターフェース内のさまざまなアクションへのアクセスも決定します。

ユーザーロール は、Super admin ユーザーによって ユーザー → ユーザーロールセクションで設定されます。

ユーザーロールは、Super admin ユーザーによって、ユーザー設定フォームの 権限タブでユーザーに割り当てられます。



ホストへのアクセス

Zabbix における任意のホストおよびテンプレートのデータへのアクセスは、ユーザーグループに対して、ホスト/テンプレートグループレベルでのみ付与されます。

これは、個々のユーザーに対してホスト（またはホストグループ）へのアクセスを直接付与することはできないことを意味します。ホストへのアクセスは、そのホストを含むホストグループへのアクセス権が付与されたユーザーグループのメンバーである場合にのみ付与できます。

同様に、ユーザーがテンプレートへのアクセス権を付与されるのは、そのテンプレートを含むテンプレートグループへのアクセス権が付与されたユーザーグループのメンバーである場合に限られます。

3 ユーザグループ

概要

ユーザーグループを使用すると、組織上の目的とデータへの権限割り当ての両方のために、ユーザーをグループ化できます。ホストグループおよびテンプレートグループのデータの閲覧権限と設定権限は、個々のユーザーではなく、ユーザーグループに割り当てら

れます。

多くの場合、あるユーザーグループに対して利用可能な情報と、別のユーザーグループに対して利用可能な情報を分けることが有効です。これは、ユーザーをグループ化し、その後ホストグループおよびテンプレートグループに対して異なる権限を割り当てることで実現できます。

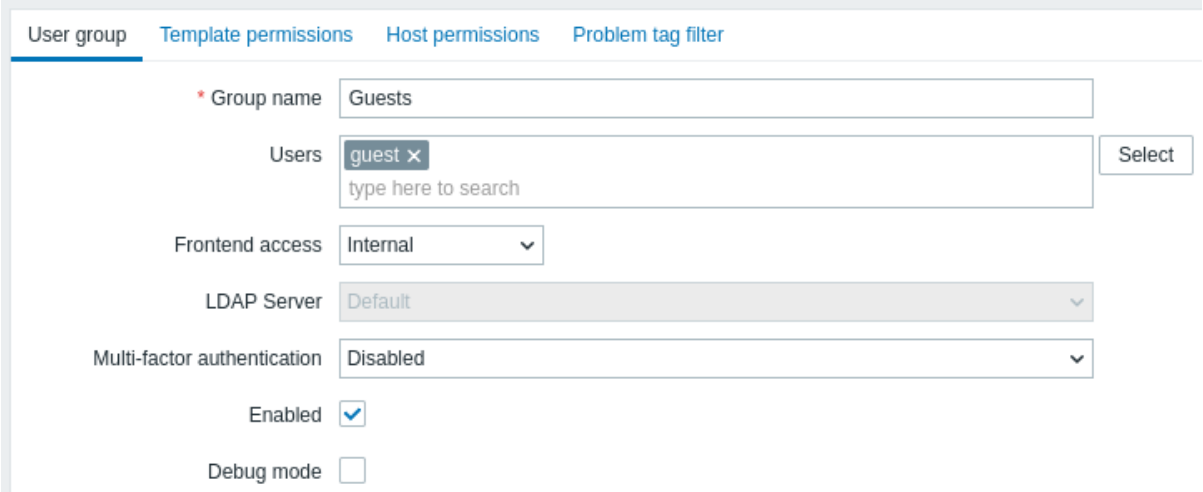
ユーザーは任意の数のグループに所属できます。

設定

ユーザーグループを設定するには、以下の手順を実行します。

- *Users* → *User groups* に移動します
- *Create user group* をクリックします（または、既存のグループを編集する場合はグループ名をクリックします）
- フォームでグループ属性を編集します

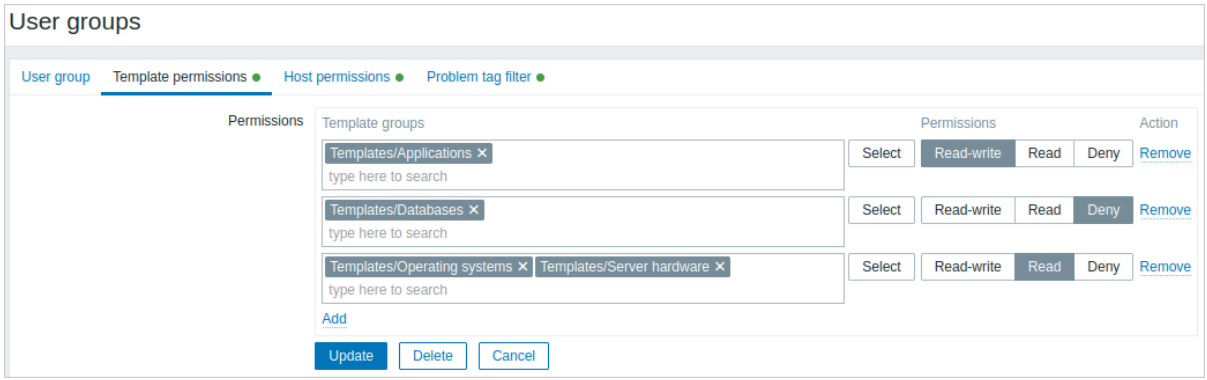
User group タブには、グループの一般属性が含まれています。



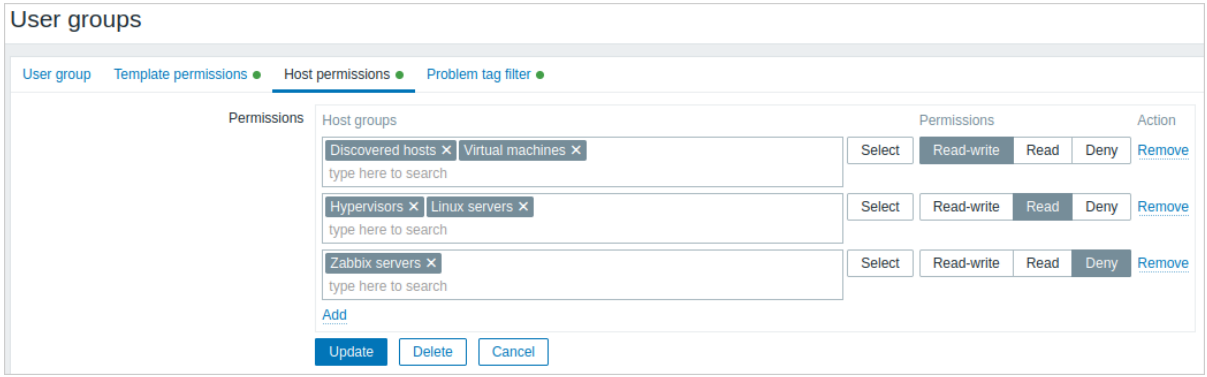
必須入力フィールドには、すべて赤いアスタリスクが付いています。

Parameter	Description
Group name	一意のグループ名。
Users	グループにユーザーを追加するには、既存ユーザーの名前を入力し始めます。一致するユーザー名のドロップダウンが表示されたら、スクロールして選択します。 または、*Select* ボタンをクリックして、ポップアップからユーザーを選択することもできます。
Frontend access	グループのユーザーをどのように認証するかを指定します。 **System default** - デフォルトの認証方式を使用します（グローバル に設定されたもの）。 **Internal** - Zabbix の内部認証を使用します（グローバルで LDAP 認証が使用されている場合でも）。HTTP 認証がグローバルデフォルトの場合は無視されます。 **LDAP** - LDAP 認証を使用します（グローバルで内部認証が使用されている場合でも）。HTTP 認証がグローバルデフォルトの場合は無視されます。 **Disabled** - このグループの Zabbix Web インターフェースへのアクセスを禁止します
LDAP server	ユーザー認証に使用するLDAP server を選択します。 このフィールドは、*Frontend access* が LDAP または System default に設定されている場合にのみ有効です。
Multi-factor authentication	ユーザー認証に使用する多要素認証のmethod を選択します。 **Default** - MFA 設定でデフォルトとして設定された方式を使用します。このオプションは、MFA が有効な場合、新しいユーザーグループでデフォルトで選択されます。 **<Method name>** - 選択した方式を使用します（例: ”Zabbix TOTP”）。 **Disabled** - このグループでは MFA は無効です。このオプションは、MFA が無効な場合、新しいユーザーグループでデフォルトで選択されます。 ユーザーが MFA が有効な複数のユーザーグループに属している場合（または少なくとも 1 つのグループで MFA が有効な場合）、次の認証ルールが適用されることに注意してください。いずれかのグループが”Default” MFA 方式を使用している場合、その方式でユーザーを認証します。それ以外の場合は、アルファベット順で最初の方式が認証に使用されます。
Enabled	ユーザーグループおよびグループメンバーのステータス。 *Checked* - ユーザーグループとユーザーは有効です *Unchecked* - ユーザーグループとユーザーは無効です
Debug mode	このチェックボックスをオンにすると、ユーザーに対してdebug mode が有効になります。

Template permissions タブでは、テンプレートグループ（およびそれに含まれるテンプレート）データに対するユーザーグループのアクセス権を指定できます。



Host permissions タブでは、ホストグループ（およびそれに含まれるホスト）データに対するユーザーグループのアクセス権を指定できます。



Add をクリックして、テンプレート/ホストグループ（親グループまたはネストされたグループ）を選択し、それらに権限を割り当てます。グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。または、*Select* をクリックして、すべてのグループを一覧表示するポップアップウィンドウを開くこともできます。

その後、オプションボタンを使用して、選択したグループに権限を割り当てます。設定可能な権限は次のとおりです。

- **Read-write** - グループへの読み書きアクセス
- **Read** - グループへの読み取り専用アクセス
- **Deny** - グループへのアクセスを拒否

同じテンプレート/ホストグループが異なる権限設定で複数行に追加された場合は、最も厳しい権限が適用されます。

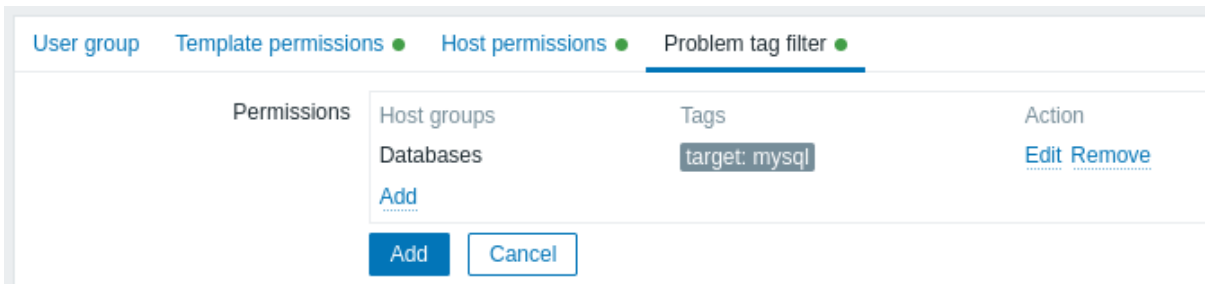
Super admin ユーザーは、ネストされたグループに親グループと同じ権限レベルを強制できることに注意してください。これは、host/template グループ設定フォームで行えます。

Template permissions タブと **Host permissions** タブは、同じパラメータセットをサポートしています。

グループに対する現在の権限は *Permissions* ブロックに表示され、そこで変更または削除できます。

Note:
ユーザーグループがホストに対して **Read-write** 権限を持ち、そのホストにリンクされたテンプレートに対して **Deny** または権限なしである場合、そのグループのユーザーはホスト上のテンプレート化されたアイテムを編集できず、テンプレート名は *Inaccessible template* と表示されます。

Problem tag filter タブでは、タグ名と値でフィルタリングされた障害をユーザーグループが表示できるようにする、タグベースの権限を設定できます。

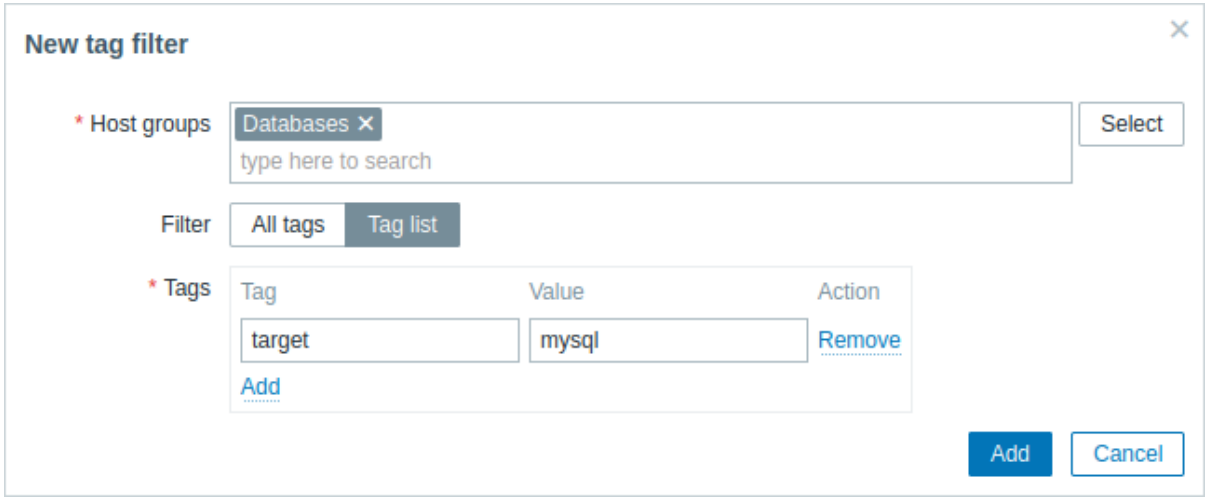


Add をクリックして、ホストグループを選択します。タグフィルターを適用するホストグループを選択するには、*Select* をクリックして既存のホストグループの完全な一覧を表示するか、ホストグループ名の入力を開始して一致するグループのドロップダウンを表示します。障害タグフィルターはテンプレートグループには適用できないため、表示されるのはホストグループのみです。

その後、*All tags* から *Tag list* に切り替えて、フィルタリング用の特定のタグとその値を設定できます。値のないタグ名は追加できますが、名前のない値は追加できません。*Permissions* ブロックには最初の 3 つのタグのみ（値がある場合はその値も含む）が表示されます。さらに多くある場合は、 ••• アイコンをクリックするか、その上にマウスカーソルを置くことで確認できます。

タグフィルターを使用すると、ホストグループへのアクセス権と、障害を表示できるかどうかを分離できます。

たとえば、データベース管理者が"MySQL" データベースの障害だけを確認する必要がある場合、まずデータベース管理者用のユーザーグループを作成し、その後でタグ名"target" と値"mysql" を指定する必要があります。



タグ名"target" が指定され、値フィールドが空白のままの場合、そのユーザーグループは、選択したホストグループに対してタグ名"target" を持つすべての障害を表示できます。*All tags* が選択されている場合、ユーザーグループは指定されたホストグループのすべての障害を表示できます。

タグ名とタグ値が正しく指定されていることを確認してください。そうでない場合、ユーザーグループには障害がまったく表示されません。

ユーザーが複数のユーザーグループのメンバーである場合の例を見てみましょう。この場合、フィルタリングではタグに対して OR 条件が使用されます。

User group A			**User group B**			両方のグループに属するユーザー（メンバー）に表示される結果
Tag filter						
Host group	*Tag name*	*Tag value*	*Host group*	*Tag name*	*Tag value*	
Databases	target	mysql	Databases	target	oracle	target:mysql または target:oracle の障害が表示される
Databases	set to: *All tags*		Databases	target	oracle	すべての障害が表示される
Problem tag filter で未設定			Databases	target	oracle	target:oracle の障害が表示される

Attention:
フィルターを追加すると（たとえば、特定のホストグループ"Databases" ですべてのタグを対象にした場合）、他のホストグループの障害は表示できなくなります。

複数のユーザーグループからのアクセス

ユーザーは任意の数のユーザーグループに所属できます。これらのグループは、ホストまたはテンプレートに対して異なるアクセス権限を持つ場合があります。

そのため、権限のないユーザーが結果としてどのエンティティにアクセスできるようになるのかを把握することが重要です。以下の例では、ユーザーグループ A および B に所属するユーザーについて、ホスト **X**（ホストグループ 1 内）へのアクセスがさまざまな状況でどのような影響を受けるかを考えます。

- グループ A がホストグループ 1 に対して *Read* アクセスのみを持ち、グループ B がホストグループ 1 に対して *Read-write* アクセスを持つ場合、ユーザーは'X' に対して **Read-write** アクセスを取得します。

> **Attention:**
> 「Read-write」権限は「Read」権限より優先されます。

- 上記と同じシナリオで、'X' が同時にグループ A または B に対して 拒否されているホストグループ 2 にも属している場合、ホストグループ 1 に対する *Read-write* アクセスがあっても、'X' へのアクセスは 利用不可になります。
- グループ A に権限が定義されておらず、グループ B がホストグループ 1 に対して *Read-write* アクセスを持つ場合、ユーザーは'X' に対して **Read-write** アクセスを取得します。
- グループ A がホストグループ 1 に対して *Deny* アクセスを持ち、グループ B がホストグループ 1 に対して *Read-write* アクセスを持つ場合、ユーザーの'X' へのアクセスは 拒否されます。

その他の詳細

- ホストへの読み書きアクセス権を持つ Admin レベルのユーザーでも、それらが属するテンプレートグループへのアクセス権がない場合、テンプレートのリンク/リンク解除はできません。テンプレートグループへの読み取りアクセス権がある場合は、ホストへのテンプレートのリンク/リンク解除は可能ですが、テンプレート一覧にはテンプレートは表示されず、他の場所でテンプレートを操作することもできません。
- ホストへの読み取りアクセス権を持つ Admin レベルのユーザーは、設定セクションのホスト一覧にそのホストを表示できません。ただし、そのホストのトリガーは IT サービス設定でアクセス可能です。
- Super Admin 以外のすべてのユーザー（'guest' を含む）は、マップが空であるか、画像のみを含む場合に限り、ネットワークマップを表示できます。ホスト、ホストグループ、またはトリガーがマップに追加されると、権限設定が適用されます。
- 対象のホストへのアクセスが明示的に"denied" されている場合、Zabbix サーバーはアクションの実行先受信者として定義されたユーザーに通知を送信しません。

13 secrets の保存

概要

Zabbix は、安全な vault から機密情報を取得するように設定できます。以下のシークレット管理サービスがサポートされています: HashiCorp Vault KV Secrets Engine - Version 2、CyberArk Vault CV12。

シークレットは、以下の取得に使用できます:

- ユーザーマクロの値
- データベースアクセス認証情報

Zabbix は、シークレットが他者によって管理されていることを前提として、vault 内のシークレットへの読み取り専用アクセスを提供します。

特定の vault プロバイダーの設定については、以下を参照してください。

- HashiCorp の設定
- CyberArk の設定

シークレット値のキャッシュ

デフォルトでは、vault シークレットマクロの値は設定データの更新のたびに Zabbix サーバーによって取得され、その後設定キャッシュに保存されます。Zabbix プロキシは、設定同期のたびに Zabbix サーバーから vault シークレットマクロの値を受け取り、自身の設定キャッシュに保存します。

> **Attention:**
> Zabbix サーバーとプロキシ間では暗号化を有効にする必要があります。そうしないと、サーバーの警告メッセージがログに記録されます。

また、マクロ値が Zabbix サーバーと Zabbix プロキシによってそれぞれ独立して取得されるように設定することも可能です。

vault からキャッシュされたシークレット値の更新を手動でトリガーするには、コマンドラインオプション 'secrets_reload' を使用します。

Zabbix Web インターフェースのデータベース認証情報のキャッシュはデフォルトで無効になっていますが、zabbix.conf.php でオプション `$DB['VAULT_CACHE'] = true` を設定することで有効にできます。認証情報は、ファイルシステムの一時ファイルディレ

クトリを使用するローカルキャッシュに保存されます。Web サーバーでは、プライベートな一時フォルダーへの書き込みが許可されている必要があります（たとえば、Apache では設定オプション PrivateTmp=True を設定する必要があります）。データキャッシュを更新/無効化する頻度を制御するには、ZBX_DATA_CACHE_TTL 定数 を使用します。

TLS 設定

Zabbix コンポーネントと vault 間の通信で TLS を設定するには、認証局（CA）によって署名された証明書をシステム全体のデフォルト CA ストアに追加します。
別の場所を使用するには、Zabbix サーバー/プロキシ の設定パラメータ SSLCALocation でディレクトリを指定し、
そのディレクトリ内に証明書ファイルを配置してから、CLI コマンドを実行します:

```
c_rehash .
```

1 CyberArk の設定

このセクションでは、CyberArk Vault CV12 からシークレットを取得するように Zabbix を設定する方法について説明します。

vault は、公式のCyberArk ドキュメントに記載されているとおりにインストールおよび設定する必要があります。

Zabbix で TLS を設定する方法については、シークレットの保存を参照してください。

データベース認証情報

データベース認証情報を含むシークレットへのアクセスは、各 Zabbix コンポーネントごとに個別に設定します。

サーバーとプロキシ

Zabbix server またはproxy のデータベース認証情報を vault から取得するには、設定ファイルで次の設定パラメータを指定します。

- Vault - 使用する vault プロバイダー;
- VaultURL - vault サーバーの HTTP[S] URL;
- VaultDBPath - キー”Content” および”UserName” によって取得されるデータベース認証情報を含む vault シークレットへのクエリ（このオプションは、DBUser および DBPassword が指定されていない場合にのみ使用できます）;
- VaultTLSCertFile, VaultTLSKeyFile - SSL 証明書およびキーファイル名; これらのオプションの設定は必須ではありませんが、強く推奨されます;
- VaultPrefix - vault に応じた、vault パスまたはクエリ用のカスタムプレフィックス; 指定しない場合は、最適なデフォルト値が使用されます。

Attention:
設定パラメータ Vault、VaultURL、VaultTLSCertFile、VaultTLSKeyFile、VaultPrefix は、Zabbix サーバーがシークレット vault マクロを処理する際の vault 認証にも使用されます（configured されている場合は Zabbix プロキシでも使用されます）。Zabbix サーバーおよび Zabbix プロキシは、VaultDBPath の DB 認証情報を含む vault シークレットマクロを展開しません。

Zabbix サーバーおよび Zabbix プロキシは、起動時に *zabbix_server.conf* および *zabbix_proxy.conf* ファイルから vault 関連の設定パラメータを読み取ります。

例

1. *zabbix_server.conf* で、以下のパラメータを指定します。

```
Vault=CyberArk
VaultURL=https://127.0.0.1:1858
VaultDBPath=AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
VaultTLSCertFile=cert.pem
VaultTLSKeyFile=key.pem
VaultPrefix=/AIMWebService/api/Accounts?
```

2. Zabbix は、vault に以下の API リクエストを送信します。

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. vault のレスポンスには、キー”Content” と”UserName” が含まれます。

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

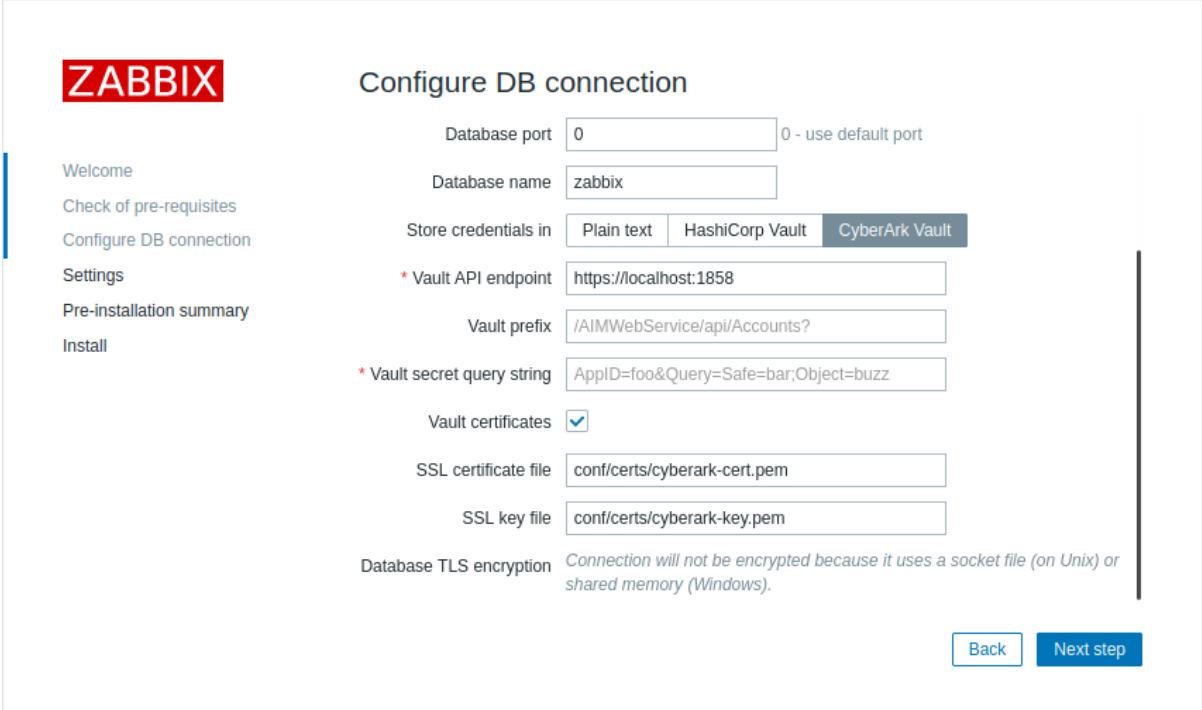
4. その結果、Zabbix はデータベース認証に以下の認証情報を使用します。

- ユーザー名: <username>
- パスワード: <password>

Web インターフェース

Zabbix Web インターフェース用に vault からデータベース認証情報を取得するには、Web インターフェースのインストール時に以下のパラメータを指定します。

1. *Configure DB Connection* ステップで、*Store credentials in* パラメータを”CyberArk Vault” に設定します。



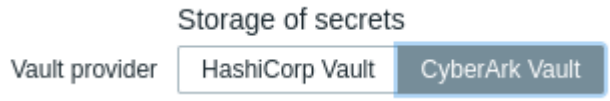
2. 次に、追加のパラメータを入力します。

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:1858	`scheme://host:port` 形式で、vault への接続に使用する URL を指定します。
Vault prefix	no	/AIMWebService/api/Accounts?	vault のパスまたはクエリに使用するカスタムプレフィックスを指定します。指定しない場合は、デフォルト値が使用されます。
Vault secret query string	yes		データベース認証情報の取得元を指定するクエリです。 例: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		チェックボックスをオンにすると、クライアント認証を設定するための追加パラメータが表示されます。このパラメータは任意ですが、CyberArk Vault との通信では有効にすることを強く推奨します。
SSL certificate file	no	conf/certs/cyberark-cert.pem	SSL 証明書ファイルへのパスです。ファイルは PEM 形式である必要があります。 証明書ファイルに秘密鍵も含まれている場合は、SSL key file パラメータは空のままにしてください。
SSL key file	no	conf/certs/cyberark-key.pem	クライアント認証に使用する SSL 秘密鍵ファイルの名前です。ファイルは PEM 形式である必要があります。

ユーザーマクロ値

Vault secret ユーザーマクロ値の保存に CyberArk Vault を使用するには、次のことを確認してください。

- Zabbix サーバーが CyberArk Vault と連携するように設定されていること。
- 管理 → 一般 → その他の *Vault provider* パラメータが「CyberArk Vault」に設定されていること。



Note:
Zabbix サーバー（および、設定されている場合は Zabbix プロキシ）には、vault から *Vault secret* マクロ値へアクセスするための権限が必要です。Zabbix Web インターフェースには、そのようなアクセスは必要ありません。

マクロ値には、クエリ（query:key の形式）を含める必要があります。

マクロ値の Zabbix による処理の詳細については、*Vault secret macros*を参照してください。

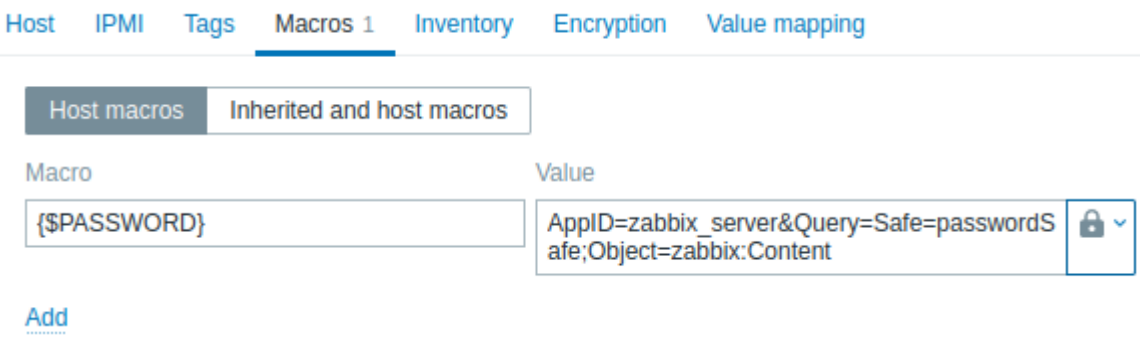
クエリ構文

コロン記号（":"）は、クエリとキーを区切るために予約されています。

クエリ自体にスラッシュまたはコロンが含まれる場合、これらの記号は URL エンコードする必要があります（"/" は"%2F"、":" は"%3A" にエンコードされます）。

例

1. Zabbix で、タイプが *Vault secret*、値が `AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content` のユーザーマクロ {$PASSWORD} を追加します。



2. Zabbix は vault に対して次の API リクエストを送信します。

```
curl \
--header "Content-Type: application/json" \
--cert cert.pem \
--key key.pem \
https://127.0.0.1:1858/AIMWebService/api/Accounts?AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbi
```

3. vault のレスポンスにはキー"Content" が含まれます。

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. その結果、Zabbix はマクロ {$PASSWORD} を値 - <password> に解決します。

既存の設定を更新する

CyberArk Vault からシークレットを取得するための既存の設定を更新するには、次の手順を実行します。

1. データベース認証情報 セクションで説明されているとおりに、Zabbix サーバーまたはプロキシの設定ファイルのパラメータを更新します。

2. *Web インターフェース* セクションで説明されている必要なパラメータを指定して、Zabbix Web インターフェースを再設定し、DB 接続設定を更新します。
   Zabbix Web インターフェースを再設定するには、ブラウザで Web インターフェースのセットアップ URL を開きます。

- Apache の場合: http://<server_ip_or_name>/zabbix/setup.php
- Nginx の場合: http://<server_ip_or_name>/setup.php

または、これらのパラメータはWeb インターフェース設定ファイル (*zabbix.conf.php*) で設定することもできます。

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. 必要に応じて、ユーザーマクロの値 セクションで説明されているとおりにユーザーマクロを設定します。

HashiCorp Vault からシークレットを取得するための既存の設定を更新する方法については、*HashiCorp の設定* を参照してください。

2 HashiCorp の設定

概要

このセクションでは、HashiCorp Vault KV Secrets Engine - Version 2 からシークレットを取得するために Zabbix を設定する方法について説明します。

vault は、公式の HashiCorp ドキュメント に記載されているとおりにデプロイおよび設定する必要があります。

Zabbix で TLS を設定する方法については、シークレットの保存 を参照してください。

データベース認証情報の取得

データベース認証情報を含むシークレットを正常に取得するには、以下の両方を設定する必要があります。

- Zabbix サーバー/プロキシ
- Zabbix Web インターフェース

サーバー/プロキシ

Zabbix サーバー またはプロキシ を設定するには、
設定ファイルで以下の設定パラメータを指定します。

- `Vault` - 使用する vault プロバイダー。
- `VaultToken` - vault 認証トークン（詳細は Zabbix サーバー/プロキシの設定ファイルを参照）。
- `VaultURL` - vault サーバーの HTTP[S] URL。
- `VaultDBPath` - データベース認証情報を含む vault シークレットへのパス（このオプションは、DBUser および DBPassword が指定されていない場合にのみ使用できます）。Zabbix サーバーまたはプロキシは、キー”password” および”username” によって認証情報を取得します。
- `VaultPrefix` - vault に応じた、vault パスまたはクエリ用のカスタムプレフィックス。指定しない場合は、最適なデフォルト値が使用されます。

Attention:
`Vault`、`VaultToken`、`VaultURL`、`VaultPrefix` の各設定パラメータは、Zabbix サーバーがシークレット vault マクロを処理する際の vault 認証にも使用されます（設定されている 場合は Zabbix プロキシでも使用されます）。Zabbix サーバーおよび Zabbix プロキシは、VaultDBPath の DB 認証情報を含む vault シークレットマクロを展開しません。

 プロキシごとに異なるトークンを使用することを強く推奨します。

Zabbix サーバーおよび Zabbix プロキシは、起動時に *zabbix_server.conf* および *zabbix_proxy.conf* から vault 関連の設定パラメータを読み取ります。
さらに、Zabbix サーバーおよび Zabbix プロキシは、起動時に `VAULT_TOKEN` 環境変数も一度だけ読み取り、その後 fork されたスクリプトから参照できないように unset します。`VaultToken` と `VAULT_TOKEN` の両方に値が含まれている場合はエラーになります。

例

1. *zabbix_server.conf* で、以下のパラメータを指定します。

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
```

```
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. vault に必要なシークレットを作成するため、以下の CLI コマンドを実行します。

```
#### まだ有効になっていない場合は "secret/" マウントポイントを有効にします。"kv-v2" を使用する必要があることに注意してくださ
vault secrets enable -path=secret/ kv-v2

#### マウントポイント "secret/" およびパス "zabbix/database" の下に、username と password をキーとする新しいシークレ
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### シークレットが正常に追加されたことをテストします。
vault kv get secret/zabbix/database

#### 最後に Curl でテストします。"data" はマウントポイントの後に手動で追加し、"/v1" はマウントポイントの前に追加する必要があ
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. その結果、Zabbix サーバーはデータベース認証用に以下の認証情報を取得します。

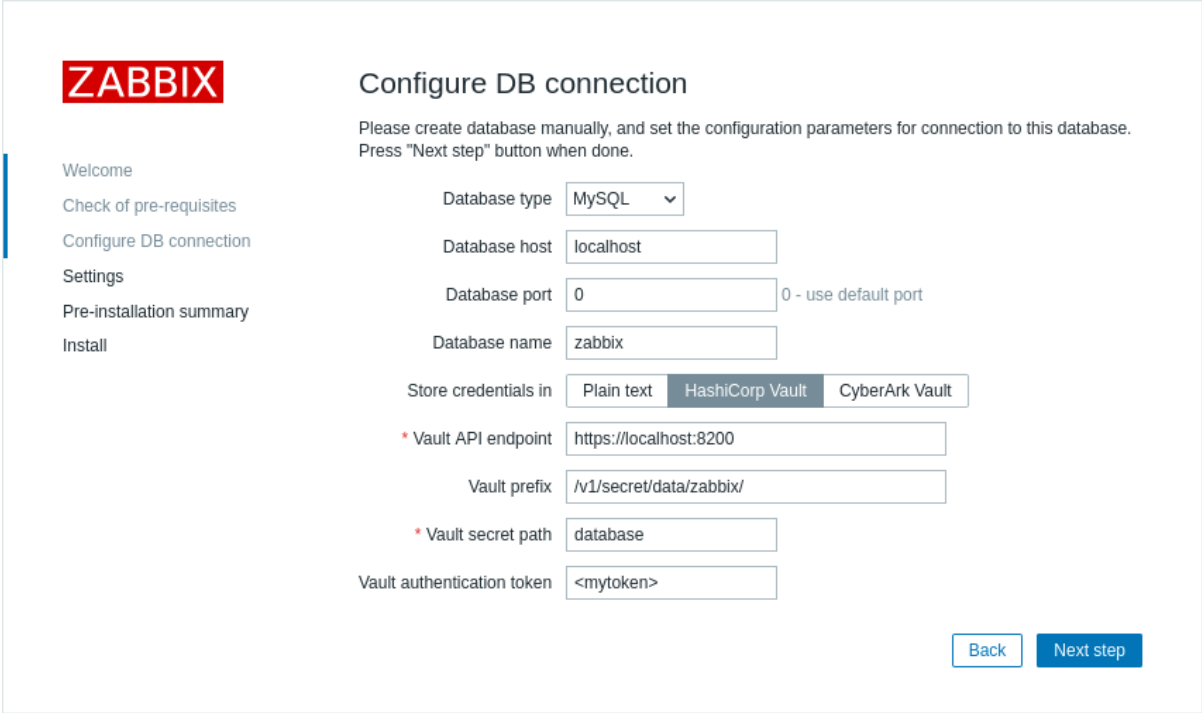
- ユーザー名: zabbix
- パスワード: <password>

Web インターフェース

Zabbix の Web インターフェースは、Web インターフェースのインストール時、または Web インターフェース設定ファイル (`zabbix.conf.php`) を更新することで、vault からデータベース認証情報を取得するよう設定できます。

Attention:
前回の Web インターフェースのインストール以降に vault 認証情報が変更されている場合は、Web インターフェースのインストールを再実行するか、`zabbix.conf.php` を更新してください。あわせて既存設定の更新も参照してください。

Web インターフェースのインストール中は、*Configure DB Connection* ステップで設定パラメータを指定する必要があります。



- *Store credentials in* パラメータを「HashiCorp Vault」に設定します。
- 接続パラメータを指定します。

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:8200	`scheme://host:port` 形式で、vault への接続に使用する URL を指定します

Parameter	Mandatory	Default value	Description
Vault prefix	no	/v1/secret/data/	vault のパスまたはクエリに使用するカスタムプレフィックスを指定します。指定しない場合は、デフォルト値が使用されます。 例: `/v1/secret/data/zabbix/`
Vault secret path	no		データベースの認証情報を取得するシークレットへのパスです。認証情報は、キー「password」および「username」から取得されます。 例: `database`
Vault authentication token	no		シークレットパスへの読み取り専用アクセス用の認証トークンを指定します。 トークンおよび vault ポリシーの作成方法については、HashiCorp documentationを参照してください。

ユーザーマクロ値の取得

Vault secret ユーザーマクロ値の保存に HashiCorp Vault を使用するには、次の点を確認してください。

- Zabbix サーバー/プロキシが、HashiCorp Vault と連携するように設定されていること。
- 管理 → 一般 → その他 の *Vault provider* パラメータが"HashiCorp Vault"（デフォルト）に設定されていること。



Note:
Zabbix サーバー（および、設定されている場合は Zabbix プロキシ）には、vault から *Vault secret* マクロ値へアクセスする権限が必要です。
Zabbix Web インターフェースには、そのようなアクセス権は必要ありません。

マクロ値には参照パス（`path:key` の形式。例: `macros:password`）を含める必要があります。
Zabbix サーバー/プロキシの設定時に指定する認証トークン（`VaultToken` パラメータ）は、このパスに対する読み取り専用アクセス権を持っている必要があります。

マクロ値の Zabbix による処理の詳細については、*Vault secret macros*を参照してください。

パス構文

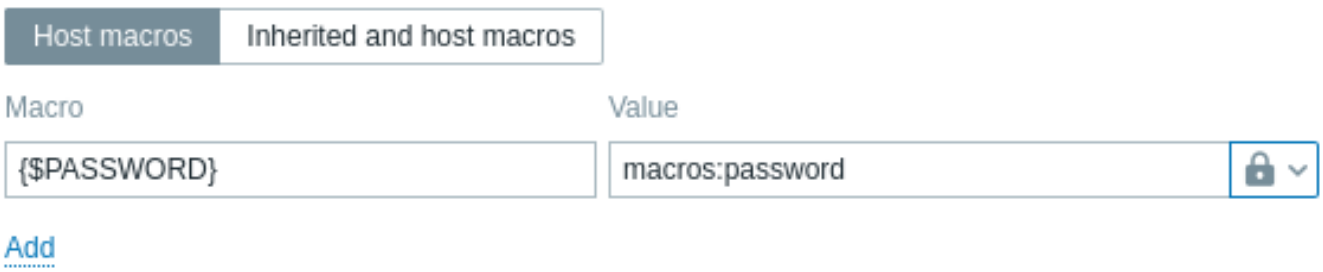
スラッシュ ("/") とコロン (":") は予約済みの記号です。

スラッシュは、マウントポイントとパスを区切るためにのみ使用できます（例: *secret/zabbix*。この場合、マウントポイントは"secret"、パスは"zabbix" です）。Vault マクロの場合、コロンはパス/クエリとキーを区切るためにのみ使用できます。

スラッシュで区切られた名前のマウントポイントを作成する必要がある場合（例: *foo/bar/zabbix*。この場合、マウントポイントは"foo/bar"、パスは"zabbix" であり、"foo%2Fbar/zabbix" としてエンコードできます）や、マウントポイント名またはパスにコロンを含める必要がある場合は、スラッシュおよびコロン記号を URL エンコードできます。

例

1. Zabbix で、タイプが「Vault secret」、値が `macros:password` のユーザーマクロ `{$PASSWORD}` を追加します。



2. vault に必要なシークレットを作成するため、以下の CLI コマンドを実行します。

```
#### まだ有効になっていない場合は "secret/" マウントポイントを有効にします。なお、"kv-v2" を使用する必要があります。
vault secrets enable -path=secret/ kv-v2

#### マウントポイント "secret/"、パス "zabbix/macros" の下に、キー "password" を持つ新しいシークレットを登録します。
vault kv put -mount=secret zabbix/macros password=<password>

#### シークレットが正常に追加されたことをテストします。
vault kv get secret/zabbix/macros

#### 最後に Curl でテストします。"data" はマウントポイントの後に、"/v1" はマウントポイントの前に手動で追加する必要がある点に
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. その結果、Zabbix はマクロ {$PASSWORD} を次の値に解決します: <password>

既存の設定の更新

HashiCorp Vault からシークレットを取得するための既存の設定を更新するには、次の手順を実行します。

1. データベース認証情報 セクションで説明されているように、Zabbix サーバーまたはプロキシの設定ファイルのパラメータを更新します。
2. *Web インターフェース* セクションで説明されているように、Zabbix Web インターフェースを再設定して必要なパラメータを指定し、DB 接続設定を更新します。
   Zabbix Web インターフェースを再設定するには、ブラウザで Web インターフェースのセットアップ URL を開きます。

- Apache の場合: http://<server_ip_or_name>/zabbix/setup.php
- Nginx の場合: http://<server_ip_or_name>/setup.php

または、これらのパラメータはWeb インターフェース設定ファイル (*zabbix.conf.php*) で設定することもできます。

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

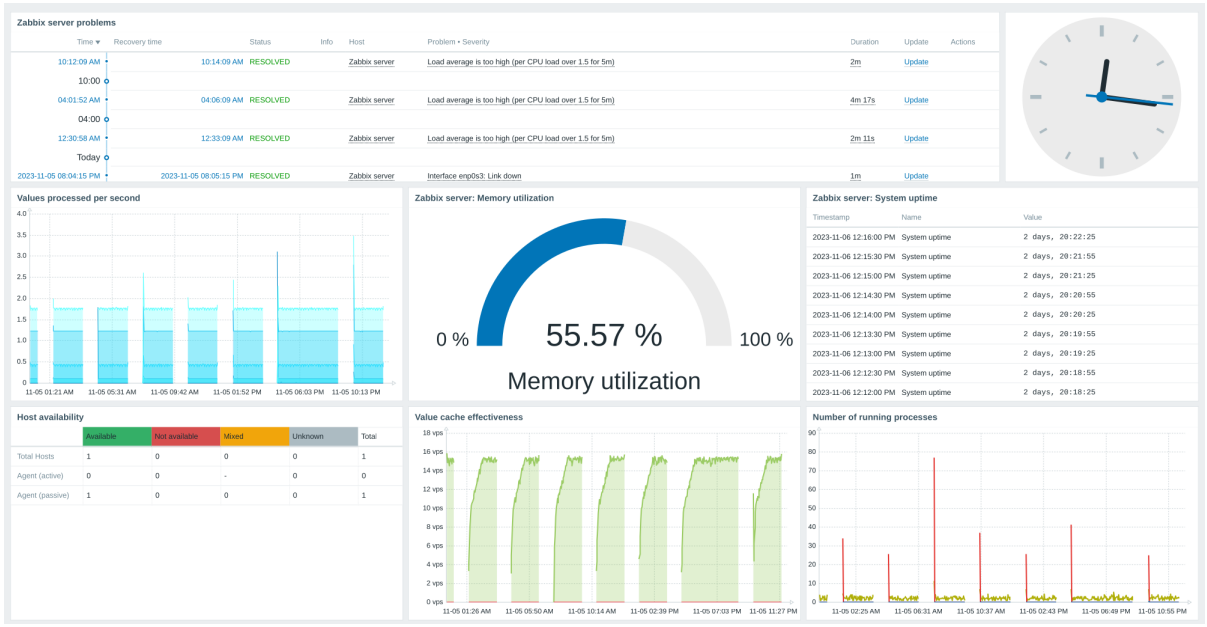
3. 必要に応じて、ユーザーマクロの値 セクションで説明されているように、ユーザーマクロを設定します。

CyberArk Vault からシークレットを取得するための既存の設定を更新するには、*CyberArk の設定* を参照してください。

14 スケジュールレポート

概要

定期レポート機能を使用すると、指定したダッシュボードの PDF 版を、指定した受信者に対して定期的に送信するよう設定できます。



前提条件:

- 定期レポート生成を有効にするには、Zabbix web service がインストールされ、正しく設定されている必要があります。手順については、定期レポートの設定を参照してください。
- ユーザーには、タイプが *Admin* または *Super admin* のユーザーロールが割り当てられており、以下の権限を持っている必要があります:
  - *Access to UI elements* ブロック内の *Scheduled reports*（レポート設定を表示するため）
  - *Access to actions* ブロック内の *Manage scheduled reports*（レポートを作成/編集するため）

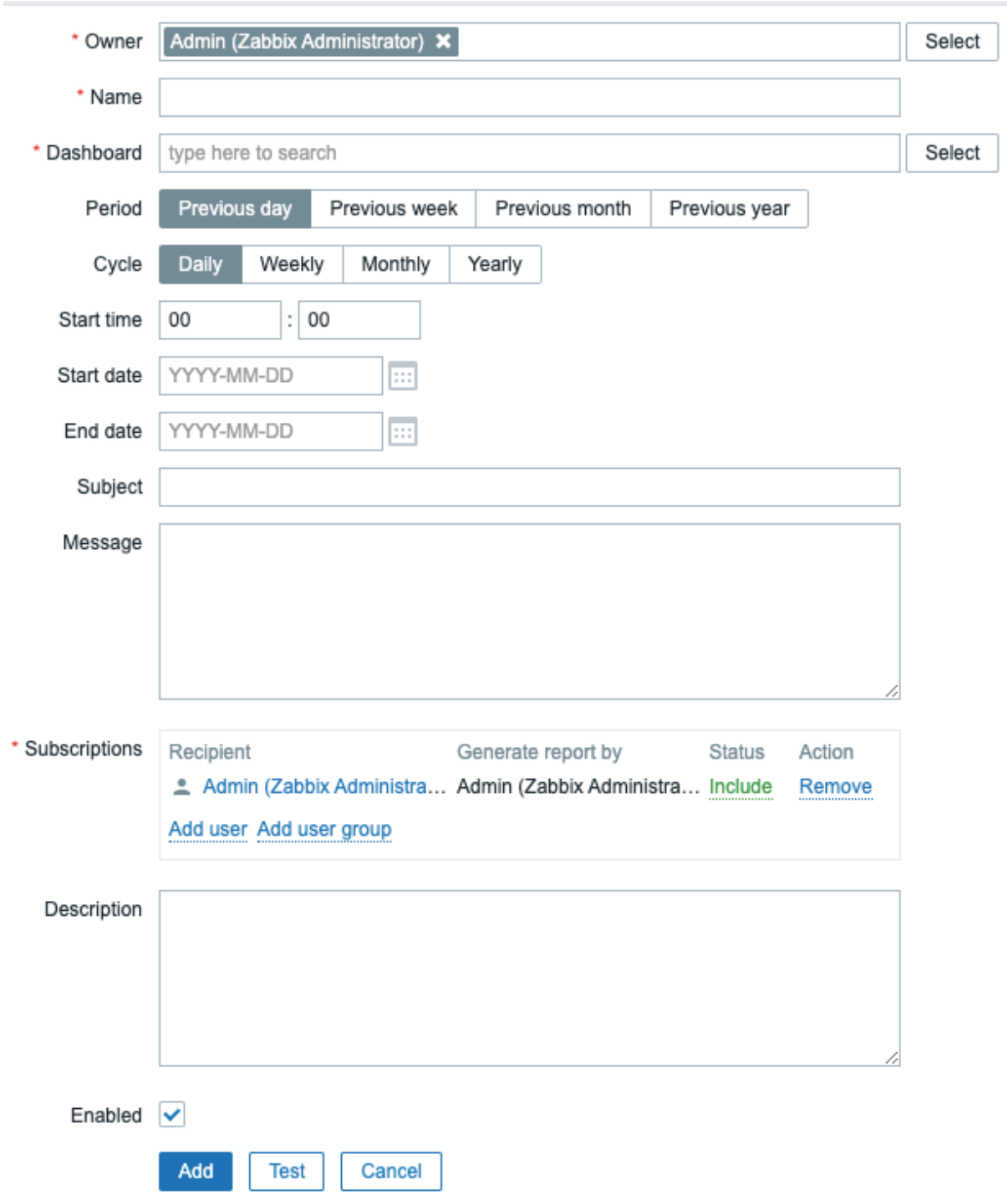
Zabbix Web インターフェースで定期レポートを作成するには、以下を実行します。

- レポート > 定期レポートに移動します。
- 画面右上の レポートの作成をクリックします。
- フォームにレポートのパラメータを入力します。

既存のレポートを開き、複製ボタンをクリックしてから別の名前で保存することでも、レポートを作成できます。

設定

定期レポートタブには、レポートの一般的な属性が含まれています。



必須の入力フィールドには、すべて赤いアスタリスクが付いています。

Parameter	Description
Owner	レポートを作成するユーザーです。*Super admin* レベルのユーザーは所有者を変更できます。*Admin* レベルのユーザーの場合、このフィールドは読み取り専用です。
Name	レポート名です。一意である必要があります。
Dashboard	レポートの基になるダッシュボードです。一度に選択できるダッシュボードは 1 つだけです。ダッシュボードを選択するには、名前の入力を開始すると一致するダッシュボードの一覧が表示されるので、スクロールして選択します。あるいは、フィールドの横にある *Select* をクリックし、表示された一覧からダッシュボードを選択することもできます。
Period	レポートを作成する対象期間です。前日、前週、前月、または前年を選択します。
Cycle	レポート生成の頻度です。レポートは日次、週次、月次、または年次で送信できます。”Weekly” モードでは、レポートを送信する曜日を選択できます。
Start time	レポートを作成する時刻です。形式は hh:mm です。Zabbix サーバーのタイムゾーンが使用される点に注意してください。
Repeat on	レポートを送信する曜日です。このフィールドは、*Cycle* が”Weekly” に設定されている場合にのみ使用できます。

Parameter	Description
Start date	定期的なレポート生成を開始する日付です。
End date	定期的なレポート生成を停止する日付です。
Subject	レポートメールの件名です。サポートされるマクロ: {TIME}, {TIMESTAMP}.
Message	レポートメールの本文です。サポートされるマクロ: {TIME}, {TIMESTAMP}.
Subscriptions	レポート受信者の一覧です。デフォルトでは、レポート所有者のみが含まれます。メールメディアが設定されている任意の Zabbix ユーザーをレポート受信者として指定できます。 *Add user* または *Add user group* をクリックして、受信者を追加します。 設定を編集するには、ユーザー名をクリックします。 *Generate report by* - レポートデータを現在のユーザーのダッシュボード権限に基づいて生成するか、受信者の権限に基づいて生成するかを指定します。 *Status* - ユーザーにレポートを送信するには”Include” を選択し、このユーザーへの送信を防ぐには”Exclude” を選択します。少なくとも 1 人のユーザーが”Include” ステータスである必要があります。”Exclude” ステータスは、含まれているユーザーグループから特定のユーザーを除外するために使用できます。 権限が不十分なユーザー（つまり、受信者またはレポート所有者と同じユーザーグループのメンバーではない、*Admin* ユーザータイプに基づくロールを持つユーザー）は、*Recipient* および *Generate report by* フィールドで実際の名前の代わりに”Inaccessible user” または”Inaccessible user group” と表示されます。*Status* および *Action* フィールドは読み取り専用として表示されます。
Enabled	レポートのステータスです。このチェックボックスをオフにすると、レポートは無効になります。
Description	レポートの任意の説明です。この説明は内部利用のためのものであり、レポート受信者には送信されません。

フォームボタン

フォームの下部にあるボタンでは、いくつかの操作を実行できます。

Add	レポートを追加します。このボタンは新規レポートでのみ使用できます。
Update	レポートのプロパティを更新します。
Clone	現在のレポートのプロパティに基づいて別のレポートを作成します。
Test	現在のユーザーにレポートを送信して、レポート設定が正しいかどうかをテストします。
Delete	レポートを削除します。
Cancel	レポートのプロパティの編集をキャンセルします。

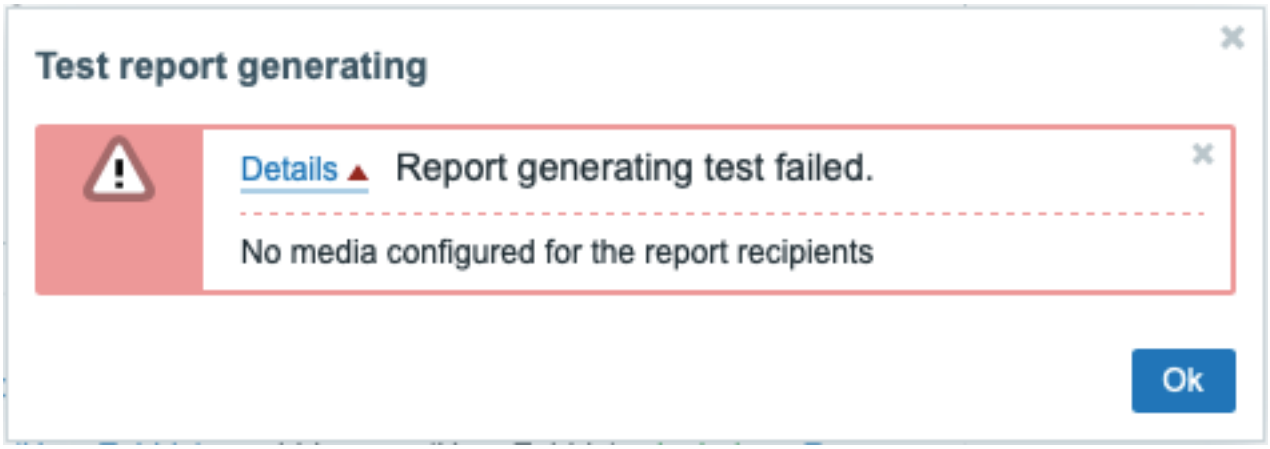
テスト

レポートをテストするには、レポート設定フォームの下部にある テストボタンをクリックします。

Note:
ダッシュボードのaction menuからレポート設定フォームを開いた場合、テストボタンは使用できません。

設定が正しい場合、テストレポートは現在のユーザーに直ちに送信されます。
テストレポートでは、購読者および *Generate report by* ユーザー設定は無視されます。

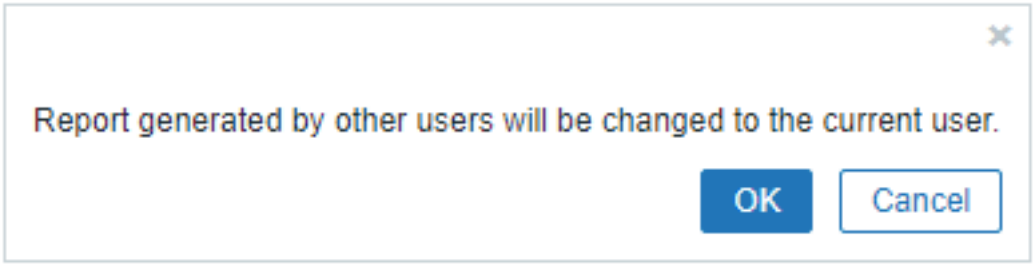
設定が正しくない場合、考えられる原因を示すエラーメッセージが表示されます。



レポートの更新

既存のレポートを更新するには、レポート名をクリックし、必要な設定変更を行ってから、*Update* ボタンをクリックします。

既存のレポートが別のユーザーによって更新され、このユーザーがダッシュボードを変更した場合、*Update* ボタンをクリックすると、「Report generated by other users will be changed to the current user」という警告メッセージが表示されます。



この段階で *OK* をクリックすると、次の変更が行われます。

- *Generate report by* 設定が更新され、最後にレポートを編集したユーザーが表示されるようになります（*Generate report by* が受信者に設定されている場合を除く）。
- 「Inaccessible user」または「Inaccessible user group」と表示されているユーザーは、レポート購読者の一覧から削除されます。

Cancel をクリックすると、設定フォームが閉じられ、レポートの更新はキャンセルされます。

レポートの複製

既存のレポートをすばやく複製するには、既存のレポート設定フォームの下部にある *Clone* ボタンをクリックします。
他のユーザーが作成したレポートを複製する場合、現在のユーザーが新しいレポートの所有者になります。

ユーザー権限に応じて、レポート設定は新しいレポート設定フォームにコピーされます。

- レポートを複製するユーザーにダッシュボードへの権限がない場合、*Dashboard* フィールドはクリアされます。
- レポートを複製するユーザーに *Subscriptions* リスト内の一部のユーザーまたはユーザーグループへの権限がない場合、アクセスできない受信者は複製されません。
- *Generate report by* の設定は、現在のユーザーが表示されるように更新されます（*Generate report by* が受信者に設定されている場合を除く）。

必要な設定とレポート名を変更し、*Add* をクリックします。

15 データのエクスポート

概要

Zabbix は、次の 2 つの方法でリアルタイムでのデータエクスポートをサポートします。

- ファイルにエクスポート
- 外部システムへのストリーミング

次のエンティティをエクスポートできます。

- トリガーイベント
- アイテムの値
- トレンド (ファイルにエクスポートのみ)

1 ファイルにエクスポート

概要

トリガーイベント、アイテムの値、およびトレンドを、改行区切りの JSON 形式でリアルタイムにエクスポートするよう設定できます。

エクスポートはファイルに対して行われ、エクスポートファイルの各行は JSON オブジェクトになります。値のマッピングは適用されません。

エラーが発生した場合（データをエクスポートファイルに書き込めない、またはエクスポートファイルの名前を変更できない、あるいは名前変更後に新しいファイルを作成できない場合）、そのデータアイテムは破棄され、エクスポートファイルには書き込まれません。データは Zabbix データベースにのみ書き込まれます。書き込みの問題が解消されると、エクスポートファイルへのデータ書き込みが再開されます。

Zabbix 7.4.6 以降、エクスポートファイルはファイル所有者のみに読み書き権限を付与して作成されます。さらに、そのファイルは所有グループによる読み取りが可能です。その他の権限はすべて拒否されます。

エクスポートされる情報の正確な詳細については、エクスポートプロトコルページを参照してください。

データ受信後、サーバーがデータをエクスポートする前にホスト/アイテムが削除された場合、ホスト/アイテムにはメタデータ（ホストグループ、ホスト名、アイテム名）が存在しないことがある点に注意してください。

設定

トリガーイベント、アイテム値、およびトレンドのリアルタイムエクスポートは、エクスポートファイル用のディレクトリを指定して設定します。詳細は、サーバー設定のExportDirパラメータを参照してください。

利用可能なパラメータは、ほかに 2 つあります。

- ExportFileSize は、個々のエクスポートファイルに許可される最大サイズの設定に使用できます。プロセスがファイルに書き込む必要がある場合、まずそのファイルのサイズを確認します。設定されたサイズ上限を超えている場合、ファイル名の末尾に.old を付加してファイル名を変更し、元の名前で新しいファイルを作成します。

> **Attention:**
> データを書き込む各プロセスごとに 1 つのファイルが作成されます（つまり、およそ 4～30 ファイルです）。エクスポートファイルのデフォルトサイズは 1G であるため、大きなエクスポートファイルを保持していると、ディスク容量を急速に消費する可能性があります。

- ExportType では、どのエンティティタイプ（events、history、trends）をエクスポートするかを指定できます。

2 外部システムへのストリーミング

概要

HTTP 経由で、Zabbix から外部システムへアイテムの値およびイベントをストリーミングできます（プロトコルの詳細を参照）。

タグフィルターを使用して、アイテムの値またはイベントのサブセットをストリーミングできます。

データストリーミングを担当する Zabbix サーバーのプロセスは、`connector manager` と `connector worker` の 2 つです。Zabbix の内部アイテム `zabbix[connector_queue]` を使用すると、コネクターキューに追加された値の件数を監視できます。

設定

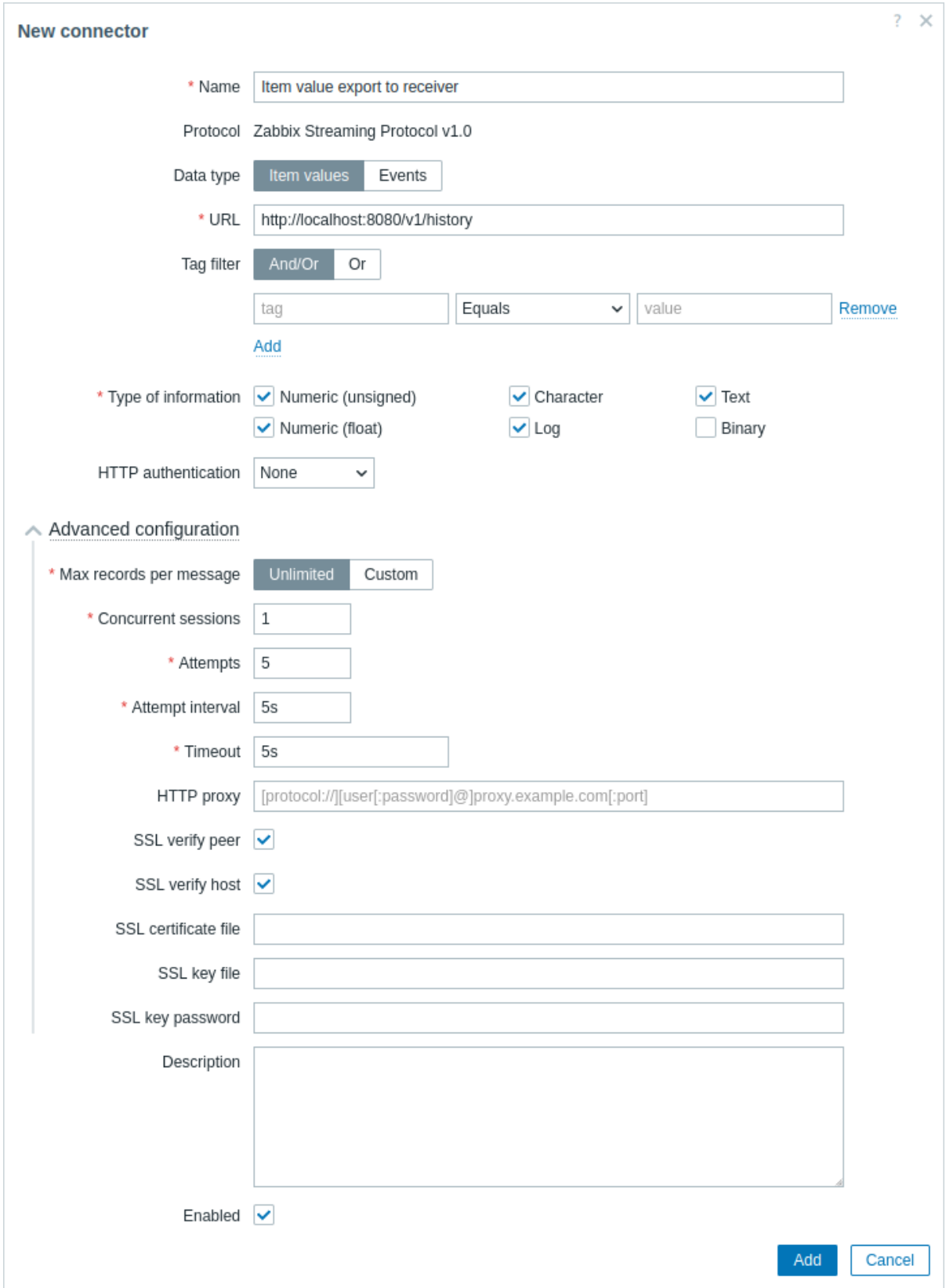
外部システムへのデータストリーミングを設定するには、以下の手順が必要です。

1. Zabbix からのデータを受信するためのリモートシステムを準備します。
この目的のために、以下のツールを利用できます。

- 受信した情報を `events.ndjson` および `history.ndjson` ファイルに記録する、シンプルな受信側の例。
- Zabbix server 用 Kafka connector - Go で記述された軽量サーバーで、Zabbix サーバーから Kafka ブローカーへアイテムの値およびイベントを転送するよう設計されています。

2. `zabbix_server.conf` のStartConnectors パラメータを調整して、Zabbix で必要な数のコネクタワーカーを設定します。
コネクタワーカー数は、Zabbix Web インターフェースで設定されたコネクタ数と一致させる必要があります（同時セッション数が 1 を超える場合は、それ以上にする必要があります）。
その後、Zabbix サーバーを再起動します。

3. Zabbix Web インターフェースで新しいコネクタを設定し（*Administration* > *General* > *Connectors*）、`zabbix_server -R config_cache_reload` コマンドでサーバーキャッシュを再読み込みします。



必須フィールドにはアスタリスクが付いています。

Parameter	Description
Name	コネクタ名を入力します。
Data type	ストリーミングするデータタイプを選択します。 **Item values** - Zabbix から外部システムへアイテムの値をストリーミングします。 **Events** - Zabbix から外部システムへイベントをストリーミングします。
URL	受信側の URL を入力します。ユーザーマクロを使用できます。

Parameter	Description
Tag filter	タグフィルターに一致するアイテムの値またはイベントのみをエクスポートします。設定しない場合は、すべてをエクスポートします。 特定のタグおよびタグ値を含めることも除外することも可能です。複数の条件を設定できます。タグ名の一致は常に大文字・小文字を区別します。 各条件には複数の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名および値を含めます（大文字・小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字・小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名および値を除外します（大文字・小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字・小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
Type of information	コネクタがストリーミングするアイテムの値をフィルタリングするための情報タイプ（numeric (unsigned)、numeric (float)、character など）を選択します。 このフィールドは、*Data type* が”Item values” に設定されている場合に使用できます。
HTTP authentication	認証オプションを選択します。 **None** - 認証は使用しません。 **Basic** - Basic 認証を使用します。 **NTLM** - NTLM（Windows NT LAN Manager）認証を使用します。 **Kerberos** - Kerberos 認証を使用します（あわせて参照: Configuring Kerberos with Zabbix）。 **Digest** - Digest 認証を使用します。 **Bearer** - Bearer 認証を使用します。
Username	ユーザー名を入力します（最大 255 文字）。ユーザーマクロを使用できます。 このフィールドは、*HTTP authentication* が”Basic”、”NTLM”、”Kerberos”、または”Digest” に設定されている場合に使用できます。
Password	ユーザーパスワードを入力します（最大 255 文字）。ユーザーマクロを使用できます。 このフィールドは、*HTTP authentication* が”Basic”、”NTLM”、”Kerberos”、または”Digest” に設定されている場合に使用できます。
Bearer token	Bearer トークンを入力します。ユーザーマクロを使用できます。 このフィールドは、*HTTP authentication* が”Bearer” に設定されている場合に使用可能かつ必須です。
Advanced configuration	*Advanced configuration* ラベルをクリックすると、高度な設定オプションが表示されます（以下を参照）。
Max records per message	1 つのメッセージ内でストリーミングできる値またはイベントの最大数を指定します。
Concurrent sessions	このコネクタで実行する送信プロセス数を選択します。最大 100 セッションまで指定でき、デフォルト値は”1” です。
Attempts	データストリーミングの試行回数です。最大 5 回まで指定でき、デフォルト値は”1” です。
Attempt interval	データストリーミングの試行に失敗した後、コネクタが待機する時間を指定します。最大 10s まで指定でき、デフォルト値は”5s” です。 このフィールドは、*Attempts* が”2” 以上に設定されている場合に使用できます。 失敗した試行とは、接続の確立に失敗した場合、または HTTP レスポンスコードが 200、201、202、203、204 でない場合です。再試行は、通信エラーが発生した場合、または HTTP レスポンスコードが 200、201、202、203、204、400、401、403、404、405、415、422 でない場合にトリガーされます。リダイレクトは追跡されるため、302 -> 200 は正常なレスポンスですが、302 -> 503 は再試行を引き起こします。
Timeout	メッセージのタイムアウトを指定します（1～60 秒、デフォルトは 5 秒）。 時間サフィックスを使用できます。例: 30s、1m。ユーザーマクロを使用できます。

Parameter	Description
HTTP proxy	以下の形式で使用する HTTP プロキシを指定できます。 `[protocol://][username[:password]@]proxy.example.com[:port]` ユーザーマクロを使用できます。 オプションの `protocol://` プレフィックスを使用して、代替のプロキシプロトコルを指定できます（プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました）。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。デフォルトでは 1080 ポートが使用されます。 *HTTP proxy* が指定されている場合、`http_proxy`、`HTTPS_PROXY` などのプロキシ関連環境変数よりもこのプロキシ設定が優先されます。指定されていない場合、プロキシはプロキシ関連環境変数を上書きしません。入力された値はそのまま渡され、妥当性チェックは行われません。 SOCKS プロキシアドレスを入力することもできます。誤ったプロトコルを指定すると、接続は失敗し、アイテムは未サポートになります。 HTTP プロキシでは単純認証のみサポートされることに注意してください。
SSL verify peer	Web サーバーの SSL 証明書を検証するには、チェックボックスをオンにします。 サーバー証明書は、システム全体の認証局（CA）の場所から自動的に取得されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータ`SSLCALocation` を使用して上書きできます。
SSL verify host	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスをオンにします。 これにより、`CURLOPT_SSL_VERIFYHOST` cURL オプションが設定されます。
SSL certificate file	クライアント認証に使用する SSL 証明書ファイル名です。証明書ファイルは PEM[1] 形式である必要があります。ユーザーマクロを使用できます。 証明書ファイルに秘密鍵も含まれている場合は、*SSL key file* フィールドを空のままにしてください。鍵が暗号化されている場合は、*SSL key password* フィールドにパスワードを指定してください。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ`SSLCertLocation` で指定します。
SSL key file	クライアント認証に使用する SSL 秘密鍵ファイル名です。秘密鍵ファイルは PEM[1] 形式である必要があります。ユーザーマクロを使用できます。 このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータ`SSLKeyLocation` で指定します。
SSL key password	SSL 秘密鍵ファイルのパスワードです。ユーザーマクロを使用できます。
Description	コネクタの説明を入力します。
Enabled	チェックボックスをオンにしてコネクタを有効にします。

Kafka コネクタがブートストラップブローカーアドレスのカンマ区切りリストで設定されている場合(例: `Kafka.URL=kafka1.example.com:`
Kafka クライアントは最初に応答したブローカーに接続し、そのクラスターメタデータを使用します。
リストに異なる Kafka クラスターのアドレスが含まれている場合、最も速く応答したクラスターのみが使用され、他のアドレスは利用不可としてログに記録されます。その結果、コネクタが接続されていても、以下のような起動時警告が表示されることがあります。

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

一部の環境（プライベートネットワーク、コンテナネットワーク、または標準的でない DNS/hosts 設定）では、ホスト名または IP がループバックアドレス（例: 127.0.0.1/localhost）に解決されたり、クライアントによって正規化されたりすることがあり、その結果、このような警告が誤解を招く場合があります。
混乱を減らすには、すべての `Kafka.URL` アドレスが同じ Kafka クラスターに属していることを確認し、コネクタホストからの DNS 名前解決とブローカーの `advertised.listeners` を確認し、ブローカーの通知アドレスに解決されるアドレスを優先して使用してください。

プロトコル

サーバーと受信側の通信は、REST API、NDJSON、`Content-Type: application/x-ndjson` を使用した HTTP 経由で行われます。

詳細については、改行区切り JSON エクスポートプロトコル を参照してください。

サーバーリクエスト

アイテム値のストリーミングの例:

```
POST /v1/history HTTP/1.1
Host: localhost:8080
Accept: */*
```

```
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 628
Content-Type: application/x-ndjson

{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"f
{"host":{"host":"Zabbix server","name":"Zabbix server"},"groups":["Zabbix servers"],"item_tags":[{"tag":"b
```

イベントのストリーミングの例:

```
POST /v1/events HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Encoding: deflate, gzip, br, zstd
Content-Length: 333
Content-Type: application/x-ndjson

{"clock":1673454303,"ns":800155804,"value":1,"eventid":5,"name":"trigger for foo being 0","severity":0,"ho
{"clock":1673454303,"ns":832290669,"value":0,"eventid":6,"p_eventid":5}
```

Receiver のレスポンス

レスポンスは、HTTP レスポンスのステータスコードと JSON ペイロードで構成されます。正常に処理されたリクエストでは、HTTP レスポンスのステータスコードは”200”、”201”、”202”、”203”、または”204” である必要があります。その他のコードは、リクエストの失敗を示します。

成功時の例:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

エラーがある場合の例:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 SNMP ゲートウェイ

概要

Zabbix SNMP gateway は、SNMP ポーリングとトラップの両方をサポートする、snmpd 向けの AgentX 拡張です。

Zabbix SNMP gateway を使用すると、SNMP プロトコルを使って以下を取得できます。

- トリガーデータ
- 障害トリガーデータ
- ホストグループのステータス（グループごとのトリガーステータス別トリガー数）

データは OID によって取得されます。OID は、共通のベースと固有のサフィックスを組み合わせたものです。共通のベースは SNMP gateway の設定ファイルで設定します。例えば、次のようになります。

- BaseOID=1.3.6.1.4.1.3043.7.55 - 任意のトリガーデータ用
- ProblemBaseOID=1.3.6.1.4.1.3047.7.55 - 障害トリガーデータ用
- BaseOID=1.3.6.1.4.1.3046.7.55 - ホストグループのステータス用

OID のサフィックスは、Web インターフェースでホストのトリガー設定にタグとして設定します（例: OIDSuffix:3）。

この場合、トリガーに関するすべての情報は OID=1.3.6.1.4.1.3043.7.55.X.3 で利用可能になります。ここでの”X” はトリガーデータフィールドの番号です（つまり、1 - サフィックス、2 - ID、3 - 条件式、4 - 説明、など）。

より詳細な説明と設定ファイルの例については、SNMP gateway readme fileを参照してください。

インストールとセットアップ

以下の手順については、SNMP ゲートウェイのreadmeファイルを参照してください。

- snmpd のインストールと設定
- AgentX サポートの有効化
- Zabbix SNMP ゲートウェイの設定
- トリガー状態の変更に対する SNMP トラップの設定

データの取得

すべてが正しく設定されていれば、snmpwalk および snmpget コマンドを使用してデータを取得できます。

```
[user@localhost ~]# snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55.2.3
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
```

```
[user@localhost ~]# snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.3043.7.55
SNMPv2-SMI::enterprises.3043.7.55.1.1 = INTEGER: 1
SNMPv2-SMI::enterprises.3043.7.55.1.3 = INTEGER: 3
SNMPv2-SMI::enterprises.3043.7.55.1.4 = INTEGER: 4
SNMPv2-SMI::enterprises.3043.7.55.1.5 = INTEGER: 5
SNMPv2-SMI::enterprises.3043.7.55.1.6 = INTEGER: 6
SNMPv2-SMI::enterprises.3043.7.55.1.10 = INTEGER: 10
SNMPv2-SMI::enterprises.3043.7.55.2.1 = INTEGER: 15367
SNMPv2-SMI::enterprises.3043.7.55.2.3 = INTEGER: 15247
SNMPv2-SMI::enterprises.3043.7.55.2.4 = INTEGER: 15365
SNMPv2-SMI::enterprises.3043.7.55.2.5 = INTEGER: 15366
SNMPv2-SMI::enterprises.3043.7.55.2.6 = INTEGER: 13493
SNMPv2-SMI::enterprises.3043.7.55.2.10 = INTEGER: 13503
...
```

フィルタリングオプション

SNMP ゲートウェイ設定では、問題のトリガー情報を次の条件で制限できます。

- 深刻度別（デフォルトでは ProblemMinSeverity=-1）
- 確認済みの問題を非表示にする（デフォルトでは ProblemHideAck=false）

SNMP ゲートウェイ設定では、ホストグループごとの問題数を次の条件で制限できます。

- 不明状態のトリガー別（デフォルトでは CountUnknown=false）
- 確認済み／未確認／すべての問題を持つトリガー別（デフォルトでは CountAcknowledgeStatus=all）

8. サービス監視

概要　サービス監視は、ビジネスレベルの監視機能であり、IT インフラストラクチャ全体のサービスツリーの概要を把握し、インフラストラクチャの弱点を特定し、さまざまな IT サービスの SLA を計算し、その他の情報をより高いレベルで確認するために使用できます。サービス監視は、ディスク容量不足やプロセッサー負荷の上昇などの低レベルの詳細ではなく、サービス全体の可用性に重点を置いています。サービス監視はまた、サービスが期待どおりに動作していない場合に、問題の根本原因を特定するための機能も提供します。

サービス監視では、監視データの階層表現を作成できます。

非常に単純なサービス構造の例を以下に示します。

```
サービス
|
|-ワークステーション
| |
| |-ワークステーション1
| |
| |-ワークステーション2
|
|-サーバー
```

構造内の各ノードにはステータス属性があります。ステータスは、選択したアルゴリズムに従って計算され、上位レベルへ伝播されます。個々のノードのステータスは、マッピングされた障害のステータスの影響を受けます。障害のマッピングは、タグ付けによって行われます。

Zabbix は、サービスステータスの変更が検出された場合に、通知を送信したり、Zabbix サーバー上でスクリプトを自動実行したりできます。子サービスのステータスに基づいて、親サービスを「障害状態」にするかどうかについて、柔軟なルールを定義できます。サービスの障害データは、その後、SLA の計算や、柔軟な条件セットに基づく SLA レポートの送信に使用できます。

サービス監視は「サービス」メニューで設定します。このメニューは、以下のセクションで構成されています。

- サービス

サービスセクションでは、親サービスを追加し、次にその親サービスに子サービスを追加することで、監視対象インフラストラクチャの階層を構築できます。

サービスツリーの設定に加えて、このセクションではインフラストラクチャ全体の概要を提供し、サービスステータスの変更につながった問題をすばやく特定できます。

- SLA

このセクションでは、サービスレベル契約を定義し、特定のサービスに対してサービスレベル目標を設定できます。

- SLA レポート

このセクションでは、SLA レポートを表示できます。

サービスアクション

サービスアクションを設定することもできます。

サービスアクションは任意であり、以下を実行できます。

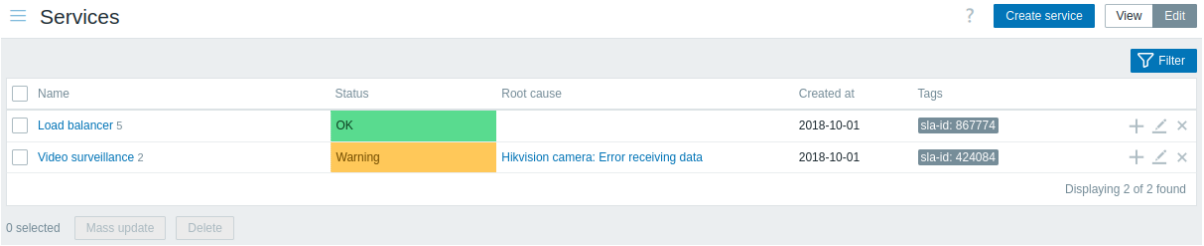
- サービスが停止したことを通知する
- サービスステータスの変更時に Zabbix サーバー上でリモートコマンドを実行する
- サービスが再び稼働したときに復旧通知を送信する

関連情報**:**

- SLA 監視設定の例
- 6.0 未満の Zabbix バージョンからのサービスのアップグレードに関する注意事項

1 サービスツリー

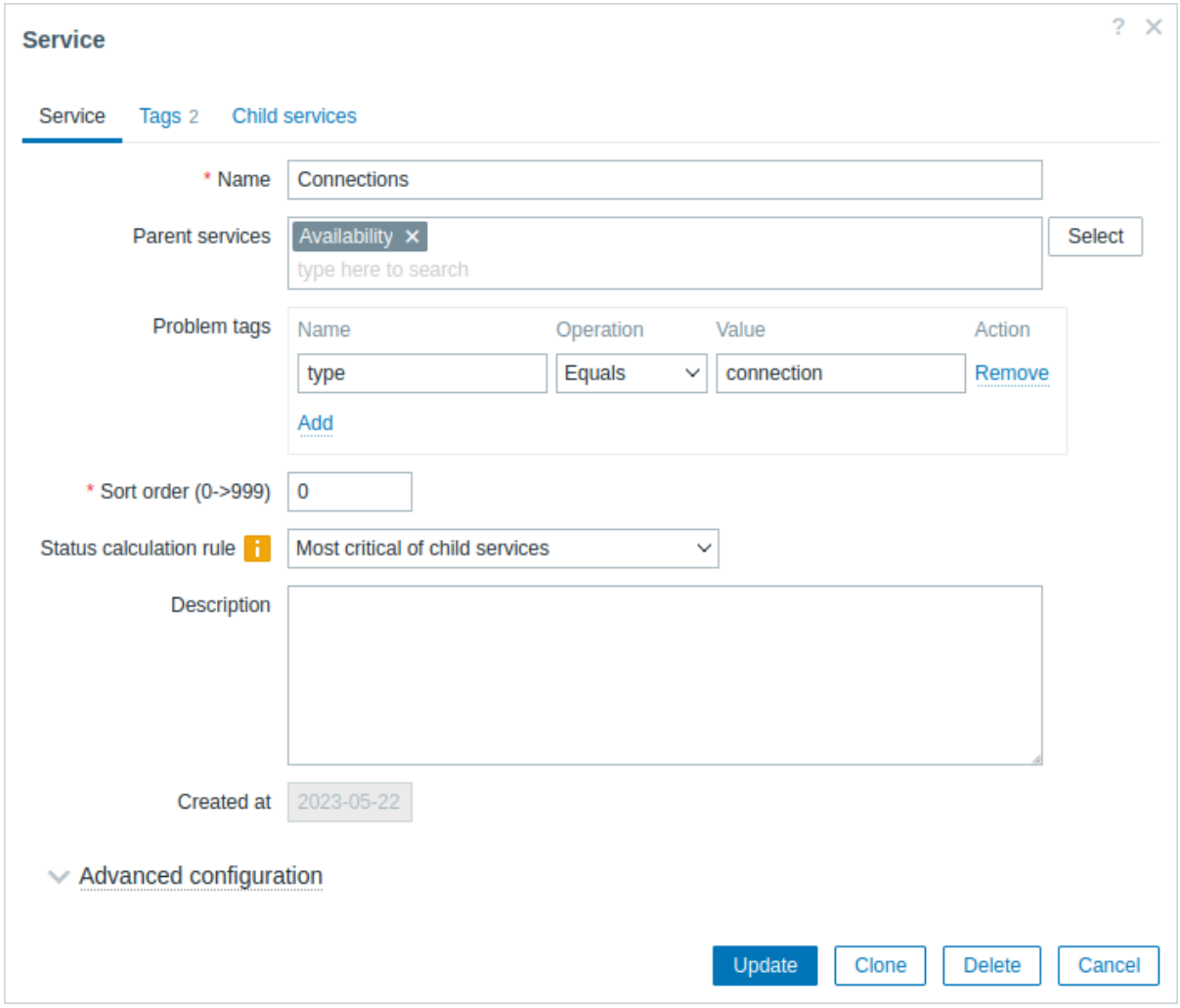
サービスツリーは、*Services -> Services* メニューセクションで設定します。右上隅で、View から編集モードに切り替えます。



新しいサービスを設定するには、右上隅の *Create service* ボタンをクリックします。

子サービスをすばやく追加するには、親サービスの横にあるプラスアイコンを押すこともできます。
これにより同じサービス設定フォームが開きますが、*Parent services* パラメータはあらかじめ入力された状態になります。

サービスの設定　**Service** タブで必要なサービスパラメータを設定します。

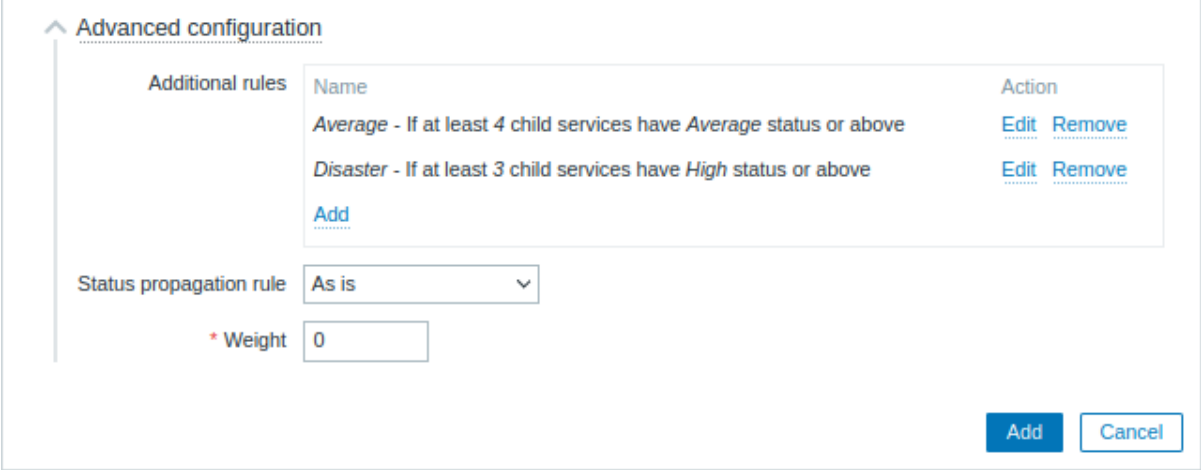


すべての必須入力フィールドは、赤いアスタリスクでマークされています。

パラメーター	説明
名前	サービス名
親サービス	サービスが属する親サービス。 一番レベルの高いサービスを追加する場合は、このフィールドを空のままにします。 1 つのサービスに複数の親サービスがある場合があります。その場合は各親サービスのサービスツリーに表示されます。
障害タグ	障害をサービスにマップするタグを指定します： 等しい - 指定されたタグ名が値と一致するとき対象とします（大文字と小文字を区別します） 含む - 指定されたタグ名が値に入力された文字列が含まれる時に対象とします（部分文字列の一致、大文字と小文字は区別されます） タグ名の一致では常に大文字と小文字が区別されます。
ソート順	表示の並べ替え順序。最も低いものが最初になります。
ステータス計算ルール	サービスステータスの計算ルール すべての子が障害であったときにもっとも重要（Zabbix バージョン 6.0 以前の”Problem, if all children have problems” と同じ）-すべての子に問題がある場合、子サービスの最も深刻な障害の色でサービスステータスが表示されます 子サービスで最も重要（Zabbix バージョン 6.0 以前の”Problem, if one child has a problem” と同じ）-子ノードの最も深刻な障害の色でサービスステータスが表示されます ステータスを正常に設定-サービスステータスを計算しません 下の高度な設定チェックボックスをオンにして、追加のステータス計算ルールを設定します。
説明	サービスの説明

パラメーター	説明
高度な設定	高度な設定オプションにアクセスするには、チェックボックスをオンにします。

詳細設定



パラメータ	説明
Additional rules	追加のステータス計算ルールを定義するには *Add* をクリックします。
Set status to	条件が一致する場合は、サービスステータスを *OK*（default）、未分類、情報、警告、軽度の障害、重度の障害、致命的な障害のいずれかに設定します。
Condition	直接の子サービスの条件を選択します。 少なくとも **(N)** 個の子サービスが **(Status)** ステータス以上の場合 ** 少なくとも (N%) の子サービスが (Status) ステータス以上の場合 ** **(N) 個未満の子サービスが (Status) ステータス以下の場合 ** **(ステータス) 以下の子サービスが (N%) 未満の場合 ** **(Status) ステータス以上の子サービスの重みが (W) 以上の場合 ** ** (Status) ステータス以上の子サービスの重みが (N%) 以上の場合 * * **(Status)** ステータス以下の子サービスの重みが **(W)** 未満の場合 **(Status)** ステータス以下の子サービスの重みが **(N%)** 未満の場合 複数の条件が指定され、状況が複数の条件に一致する場合、最も高い重大度が設定されます。
N (W)	条件で N または W（1-100000）、または N％(1-100）の値を設定します。
Status	条件となるステータスの値を選択します：*OK*（default）、未分類、情報、警告、軽度の障害、重度の障害、致命的な障害
Status propagation rule	サービスステータスを親サービスに伝達するためのルール: 現状のまま - ステータスは変更されずに伝達されます。 増加 - 伝播されたステータスの深刻度を 1 ～5 増加させて伝達されます 減少 - 伝播されたステータスの深刻度を 1 ～5 現象させて伝達されます このサービスを無視 - ステータスは親サービスにまったく伝達されません 固定ステータス - ステータスは静的に伝達されます。つまり、常に同じステータスです。
Weight	サービスの重み（0（デフォルト）から 1000000 の範囲の整数）

Note:
追加のステータス計算ルールは、メインの *Status calculation rule* パラメーターに従って計算されたレベルよりも重大度レベルを上げるためにのみ使用できます。Additional rules では警告だが、*Status calculation rule* では致命的な障害である場合、サービスのステータスは致命的な障害になります。

Tags タブにはservice-level tagsが含まれています。サービスレベルのタグは、サービスを識別するために使用されます。このタイプのタグは問題をサービスにマップするためには使用されません。（そのためには、最初のタブから *Problem tags(#problem-tags)* を使用します）

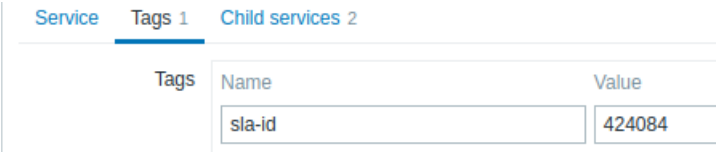
Child services タブでは、依存サービスを指定できます。*Add* をクリックして、既存のサービスのリストからサービスを追加します。新しい子サービスを追加する場合は、最初にこのサービスを保存してから、作成したサービスの横にあるプラスアイコンをクリックします。

タグ　サービスには、2 種類の異なるタグがあります。

- サービスタグ
- 障害タグ

サービスタグ

サービスタグは、サービスをサービスアクションおよびSLAに一致させるために使用されます。これらのタグは、サービス設定のタグタブで指定します。
SLA のマッピングには、*OR* ロジックが使用されます。サービスは、一致するタグを少なくとも 1 つ持っている場合、SLA にマッピングされます。
サービスアクションでは、マッピングルールを設定でき、*AND*、*OR*、または *AND/OR* ロジックのいずれかを使用できます。

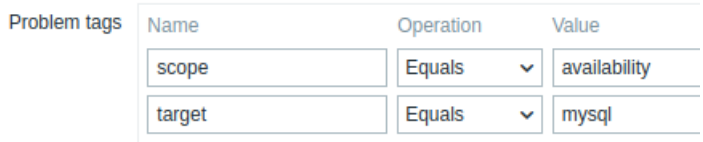


障害タグ

障害タグは、障害とサービスを照合するために使用されます。これらのタグは、プライマリサービスの設定タブで指定します。

障害タグを定義して障害と直接関連付けできるのは、最下位の階層レベルにある子サービスのみです。障害タグが一致すると、サービスのステータスはその障害と同じステータスに変わります。複数の障害がある場合、サービスはその中で最も深刻な障害のステータスになります。親サービスのステータスは、その後、ステータス計算ルールに従って子サービスのステータスに基づいて計算されます。

複数のタグが指定されている場合は、*AND* ロジックが使用されます。サービスにマッピングされるには、障害がサービス設定で指定されたすべてのタグを持っている必要があります。



Note:
Zabbix の障害は、テンプレート、ホスト、アイテム、Web シナリオ、トリガーのチェーン全体からタグを継承します。これらのタグはいずれも、障害をサービスに照合するために使用できます。

例:

障害 *Web camera 3 is down* には、タグ `type:video-surveillance`、`floor:1`、`name:webcam-3` があり、ステータスは *Warning* です。

サービス **Web camera 3** には、唯一の障害タグとして `name:webcam-3` が指定されています。



この障害が検出されると、サービスのステータスは *OK* から *Warning* に変わります。

サービス **Web camera 3** に障害タグ `name:webcam-3` と `floor:2` があった場合、この障害が検出されても条件が部分的にしか満たされないため、ステータスは変更されません。

既存のサービスの変更

Note:
以下で説明するボタンは、サービスセクションが編集モードの場合にのみ表示されます。

既存のサービスを編集するには、サービスの横にある鉛筆アイコンを押します。

既存のサービスを複製するには、鉛筆アイコンを押してその設定を開き、次に **Clone** ボタンを押します。サービスを複製すると、親リンクは保持されますが、子リンクは保持されません。

サービスを削除するには、その横にある x アイコンを押します。親サービスを削除しても、その子サービスは削除されず、サービツリー内で 1 つ上のレベルに移動します（第 1 レベルの子は、削除された親サービスと同じレベルになります）。

サービス一覧の下にある 2 つのボタンでは、一括編集オプションを利用できます。

- *Mass update* - サービスのプロパティを一括更新します

- *Delete* - サービスを削除します

これらのオプションを使用するには、対象のサービスの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

2 SLA

概要　サービスを作成すると、サービスレベル契約（SLA）に照らして、そのパフォーマンスが目標どおりであるかの監視を開始できます。

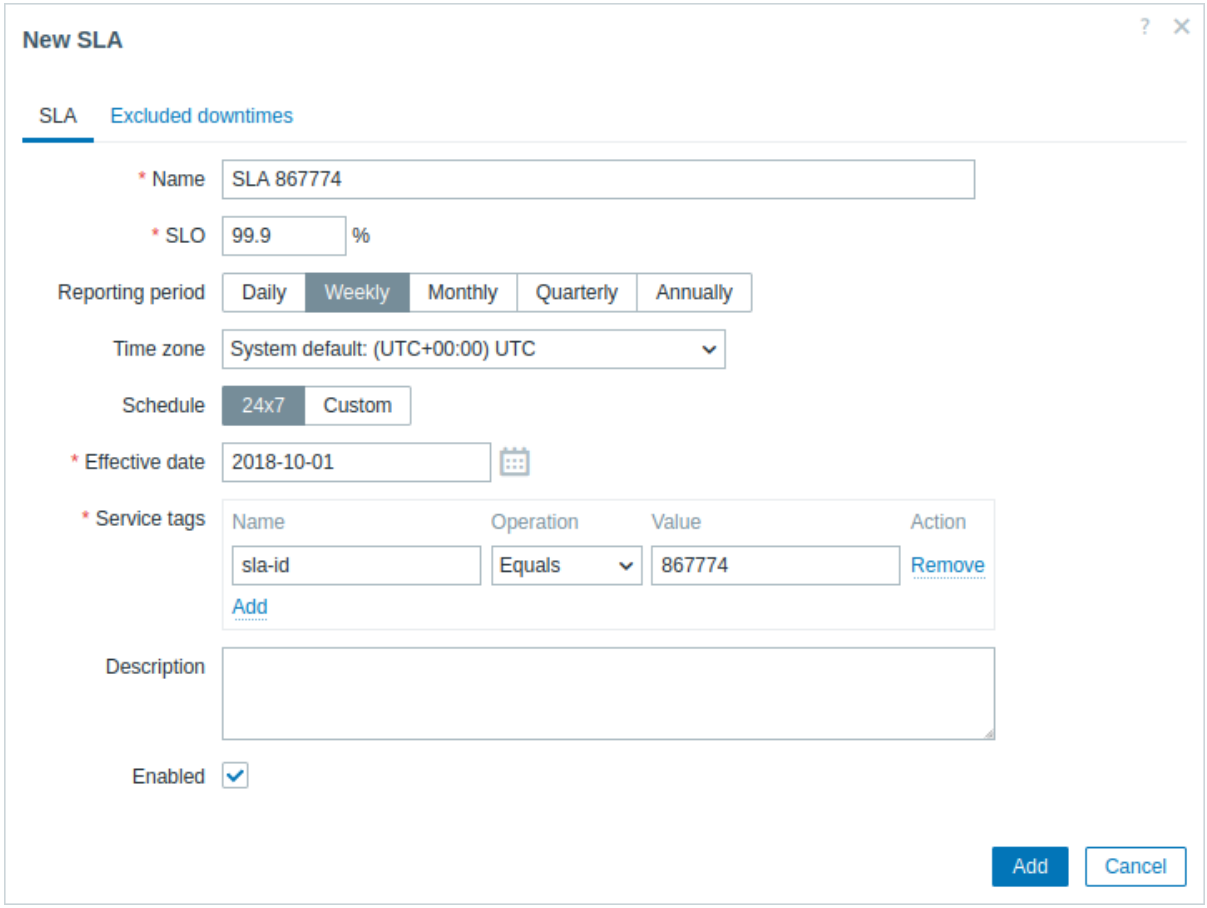
サービス->*SLA* メニューセクションでは、さまざまなサービスの SLA を設定できます。Zabbix の SLA では、サービスレベル目標（SLO）、想定稼働時間のスケジュール、および計画停止時間を定義します。

SLA とサービスは、サービスタグによって関連付けられます。同じ SLA を複数のサービスに適用でき、パフォーマンスは一致する各サービスごとに個別に測定されます。1 つのサービスに複数の SLA を割り当てることもでき、その場合は各 SLA のデータが個別に表示されます。

SLA レポートでは、Zabbix は実際のサービス可用性を測定するサービスレベル指標（SLI）データを提供します。サービスが SLA の目標を満たしているかどうかは、SLO（期待される可用性の%）と SLI（実際の可用性の%）を比較して判断されます。

設定　新しい SLA を作成するには、*SLA* の作成ボタンをクリックします。

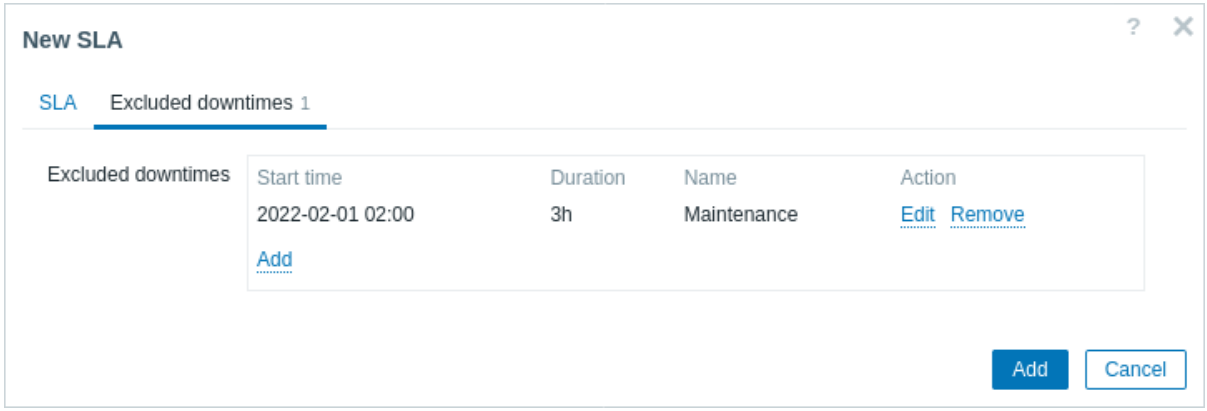
SLA タブでは、一般的な SLA パラメータを指定できます。



パラメータ	説明
名前	SLA 名を入力します。
SLO	サービスレベル目標（SLO）をパーセンテージで入力します。
レポート期間	SLA レポートで SLI データを計算するために使用する SLI レポート期間を選択します。 日次 - 各日、00:00:00 から 23:59:59 まで。 週次 - 各週、日曜日の 00:00:00 から土曜日の 23:59:59 まで。 月次 - 各月、1 日目の 00:00:00 から末日の 23:59:59 まで。 四半期ごと - 各暦四半期: Q1（1 月～3 月）、Q2（4 月～6 月）、Q3（7 月～9 月）、Q4（10 月～12 月）。 年次 - 各暦年、1 月 1 日 00:00:00 から 12 月 31 日 23:59:59 まで。

パラメータ	説明
タイムゾーン	SLA のタイムゾーンを選択します。
スケジュール	SLA のスケジュールとして、24x7 またはカスタムを選択します。
有効開始日	SLA 計算を開始する日付を選択します。
サービスタグ	この SLA を適用するサービスを識別するためのサービスタグを追加します。 名前 - サービスタグ名。完全一致で、大文字と小文字が区別されます。 操作 - タグ値が完全に一致する必要がある場合は *Equals* を、タグ値の一部が一致する必要がある場合は *Contains* を選択します（*Equals* は大文字と小文字を区別し、*Contains* は区別しません）。 値 - 選択した操作に従って検索するサービスタグ値。 少なくとも 1 つのサービスタグが一致した場合、そのサービスに SLA が適用されます。
説明	SLA の説明を追加します。
有効	チェックボックスをオンにして、SLA 計算を有効にします。

除外するダウンタイムタブでは、SLA 計算から除外するダウンタイムを指定できます。



追加をクリックして除外するダウンタイムを設定し、期間名、開始日、期間を入力します。

SLA レポート　サービスが SLA と比較してどのように稼働しているかは、SLA レポートで確認できます。SLA レポートは、次の場所から表示できます。

- *SLA* セクションで、SLA レポートのハイパーリンクをクリックする。
- *Services* セクションで、情報タブ内の SLA 名をクリックする。
- ダッシュボードのウィジェット *SLA report*。

SLA を設定すると、services セクションの *Info* タブにもサービスの稼働状況に関する情報が表示されます。

3 セットアップ例

概要　このセクションでは、Zabbix 高可用性クラスターをサービスとして監視するためのセットアップについて説明します。

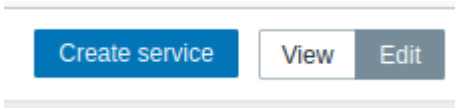
前提条件　サービス監視を設定する前に、ホストを設定しておく必要があります。

- *HA ノード 1* に少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）`component:ha-node-1`
- *HA ノード 2* に少なくとも 1 つのトリガーとタグ（できればトリガーレベルで設定）`component:ha-node-2`

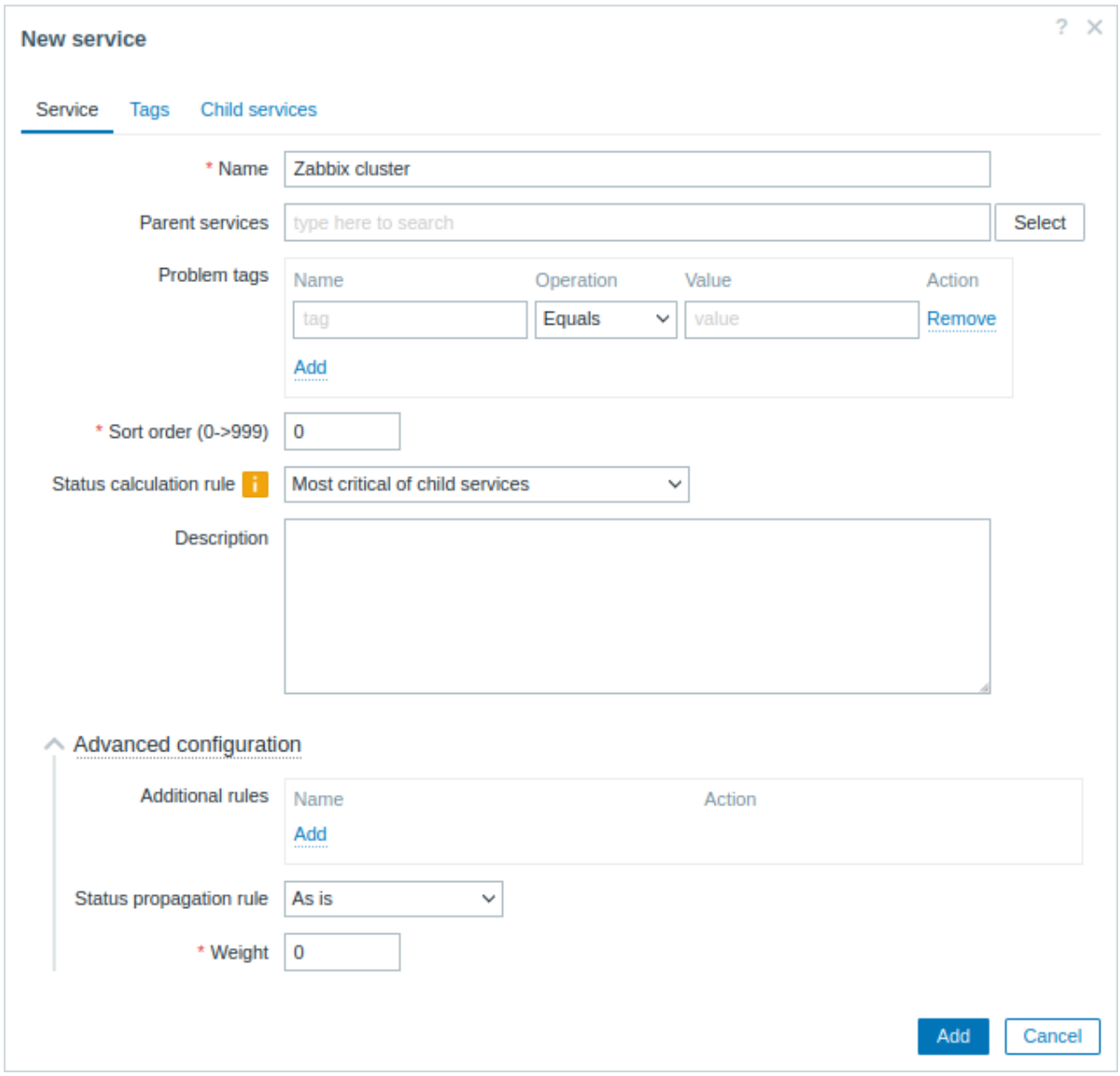
サービスツリー　次のステップは、サービスツリーを構築することです。この例では、インフラストラクチャは非常に基本的なもので、3 つのサービスで構成されています。*Zabbix cluster*（親）と、2 つの子サービス *Zabbix server node 1* および *Zabbix server node 2* です。

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

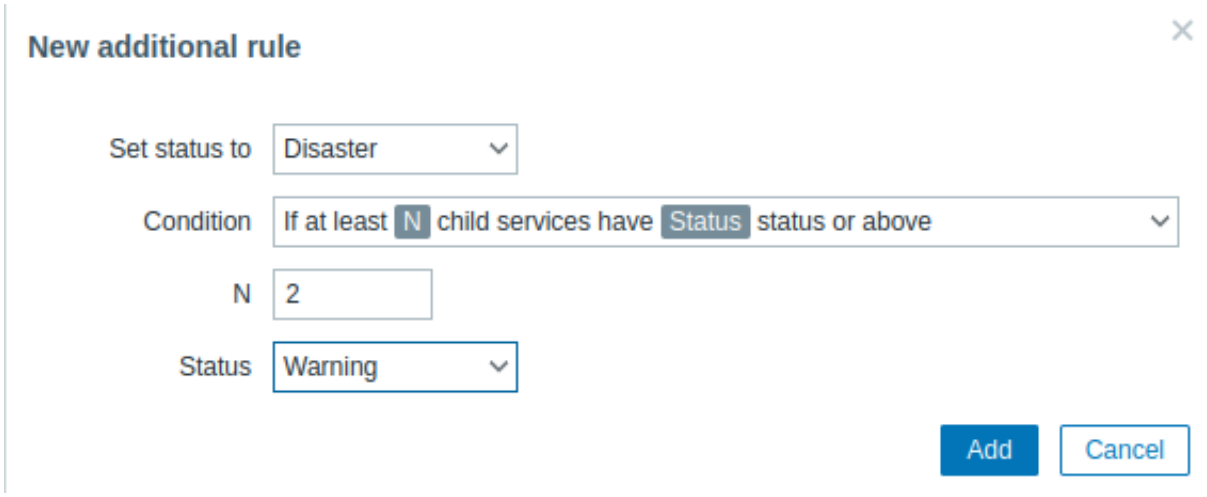
サービスページで、*Edit* モードを有効にし、Create service を押します。



サービス設定ウィンドウで、名前 *Zabbix cluster* を入力し、*Advanced configuration* ラベルをクリックして詳細設定オプションを表示します。



追加ルールを設定します。

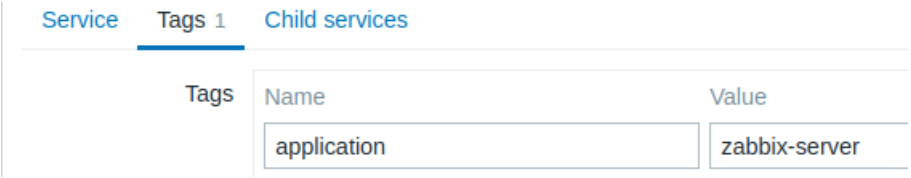


Zabbix cluster には 2 つの子サービスがあり、HA ノードごとに 1 つずつ対応します。両方の HA ノードで少なくとも *Warning* ステータスの問題が発生した場合、親サービスのステータスは *Disaster* に設定される必要があります。これを実現するには、追加ルール

を次のように設定します。

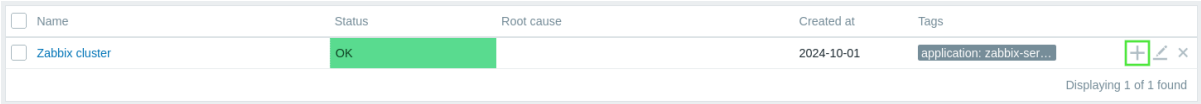
- Set status to: Disaster
- Condition: N 個以上の子サービスが Status ステータス以上である場合
- N: 2
- Status: Warning

Tags タブに切り替え、タグ `application:zabbix-server` を追加します。このタグは、後でサービスアクションおよび SLA レポートに使用されます。



新しいサービスを保存します。

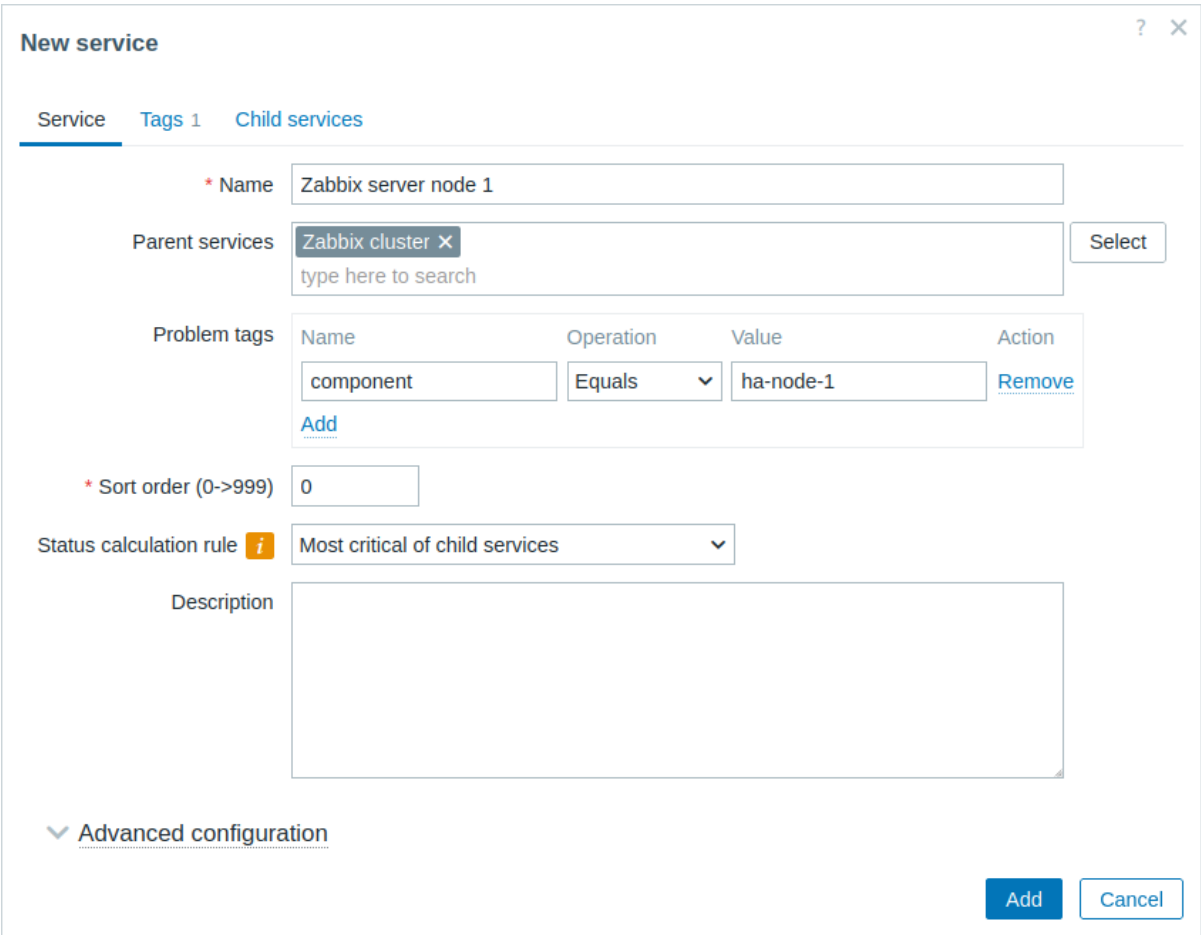
子サービスを追加するには、Zabbix cluster サービスの横にあるプラスアイコンを押します（このアイコンは編集モードでのみ表示されます）。



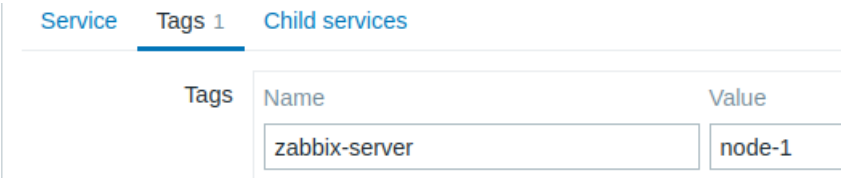
サービス設定ウィンドウで、名前 *Zabbix server node 1* を入力します。Parent services パラメータには、すでに *Zabbix cluster* が事前入力されていることに注意してください。

このサービスの可用性は、ホスト *HA node 1* 上の問題の影響を受けます。この問題には `component:ha-node-1` の障害タグが付いています。Problem tags パラメータには、次のように入力します。

- Name: component
- Operation: Equals
- Value: ha-node-1



Tags タブに切り替え、サービスタグ `zabbix-server:node-1` を追加します。このタグは、後でサービスアクションおよび SLA レポートで使用されます。



新しいサービスを保存します。

名前を「Zabbix server node 2」として、Zabbix クラスターの別の子サービスを作成します。

Problem タグを以下のように設定します。

- 名前: component
- 操作: 等しい
- 値: ha-node-2

タグタブに切り替え、サービスタグ `zabbix-server:node-2` を追加します。

新しいサービスを保存します。

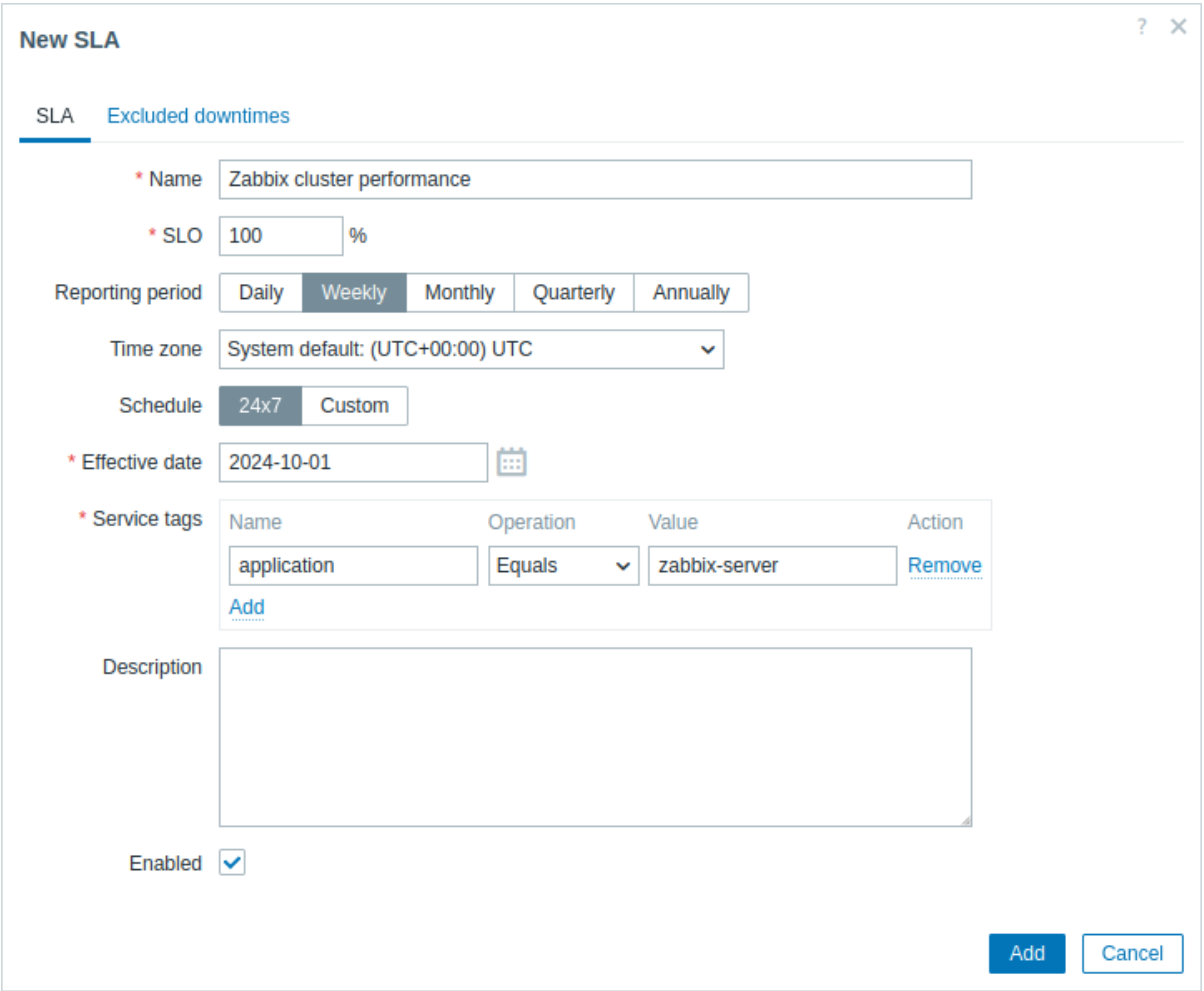
SLA　この例では、想定される Zabbix クラスターのパフォーマンスは、半年ごとの 1 時間のメンテナンス期間を除いて 100% です。

まず、新しいサービスレベル契約を追加する必要があります。

Services->SLA メニューセクションに移動し、**Create SLA** を押します。名前に *Zabbix cluster performance* を入力し、SLO を 100% に設定します。

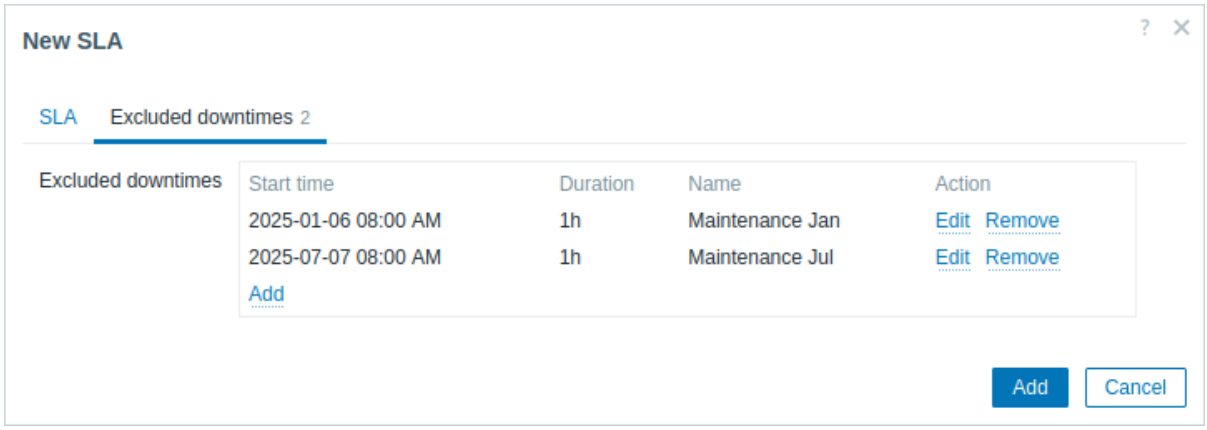
サービス Zabbix cluster には、サービスタグ `application:zabbix-server` があります。Zabbix cluster のパフォーマンス測定にこの SLA を使用するには、*Service tags* パラメータで次のように指定します。

- Name: application
- Operation: Equals
- Value: zabbix-server



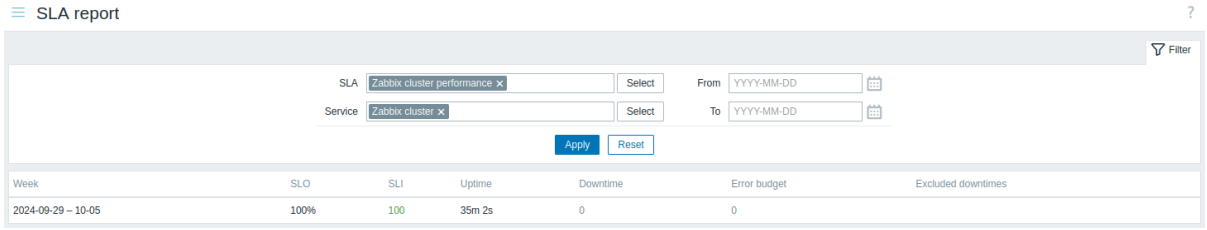
実際の環境では、必要に応じてレポート期間、タイムゾーン、開始日を更新したり、スケジュールを 24 時間 365 日からカスタムに変更したりすることもできます。この例では、デフォルト設定のままで十分です。

Excluded downtimes タブに切り替え、計画メンテナンス期間を SLA 計算から除外するため、これらの期間のダウンタイムを追加します。Excluded downtimes セクションで **Add** リンクを押し、ダウンタイム名、予定開始時刻、継続時間を入力します。

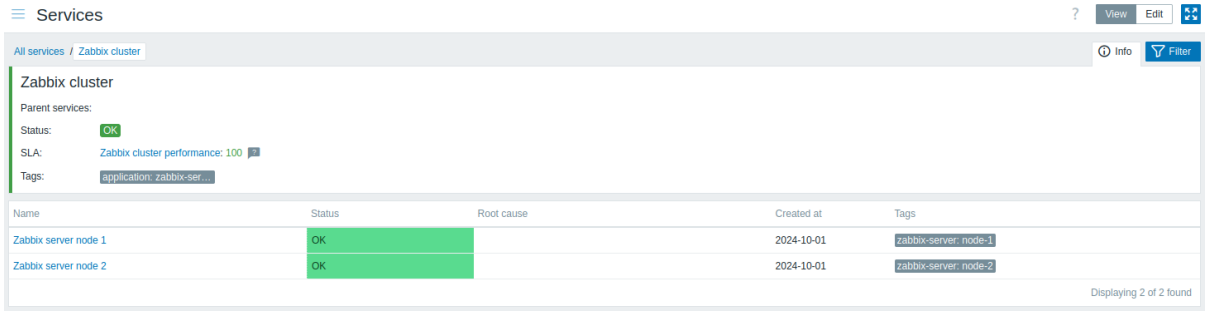


Add を押して、新しい SLA を保存します。

SLA レポートセクションに切り替えると、Zabbix cluster の SLA レポートを確認できます。



SLA 情報は *Services* セクションでも確認できます。



9. Web 監視

概要　Zabbix では、Web サイトの可用性に関する複数の側面を確認できます。

> **Attention:**
> Web 監視を実行するには、Zabbix サーバーが事前に cURL（libcurl）サポート付きで設定 されている必要があります。

Web 監視を有効にするには、Web シナリオを定義する必要があります。Web シナリオは、1 つまたは複数の HTTP リクエスト、つまり「ステップ」で構成されます。ステップは、事前定義された順序で Zabbix サーバーによって定期的に実行されます。ホストがプロキシによって監視されている場合、ステップはプロキシによって実行されます。

Web シナリオは、アイテム、トリガーなどと同じ方法でホスト/テンプレートに関連付けられます。つまり、Web シナリオはテンプレートレベルでも作成でき、その後 1 回の操作で複数のホストに適用できます。

任意の Web シナリオでは、次の情報が収集されます。

- シナリオ全体のすべてのステップにおける 1 秒あたりの平均ダウンロード速度
- 失敗したステップ番号
- 直近のエラーメッセージ

任意の Web シナリオのステップでは、次の情報が収集されます。

- 1 秒あたりのダウンロード速度
- 応答時間
- 応答コード

詳細については、Web 監視アイテムを参照してください。

Web シナリオの実行によって収集されたデータはデータベースに保持されます。このデータは、グラフ、トリガー、通知に自動的に使用されます。

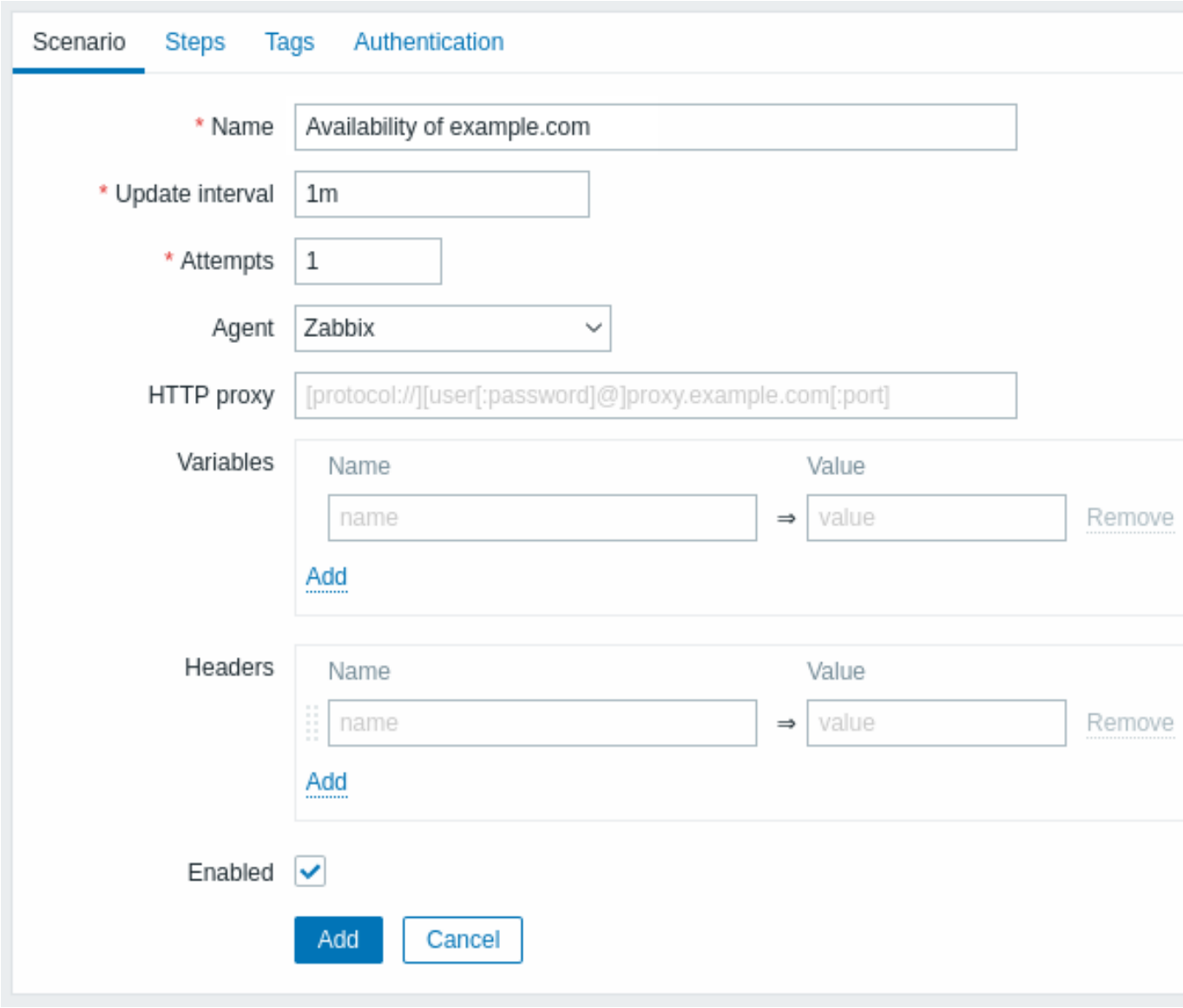
Zabbix では、取得した HTML ページに事前定義された文字列が含まれているかどうかも確認できます。ログインをシミュレートして実行し、ページ上でマウスクリックをシミュレートした経路をたどることもできます。

Zabbix の Web 監視は HTTP と HTTPS の両方をサポートしています。Web シナリオの実行時には、Zabbix は必要に応じてリダイレクトを追跡します（以下の *Follow redirects* オプションを参照）。リダイレクトの最大数は 10 にハードコードされています（cURL オプション CURLOPT_MAXREDIRS を使用）。単一のシナリオの実行中は、すべての Cookie が保持されます。

Web シナリオの設定　Web シナリオを設定するには、以下の手順を実行します。

- データ収集 → ホスト（または テンプレート）に移動します
- ホスト/テンプレートの行にある *Web* をクリックします
- 右側の *Web* シナリオの作成をクリックします（既存のシナリオを編集する場合はシナリオ名をクリックします）
- フォームにシナリオのパラメータを入力します

シナリオタブでは、Web シナリオの一般的なパラメータを設定できます。



必須の入力フィールドには、赤いアスタリスクが付いています。

シナリオのパラメータ:

Parameter	Description
名前	一意のシナリオ名です。 ユーザーマクロをサポートしています。ユーザーマクロを使用した場合、Web 監視アイテム名ではこれらのマクロは未解決のまま残る点に注意してください。
更新間隔	シナリオを実行する頻度です。 時間サフィックスをサポートしています。例: 30s、1m、2h、1d。 ユーザーマクロをサポートしています。ユーザーマクロを使用してその値が変更された場合（例: 5m → 30s）、次回のチェックは以前の値に従って実行されます（この例では、より先の時刻になります）。 新しい Web シナリオは、作成後 60 秒以内にチェックされます。
試行回数	Web シナリオのステップを実行する試行回数です。ネットワークの問題（タイムアウト、接続不可など）が発生した場合、Zabbix はステップの実行を複数回繰り返すことができます。設定した値はシナリオ内の各ステップに等しく適用されます。指定できる試行回数は最大 10 回で、デフォルト値は 1 です。 注: Zabbix は、誤ったレスポンスコードや必要な文字列の不一致を理由にステップを再試行することはありません。
エージェント	クライアントエージェントを選択します。 Zabbix は選択したブラウザとして動作します。これは、Web サイトがブラウザごとに異なるコンテンツを返す場合に便利です。 このフィールドではユーザーマクロを使用できます。
HTTP プロキシ	使用する HTTP プロキシを、 `[protocol://][username[:password]@]proxy.example.com[:port]` の形式で指定できます。 これは CURLOPT_PROXY cURL オプションを設定します。 オプションの `protocol://` プレフィックスを使用して、代替のプロキシプロトコルを指定できます（プロトコルプレフィックスのサポートは cURL 7.21.7 で追加されました）。プロトコルを指定しない場合、プロキシは HTTP プロキシとして扱われます。 デフォルトでは、ポート 1080 が使用されます。 指定した場合、このプロキシは http_proxy、HTTPS_PROXY などのプロキシ関連環境変数を上書きします。指定しない場合、プロキシはプロキシ関連環境変数を上書きしません。入力した値はそのまま渡され、妥当性チェックは行われません。 SOCKS プロキシのアドレスを入力することもできます。誤ったプロトコルを指定すると接続は失敗し、アイテムは未サポートになります。 注: HTTP プロキシでは単純認証のみサポートされています。 このフィールドではユーザーマクロを使用できます。
変数	シナリオのステップ（URL、POST 変数）で使用できる変数です。 形式は次のとおりです。 **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()}（マクロ関数を参照） 例: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} これらのマクロは、ステップ内で {username}、{password}、{hostid} などとして参照できます。Zabbix はそれらを実際の値に自動的に置き換えます。`regex:` を持つ変数は、正規表現の値を取得するために 1 つのステップが必要になるため、抽出された値を適用できるのは次のステップからである点に注意してください。 値の部分が `regex:` で始まる場合、その後ろの部分は Web ページを検索する正規表現として扱われ、一致した場合はその一致内容が変数に保存されます。一致した値を抽出できるように、少なくとも 1 つのサブグループが存在している必要があります。 ユーザーマクロおよび {HOST.*} マクロをサポートしています。 変数は、クエリフィールドまたは POST 変数のフォームデータで使用される場合は自動的に URL エンコードされますが、生の POST や URL 内で直接使用する場合は手動で URL エンコードする必要があります。

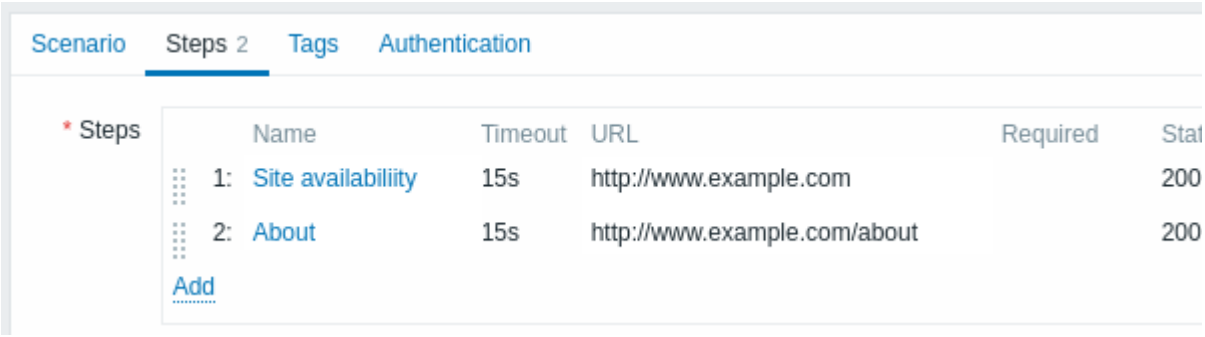
Parameter	Description
ヘッダー	HTTP ヘッダーは、リクエスト実行時に使用されます。デフォルトヘッダーとカスタムヘッダーの両方を使用できます。 ヘッダーは、シナリオレベルでドロップダウンリストから選択したエージェントタイプに応じたデフォルト設定で割り当てられ、ステップレベルでカスタム定義されていない限り、すべてのステップに適用されます。 ステップレベルでヘッダーを定義すると、シナリオレベルでドロップダウンリストから**'User-Agent'** を選択して割り当てられるデフォルトヘッダーを除き、それまでに定義されたすべてのヘッダーが自動的に破棄される点に注意してください。 ただし、'User-Agent' のデフォルトヘッダーであっても、ステップレベルで指定することで上書きできます。 シナリオレベルでヘッダーを解除するには、ステップレベルでそのヘッダー名を指定し、値を空にする必要があります。 ヘッダーは、HTTP プロトコルで表示されるのと同じ構文で記述する必要があります。また、CURLOPT_HTTPHEADER cURL オプションでサポートされる追加機能を任意で使用できます。 例: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 ユーザーマクロおよび {HOST.*} マクロをサポートしています。
有効	このチェックボックスがオンの場合、シナリオはアクティブです。オフの場合は無効です。

既存のシナリオを編集する場合、フォームでは次の 2 つの追加ボタンを使用できます。

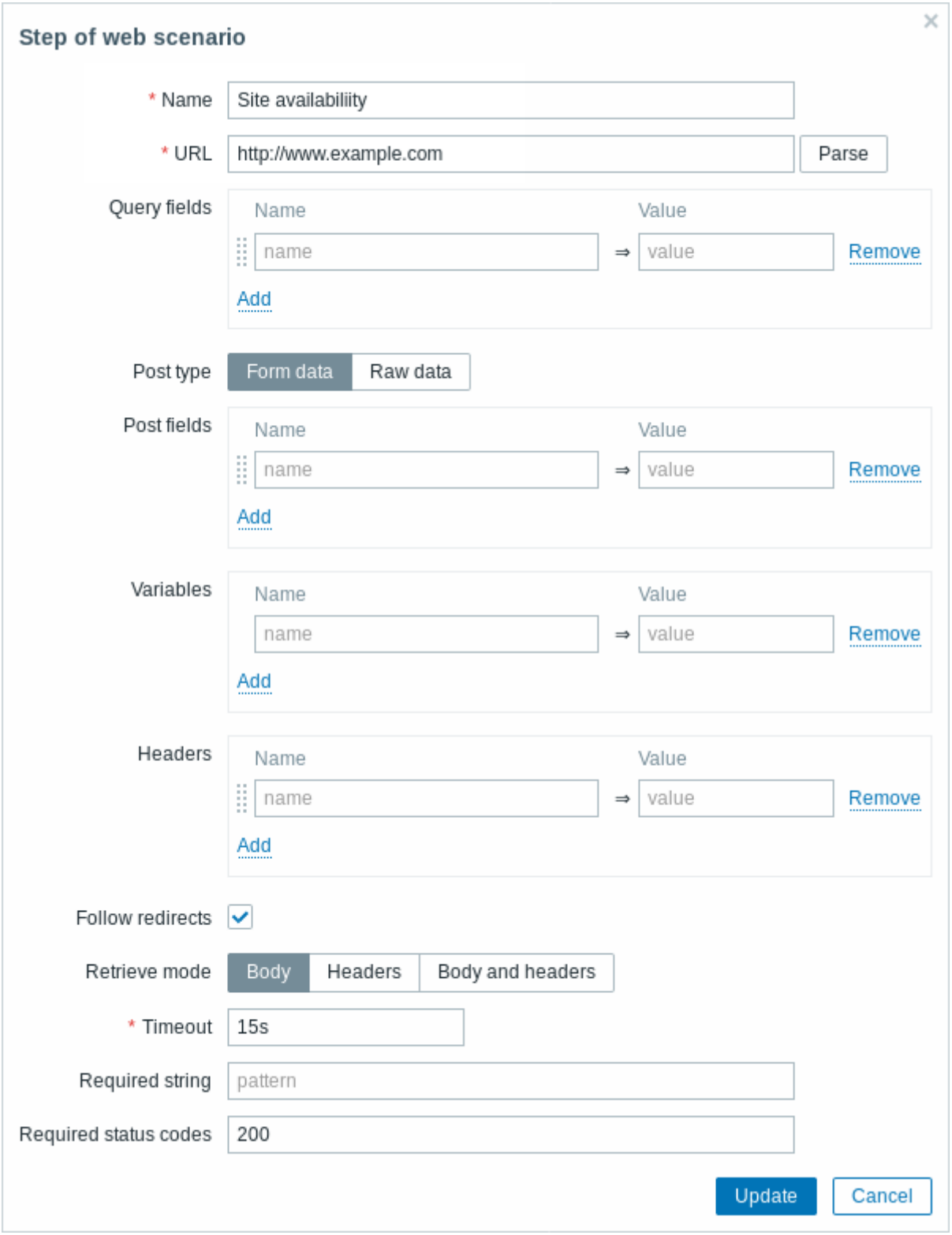
Clone	既存のシナリオのプロパティに基づいて別のシナリオを作成します。
Clear history and trends	シナリオの履歴データとトレンドデータを削除します。これにより、データ削除直後にサーバーがシ

Note:
HTTP プロキシフィールドを空のままにした場合、HTTP プロキシを使用する別の方法として、プロキシ関連の環境変数を設定できます。
HTTP チェックの場合は、Zabbix server ユーザーに対して **http_proxy** 環境変数を設定します。例: `http_proxy=http://proxy_ip:proxy_port`。
HTTPS チェックの場合は、**HTTPS_PROXY** 環境変数を設定します。例: `HTTPS_PROXY=http://proxy_ip:proxy_port`。
詳細は、シェルコマンド *# man curl* を実行して確認できます。

ステップタブでは、Web シナリオのステップを設定できます。Web シナリオのステップを追加するには、ステップブロック内の 追加をクリックします。



Note:
シークレットユーザーマクロは、URL 内で使用してはいけません。"******" に解決されるためです。



ステップの設定

ステップのパラメータ:

パラメータ	説明
名前	一意のステップ名。 ユーザーマクロをサポートしています。ユーザーマクロを使用した場合、Web 監視アイテム名ではこれらのマクロは未解決のまま残る点に注意してください。

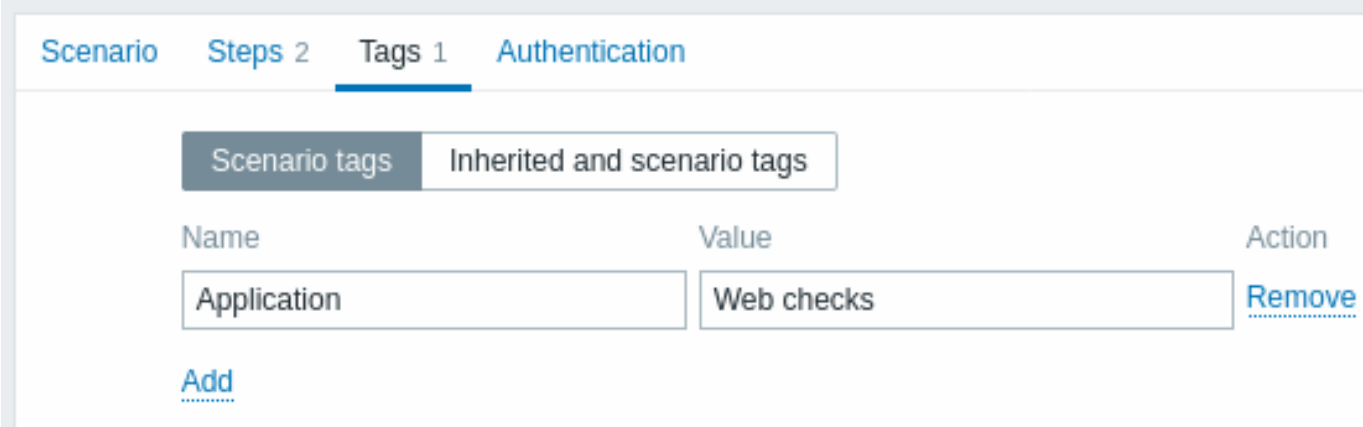
パラメータ	説明
URL	接続してデータを取得する URL。例: https://www.example.com http://www.example.com/download ドメイン名は Unicode 文字で指定できます。Web シナリオのステップ実行時に、自動的に ASCII へ punycode 変換されます。 *Parse* ボタンを使用すると、オプションのクエリフィールド（例: ?name=Admin&password=mypassword）を URL から分離し、属性と値を *Query fields* へ移動して自動的に URL エンコードできます。 URL では、{macro} 構文を使用して変数を利用できます。変数は {{macro}.urlencode()} 構文を使用して手動で URL エンコードできます。 ユーザーマクロおよび {HOST.*}マクロをサポートしています。 2048 文字までに制限されています。
Query fields	URL 用の HTTP GET 変数。 属性と値のペアとして指定します。 値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、または {HOST.*} マクロの値は解決された後、自動的に URL エンコードされます。{{macro}.urlencode()} 構文を使用すると、二重に URL エンコードされます。 ユーザーマクロおよび {HOST.*}マクロをサポートしています。
Post	HTTP POST 変数。 **Form data** モードでは、属性と値のペアとして指定します。 値は自動的に URL エンコードされます。シナリオ変数、ユーザーマクロ、または {HOST.*} マクロの値は解決された後、自動的に URL エンコードされます。 **Raw data** モードでは、属性/値は 1 行で表示され、**&** 記号で連結されます。 生の値は {{macro}.urlencode()} または {{macro}.urldecode()} 構文を使用して手動で URL エンコード/デコードできます。 例: id=2345&userid={user} {user} が Web シナリオの変数として定義されている場合、ステップ実行時にその値へ置き換えられます。変数を URL エンコードしたい場合は、{user} を {{user}.urlencode()} に置き換えてください。 ユーザーマクロおよび {HOST.*}マクロをサポートしています。
Variables	GET および POST 機能で使用できるステップレベルの変数。 属性と値のペアとして指定します。 ステップレベルの変数は、シナリオレベルの変数または前のステップからの変数を上書きします。ただし、ステップレベルの変数の値が影響するのは次のステップのみであり、現在のステップには影響しません。 形式は次のとおりです: **{macro}**=value **{macro}**=regex:<regular expression> 詳細は、シナリオレベルの変数の説明を参照してください。 変数は、クエリフィールドまたは POST 変数のフォームデータで使用する場合は自動的に URL エンコードされますが、生の POST または URL 内で直接使用する場合は手動で URL エンコードする必要があります。
Headers	リクエスト実行時に送信されるカスタム HTTP ヘッダー。 属性と値のペアとして指定します。 ステップレベルで定義されたヘッダーは、その特定のステップで使用されます。 ステップレベルでヘッダーを定義すると、シナリオレベルのドロップダウンリストで**'User-Agent'** を選択して割り当てられるデフォルトヘッダーを除き、それまでに定義されていたすべてのヘッダーは自動的に破棄される点に注意してください。 ただし、'User-Agent' のデフォルトヘッダーであっても、ステップレベルで指定することで上書きできます。 たとえば、ヘッダー名を指定して値を設定しない場合、シナリオレベルのデフォルトヘッダーは解除されます。 ユーザーマクロおよび {HOST.*} マクロをサポートしています。 これは cURL オプションCURLOPT_HTTPHEADERを設定します。
Follow redirects	HTTP リダイレクトに従うには、チェックボックスをオンにします。 これは cURL オプションCURLOPT_FOLLOWLOCATIONを設定します。
Retrieve mode	取得モードを選択します: **Body** - HTTP レスポンスから本文のみを取得 **Headers** - HTTP レスポンスからヘッダーのみを取得 **Body and headers** - HTTP レスポンスから本文とヘッダーを取得

パラメータ	説明
Timeout	Zabbix は、URL の処理に設定された時間（1 秒から最大 1 時間）を超えて費やすことはありません。実際には、このパラメータは URL への接続を確立する最大時間と、HTTP リクエストを実行する最大時間を定義します。したがって、Zabbix はこのステップに **2 x Timeout** 秒を超えて費やすことはありません。 時間サフィックスをサポートしています（例: 30s、1m、1h）。ユーザーマクロをサポートしています。
Required string	必須の正規表現パターン。 取得したコンテンツ（HTML）が必須パターンに一致しない場合、このステップは失敗します。空の場合、必須文字列のチェックは実行されません。 例: Homepage of Zabbix Welcome.*admin 注: このフィールドでは、Zabbix Web インターフェースで作成した正規表現の参照はサポートされていません。 ユーザーマクロおよび {HOST.*}マクロをサポートしています。
Required status codes	期待される HTTP ステータスコードの一覧。Zabbix が一覧にないコードを受け取った場合、このステップは失敗します。 空の場合、ステータスコードのチェックは実行されません。 例: 200,201,210-299 ユーザーマクロをサポートしています。

Note:
Web シナリオのステップに対する変更は、シナリオ全体を保存したときにのみ保存されます。

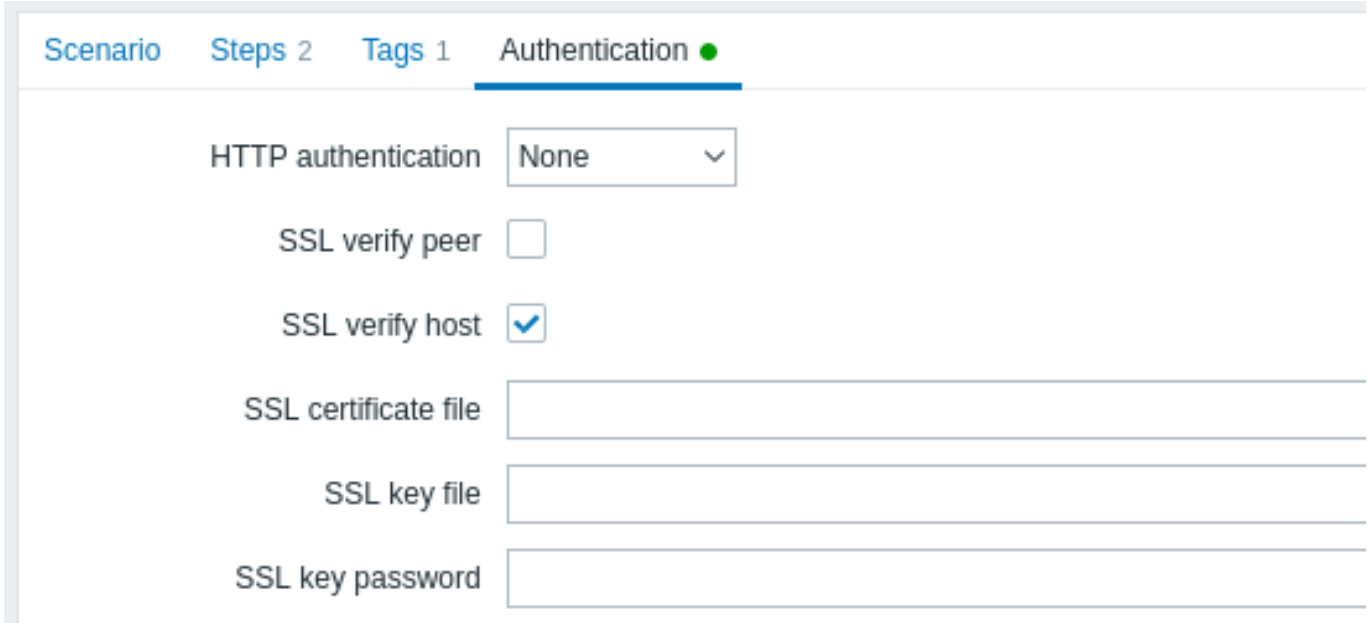
Web 監視ステップの設定方法については、実際の例も参照してください。

タグの構成　**Tags** タブでは、シナリオレベルのタグを定義できます。



タグ付けにより、Web シナリオと Web 監視アイテムをフィルタリングできます。

認証の設定　認証タブでは、シナリオの認証オプションを設定できます。
タブ名の横にある緑色の点は、何らかの HTTP 認証が有効になっていることを示します。



認証パラメータ:

Parameter	Description
HTTP 認証	認証オプションを選択します: なし - 認証は使用されません。 **Basic** - Basic 認証を使用します。 **NTLM** - NTLM（Windows NT LAN Manager）認証を使用します。 **Kerberos** - Kerberos 認証を使用します（参照: Zabbix での Kerberos の設定）。 **Digest** - Digest 認証を使用します。
ユーザー	ユーザー名を入力します（最大 255 文字）。 *HTTP* 認証が Basic、NTLM、Kerberos、または Digest に設定されている場合に使用できます。ユーザーマクロをサポートしています。
パスワード	ユーザーのパスワードを入力します（最大 255 文字）。 *HTTP* 認証が Basic、NTLM、Kerberos、または Digest に設定されている場合に使用できます。ユーザーマクロをサポートしています。
SSL verify peer	Web サーバーの SSL 証明書を検証するには、チェックボックスをオンにします。 サーバー証明書は、システム全体の認証局（CA）の場所から自動的に取得されます。CA ファイルの場所は、Zabbix サーバーまたはプロキシの設定パラメータSSLCALocation を使用して上書きできます。 これは、CURLOPT_SSL_VERIFYPEER cURL オプションを設定します。
SSL verify host	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証するには、チェックボックスをオンにします。 これは、CURLOPT_SSL_VERIFYHOST cURL オプションを設定します。
SSL certificate file	クライアント認証に使用する SSL 証明書ファイルの名前です。証明書ファイルは PEM[1] 形式である必要があります。証明書ファイルに秘密鍵も含まれている場合は、*SSL key file* フィールドは空のままにしてください。鍵が暗号化されている場合は、*SSL key password* フィールドにパスワードを指定してください。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータSSLCertLocation で指定します。 `HOST.*` マクロおよびユーザーマクロをこのフィールドで使用できます。 これは、CURLOPT_SSLCERT cURL オプションを設定します。
SSL key file	クライアント認証に使用する SSL 秘密鍵ファイルの名前です。秘密鍵ファイルは PEM[1] 形式である必要があります。このファイルを含むディレクトリは、Zabbix サーバーまたはプロキシの設定パラメータSSLKeyLocation で指定します。 `HOST.*` マクロおよびユーザーマクロをこのフィールドで使用できます。 これは、CURLOPT_SSLKEY cURL オプションを設定します。
SSL key password	SSL 秘密鍵ファイルのパスワードです。 ユーザーマクロをこのフィールドで使用できます。 これは、CURLOPT_KEYPASSWD cURL オプションを設定します。

Attention:
[1] Zabbix は、PEM 形式の証明書ファイルおよび秘密鍵ファイルのみをサポートしています。証明書および秘密鍵のデータが PKCS #12 形式のファイル（通常は拡張子 *.p12 または *.pfx）にある場合は、次のコマンドを使用して PEM ファイルを生成できます:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Note:
Zabbix サーバーは、再起動なしで証明書の変更を反映します。

Note:
クライアント証明書と秘密鍵が 1 つのファイルに含まれている場合は、それを”SSL certificate file” フィールドに指定し、”SSL key file” フィールドは空のままにしてください。証明書と鍵は引き続き PEM 形式である必要があります。証明書と鍵の結合は簡単です:

```
cat client.crt client.key > client.pem
```

表示　ホストに設定された Web シナリオを表示するには、監視 → ホストに移動し、一覧でホストを見つけて、最後の列にある *Web* ハイパーリンクをクリックします。詳細情報を表示するには、シナリオ名をクリックします。



Web シナリオの概要は、ダッシュボードでも Web 監視ウィジェットを使用して表示できます。

Web シナリオ実行の最近の結果は、監視 → 最新データセクションで確認できます。

拡張監視　受信した HTML ページの内容をログに記録する必要がある場合があります。
これは、Web シナリオのいずれかのステップが失敗した場合に特に有用です。
その目的には、デバッグレベル 5（trace）を使用します。
このレベルは、サーバーおよびプロキシの設定ファイルで設定するか、ランタイム制御オプション（`-R log_level_increase="http poller,N"`、ここで N はプロセス番号）を使用して設定できます。
以下の例は、デバッグレベル 4 がすでに設定されている場合に、拡張監視を開始する方法を示しています。

```
# すべてのhttp pollerのログレベルを上げる：
zabbix_server -R log_level_increase="http poller"

# 2番目のhttp pollerのログレベルを上げる：
zabbix_server -R log_level_increase="http poller,2"
```

拡張 Web 監視が不要な場合は、`-R log_level_decrease` オプションを使用して停止できます。

1 Web 監視アイテム

概要

Web シナリオが作成されると、モニタリング用にいくつかの新しいアイテムが自動的に追加されます。

すべてのアイテムは、Web シナリオからタグを継承します

シナリオアイテム

シナリオが作成されると、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for scenario <Scenario>	このアイテムはシナリオ全体のダウンロード速度 (1 秒あたりのバイト数)、つまりすべてのステップの平均速度を収集します。 アイテムキー: web.test.in[Scenario„bps] タイプ: *Numeric(float)*
Failed step of scenario <Scenario>	このアイテムはシナリオで失敗したステップの番号を表示します。すべてのステップが正常に実行されると、0 が返されます。 アイテムキー: web.test.fail[Scenario] タイプ: *Numeric(unsigned)*
Last error message of scenario <Scenario>	このアイテムはシナリオの最後のエラーメッセージテキストを返します。新しい値は、シナリオに失敗したステップがある場合にのみ保存されます。すべての手順に問題がない場合、新しい値は収集されません。 アイテムキー: web.test.error[Scenario] タイプ: *Character*

”Scenario” の代わりに実際のシナリオ名が使用されます。

Note:
シナリオ名にユーザーマクロが含まれている場合、これらのマクロは Web 監視アイテム名で未解決のままになります。

 シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

Note:
Web 監視アイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

これらのアイテムを使用して、トリガーを作成し、通知条件を定義できます。

例 1

「Web シナリオに失敗しました」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.fail[Scenario])<>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 2

トリガー名に問題の説明を含む"Web scenario failed" トリガーを作成するには、次の名前でトリガーを定義します。

```
Web scenario "Scenario" failed: {ITEM.VALUE}
```

さらに次のようにトリガー式を定義します

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

例 3

「Web アプリケーションが遅い」トリガーを作成するには、次のようにトリガー式を定義します。

```
last(/host/web.test.in[Scenario,,bps])<10000
```

必ず'Scenario' を本来のシナリオ名に置き換えてください。

シナリオのステップアイテム

ステップが作成されるとすぐに、Zabbix は監視用に次のアイテムを自動的に追加します。

アイテム	説明
Download speed for step <Step> of scenario <Scenario>	このアイテムはステップのダウンロード速度（バイト/秒）を取得します。 Item key: web.test.in[Scenario,Step,bps] Type: *Numeric(float)*
Response time for step <Step> of scenario <Scenario>	このアイテムはステップの応答時間を秒単位で取得します。リクエストの開始からすべての情報が転送されるまでがカウントされます。 Item key: web.test.time[Scenario,Step,resp] Type:*Numeric(float)*
Response code for step <Step> of scenario <Scenario>	このアイテムはステップの応答コードを取得します。 Item key:web.test.rspcode[Scenario,Step]Type: *Numeric(unsigned)*

"Scenario" と"Step" の代わりに、実際のシナリオとステップ名がそれぞれ使用されます。

Note:
Web モニタリングアイテムには、30 日の履歴と 90 日のトレンド保持期間が追加されます。

Note:
シナリオ名が二重引用符で始まるか、カンマまたは角括弧が含まれている場合、アイテムキーで適切に引用符で囲まれます。その他の場合、追加の見積もりは実行されません。

これらのアイテムを使用してトリガーを作成し、通知条件を定義できます。たとえば「ZabbixGUI ログインが遅すぎる」トリガーを作成するには次のようにトリガー式を定義します。

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 実際のシナリオ

概要

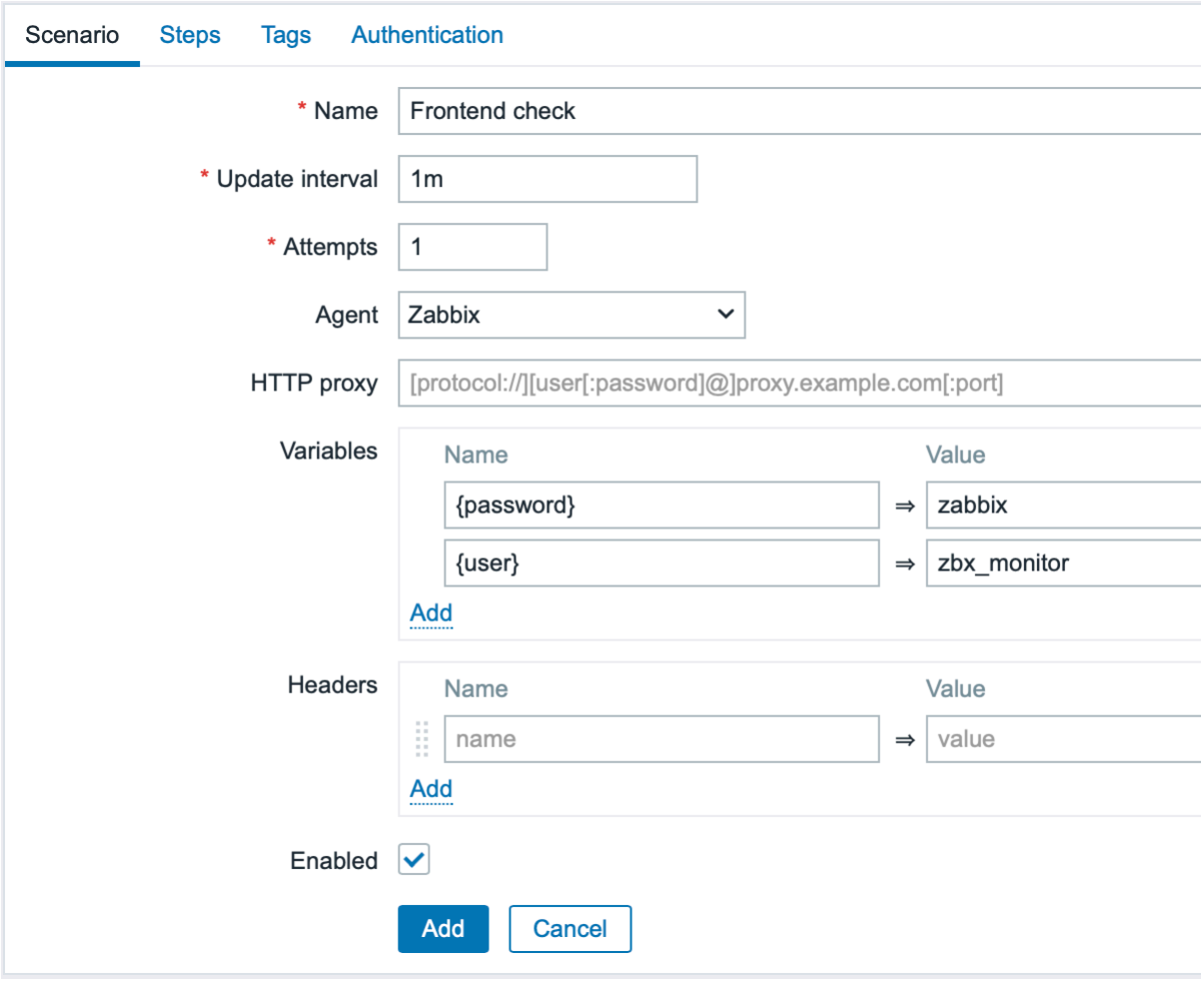
このセクションでは、Web 監視をどのように使用できるかについて、実際の手順に沿った例を紹介します。

Zabbix の Web 監視を使用して、Zabbix Web インターフェースを監視します。目的は、それが利用可能であるか、正しいコンテンツを提供しているか、そしてどの程度迅速に動作するかを判断することです。これを実現するには、最初のページの可用性の確認、ユーザー名とパスワードを使用したログイン、ログイン成功の検証、ログアウト、そしてログアウトの確認など、いくつかのステップが必要です。

シナリオ

新しい Web シナリオを追加する

データ収集 → ホストに移動し、ホストを選択して、そのホストの行にある *Web* をクリックします。
次に、*Web* シナリオの作成をクリックします。



新しいシナリオのフォームで、以下のフィールドを入力します。

- 名前 - Frontend check
- 更新間隔 - 1m
- 試行回数 - 1
- エージェント - Zabbix

変数セクションで、*{password}* と *{user}* の 2 つの変数を追加します。
値として、既存の Zabbix ユーザーの認証情報を入力します。

安全上の理由から、監視用途で使用するために、最小限の権限を持つ別のユーザーを作成することを推奨します。

必要に応じて、タグタブに切り替えて Web シナリオのタグを追加します。

Note:
設定が完了すると、この Web シナリオは自動的に Zabbix trapper アイテムをホストに追加します。
Web シナリオのタグを使用すると、関連するアイテムやトリガーをすばやく識別したり、収集されたデータを検索したりできます。
たとえば、このチュートリアルに適したタグは `component: web-scenario` および/または `target: frontend` です。

Web シナリオのステップを設定する

Steps タブに切り替え、シナリオのステップを定義します。個々のステップを追加するには、*Add* ボタンをクリックします。

共通フィールド

以下で説明する各ステップでは、ステップ固有のフィールドに加えて、次のフィールドを入力します。

- **URL** - Zabbix Web インターフェースの URL
- **Timeout** - 15s
- **Required status codes** - 200

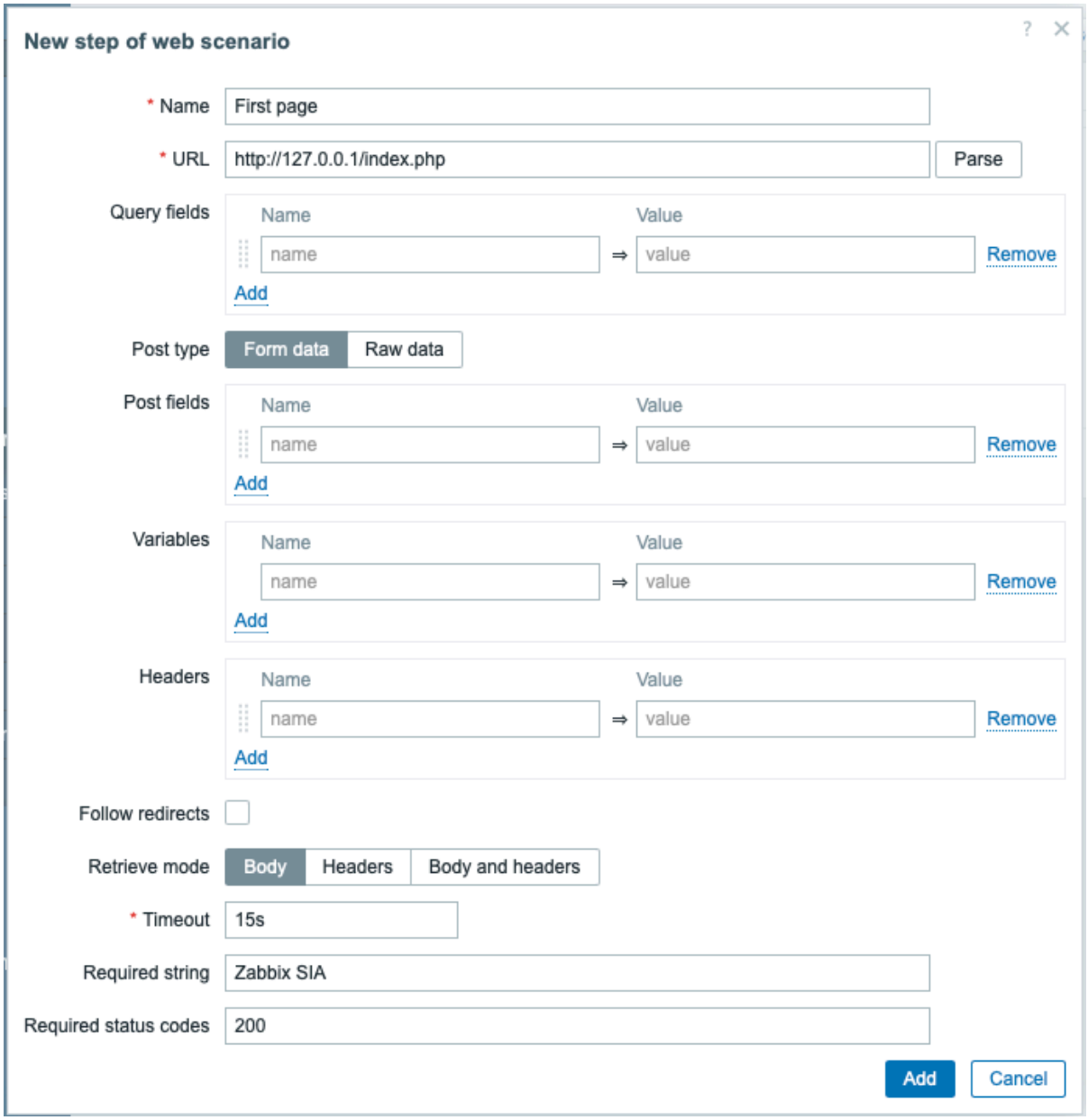
Web シナリオのステップ 1

最初のページが正しく応答し、HTTP レスポンスコード 200 を返し、テキスト「Zabbix SIA」を含んでいることを確認します。

- **Name** フィールドに、*First page* と入力します。

- **Required string** フィールドに、*Zabbix SIA* と入力します。
- 共通フィールドを入力します。

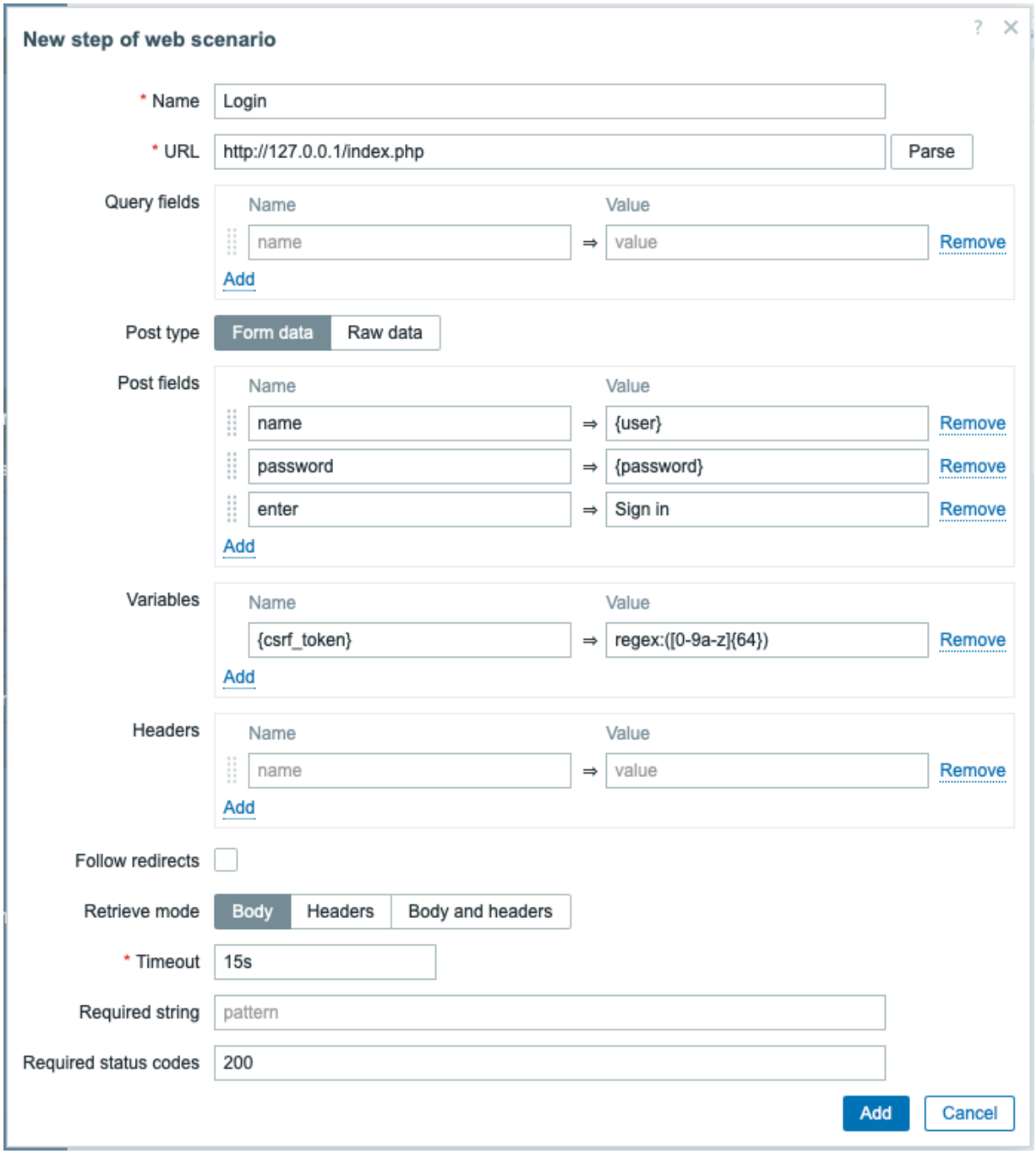
ステップの設定が完了したら、*Add* ボタンを押します。



Web シナリオのステップ 2

シナリオレベルで定義したマクロ（変数）*{user}* および *{password}* を使用して、Zabbix の Web インターフェースにログインします。

- **Name** フィールドに *Login* と入力します。
- **Post fields** セクションで、3 つの post フィールドを追加します。
  - *name* に値 *{user}*
  - *password* に値 *{password}*
  - *enter* に値 *Sign in*
- **Variables** セクションで、新しい変数 *{csrf_token}* を値 *regex:([0-9a-z]{64})* で追加します。この変数は、割り当てられた CSRF トークンの値を取得し、ステップ 4 で再利用するためのものです。
- 共通フィールド を入力します。

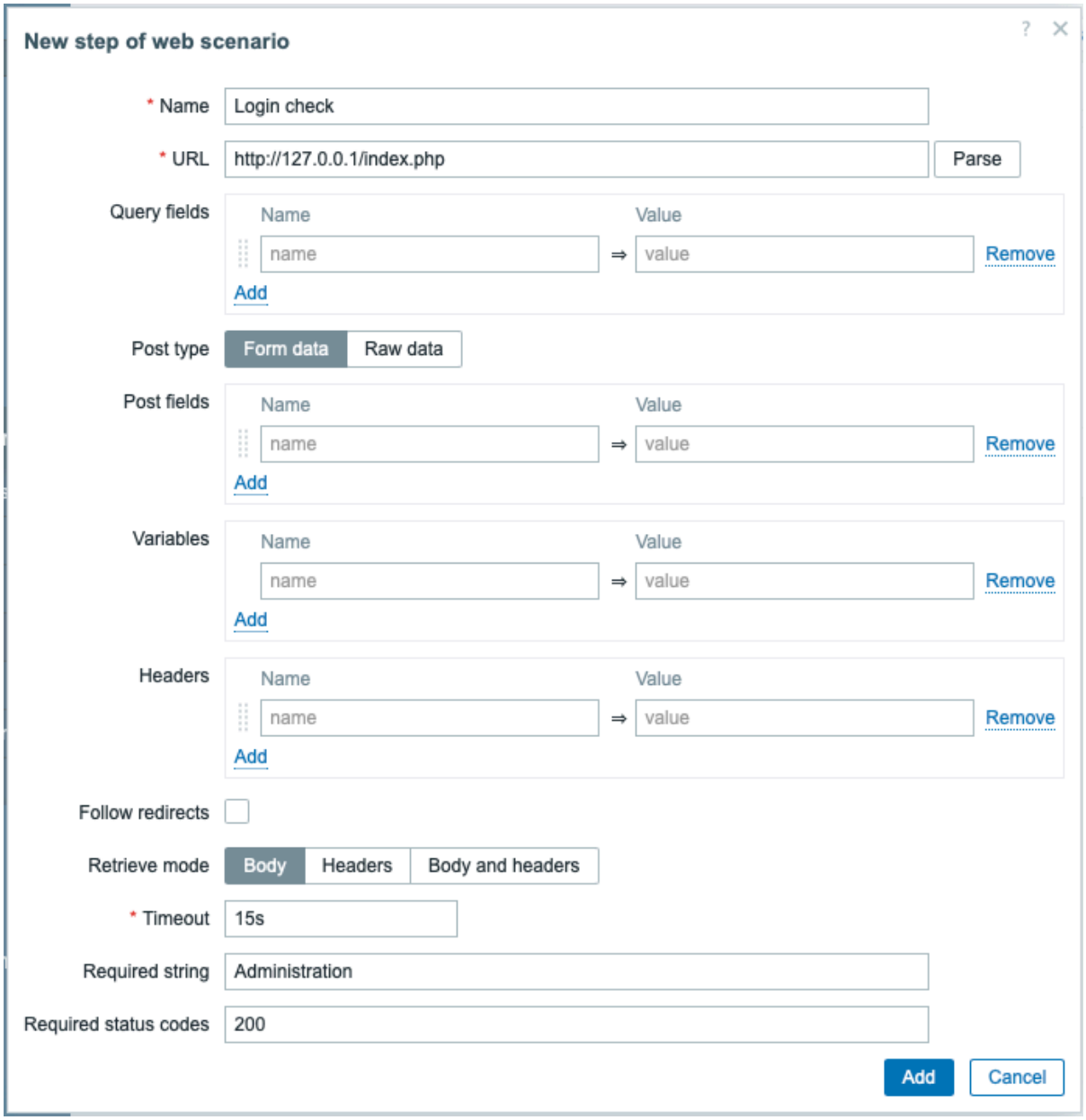


Attention:
Zabbix の Web インターフェースでは、ログイン時に JavaScript によるリダイレクトが使用されるため、まずログインを実行する必要があり、ログイン後の機能は後続のステップでのみ確認できます。さらに、ログインステップでは **index.php** ファイルへの完全な URL を使用する必要があります。

Web シナリオのステップ 3

ログイン後、ログイン時にのみ表示される文字列（たとえば *Administration*）を確認して、成功を検証します。

- **Name** フィールドに、*Login check* と入力します。
- **Required string** フィールドに、*Administration* と入力します。
- 共通フィールド を入力します。

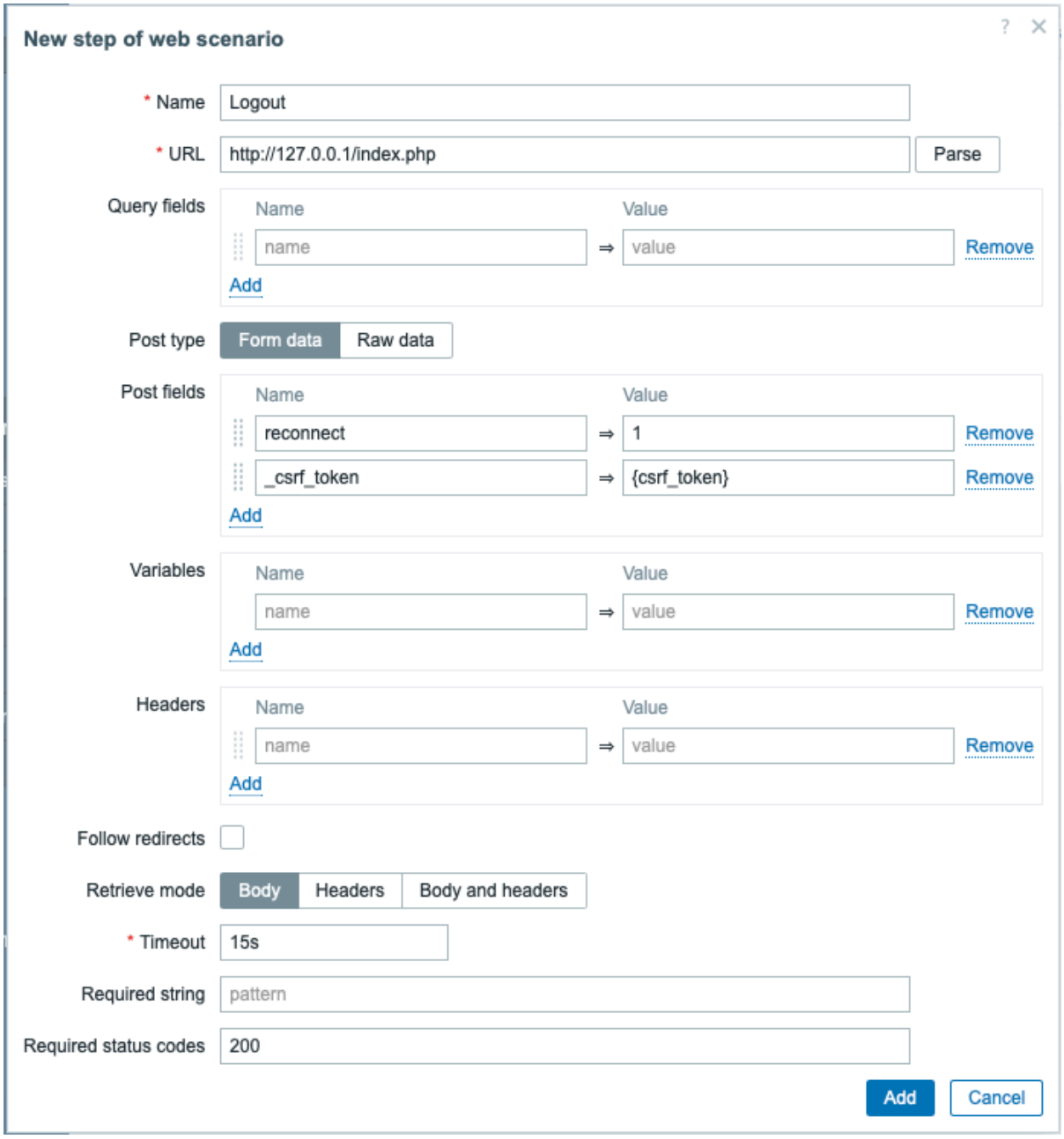


Web シナリオのステップ 4

Web インターフェースへのアクセス性とログインが確認できたら、ログアウトのステップを追加します。追加しない場合、Zabbix データベースに未終了のセッションレコードが多数蓄積されます。

- **Name** フィールドに *Logout* と入力します。
- **Post fields** セクションで、2 つの POST フィールドを追加します。
  - *reconnect* に値 *1* を設定します。
  - *_csrf_token* に値 *{csrf_token}* を設定します。
- 共通フィールド を入力します。

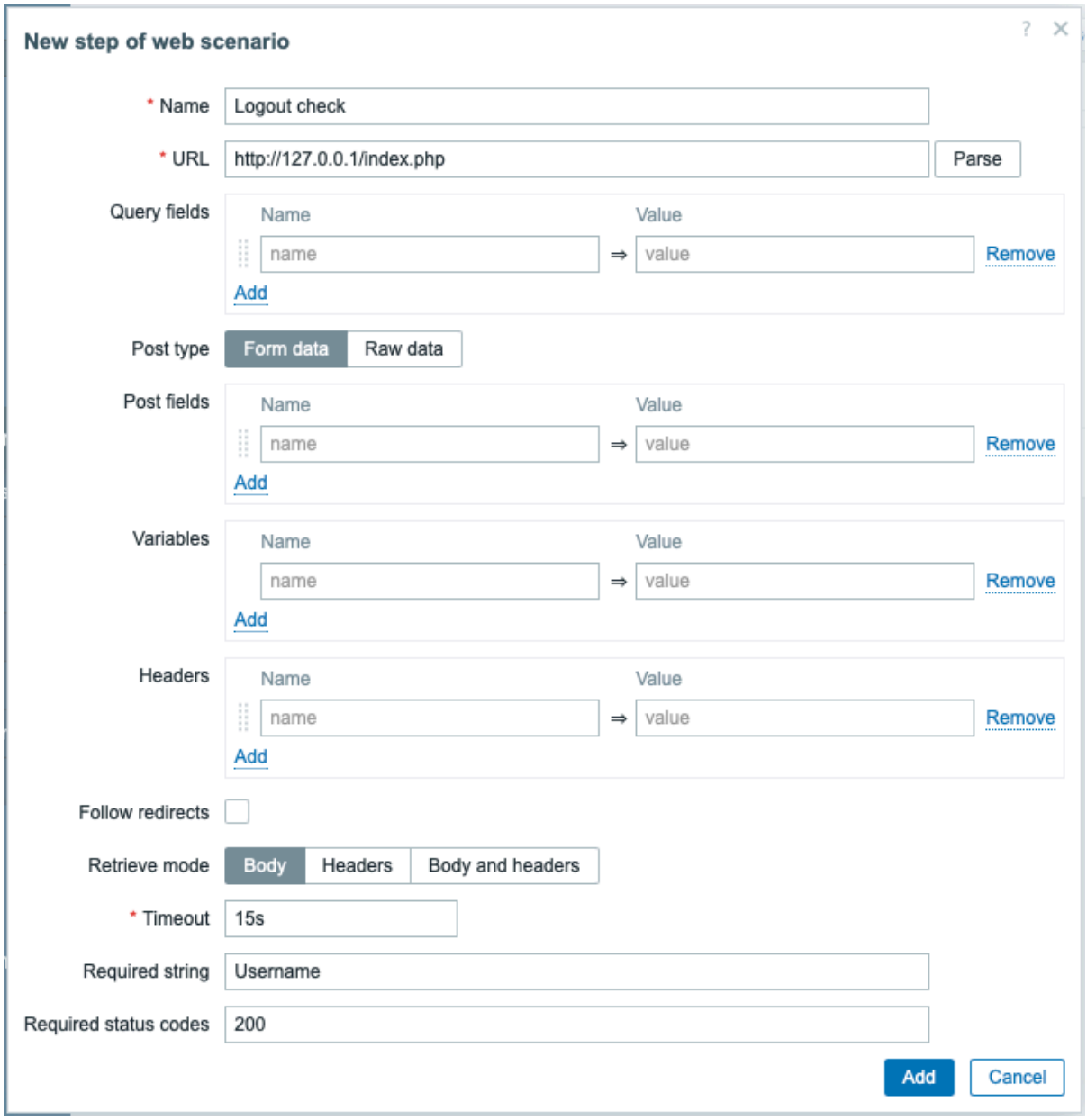
このステップでは、ステップ 2 で取得した変数 {csrf_token} を使用します。



Web シナリオのステップ 5

ログアウトを確認するには、**Username** 文字列を確認します。

- **Name** フィールドに、*Logout check* と入力します。
- **Required string** フィールドに、*Username* と入力します。
- 共通フィールド を入力します。



ステップの完全な設定

Web シナリオのステップの完全な設定は、次のようになります。

	Name	Timeout	URL	Required	Status codes	Action
1:	First page	15s	http://127.0.0.1/index.php	Zabbix SIA	200	Remove
2:	Log in	15s	http://127.0.0.1/index.php		200	Remove
3:	Log in Check	15s	http://127.0.0.1/index.php	Administrati...	200	Remove
4:	Log out	15s	http://127.0.0.1/index.php		200	Remove
5:	Log out check	15s	http://127.0.0.1/index.php	Username	200	Remove

Add

結果を確認する

完成した Web 監視シナリオを保存します。

シナリオはホストに追加されます。Web シナリオの情報を表示するには、監視 → ホストに移動し、一覧でホストを見つけて、最後の列にある Web のハイパーリンクをクリックします。



シナリオ名をクリックすると、より詳細な統計情報を確認できます。



10. 仮想マシン監視

概要　Zabbix は、ローレベルディスカバリ ルールを使用して VMware ハイパーバイザーおよび仮想マシンを自動的に検出し、事前定義されたホストプロトタイプに基づいて、それらを監視するためのホストを作成できます。

Zabbix には、VMware vCenter または ESXi ハイパーバイザーを監視するためのすぐに使えるテンプレートも含まれています。

必要な VMware vCenter または vSphere の最小バージョンは 5.1 です。

データ収集　仮想マシンの監視は、次の 2 つのステップで構成されます。

1. Zabbix の *vmware collector* プロセスが仮想マシンのデータを収集します。これらのプロセスは、SOAP プロトコルを介して VMware Web サービスから必要な情報を取得し、前処理を行ったうえで、Zabbix サーバーの共有メモリに保存します。
2. Zabbix の *poller* プロセスが、Zabbix のシンプルチェックVMware monitoring item keys を使用してデータを取得します。

Zabbix は、収集したデータを VMware の設定データと VMware のパフォーマンスカウンターデータに分類します。
これら 2 種類のデータは、いずれも *vmware collector* プロセスによって独立して収集されます。

VMware のパフォーマンスカウンター情報に基づいて、次の統計情報を利用できます。

- データストア
- ディスクデバイス
- CPU
- 電源
- ネットワークインターフェース
- カスタムパフォーマンスカウンターアイテム

VMware のパフォーマンスカウンターからデータを取得するアイテムの完全な一覧については、*VMware monitoring item keys* を参照してください。

VMware イベントの取得頻度はvmware.eventlog の監視間隔に依存しますが、5 秒未満にはできないことに注意してください。

設定　Zabbix サーバーをsourcesからコンパイルする場合、仮想マシン監視を有効にするには、`--with-libcurl --with-libxml2` 設定オプションを指定してコンパイルする必要があります。Zabbix パッケージは、これらのオプションがすでに有効な状態でコンパイルされています。

仮想マシン監視では、以下の Zabbix サーバー設定ファイルパラメータを変更できます。

- StartVMwareCollectors

Note:
監視対象の VMware サービス数より多くのコレクターを有効にすることを推奨します。そうしないと、VMware 設定データの取得（大規模な環境では時間がかかります）によって、VMware パフォーマンスカウンター統計の取得が遅延する可能性があります。

 一般的に、StartVMwareCollectors の値は 2 を下回らず、監視対象の VMware サービス数の 2 倍を超えないようにする必要があります: サービス数 < StartVMwareCollectors < (サービス数 * 2)。たとえば、1 つの VMware サービスを監視する場合は StartVMwareCollectors を 2 に設定し、3 つのサービスを監視する場合は 5 に設定します。

 必要なコレクター数は、VMware 環境の規模、および VMwareFrequency と VMwarePerfFrequency の設定パラメータにも依存することに注意してください。

- VMwareCacheSize
- VMwareFrequency
- VMwarePerfFrequency
- VMwareTimeout

Attention:
データストア容量メトリクスをサポートするには、VMware の `vpxd.stats.maxQueryMetrics` キーの値が少なくとも 64 に設定されていることを確認してください。詳細については、*VMware Knowledge Base* article を参照してください。

ディスカバリ

Zabbix は、低レベルディスカバリルール（例: *vmware.hv.discovery[{$VMWARE.URL}]*）を使用して、VMware ハイパーバイザーおよび仮想マシンを自動的に検出できます。さらに、Zabbix はホストプロトタイプを使用して、検出されたエンティティに対応する実際のホストを自動的に生成できます。詳細については、ホストプロトタイプを参照してください。

Configuration examples

For a basic example of how to set up Zabbix for monitoring VMware using the *VMware FQDN* template, see *Monitor VMware with Zabbix*.

For a more detailed example of how to create a host, a low-level discovery rule, and a host prototype for monitoring VMware, see *Setup example*.

拡張ログ　*vmware collector* プロセスによって収集されたデータは、デバッグレベル 5 を使用して詳細なデバッグのためにログへ記録できます。デバッグレベルは、サーバー およびプロキシ の設定ファイルで設定するか、ランタイム制御オプション `-R log_level_increase="vmware collector,N"` を使用して設定できます。ここで”N” はプロセス番号です。

たとえば、すべての *vmware collector* プロセスのデバッグレベルを 4 から 5 に上げるには、次のコマンドを実行します。

```
zabbix_server -R log_level_increase="vmware collector"
```

2 番目の *vmware collector* プロセスのデバッグレベルを 4 から 5 に上げるには、次のコマンドを実行します。

```
zabbix_server -R log_level_increase="vmware collector,2"
```

VMware collector データの拡張ログが不要になった場合は、`-R log_level_decrease` コマンドを実行して、デバッグレベルをデフォルト (3) に下げることを推奨します。

トラブルシューティング

- メトリクスを利用できない場合は、最近の VMware vSphere バージョンでそれらがデフォルトで利用不可または無効になっていないか、あるいはパフォーマンスメトリクスのデータベースクエリに何らかの制限が設定されていないかを確認してください。
  詳細については、ZBX-12094を参照してください。
- `config.vpxd.stats.maxQueryMetrics` が無効である、または許可されている最大文字数を超えているというエラーが表示される場合は、vCenter Server の設定に `config.vpxd.stats.maxQueryMetrics` パラメータを追加してください。
  このパラメータの値は、VMware の *web.xml* ファイル内の `maxQuerysize` の値と同じである必要があります。
  詳細については、*VMware Knowledge Base* の記事を参照してください。
- Zabbix インストール環境でメモリを過剰に使用している疑いがある場合は、tcmalloc を使用した過剰なメモリ使用量のプロファイリングを参照してください。

1 VMware 監視アイテムキー

概要　このページでは、VMware 環境 の監視に使用できるシンプルチェックの詳細を説明します。
メトリクスは監視対象ごとにグループ化されています。

サポートされているアイテムキー　アイテムキーは、パラメータおよび追加情報を省略して一覧表示されています。完全な詳細を表示するには、アイテムキーをクリックしてください。

アイテムキー	説明	アイテムグループ
vmware.eventlog	VMware イベントログ。	一般サービス
vmware.fullname	VMware サービスの完全名。	
vmware.version	VMware サービスのバージョン。	
vmware.cl.perfcounter	VMware クラスターパフォーマンスカウンターメトリクス。	クラスター
vmware.cluster.alarms.get	VMware クラスターのアラームデータ。	
vmware.cluster.discovery	VMware クラスターのディスカバリ。	
vmware.cluster.property	VMware クラスターのプロパティ。	
vmware.cluster.status	VMware クラスターのステータス。	
vmware.cluster.tags.get	VMware クラスターのタグ配列。	
vmware.datastore.alarms.get	VMware データストアのアラームデータ。	データストア
vmware.datastore.discovery	VMware データストアのディスカバリ。	
vmware.datastore.hv.list	データストアのハイパーバイザー一覧。	
vmware.datastore.perfcounter	VMware データストアのパフォーマンスカウンター値。	
vmware.datastore.property	VMware データストアのプロパティ。	
vmware.datastore.read	データストアからの読み取り操作にかかる時間。	
vmware.datastore.size	VMware データストアの容量（バイトまたは全体に対する割合）。	
vmware.datastore.tags.get	VMware データストアのタグ配列。	
vmware.datastore.write	データストアへの書き込み操作にかかる時間。	
vmware.dc.alarms.get	VMware データセンターのアラームデータ。	データセンター
vmware.dc.discovery	VMware データセンターのディスカバリ。	
vmware.dc.tags.get	VMware データセンターのタグ配列。	
vmware.dvswitch.discovery	VMware vSphere Distributed Switch のディスカバリ。	vSphere Distributed Switch
vmware.dvswitch.fetchports	VMware vSphere Distributed Switch のポートデータ。	
vmware.hv.alarms.get	VMware ハイパーバイザーのアラームデータ。	ハイパーバイザー
vmware.hv.cluster.name	VMware ハイパーバイザーのクラスター名。	

アイテムキー	説明	アイテムグループ
vmware.hv.connectionstate	VMware ハイパーバイザーの接続状態。	
vmware.hv.cpu.usage	VMware ハイパーバイザーのプロセッサー使用率（Hz）。	
vmware.hv.cpu.usage.perf	VMware ハイパーバイザーの、期間中の割合としてのプロセッサー使用率。	
vmware.hv.cpu.utilization	VMware ハイパーバイザーの、期間中の割合としてのプロセッサー使用率。電源管理または HT に依存します。	
vmware.hv.datacenter.name	VMware ハイパーバイザーのデータセンター名。	
vmware.hv.datastore.discovery	VMware ハイパーバイザーデータストアのディスカバリ。	
vmware.hv.datastore.list	VMware ハイパーバイザーデータストアの一覧。	
vmware.hv.datastore.multipath	利用可能なデータストアパスの数。	
vmware.hv.datastore.read	データストアからの読み取り操作にかかる平均時間。	
vmware.hv.datastore.size	VMware データストアの容量（バイトまたは全体に対する割合）。	
vmware.hv.datastore.write	データストアへの書き込み操作にかかる平均時間。	
vmware.hv.discovery	VMware ハイパーバイザーのディスカバリ。	
vmware.hv.diskinfo.get	VMware ハイパーバイザーのディスクデータ。	
vmware.hv.fullname	バージョン情報を含む完全な製品名。	
vmware.hv.hw.cpu.freq	VMware ハイパーバイザーのプロセッサー周波数。	
vmware.hv.hw.cpu.model	VMware ハイパーバイザーのプロセッサーモデル。	
vmware.hv.hw.cpu.num	VMware ハイパーバイザー上のプロセッサーコア数。	
vmware.hv.hw.cpu.threads	VMware ハイパーバイザー上のプロセッサースレッド数。	
vmware.hv.hw.memory	VMware ハイパーバイザーの総メモリサイズ。	
vmware.hv.hw.model	VMware ハイパーバイザーのモデル。	
vmware.hv.hw.sensors.get	VMware ハイパーバイザーのハードウェアセンサー値。	
vmware.hv.hw.serialnumber	VMware ハイパーバイザーのシリアル番号。	
vmware.hv.hw.uuid	VMware ハイパーバイザーの BIOS UUID。	
vmware.hv.hw.vendor	VMware ハイパーバイザーのベンダー名。	
vmware.hv.maintenance	VMware ハイパーバイザーのメンテナンス状態。	
vmware.hv.memory.size.ballooned	VMware ハイパーバイザーのバルーニングされたメモリサイズ。	
vmware.hv.memory.used	VMware ハイパーバイザーの使用済みメモリサイズ。	
vmware.hv.net.if.discovery	VMware ハイパーバイザーネットワークインターフェースのディスカバリ。	
vmware.hv.network.in	VMware ハイパーバイザーのネットワーク入力統計。	
vmware.hv.network.linkspeed	VMware ハイパーバイザーのネットワークインターフェース速度。	
vmware.hv.network.out	VMware ハイパーバイザーのネットワーク出力統計。	
vmware.hv.perfcounter	VMware ハイパーバイザーのパフォーマンスカウンター値。	
vmware.hv.property	VMware ハイパーバイザーのプロパティ。	
vmware.hv.power	VMware ハイパーバイザーの消費電力。	
vmware.hv.sensor.health.state	VMware ハイパーバイザーのヘルス状態ロールアップセンサー。	
vmware.hv.sensors.get	VMware ハイパーバイザーの HW ベンダー状態センサー。	
vmware.hv.status	VMware ハイパーバイザーのステータス。	
vmware.hv.tags.get	VMware ハイパーバイザーのタグ配列。	
vmware.hv.uptime	VMware ハイパーバイザーの稼働時間。	
vmware.hv.version	VMware ハイパーバイザーのバージョン。	
vmware.hv.vm.num	VMware ハイパーバイザー上の仮想マシン数。	
vmware.rp.cpu.usage	VMware リソースプール上の、期間中のヘルツ単位の CPU 使用率。	リソースプール
vmware.rp.memory	VMware リソースプールのメモリメトリクス。	
vmware.alarms.get	VMware Virtual Center のアラームデータ。	Virtual center
vmware.vm.alarms.get	VMware 仮想マシンのアラームデータ。	仮想マシン
vmware.vm.attribute	VMware 仮想マシンのカスタム属性値。	
vmware.vm.cluster.name	VMware 仮想マシン名。	
vmware.vm.consolidationneeded	VMware 仮想マシンのディスクで統合が必要です。	
vmware.vm.cpu.latency	物理 CPU へのアクセス競合により、仮想マシンを実行できなかった時間の割合。	
vmware.vm.cpu.num	VMware 仮想マシン上のプロセッサー数。	
vmware.vm.cpu.readiness	仮想マシンの実行準備はできていたものの、物理 CPU 上で実行するようスケジュールされなかった時間の割合。	
vmware.vm.cpu.ready	仮想マシンの実行準備はできていたものの、物理 CPU 上で実行するようスケジュールされなかった時間。	
vmware.vm.cpu.swapwait	スワップイン待機に費やされた CPU 時間の割合。	
vmware.vm.cpu.usage	VMware 仮想マシンのプロセッサー使用率（Hz）。	
vmware.vm.cpu.usage.perf	VMware 仮想マシンの、期間中の割合としてのプロセッサー使用率。	
vmware.vm.datacenter.name	VMware 仮想マシンのデータセンター名。	

アイテムキー	説明	アイテムグループ
vmware.vm.discovery	VMware 仮想マシンのディスカバリ。	
vmware.vm.guest.memory.size.swapped	スワップ領域にスワップアウトされたゲスト物理メモリ量。	
vmware.vm.guest.osuptime	前回のオペレーティングシステム起動以降の経過時間の合計。	
vmware.vm.hv.maintenance	VMware 仮想マシンのハイパーバイザーメンテナンス状態。	
vmware.vm.hv.name	VMware 仮想マシンのハイパーバイザー名。	
vmware.vm.memory.size	VMware 仮想マシンの総メモリサイズ。	
vmware.vm.memory.size.ballooned	VMware 仮想マシンのバルーニングされたメモリサイズ。	
vmware.vm.memory.size.compressed	VMware 仮想マシンの圧縮メモリサイズ。	
vmware.vm.memory.size.consumed	ゲスト物理メモリページをバックアップするために消費されたホスト物理メモリ量。	
vmware.vm.memory.size.private	VMware 仮想マシンのプライベートメモリサイズ。	
vmware.vm.memory.size.shared	VMware 仮想マシンの共有メモリサイズ。	
vmware.vm.memory.size.swapped	VMware 仮想マシンのスワップ済みメモリサイズ。	
vmware.vm.memory.size.usage.guest	VMware 仮想マシンのゲストメモリ使用量。	
vmware.vm.memory.size.usage.host	VMware 仮想マシンのホストメモリ使用量。	
vmware.vm.memory.usage	消費されたホスト物理メモリの割合。	
vmware.vm.net.if.discovery	VMware 仮想マシンネットワークインターフェースのディスカバリ。	
vmware.vm.net.if.in	VMware 仮想マシンネットワークインターフェースの入力統計。	
vmware.vm.net.if.out	VMware 仮想マシンネットワークインターフェースの出力統計。	
vmware.vm.net.if.usage	VMware 仮想マシンの、期間中のネットワーク使用率。	
vmware.vm.perfcounter	VMware 仮想マシンのパフォーマンスカウンター値。	
vmware.vm.powerstate	VMware 仮想マシンの電源状態。	
vmware.vm.property	VMware 仮想マシンのプロパティ。	
vmware.vm.snapshot.get	VMware 仮想マシンのスナップショット状態。	
vmware.vm.state	VMware 仮想マシンの状態。	
vmware.vm.storage.committed	VMware 仮想マシンのコミット済みストレージ容量。	
vmware.vm.storage.readoio	収集間隔中に仮想ディスクに対して未処理となっていた読み取り要求の平均数。	
vmware.vm.storage.totalreadlatency	仮想ディスクからの読み取りにかかる平均時間。	
vmware.vm.storage.totalwritelatency	仮想ディスクへの書き込みにかかる平均時間。	
vmware.vm.storage.uncommitted	VMware 仮想マシンの未コミットストレージ容量。	
vmware.vm.storage.unshared	VMware 仮想マシンの非共有ストレージ容量。	
vmware.vm.storage.writeoio	収集間隔中に仮想ディスクに対して未処理となっていた書き込み要求の平均数。	
vmware.vm.tags.get	VMware 仮想マシンのタグ配列。	
vmware.vm.tools	VMware 仮想マシンのゲストツールの状態またはバージョン。	
vmware.vm.uptime	VMware 仮想マシンの稼働時間。	
vmware.vm.vfs.dev.discovery	VMware 仮想マシンディスクデバイスのディスカバリ。	
vmware.vm.vfs.dev.read	VMware 仮想マシンディスクデバイスの読み取り統計。	
vmware.vm.vfs.dev.write	VMware 仮想マシンディスクデバイスの書き込み統計。	
vmware.vm.vfs.fs.discovery	VMware 仮想マシンファイルシステムのディスカバリ。	
vmware.vm.vfs.fs.size	VMware 仮想マシンファイルシステムの統計。	

アイテムキーの詳細　山かっこなしのパラメータは必須です。山かっこ **< >** で示されたパラメータは任意です。

vmware.eventlog[url,<mode>,<severity>]

 VMware イベントログ。
 戻り値: ログ。

パラメータ:

- **url** - VMware サービスの URL。
- **mode** - *all*（デフォルト）または *skip* - 古いデータの処理をスキップします。
- **severity** - 重大度でフィルタリングします: *error*、*warning*、*info*、または *user*。複数の重大度をカンマ区切りのリストで指定する場合、このパラメータは引用符で囲む必要があります（例: `"error,warning,info,user"`）。デフォルトでは無効です。

コメント:

- URL ごとに `vmware.eventlog` のアイテムキーは 1 つだけでなければなりません。
- VMware イベントログレコードのフィルタリングの例も参照してください。

vmware.fullnameurl

 VMware サービスの完全名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。

vmware.versionurl

 VMware サービスのバージョン。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。

vmware.cl.perfcounter[url,id,path,<instance>]

 VMware クラスタのパフォーマンスカウンターメトリクスです。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **id** - VMware クラスタ ID。`id` は `vmware.cluster.discovery[]` から {#CLUSTER.ID} として取得できます。
- **path** - パフォーマンスカウンターのパス[1]。
- **instance** - パフォーマンスカウンターのインスタンス。

vmware.cluster.alarms.get[url,id]

 VMware クラスタのアラームデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **id** - VMware クラスタ ID。

vmware.cluster.discoveryurl

 VMware クラスターのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.cluster.property[url,id,prop]

 VMware クラスターのプロパティです。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **id** - VMware クラスター ID。
- **prop** - VMware SDK で定義されている VM オブジェクトのプロパティ名であるプロパティパス。

例:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 VMware クラスターのステータス。
 戻り値: *0* - グレー; *1* - 緑; *2* - 黄; *3* - 赤。

パラメータ:

- **url** - VMware サービスの URL;
- **name** - VMware クラスター名。

vmware.cluster.tags.get[url,id]

 VMware クラスタタグの配列。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービス URL。
- **id** - VMware クラスタ ID。

このアイテムは、vSphere 6.5 以降で動作します。

vmware.datastore.alarms.get[url,uuid]

 VMware データストアのアラームデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware データストアのグローバル一意識別子。

vmware.datastore.discovery[url,<filter_uuid>]

 VMware データストアのディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービス URL。
- **filter_uuid** - 単一の HV または VM の一意な識別子（デフォルト: 空）。指定した場合、リンクされたデータストアのみがディスカバリされます。

vmware.datastore.hv.list[url,datastore]

 データストアのハイパーバイザーの一覧。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL。
- **datastore** - データストアの UUID または名前。

出力例:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 VMware データストアのパフォーマンスカウンター値です。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware データストアのグローバル一意識別子;
- **path** - パフォーマンスカウンターのパス[1];
- **instance** - パフォーマンスカウンターのインスタンス。集計値には空のインスタンスを使用します（デフォルト）。`instance` は、{#DATASTORE.EXTENT} 配列の一部として `vmware.datastore.discovery[]` から取得できます。

vmware.datastore.property[url,uuid,prop]

 VMware データストアのプロパティです。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware データストアのグローバル一意識別子。
- **prop** - プロパティパス。

vmware.datastore.read[url,datastore,<mode>]

 データストアからの読み取り操作に要する時間（ミリ秒）。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **datastore** - データストアの UUID または名前。
- **mode** - *latency*（平均値、デフォルト）または *maxlatency*（最大値）。

vmware.datastore.size[url,datastore,<mode>]

 VMware データストアの容量。単位はバイト、または総容量に対する割合（パーセンテージ）です。
 戻り値: バイトの場合は *Integer*、パーセンテージの場合は *Float*。

パラメータ:

- **url** - VMware サービスの URL;
- **datastore** - データストアの UUID または名前;
- **mode** - 指定可能な値: *total*（デフォルト）、*free*、*pfree*（空き容量の割合）、*uncommitted*。

vmware.datastore.tags.get[url,uuid]

 VMware データストアタグの配列。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware データストアのグローバル一意識別子。

このアイテムは、vSphere 6.5 以降で動作します。

vmware.datastore.write[url,datastore,<mode>]

 データストアへの書き込み操作に要する時間（ミリ秒）。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **datastore** - データストアの UUID または名前;
- **mode** - *latency*（平均値、デフォルト）または *maxlatency*（最大値）。

vmware.dc.alarms.get[url,id]

 VMware データセンターのアラームデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **id** - VMware データセンター ID。

vmware.dc.discoveryurl

 VMware データセンターのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。

vmware.dc.tags.get[url,id]

 VMware データセンターのタグ配列。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービス URL。
- **id** - VMware データセンター ID。

このアイテムは vSphere 6.5 以降で動作します。

vmware.dvswitch.discoveryurl

 VMware vSphere Distributed Switch のディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 VMware vSphere Distributed Switch のポートデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware DVSwitch のグローバル一意識別子。
- **filter** - ポートを選択するための条件をカンマ区切りで指定した単一の文字列。
- **mode** - *state*（”config” XML ノードを除くすべての XML、デフォルト）または *full*。

filter パラメータは、VMware データオブジェクト DistributedVirtualSwitchPortCriteria で利用可能な条件をサポートします。

例:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 VMware ハイパーバイザーのアラームデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.cluster.name[url,uuid]

 VMware ハイパーバイザークラスター名。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービス URL。

- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.connectionstate[url,uuid]

 VMware ハイパーバイザーの接続状態です。
 戻り値: 文字列: *connected*、*disconnected*、または *notResponding*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.cpu.usage[url,uuid]

 VMware ハイパーバイザーのプロセッサ使用量（Hz）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.cpu.usage.perf[url,uuid]

 VMware ハイパーバイザーのプロセッサ使用率（期間中のパーセンテージ）です。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.cpu.utilization[url,uuid]

 VMware ハイパーバイザーのプロセッサ使用率を、期間中のパーセンテージで示します。電源管理または HT に依存します。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.datacenter.name[url,uuid]

 VMware ハイパーバイザーのデータセンター名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.datastore.discovery[url,uuid]

 VMware ハイパーバイザーのデータストアのディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.datastore.list[url,uuid]

 VMware ハイパーバイザーのデータストアの一覧。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

出力例:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 利用可能なデータストアパスの数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

- **datastore** - データストアの UUID または名前。
- **partitionid** - `vmware.hv.datastore.discovery` の物理デバイスの内部 ID。

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 データストアからの読み取り操作に要する平均時間（ミリ秒）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。
- **datastore** - データストアの UUID または名前。
- **mode** - *latency*（デフォルト）。

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 VMware データストアの容量を、バイト単位または総容量に対する割合で返します。
 戻り値: バイトの場合は *Integer*、割合の場合は *Float*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。
- **datastore** - データストアの UUID または名前。
- **mode** - 指定可能な値: *total* (デフォルト)、*free*、*pfree* (空き容量の割合)、*uncommitted*。

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 データストアへの書き込み操作に要する平均時間（ミリ秒）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。
- **datastore** - データストアの UUID または名前。
- **mode** - *latency*（デフォルト）。

vmware.hv.discoveryurl

 VMware ハイパーバイザーのディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。

vmware.hv.diskinfo.get[url,uuid]

 VMware ハイパーバイザーのディスクデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.fullname[url,uuid]

 バージョン情報を含む完全な製品名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.cpu.freq[url,uuid]

 VMware ハイパーバイザーのプロセッサ周波数（Hz）。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.cpu.model[url,uuid]

 VMware ハイパーバイザーのプロセッサーモデル。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。

- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.cpu.num[url,uuid]

 VMware ハイパーバイザー上のプロセッサコア数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.cpu.threads[url,uuid]

 VMware ハイパーバイザー上のプロセッサスレッド数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.memory[url,uuid]

 VMware ハイパーバイザーの合計メモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.model[url,uuid]

 VMware ハイパーバイザーのモデル。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.sensors.get[url,uuid]

 VMware ハイパーバイザーのハードウェアセンサーの値。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.hw.serialnumber[url,uuid]

 VMware ハイパーバイザーのシリアル番号。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは、vSphere API 6.7 以降で動作します。

vmware.hv.hw.uuid[url,uuid]

 VMware ハイパーバイザーの BIOS UUID。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービス URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは、vSphere API 6.7 以降で動作します。

vmware.hv.hw.vendor[url,uuid]

 VMware ハイパーバイザーのベンダー名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは、vSphere API 6.7 以降で動作します。

vmware.hv.maintenance[url,uuid]

 VMware ハイパーバイザーのメンテナンス状態。
 戻り値: *0* - メンテナンス中ではない; *1* - メンテナンス中。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.memory.size.ballooned[url,uuid]

 VMware ハイパーバイザーのバルーニングされたメモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.memory.used[url,uuid]

 VMware ハイパーバイザーで使用中のメモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.net.if.discovery[url,uuid]

 VMware ハイパーバイザーのネットワークインターフェースのディスカバリ。
 戻り値: *JSON object*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.network.in[url,uuid,<mode>]

 VMware ハイパーバイザーのネットワーク入力統計情報（1 秒あたりのバイト数）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。
- **mode** - *bps*（デフォルト）、*packets*、*dropped*、*errors*、*broadcast*。

vmware.hv.network.linkspeed[url,uuid,ifname]

 VMware ハイパーバイザーのネットワークインターフェース速度。
 戻り値: *Integer*。ネットワークインターフェースがダウンしている場合は *0* を返し、それ以外の場合はインターフェースの速度値を返します。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子;
- **ifname** - インターフェース名。

vmware.hv.network.out[url,uuid,<mode>]

 VMware ハイパーバイザーのネットワーク送信統計情報（1 秒あたりのバイト数）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。
- **mode** - *bps*（デフォルト）、*packets*、*dropped*、*errors*、*broadcast*。

vmware.hv.perfcounter[url,uuid,path,<instance>]

 VMware ハイパーバイザーのパフォーマンスカウンター値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子;
- **path** - パフォーマンスカウンターのパス [1];
- **instance** - パフォーマンスカウンターのインスタンス。集計値には空のインスタンスを使用します（デフォルト）。

vmware.hv.property[url,uuid,prop]

 VMware ハイパーバイザーのプロパティ。
 戻り値: 文字列。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子;
- **prop** - プロパティパス。

vmware.hv.power[url,uuid,<max>]

 VMware ハイパーバイザーの消費電力（W）。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。
- **max** - 許容される最大消費電力。

vmware.hv.sensor.health.state[url,uuid]

 VMware ハイパーバイザーのヘルス状態ロールアップセンサーです。
 戻り値: *Integer*: *0* - グレー、*1* - グリーン、*2* - イエロー、*3* - レッド。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子

このアイテムは、VMware が *VMware Rollup Health State* センサーを非推奨にしたため、VMware vSphere 6.5 以降では動作しない可能性があることに注意してください。

vmware.hv.sensors.get[url,uuid]

 VMware ハイパーバイザーの HW ベンダー状態センサーです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.status[url,uuid]

 VMware ハイパーバイザーのステータス。
 戻り値: *Integer*: *0* - グレー; *1* - 緑; *2* - 黄; *3* - 赤。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは、ホストシステム全体のステータスプロパティを使用します。

vmware.hv.tags.get[url,uuid]

 VMware ハイパーバイザーのタグ配列。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは、vSphere 6.5 以降で動作します。

vmware.hv.uptime[url,uuid]

 VMware ハイパーバイザーの稼働時間（秒）です。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

このアイテムは、ホストシステム全体のステータスプロパティを使用します。

vmware.hv.version[url,uuid]

 VMware ハイパーバイザーのバージョン。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.hv.vm.num[url,uuid]

 VMware ハイパーバイザー上の仮想マシン数。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware ハイパーバイザーのグローバル一意識別子。

vmware.rp.cpu.usage[url,rpid]

 VMware リソースプールの期間中の CPU 使用量（ヘルツ単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **rpid** - VMware リソースプール ID。

vmware.rp.memory[url,rpid,<mode>]

 VMware リソースプールのメモリメトリクス。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL;
- **rpid** - VMware リソースプール ID;
- **mode** - 指定可能な値:
consumed (デフォルト) - ゲスト物理メモリページをバックアップするために消費されるホスト物理メモリ量
 ballooned - ゲスト内のバルーンドライバーによって仮想マシンから回収されたゲスト物理メモリ量
overhead - 仮想マシンの実行のために ESXi データ構造によって消費されるホスト物理メモリ量

vmware.alarms.geturl

 VMware vCenter のアラームデータ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。

vmware.vm.alarms.get[url,uuid]

 VMware 仮想マシンのアラームデータです。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.attribute[url,uuid,name]

 VMware 仮想マシンのカスタム属性値。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービス URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子;
- **name** - カスタム属性名。

vmware.vm.cluster.name[url,uuid]

 VMware 仮想マシン名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービス URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.consolidationneeded[url,uuid]

 VMware 仮想マシンのディスクは統合が必要です。
 戻り値: *String*: *true* - 統合が必要です; *false* - 統合は不要です。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.cpu.latency[url,uuid]

 仮想マシンが物理 CPU へのアクセス競合により実行できない時間の割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.cpu.num[url,uuid]

 VMware 仮想マシン上のプロセッサ数。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.cpu.readiness[url,uuid,<instance>]

 仮想マシンが実行可能な状態であったにもかかわらず、物理 CPU 上での実行にスケジュールされなかった時間の割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - CPU インスタンス。

vmware.vm.cpu.ready[url,uuid]

 仮想マシンは実行可能な状態であったものの、物理 CPU 上で実行するようスケジュールできなかった時間（ミリ秒）です。CPU ready time は、ホスト上の仮想マシン数とそれらの CPU 負荷（%）に依存します。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 スワップインの待機に費やされた CPU 時間の割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - CPU インスタンス。

vmware.vm.cpu.usage[url,uuid]

 VMware 仮想マシンのプロセッサ使用率（Hz）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.cpu.usage.perf[url,uuid]

 VMware 仮想マシンのプロセッサ使用率を、対象期間中のパーセンテージで示します。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.datacenter.name[url,uuid]

 VMware 仮想マシンのデータセンター名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.discoveryurl

 VMware 仮想マシンのディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。

vmware.vm.guest.memory.size.swapped[url,uuid]

 スワップ領域にスワップアウトされたゲスト物理メモリの量（KB）。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.guest.osuptime[url,uuid]

 最後にオペレーティングシステムが起動してからの経過時間の合計（秒単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.hv.maintenance[url,uuid]

 VMware 仮想マシンのハイパーバイザーのメンテナンス状態。
 戻り値: *0* - メンテナンス中ではない; *1* - メンテナンス中。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.hv.name[url,uuid]

 VMware 仮想マシンのハイパーバイザー名。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size[url,uuid]

 VMware 仮想マシンの合計メモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.ballooned[url,uuid]

 VMware 仮想マシンのバルーニングされたメモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.compressed[url,uuid]

 VMware 仮想マシンの圧縮メモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.consumed[url,uuid]

 ゲスト物理メモリページをバックアップするために消費されるホストの物理メモリ量（KB）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.private[url,uuid]

 VMware 仮想マシンのプライベートメモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.shared[url,uuid]

 VMware 仮想マシンの共有メモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.swapped[url,uuid]

 VMware 仮想マシンのスワップ済みメモリサイズ（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.usage.guest[url,uuid]

 VMware 仮想マシンのゲストメモリ使用量（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.size.usage.host[url,uuid]

 VMware 仮想マシンのホストメモリ使用量（バイト単位）。
 戻り値: 整数。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.memory.usage[url,uuid]

 消費されたホスト物理メモリの割合。
 戻り値: *Float*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.net.if.discovery[url,uuid]

 VMware 仮想マシンのネットワークインターフェースのディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 VMware 仮想マシンのネットワークインターフェースの受信統計情報（1 秒あたりのバイト数/パケット数）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子;
- **instance** - ネットワークインターフェースのインスタンス;
- **mode** - *bps*（デフォルト）または *pps* - 1 秒あたりのバイト数またはパケット数。

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 VMware 仮想マシンのネットワークインターフェースの送信統計情報（1 秒あたりのバイト数/パケット数）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子;
- **instance** - ネットワークインターフェースのインスタンス;
- **mode** - *bps*（デフォルト）または *pps* - 1 秒あたりのバイト数またはパケット数。

vmware.vm.net.if.usage[url,uuid,<instance>]

 VMware 仮想マシンのネットワーク使用率（送信レートと受信レートの合計）を、指定された間隔において示します（KBps）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - ネットワークインターフェースのインスタンス。

vmware.vm.perfcounter[url,uuid,path,<instance>]

 VMware 仮想マシンのパフォーマンスカウンター値。
 戻り値: 整数 [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子;
- **path** - パフォーマンスカウンターのパス [1];
- **instance** - パフォーマンスカウンターのインスタンス。集計値には空のインスタンスを使用します（デフォルト）。

vmware.vm.powerstate[url,uuid]

 VMware 仮想マシンの電源状態。
 戻り値: *0* - poweredOff; *1* - poweredOn; *2* - suspended。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.property[url,uuid,prop]

 VMware 仮想マシンのプロパティです。
 戻り値: *String*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **prop** - プロパティパス。

vmware.vm.snapshot.get[url,uuid]

 VMware 仮想マシンのスナップショット状態。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.state[url,uuid]

 VMware 仮想マシンの状態。
 戻り値: 文字列: *notRunning*、*resetting*、*running*、*shuttingDown*、*standby*、または *unknown*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.storage.committed[url,uuid]

 VMware 仮想マシンのコミット済みストレージ容量（バイト）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.storage.readoio[url,uuid,instance]

 収集間隔中における仮想ディスクへの未処理の読み取り要求の平均数。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - ディスクデバイスのインスタンス。

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 仮想ディスクからの読み取りにかかる平均時間（ミリ秒）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - ディスクデバイスのインスタンス。

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 仮想ディスクへの書き込みにかかる平均時間（ミリ秒）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - ディスクデバイスのインスタンス。

vmware.vm.storage.uncommitted[url,uuid]

 VMware 仮想マシンの未コミットストレージ容量（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.storage.unshared[url,uuid]

 VMware 仮想マシンの非共有ストレージ容量（バイト単位）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.storage.writeoio[url,uuid,instance]

 収集間隔中の仮想ディスクに対する未処理の書き込み要求の平均数。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - ディスクデバイスのインスタンス。

vmware.vm.tags.get[url,uuid]

 VMware 仮想マシンのタグ配列。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

このアイテムは、vSphere 6.5 以降で動作します。

vmware.vm.tools[url,uuid,<mode>]

 VMware 仮想マシンのゲストツールの状態。
 戻り値: 文字列。
 status モード: *guestToolsExecutingScripts* - VMware Tools は起動中です。*guestToolsNotRunning* - VMware Tools は実行されていません。*guestToolsRunning* - VMware Tools は実行中です。
 version モード: バージョン。

パラメータ:

- **url** - VMware サービスの URL
- **uuid** - VMware 仮想マシンのグローバル一意識別子
- **mode** - *version*、*status* (デフォルト)。

vmware.vm.uptime[url,uuid]

 VMware 仮想マシンの稼働時間（秒）。
 戻り値: *Integer*。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.vfs.dev.discovery[url,uuid]

 VMware 仮想マシンのディスクデバイスのディスカバリ。
 戻り値: JSON オブジェクト。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 VMware 仮想マシンのディスクデバイスの読み取り統計情報（1 秒あたりのバイト数/操作数）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。
- **instance** - ディスクデバイスのインスタンス。
- **mode** - *bps*（デフォルト）または *ops* - 1 秒あたりのバイト数または操作数。

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 VMware 仮想マシンのディスクデバイス書き込み統計（1 秒あたりのバイト数/操作数）。
 戻り値: *Integer* [2]。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子;
- **instance** - ディスクデバイスのインスタンス;
- **mode** - *bps*（デフォルト）または *ops* - 1 秒あたりのバイト数または操作数。

vmware.vm.vfs.fs.discovery[url,uuid]

 VMware 仮想マシンのファイルシステムのディスカバリ。
 戻り値: JSON object。

パラメータ:

- **url** - VMware サービスの URL。
- **uuid** - VMware 仮想マシンのグローバル一意識別子。

このアイテムを動作させるには、ゲスト仮想マシンに VMware Tools がインストールされている必要があります。

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 VMware 仮想マシンのファイルシステム統計情報（バイト/パーセンテージ）です。
 戻り値: バイトの場合は *Integer*、パーセンテージの場合は *Float*。

パラメータ:

- **url** - VMware サービスの URL;
- **uuid** - VMware 仮想マシンのグローバル一意識別子;
- **fsname** - ファイルシステム名;
- **mode** - *total*、*free*、*used*、*pfree*、または *pused*。

このアイテムを機能させるには、ゲスト仮想マシンに VMware Tools がインストールされている必要があります。

脚注

[1] VMware のカスタムパフォーマンスカウンター名の作成を参照してください。

[2] これらのアイテムの値は VMware パフォーマンスカウンターから取得され、Zabbix VMware キャッシュ内のデータを更新するために、VMwarePerfFrequency parameterが使用されます。

- vmware.cl.perfcounter
- vmware.hv.datastore.read
- vmware.hv.datastore.write
- vmware.hv.network.in
- vmware.hv.network.out

- vmware.hv.perfcounter
- vmware.vm.cpu.ready
- vmware.vm.net.if.in
- vmware.vm.net.if.out
- vmware.vm.perfcounter
- vmware.vm.vfs.dev.read
- vmware.vm.vfs.dev.write

詳細

Zabbix で VMware 環境を監視するための詳細な設定方法は、仮想マシン監視 を参照してください。

2 仮想マシンディスカバリキーフィールド

次の表は、仮想マシン関連のディスカバリキーが返すフィールドの一覧です。

アイテムキー		
説明	フィールド	取得したコンテンツ
vmware.cluster.discovery		
Performs cluster discovery.	{#CLUSTER.ID}	Cluster identifier.
	{#CLUSTER.NAME}	Cluster name.

	”resource_pool”	An array containing resource pool data, including resource group ID, tags array, resource pool path, number of virtual machines.
		
		Array structure:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`

	”tags”	For `"tags"` array structure, see the ”tags” field.
		An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Performs datastore discovery.	{#DATASTORE}	Datastore name.

	{#DATASTORE.EXTENT}	An array containing datastore extent partition ID and instance name.
		
		Array structure:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Datastore type. Value examples: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Datastore identifier.

アイテムキー		
	”tags”	An array containing tags with tag name, description and category. Array structure: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dc.discovery		
Performs datacenter discovery.	{#DATACENTER}	Datacenter name.
	{#DATACENTERID}	Datacenter identifier.
	”tags”	An array containing tags with tag name, description and category. Array structure: [{ "tag":"tag name", "tag_description":"tag description", "category":"tag category" }]
vmware.dvswitch.discovery		
Performs vSphere distributed switches discovery.	{#DVS.NAME}	Switch name.
	{#DVS.UUID}	Switch identifier.
vmware.hv.discovery		
Performs hypervisor discovery.	{#HV.UUID}	Unique hypervisor identifier.
	{#HV.ID}	Hypervisor identifier (HostSystem managed object name).
	{#HV.NAME}	Hypervisor name.
	{#HV.NETNAME}	Hypervisor network host name.
	{#HV.IP}	Hypervisor IP address, might be empty. In case of an HA configuration with multiple net interfaces, the following selection priority for interface is observed: - prefer the IP which shares the IP-subnet with the vCenter IP; - prefer the IP from IP-subnet with default gateway; - prefer the IP from interface with the lowest ID.
	{#CLUSTER.NAME}	Cluster name, might be empty.
	{#DATACENTER.NAME}	Datacenter name.
	{#PARENT.NAME}	Name of container that stores the hypervisor.
	{#PARENT.TYPE}	Type of container in which the hypervisor is stored. The values could be `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, where ”VMware” stands for unknown container type.
	”resource_pool”	An array containing resource pool data, including resource group ID, tags array, resource pool path, number of virtual machines. Array structure: [{ "rpid":"resource group id", "tags":[{}], "rpath":"resource group path", "vm_count":0 }] For `"tags"` array structure, see the ”tags” field.

アイテムキー		

	”tags”	An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.datastore.discovery		
Performs hypervisor datastore discovery. Note that multiple hypervisors can use the same datastore.	{#DATASTORE}	Datastore name.
	{#DATASTORE.TYPE}	Datastore type. Value examples: `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Datastore identifier.
	{#MULTIPATH.COUNT}	Registered number of datastore paths.
	{#MULTIPATH.PARTITION.COUNT}	Number of available disk partitions.

	”datastore_extent”	An array containing datastore extent instance name and partition ID.
		
		Array structure:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	”tags”	An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.net.if.discovery		
Performs hypervisor network interfaces discovery.	{#IFNAME}	Interface name.
	{#IFDRIVER}	Interface driver.
	{#IFDUPLEX}	Interface duplex settings.
	{#IFSPEED}	Interface speed.
	{#IFMAC}	Interface mac address.
vmware.vm.discovery		
Performs virtual machine discovery.	{#VM.UUID}	Unique virtual machine identifier.
	{#VM.ID}	Virtual machine identifier (VirtualMachine managed object name).
	{#VM.NAME}	Virtual machine name.
	{#HV.NAME}	Hypervisor name.
	{#HV.UUID}	Unique hypervisor identifier.
	{#HV.ID}	Hypervisor identifier (HostSystem managed object name).
	{#CLUSTER.NAME}	Cluster name, might be empty.
	{#DATACENTER.NAME}	Datacenter name.
	{#DATASTORE.NAME}	Datastore name.
	{#DATASTORE.UUID}	Datastore identifier.
	{#VM.IP}	Virtual machine IP address, might be empty.
	{#VM.DNS}	Virtual machine DNS name, might be empty.
	{#VM.GUESTFAMILY}	Guest virtual machine OS family, might be empty.
	{#VM.GUESTFULLNAME}	Full guest virtual machine OS name, might be empty.

アイテムキー		
	{#VM.FOLDER}	The chain of virtual machine parent folders, can be used as value for nested groups; folder names are combined with "/". Might be empty.
	{#VM.TOOLS.STATUS}	VMware virtual machine tools state.
	{#VM.POWERSTATE}	VMware virtual machine power state (`poweredOff`, `poweredOn`, or `suspended`).
	{#VM.RPOOL.ID}	Resource pool identifier.
	{#VM.RPOOL.PATH}	Full resource pool path excluding the "root" name "Resources". Folder names are combined with "/".
	{#VM.SNAPSHOT.COUNT}	Number of VM snapshots.

	"tags"	An array containing tags with tag name, description and category.
		
		Array structure:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

	"vm_customattribute"	An array of virtual machine custom attributes (if defined).
		
		Array structure:
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

	"net_if"	An array of virtual machine network interfaces.
		
		Array structure:
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		For the description of returned data, see the "vmware.vm.net.if.discovery" item key.

vmware.vm.net.if.discovery		
Performs virtual machine network interface discovery.	{#IFNAME}	Network interface name.
	{#IFDESC}	Interface description.
	{#IFMAC}	Interface mac address.
	{#IFCONNECTED}	Interface connection status (`false` - disconnected; `true` - connected).
	{#IFTYPE}	Interface type.
	{#IFBACKINGDEVICE}	Name of the backing device.
	{#IFDVSWITCH.UUID}	Unique vSphere Distributed Switch identifier.
	{#IFDVSWITCH.PORTGROUP}	Distributed port group.
	{#IFDVSWITCH.PORT}	vSphere Distributed Switch port.
	"ifip"	An array of interface addresses.
vmware.vm.vfs.dev.discovery		
Performs virtual machine disk device discovery.	{#DISKNAME}	Disk device name.

アイテムキー		
vmware.vm.vfs.fs.discovery Performs virtual machine file system discovery.	{#FSNAME}	File system name.

3 VMware アイテムの **JSON** の例

概要　このセクションでは、さまざまな VMwareアイテムによって返される JSON オブジェクトに関する追加情報を提供します。

vmware.*.alarms.get　**vmware.alarms.get[]**、**vmware.cluster.alarms.get[]**、**vmware.datastore.alarms.get[]**、**vmware.dc.alarms.**
vmware.hv.alarms.get[]、**vmware.vm.alarms.get[]** の各アイテムは、次の構造を持つ JSON オブジェクトを返します（値は例として示しています）。

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get　アイテム　**vmware.cluster.tags.get[]**、**vmware.datastore.tags.get[]**、**vmware.dc.tags.get[]**、**vmware.hv.tags.get[]**、**vmware.vm.tags.get[]** は、次の構造を持つ JSON オブジェクトを返します（値は例として示しています）。

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "OS type カテゴリ用のタグ",
        "category": "OS type"
    },
    {
            "name": "SQL Server",
        "description": "application name カテゴリ用のタグ",
        "category": "application name"
        },
    {
            // その他のタグ
```

```
        }
    ]
}
```

vmware.hv.diskinfo.get　アイテム **vmware.hv.diskinfo.get[]** は、次の構造の JSON オブジェクトを返します（値は例として示しています）：

```
[
  {
    "instance": "mpx.vmhba32:C0:T0:L0",
    "hv_uuid": "8002299e-d7b9-8728-d224-76004bbb6100",
    "datastore_uuid": "",
    "operational_state": [
      "ok"
    ],
    "lun_type": "disk",
    "queue_depth": 1,
    "model": "USB DISK",
    "vendor": "SMI Corp",
    "revision": "1100",
    "serial_number": "CCYYMMDDHHmmSS9S62CK",
    "vsan": {}
  },
  {
    // other instances
  }
]
```

vmware.dvswitch.fetchports.get　アイテム **vmware.dvswitch.fetchports.get[]** は、次の構造を持つ JSON オブジェクトを返します（値は例として示しています）。

```
{
    "FetchDVPortsResponse":
    {
        "returnval": [
            {
                "key": "0",
                "dvsUuid": "50 36 6a 24 25 c0 10 9e-05 4a f6 ea 4e 3d 09 88",
                "portgroupKey": "dvportgroup-2023",
                "proxyHost":
                {
                    "@type": "HostSystem",
                    "#text": "host-2021"
                },
                "connectee":
                {
                    "connectedEntity":
                    {
                        "@type": "HostSystem",
                        "#text": "host-2021"
                    },
                    "nicKey": "vmnic0",
                    "type": "pnic"
                },
                "conflict": "false",
                "state":
                {
                    "runtimeInfo":
                    {
                        "linkUp": "true",
                        "blocked": "false",
                        "vlanIds":
                        {
```

```
                            "start": "0",
                            "end": "4094"
                        },
                        "trunkingMode": "true",
                        "linkPeer": "vmnic0",
                        "macAddress": "00:00:00:00:00:00",
                        "statusDetail": null,
                        "vmDirectPathGen2Active": "false",
                        "vmDirectPathGen2InactiveReasonOther": "portNptIncompatibleConnectee"
                    },
                    "stats":
                    {
                        "packetsInMulticast": "2385470",
                        "packetsOutMulticast": "45",
                        "bytesInMulticast": "309250248",
                        "bytesOutMulticast": "5890",
                        "packetsInUnicast": "155601537",
                        "packetsOutUnicast": "113008658",
                        "bytesInUnicast": "121609489384",
                        "bytesOutUnicast": "47240279759",
                        "packetsInBroadcast": "1040420",
                        "packetsOutBroadcast": "7051",
                        "bytesInBroadcast": "77339771",
                        "bytesOutBroadcast": "430392",
                        "packetsInDropped": "0",
                        "packetsOutDropped": "0",
                        "packetsInException": "0",
                        "packetsOutException": "0"
                    }
                },
                "connectionCookie": "1702765133",
                "lastStatusChange": "2022-03-25T14:01:11Z",
                "hostLocalPort": "false"
            },
            {
                //その他のキー
            }
        ]
    }
}
```

vmware.hv.hw.sensors.get アイテム **vmware.hv.hw.sensors.get[]** は、次の構造の JSON オブジェクトを返します（値は例として示しています）。

```
{
    "val":
        {
            "@type": "HostHardwareStatusInfo",
            "storageStatusInfo": [
                {
                    "name": "Intel Corporation HD Graphics 630 #2",
                "status":
                        {
                            "label": "Unknown",
                    "summary": "Cannot report on the current status of the physical element",
                    "key": "Unknown"
                }
            },
            {
                    "name": "Intel Corporation 200 Series/Z370 Chipset Family USB 3.0 xHCI Controller #20"
                "status":
                        {
                     "label": "Unknown",
```

```
                    "summary": "Cannot report on the current status of the physical element",
                    "key": "Unknown"
                }
            },
            {
                    // その他の hv hw センサー
                }
        ]
        }
}
```

vmware.hv.sensors.get　アイテム **vmware.hv.sensors.get[]** は、次の構造の JSON オブジェクトを返します（値は例として示しています）：

```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10500",
                "unitModifier": "-2",
                "baseUnits": "Watts",
                "sensorType": "other"
            },
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "Power Supply 1 PS 1 Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                    "summary": "Sensor is operating under normal conditions",
                    "key": "green"
                },
                    "currentReading": "10000",
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                    // その他の hv センサー
                }
        ]
        }
}
```

vmware.vm.snapshot.get　スナップショットが存在する場合、アイテム **vmware.snapshot.get[]** は次の構造の JSON オブジェクトを返します（値は例として示しています）。

```
{
  "snapshot": [
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:16:39 AM",
      "description": "Descr 1",
      "createtime": "2022-04-01T06:16:51.761Z",
      "size": 5755795171,
```

```
      "uniquesize": 5755795171
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:18:21 AM",
      "description": "Descr 2",
      "createtime": "2022-04-01T06:18:29.164999Z",
      "size": 118650595,
      "uniquesize": 118650595
    },
    {
      "name": "VM Snapshot 4%2f1%2f2022, 9:37:29 AM",
      "description": "Descr 3",
      "createtime": "2022-04-01T06:37:53.534999Z",
      "size": 62935016,
      "uniquesize": 62935016
    }
  ],
  "count": 3,
  "latestdate": "2022-04-01T06:37:53.534999Z",
  "latestage": 22729203,
  "oldestdate": "2022-04-01T06:16:51.761Z",
  "oldestage": 22730465,
  "size": 5937380782,
  "uniquesize": 5937380782
}
```


スナップショットが存在しない場合、アイテム **vmware.snapshot.get[]** は空の値を持つ JSON オブジェクトを返します。


```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```


4 VMware 監視設定例

概要

以下の例では、VMware 仮想マシンを監視するために Zabbix を設定する方法について説明します。これには以下が含まれます。

- VMware 環境を表すホストを作成すること。
- VMware 環境内の仮想マシンを検出するローレベルディスカバリルールを作成すること。
- ホストのプロトタイプを作成すること。これに基づいて、Zabbix はローレベルディスカバリルールによって検出された仮想マシン用の実際のホストを生成します。

前提条件

> **Note:**
> この例では、VMware の設定については扱いません。VMware はすでに設定済みであることを前提としています。

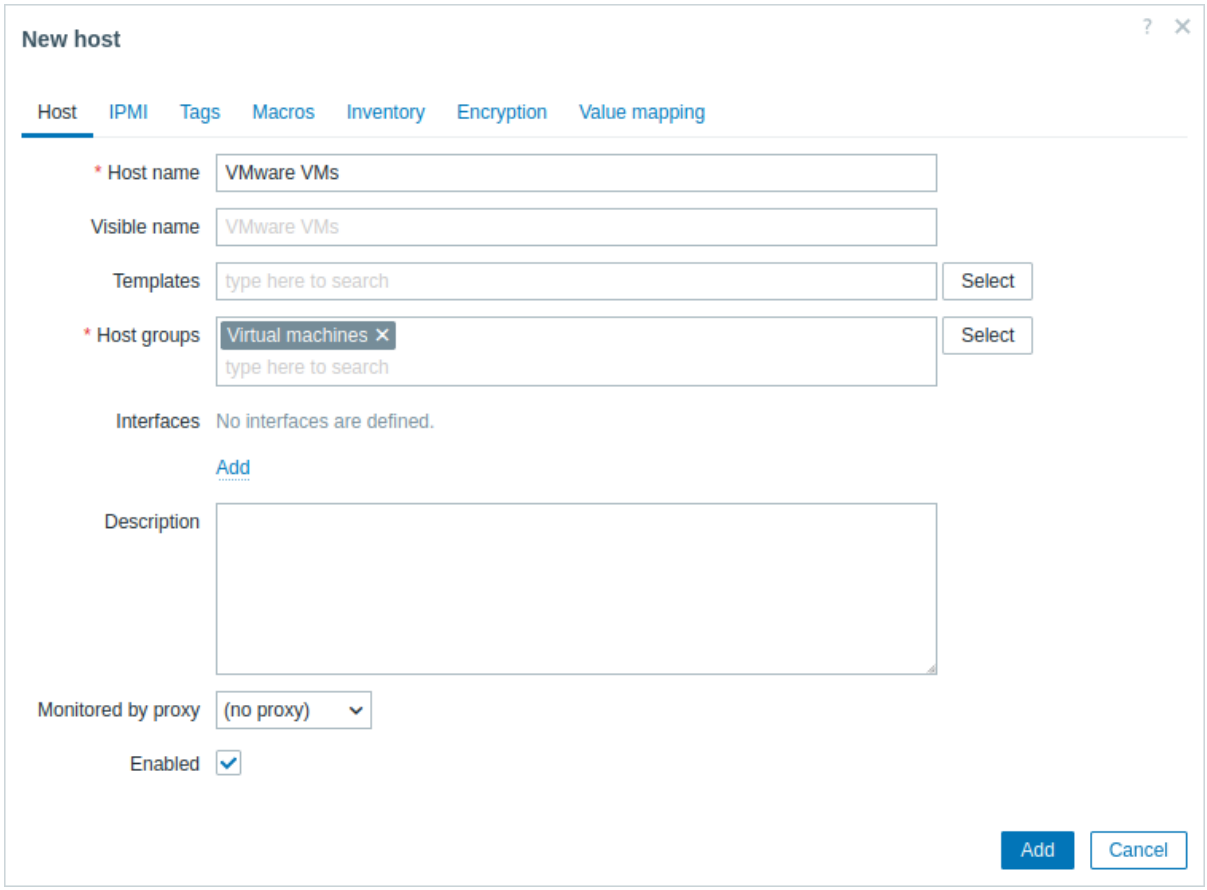
先に進む前に、Zabbix サーバーの設定ファイルで`StartVMwareCollectors` パラメータを 2 以上 に設定してください（デフォルト値は 0 です）。

ホストの作成

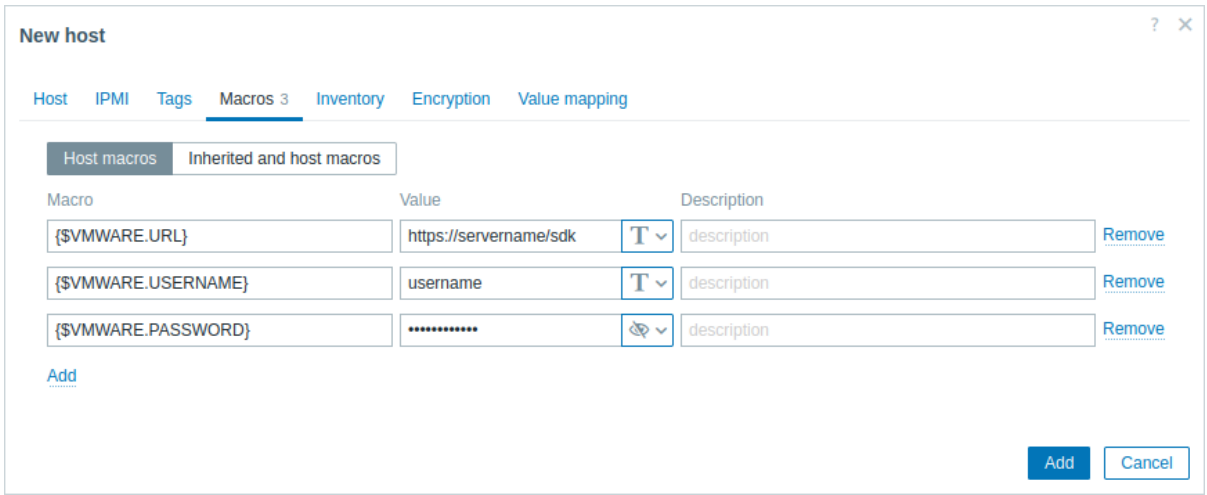
1. データ収集 → ホスト に移動します。

2. ホストを作成します。

- ホスト名フィールドに、ホスト名を入力します（例: 「VMware VMs」）。
- ホストグループフィールドで、ホストグループを入力または選択します（例: 「Virtual machines」）。



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



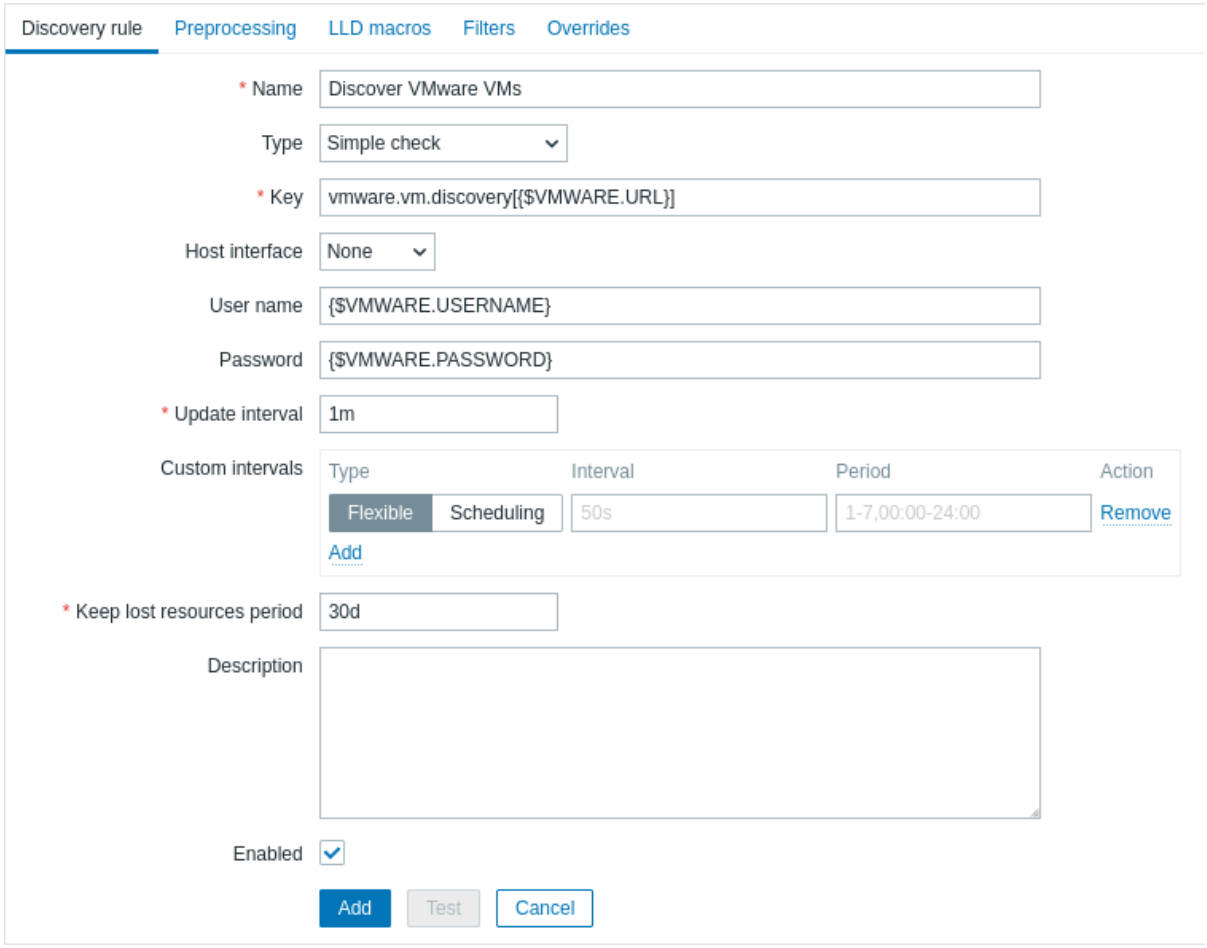
3. 追加ボタンをクリックしてホストを作成します。このホストは、VMware 環境を表します。

ローレベルディスカバリルールの作成

1. 作成したホストの *Discovery* をクリックして、そのホストのローレベルディスカバリルールの一覧に移動します。

2. ローレベルディスカバリルールを作成します。

- *Name* フィールドに、ローレベルディスカバリルール名（例: 「VMware VM を検出」）を入力します。
- *Type* フィールドで、"Simple check" を選択します。
- *Key* フィールドに、VMware 仮想マシンを検出するための組み込みアイテムキー`vmware.vm.discovery[{$VMWARE.URL}]`を入力します。
- *User name* フィールドと *Password* フィールドに、事前にホストで設定した対応するマクロを入力します。



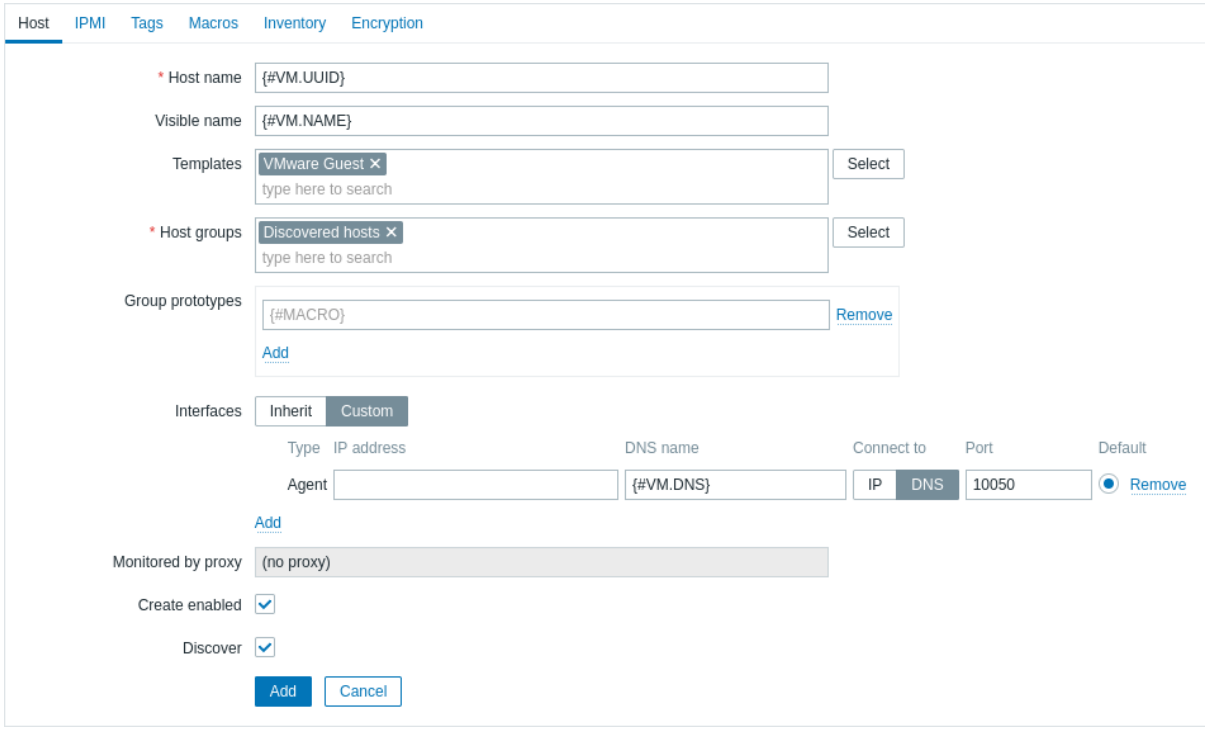
3. *Add* ボタンをクリックして、ローレベルディスカバリルールを作成します。
このディスカバリルールにより、VMware 環境内の仮想マシンが検出されます。

ホストプロトタイプの作成

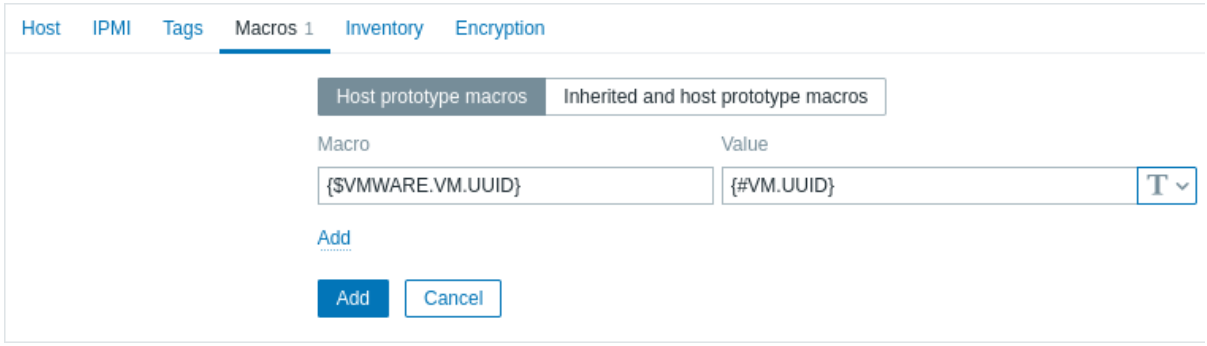
1. ローレベルディスカバリルールの一覧で、先ほど作成したローレベルディスカバリルールの *Host prototypes* をクリックします。

2. ホストプロトタイプを作成します。ホストプロトタイプは、ローレベルディスカバリルールを通じてホストを作成するためのひな形であるため、ほとんどのフィールドにはローレベルディスカバリマクロが含まれます。これにより、先ほど作成したローレベルディスカバリルールによって取得された内容に基づくプロパティでホストが作成されます。

- *Host name* フィールドに、マクロ {#VM.UUID} を入力します。
- *Visible name* フィールドに、マクロ {#VM.NAME} を入力します。
- *Templates* フィールドで、”VMware Guest” テンプレートを入力または選択します。このテンプレートには、仮想マシンの電源状態、CPU 使用率、メモリ使用率、ネットワークデバイスなどを監視するためのVMware アイテムおよびディスカバリルールが含まれています。
- *Host groups* フィールドで、ホストグループ（例: ”Discovered hosts”）を入力または選択します。
- *Interfaces* フィールドで、カスタムのホストインターフェースを追加します。その後、*DNS name* フィールドにマクロ {#VM.DNS} を入力するか、*IP address* フィールドにマクロ {#VM.IP} を入力します。または、VMware 環境の仮想マシンに複数のインターフェースがある場合は、*Advanced host interface configuration* セクションに進んでください。カスタムのホストインターフェースの設定は、*VMware Guest* テンプレートを正しく機能させるために必要です。



- *Macros* タブで、{$VMWARE.VM.UUID} マクロに値 {#VM.UUID} を設定します。これは、このマクロをアイテムパラメータ内のホストレベルユーザーマクロとして使用する *VMware Guest* テンプレートを正しく機能させるために必要です（例: *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*）。

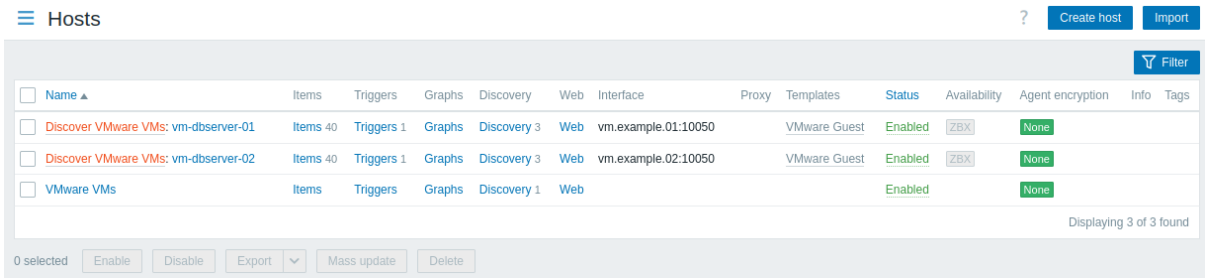


3. *Add* ボタンをクリックして、ホストプロトタイプを作成します。このホストプロトタイプは、先ほど作成したローレベルディスカバリルールによって検出された仮想マシン用のホストを作成するために使用されます。

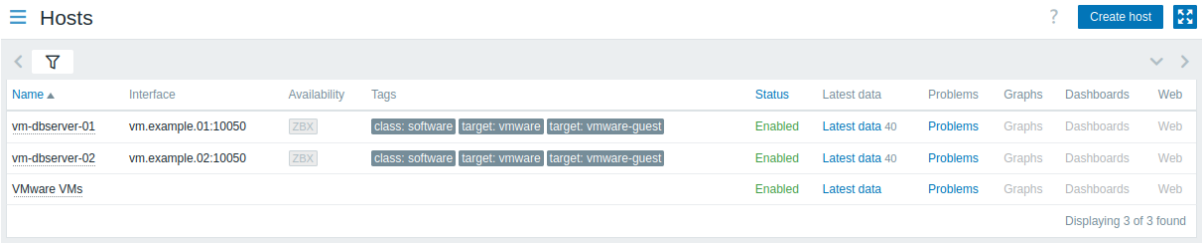
ホストとメトリクスの表示

ホストプロトタイプが作成されると、ローレベルディスカバリルールによって検出された VMware 仮想マシン用のホストが作成され、Zabbix による監視が開始されます。必要に応じて、ホストの検出と作成は手動で実行することもできます。

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



収集されたメトリクスを表示するには、監視 → ホストメニューセクションに移動し、いずれかのホストの 最新データをクリックします。



ホストインターフェースの高度な設定

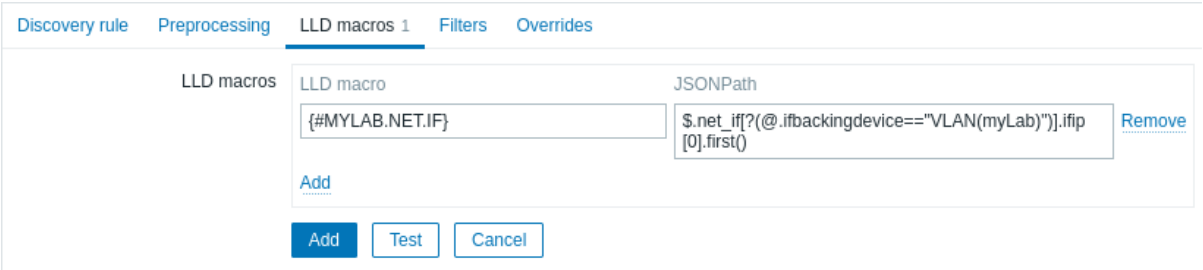
低レベルディスカバリルールの作成 セクションで設定した `vmware.vm.discovery[{$VMWARE.URL}]` アイテムキーは、"net_if" フィールド にネットワークインターフェースのデータを返します。

```
"net_if": [
      {
        "ifname": "5000",
        "ifdesc": "Network adapter 1",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": true,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": [
          "127.0.0.1",
          "::1"
        ]
      },
      {
        "ifname": "5001",
        "ifdesc": "Network adapter 2",
        "ifmac": "00:11:22:33:44:55",
        "ifconnected": false,
        "iftype": "VirtualVmxnet3",
        "ifbackingdevice": "VLAN(myLab2)",
        "ifdvswitch_uuid": "",
        "ifdvswitch_portgroup": "",
        "ifdvswitch_port": "",
        "ifip": []
      }
    ]
```

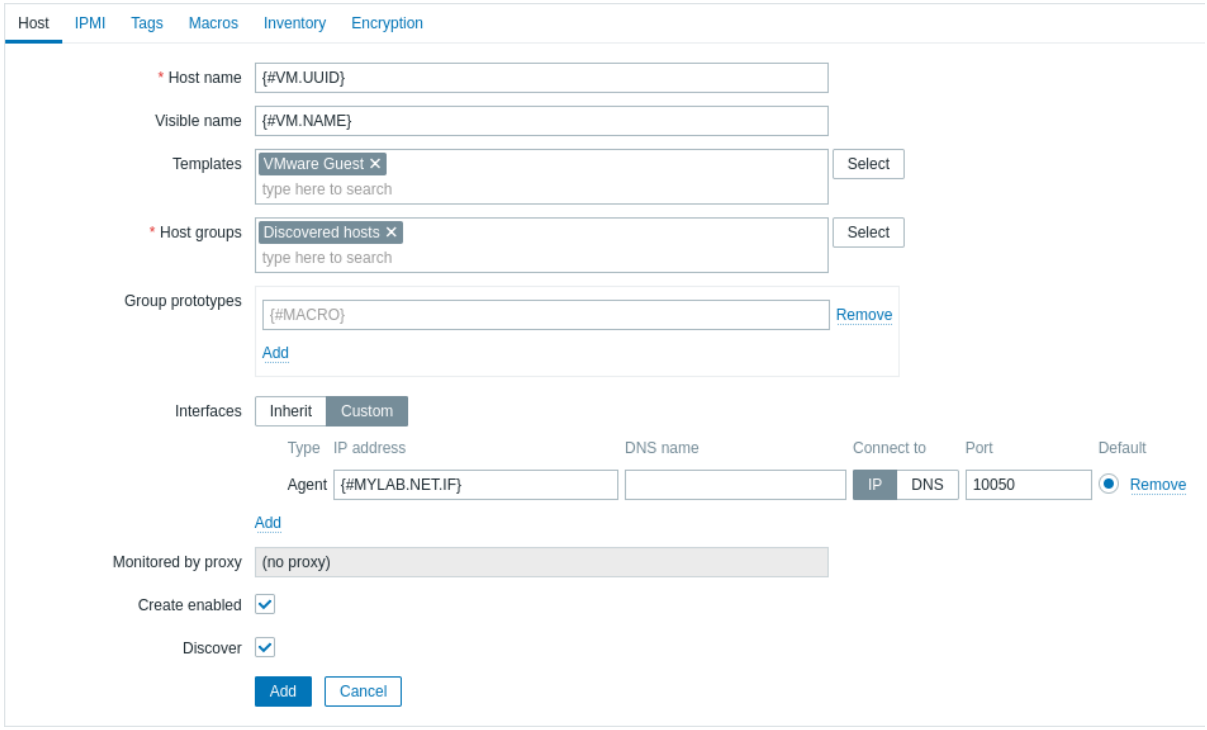
このデータは、カスタムホストインターフェースの設定に使用できます。

1. 低レベルディスカバリルールを作成する際に、追加で低レベルディスカバリマクロ も設定します。
LLD macros タブで、JSONPath の値を持つカスタム LLD マクロを作成します。例:

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. ホストプロトタイプを作成する際に、カスタムホストインターフェースを追加し、*DNS name* または *IP address* フィールドに LLD マクロを入力します。



11. メンテナンス

概要　Zabbix では、ホストおよびホストグループに対してメンテナンス期間を定義できます。

さらに、トリガータグを指定することで、単一のトリガー（またはトリガーのサブセット）に対してのみメンテナンスを定義することも可能です。この場合、メンテナンスはそれらのトリガーに対してのみ有効になり、ホストまたはホストグループ内のその他すべてのトリガーはメンテナンス中にはなりません。

メンテナンスには、データ収集ありとデータ収集なしの 2 種類があります。

「データ収集あり」のメンテナンス中は、トリガーは通常どおり処理され、必要に応じてイベントが作成されます。ただし、アクション設定で *Pause operations for suppressed problems* オプションがチェックされている場合、メンテナンス中のホスト/トリガーに対する障害のエスカレーションは一時停止されます。この場合、通知の送信やリモートコマンドの実行を含む可能性のあるエスカレーションステップは、メンテナンス期間が終了するまで無視されます。メンテナンス中に抑止されるのはエスカレーションのみであり、障害の復旧および更新操作は抑止されないことに注意してください。

たとえば、障害の発生後 0 分、30 分、60 分にエスカレーションステップが予定されており、実際の障害発生から 10 分後から 40 分後までの 30 分間のメンテナンスがある場合、2 番目と 3 番目のステップは 30 分遅れて、つまり 60 分後と 90 分後に実行されます(障害がまだ継続している場合)。同様に、メンテナンス中に障害が発生した場合、エスカレーションはメンテナンス終了後に開始されます。

メンテナンス中も通常どおり（遅延なく）障害通知を受け取るには、アクション設定で *Pause operations for suppressed problems* オプションのチェックを外す必要があります。

> **Note:**
> （トリガー式で使用される）少なくとも 1 つのホストがメンテナンスモードでない場合、Zabbix は障害通知を送信します。

メンテナンス中は Zabbix サーバーが稼働している必要があります。メンテナンスは毎分再計算されます。また、メンテナンス期間に変更がある場合は、設定キャッシュが再読み込みされるとすぐに再計算されます。

タイマープロセスは、毎分 0 秒時点でホストの状態をメンテナンスに変更する必要があるか、またはメンテナンスから戻す必要があるかを確認します。さらに、設定更新後に [maintenance periods] に変更があるかどうかに基づいて、開始/停止すべきメンテナンスがあるかをタイマープロセスが毎秒確認します。したがって、メンテナンス期間の開始/停止の速度は、設定のupdate interval（デフォルトでは 10 秒）に依存します。メンテナンス期間の変更には、*Active since/Active till* 設定は含まれないことに注意してください。また、既存の有効なメンテナンス期間にホスト/ホストグループが追加された場合、その変更は次の分の開始時にのみタイマープロセスによって有効化されます。

ホストがメンテナンスに入ると、Zabbix サーバーのタイマープロセスは、抑止が必要かどうかを確認するために、すべての未解決の障害を読み取ることに注意してください。未解決の障害が多数ある場合、これはパフォーマンスに影響する可能性があります。また、

Zabbix サーバーは起動時にも、たとえその時点でメンテナンスが設定されていなくても、すべての未解決の障害を読み取ります。

メンテナンスの種類（「データなし」メンテナンスを含む）に関係なく、Zabbix サーバー（またはプロキシ）は常にデータを収集することに注意してください。その後、'no data collection' が設定されている場合、そのデータはサーバーによって無視されます。

「データなし」メンテナンスが終了したとき、nodata() 関数を使用するトリガーは、その監視期間内の次回チェックまでは発火しません。

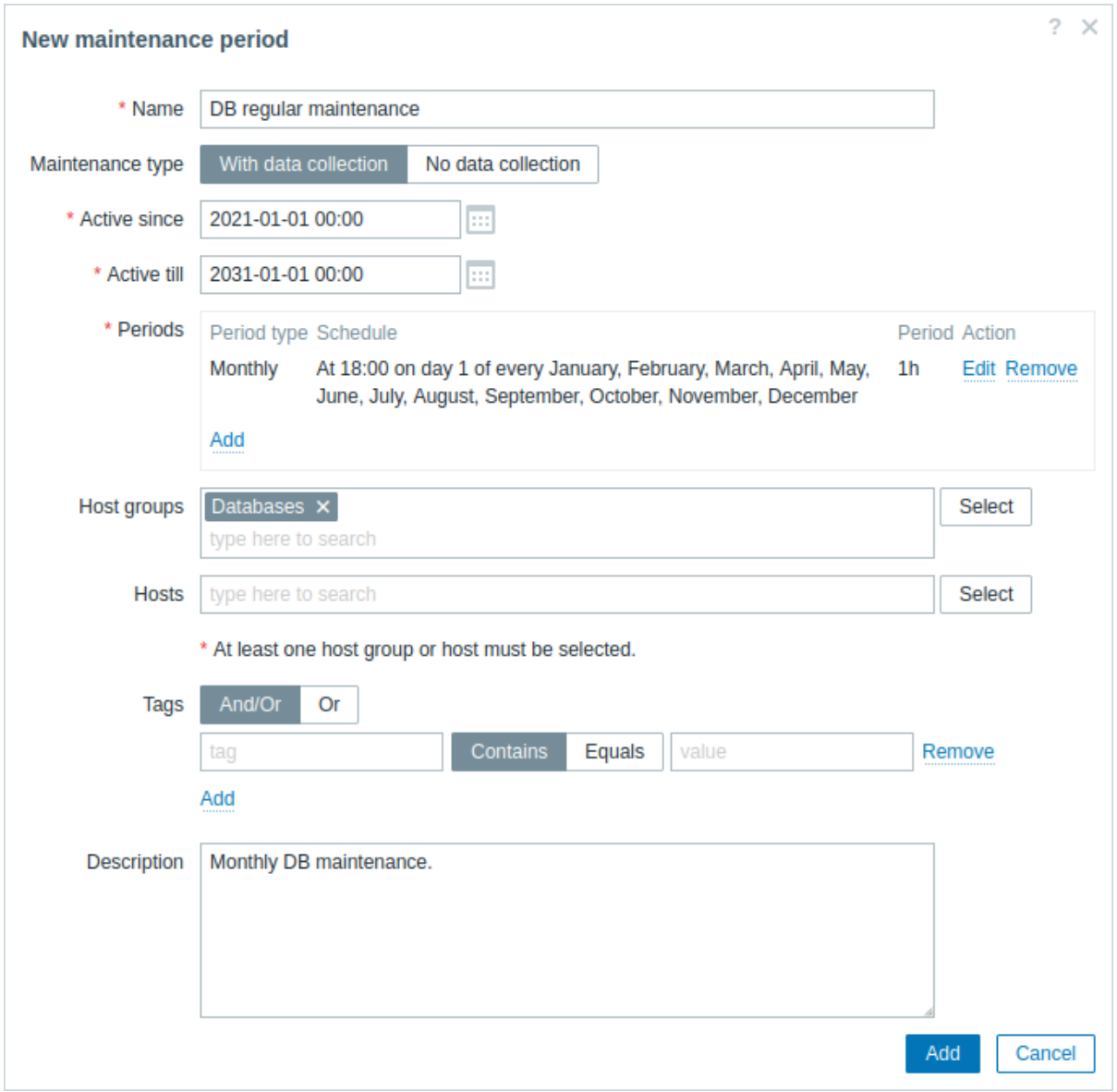
ホストがメンテナンス中にログアイテムが追加され、その後メンテナンスが終了した場合、収集されるのはメンテナンス終了以降の新しいログファイルエントリのみです。

「データなし」タイプのメンテナンス中のホストに対してタイムスタンプ付きの値が送信された場合（たとえばZabbix sender を使用した場合）、その値は破棄されます。ただし、期限切れのメンテナンス期間に対してタイムスタンプ付きの値を送信することは可能であり、その場合は受け入れられます。

メンテナンス期間、ホスト、グループ、またはタグがユーザーによって変更された場合、その変更は設定キャッシュの同期後にのみ有効になります。

設定　メンテナンス期間を設定するには、次のようにします。

1. データ収集 > メンテナンスに移動します。
2. メンテナンス期間の作成をクリックします（または既存のメンテナンス期間の名前をクリックします）。
3. フォームにメンテナンスパラメータを入力します。



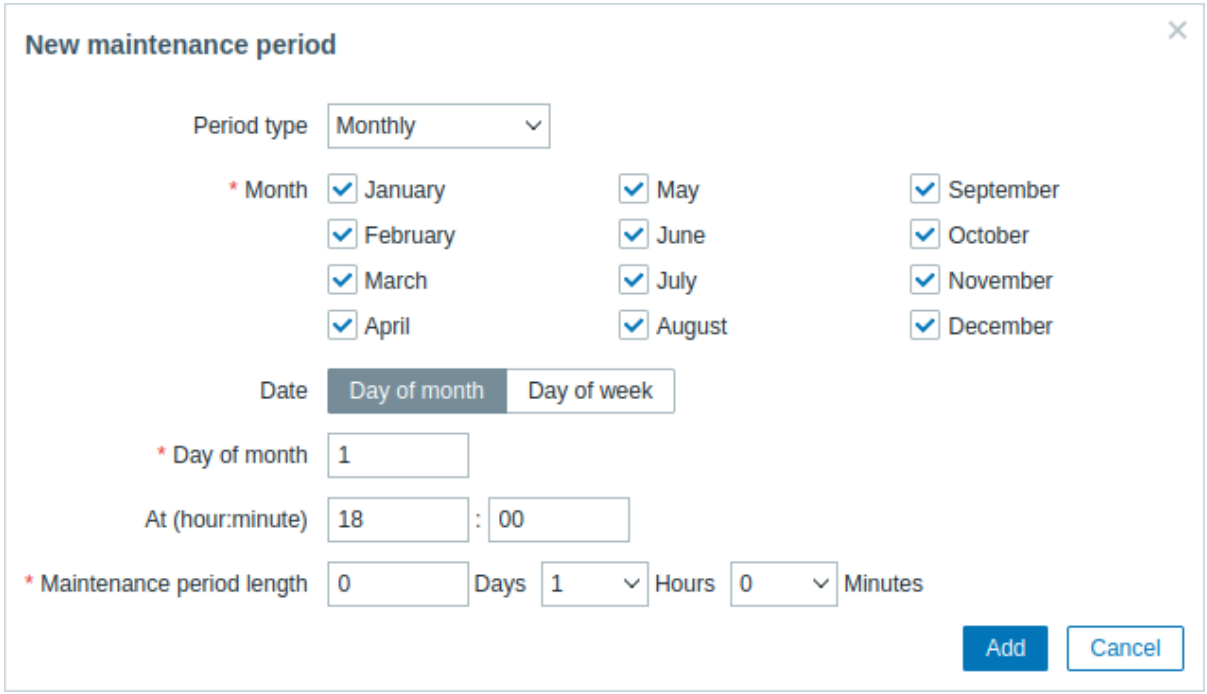
必須入力フィールドには、赤いアスタリスクが付いています。

Parameter	Description
Name	メンテナンス期間の名前。

Parameter	Description
Maintenance type	設定できるメンテナンスタイプは 2 種類あります。 データ収集あり - メンテナンス中もサーバーによってデータが収集され、トリガーが処理されます。 データ収集なし - メンテナンス中はサーバーによってデータは収集されません。 各タイプが可用性レポートに与える影響については、メンテナンス期間の影響を参照してください。
Active since	メンテナンス期間の実行が有効になる日時。 注: この時刻を設定しただけではメンテナンス期間は有効になりません。メンテナンス期間は *Periods* で設定する必要があります（以下を参照）。
Active till	メンテナンス期間の実行が無効になる日時。
Periods	このブロックでは、メンテナンスを実施する正確な曜日と時間を定義できます。Add をクリックすると、柔軟にメンテナンススケジュールを定義できる *Maintenance period* フォームを含むポップアップウィンドウが開きます。詳細はメンテナンス期間を参照してください。
Host groups	メンテナンスを有効にするホストグループを選択します。指定したホストグループ内のすべてのホストに対してメンテナンスが有効になります。このフィールドはオートコンプリート対応のため、入力を開始すると利用可能なすべてのホストグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。そのため、ネストされたグループ内のホストでもメンテナンスが有効になります。
Hosts	メンテナンスを有効にするホストを選択します。このフィールドはオートコンプリート対応のため、入力を開始すると利用可能なすべてのホストのドロップダウンが表示されます。
Tags	メンテナンス中のホストで、一致するタグを持つ障害を抑制するためのタグを指定します。 複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の 2 つの演算子を使用できます。 含む - 指定したタグ名を対象とし、タグ値に入力した文字列が含まれるものを含めます（部分一致、大文字と小文字を区別）。 等しい - 指定したタグ名とタグ値を含めます（大文字と小文字を区別）。 条件には次の 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 タグを指定できるのは、*With data collection* モードが選択されている場合のみです。
Description	メンテナンス期間の説明。

メンテナンス期間

メンテナンス期間のウィンドウは、定期または 1 回限りのメンテナンスの時間をスケジュールするためのものです。
このフォームは動的で、使用可能なフィールドは選択した *Period type* に応じて変化します。



Period type	説明
One time only	1 回限りのメンテナンス期間を設定します： *Date* - メンテナンス期間の日時； *Maintenance period length* - メンテナンスを有効にする期間。
Daily	毎日のメンテナンス期間を設定します： *Every day(s)* - メンテナンスの頻度（1 - *(デフォルト)* 毎日、2 - 2 日ごと、など）； *At (hour:minute)* - メンテナンスを開始する時刻； *Maintenance period length* - メンテナンスを有効にする期間。 *Every day(s)* パラメータが”1” より大きい場合、開始日は *Active since* の時刻が属する日になります。例： - *Active since* が”2021-01-01 12:00”、*Every day(s)* が”2”、*At (hour:minute)* が”23:00” に設定されている場合、最初のメンテナンス期間は 1 月 1 日 23:00 に開始し、2 回目のメンテナンス期間は 1 月 3 日 23:00 に開始します； - *Active since* が”2021-01-01 12:00”、*Every day(s)* が”2”、*At (hour:minute)* が”01:00” に設定されている場合、最初のメンテナンス期間は 1 月 3 日 01:00 に開始し、2 回目のメンテナンス期間は 1 月 5 日 01:00 に開始します。
Weekly	毎週のメンテナンス期間を設定します： *Every week(s)* - メンテナンスの頻度（1 - *(デフォルト)* 毎週、2 - 2 週間ごと、など）； *Day of week* - メンテナンスを実施する曜日； *At (hour:minute)* - メンテナンスを開始する時刻； *Maintenance period length* - メンテナンスを有効にする期間。 *Every week(s)* パラメータが”1” より大きい場合、開始週は *Active since* の時刻が属する週になります。例については、上記の *Daily* パラメータの説明を参照してください。
Monthly	毎月のメンテナンス期間を設定します： *Month* - 定期メンテナンスを実施するすべての月を選択します； *Date:* ***Day of month*** - メンテナンスを毎月同じ日付（例えば毎月 1 日）に実施する場合はこのオプションを選択し、表示される *Day of month* フィールドで必要な日を選択します； *Date:* ***Day of week*** - メンテナンスを特定の日（例えば毎月第 1 月曜日）にのみ実施する場合はこのオプションを選択し、次に（ドロップダウンで）必要な週（第 1、第 2、第 3、第 4、または最終）を選択し、その後メンテナンスを実施する曜日のチェックボックスをオンにします； *At (hour:minute)* - メンテナンスを開始する時刻； *Maintenance period length* - メンテナンスを有効にする期間。

Attention:
メンテナンス期間を作成する際は、それを作成したユーザーのタイムゾーンが使用されます。ただし、定期的なメンテナンス期間（毎日、毎週、毎月）をスケジュールする場合は、Zabbix サーバーのタイムゾーンが使用されます。定期的なメンテナンス期間の動作を予測可能にするため、Zabbix のすべての部分で共通のタイムゾーンを使用する必要があります。

完了したら、*Add* を押してメンテナンス期間を *Periods* ブロックに追加します。

サマータイム（DST）の変更は、メンテナンスの継続時間には影響しないことに注意してください。例えば、通常は 01:00 に開始して 03:00 に終了する、2 時間のメンテナンスが設定されているとします。

- メンテナンス開始から 1 時間後（02:00）に DST の変更が発生し、現在時刻が 02:00 から 03:00 に変わった場合、メンテナンスはさらに 1 時間継続します（04:00 まで）。
- メンテナンス開始から 2 時間後（03:00）に DST の変更が発生し、現在時刻が 03:00 から 02:00 に変わった場合、2 時間が経過しているため、メンテナンスは停止します。
- メンテナンス期間が、DST の変更によってスキップされる時間帯に開始するよう設定されている場合、そのメンテナンスは開始されません。

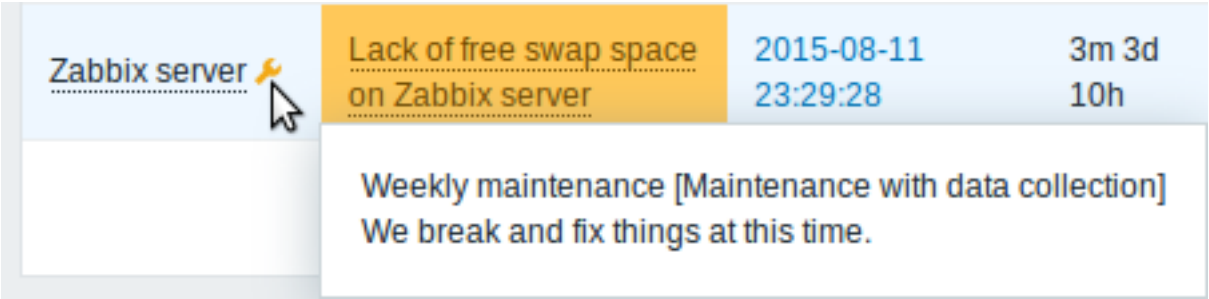
メンテナンス期間が「1 day」に設定されている場合（Zabbix は日数を時間単位で計算するため、実際のメンテナンス期間は 24 時間です）、00:00 に開始して翌日の 00:00 に終了するとします。

- 現在時刻が 1 時間進んだ場合、メンテナンスは翌日の 01:00 に停止します。
- 現在時刻が 1 時間戻った場合、メンテナンスはその日の 23:00 に停止します。

表示　メンテナンス中のホストの表示

ホスト名の横にあるオレンジ色のレンチアイコン は、このホストが以下でメンテナンス中であることを示します。

- ダッシュボード
- 監視 > 障害
- インベントリ > ホスト > ホストインベントリの詳細
- データ収集 > ホスト（「ステータス」列を参照）



マウスポインターをアイコンの上に置くと、メンテナンスの詳細が表示されます。

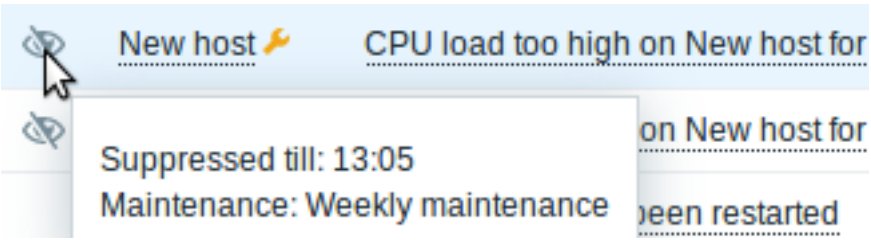
さらに、メンテナンス中のホストは 監視 > マップでオレンジ色の背景で表示されます。

抑制された障害の表示

通常、メンテナンス中のホストの障害は抑制され、つまり Web インターフェースには表示されません。ただし、以下の場所で *Show suppressed problems* オプションを選択することで、抑制された障害を表示するよう設定することもできます。

- ダッシュボード（*Problem hosts*、*Problems*、*Problems by severity*、*Trigger overview* ウィジェット設定内）
- 監視 > 障害（フィルター内）
- 監視 > マップ（マップ設定内）
- グローバル通知 （ユーザープロファイル設定内）

抑制された障害が表示される場合、次のアイコンが表示されます: 。アイコンにマウスカーソルを合わせると、さらに詳細が表示されます。



メンテナンス中のキューの計算　Zabbix の Web インターフェース（Administration > Queue）に表示されるキューは、Zabbix サーバーによって計算されます。
データ収集を行わないメンテナンス中のアイテムは含まれません。このようなアイテムは、値の取得が遅延している場合でも、キュー長は常にゼロです。
データ収集を行うメンテナンス中の遅延アイテムは、引き続きキューにカウントされます。

Zabbix のプロキシは、Zabbix サーバーとプロキシの間でメンテナンス設定が同期されないため、メンテナンス期間を認識しません。Zabbix プロキシ上で計算される内部チェック（たとえば、`zabbix[queue,,]` および`zabbix[stats,,,queue,,]`）は、Zabbix サーバー上のメンテナンス状態に関係なく、遅延アイテムを報告します。

その結果、データ収集を行わないメンテナンス中の同じアイテムについて、Zabbix の Web インターフェースと Zabbix プロキシ上の内部チェックとで、異なるキュー長が報告される場合があります。

12. 正規表現

概要　Perl 互換の正規表現 (PCRE、PCRE2) が Zabbix でサポートされています。

Zabbix で正規表現を使用するには 2 つの方法があります。

- 正規表現を手動で入力する
- Zabbix で作成したグローバルな正規表現を使用する。

正規表現　サポートされている箇所では、正規表現を手動で入力できます。
ただし、この記号は Zabbix でグローバル正規表現の参照に使用されるため、式の先頭を @ にすることはできません。

> **Warning:**
> 正規表現の使用時にスタック不足が発生する可能性があります。詳細は pcrestack man page を参照してください。

複数行マッチングでは、アンカー ^ と $ は文字列全体の先頭/末尾ではなく、それぞれ各行の先頭/末尾にマッチすることに注意してください。

あわせて、さまざまなコンテキストにおける正しいエスケープの例も参照してください。

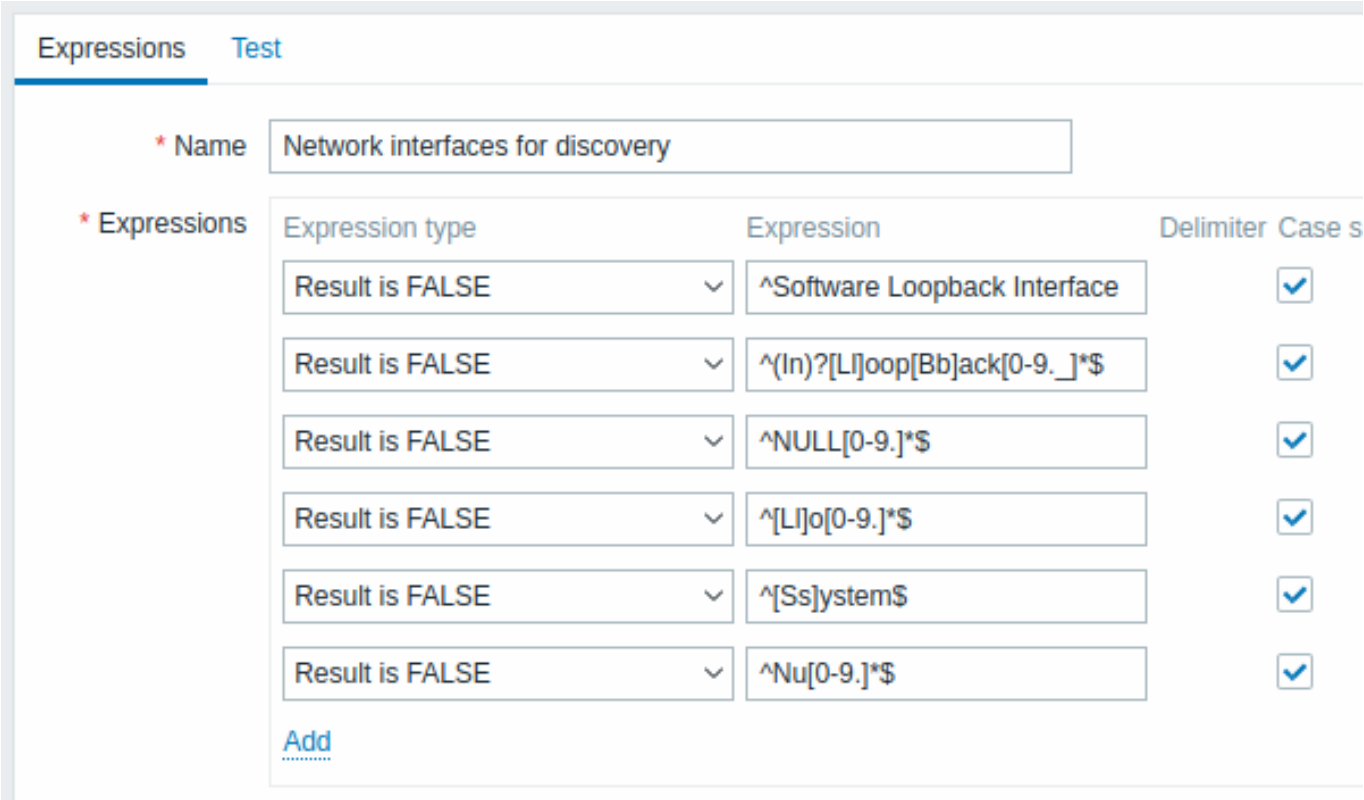
グローバルな正規表現　Zabbix フロントエンドで複雑な正規表現を作成およびテストするための高度なエディターがあります。

このようにして作成した正規表現は、*@mycustomregexp* のように、先頭に @ を付けたその名前を参照することで、フロントエンドのさまざまな場所で使用できます。

グローバルな正規表現を作成するには:

- 管理 → 一般設定に移動。
- ドロップダウンから正規表現を選択。
- 正規表現の作成をクリック。

条件式タブでは、正規表現名を設定し、部分式を追加できます。



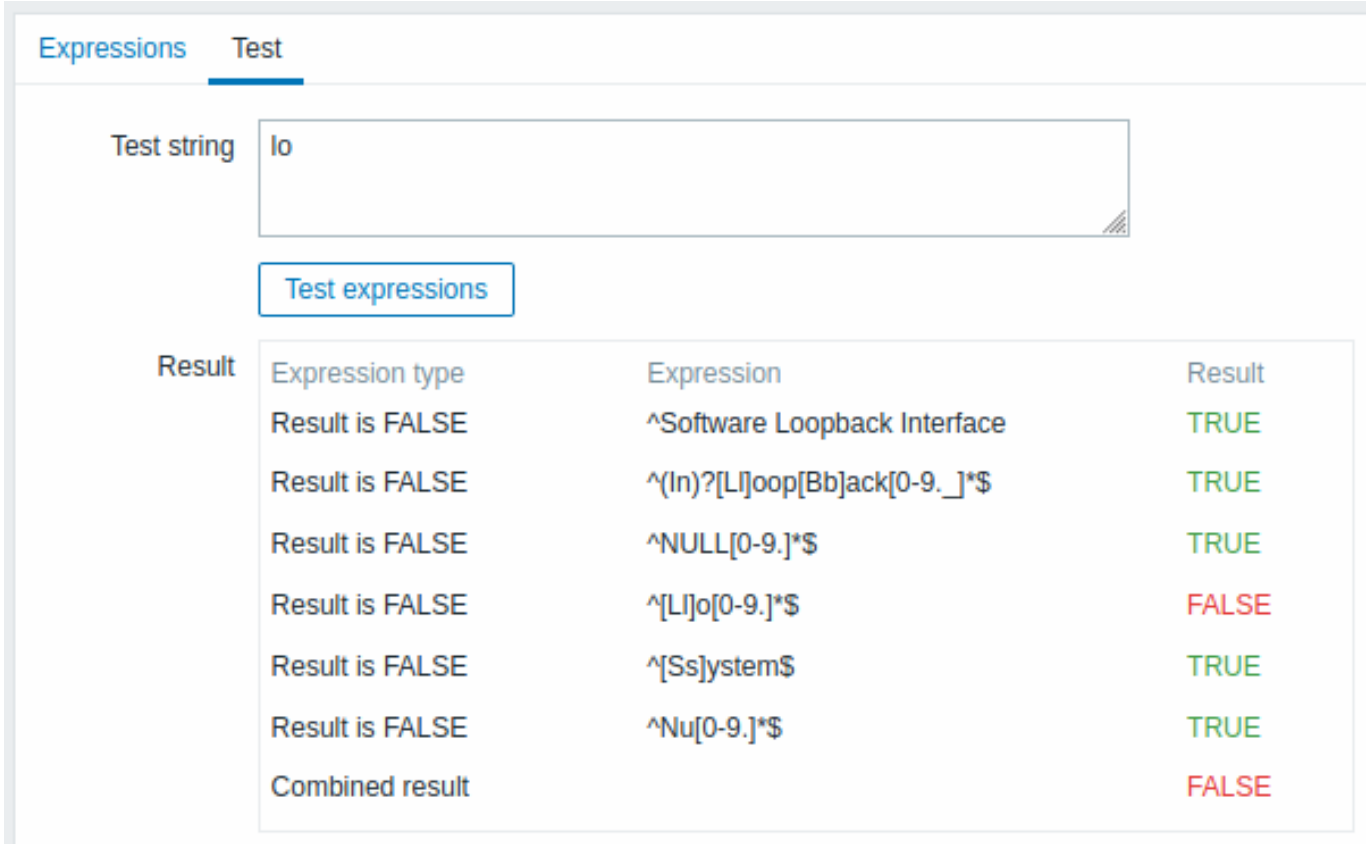
必須入力項目には、赤いアスタリスクが表示されています。

パラメータ		説明
名前		正規表現名を設定します。すべての Unicode 文字を使用できます。
条件式		条件式ブロックの 追加をクリックして、新しい部分式を追加します。
	条件式の形式	条件式の形式を選択します: 文字列が含まれる - 部分文字列と一致します いずれかの文字列が含まれる - 区切りリストの任意の部分文字列と一致します。区切りリストには、コンマ (,)、ドット (.)、またはスラッシュ (/) が含まれます。 文字列が含まれない - 部分文字列以外の任意の文字列に一致します 結果が真 - 正規表現に一致する 結果が偽 - 正規表現に一致しない
	条件式	部分文字列/正規表現を入力します。
区切り文字		正規表現でテキスト文字列を区切るには、コンマ (,)、ドット (.)、またはスラッシュ (/) を使用します。このパラメーターは条件式の形式が” いずれかの文字列が含まれる” を選択している場合にのみ有効です。
大文字と小文字を区別		正規表現が文字の大文字と小文字を区別するかどうかを指定するチェックボックス

式内のスラッシュ (/) は、区切り文字ではなく文字どおりに扱われるので、スラッシュを含む式はエラーなしで保存できます。

Attention:
Zabbix のカスタム正規表現名には、カンマやスペースなどを含むことができます。そのため、参照時に誤解を招く可能性がある場合 (例えば、アイテムキーのパラメータにカンマがある場合)、参照全体を次のように引用符で囲むことができます。
”@My custom regexp for purpose1, purpose2”.
正規表現名は、他の場所（たとえば LLD ルールのプロパティ）では引用符で囲まないでください。

テストタブでは、テスト文字列を指定することで、正規表現とその部分式をテストできます。



結果には、各部分式のステータスとカスタム式の合計ステータスが表示されます。

カスタム式の合計ステータスは、結合結果として定義されます。複数のサブ式が定義されている場合、Zabbix は AND 論理演算子を使用して 結合結果を計算します。これは、少なくとも 1 つの結果が False の場合、結合された結果も False ステータスになることを意味します。

デフォルトのグローバル正規表現　Zabbix には、デフォルトのデータセットとしていくつかのグローバル正規表現が含まれています。

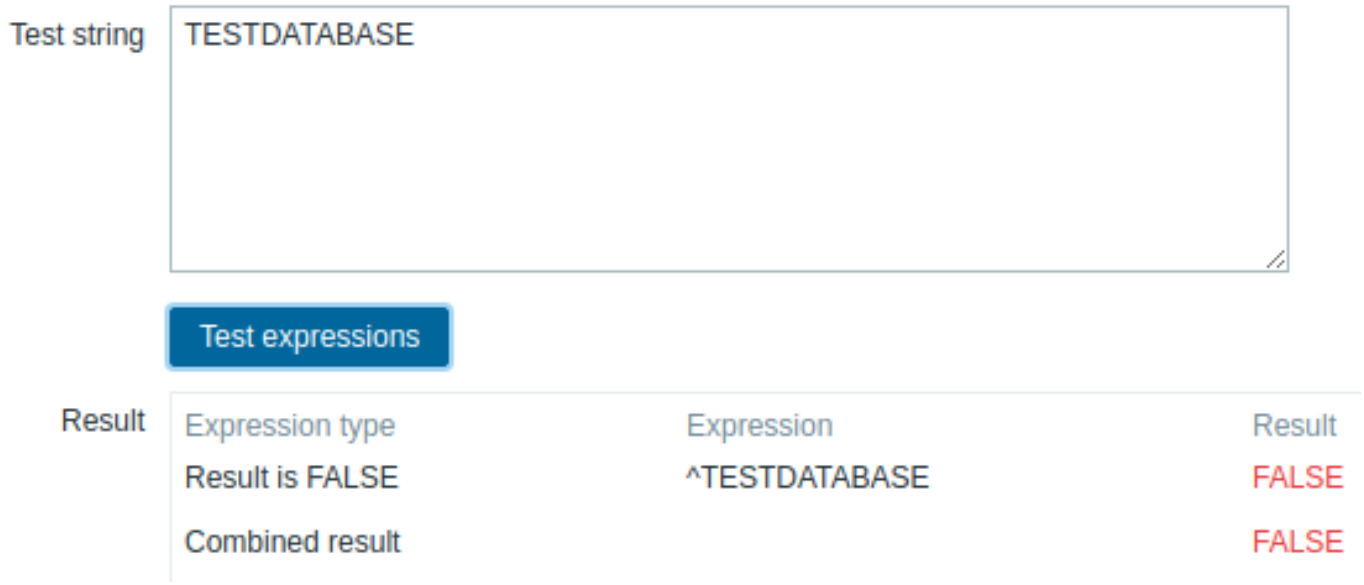
Name	Expression	Matches
ディスカバリ用ファイルシステム	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)$`	「btrfs」または「ext2」または「ext3」または「ext4」または「jfs」または「reiser」または「xfs」または「ffs」または「ufs」または「jfs」または「jfs2」または「vxfs」または「hfs」または「refs」または「apfs」または「ntfs」または「fat32」または「zfs」
ディスカバリ用ネットワークインターフェース	`^Software Loopback Interface`	「Software Loopback Interface」で始まる文字列。
	`^lo$`	「lo」
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	先頭に任意で「In」があり、その後に「L」または「l」、続いて「oop」、「B」または「b」、続いて「ack」があり、その後ろに任意の数の数字、ドット、またはアンダースコアが続く文字列。
	`^NULL[0-9.]*$`	「NULL」で始まり、その後ろに任意の数の数字またはドットが続く文字列。
	`^[Ll]o[0-9.]*$`	「Lo」または「lo」で始まり、その後ろに任意の数の数字またはドットが続く文字列。
	`^[Ss]ystem$`	「System」または「system」
	`^Nu[0-9.]*$`	「Nu」で始まり、その後ろに任意の数の数字またはドットが続く文字列。
SNMP ディスカバリ用ストレージデバイス	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	「Physical memory」または「Virtual memory」または「Memory buffers」または「Cached memory」または「Swap space」

Name	Expression	Matches
ディスカバリ用 *Windows* サービス名	^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$	「MMCSS」、「gupdate」、「SysmonLog」、「clr_optimization_v2.0.50727_32」または「clr_optimization_v4.0.30319_32」のような文字列(ドットの代わりに改行以外の任意の文字を使用可能)。
ディスカバリ用 *Windows* サービス起動状態	^(automatic|automatic delayed)$	「automatic」または「automatic delayed」

例　例 1

低レベルの検出で次の式を使用すると、特定の名前のデータベース以外を検出できます。

^TESTDATABASE$

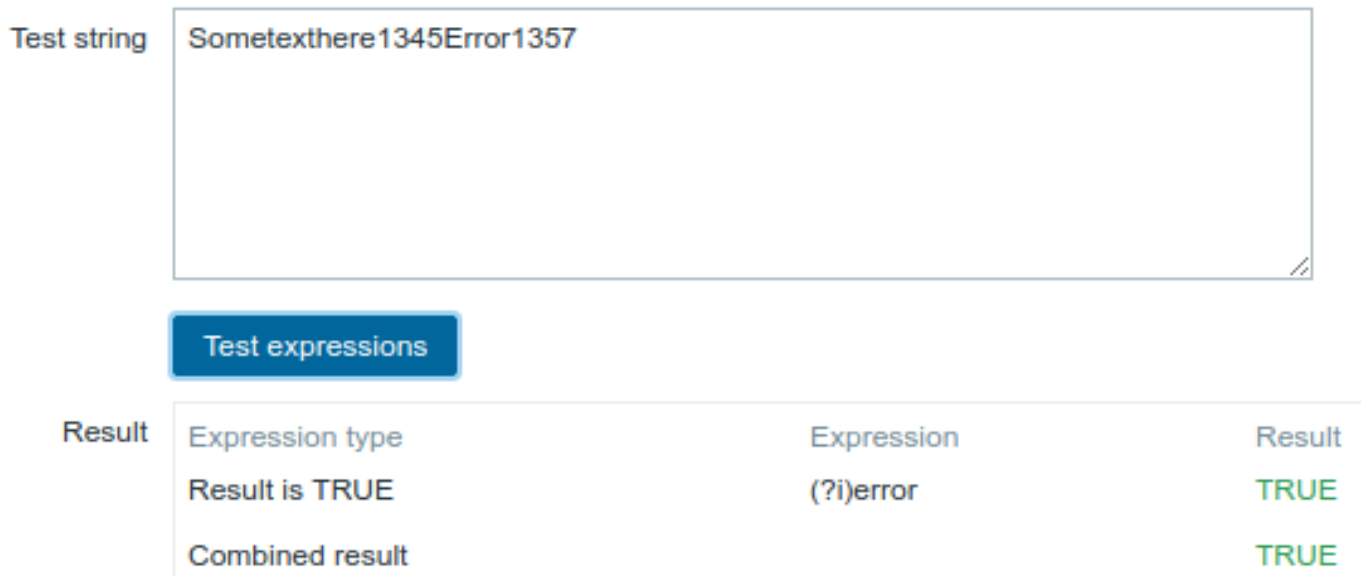


選択された *Expression type*：" 結果は FALSE " です。"*TESTDATABASE*" を含んだ名前と一致しません。

インライン型正規表現修飾子を含む例

インラインモディファイアを含む次の正規表現を使用します。
 (?i) を使用して、"error" という文字にマッチさせます。

(?i)error

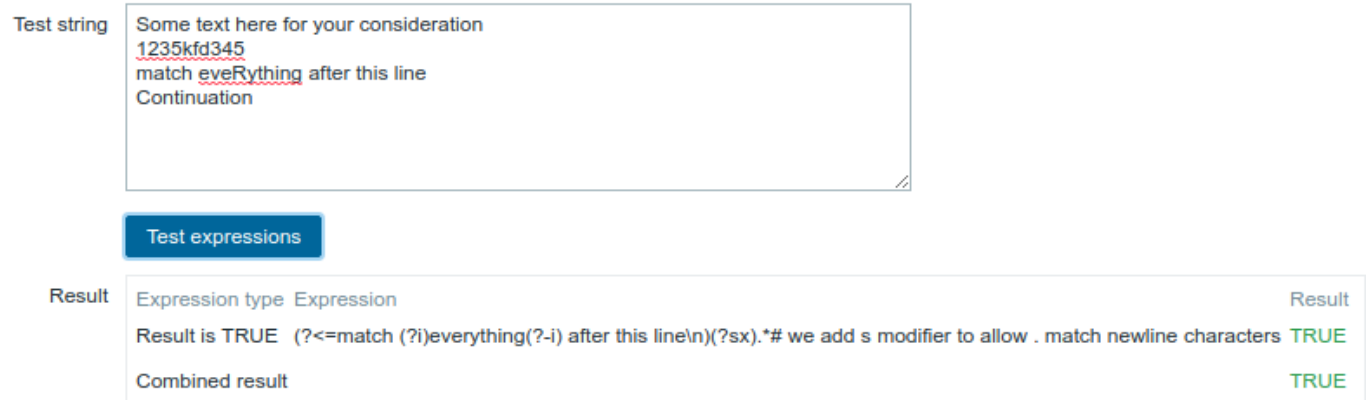


選択された *Expression type*：" 結果は TRUE " です。文字"error " にマッチする。

インライン正規表現修飾子を含む別の例

複数のインライン修飾子を含む次の正規表現を使用すると、特定の行以降の文字にマッチします。

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# we add s modifier to allow .match newline charact
```



選択された Expression type：” 結果は TRUE ” です。
 特定の行以降の文字にマッチします。

Attention:
g 修飾子は、行内で指定することはできません。利用可能な修飾子の一覧は
 利用可能な修飾子のリストは、pcresyntax manpageにあります。
 PCRE の構文に関するより詳しい情報は PCRE HTML documentation を参照してください。

場所ごとの正規表現サポート

場所		正規表現	グローバル正規表現	複数行マッチング	コメント
エージェントアイテム					
	eventlog[]	Yes	Yes	Yes	regexp、severity、source、eventid パラメータ
	eventlog.count[]				regexp、severity、source、eventid パラメータ
	log[]				regexp パラメータ
	log.count[]				
	logrt[]		Yes/No		regexp パラメータは両方をサポートし、file_regexp パラメータは非グローバル式のみをサポート
	logrt.count[]				
	proc.cpu.util[]		No	No	cmdline パラメータ
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	sensor[]				Linux 2.4 の device および sensor パラメータ
	system.hw.macaddr[]				interface パラメータ
	system.sw.packages[]				regexp パラメータ
	system.sw.packages.get[]				regexp パラメータ
	vfs.dir.count[]				regex_incl、regex_excl、regex_excl_dir パラメータ
	vfs.dir.get[]				regex_incl、regex_excl、regex_excl_dir パラメータ
	vfs.dir.size[]				regex_incl、regex_excl、regex_excl_dir パラメータ
	vfs.file.regexp[]			Yes	regexp パラメータ
	vfs.file.regmatch[]				

場所		正規表現	グローバル正規表現	複数行マッチング	コメント
	web.page.regexp[]				
SNMP トラップ					
	snmptrap[]	Yes	Yes	No	`regexp` パラメータ
		Yes	No	No	`pattern` パラメータ
アイテム値の前処理トリガー/計算アイテム用関数					
	count()	Yes	Yes	Yes	`operator` パラメータが *regexp* または *iregexp* の場合の `pattern` パラメータ
	countunique()	Yes	Yes		
	find()	Yes	Yes		
	logeventid()	Yes	Yes	No	`pattern` パラメータ
	logsource()				
ローレベルディスカバリ					
	フィルター	Yes	Yes	No	*Regular expression* フィールド
	オーバーライド	Yes	No		*Operation* 条件の *matches*、*does not match* オプション内
アクション条件		Yes	No	No	*Host name* および *Host metadata* 自動登録条件の *matches*、*does not match* オプション内
スクリプト		Yes	Yes	No	*Input validation rule* フィールド

場所		正規表現	グローバル正規表現	複数行マッチング	コメント
Web監視ユーザーマクロコンテキストマクロ関数		Yes	No	Yes	**regex:** 接頭辞付きの *Variables* *Required string* フィールド
		Yes	No	No	**regex:** 接頭辞付きのマクロコンテキスト内
	regsub() iregsub()	Yes	No	No	`pattern` パラメータ
マップ内のリンクインジケーター		Yes	No	No	*Pattern* フィールド（テキストアイテム用）
アイコンマッピング		Yes	Yes	No	*Expression* フィールド
値のマッピング		Yes	No	No	マッピングタイプが `regexp` の場合の *Value* フィールド

13. 障害イベントの確認

概要　Zabbix の障害イベントはユーザーが確認することができます。

ユーザーは障害イベントの通知を受けると、Zabbix フロントエンドに移動し、以下のいずれかの方法で障害の更新ポップアップウィンドウを開き、障害を確認することができます。確認する際に、障害に対するコメント (対応中など) を入力することができます。

これにより、他のシステムユーザーが同じ問題を発見した場合、すぐに、それが確認済かどうかや、これまでのコメントを見ることができます。

このように、複数のシステムユーザーとの問題解決のワークフローを連携して行うことができます。

確認状況は、アクション操作を定義する際にも使用されます。例えば、あるイベントがしばらく確認されなかった場合にのみ、上位のマネージャに通知を送信するように定義できます。

イベントを確認し、それについてコメントするには、ユーザーは対応するトリガーに対して最低限の読み取り権限を持っている必要があります。障害の深刻度を変更したり、障害をクローズしたりするには、ユーザーは対応するトリガーに対して読み取り/書き込みの権限を持っている必要があります。

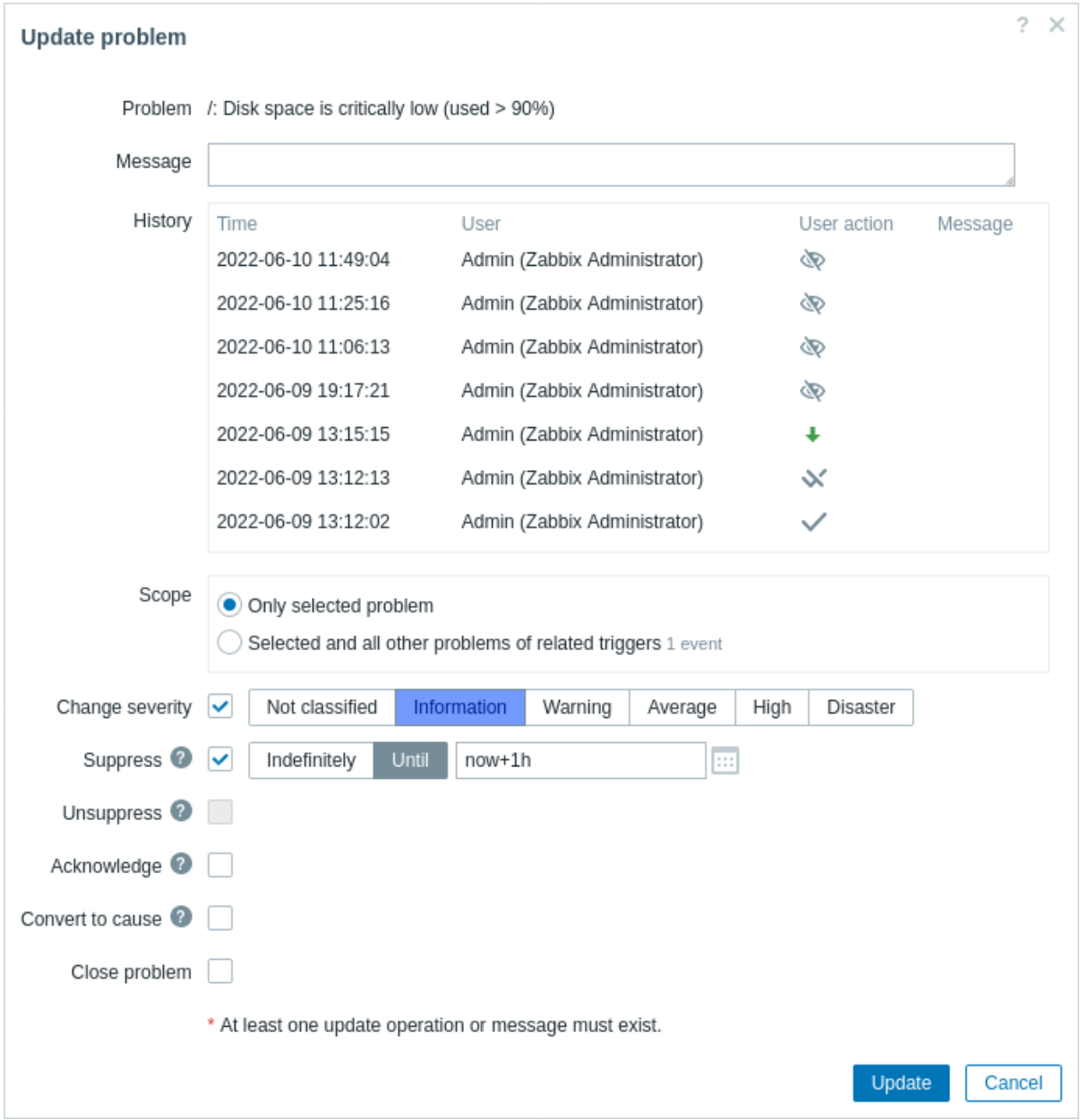
障害を確認するための、障害更新ポップアップウィンドウにアクセスするには、いくつかの方法があります。

- 監視 → 障害で障害を選択した状態で、リストの下の一括更新をクリックします。
- 障害の 更新列にある 更新をクリックします。
  - ダッシュボード (障害と 障害・深刻度ウィジェット)
  - 監視 → 障害
  - 監視 → 障害 → イベント詳細
- 未解決の障害のセルをクリックします。
  - ダッシュボード (トリガー概要ウィジェット)

ポップアップメニューの 更新オプションで、障害更新ウィンドウに遷移します。

障害の更新　障害更新ポップアップでは次のことが可能です。

- 障害に対するコメント
- これまでのコメントとアクションを表示
- 障害の深刻度を変更
- 障害の抑制状態の切り替え
- 障害の確認状態の切り替え
- 症状の障害を障害の原因に変える
- 障害を手動でクローズする



すべての必須入力フィールドには赤いアスタリスクが付いています。

パラメータ	説明
障害	障害が 1 つだけ選択されている場合は障害名が表示されます。 複数の障害が選択されている場合は、*N* 個の障害が選択されていますと表示されます。
メッセージ	障害に対するコメントをテキストで入力してください（最大 2048 文字）。
履歴	障害に対するこれまでの活動やコメントが、時間やユーザーの詳細とともに一覧表示されます。 ユーザーの操作を表すアイコンの意味については、イベント詳細のページをご覧ください。 履歴は更新対象の障害が 1 つだけ選択されている場合に表示されます。
スコープ	障害の深刻度の変更、確認、手動によるクローズなどのアクションの範囲を定義する。 選択した障害のみ - このイベントのみに影響します。 選択した障害と関連するトリガーのすべての障害 - 確認/クローズされた障害がある場合、このイベントと、これまでに確認/クローズされていない他のすべての障害に影響します。スコープにすでに確認またはクローズされた障害が含まれている場合、これらの障害が繰り返し確認/クローズされることはありません。一方、メッセージと深刻度の変更操作の数は制限されていません。
深刻度の変更	チェックボックスをマークし、深刻度ボタンをクリックすると、障害の深刻度が更新されます。 深刻度を変更するチェックボックスは、選択した障害のうち少なくとも 1 つに読み取り/書き込み権限がある場合に使用できます。更新をクリックすると、読み取り/書き込み可能な障害のみが更新されます。 選択したトリガーのいずれにも読み取り/書き込み権限が存在しない場合、チェックボックスは無効になります。

パラメータ	説明
抑制	障害を抑制するにはチェックボックスをオンにします。 **Indefinitely** - 無期限に抑制する **Until** - 一定時間抑制する。時間形式は絶対と相対の両方がサポートされています。例: now+1d - 今から 1 日間 (デフォルト) now/w - 今週末まで 2022-05-28 12:00:00 - 絶対日時まで 単純な期間 (例: '1d'、'1w') はサポートされていないことに注意してください。 このオプションが使用できるかどうかは、" 問題の抑制" のユーザーの役割設定によって異なります。 参照: 障害の抑制
抑制の解除	障害の抑制を解除するには、チェックボックスをオンにします。このチェックボックスは、選択した障害の少なくとも 1 つが抑制されている場合にのみアクティブになります。 このオプションが使用できるかどうかは、" 問題の抑制" のユーザーの役割設定によって異なります。
確認	チェックボックスをマークして、障害を確認します。 このチェックボックスは、選択した障害の中に未確認のものが 1 つでもある場合に有効です。 すでに確認されている障害に対して、さらに確認を追加することはできません（ただし、別のコメントを追加することは可能です）。
未確認	チェックボックスをマークして、障害を未確認状態にします。 このチェックボックスは、選択された中に少なくとも 1 つの確認済の障害がある場合に利用できます。
原因に変更する	症状の障害を障害の原因に変換するには、チェックボックスをオンにします。
障害のクローズ	選択した障害を手動でクローズする場合は、チェックボックスをオンにします。 障害をクローズするためのチェックボックスは、選択した障害の少なくとも 1 つについて、トリガー設定で手動クローズを許可するオプションがチェックされている場合に利用可能です。更新をクリックすると、クローズが許可されている障害のみがクローズされます。 手動でクローズできる障害がない場合、チェックボックスは無効になります。 すでにクローズした障害を、繰り返しクローズすることはできません。

表示　障害の確認情報に基づいて、ダッシュボードまたはマップに障害数を表示できます。設定するには、マップ設定と深刻度別の障害ダッシュボードウィジェットの両方で使用できる障害表示オプションを選択する必要があります。すべての障害数や未確認の障害数を分けて表示することが可能です。

障害の更新情報（確認状況など）をもとに、メッセージの送信やリモートコマンドの実行といった更新操作を設定することができます。

1 障害の抑制

概要

問題の抑制は、後程対処できる問題を一時的に非表示にする方法を提供します。これは最も緊急の問題へ最優先順位を与えるために問題リストを整理するのに役立ちます。

たとえば、週末に問題が発生することがありますが、すぐに対処するほど緊急ではないため、月曜日の朝まで「スヌーズ」することができます。

問題の抑制では、メンテナンスホストのすべての問題が非表示になるホストメンテナンスによる問題の抑制とは対照的に、単一の問題のみを非表示にできます。

抑制された問題に対するトリガーアクションの操作は、ホストメンテナンス の場合と同じように一時停止されます。

構成

問題は、**problem update** ウィンドウから抑制できます。抑制は、コメント、重大度の変更、確認などとともに、問題の更新オプションの 1 つです。

同じ問題の更新ウィンドウから、問題の抑制を解除することもできます。

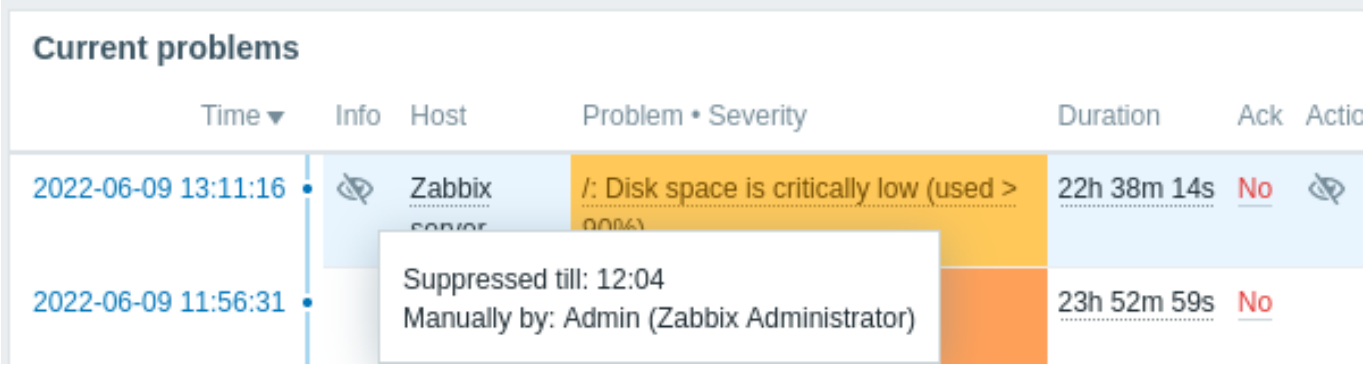
表示

抑制された問題は、情報列に点滅する抑制アイコンでマークされ、その後非表示になります。 

抑制タスクが待機リストにある間、抑制アイコンは点滅します。タスクマネージャーが問題を抑制したら、アイコンの点滅は停止します。抑制アイコンが長時間点滅し続ける場合は、サーバーに問題がある可能性があります。例えば、サーバーがダウンしていてタスクマネージャーがタスクを完了できない場合などです。抑制解除についても同様です。タスクが送信されてからサーバーがタスクを完了していない間は、抑制解除アイコンが点滅します。

抑制された問題は、問題フィルター/ウィジェットの設定に応じて、非表示または表示されることがあります。

問題リストに表示される抑制された問題には抑制アイコンが表示され、マウスオーバーすると抑制の詳細が表示されます。



アクション列の抑制アイコンにマウスを合わせると、抑制の詳細がポップアップ表示されます。

14. 設定のエクスポート/インポート

概要　Zabbix のエクスポート/インポート機能により、Zabbix システム間で様々な設定情報を共有することができます。

この機能のよくある使用例:

- テンプレートやネットワークマップの共有 - Zabbix ユーザ同士での設定パラメータの共有
- Zabbix コミュニティテンプレートにテンプレートをアップロードする - 他のユーザはテンプレートをダウンロードし、そのファイルを Zabbix にインポートできます。
- サードパーティツールとの統合 - 汎用的な YAML、XML、JSON 形式を使用することで、データのインポート/エクスポートを行ってサードパーティのツールやアプリケーションと統合することができます。

エクスポート/インポートできるもの

エクスポート/インポートできるオブジェクトは次の通りです。

- ホストグループ (*Zabbix API* 経由のみ)
- テンプレートグループ (*Zabbix API* 経由のみ)
- テンプレート
- ホスト
- ネットワークマップ
- メディアタイプ
- イメージ

エクスポート形式

Zabbix の Web インターフェースまたはZabbix APIを使用してデータをエクスポートすることができます。サポートされているエクスポート形式は YAML、XML、JSON です。

エクスポートの詳細

- サポートされているすべての要素は、1 つのファイルにエクスポートされます。
- リンクされたテンプレートから継承されたホストおよびテンプレートのエンティティ（アイテム、トリガー、グラフ、ディスカバリルール）はエクスポートされません。ホストレベルでそれらのエンティティに加えられた変更（たとえば、変更されたアイテムの監視間隔、修正された正規表現、またはローレベルディスカバリルールに追加されたプロトタイプ）は、エクスポート時に失われます。インポート時には、リンクされたテンプレート由来のすべてのエンティティが、元のリンクされたテンプレートと同じ状態で再作成されます。
- ローレベルディスカバリによって作成されたエンティティと、それらに依存するすべてのエンティティはエクスポートされません。たとえば、LLD ルールによって生成されたアイテム用に作成されたトリガーはエクスポートされません。
- エクスポートされたホスト/テンプレートにタイムアウトをサポートするエンティティが含まれている場合、それらのエンティティに独自のタイムアウトが設定されていれば、そのタイムアウト値もエクスポートされます。

インポートに関する詳細

- インポートは最初のエラーで停止します。
- イメージのインポート時に既存のイメージを更新する場合、”imagetype” フィールドは無視されます。つまり、インポートによってイメージタイプを変更することはできません。
- ”Delete missing” オプションを使用してホスト/テンプレートをインポートする場合、インポートファイルに存在しないホスト/テンプレートマクロは、インポート後にホスト/テンプレートから削除されます。
- アイテム、トリガー、グラフ、discoveryRules、itemPrototypes、triggerPrototypes、graphPrototypes の空のタグには意味がありません。つまり、存在しない場合と同じです。
- インポートされたホスト/テンプレートのエンティティに独自のタイムアウトが設定されている場合はそれが適用され、そうでない場合はプロキシ/グローバルタイムアウトが適用されます。
- インポートは YAML、XML、JSON をサポートしており、インポートファイルには正しい拡張子が必要です。YAML は.yaml および.yml、XML は.xml、JSON は.json です。サポートされる XML バージョンについては、互換性情報 を参照してください。
- インポートは UTF-8 エンコーディングの設定ファイルのみをサポートします（BOM の有無は問いません）。その他のエンコーディング（UTF16LE、UTF16BE、UTF32LE、UTF32BE など）では、インポート変換エラーが発生します。

YAML の基本形式　YAML エクスポート形式には、次のノードが含まれます。

- Zabbix YAML エクスポートのルートノード
- エクスポートのバージョン

```
zabbix_export:
  version: '7.4'
```

その他のノードは、エクスポートされるオブジェクトに応じて異なります。

XML 形式　XML エクスポート形式には、次のタグが含まれます。

- XML ドキュメントのデフォルトヘッダー
- Zabbix XML エクスポートのルートタグ
- エクスポートのバージョン

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>7.4</version>
</zabbix_export>
```

その他のタグは、エクスポートされるオブジェクトによって異なります。

JSON 形式　JSON エクスポート形式には、以下のオブジェクトが含まれます。

- Zabbix JSON エクスポートのルートオブジェクト
- エクスポートのバージョン

```
{
    "zabbix_export": {
        "version": "7.4"
    }
}
```

その他のオブジェクトは、エクスポートされたオブジェクトに依存します。

1 テンプレートグループ

概要

フロントエンドでは、テンプレートグループはテンプレートエクスポートでのみエクスポートできます。テンプレートをエクスポートすると、そのテンプレートが属するすべてのグループも自動的にエクスポートされます。

API を使用すると、テンプレートグループをテンプレートから独立してエクスポートできます。

エクスポートフォーマット

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Network devices'
```

エクスポートされた要素

要素	タイプ	説明
uuid	string	テンプレートグループのユニーク ID。
name	string	テンプレートグループの名前。

2 ホストグループ

概要

Web インターフェースでは、ホストグループはホストのエクスポートでのみエクスポートできます。ホストがエクスポートされると、それが属するすべてのグループも自動的にエクスポートされます。

API を使用すると、ホストから独立してホストグループをエクスポートできます。

エクスポート形式

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix servers'
```

エクスポートされる要素

要素	タイプ	説明
uuid	文字列	ホストグループのユニーク ID。
name	文字列	グループ名。

3 テンプレート

概要

テンプレートは、多くの関連オブジェクトおよびオブジェクト間の関係とともにエクスポートされます。

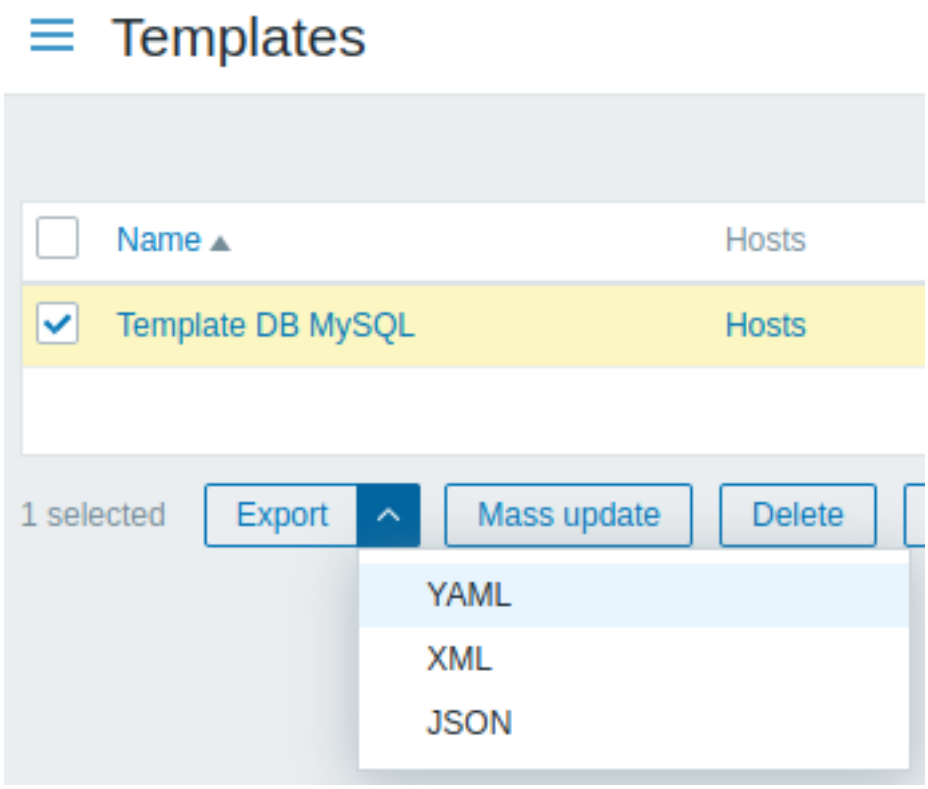
テンプレートのエクスポートには、以下が含まれます。

- リンクされたテンプレートグループ
- リンクされたホストグループ（ホストプロトタイプ設定で使用されている場合）
- テンプレートデータ
- 他のテンプレートへのリンク
- テンプレートグループへのリンク
- 直接リンクされたアイテム
- 直接リンクされたトリガー
- 直接リンクされたグラフ
- 直接リンクされたダッシュボード
- すべてのプロトタイプを含む、直接リンクされたディスカバリルール
- 直接リンクされた Web シナリオ
- 値のマッピング

エクスポート

テンプレートをエクスポートするには、次の手順を実行します。

1. データ収集 → テンプレートに移動します。
2. エクスポートするテンプレートのチェックボックスを選択します。
3. 一覧の下にある エクスポートをクリックします。



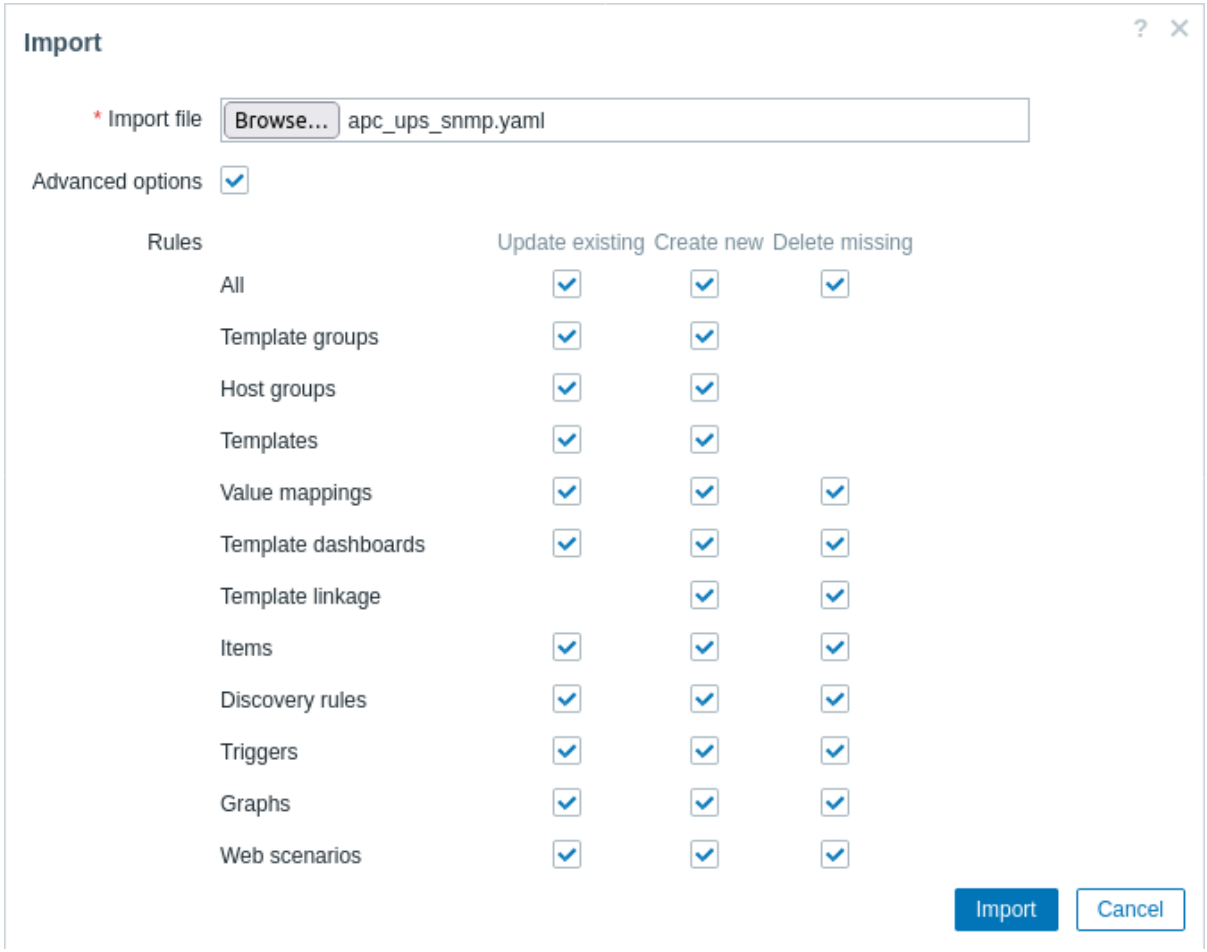
選択した形式に応じて、テンプレートはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_templates.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）
- `zabbix_export_templates.xml` - XML エクスポートの場合
- `zabbix_export_templates.json` - JSON エクスポートの場合

インポート

テンプレートをインポートするには、次の手順を実行します。

1. データ収集 → テンプレートに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. 設定フォームの右下隅にある インポートをクリックします。



詳細オプションチェックボックスをオンにすると、インポート可能なすべての要素の詳細な一覧が表示されます。必要に応じて、各インポートルールをオンまたはオフにしてください。

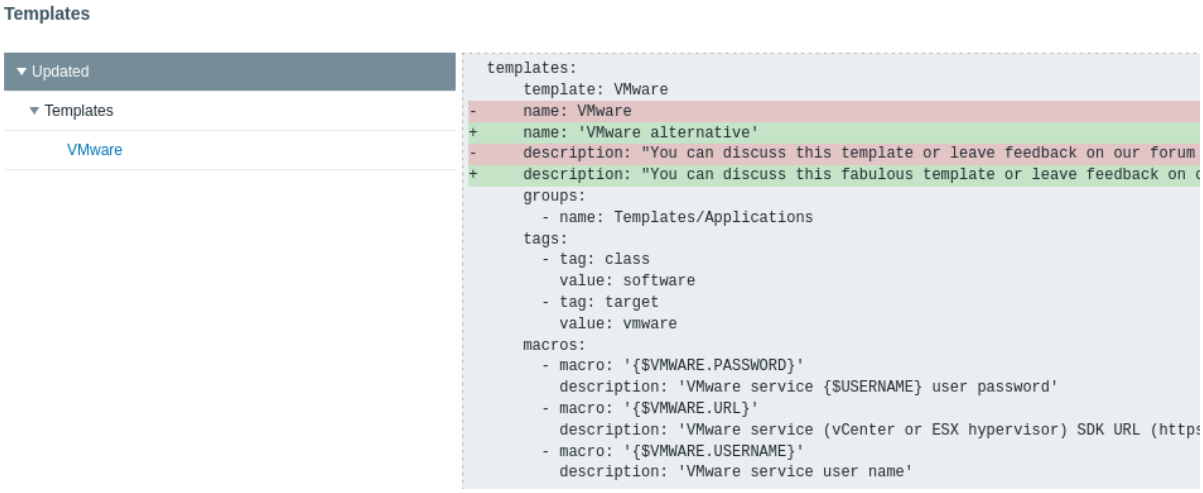
すべて行のチェックボックスをクリックすると、その下にあるすべての要素がオン/オフされます。

インポートルール:

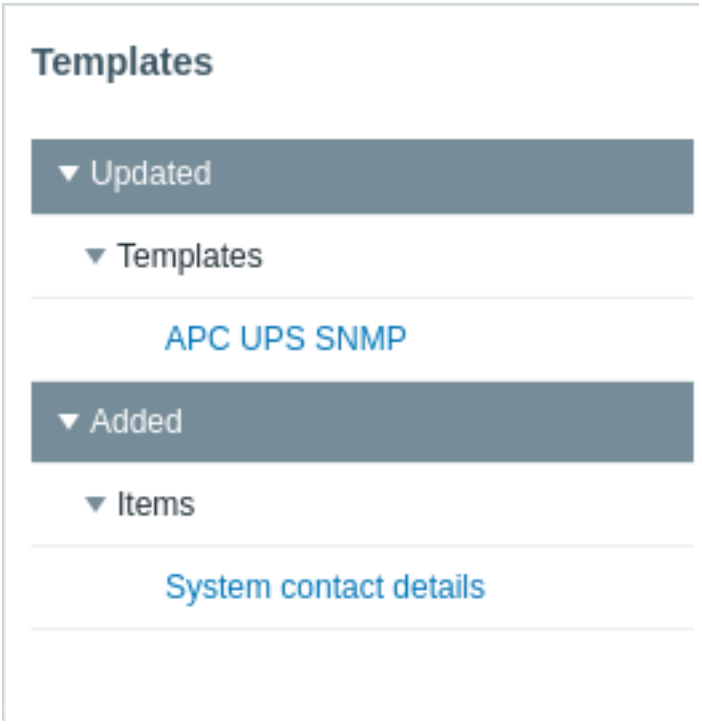
ルール	説明
既存を更新	既存の要素は、インポートファイルのデータを使用して更新されます。そうでない場合、更新されません。
新規作成	新しい要素は、インポートファイルのデータを使用して作成されます。そうでない場合、作成されません。
不足分を削除	インポートファイルに存在しない既存の要素は削除されます。そうでない場合、削除されません。 テンプレートのリンクに対して 不足分を削除がオンになっている場合、インポートファイルに存在しない現在のテンプレートリンクは解除されます。リンク解除されたテンプレートから継承されたエンティティ（アイテム、トリガー、グラフなど）は削除されません（各エンティティに対しても 不足分を削除オプションが選択されている場合を除きます）。

Attention:
同じ名前のテンプレートがすでに存在する場合、クリーンなインポートを実現するには、インポート時に 不足分を削除オプションをチェックする必要があります。こうすることで、更新後のテンプレートに含まれなくなった古いアイテムが削除されます（これらの古いアイテムの履歴が失われることに注意してください）。

次の画面では、インポートされるテンプレートの内容を確認できます。
新しいテンプレートの場合、すべての要素が緑色で表示されます。
既存のテンプレートを更新する場合、新しいテンプレート要素は緑色で強調表示され、削除されたテンプレート要素は赤色で強調表示され、変更されていない要素は灰色の背景で表示されます。



左側のメニューを使用して、変更一覧内を移動できます。
更新済みセクションでは、既存のテンプレート要素に加えられたすべての変更が強調表示されます。
追加済みセクションには、新しいテンプレート要素が一覧表示されます。
各セクション内の要素は要素タイプごとにグループ化されています。灰色の矢印をクリックすると、要素グループを展開または折りたたむことができます。



テンプレートの変更内容を確認したら、インポートをクリックしてテンプレートのインポートを実行します。
インポートの成功または失敗メッセージが Web インターフェースに表示されます。

エクスポート形式　テンプレートをエクスポートした際に、Zabbix は構造化された YAML、JSON、または XML 形式を生成します。エクスポートには、テンプレートメタデータ、アイテム、マクロ、トリガー、ダッシュボードなどのテンプレートの要素が含まれます。

各要素は特定の目的を果たし、ネストされた要素を含む場合があります。

以下のセクションでは、エクスポート形式の各要素について説明します。例では、Linux by Zabbix agent テンプレートを使用しています。

省略記号（...）は、簡潔さを重視して省略された要素を示します。（see table below）という注記は、要素が後述のセクションで詳細に説明されている場合に使用されます。

```
zabbix_export:
  version: '7.4'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
```

```
      name: 'Templates/Operating systems'
  templates: (see table below)
  triggers: (see table below)
  graphs: (see table below)
```

テンプレートのメタデータ

要素		型	説明
version		string	(必須) エクスポートのバージョン。
template_groups			(必須) テンプレートグループのルート要素。
	uuid	string	(必須) このテンプレートグループの一意の識別子。
	name	string	(必須) テンプレートグループ名。
host_groups			ホストプロトタイプで使用されるホストグループのルート要素。
	uuid	string	(必須) このホストグループの一意の識別子。
	name	string	(必須) ホストグループ名。
templates			テンプレート のルート要素。
triggers			テンプレートのトリガー要素のルート要素。これはテンプレートアイテムのトリガー と同じです。
graphs			テンプレートのグラフ要素のルート要素。これはホストグラフ と同じです。

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        これは公式の Linux テンプレートです。Zabbix エージェント 7.4 以降が必要です。(...)
      wizard_ready: 'YES'
      readme: |
        ## 概要

        これは公式の Linux テンプレートです。Zabbix エージェント 7.4 以降が必要です。(...)
      vendor:
        name: Zabbix
        version: 7.4-2
      groups:
        - name: 'Templates/Operating systems'
      items: (以下の表を参照)
      discovery_rules: (以下の表を参照)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'このタイムアウトを過ぎると、エージェントは利用不可と見なされます。Zabbix サーバー/プロキシから到達
          config: (以下の表を参照)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'CPU 使用率の重大しきい値 (% 単位)。'
          config: (以下の表を参照)
        (...)
      dashboards: (以下の表を参照)
      valuemaps: (以下の表を参照)
    (...)
```

テンプレート

要素	Type	説明
uuid	string	(必須) このテンプレートの一意の識別子。
template	string	(必須) 一意のテンプレート名。
name	string	表示用テンプレート名。
description	text	テンプレートの説明。
wizard_ready	text	テンプレートをホストウィザードで選択可能にするかどうか。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。 関連項目: テンプレートオブジェクト (wizard_ready)。
readme	text	ホストウィザードに表示する、テンプレート固有の設定手順。Markdown 形式をサポートします。
vendor		テンプレートベンダーのルート要素 (エクスポートされたテンプレートにベンダーデータが含まれる場合に存在)。
name	string	(必須) テンプレートベンダー名。
version	string	(必須) テンプレートのバージョン。 標準搭載テンプレートでは、バージョンは次の形式で表示されます: Zabbix のメジャーバージョン、区切り文字 ("-")、リビジョン番号 (テンプレートの新しいバージョンごとに増加し、Zabbix のメジャーバージョンごとにリセットされます)。例: 7.0-0、7.0-3、7.4-0、7.4-3。
templates		リンクされたテンプレートのルート要素。
name	string	(必須) テンプレート名。
groups		テンプレートグループのルート要素。
name	string	(必須) テンプレートグループ名。
items		テンプレートアイテムのルート要素。
discovery_rules		テンプレートのローレベルディスカバリルールのルート要素。
httptests		テンプレートの Web シナリオのルート要素。
tags		テンプレートタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
macros		テンプレートユーザーマクロのルート要素。
macro	string	(必須) ユーザーマクロ名。
type	string	ユーザーマクロのタイプ。 指定可能な値:[1] TEXT (0、デフォルト)、SECRET_TEXT (1)、VAULT (2)。 関連項目: ユーザーマクロオブジェクト (type)。
value	string	ユーザーマクロ値。
description	string	ユーザーマクロの説明。
config		ホストウィザードでのマクロの表示方法を制御する、テンプレートマクロ設定のルート要素。
dashboards		テンプレートダッシュボードのルート要素。
valuemaps		テンプレートの値マップのルート要素。

テンプレートアイテム

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix エージェント ping'
          key: agent.ping
          description: 'このアイテムに対して、エージェントは常に "1" を返します。可用性チェックのために `nodata()` と組み
          valuemap:
            name: 'Zabbix エージェント ping ステータス'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'インストール済みパッケージ数'
          key: system.sw.packages.get
          delay: 1h
```

```
      preprocessing: （以下の表を参照）
      tags:
        - tag: component
          value: os
      triggers: （以下の表を参照）
    - uuid: 403cebed115441369e94d35d070ca7b8
      name: 'メモリ使用率'
      type: DEPENDENT
      key: vm.memory.utilization
      value_type: FLOAT
      units: '%'
      description: '使用メモリの割合は `100-pavailable` として計算されます。'
      preprocessing: （以下の表を参照）
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: （以下の表を参照）
    (...)
```

Element	Type	説明
uuid	string	（必須）このアイテムの一意の識別子。
name	string	（必須）アイテム名。
type	string	アイテムタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0、デフォルト)、TRAP (2)、SIMPLE (3)、INTERNAL (5)、ZABBIX_ACTIVE (7)、EXTERNAL (10)、ODBC (11)、IPMI (12)、SSH (13)、TELNET (14)、CALCULATED (15)、JMX (16)、SNMP_TRAP (17)、DEPENDENT (18)、HTTP_AGENT (19)、SNMP_AGENT (20)、ITEM_TYPE_SCRIPT (21)、ITEM_TYPE_BROWSER (22)。
snmp_oid	string	（SNMP_AGENT アイテムで必須）SNMP オブジェクト ID。
key	string	（必須）アイテムキー。
delay	string	アイテムの更新間隔。 デフォルト: 1m。TRAP アイテムでは値は常に 0 になります。
history	string	履歴データを保存する期間（time suffix、user macro またはLLD macro を使用）。 デフォルト: 31d。
trends	string	トレンドデータを保存する期間（time suffix、user macro またはLLD macro を使用）。 デフォルト: 365d。
status	string	アイテムのステータス。 指定可能な値:[1] ENABLED (0、デフォルト)、DISABLED (1)。
value_type	string	受信する値の型。 指定可能な値:[1] FLOAT (0)、CHAR (1)、LOG (2)、UNSIGNED (3、デフォルト)、TEXT (4)、BINARY (5)。
allowed_hosts	string	このアイテムに対してデータ送信を許可するホストの IP アドレスをカンマ区切りで指定したリスト。 TRAP および HTTP_AGENT アイテムでサポートされます。
units	string	受信する値の単位（bps、B など）。
params	text	アイテムタイプに応じた追加パラメータ（SSH および TELNET アイテムでは実行するスクリプト、ODBC アイテムでは SQL クエリ、CALCULATED アイテムでは数式、ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムではスクリプト）。
ipmi_sensor	string	IPMI センサー。 IPMI アイテムでサポートされます。
authtype	string	認証タイプ。 SSH および HTTP_AGENT アイテムでサポートされます。 SSH アイテムで指定可能な値:[1] PASSWORD (0、デフォルト)、PUBLIC_KEY (1)。 HTTP_AGENT アイテムで指定可能な値:[1] NONE (0、デフォルト)、BASIC (1)、NTLM (2)、Kerberos (3)、または Digest (4)。
username	string	（SSH および TELNET アイテムで必須）認証用ユーザー名。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、password（以下参照）も指定するか、両方の要素を空欄のままにする必要があります。

Element	Type	説明
password	string	(SSH および TELNET アイテムで必須) 認証用パスワード。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、username (上記参照) も指定するか、両方の要素を空欄のままにする必要があります。
publickey	string	(SSH アイテムで必須) 公開鍵ファイル名。
privatekey	string	(SSH アイテムで必須) 秘密鍵ファイル名。
description	text	アイテムの説明。
inventory_link	string	アイテムによって設定されるホストインベントリフィールド。 指定可能な値:[1] NONE (0)、ALIAS (4) など (サポートされるフィールドについてはHost inventory を参照)。
valuemap		アイテムの値マッピングのルート要素。
name	string	(必須) アイテムで使用する値マッピング名。
logtimefmt	string	ログエントリ内の時刻の形式。 LOG 値型のアイテムでサポートされます。
preprocessing		アイテム値の前処理のルート要素。
step		テンプレートアイテム値の前処理ステップ のルート要素。
jmx_endpoint	string	JMX エンドポイント。 JMX アイテムでサポートされます。
master_item		(DEPENDENT アイテムで必須) 依存アイテムのマスターアイテムのルート要素。
key	string	(必須) 依存アイテムのマスターアイテムキー。
timeout	string	アイテムデータのポーリング要求タイムアウト。 Timeouts のアイテムタイプ一覧でサポートされます。
url	string	(HTTP_AGENT アイテムで必須) URL 文字列。
query_fields		クエリパラメータのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムで必須) クエリパラメータ名。
value	string	クエリパラメータ値。 HTTP_AGENT アイテムでサポートされます。
parameters		ユーザー定義パラメータのルート要素。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
name	string	(ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムで必須) ユーザー定義パラメータ名。
value	string	ユーザー定義パラメータ値。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
posts	string	HTTP(S) リクエスト本文データ。 HTTP_AGENT アイテムでサポートされます。
status_codes	string	必要な HTTP ステータスコードの範囲をカンマで区切って指定します。 HTTP_AGENT アイテムでサポートされます。
follow_redirects	string	データ取得時にレスポンスのリダイレクトに従うかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0)、YES (1、デフォルト)。
post_type	string	POST データ本文のタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0、デフォルト)、JSON (2)、XML (3)。
http_proxy	string	HTTP(S) プロキシ接続文字列。 HTTP_AGENT アイテムでサポートされます。
headers		HTTP(S) リクエストヘッダーのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムで必須) ヘッダー名。
value	string	(HTTP_AGENT アイテムで必須) ヘッダー値。
retrieve_mode	string	レスポンスのどの部分を保存するか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] BODY (0、デフォルト)、HEADERS (1)、BOTH (2)。
request_method	string	リクエストメソッドのタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] GET (0、デフォルト)、POST (1)、PUT (2)、HEAD (3)。
output_format	string	レスポンスの処理方法。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0、デフォルト)、JSON (1)。
allow_traps	string	trapper アイテムと同様に値を設定できるようにします。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。

Element	Type	説明
ssl_cert_file	string	公開 SSL キーファイルのパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_file	string	秘密 SSL キーファイルのパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_password	string	SSL キーファイルのパスワード。 HTTP_AGENT アイテムでサポートされます。
verify_peer	string	ホストの証明書が正当なものであるかを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
verify_host	string	接続先のホスト名がホストの証明書内のものと一致するかを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
tags		アイテムタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
triggers		テンプレートアイテムトリガー のルート要素。

Note:
関連項目: Item object (一致する名前の関連プロパティを参照してください)。

テンプレートのアイテム値前処理ステップ

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

要素	型	説明
type	string	(必須) アイテム値前処理ステップのタイプ。 指定可能な値:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9、計算式: 受信値 - 前回値), CHANGE_PER_SECOND (10、計算式: (受信値 - 前回値)/(現在時刻 - 前回チェック時刻)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30)。
parameters		(必須) アイテム値前処理ステップのパラメータのルート要素。
parameter	string	アイテム値前処理ステップの個別パラメータ。
error_handler	string	前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値:[1] ORIGINAL_ERROR (0、デフォルト), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3)。
error_handler_params	string	エラーハンドラーのパラメータ。

Note:
関連項目: アイテム前処理オブジェクト（一致する名前の関連プロパティを参照してください）。

テンプレートアイテムのトリガー

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Number of installed packages has been changed'
              priority: WARNING
              manual_close: 'YES'
              tags:
                - tag: scope
                  value: notice
        (...)
        - uuid: 403cebed115441369e94d35d070ca7b8
          (...)
          triggers:
            - uuid: cfd395b1cde74ef18a5e5f840bd5142a
              expression: 'min(/Linux by Zabbix agent/vm.memory.utilization,5m)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: High memory utilization'
              event_name: 'Linux: High memory utilization (>{$MEMORY.UTIL.MAX}% for 5m)'
              priority: AVERAGE
              description: 'The system is running out of free memory.'
              dependencies:
                - name: 'Linux: Lack of available memory'
                  expression: 'max(/Linux by Zabbix agent/vm.memory.size[available],5m)<{$MEMORY.AVAILABLE
              tags:
                - tag: scope
                  value: capacity
                - tag: scope
                  value: performance
        (...)
```

要素	Type	説明
uuid	string	(必須) このトリガーの一意な識別子。
expression	string	(必須) トリガー条件式。
recovery_mode	string	OK イベントを生成する基準。 指定可能な値:[1] EXPRESSION (0, デフォルト), RECOVERY_EXPRESSION (1), NONE (2)。
recovery_expression	string	トリガーの復旧条件式。
correlation_mode	string	相関モード（イベント相関なし、またはタグによるイベント相関）。 指定可能な値:[1] DISABLED (0, デフォルト), TAG_VALUE (1)。
correlation_tag	string	イベント相関に使用するタグ名。
name	string	(必須) トリガー名。
event_name	string	イベント名。
opdata	string	運用データ。
url_name	string	トリガーに関連付けられた URL のラベル。
url	string	トリガーに関連付けられた URL。
status	string	トリガーのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
priority	string	トリガーの深刻度。 指定可能な値:[1] NOT_CLASSIFIED (0, デフォルト), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5)。

要素	Type	説明
description	text	トリガーの説明。
type	string	イベント生成タイプ（単一の障害イベントまたは複数の障害イベント）。 指定可能な値:[1] SINGLE (0, デフォルト), MULTIPLE (1)。
manual_close	string	障害イベントの手動クローズ。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
dependencies		依存関係のルート要素。
name	string	(必須) 依存先トリガー名。
expression	string	(必須) 依存先トリガー条件式。
recovery_expression	string	依存先トリガーの復旧条件式。
tags		トリガータグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
参照: Trigger object（一致する名前の関連プロパティを参照してください）。

テンプレートのローレベルディスカバリルール

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'ネットワークインターフェースのディスカバリ'
          key: net.if.discovery
          delay: 1h
          filter: （下の表を参照）
          description: 'ネットワークインターフェースをディスカバリします。'
          item_prototypes: （下の表を参照）
          trigger_prototypes: （下の表を参照）
          graph_prototypes: （下の表を参照）
        (...)
```

Attention:
テンプレートのローレベルディスカバリルール要素の大部分は、テンプレートアイテム と同じです。以下の表では、テンプレートアイテムと異なる要素について説明します。

Element	Type	Description
type	string	アイテムのタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0, デフォルト), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), JMX (16), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22)。
key	string	(必須) ローレベルディスカバリルールのキー。
filter		テンプレートのローレベルディスカバリルールフィルターのルート要素。
lifetime	string	ディスカバリされなくなったリソースが削除されるまでの期間（秒、time suffix、またはuser macro を使用)。 デフォルト: 7d。
lifetime_type	string	失われた LLD リソースを削除するシナリオ。 指定可能な値: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER。
enabled_lifetime	string	ディスカバリされなくなったリソースが無効化されるまでの期間（秒、time suffix、またはuser macro を使用)。
enabled_lifetime_type	string	失われた LLD リソースを無効化するシナリオ。 指定可能な値: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER。
item_prototypes		テンプレートのアイテムプロトタイプ要素のルート要素。これはテンプレートアイテム と同じです。

Element	Type	Description
trigger_prototypes		テンプレートのトリガープロトタイプ要素のルート要素。これはテンプレートアイテムのトリガー と同じです。
graph_prototypes		テンプレートのグラフプロトタイプ要素のルート要素。これはホストグラフ と同じです。
host_prototypes		テンプレートのホストプロトタイプ要素のルート要素。これはホスト と同じです。
parent_discovery_rule		ローレベルディスカバリルールプロトタイプの親ローレベルディスカバリルール（またはルールプロトタイプ）のルート要素。このプロパティは、参照されたルール/ルールプロトタイプの直接の子である LLD ルールプロトタイプであることを示します。
key	string	（必須）親ローレベルディスカバリルール（またはルールプロトタイプ）のキー。
master_item	string	（DEPENDENT ルールで必須）依存ルールのマスターアイテムのルート要素。
lld_macro_paths		ローレベルディスカバリルールのマクロパスのルート要素。
lld_macro	string	（必須）ローレベルディスカバリルールのマクロ名。
path	string	（必須）対応するマクロに割り当てられる値のセレクター。
preprocessing		ローレベルディスカバリルールの値の前処理のルート要素。
step		ローレベルディスカバリルールの値の前処理ステップ要素のルート要素。これはテンプレートアイテムの値の前処理ステップ と同じですが、指定可能な値はより少なくなります。 参照: LLD rule preprocessing object。
overrides		ローレベルディスカバリルールのオーバーライドルールのルート要素。
name	string	（必須）一意のオーバーライド名。
step	string	（必須）オーバーライドの一意の順序番号。
stop	string	一致した場合に後続のオーバーライドの処理を停止します。
filter		テンプレートのローレベルディスカバリルールのオーバーライドルールフィルター要素のルート要素。これはテンプレートのローレベルディスカバリルールフィルター と同じです。
operations		テンプレートのローレベルディスカバリルールのオーバーライド操作 のルート要素。

Note:
参照: LLD rule object（一致する名前の関連プロパティを参照してください）。

テンプレートのローレベルディスカバリルールのフィルター

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.MATCHES}'
              - macro: '{#IFNAME}'
                value: '{$NET.IF.IFNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
        - uuid: 2bbdc79f082d4c618e01bec625e9c90a
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.MATCHES}'
              - macro: '{#DEVNAME}'
                value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
              - macro: '{#DEVTYPE}'
                value: disk
          (...)
```

Element	Type	Description
evaltype	string	フィルター条件の評価方法を上書きします。 指定可能な値:[1] AND_OR (0、デフォルト)、AND (1)、OR (2)、FORMULA (3)。
formula	string	フィルター条件用のカスタム計算式。
conditions		フィルター条件のルート要素。
macro	string	(必須) チェック対象のローレベルディスカバリマクロ名。
value	string	比較する値。
operator	string	条件演算子。 指定可能な値:[1] MATCHES_REGEX (8、デフォルト)、NOT_MATCHES_REGEX (9)。
formulaid	string	(必須) カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際、この ID はユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。

Note:
関連項目: LLD rule filter object (一致する名前の関連プロパティを参照してください)。

テンプレートのローレベルディスカバリルールのオーバーライド操作

要素	型	説明
operationobject	string	操作が適用されるオブジェクト。 指定可能な値:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3)。
operator	string	オーバーライド条件の演算子。 指定可能な値:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6)。
value	string	オーバーライド条件の演算子に使用する正規表現または文字列。
status	string	オーバーライド操作時のオブジェクトのステータス。
discover	string	ディスカバリの結果としてオブジェクトを追加するかどうか。
delay	string	オーバーライド操作時にアイテムのプロトタイプに設定される更新間隔。
history	string	オーバーライド操作時にアイテムのプロトタイプに設定される履歴保存期間。
trends	string	オーバーライド操作時にアイテムのプロトタイプに設定されるトレンド保存期間。
severity	string	オーバーライド操作時に設定されるトリガーのプロトタイプの深刻度。
tags		オーバーライド操作時にオブジェクトに設定されるタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグの値。
templates		オーバーライド操作時にホストのプロトタイプにリンクされるテンプレートのルート要素。
name	string	(必須) テンプレート名。
inventory_mode	string	オーバーライド操作時に設定されるホストのプロトタイプのインベントリモード。

Note:
参照: LLD rule override operation object (一致する名前の関連プロパティを参照してください)。

テンプレートの Web シナリオ

要素	Type	説明
uuid	string	(必須) この Web シナリオの一意な識別子。
name	string	(必須) Web シナリオ名。
delay	string	Web シナリオの実行頻度 (秒、time suffix、またはuser macro を使用)。 デフォルト: 1m。
attempts	integer	Web シナリオのステップを実行する試行回数。 指定可能な値:[1] 1～10 (デフォルト: 1)。
agent	string	クライアントエージェント。Zabbix は選択したブラウザーを装って動作します。 これは、Web サイトがブラウザーごとに異なるコンテンツを返す場合に便利です。 デフォルト: Zabbix。
http_proxy	string	Web シナリオで使用されるプロキシ。形式: `http://[username[:password]@]proxy.example.com[:port]`

要素	Type	説明
variables		シナリオのステップで使用できる Web シナリオ変数のルート要素。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。 ヘッダーは、HTTP プロトコルで表示されるものと同じ構文で記述する必要があります。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
status	string	Web シナリオのステータス。 指定可能な値:[1] ENABLED (0、デフォルト)、DISABLED (1)。
authentication	string	認証方式。 指定可能な値:[1] NONE (0、デフォルト)、BASIC (1)、NTLM (2)、Kerberos (3)、または Digest (4)。
http_user	string	BASIC (HTTP)、NTLM、Kerberos、または Digest 認証で使用するユーザー名。
http_password	string	BASIC (HTTP)、NTLM、Kerberos、または Digest 認証で使用するパスワード。
verify_peer	string	Web サーバーの SSL 証明書を検証します。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
verify_host	string	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証します。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
ssl_cert_file	string	クライアント認証に使用する SSL 証明書ファイル名（PEM 形式である必要があります）。
ssl_key_file	string	クライアント認証に使用する SSL 秘密鍵ファイル名（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵ファイルのパスワード。
steps		(必須) テンプレートの Web シナリオステップ のルート要素。
tags		Web シナリオタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
関連項目: Web scenario object（一致する名前の関連プロパティを参照してください）。

テンプレート Web シナリオのステップ

要素	Type	説明
name	string	(必須) Web シナリオステップ名。
url	string	(必須) 監視対象の URL。
query_fields		クエリパラメータのルート要素（リクエスト実行時に URL へ追加される HTTP フィールドの配列）。
name	string	(必須) クエリパラメータ名。
value	string	クエリパラメータ値。
posts		HTTP POST 変数のルート要素（文字列（生の POST データ）または HTTP フィールドの配列（フォームフィールドデータ））。
name	string	(必須) POST フィールド名。
value	string	(必須) POST フィールド値。
variables		このステップの後に適用されるステップレベルの変数（マクロ）のルート要素。 変数値に'regex:' プレフィックスがある場合、その値は'regex:' プレフィックスに続く正規表現パターンに従って、このステップで返されたデータから抽出されます。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
follow_redirects	string	HTTP リダイレクトに従います。 指定可能な値:[1] NO (0)、YES (1、デフォルト)。
retrieve_mode	string	HTTP レスポンスの取得モード。 指定可能な値:[1] BODY (0、デフォルト)、HEADERS (1)、BOTH (2)。
timeout	string	ステップ実行のタイムアウト（秒、time suffix、またはuser macro を使用)。 デフォルト: 15s。
required	string	レスポンス内に存在する必要があるテキスト（空の場合は無視されます)。

要素	Type	説明
status_codes	string	許可される HTTP ステータスコードのカンマ区切りリスト（例: 200-201,210-299。空の場合は無視されます）。

Note:
参照: Web scenario step object（一致する名前の関連プロパティを参照してください）。

テンプレートマクロ設定

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: '最後に Zabbix エージェントが確認されてからの秒数'
            description: 'エージェントが利用不可と見なされるまでのタイムアウト。'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Thresholds
            label: 'CPU 使用率のしきい値'
            description: 'CPU 使用率の重大なしきい値 (%)。0 から 100 までの範囲です。'
            regex: '^-?([0-9]+|([0-9]+)\.([0-9]+))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'インターフェース制御'
            description: 'インターフェースの動作状態が "Link down" に変化したときにトリガーを生成します。'
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

要素	Type	説明
type	string	(必須）マクロ入力フィールドのタイプ。 使用可能な値:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3)。
priority	string	マクロ一覧内でのマクロの位置。
section_name	string	マクロがグループ化される折りたたみ可能なセクションのラベル。
label	string	(TEXT、LIST、CHECKBOX マクロで必須）マクロのラベル。
description	text	マクロのヘルプテキスト。Markdown 形式をサポートします。
required	string	マクロを必須としてマークします。 使用可能な値:[1] NO (0), YES (1)。 TEXT および LIST マクロでサポートされます。
regex	string	テキストボックスフィールド内のユーザー入力を検証するための正規表現。 TEXT マクロでサポートされます。
options		LIST 項目または CHECKBOX 値のルート要素。
value	string	(LIST マクロで必須）LIST 項目の値。
text	string	(LIST マクロで必須）LIST 項目のテキスト。

要素	Type	説明
checked	string	(CHECKBOX マクロで必須) チェック状態を表す値。 使用可能な値:[1] 0 (false), 1 (true)。 CHECKBOX マクロでサポートされます。
unchecked	string	(CHECKBOX マクロで必須) 未チェック状態を表す値。 使用可能な値:[1] 0 (false), 1 (true)。 CHECKBOX マクロでサポートされます。

Note:
関連項目: Macro configuration object (一致する名前の関連プロパティを参照してください)。

テンプレートダッシュボード

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (以下の表を参照)
        (...)
```

要素		型	説明
uuid		string	(必須) このダッシュボードの一意の識別子。
name		string	(必須) テンプレートダッシュボード名。
display period		integer	ダッシュボードページの表示期間。
auto_start		string	スライドショーの自動開始。 指定可能な値:[1] NO (0), YES (1、デフォルト)。
pages			テンプレートダッシュボードページのルート要素。
	name	string	ページ名。
	display period	integer	ページの表示期間。
	sortorder	integer	ページの並び順。
widgets			テンプレートダッシュボードウィジェットのルート要素。

Note:
関連項目: テンプレートダッシュボードオブジェクト (一致する名前の関連プロパティを参照してください)。

テンプレートダッシュボードウィジェット

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
```

```
                  - type: GRAPH_PROTOTYPE
                    name: graphid.0
                    value:
                      host: 'Linux by Zabbix agent'
                      name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                  - type: STRING
                    name: reference
                    value: AAAEB
                - type: graphprototype
                  'y': '5'
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '1'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

要素	Type	説明
type	string	(必須) ウィジェットタイプ。
name	string	ウィジェット名。
x	integer	テンプレートダッシュボードの左側からの水平位置。 指定可能な値:[1] 0-71。
y	integer	テンプレートダッシュボードの上側からの垂直位置。 指定可能な値:[1] 0-63。
width	integer	ウィジェットの幅。 指定可能な値:[1] 1-72。
height	integer	ウィジェットの高さ。 指定可能な値:[1] 1-64。
hide_header	string	ウィジェットヘッダーを非表示にします。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
fields		テンプレートダッシュボードウィジェットフィールドのルート要素。
type	string	(必須) ウィジェットフィールドタイプ。 指定可能な値:[1] INTEGER (0)、STRING (1)、ITEM (4)、ITEM_PROTOTYPE (5)、GRAPH (6)、GRAPH_PROTOTYPE (7)、MAP (8)、SERVICE (9)、SLA (10)、USER (11)、ACTION (12)、MEDIA_TYPE (13)。
name	string	(必須) ウィジェットフィールド名。
value	mixed	(必須) フィールドタイプに応じたウィジェットフィールド値。

Note:
参照: Template dashboard widget object (一致する名前の関連プロパティを参照してください)。

テンプレートの値マップ

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
```

```
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
              newvalue: up
        (...)
```

要素	Type	説明
uuid	string	(必須) この値マップの一意の識別子。
name	string	(必須) 値マップ名。
mapping		マッピングのルート要素。
type	string	マッピングの一致タイプ。 使用可能な値:[1] EQUAL (0、デフォルト)、GREATER_OR_EQUAL (2)、LESS_OR_EQUAL (3)、IN_RANGE (4)、REGEXP (5)、DEFAULT (6)。
value	string	元の値。
newvalue	string	(必須) 元の値がマッピングされる先の値。

Note:
参照: Value map object（一致する名前の関連プロパティを参照してください）。

脚注 [1] かっこ内の API integer 値（例: ENABLED (0)）は、参照用としてのみ記載されています。詳細については、表のエントリ内のリンク先 API オブジェクトページ、または各セクションの末尾を参照してください。

4 ホスト

概要

ホストは、多くの関連オブジェクトおよびオブジェクト間の関係とともにエクスポートされます。

ホストのエクスポートには、以下が含まれます。

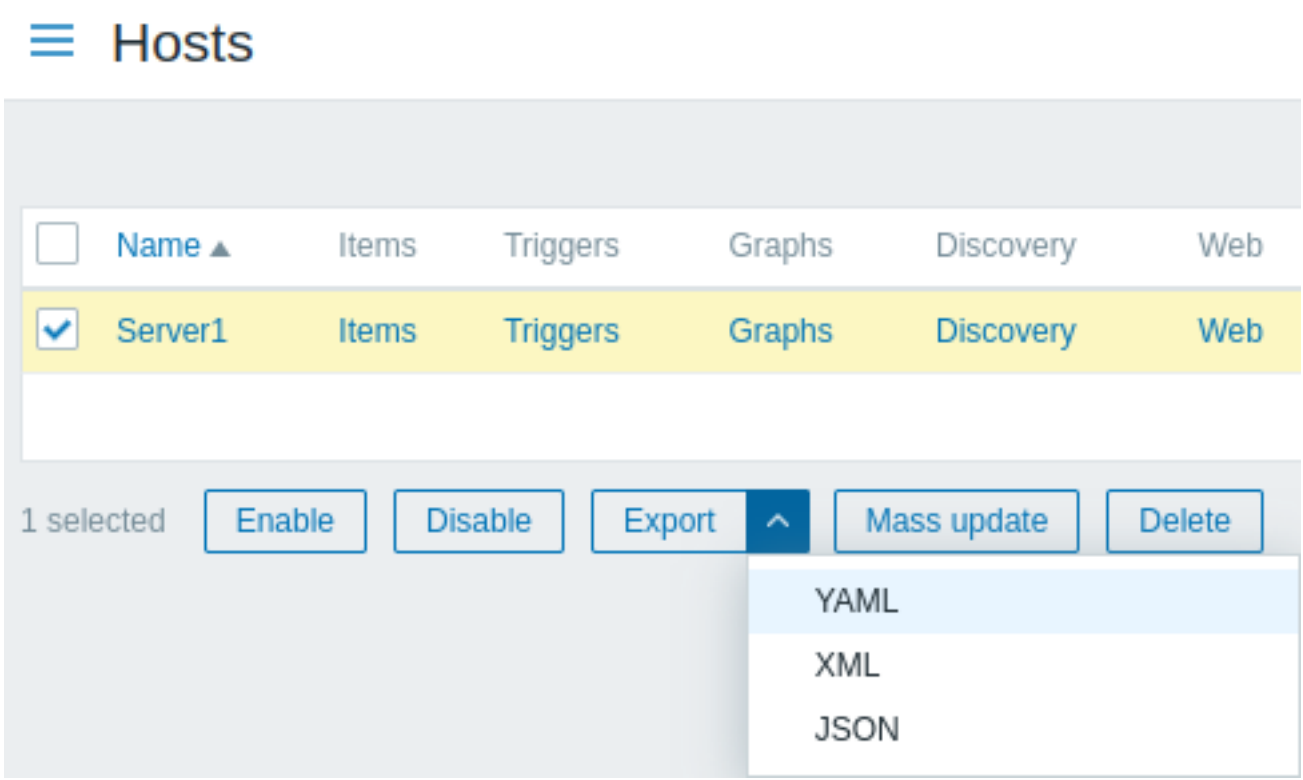
- リンクされたホストグループ
- ホストデータ
- テンプレートのリンク
- ホストグループのリンク
- ホストインターフェース
- 直接リンクされたアイテム
- 直接リンクされたトリガー
- すべてのプロトタイプを含む、直接リンクされたディスカバリルール
- 直接リンクされた Web シナリオ
- ホストマクロ
- ホストインベントリデータ
- 値マップ
- リンクされたグラフ

エクスポート

ホストをエクスポートするには、次の手順を実行します。

1. データ収集 → ホストに移動します。
2. エクスポートするホストのチェックボックスをオンにします。

3. 一覧の下にある エクスポートをクリックします。



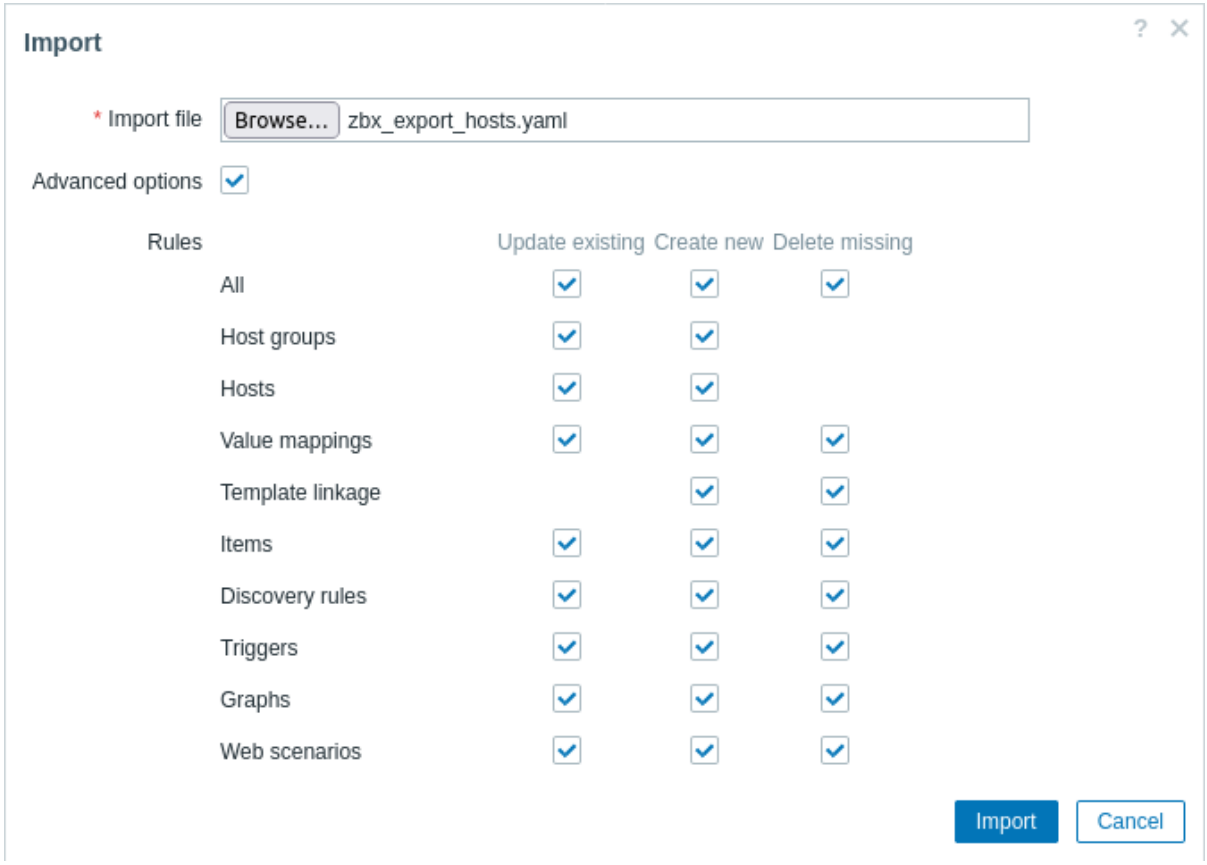
選択した形式に応じて、ホストはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_hosts.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）
- `zabbix_export_hosts.xml` - XML エクスポートの場合
- `zabbix_export_hosts.json` - JSON エクスポートの場合。

インポート

ホストをインポートするには、次の手順を実行します。

1. データ収集 → ホストに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. 設定フォームの右下隅にある インポートをクリックします。



詳細オプションチェックボックスをオンにすると、インポート可能なすべての要素の詳細な一覧が表示されます。必要に応じて各インポートルールをオンまたはオフにしてください。

すべて行のチェックボックスをクリックすると、その下にあるすべての要素がオン/オフされます。

インポートルール:

Rule	Description
既存を更新	既存の要素は、インポートファイルのデータを使用して更新されます。そうでない場合は、更新されません。
新規作成	新しい要素は、インポートファイルのデータを使用して作成されます。そうでない場合は、作成されません。
存在しないものを削除	インポートファイルに存在しない既存の要素は削除されます。そうでない場合は、削除されません。 テンプレートのリンクに対して 存在しないものを削除がオンになっている場合、インポートファイルに存在しない現在のテンプレートリンクは解除されます。リンク解除されたテンプレートから継承されたエンティティ（アイテム、トリガー、グラフなど）は削除されません（各エンティティに対しても 存在しないものを削除オプションが選択されている場合を除きます）。

インポートの成功または失敗のメッセージが Web インターフェースに表示されます。

エクスポート形式

ホストがエクスポートされると、Zabbix は構造化された YAML、JSON、または XML 形式を生成します。エクスポートには、ホストのメタデータ、アイテム、マクロ、トリガーなどのホスト要素が含まれます。

各要素は特定の目的を持ち、ネストされた要素を含む場合があります。

以下のセクションでは、エクスポート形式の各要素について説明します。例では、MySQL by Zabbix agent 2 テンプレートを使用するホストを用いています。要素の説明を目的として、ホストの作成後にテンプレートのリンクを解除しています。リンクを解除しても、すべての要素は設定内に保持されます（リンク解除とクリアを行った場合とは異なり、その場合は要素が削除されます）。

省略記号（`...`）は、簡潔にするために省略された要素を示します。注記（`see table below`）は、その要素が後続のセクションでより詳しく説明されている場合に使用されます。

```
zabbix_export:
  version: '7.4'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: （以下の表を参照）
  graphs: （以下の表を参照）
```

ホストメタデータ

要素		Type	説明
version		string	(必須) Zabbix のバージョン。
host_groups			(必須) ホストグループのルート要素。
	uuid	string	(必須) このホストグループの一意の識別子。
	name	string	(必須) ホストグループ名。
hosts			ホスト のルート要素。
graphs			ホストグラフ のルート要素。

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: （以下の表を参照）
      items: （以下の表を参照）
      discovery_rules: （以下の表を参照）
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL user password.'
        (...)
      valuemaps: （以下の表を参照）
  (...)
```

ホスト

Element	Type	説明
host	string	（必須）一意のホスト名。
name	string	表示用ホスト名。
description	text	ホストの説明。
monitored_by	string	ホストの監視方法。 指定可能な値:[1] SERVER (0、デフォルト)、PROXY (1)、または PROXY_GROUP (2)。
proxy		プロキシのルート要素。
name	string	（必須）ホストを監視するプロキシの名前（存在する場合）。
proxy_group		プロキシグループのルート要素。
name	string	（必須）ホストの監視に使用されるプロキシグループの名前（存在する場合）。
status	string	ホストのステータス。 指定可能な値:[1] ENABLED (0、デフォルト)、DISABLED (1)。

Element	Type	説明
ipmi_authtype	string	IPMI セッションの認証タイプ。 指定可能な値:[1] DEFAULT (-1、デフォルト)、NONE (0)、MD2 (1)、MD5 (2)、STRAIGHT (4)、OEM (5)、RMCP_PLUS (6)。
ipmi_privilege	string	IPMI セッションの権限レベル。 指定可能な値:[1] CALLBACK (1)、USER (2、デフォルト)、OPERATOR (3)、ADMIN (4)、OEM (5)。
ipmi_username	string	IPMI チェック用のユーザー名。
ipmi_password	string	IPMI チェック用のパスワード。
templates		リンクされたテンプレートのルート要素。
name	string	(必須) テンプレート名。
groups		ホストグループのルート要素。ホストが所属するホストグループを指定します。
name	string	(必須) ホストグループ名。
interfaces		ホストインターフェース のルート要素。
items		ホストアイテム のルート要素。
discovery_rules		ホストのローレベルディスカバリルール のルート要素。
http_tests		ホストの Web シナリオ のルート要素。
tags		ホストタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
macros		ホストマクロのルート要素。
macro	string	(必須) ユーザーマクロ名。
type	string	ユーザーマクロのタイプ。 指定可能な値:[1] TEXT (0、デフォルト)、SECRET_TEXT (1)、VAULT (2)。
value	string	ユーザーマクロ値。
description	string	ユーザーマクロの説明。
inventory		ホストインベントリのルート要素。
<inventory_property>	string	インベントリプロパティ。すべてのプロパティには対応する要素があります (type、name、os など。例についてはExport format を参照)。
inventory_mode	string	インベントリモード。 指定可能な値:[1] DISABLED (-1)、MANUAL (0、デフォルト)、AUTOMATIC (1)。
valuemaps		ホストの値マップ のルート要素。

Note:
関連項目: Host object (一致する名前の関連プロパティを参照してください)。

ホストインターフェース

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          interface_ref: if1
      (...)
```

要素	Type	説明
default	string	これがプライマリホストインターフェースであるかどうか。1 つのホスト上では、1 つのタイプにつきプライマリインターフェースは 1 つだけ設定できます。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
type	string	インターフェースのタイプ。 指定可能な値:[1] ZABBIX (1, デフォルト), SNMP (2), IPMI (3), JMX (4)。
useip	string	ホストへの接続に IP をインターフェースとして使用するかどうか (使用しない場合は DNS が使用されます)。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
ip	string	(IP 接続で必須) IP アドレス (IPv4 または IPv6)。
dns	string	(DNS 接続で必須) DNS 名。
port	string	ポート番号。

要素	Type	説明
details		インターフェース詳細のルート要素。
version	string	使用する SNMP バージョン。 指定可能な値:[1] SNMPV1 (1), SNMP_V2C (2, デフォルト), SNMP_V3 (3)。
community	string	(SNMPv1 および SNMPv2 アイテムで必須) SNMP コミュニティ。
max_repetitions	string	ネイティブ SNMP バルクリクエスト (GetBulkRequest-PDUs) の最大繰り返し値。 SNMPv2 および SNMPv3 アイテム (`discovery[]` および `walk[]` アイテム) でサポートされます。 デフォルト: 10。
contextname	string	SNMPv3 コンテキスト名。 SNMPv3 アイテムでサポートされます。
securityname	string	SNMPv3 セキュリティ名。 SNMPv3 アイテムでサポートされます。
securitylevel	string	SNMPv3 セキュリティレベル。 SNMPv3 アイテムでサポートされます。 指定可能な値:[1] NOAUTHNOPRIV (0, デフォルト), AUTHNOPRIV (1), AUTHPRIV (2)。
authprotocol	string	SNMPv3 認証プロトコル。 SNMPv3 アイテムでサポートされます。 指定可能な値:[1] MD5 (0, デフォルト), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5)。
authpassphrase	string	SNMPv3 認証パスフレーズ。 SNMPv3 アイテムでサポートされます。
privprotocol	string	SNMPv3 プライバシープロトコル。 SNMPv3 アイテムでサポートされます。 指定可能な値:[1] DES (0, デフォルト), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5)。
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。 SNMPv3 アイテムでサポートされます。
bulk	string	SNMP にバルクリクエストを使用するかどうか。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
interface_ref	string	アイテムで使用するインターフェース参照名 (形式: `if<N>`)。

Note:
参照: Host interface object (一致する名前の関連プロパティを参照してください)。

ホストのアイテム

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: '`binlog_cache_size` に収まらないため、通常のバイナリログキャッシュではなく一時ディスクキャッシュ
          preprocessing: (下表を参照)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Buffer pool utilization'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
```

```
      params: |
        ( last(//mysql.innodb_buffer_pool_pages_total) -
        last(//mysql.innodb_buffer_pool_pages_free) ) /
        ( last(//mysql.innodb_buffer_pool_pages_total) +
        ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
        ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
      description: 'バッファプール内の総ページ数に対する使用済みページ数の比率。'
      tags:
        - tag: component
          value: memory
      triggers: (下表を参照)
    (...)
    - name: Uptime
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'サーバーが稼働している秒数。'
      preprocessing: (下表を参照)
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: (下表を参照)
    (...)
```

Element	Type	説明
name	string	(必須) アイテム名。
type	string	アイテムタイプ。 指定可能な値:[1] ZABBIX_PASSIVE (0, デフォルト), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23)。
snmp_oid	string	(SNMP_AGENT アイテムで必須) SNMP オブジェクト ID。
key	string	(必須) アイテムキー。
delay	string	アイテムの更新間隔。 デフォルト: 1m。TRAP アイテムでは値は常に 0 になります。
history	string	履歴データを保存する期間（time suffix、user macro またはLLD macro を使用)。 デフォルト: 31d。
trends	string	トレンドデータを保存する期間（time suffix、user macro またはLLD macro を使用)。 デフォルト: 365d。
status	string	アイテムのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。
value_type	string	受信する値の型。 指定可能な値:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, デフォルト), TEXT (4), BINARY (5)。
allowed_hosts	string	このアイテムに対してデータ送信を許可するホストの IP アドレスをカンマ区切りで指定したリスト。 TRAP および HTTP_AGENT アイテムでサポートされます。
units	string	受信する値の単位（bps、B など)。
params	text	アイテムタイプに応じた追加パラメータ（SSH および TELNET アイテムでは実行するスクリプト、ODBC アイテムでは SQL クエリ、CALCULATED アイテムでは数式、ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムではスクリプト)。
ipmi_sensor	string	IPMI センサー。 IPMI アイテムでサポートされます。
authtype	string	認証タイプ。 SSH および HTTP_AGENT アイテムでサポートされます。 SSH アイテムで指定可能な値:[1] PASSWORD (0, デフォルト), PUBLIC_KEY (1)。 HTTP_AGENT アイテムで指定可能な値:[1] NONE (0, デフォルト), BASIC (1), NTLM (2), Kerberos (3), または Digest (4)。

Element	Type	説明
username	string	(SSH および TELNET アイテムで必須) 認証用ユーザー名。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、password（下記参照）も指定するか、両方の要素を空欄のままにする必要があります。
password	string	(SSH および TELNET アイテムで必須) 認証用パスワード。 SIMPLE、ODBC、JMX、HTTP_AGENT アイテムでサポートされます。 JMX アイテムで使用する場合は、username（上記参照）も指定するか、両方の要素を空欄のままにする必要があります。
publickey	string	(SSH アイテムで必須) 公開鍵ファイル名。
privatekey	string	(SSH アイテムで必須) 秘密鍵ファイル名。
description	text	アイテムの説明。
inventory_link	string	アイテムによって設定されるホストインベントリフィールド。 指定可能な値:[1] NONE (0), ALIAS (4) など（サポートされるフィールドについてはHost inventory を参照）。
valuemap		アイテムの値マップのルート要素。
name	string	(必須) アイテムで使用する値マップ名。
logtimefmt	string	ログエントリ内の時刻の形式。 LOG 値型のアイテムでサポートされます。
preprocessing		アイテム値の前処理のルート要素。
step		ホストのアイテム値前処理ステップ のルート要素。
interface_ref	string	ホストインターフェースへの参照（形式: if<N>）。
jmx_endpoint	string	JMX エンドポイント。 JMX アイテムでサポートされます。
master_item		(DEPENDENT アイテムで必須) 依存アイテムのマスターアイテムのルート要素。
key	string	(必須) 依存アイテムのマスターアイテムキー。
timeout	string	アイテムデータのポーリング要求タイムアウト。 Timeouts のアイテムタイプ一覧でサポートされます。
url	string	(HTTP_AGENT アイテムで必須) URL 文字列。
query_fields		クエリパラメータのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムで必須) クエリパラメータ名。
value	string	クエリパラメータ値。 HTTP_AGENT アイテムでサポートされます。
parameters		ユーザー定義パラメータのルート要素。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
name	string	(ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムで必須) ユーザー定義パラメータ名。
value	string	ユーザー定義パラメータ値。 ITEM_TYPE_SCRIPT および ITEM_TYPE_BROWSER アイテムでサポートされます。
posts	string	HTTP(S) リクエスト本文データ。 HTTP_AGENT アイテムでサポートされます。
status_codes	string	必要な HTTP ステータスコードの範囲をカンマ区切りで指定します。 HTTP_AGENT アイテムでサポートされます。
follow_redirects	string	データ取得時にレスポンスのリダイレクトに従うかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0), YES (1, デフォルト)。
post_type	string	POST データ本文のタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0, デフォルト), JSON (2), XML (3)。
http_proxy	string	HTTP(S) プロキシ接続文字列。 HTTP_AGENT アイテムでサポートされます。
headers		HTTP(S) リクエストヘッダーのルート要素。 HTTP_AGENT アイテムでサポートされます。
name	string	(HTTP_AGENT アイテムで必須) ヘッダー名。
value	string	(HTTP_AGENT アイテムで必須) ヘッダー値。
retrieve_mode	string	レスポンスのどの部分を保存するか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] BODY (0, デフォルト), HEADERS (1), BOTH (2)。
request_method	string	リクエストメソッドのタイプ。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] GET (0, デフォルト), POST (1), PUT (2), HEAD (3)。

Element	Type	説明
output_format	string	レスポンスの処理方法。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] RAW (0, デフォルト), JSON (1)。
allow_traps	string	trapper アイテムと同様に値を設定できるようにします。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
ssl_cert_file	string	公開 SSL キーファイルのパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_file	string	秘密 SSL キーファイルのパス。 HTTP_AGENT アイテムでサポートされます。
ssl_key_password	string	SSL キーファイルのパスワード。 HTTP_AGENT アイテムでサポートされます。
verify_peer	string	ホストの証明書が正当なものであることを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
verify_host	string	接続先のホスト名がホストの証明書内のものと一致することを検証するかどうか。 HTTP_AGENT アイテムでサポートされます。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
tags		アイテムタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。
triggers		ホストのアイテムトリガー のルート要素。

Note:
関連項目: Item object（一致する名前の関連プロパティを参照）。

ホストのアイテム値の前処理ステップ

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.Binlog_cache_disk_use
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 6h
          (...)
```

要素	Type	説明
type	string	(必須) アイテム値の前処理ステップのタイプ。 使用可能な値:[1] MULTIPLIER (1), RTRIM (2), LTRIM (3), TRIM (4), REGEX (5), BOOL_TO_DECIMAL (6), OCTAL_TO_DECIMAL (7), HEX_TO_DECIMAL (8), SIMPLE_CHANGE (9、計算式: 受信値 - 前回値), CHANGE_PER_SECOND (10、計算式: (受信値 - 前回値)/(現在時刻 - 前回チェック時刻)), XMLPATH (11), JSONPATH (12), IN_RANGE (13), MATCHES_REGEX (14), NOT_MATCHES_REGEX (15), CHECK_JSON_ERROR (16), CHECK_XML_ERROR (17), CHECK_REGEX_ERROR (18), DISCARD_UNCHANGED (19), DISCARD_UNCHANGED_HEARTBEAT (20), JAVASCRIPT (21), PROMETHEUS_PATTERN (22), PROMETHEUS_TO_JSON (23), CSV_TO_JSON (24), STR_REPLACE (25), CHECK_NOT_SUPPORTED (26), XML_TO_JSON (27), SNMP_WALK_VALUE (28), SNMP_WALK_TO_JSON (29), SNMP_GET_VALUE (30)。
parameters		(必須) アイテム値の前処理ステップのパラメータのルート要素。

要素	Type	説明
parameter	string	アイテム値の前処理ステップの個別パラメータ。
error_handler	string	前処理ステップの失敗時に使用されるアクションタイプ。 使用可能な値:[1] ORIGINAL_ERROR (0、デフォルト), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3)。
error_handler_params	string	エラーハンドラーのパラメータ。

Note:
参照: アイテム前処理オブジェクト（一致する名前の関連プロパティを参照してください）。

ホストのアイテムのトリガー

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: Uptime
          (...)
          triggers:
            - expression: 'nodata(/MySQL server/mysql.uptime,30m)=1'
              name: 'MySQL: Failed to fetch info data'
              event_name: 'MySQL: Failed to fetch info data (or no data for 30m)'
              priority: INFO
              description: 'Zabbix has not received any data for items for the last 30 minutes.'
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

要素	Type	説明
uuid	string	(必須) このトリガーの一意な識別子。
expression	string	(必須) トリガー式。
recovery_mode	string	OK イベントを生成する基準。 指定可能な値:[1] EXPRESSION (0, デフォルト), RECOVERY_EXPRESSION (1), NONE (2)。
recovery_expression	string	トリガーの復旧式。
correlation_mode	string	相関モード（イベント相関なし、またはタグによるイベント相関）。 指定可能な値:[1] DISABLED (0, デフォルト), TAG_VALUE (1)。
correlation_tag	string	イベント相関に使用するタグ名。
name	string	(必須) トリガー名。
event_name	string	イベント名。
opdata	string	運用データ。
url_name	string	トリガーに関連付けられた URL のラベル。
url	string	トリガーに関連付けられた URL。
status	string	トリガーのステータス。 指定可能な値:[1] ENABLED (0, デフォルト), DISABLED (1)。

要素	Type	説明
priority	string	トリガーの深刻度。 指定可能な値:[1] NOT_CLASSIFIED (0, デフォルト), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5)。
description	text	トリガーの説明。
type	string	イベント生成タイプ（単一の障害イベントまたは複数の障害イベント）。 指定可能な値:[1] SINGLE (0, デフォルト), MULTIPLE (1)。
manual_close	string	障害イベントの手動クローズ。 指定可能な値:[1] NO (0, デフォルト), YES (1)。
dependencies		依存関係のルート要素。
name	string	（必須）依存先トリガー名。
expression	string	（必須）依存先トリガー式。
recovery_expression	string	依存先トリガーの復旧式。
tags		トリガータグのルート要素。
tag	string	（必須）タグ名。
value	string	タグ値。

Note:
参照: Trigger object（一致する名前の関連プロパティを参照してください）。

ホストのローレベルディスカバリルール

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (see table below)
          description: 'Scanning databases in DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Size of database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Database size.'
              preprocessing:
                - type: DISCARD_UNCHANGED_HEARTBEAT
                  parameters:
                    - 1h
              interface_ref: if1
              tags:
                - tag: component
                  value: storage
                - tag: database
                  value: '{#DATABASE}'
          lld_macro_paths:
            - lld_macro: '{#DATABASE}'
              path: $.Database
          preprocessing:
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 1d
        (...)
```

Attention:
ホストのローレベルディスカバリルール要素の大部分は、ホストアイテム と同じです。以下の表では、ホストアイテムと異なる要素について説明します。

Element	Type	Description
type	string	アイテムタイプ。 使用可能な値:[1] ZABBIX_PASSIVE (0、デフォルト)、TRAP (2)、SIMPLE (3)、INTERNAL (5)、ZABBIX_ACTIVE (7)、EXTERNAL (10)、ODBC (11)、IPMI (12)、SSH (13)、TELNET (14)、JMX (16)、DEPENDENT (18)、HTTP_AGENT (19)、SNMP_AGENT (20)、ITEM_TYPE_SCRIPT (21)、ITEM_TYPE_BROWSER (22)。
key	string	(必須) ローレベルディスカバリルールのキー。
filter		ホストのローレベルディスカバリルールフィルターのルート要素。
lifetime	string	検出されなくなったリソースが削除されるまでの期間 (秒、time suffix、またはuser macro を使用)。 デフォルト: 7d。
lifetime_type	string	失われた LLD リソースを削除するシナリオ。 使用可能な値: DELETE_NEVER、DELETE_IMMEDIATELY、DELETE_AFTER。
enabled_lifetime	string	検出されなくなったリソースが無効化されるまでの期間 (秒、time suffix、またはuser macro を使用)。
enabled_lifetime_type	string	失われた LLD リソースを無効化するシナリオ。 使用可能な値: DISABLE_NEVER、DISABLE_IMMEDIATELY、DISABLE_AFTER。
item_prototypes		ホストアイテムプロトタイプ要素のルート要素。内容はホストアイテム と同じです。
trigger_prototypes		ホストトリガープロトタイプ要素のルート要素。内容はホストアイテムトリガー と同じです。
graph_prototypes		ホストグラフプロトタイプのルート要素。内容はホストグラフ と同じです。
host_prototypes		ホストプロトタイプのルート要素。内容はホスト と同じです。
parent_discovery_rule		ローレベルディスカバリルールプロトタイプの親ローレベルディスカバリルール (またはルールプロトタイプ) のルート要素。このプロパティは、それが参照されたルール/ルールプロトタイプの直接の子である LLD ルールプロトタイプであることを示します。
key	string	(必須) 親ローレベルディスカバリルール (またはルールプロトタイプ) のキー。
master_item	string	(DEPENDENT ルールで必須) 依存ルールのマスターアイテムのルート要素。
lld_macro_paths		ローレベルディスカバリルールのマクロパスのルート要素。
lld_macro	string	(必須) ローレベルディスカバリルールのマクロ名。
path	string	(必須) 対応するマクロに割り当てられる値のセレクター。
preprocessing		ローレベルディスカバリルールの値の前処理のルート要素。
step		ローレベルディスカバリルールの値の前処理ステップ要素のルート要素。内容はホストアイテムの値の前処理ステップ と同じですが、使用可能な値はより少なくなります。あわせて参照: LLD rule preprocessing object。
overrides		ローレベルディスカバリルールのオーバーライドルールのルート要素。
name	string	(必須) 一意のオーバーライド名。
step	string	(必須) オーバーライドの一意の順序番号。
stop	string	一致した場合、以降のオーバーライドの処理を停止します。
filter		ローレベルディスカバリルールのオーバーライドルールフィルター要素のルート要素。内容はホストのローレベルディスカバリルールフィルター と同じです。
operations		ホストのローレベルディスカバリルールのオーバーライド操作のルート要素。

Note:
あわせて参照: LLD rule object (一致する名前の関連プロパティを参照してください)。

ホストのローレベルディスカバリルールのフィルター

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          (...)
          filter:
```

```
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

要素	Type	説明
evaltype	string	フィルター条件の評価方法を上書きします。 指定可能な値:[1] AND_OR (0、デフォルト)、AND (1)、OR (2)、FORMULA (3)。
formula	string	フィルター条件のカスタム計算式。
conditions		フィルター条件のルート要素。
macro	string	(必須) チェック対象のローレベルディスカバリマクロ名。
value	string	比較する値。
operator	string	条件演算子。 指定可能な値:[1] MATCHES_REGEX (8、デフォルト)、NOT_MATCHES_REGEX (9)。
formulaid	string	(必須) カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件の変更時にはこの ID をユーザーが定義する必要がありますが、その後に取得を要求すると新たに生成されます。

Note:
関連項目: LLD rule filter object (一致する名前の関連プロパティを参照してください)。

ホストのローレベルディスカバリルールの上書き操作

要素	Type	説明
operationobject	string	操作が適用されるオブジェクト。 使用可能な値:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4)。
operator	string	上書き条件の演算子。 使用可能な値:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6)。
value	string	上書き条件の演算子に使用する正規表現または文字列。
status	string	上書き操作時のオブジェクトのステータス。
discover	string	ディスカバリの結果としてオブジェクトが追加されるかどうか。
delay	string	上書き操作時にアイテムのプロトタイプに設定される更新間隔。
history	string	上書き操作時にアイテムのプロトタイプに設定される履歴保存期間。
trends	string	上書き操作時にアイテムのプロトタイプに設定されるトレンド保存期間。
severity	string	上書き操作時に設定されるトリガーのプロトタイプの深刻度。
tags		上書き操作時にオブジェクトに設定されるタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグの値。
templates		上書き操作時にホストのプロトタイプにリンクされるテンプレートのルート要素。
name	string	(必須) テンプレート名。
inventory_mode	string	上書き操作時に設定されるホストのプロトタイプのインベントリモード。

Note:
関連項目: LLD rule override operation object (一致する名前の関連プロパティを参照してください)。

ホストの Web シナリオ

Element	Type	説明
uuid	string	(必須) この Web シナリオの一意の識別子。
name	string	(必須) Web シナリオ名。

Element	Type	説明
delay	string	Web シナリオを実行する頻度（秒、time suffix、またはuser macro を使用）。 デフォルト: 1m。
attempts	integer	Web シナリオのステップを実行する試行回数。 指定可能な値: 1～10（デフォルト: 1）。
agent	string	クライアントエージェント。Zabbix は選択したブラウザーを装います。 これは、Web サイトがブラウザーごとに異なるコンテンツを返す場合に便利です。 デフォルト: Zabbix。
http_proxy	string	Web シナリオで使用されるプロキシ。形式: `http://[username[:password]@]proxy.example.com[:port]`
variables		シナリオのステップで使用できる Web シナリオ変数のルート要素。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。 ヘッダーは、HTTP プロトコルで記述される場合と同じ構文で列挙する必要があります。
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
status	string	Web シナリオのステータス。 指定可能な値:[1] ENABLED (0、デフォルト)、DISABLED (1)。
authentication	string	認証方式。 指定可能な値:[1] NONE (0、デフォルト)、BASIC (1)、NTLM (2)、Kerberos (3)、または Digest (4)。
http_user	string	BASIC (HTTP)、NTLM、Kerberos、または Digest 認証で使用するユーザー名。
http_password	string	BASIC (HTTP)、NTLM、Kerberos、または Digest 認証で使用するパスワード。
verify_peer	string	Web サーバーの SSL 証明書を検証します。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
verify_host	string	Web サーバー証明書の *Common Name* フィールドまたは *Subject Alternate Name* フィールドが一致することを検証します。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
ssl_cert_file	string	クライアント認証に使用する SSL 証明書ファイル名（PEM 形式である必要があります）。
ssl_key_file	string	クライアント認証に使用する SSL 秘密鍵ファイル名（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵ファイルのパスワード。
steps		(必須) ホストの Web シナリオステップ のルート要素。
tags		Web シナリオタグのルート要素。
tag	string	(必須) タグ名。
value	string	タグ値。

Note:
関連項目: Web scenario object（一致する名前の関連プロパティを参照してください）。

ホストの Web シナリオのステップ

要素	型	説明
name	string	(必須) Web シナリオのステップ名。
url	string	(必須) 監視対象の URL。
query_fields		クエリパラメータのルート要素（リクエスト実行時に URL へ追加される HTTP フィールドの配列）。
name	string	(必須) クエリパラメータ名。
value	string	クエリパラメータ値。
posts		HTTP POST 変数のルート要素（文字列（生の POST データ）または HTTP フィールドの配列（フォームフィールドデータ））。
name	string	(必須) POST フィールド名。
value	string	(必須) POST フィールド値。
variables		このステップの後に適用されるステップレベルの変数（マクロ）のルート要素。 変数値に'regex:' プレフィックスがある場合、その値は'regex:' プレフィックスに続く正規表現パターンに従って、このステップで返されたデータから抽出されます。
name	string	(必須) 変数名。
value	text	(必須) 変数値。
headers		リクエスト実行時に送信される HTTP ヘッダーのルート要素。

要素	型	説明
name	string	(必須) ヘッダー名。
value	text	(必須) ヘッダー値。
follow_redirects	string	HTTP リダイレクトに従います。 指定可能な値:[1] NO (0)、YES (1、デフォルト)。
retrieve_mode	string	HTTP レスポンスの取得モード。 指定可能な値:[1] BODY (0、デフォルト)、HEADERS (1)、BOTH (2)。
timeout	string	ステップ実行のタイムアウト（秒、time suffix、またはuser macro を使用)。 デフォルト: 15s。
required	string	レスポンス内に存在する必要があるテキスト（空の場合は無視されます)。
status_codes	string	許可される HTTP ステータスコードのカンマ区切りリスト（例: 200-201,210-299。空の場合は無視されます)。

Note:
参照: Web scenario step object（一致する名前の関連プロパティを参照してください)。

ホストグラフ

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: （以下の表を参照）
    (...)
```

要素	型	説明
uuid	string	グラフの一意な識別子。
name	string	(必須）グラフ名。
width	integer	グラフの幅（ピクセル単位)。 プレビューおよび円/分離円グラフで使用されます。 指定可能な値:[1] 20-65535（デフォルト: 900)。
height	integer	グラフの高さ（ピクセル単位)。 プレビューおよび円/分離円グラフで使用されます。 指定可能な値:[1] 20-65535（デフォルト: 900)。
yaxismin	double	Y 軸の最小値。 Y 軸の最小値が FIXED の場合にサポートされます。 デフォルト: 0。
yaxismax	double	Y 軸の最大値。 X 軸の最小値が FIXED の場合にサポートされます。 デフォルト: 0。
show_work_period	string	非稼働時間を強調表示します。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] NO (0), YES (1、デフォルト)。
show_triggers	string	単純トリガーの値を線として表示します。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] NO (0), YES (1、デフォルト)。
type	string	グラフの種類。 指定可能な値:[1] NORMAL (0、デフォルト), STACKED (1), PIE (2), EXPLODED (3)。
show_legend	string	グラフの凡例を表示します。 指定可能な値:[1] NO (0), YES (1、デフォルト)。
show_3d	string	3D スタイルを有効にします。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] NO (0、デフォルト), YES (1)。
percent_left	double	左軸のパーセンタイル線を表示します。 NORMAL グラフでサポートされます。 デフォルト: 0。
percent_right	double	右軸のパーセンタイル線を表示します。 NORMAL グラフでサポートされます。 デフォルト: 0。

要素	型	説明
ymin_type_1	string	Y 軸の最小値。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] CALCULATED (0、デフォルト), FIXED (1), ITEM (2)。
ymin_item_1		(ymin_type_1 が ITEM に設定されている場合は必須) 個々のアイテム詳細のルート要素。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムキー。
ymax_type_1	string	Y 軸の最大値。 NORMAL および STACKED グラフでサポートされます。 指定可能な値:[1] CALCULATED (0、デフォルト), FIXED (1), ITEM (2)。
ymax_item_1		(ymax_type_1 が ITEM に設定されている場合は必須) 個々のアイテム詳細のルート要素。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムキー。
graph_items		(必須) ホストグラフアイテムのルート要素。

Note:
参照: Graph object (一致する名前を持つ関連プロパティを参照してください)。

ホストのグラフアイテム

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items:
        - drawtype: GRADIENT_LINE
          color: 199C0D
          item:
            host: 'MySQL server'
            key: mysql.bytes_received.rate
        - sortorder: '1'
          drawtype: GRADIENT_LINE
          color: F63100
          item:
            host: 'MySQL server'
            key: mysql.bytes_sent.rate
    (...)
```

要素	Type	説明
sortorder	integer	描画順序。値が小さいものから先に描画されます。ほかの線や領域の背後 (または前面) に描画するために使用できます。
drawtype	string	グラフアイテムの描画スタイル。 NORMAL グラフでサポートされています。 指定可能な値:[1] SINGLE_LINE (0、デフォルト)、FILLED_REGION (1)、BOLD_LINE (2)、DOTTED_LINE (3)、DASHED_LINE (4)、GRADIENT_LINE (5)。
color	string	要素の色 (16 進数 6 文字)。
yaxisside	string	グラフアイテムの Y 軸スケールを描画するグラフの側。 NORMAL および STACKED グラフでサポートされています。
calc_fnc	string	アイテムに複数の値が存在する場合に描画するデータ。 指定可能な値:[1] MIN (1)、AVG (2、デフォルト)、MAX (4)、ALL (7、最小値・平均値・最大値。シンプルグラフでサポート)、LAST (9、円グラフ/分解円グラフでサポート)。
type	string	グラフアイテムのタイプ。 指定可能な値:[1] SIMPLE (0、デフォルト)、GRAPH_SUM (2、アイテムの値が円グラフ全体を表します。円グラフ/分解円グラフでサポート)。
item		(必須) 個別のアイテム。
host	string	(必須) アイテムのホスト。
key	string	(必須) アイテムキー。

Note:
関連情報: Graph item object（一致する名前の関連プロパティを参照してください）。

ホストの値のマッピング

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: 値のマッピングの例
          mappings:
            - value: '1'
              newvalue: 変換後の値の例
  (...)
```

要素	Type	説明
uuid	string	(必須) この値のマッピングの一意な識別子。
name	string	(必須) 値のマッピング名。
mapping		マッピングのルート要素。
type	string	マッピングの一致タイプ。 指定可能な値:[1] EQUAL (0、デフォルト)、GREATER_OR_EQUAL (2)、LESS_OR_EQUAL (3)、IN_RANGE (4)、REGEXP (5)、DEFAULT (6)。
value	string	元の値。
newvalue	string	(必須) 元の値がマッピングされる先の値。

Note:
関連項目: Value map object（一致する名前の関連プロパティを参照してください）。

脚注 [1] かっこ内の API integer 値（例: ENABLED (0)）は、参照用としてのみ記載されています。詳細については、表の該当エントリにあるリンク先の API オブジェクトページ、または各セクションの末尾を参照してください。

5 ネットワークマップ

概要

ネットワークマップのエクスポートには、以下が含まれます。

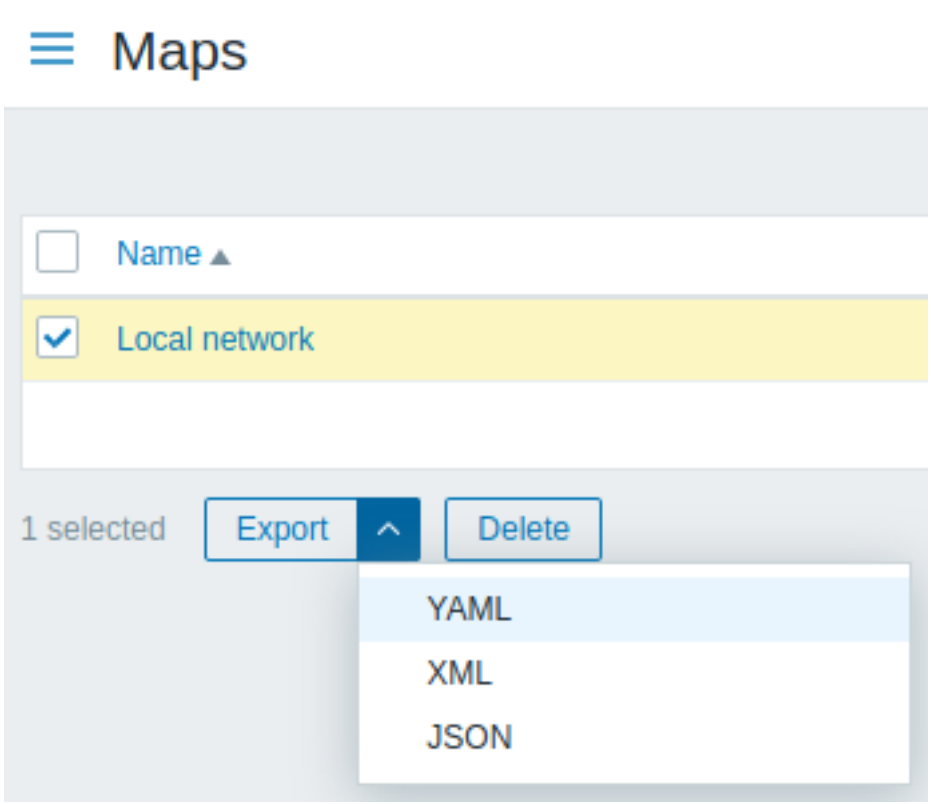
- 関連するすべての画像
- マップ構造（すべてのマップ設定、含まれるすべての要素とその設定、マップリンク、およびマップリンクのステータスインジケーター）

Warning:
エクスポートされたマップに関連している可能性のあるホストグループ、ホスト、トリガー、その他のマップ、またはその他の要素はエクスポートされません。そのため、マップが参照する要素のうち少なくとも 1 つが欠けている場合、インポートは失敗します。

エクスポート

ネットワークマップをエクスポートするには、次の手順を実行します。

1. 監視 → マップに移動します。
2. エクスポートするネットワークマップのチェックボックスをオンにします。
3. 一覧の下にある エクスポートをクリックします。



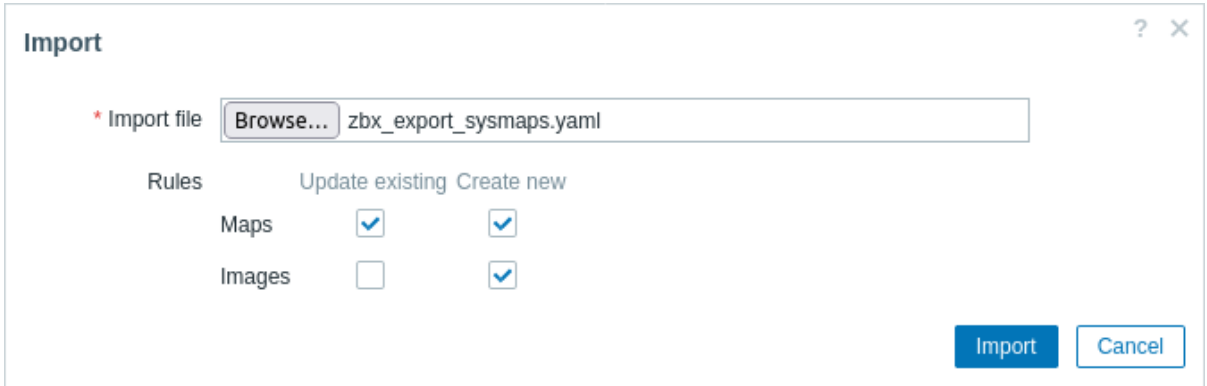
選択した形式に応じて、マップはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_maps.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）
- `zabbix_export_maps.xml` - XML エクスポートの場合
- `zabbix_export_maps.json` - JSON エクスポートの場合

インポート

ネットワークマップをインポートするには、次の手順を実行します。

1. 監視 → マップに移動します。
2. 右上隅の インポートをクリックします。
3. インポートファイルを選択します。
4. インポートルールで必要なオプションをチェックします。
5. 設定フォームの右下隅にある インポートをクリックします。



インポートルール:

ルール	説明
既存のものを更新	既存のマップは、インポートファイルのデータを使用して更新されます。それ以外の場合、更新されません。
新規作成	新しいマップは、インポートファイルのデータを使用して作成されます。それ以外の場合、作成されません。

マップの両方のオプションのチェックを外し、画像に対応するオプションをチェックすると、画像のみがインポートされます。画像のインポートは *Super admin* ユーザーのみが利用できます。

インポートの成功または失敗を示すメッセージが、Web インターフェースに表示されます。

Warning:
既存の画像を置き換えると、この画像を使用しているすべてのマップに影響します。

エクスポート形式

YAML へのエクスポート:

```yaml
zabbix_export:
  version: '7.4'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'ローカルネットワーク'
      width: '680'
      height: '200'
      label_type: '0'
      label_location: '0'
      highlight: '1'
      expandproblem: '1'
      markelements: '1'
      show_unack: '0'
      severity_min: '0'
      show_suppressed: '0'
      grid_size: '50'
      grid_show: '1'
      grid_align: '1'
      label_format: '0'
      label_type_host: '2'
      label_type_hostgroup: '2'
      label_type_trigger: '2'
      label_type_map: '2'
      label_type_image: '2'
      label_string_host: ''
      label_string_hostgroup: ''
      label_string_trigger: ''
      label_string_map: ''
      label_string_image: ''
      expand_macros: '1'
      background: {  }
      iconmap: {  }
      urls: {  }
      selements:
        - elementtype: '0'
          elements:
            - host: 'Zabbix server'
          label: |
            {HOST.NAME}
            {HOST.CONN}
          label_location: '0'
          x: '111'
          'y': '61'
          elementsubtype: '0'
          areatype: '0'
          width: '200'
          height: '200'
          viewtype: '0'
          use_iconmap: '0'
          selementid: '1'
          icon_off:
            name: Zabbix_server_3D_(128)
          icon_on: {  }
```

```
        icon_disabled: {  }
        icon_maintenance: {  }
        urls: {  }
        evaltype: '0'
    shapes:
      - type: '0'
        x: '0'
        'y': '0'
        width: '680'
        height: '15'
        text: '{MAP.NAME}'
        font: '9'
        font_size: '11'
        font_color: '000000'
        text_halign: '0'
        text_valign: '0'
        border_type: '0'
        border_width: '0'
        border_color: '000000'
        background_color: ''
        zindex: '0'
    lines: {  }
    links: {  }
```

エクスポートされる要素　エクスポートされる要素については、以下の表で説明します。

Element		Type	Description
images			画像のルート要素。
	name	string	一意の画像名。
	imagetype	integer	画像タイプ。 指定可能な値: 1 - 画像; 2 - 背景。
	encodedImage	string	Base64 でエンコードされた画像。
maps			マップ のルート要素。

マップ

Element	Type	説明
name	string	一意のマップ名。
width	integer	マップの幅（ピクセル単位）。
height	integer	マップの高さ（ピクセル単位）。
label_type	integer	マップ要素ラベルの種類。 指定可能な値: 0 - ラベル; 1 - ホスト IP アドレス; 2 - 要素名; 3 - ステータスのみ; 4 - なし。
label_location	integer	デフォルトのマップ要素ラベルの位置。 指定可能な値: 0 - 下; 1 - 左; 2 - 右; 3 - 上。
highlight	integer	アクティブなトリガーおよびホストステータスのアイコン強調表示を有効にします。 指定可能な値: 0 - いいえ; 1 - はい。

Element	Type	説明
expandproblem	integer	問題が 1 つだけある要素について、障害トリガーを表示します。 指定可能な値: 0 - いいえ; 1 - はい。
markelements	integer	最近ステータスが変更されたマップ要素を強調表示します。 指定可能な値: 0 - いいえ; 1 - はい。
show_unack	integer	障害の表示方法。 指定可能な値: 0 - すべての障害数; 1 - 未確認の障害数; 2 - 確認済みと未確認の障害数を別々に表示。
severity_min	integer	デフォルトでマップに表示する最小トリガー深刻度。 指定可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害。
show_suppressed	integer	ホストのメンテナンスにより通常は抑制される（表示されない）障害を表示します。 指定可能な値: 0 - いいえ; 1 - はい。
grid_size	integer	マップグリッドのセルサイズ（ピクセル単位）。 `grid_show` が 0 に設定されている場合にサポートされます。 指定可能な値: 20、40、50、75、100。
grid_show	integer	マップ設定でグリッドを表示します。 指定可能な値: 0 - はい; 1 - いいえ。
grid_align	integer	マップ設定でアイコンを自動的に整列します。 指定可能な値: 0 - はい; 1 - いいえ。
label_format	integer	高度なラベル設定を使用します。 指定可能な値: 0 - いいえ; 1 - はい。
label_type_host	integer	ラベルをホストラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - ラベル; 1 - ホスト IP アドレス; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_hostgroup	integer	ラベルをホストグループラベルとして表示します。 `label_format` が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。

Element	Type	説明
label_type_trigger	integer	ラベルをトリガーラベルとして表示します。 label_format が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_map	integer	ラベルをマップラベルとして表示します。 label_format が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - ラベル; 2 - 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタムラベル。
label_type_image	integer	ラベルをイメージラベルとして表示します。 label_format が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - ラベル; 2 - 要素名; 4 - なし; 5 - カスタムラベル。
label_string_host	string	ホスト要素用のカスタムラベル。 label_type_host が 5 に設定されている場合にサポートされます。
label_string_hostgroup	string	ホストグループ要素用のカスタムラベル。 label_type_hostgroup が 5 に設定されている場合にサポートされます。
label_string_trigger	string	トリガー要素用のカスタムラベル。 label_type_trigger が 5 に設定されている場合にサポートされます。
label_string_map	string	マップ要素用のカスタムラベル。 label_type_map が 5 に設定されている場合にサポートされます。
label_string_image	string	イメージ要素用のカスタムラベル。 label_type_image が 5 に設定されている場合にサポートされます。
expand_macros	integer	マップ設定でラベル内のマクロを展開します。 指定可能な値: 0 - いいえ; 1 - はい。
background		背景画像のルート要素（存在する場合）。 imagetype が 2 に設定されている場合にサポートされます。
name	string	背景画像名。
iconmap		アイコンマッピングのルート要素（存在する場合）。
name	string	アイコンマッピング名。
urls		マップまたは各マップ要素で使用される URL のルート要素。
name	string	リンク名。
url	string	リンク URL。
elementtype	integer	リンクが属するマップアイテムの種類。 指定可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - イメージ。
selements		マップ selements のルート要素。
shapes		マップ図形のルート要素。
type	integer	図形の種類。 指定可能な値: 0 - 長方形; 1 - 楕円。
x	integer	図形の X 座標（ピクセル単位）。
y	integer	図形の Y 座標（ピクセル単位）。
width	integer	図形の幅。

Element		Type	説明
	height	integer	図形の高さ。
	text	string	図形内のテキスト。
	font	integer	テキストのフォントスタイル。 指定可能な値: 0 - Georgia, serif; 1 - ”Palatino Linotype”, ”Book Antiqua”, Palatino, serif; 2 - ”Times New Roman”, Times, serif; 3 - Arial, Helvetica, sans-serif; 4 - ”Arial Black”, Gadget, sans-serif; 5 - ”Comic Sans MS”, cursive, sans-serif; 6 - Impact, Charcoal, sans-serif; 7 - ”Lucida Sans Unicode”, ”Lucida Grande”, sans-serif; 8 - Tahoma, Geneva, sans-serif; 9 - ”Trebuchet MS”, Helvetica, sans-serif; 10 - Verdana, Geneva, sans-serif; 11 - ”Courier New”, Courier, monospace; 12 - ”Lucida Console”, Monaco, monospace。
	font_size	integer	フォントサイズ（ピクセル単位）。
	font_color	string	16 進コードで表されるフォント色。
	text_halign	integer	テキストの水平方向の配置。 指定可能な値: 0 - 中央; 1 - 左; 2 - 右。
	text_valign	integer	テキストの垂直方向の配置。 指定可能な値: 0 - 中央; 1 - 上; 2 - 下。
	border_type	integer	図形の枠線の種類。 指定可能な値: 0 - なし; 1 - 太線; 2 - 点線; 3 - 破線。
	border_width	integer	枠線の幅（ピクセル単位）。
	border_color	string	16 進コードで表される枠線色。
	background_color	string	16 進コードで表される背景（塗りつぶし）色。
	zindex	integer	すべての図形と線の順序付けに使用する値（z-index）。
lines			マップ線のルート要素。
	x1	integer	線の点 1 の X 座標（ピクセル単位）。
	y1	integer	線の点 1 の Y 座標（ピクセル単位）。
	x2	integer	線の点 2 の X 座標（ピクセル単位）。
	y2	integer	線の点 2 の Y 座標（ピクセル単位）。
	line_type	integer	線の種類。 指定可能な値: 0 - なし; 1 - 太線; 2 - 点線; 3 - 破線。
	line_width	integer	線の幅（ピクセル単位）。
	line_color	string	16 進コードで表される線の色。
	zindex	integer	すべての図形と線の順序付けに使用する値（z-index）。
links			マップ要素間のリンクのルート要素。
	drawtype	integer	リンクスタイル。 指定可能な値: 0 - 線; 2 - 太線; 3 - 点線; 4 - 破線。
	color	string	リンク色（16 進数 6 文字）。

Element	Type	説明
label	string	リンクラベル。
selementid1	id	接続する一方の要素の ID。
selementid2	id	接続するもう一方の要素の ID。
linktriggers		リンクステータスインジケーター のルート要素。

Note:
関連項目: Map object（一致する名前の関連プロパティを参照してください）。

マップの selement

要素	型	説明
elementtype	integer	マップ要素のタイプ。 指定可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - 画像。
elements		マップ上に表示される Zabbix エンティティ（ホスト、ホストグループ、マップなど）のルート要素。 すべてのエンティティは、それぞれ対応する要素（host など。例としてExport format を参照）を持ちます。
label	string	アイコンラベル。
label_location	integer	ラベルの位置。 指定可能な値: -1 - マップのデフォルトを使用; 0 - 下; 1 - 左; 2 - 右; 3 - 上。
x	integer	X 軸上の位置。
y	integer	Y 軸上の位置。
elementsubtype	integer	要素のサブタイプ。 elementtype が 3 に設定されている場合にサポートされます。 指定可能な値: 0 - 単一のホストグループ; 1 - すべてのホストグループ。
areatype	integer	エリアサイズ。 elementtype が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - マップ全体と同じ; 1 - カスタムサイズ。
width	integer	エリアの幅。 areatype が 1 に設定されている場合にサポートされます。
height	integer	エリアの高さ。 areatype が 1 に設定されている場合にサポートされます。
viewtype	integer	エリア配置アルゴリズム。 elementsubtype が 1 に設定されている場合にサポートされます。 指定可能な値: 0 - エリア内に均等に配置。
use_iconmap	integer	この要素にアイコンマッピングを使用します。マップレベルでアイコンマッピングが有効化されている場合にのみ関係します。 指定可能な値: 0 - いいえ; 1 - はい。
selementid	id	一意の要素レコード ID。
icon_off		要素が「OK」ステータスのときに使用する画像のルート要素。
name	string	一意の画像名。

要素	型	説明
icon_on		要素が「障害」ステータスのときに使用する画像のルート要素。
name	string	一意の画像名。
icon_disabled		要素が無効化されているときに使用する画像のルート要素。
name	string	一意の画像名。
icon_maintenance		要素がメンテナンス中のときに使用する画像のルート要素。
name	string	一意の画像名。
urls		マップまたは各マップ要素で使用される URL のルート要素。
name	string	リンク名。
url	string	リンク URL。
evaltype	integer	タグの評価タイプ。
tags		障害タグ（ホスト要素およびホストグループ要素用）のルート要素。 タグが指定されている場合、これらのタグを持つ障害のみがマップに表示されます。
tag	string	タグ名。
value	string	タグ値。
operator	integer	演算子。

Note:
関連項目: Map element object（一致する名前を持つ該当プロパティを参照してください）。

マップリンクのステータスインジケーター

要素	型	説明
drawtype	integer	トリガーが「障害」状態のときのリンクスタイル。 使用可能な値: 0 - 線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	トリガーが「障害」状態のときのリンクの色（16 進数 6 文字）。
trigger		リンクのステータス表示に使用されるトリガーのルート要素。
description	string	トリガー名。
expression	string	トリガー条件式。
recovery_expression	string	トリガーの復旧条件式。

Note:
関連項目: Map link trigger object（一致する名前の関連プロパティを参照してください）。

6 メディアタイプ

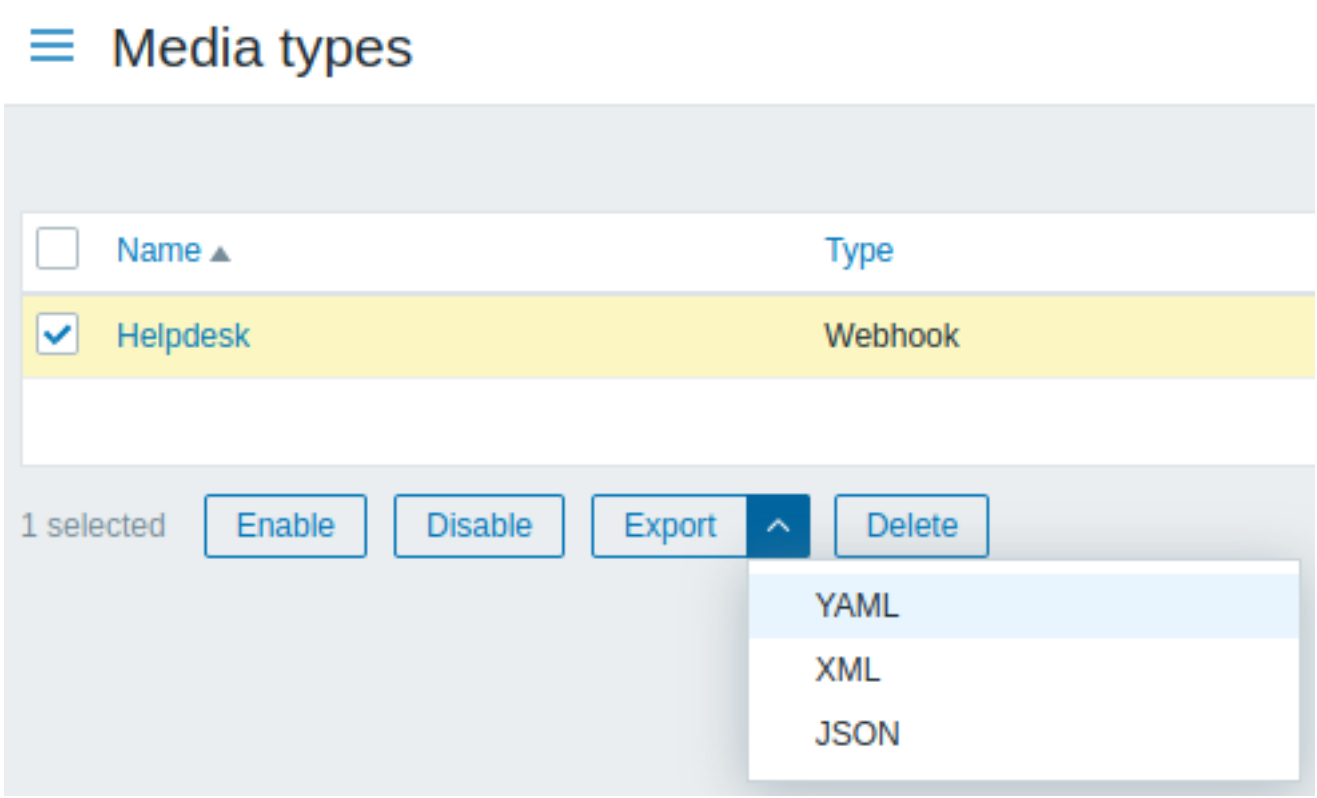
概要

メディアタイプは、関連するすべてのオブジェクトおよびオブジェクト間の関係とともにエクスポートされます。

エクスポート

メディアタイプをエクスポートするには、次の手順を実行します。

1. *Alerts* → *Media types* に移動します。
2. エクスポートするメディアタイプのチェックボックスをオンにします。
3. 一覧の下にある *Export* をクリックします。



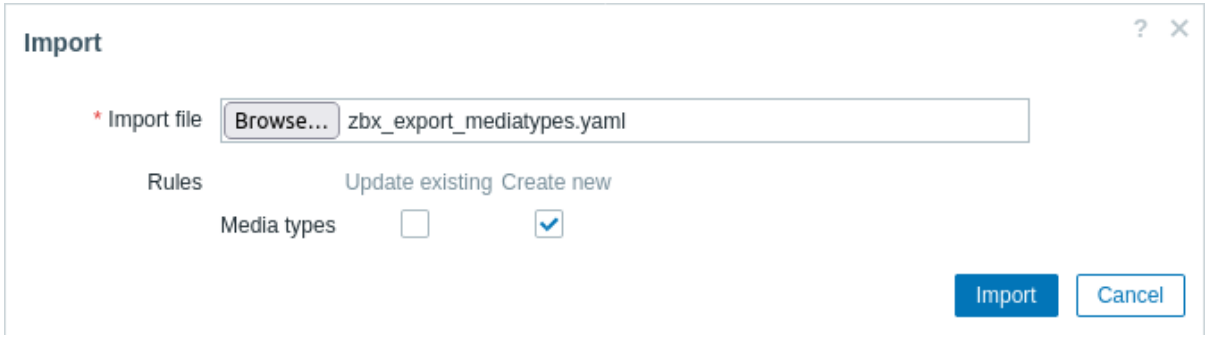
選択した形式に応じて、メディアタイプはデフォルト名のローカルファイルにエクスポートされます。

- `zabbix_export_mediatypes.yaml` - YAML エクスポートの場合（エクスポートのデフォルトオプション）
- `zabbix_export_mediatypes.xml` - XML エクスポートの場合
- `zabbix_export_mediatypes.json` - JSON エクスポートの場合。

インポート

メディアタイプをインポートするには、次の手順を実行します。

1. *Alerts* → *Media types* に移動します。
2. 右上隅の *Import* をクリックします。
3. インポートファイルを選択します。
4. インポートルールで必要なオプションをチェックします。
5. 設定フォームの右下隅にある *Import* をクリックします。



インポートルール:

ルール	説明
Update existing	既存の要素は、インポートファイルのデータを使用して更新されます。それ以外の場合は、更新されません。
Create new	新しい要素は、インポートファイルのデータを使用して作成されます。それ以外の場合は、作成されません。

インポートの成功または失敗のメッセージが、Web インターフェースに表示されます。

エクスポート形式

YAML にエクスポート:

```yaml
zabbix_export:
  version: '7.4'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
                    {name: 'average', color: '#FFA059'},
                    {name: 'high', color: '#E97659'},
```

```
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 1
    }

    if ([0, 1, 2, 3, 4, 5].indexOf(parseInt(params.event_nseverity)) === -1) {
        params.event_nseverity = '7';
    }

    if (params.event_value === '0') {
        params.event_nseverity = '6';
    }

    priority = params['priority_' + severities[params.event_nseverity].name] || params.priority_de

    if (isNaN(priority) || priority < -2 || priority > 2) {
        throw '"priority" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    data = {
        token: params.token,
        user: params.user,
        title: params.title,
        message: params.message,
        url: (params.event_source === '0')
            ? params.url + '/tr_events.php?triggerid=' + params.triggerid + '&eventid=' + params.e
            : params.url,
        url_title: params.url_title,
        priority: priority
    };

    if (priority == 2) {
        if (isNaN(params.retry) || params.retry < 30) {
            throw 'field "retry" should be a number with value of at least 30 if "priority" is set
        }
```

```
            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    セットアップガイドはこちらを参照してください: https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/temp

    token パラメータには、Pushover アプリケーションキーを設定してください。
    Zabbix ユーザーに Pushover メディアを割り当てる際は、送信先フィールドにユーザーキーを追加してください。
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: '障害: {EVENT.NAME}'
      message: |
        障害発生日時 {EVENT.DATE} {EVENT.TIME}
        障害名: {EVENT.NAME}
        ホスト: {HOST.NAME}
        深刻度: {EVENT.SEVERITY}
        運用データ: {EVENT.OPDATA}
        元の障害 ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: RECOVERY
      subject: '{EVENT.DURATION} で復旧: {EVENT.NAME}'
```

```
      message: |
        障害は {EVENT.RECOVERY.DATE} {EVENT.RECOVERY.TIME} に復旧しました
        障害名: {EVENT.NAME}
        障害継続時間: {EVENT.DURATION}
        ホスト: {HOST.NAME}
        深刻度: {EVENT.SEVERITY}
        元の障害 ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: '{EVENT.AGE} 経過した障害を更新: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} が {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME} に障害を {EVENT.UPDATE.ACTION} しまし
        {EVENT.UPDATE.MESSAGE}

        現在の障害ステータスは {EVENT.STATUS}、経過時間は {EVENT.AGE}、確認済み: {EVENT.ACK.STATUS} です。
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'ディスカバリ: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        ディスカバリルール: {DISCOVERY.RULE.NAME}

        デバイス IP: {DISCOVERY.DEVICE.IPADDRESS}
        デバイス DNS: {DISCOVERY.DEVICE.DNS}
        デバイスステータス: {DISCOVERY.DEVICE.STATUS}
        デバイス稼働時間: {DISCOVERY.DEVICE.UPTIME}

        デバイスサービス名: {DISCOVERY.SERVICE.NAME}
        デバイスサービスポート: {DISCOVERY.SERVICE.PORT}
        デバイスサービスステータス: {DISCOVERY.SERVICE.STATUS}
        デバイスサービス稼働時間: {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: '自動登録: {HOST.HOST}'
      message: |
        ホスト名: {HOST.HOST}
        ホスト IP: {HOST.IP}
        エージェントポート: {HOST.PORT}
```

エクスポートされる要素

エクスポートされる要素については、以下の表で説明します。

要素	型	説明
name	string	(必須) メディアタイプ名。
type	string	(必須) メディアタイプで使用される転送方式。 指定可能な値:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4)。
status	string	メディアタイプが有効かどうか。 指定可能な値:[1] ENABLED (0、デフォルト), DISABLED (1)
max_sessions	integer	並列処理できるアラートの最大数。 SMS の指定可能な値:[1] 1 (デフォルト)。 その他のメディアタイプの指定可能な値:[1] 0-100 (0 は無制限)。
attempts	integer	アラート送信の最大試行回数。 指定可能な値:[1] 1-10 (デフォルト: 3)。
attempt_interval	string	再試行の間隔 (秒またはtime suffixを使用)。 指定可能な値:[1] 0-60s (デフォルト: 10s)。
description	string	メディアタイプの説明。
message_templates		メディアタイプのメッセージテンプレートのルート要素。
event_source	string	(必須) イベントソース。 指定可能な値:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4)。

要素	型	説明
operation_mode	string	動作モード。 指定可能な値:[1] PROBLEM (0), RECOVERY (1), UPDATE (2)。
subject	string	メッセージの件名。
message	string	メッセージ本文。

Note:
関連項目: Media type object（一致する名前の関連プロパティを参照してください）。

メール

以下の追加要素は、メールメディアタイプに対してのみエクスポートされます。

Element	Type	Description
provider	string	メールプロバイダー。
smtp_server	string	SMTP サーバー。
smtp_port	integer	接続先の SMTP サーバーポート。 デフォルト: 25。
smtp_helo	string	SMTP helo。
smtp_email	string	通知の送信元となるメールアドレス。
smtp_security	string	使用する SMTP 接続のセキュリティレベル。 指定可能な値:[1] NONE (0、デフォルト)、STARTTLS (1)、SSL_OR_TLS (2)。
smtp_verify_host	string	SMTP の SSL ホスト検証。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
smtp_verify_peer	string	SMTP の SSL ピア検証。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
smtp_authentication	string	使用する SMTP 認証方式。 指定可能な値:[1] NONE (0、デフォルト)、PASSWORD (1)、OAUTH (2)。
username	string	ユーザー名。
password	string	認証パスワード。
redirection_url	string	OAuth 認可後にリダイレクトして戻る先の Zabbix Web インターフェース URL。
client_id	string	OAuth 認可サーバー内で登録されたクライアント識別子。
authorization_url	string	アクセスおよびリフレッシュトークンを取得するための、パラメータ付き OAuth URL。
token_url	string	認可トークンをアクセスおよびリフレッシュトークンに交換するための OAuth URL。
message_format	string	メッセージ形式。 指定可能な値:[1] TEXT (0)、HTML (1、デフォルト)。

Note:
関連項目: Media type object（一致する名前の該当プロパティを参照してください）。

SMS

以下の追加要素は、*SMS* メディアタイプに対してのみエクスポートされます。

要素	Type	説明
gsm_modem	string	（必須）GSM モデムのシリアルデバイス名。

Note:
あわせて参照: メディアタイプオブジェクト（一致する名前を持つ関連プロパティを参照してください）。

スクリプト

以下の追加要素は、スクリプトメディアタイプに対してのみエクスポートされます。

要素	型	説明
スクリプト名	string	（必須）スクリプト名。

要素		型	説明
parameters			スクリプトパラメータのルート要素。
	sortorder	string	(必須) コマンドライン引数としてスクリプトに渡されるスクリプトパラメータの順序。
	value	string	スクリプトパラメータの値。

Note:
関連項目: メディアタイプオブジェクト (一致する名前を持つ関連プロパティを参照してください)。

webhook

以下の追加要素は、*Webhook* メディアタイプに対してのみエクスポートされます。

要素	Type	説明
script	string	スクリプト。
timeout	string	Javascript スクリプトの HTTP リクエストタイムアウト間隔。 指定可能な値:[1] 1-60s (デフォルト: 30s)。
process_tags	string	返されたタグを処理するかどうか。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
show_event_menu	string	`event_menu_url` および `event_menu_name` フィールドで {EVENT.TAGS.*} マクロが正常に解決された場合に、イベントメニュー内にエントリが存在することを示します。 指定可能な値:[1] NO (0、デフォルト)、YES (1)。
event_menu_url	string	イベントメニューエントリの URL。{EVENT.TAGS.*} マクロをサポートします。
event_menu_name	string	イベントメニューエントリの名前。{EVENT.TAGS.*} マクロをサポートします。
parameters		webhook メディアタイプパラメータのルート要素。
name	string	(必須) webhook パラメータ名。
value	string	webhook パラメータ値。

Note:
関連項目: Media type object (一致する名前の関連プロパティを参照してください)。

脚注

[1] かっこ内の API 整数値 (例: ENABLED (0)) は、参照用としてのみ記載されています。
詳細については、表の該当エントリにあるリンク先の API オブジェクトページ、または各セクションの末尾を参照してください。

15. ディスカバリ

サイドバーから、参照したい項目を選択してください。

1 ネットワークディスカバリ

概要

Zabbix は、自動ネットワークディスカバリ機能を提供しており、これは効果的で非常に柔軟です。

ネットワークディスカバリを適切に設定すると、次のことが可能になります。

- Zabbix の導入を迅速化
- 管理の簡素化
- 過度な管理作業を行うことなく、変化の激しい環境で Zabbix を利用

Zabbix のネットワークディスカバリは、次の情報に基づいています。

- IP 範囲
- 外部サービスの可用性 (FTP、SSH、WEB、POP3、IMAP、TCP など)
- Zabbix エージェントから受信した情報 (サポートされるのは暗号化されていないモードのみ)
- SNMP エージェントから受信した情報

次の機能は提供されません。

- ネットワークトポロジーのディスカバリ

ネットワークディスカバリは、基本的にディスカバリとアクションの 2 つのフェーズで構成されます。

ディスカバリ

Zabbix は、ネットワークディスカバリルールで定義された IP 範囲を定期的にスキャンします。チェックの頻度は、各ルールごとに個別に設定できます。

各ルールには、その IP 範囲に対して実行するサービスチェックのセットが定義されています。

ディスカバリルールはディスカバリマネージャーによって処理されます。ディスカバリマネージャーは、各ルールごとにタスク（ネットワークチェック）のリストを持つジョブを作成します。ネットワークチェックは、利用可能なディスカバリワーカーによって並列に実行されます（数は Web インターフェースで各ルールごとに設定可能です）。同じ IP とポートに対するチェックのみが順次スケジュールされます。これは、一部のデバイスが同じポートへの並列接続を受け付けないためです。

ネットワークチェックのキューサイズは、約 2000000 件または約 4 GB のメモリに制限されています。キューがいっぱいになると、ディスカバリルールはスキップされ、警告メッセージがログに出力されます。キュー内のディスカバリチェック数を監視するには、内部アイテム `zabbix[discovery_queue]` を使用できます。

ディスカバリチェックは、他のチェックとは独立して処理されます。いずれかのチェックでサービスが見つからない場合（または失敗した場合）でも、他のチェックは引き続き処理されます。

Note:
ディスカバリルールが実行中に変更された場合、現在のディスカバリ実行は中止されます。

ネットワークディスカバリモジュールによって実行される、サービスおよびホスト（IP）の各チェックは、ディスカバリイベントを生成します。

イベント	サービスチェック結果
サービス検出	サービスが初回検出時、または'down' だった後に'up' になった場合。
サービス稼働	サービスがすでに'up' だった後も'up' である場合。
サービス消失	サービスが'up' だった後に'down' になった場合。
サービス停止	サービスがすでに'down' だった後も'down' である場合。
ホスト検出	ホストのすべてのサービスが'down' だった後に、そのホストの少なくとも 1 つのサービスが'up' になった場合、または未登録のホストに属するサービスが検出された場合。
ホスト稼働	ホストの少なくとも 1 つのサービスが、すでに少なくとも 1 つのサービスが'up' だった後も'up' である場合。
ホスト消失	ホストの少なくとも 1 つのサービスが'up' だった後に、そのホストのすべてのサービスが'down' になった場合。
ホスト停止	ホストのすべてのサービスが、すでに'down' だった後も'down' である場合。

アクション

ディスカバリイベントは、次のような関連するactionの基礎となります。

- 通知の送信
- ホストの追加/削除
- ホストの有効化/無効化
- ホストをグループに追加
- ホストをグループから削除
- ホストにタグを追加
- ホストからタグを削除
- テンプレートをホストにリンク/ホストからテンプレートのリンクを解除
- リモートスクリプトの実行

これらのアクションは、デバイスタイプ、IP、ステータス、稼働時間/停止時間などに応じて設定できます。
ネットワークディスカバリベースのイベントに対するアクション設定の詳細については、action のoperationおよびconditionsのページを参照してください。

ネットワークディスカバリアクションはイベントベースであるため、検出されたホストがオンラインの場合とオフラインの場合の両方でトリガーされます。
Service Lost/Service Down イベントで *Add host* のようなアクションがトリガーされるのを避けるため、アクションcondition *Discovery status: up* を追加することを強く推奨します。
そうしないと、検出されたホストを手動で削除した場合でも、引き続き *Service Lost/Service Down* イベントが生成され、次回のディスカバリサイクル中に再作成されます。

Note:
検出されたホストへのテンプレートのリンクは、リンク可能なテンプレートのいずれかにある一意のエンティティ（例: アイテムキー）が、そのホスト上または他のリンク可能なテンプレート上にすでに存在する一意のエンティティ（例: アイテムキー）と同一である場合、まとめて失敗します。

ホストの作成

ホストを追加操作が選択されている場合、ホストが追加されます。また、ホストを追加操作がなくても、ホストに対するアクションとなる操作を選択すると、ホストが追加されます。このような操作は次のとおりです。

- ホストを有効化
- ホストを無効化
- ホストをホストグループに追加
- テンプレートをホストにリンク

作成されたホストは *Discovered hosts* グループに追加されます（デフォルト。*Administration* > *General* > *Other* で設定可能）。ホストを別のグループに追加したい場合は、ホストグループから削除操作（”Discovered hosts” を指定）を追加し、さらに ホストグループに追加操作（別のホストグループを指定）も追加してください。ホストは必ずいずれかのホストグループに属している必要があるためです。

検出されたデバイスの IP アドレス、検出元（Zabbix サーバー、Zabbix プロキシ、またはプロキシグループ）、およびインターフェースタイプの組み合わせが、システム内でホストを特定する基準として使用されます。同じ IP アドレス、インターフェースタイプ、および検出元を持つホストがすでに存在する場合、そのホストが操作実行の対象になります。検出元が異なる場合、検出されたエンティティは別のホストとして扱われ、新しいホストが作成されることがあります。

検出されたホストの IP アドレスが変更された場合、またはインターフェースが削除された場合、次回の検出時に新しいホストが作成されます。

ホスト名の命名

ホストを追加する際、ホスト名は逆引き DNS ルックアップの結果となり、逆引きに失敗した場合は IP アドレスになります。ルックアップは、どちらがディスカバリを実行しているかに応じて、Zabbix サーバーまたは Zabbix プロキシから実行されます。プロキシでルックアップに失敗した場合、サーバーで再試行されることはありません。その名前のホストがすでに存在する場合、次のホスト名には **_2** が追加され、その次は **_3** というように続きます。

また、DNS/IP ルックアップを上書きし、代わりにホスト名としてアイテムの値を使用することも可能です。例えば、次のようなケースです。

- Zabbix エージェントが稼働している複数のサーバーを、ディスカバリ用の Zabbix エージェントアイテムを使用して検出し、このアイテムが返す文字列値に基づいて適切な名前を自動的に割り当てることができます
- 複数の SNMP ネットワークデバイスを、ディスカバリ用の SNMP エージェントアイテムを使用して検出し、このアイテムが返す文字列値に基づいて適切な名前を自動的に割り当てることができます

ホスト名がアイテムの値を使用して設定されている場合、その後のディスカバリチェックでは更新されません。アイテムの値を使用してホスト名を設定できない場合は、デフォルト値（DNS 名）が使用されます。

検出された IP アドレスを持つホストがすでに存在し、かつディスカバリ元（Zabbix サーバー、プロキシ、またはプロキシグループ）が変更されていない場合、新しいホストは作成されません。ディスカバリ元が異なる場合、検出されたエンティティは別個のものとして扱われ、新しいホストが作成されることがあります。ただし、ディスカバリアクションに操作（テンプレートのリンク、ホストグループへの追加など）が含まれている場合、それらは IP アドレス、インターフェースタイプ、およびディスカバリ元が一致する既存のホストに対して実行されます。

ホストの削除

ネットワークディスカバリルールによって検出されたホストは、検出されたエンティティがそのルールの IP 範囲内になくなった場合、*Monitoring* > *Discovery* から自動的に削除されます。
ホストは直ちに削除されます。

ホスト追加時のインターフェース作成

ネットワークディスカバリの結果としてホストが追加される場合、インターフェースは次のルールに従って作成されます。

- 検出されたサービス。例えば、SNMP チェックが成功した場合は、SNMP インターフェースが作成されます。
- ホストが Zabbix エージェント要求と SNMP 要求の両方に応答した場合は、両方の種類のインターフェースが作成されます。
- 一意性の条件が Zabbix エージェントまたは SNMP から返されたデータである場合、ホストに対して最初に見つかったインターフェースがデフォルトのインターフェースとして作成されます。
  その他の IP アドレスは追加インターフェースとして追加されます。アクションの条件（ホスト IP など）は、インターフェースの追加には影響しません。これが機能するのは、すべてのインターフェースが同じディスカバリルールによって検出される場合であることに注意してください。同じホストの別のインターフェースが別のディスカバリルールによって検出された場合は、追加のホストが追加されます。

- ホストがエージェントチェックのみに応答した場合は、エージェントインターフェースのみを持つホストとして作成されます。その後 SNMP に応答し始めた場合は、追加の SNMP インターフェースが追加されます。
- 最初に 3 つの別個のホストが”IP” 一意性条件によって検出されて作成され、その後ディスカバリルールが変更されてホスト A、B、C が同一の一意性条件の結果を持つようになった場合、B と C は最初のホストである A の追加インターフェースとして作成されます。
  個別のホスト B と C はそのまま残ります。
  Monitoring > *Discovery* では、追加されたインターフェースは”Discovered device” 列に黒いフォントでインデントされて表示されますが、”Monitored host” 列には最初に作成されたホストである A のみが表示されます。
  追加インターフェースと見なされる IP については、”Uptime/Downtime” は測定されません。

プロキシ設定の変更

異なるプロキシによってディスカバリされたホストは、常に別々のホストとして扱われるわけではありません。
ディスカバリと一意性チェックはプロキシグループの構造に依存します。プロキシがディスカバリルールを実行してホストを作成すると、そのホストはプロキシ自体に割り当てられるのではなく、そのプロキシの親プロキシグループに追加されます。
Zabbix がディスカバリ中に IP の一意性を評価する際は、親プロキシグループによって監視されているホストを確認します。
そのグループ内の個々のプロキシによって監視されているホスト（ディスカバリを実行したプロキシを含む）は、一意性チェックでは無視されます。そのため、複数のプロキシが重複するサブネットを監視している場合、重複したホストが作成される可能性があります。

この動作により、異なるサブネットで使用される重複した IP 範囲にまたがってディスカバリを実行できますが、すでに監視されているサブネットに割り当てられたプロキシを変更する場合は、重複を避けるために、ディスカバリされたホストと親プロキシグループのメンバーシップの両方に対して、プロキシの変更を一貫して適用する必要があるため、より複雑になります。

たとえば、ディスカバリルール内のプロキシを置き換える手順は次のとおりです。

1. ディスカバリルールを無効にする
2. プロキシ設定を同期する
3. ディスカバリルール内のプロキシを置き換える
4. このルールによってディスカバリされたすべてのホストのプロキシを置き換える（重複を避けるため、親プロキシグループ内のホストと、そのグループ内の個々のプロキシによって監視されているすべてのホストが更新されることを確認してください）
5. ディスカバリルールを有効にする

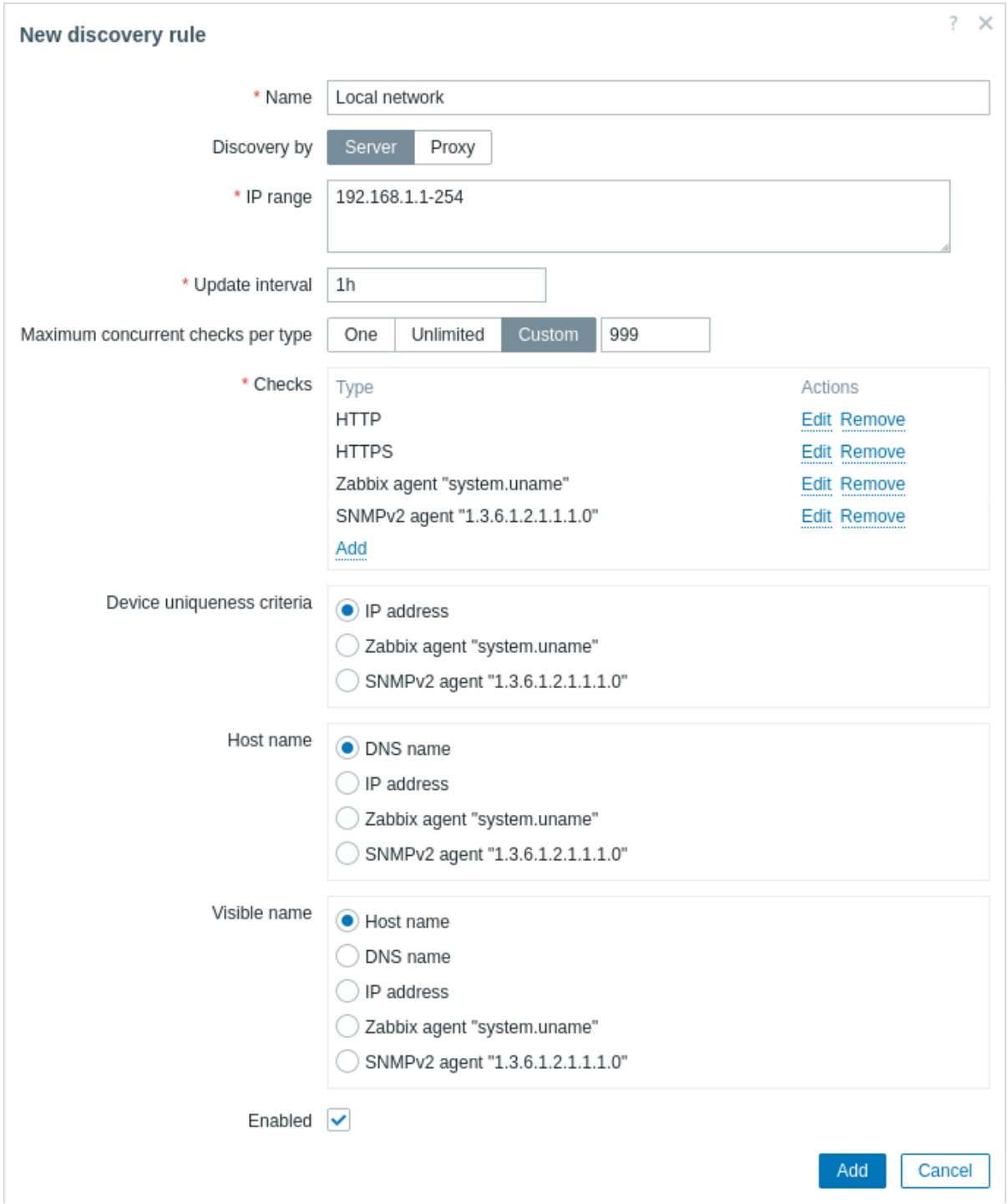
1 ネットワークディスカバリルールの設定

概要

Zabbix がホストおよびサービスを検出するために使用するネットワークディスカバリルールを設定するには、次の手順を実行します。

- データ収集 → ディスカバリに移動します
- ディスカバリルールの作成をクリックします（既存のルールを編集する場合はルール名をクリックします）
- ディスカバリルールの属性を編集します

ルール属性



必須入力フィールドには、赤いアスタリスクが付いています。

パラメータ	説明
Name	ルールの一意な名前。例えば、”Local network”。
Discovery by	ディスカバリの実行元を指定します: **Server** - Zabbix サーバーによる実行 **Proxy** - Zabbix プロキシによる実行（プロキシ名フィールドで選択）
IP range	ディスカバリ対象の IP アドレス範囲。以下の形式を使用できます: 単一 IP: 192.168.1.33 IP アドレス範囲: 192.168.1-10.1-255。範囲は、対象となるアドレス総数によって制限されます (64K 未満)。 IP マスク: 192.168.4.0/24 サポートされる IP マスク: IPv4 アドレスの場合 /16 - /30 IPv6 アドレスの場合 /112 - /128 リスト: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 このフィールドでは、スペース、タブ、および複数行を使用できます。

パラメータ	説明
Update interval	このパラメータは、Zabbix がこのルールを実行する頻度を定義します。 前回のディスカバリ実行が終了してから間隔が計測されるため、実行が重複することはありません。 時間サフィックス をサポートしています。例: 30s、1m、2h、1d。 ユーザーマクロ をサポートしています。 ユーザーマクロを使用していて、その値が変更された場合（例: 1w → 1h）、次回のチェックは変更前の値に従って実行される点に注意してください（この例ではかなり先の時刻になります）。
Maximum concurrent checks per type	ディスカバリチェックを並列処理するため、サービスチェックごとのディスカバリスレッド（ワーカー）の最大数を設定します: **One** - 1 スレッド **Unlimited** - スレッド数無制限（ただし、StartDiscoverers パラメータの値を超えません） **Custom** - カスタムのスレッド数を設定（0-999） libsnmp 実装の特性により、SNMPv3 非同期サービスチェックを持つすべてのディスカバリルールは 1 つのワーカーで処理されるため、ワーカー数を増やしてもディスカバリ速度は向上しない点に注意してください。
Checks	Zabbix はディスカバリにこのチェック一覧を使用します。Add をクリックすると、ポップアップウィンドウで新しいチェックを設定できます。 サポートされるチェック: SSH、LDAP、SMTP、FTP、HTTP、HTTPS、POP、NNTP、IMAP、TCP、Telnet、Zabbix エージェント、SNMPv1 エージェント、SNMPv2 エージェント、SNMPv3 エージェント、ICMP ping。 プロトコルベースのディスカバリでは、SNMP が SNMP OID を問い合わせる場合を除き、各ホストのテストに **net.tcp.service[]** 機能を使用します。Zabbix エージェントは、暗号化されていないモードでアイテムを問い合わせることでテストされます。詳細はエージェントアイテム を参照してください。 'Ports' パラメータには、以下のいずれかを指定できます: 単一ポート: 22 ポート範囲: 22-45 リスト: 22-45,55,60-70 Zabbix 7.0 以降、LDAP チェックを除くすべてのサービスチェックは非同期で実行されます。 Zabbix 7.0 以降、HTTP/HTTPs チェックは libcurl 経由で実行されます。Zabbix サーバー/プロキシが libcurl なしでコンパイルされている場合、HTTP チェックは以前のバージョンと同様に動作します（つまり TCP チェックとして実行されます）が、HTTPS チェックは動作しません。
Device uniqueness criteria	一意性の判定基準には、以下を指定できます: **IP address** - 単一 IP のデバイスを複数処理しません。同じ IP を持つデバイスがすでに存在する場合、それはすでにディスカバリ済みと見なされ、新しいホストは追加されません。 **<discovery check>** - Zabbix エージェントチェックまたは SNMP エージェントチェック。 ディスカバリ時に使用される一意性の判定基準は、アクション実行時にシステムで使用されるホスト識別とは異なる点に注意してください。ディスカバリ時の一意性判定基準は、2 つ以上の検出されたデバイスが同一かどうかを定義するものですが、Zabbix におけるホスト識別の基準は IP アドレスのみです（ホストの作成 を参照）。
Host name	作成されるホストの技術的なホスト名を、以下を使用して設定します: **DNS name** - DNS 名（デフォルト） **IP address** - IP アドレス **<discovery check>** - ディスカバリチェックで受信した文字列値（例: Zabbix エージェント、SNMP エージェントチェック） あわせて参照: ホスト命名。
Visible name	作成されるホストの表示名を、以下を使用して設定します: **Host name** - 技術的なホスト名（デフォルト） **DNS name** - DNS 名 **IP address** - IP アドレス **<discovery check>** - ディスカバリチェックで受信した文字列値（例: Zabbix エージェント、SNMP エージェントチェック） あわせて参照: ホスト命名。
Enabled	チェックボックスをオンにすると、このルールは有効になり、Zabbix サーバーによって実行されます。 オフの場合、このルールは無効です。実行されません。

ファイルディスクリプタの上限超過

同時チェック数が多い場合、ディスカバリマネージャー のファイルディスクリプタ上限を使い切る可能性があります。

検出に必要なファイルディスクリプタ数は、ディスカバリワーカー数 * 1000 に相当します。デフォルトではディスカバリワーカー

は 5 つで、システムのソフト上限はおよそ 1024 です。

この上限に近づくと、Zabbix は各ワーカーについてタイプごとの同時チェックのデフォルト数を減らし、ログファイルに警告を書き込みます。ただし、ユーザーが タイプごとの最大同時チェック数に Zabbix が計算した値より大きい値を設定している場合、Zabbix は 1 つのワーカーに対してユーザー定義の値を使用します。

実際に使用するシナリオ

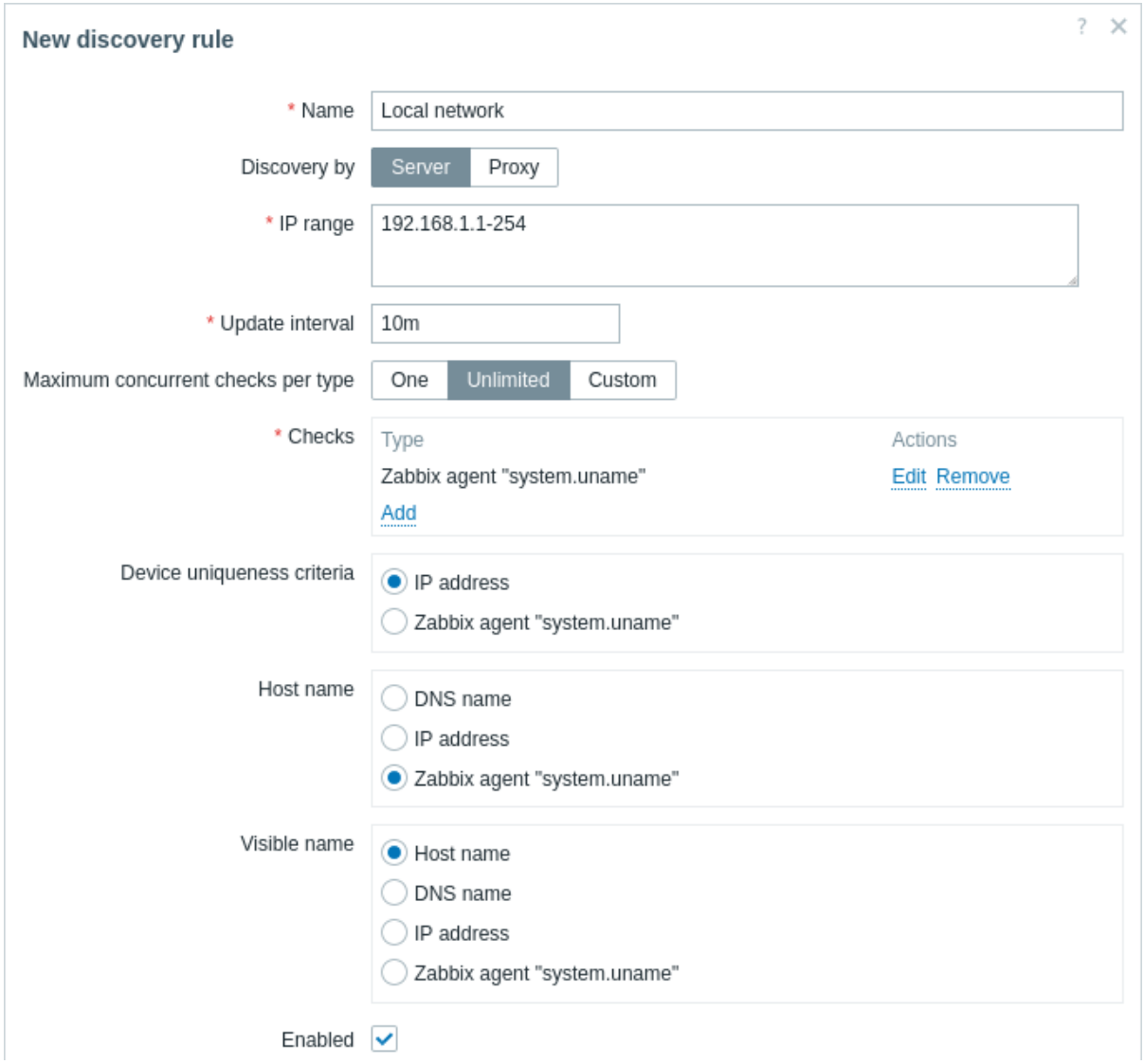
この例では、ローカルネットワークの 192.168.1.1～192.168.1.254 の範囲の IP アドレスをチェックするネットワークディスカバリを設定します。

このシナリオでは、以下のことをおこなう設定をします：

- Zabbix エージェントが動作しているホストを検出
- 検出は 10 分間隔で実行
- アップタイムが 1 時間以上のホストは監視対象に追加
- ダウンタイムが 24 時間以上のホストは監視対象から削除
- Linux ホストはグループ「Linux servers」に追加
- Windows ホストはグループ「Windows servers」に追加
- Linux ホストではテンプレート *Linux* を使用
- Windows ホストではテンプレート *Windows* を使用

ステップ 1

IP 範囲に対するネットワークディスカバリルールを定義します。



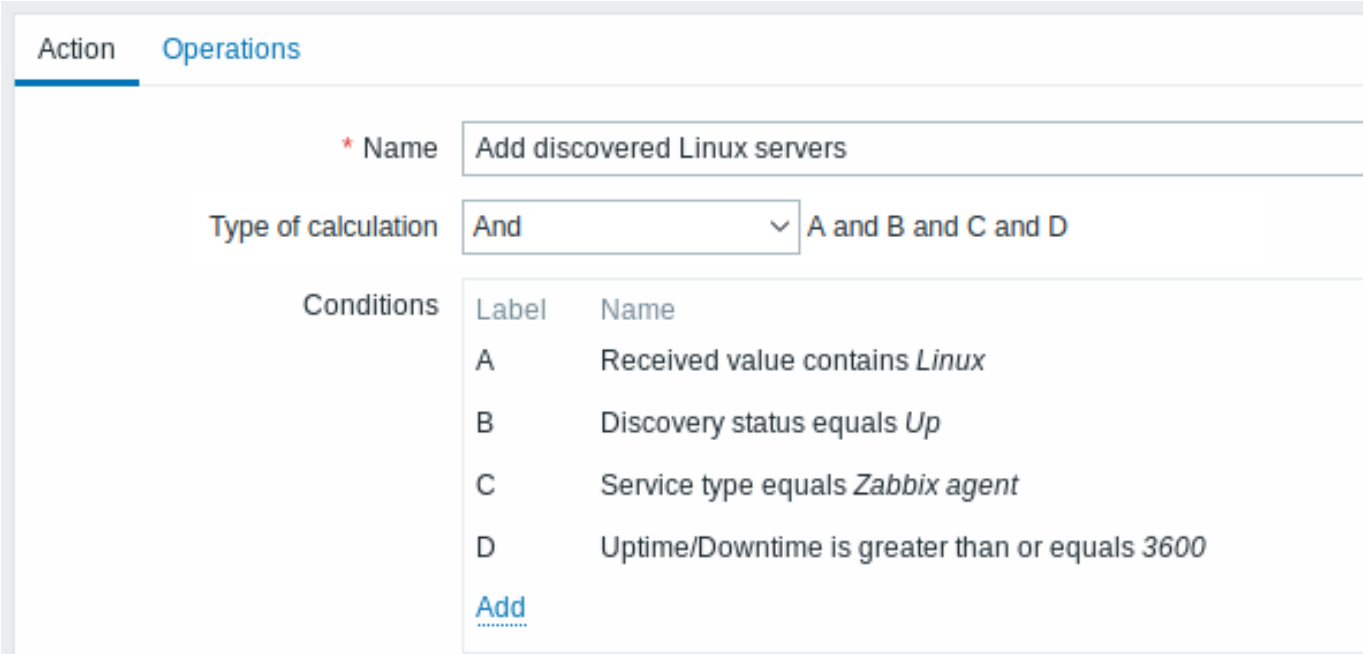
Zabbix は、Zabbix エージェントに接続して **system.uname** キーの値を取得することにより、192.168.1.1-192.168.1.254 の IP 範囲内にあるホストのディスカバリを試みます。エージェントから受信した値は、ホストの命名や、異なるオペレーティングシステムごとに異なるアクションを適用するために使用できます。たとえば、Windows サーバーをテンプレート *Windows* にリンクし、Linux サーバーをテンプレート *Linux* にリンクします。

このルールは 10 分ごとに実行されます。

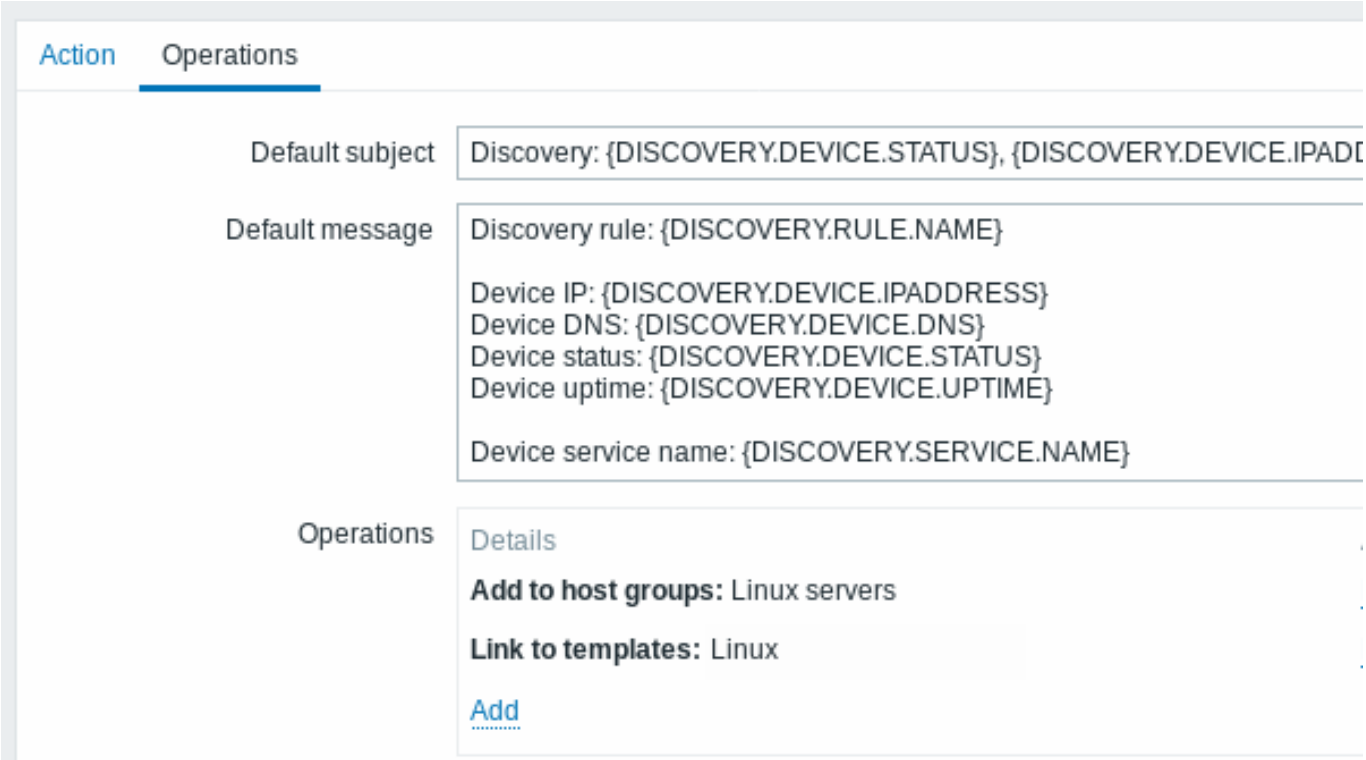
このルールを追加すると、Zabbix は自動的にディスカバリを開始し、以降の処理のためにディスカバリベースのイベントの生成を開始します。

ステップ 2

新たに検出された Linux サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。



このアクションは以下のコンディションが成立する場合有効になります。- サービス”Zabbix agent” が” 動作” している。- system.uname （ルールを定義する際に使用した Zabbix エージェントのキー）の値に”Linux” という文字列が含まれる。- アップタイムが 1 時間（3600 秒）を超えている。

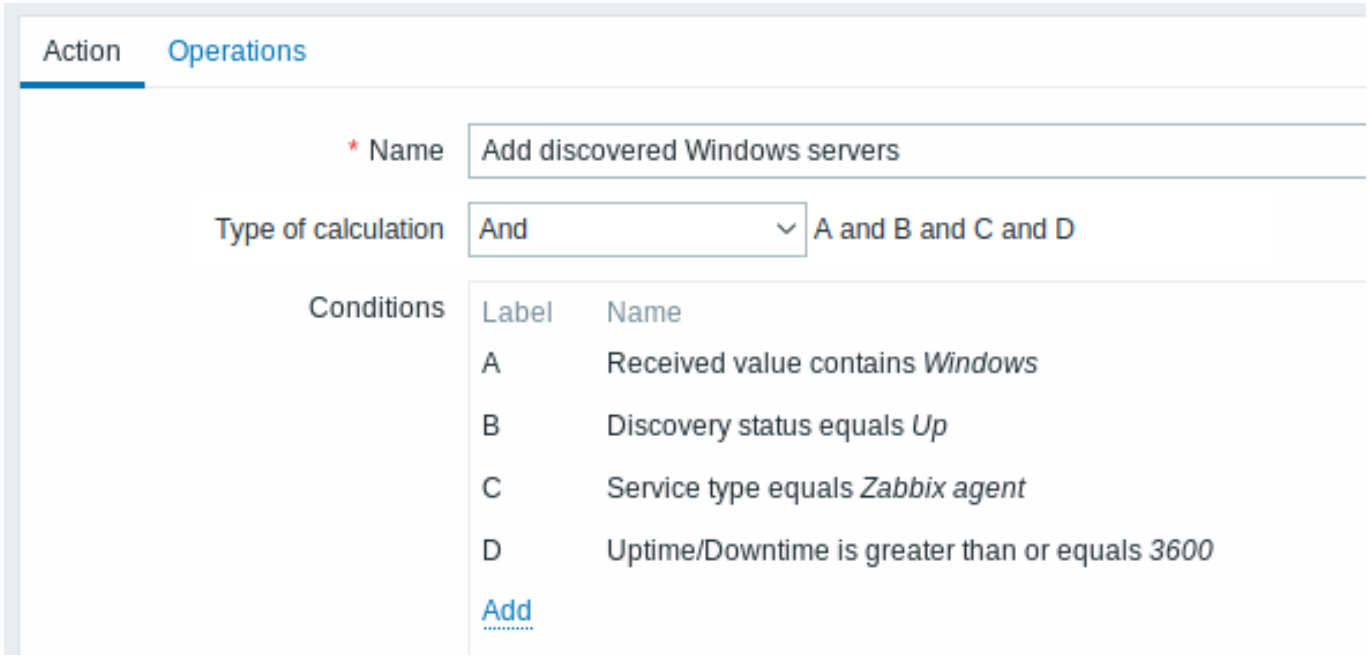


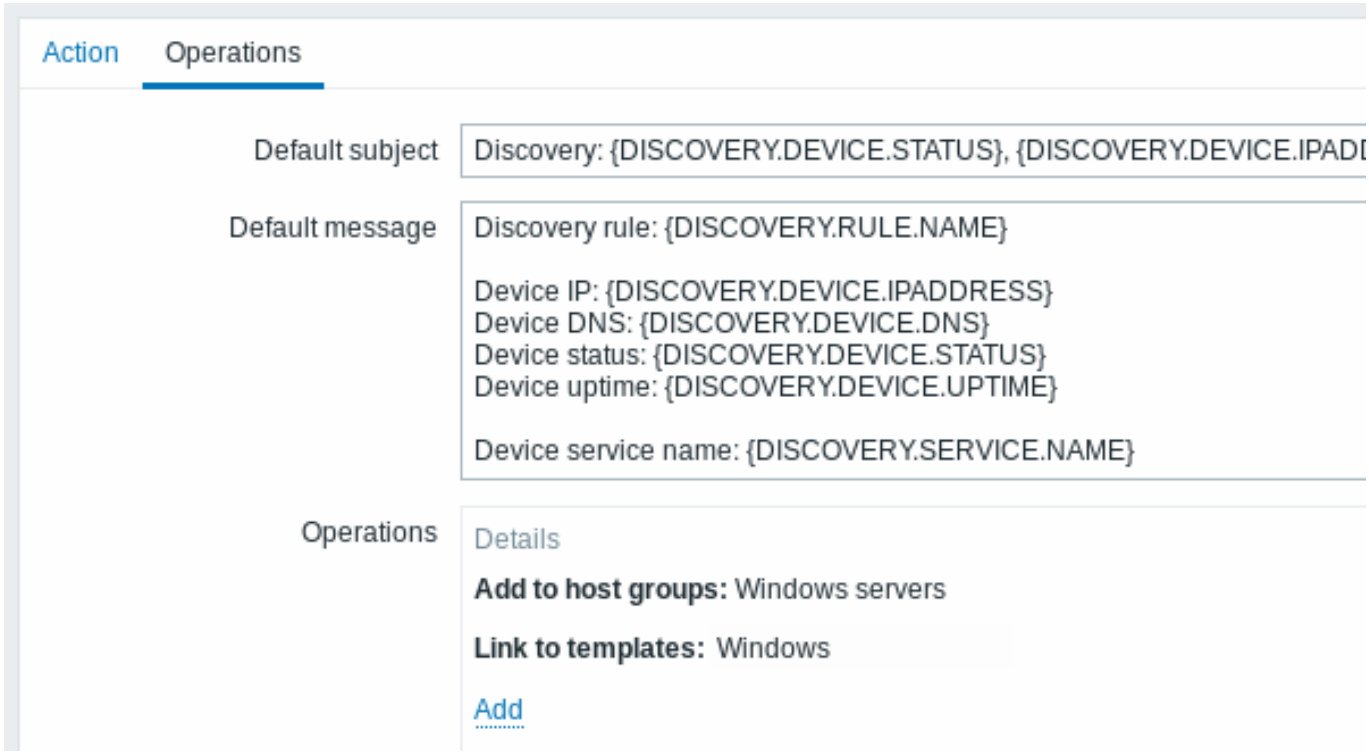
アクションによって以下のオペレーションが実行されます。

- 新たに検出したホストをグループ”Linux servers” に追加する。（まだ追加されていないホストの場合はホストの追加も実行する）
- ホストをテンプレート”Linux” にリンクする。”Linux” テンプレートに含まれるアイテムとトリガーを使用してホストの監視が自動的に開始される。

ステップ 3

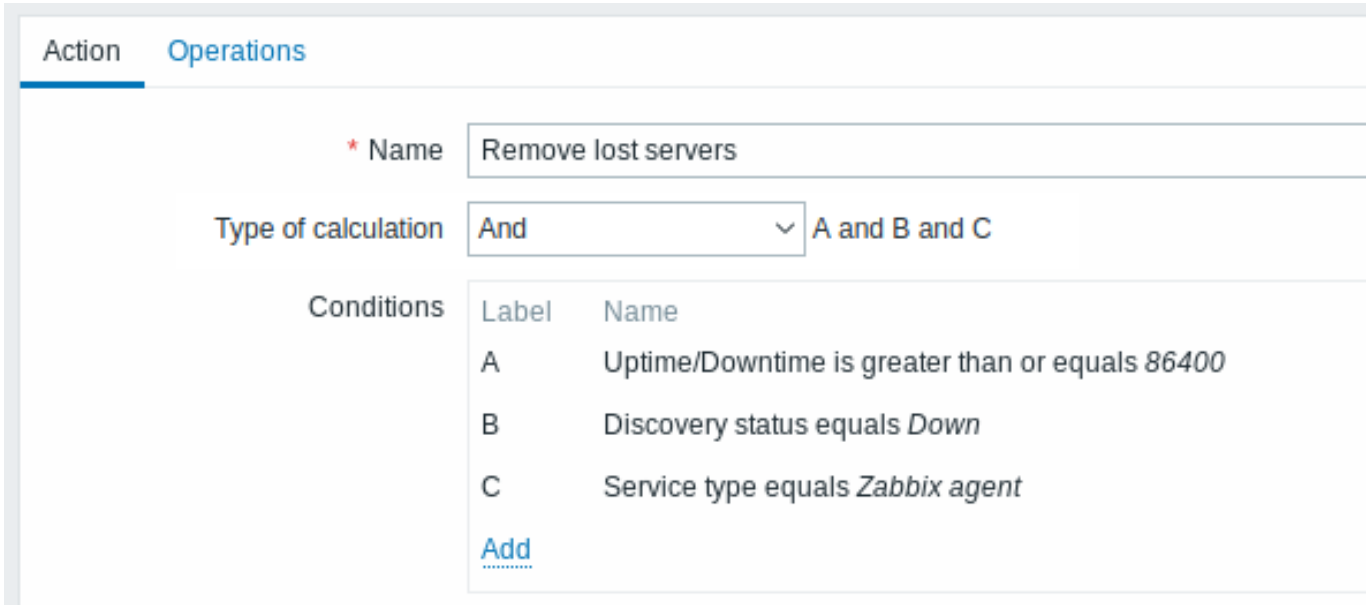
新たに検出された Windows サーバーをそれぞれのグループ/テンプレートに追加するための検出アクションを定義します。

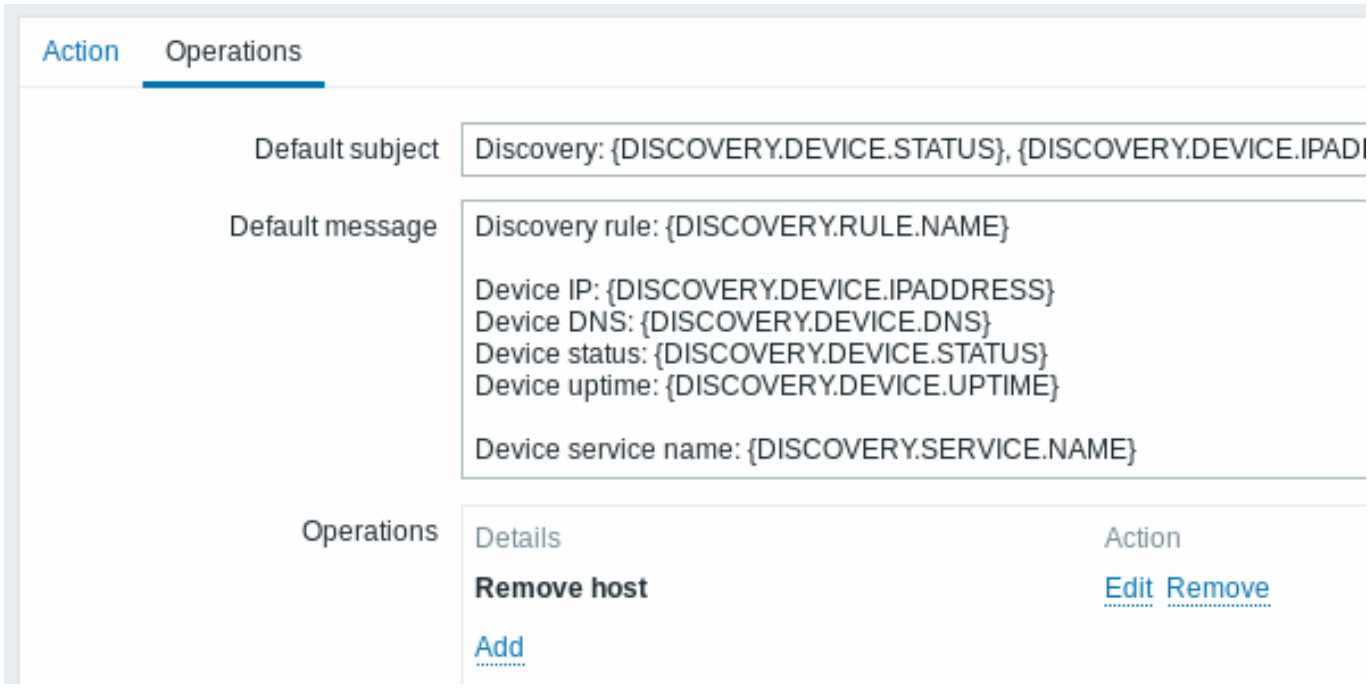




ステップ 4

動作していないサーバーを削除するための検出アクションを定義します。





”Zabbix agent” サービスが 24 時間（86400 秒）以上 down している場合、サーバーは削除されます。A server will be removed if ”Zabbix agent” service is ’down’ for more than 24 hours (86400 seconds).

2 アクティブエージェントの自動登録

概要

アクティブ Zabbix エージェントの自動登録を許可することができ、その後サーバーはそれらの監視を開始できます。これにより、新しいホストをサーバー上で手動設定することなく監視対象として追加できます。

自動登録は、これまで認識されていなかったアクティブエージェントがチェックを要求したときに実行されます。

この機能は、新しい Cloud ノードの自動監視に非常に便利です。Cloud に新しいノードが追加されるとすぐに、Zabbix はそのホストのパフォーマンスデータおよび可用性データの収集を自動的に開始します。

アクティブエージェントの自動登録では、追加されたホストをパッシブチェックで監視することもサポートしています。アクティブエージェントがチェックを要求する際、設定ファイルで`ListenIP` または`ListenPort` 設定パラメータが定義されていれば、それらもサーバーに送信されます。複数の IP アドレスが指定されている場合は、最初のものがサーバーに送信されます。

サーバーは、新しく自動登録されたホストを追加する際に、受信した IP アドレスとポートを使用してエージェントを設定します。IP アドレス値が受信されなかった場合は、受信接続に使用されたものが使われます。ポート値が受信されなかった場合は、10050 が使用されます。

ホストをデフォルトのエージェントインターフェースとしてDNS 名 で自動登録するよう指定することも可能です。

自動登録は、次の場合に再実行されます。

- ホストのメタデータ情報が変更された場合:
  - HostMetadata が変更され、エージェントが再起動された場合
  - HostMetadataItem が返す値が変更された場合
- メタデータが欠落している手動作成ホストの場合
- ホストが手動で別の Zabbix プロキシによって監視されるよう変更された場合
- 同じホストの自動登録が新しい Zabbix プロキシから行われた場合

Zabbix サーバーおよび Zabbix プロキシに対するアクティブエージェント自動登録のハートビートは 120 秒です。そのため、検出されたホストが削除された場合、自動登録は 120 秒後に再実行されます。

設定

サーバーを指定

エージェントの設定ファイル zabbix_agentd.conf で、Zabbix サーバーが指定されていることを確認してください。

```
ServerActive=10.0.0.1
```

zabbix_agentd.conf で *Hostname* を明示的に定義しない限り、エージェントが配置されているシステムのホスト名が、サーバーによってホスト名として使用されます。
Linux でシステムのホスト名を取得するには、hostname コマンドを実行します。

Zabbix エージェントの設定で *Hostname* がカンマ区切りのホスト一覧として定義されている場合、一覧に含まれるすべてのホスト名に対してホストが作成されます。

設定ファイルを変更した後は、エージェントを再起動してください。

アクティブエージェント自動登録のアクション

サーバーがエージェントから自動登録リクエストを受信すると、アクションを呼び出します。エージェントの自動登録には、イベントソース「自動登録」のアクションを設定する必要があります。

> **Note:**
> アクティブエージェントを自動登録するために、ネットワークディスカバリを設定する必要はありません。

Zabbix の Web インターフェースで、*Alerts* → *Actions* に移動し、*Autoregistration actions* を選択して *Create action* をクリックします。

- Action タブで、アクションに名前を付けます
- 必要に応じて、条件を指定します。
  条件では、ホスト名/ホストメタデータに対して部分文字列一致または正規表現一致を使用できます。
  「Host metadata」条件を使用する場合は、次のセクションを参照してください。
- Operations タブで、「Add host」、「Add to host group」(例: *Discovered hosts*)、「Link templates* など、関連するオペレーションを追加します。

> **Note:**
> 自動登録されるホストがアクティブ監視のみに対応する可能性が高い場合（たとえば、Zabbix サーバーからファイアウォールで遮断されているホストなど)、リンク用に *Template_Linux-active* のような専用テンプレートを作成するとよいでしょう。

作成されたホストは、*Discovered hosts* グループに追加されます（デフォルト。*Administration* > *General* > *Other* で設定可能)。ホストを別のグループに追加したい場合は、*Remove from host group* オペレーション（”Discovered hosts” を指定）を追加し、さらに *Add to host group* オペレーション（別のホストグループを指定）も追加してください。ホストは必ずいずれかのホストグループに属している必要があるためです。

安全な自動登録

暗号化接続で PSK ベースの認証を設定することで、安全な自動登録が可能です。

暗号化レベルは、*Administration* > *General* > *Autoregistration* でグローバルに設定します。暗号化なし、PSK 認証による TLS 暗号化、またはその両方を選択できます（つまり、一部のホストは暗号化なしで登録し、他のホストは暗号化を使用して登録できます)。

PSK による認証は、ホストを追加する前に Zabbix サーバーによって検証されます。成功した場合、ホストが追加され、*Connections from/to host* は、グローバルな自動登録設定と同じ ID/事前共有鍵を使用する「PSK」のみに設定されます。

Attention:
プロキシを使用する環境で自動登録のセキュリティを確保するには、Zabbix サーバーとプロキシ間の暗号化を有効にする必要があります。

デフォルトインターフェースとして DNS を使用する

HostInterface および HostInterfaceItem 設定パラメータ を使用すると、自動登録時のホストインターフェースにカスタム値を指定できます。

具体的には、ホストをデフォルトのエージェントインターフェースとして IP アドレスではなく DNS 名で自動登録する必要がある場合に有用です。その場合、HostInterface または HostInterfaceItem パラメータの値として DNS 名を指定するか、返す必要があります。これらのパラメータのいずれかの値が変更された場合（たとえば、IP アドレスから DNS 名へ、またはその逆）、自動登録されたホストのデフォルトインターフェースはそれに応じて更新されます。この更新は、新しいホストを作成するのではなく、既存のホストに対して適用されます。新しい値を送信するには、エージェントを再起動して自動登録プロセスを再開始させる必要があります。

Note:
HostInterface または HostInterfaceItem パラメータが設定されていない場合は、代わりに listen_dns パラメータが使用されます。この値は、エージェントの IP アドレスに対して逆引き DNS ルックアップを実行することで決定されます。逆引き DNS が適切に設定されていない場合、または無効な名前を返す場合、無効なインターフェース値により自動登録が不正確になったり失敗したりする可能性があります。

ホストメタデータの使用

エージェントがサーバーに自動登録リクエストを送信する際、ホスト名を送信します。
場合によっては（例えば、Amazon クラウドノードなど）、ホスト名だけでは Zabbix サーバーが検出されたホストを区別するのに十分ではありません。
ホストメタデータを任意で使用して、エージェントからサーバーへその他の情報を送信できます。

ホストメタデータは、エージェントの設定ファイル `zabbix_agentd.conf` で設定します。
設定ファイルでホストメタデータを指定する方法は 2 つあります。

```
HostMetadata
HostMetadataItem
```

各オプションの説明は、上記のリンクを参照してください。

HostMetadataItem パラメータは、最大 65535 個の UTF-8 コードポイントを返すことができます。
これより長い値は切り詰められます。

MySQL では、返される値にマルチバイト文字が含まれている場合、実効的な最大文字数はこれより少なくなることに注意してください。
例えば、3 バイト文字のみを含む値は合計 21844 文字まで、4 バイト文字のみを含む値は合計 16383 文字までに制限されます。

Attention:
アクティブエージェントがアクティブチェックを更新するためのリクエストをサーバーに送信するたびに、自動登録の試行が行われます。
リクエスト間の遅延は、エージェントのRefreshActiveChecks パラメータで指定します。
最初のリクエストは、エージェントの再起動直後に送信されます。

例

HostMetadata を使用した OS 別の自動登録

ホストを Zabbix サーバーによって自動登録したいとします。
ネットワーク上にはアクティブな Zabbix エージェント（上記の「Configuration」セクションを参照）が存在します。
また、ネットワーク上には Windows ホストと Linux ホストがあり、Zabbix の Web インターフェースでは「Linux by Zabbix agent」および「Windows by Zabbix agent」テンプレートを利用できます。
そのため、ホスト登録時には、登録されるホストに適切な Linux/Windows テンプレートを適用したいと考えます。
デフォルトでは、自動登録時にサーバーへ送信されるのはホスト名のみであり、それでは十分でない場合があります。
ホストに適切なテンプレートが適用されるようにするには、ホストメタデータを使用する必要があります。

Web インターフェースの設定

最初に行うことは、Web インターフェースを設定することです。
2 つのアクションを作成します。
1 つ目のアクション:

- 名前: Linux ホストの自動登録

- 条件: ホストメタデータに *Linux* が含まれる
- 実行内容: テンプレートをリンク: Linux by Zabbix agent

Note:
この場合、「ホストを追加」操作は省略できます。
ホストにテンプレートをリンクするには、まずホストを追加する必要があるため、サーバーがそれを自動的に行います。

2 つ目のアクション:

- 名前: Windows ホストの自動登録
- 条件: ホストメタデータに *Windows* が含まれる
- 実行内容: テンプレートをリンク: Windows by Zabbix agent

エージェントの設定

次に、エージェントを設定する必要があります。
以下の行をエージェントの設定ファイルに追加してください。

```
HostMetadataItem=system.uname
```

これにより、ホストメタデータには、エージェントが稼働しているホストに応じて”Linux” または”Windows” が含まれるようになります。
この場合のホストメタデータの例:

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-OPXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

設定ファイルを変更した後は、エージェントの再起動を忘れないでください。

HostMetadata を使用した自動登録とテンプレートの制御

ステップ ***1 - HostMetadata*** で登録を保護する

ホストメタデータを使用して、不要なホストの登録に対する基本的な保護を行います。

Web インターフェースの設定

推測されにくい秘密コードを使用して、不要なホストを許可しないように、Web インターフェースでアクションを作成します。

- 名前: Linux の自動登録アクション
- 条件:
  - 計算の種類: AND
  - 条件 (A): ホストメタデータに //Linux// を含む
  - 条件 (B): ホストメタデータに //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae// を含む
- 実行内容:
  - すべてのメディア経由でユーザー Admin にメッセージを送信
  - ホストグループに追加: Linux servers
  - テンプレートをリンク: Zabbix エージェントによる Linux

この方法だけでは強力な保護は提供されないことに注意してください。これは、データが平文で送信されるためです。変更を即座に反映させるには、設定キャッシュの再読み込みが必要です。

エージェントの設定

次の行をエージェントの設定ファイルに追加します。

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

ここで、”Linux” はプラットフォームであり、文字列の残りの部分は推測されにくい秘密のテキストです。

設定ファイルに変更を加えた後は、エージェントを再起動することを忘れないでください。

ステップ ***2*** - 登録済みホストにテンプレートを追加

すでに登録されているホストに、追加のテンプレートを追加できます。
この場合、MySQL by Zabbix agent テンプレートは、HostMetadata に MySQL トークンを含むホストにのみリンクされます。

Web インターフェースの設定

Web インターフェースでアクションを更新します。

- 名前: Autoregistration action Linux
- 条件:
  - 計算タイプ: AND

    - 条件 (A): ホストメタデータに Linux を含む
    - 条件 (B): ホストメタデータに 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae を含む
    - 条件 (C): ホストメタデータに MySQL を含む
- 実行内容:
    - ユーザーへのメッセージ送信: すべてのメディア経由で Admin
    - ホストグループに追加: Linux servers
    - テンプレートをリンク: Linux by Zabbix agent
    - テンプレートをリンク: MySQL by Zabbix Agent

エージェントの設定

エージェント設定ファイル内の次の行を更新します。

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

設定ファイルを変更した後は、エージェントを再起動することを忘れないでください。

3 ローレベルディスカバリ

概要　ローレベルディスカバリ（LLD）は、ホスト上のさまざまなエンティティに対して、アイテム、トリガー、グラフを自動的に作成する方法を提供します。たとえば、Zabbix は、各ファイルシステムやネットワークインターフェースごとにアイテムを手動で作成しなくても、マシン上のファイルシステムやネットワークインターフェースの監視を自動的に開始できます。LLD はホストの作成にも使用でき、たとえばハイパーバイザー上で検出された仮想マシンを追加したり、ネストされたディスカバリルールによって多段階のディスカバリを可能にしたりできます。さらに、定期的に実行されるディスカバリの実際の結果に基づいて、不要になったエンティティを自動的に削除するよう Zabbix を設定することも可能です。

ユーザーは、特定の JSON プロトコルに従う限り、独自のディスカバリタイプを定義できます。

ディスカバリプロセスの一般的なアーキテクチャは次のとおりです。

まず、ユーザーは データ収集 > テンプレートの ディスカバリ列でディスカバリルールを作成します。ディスカバリルールは、(1) 必要なエンティティ（たとえば、ファイルシステムやネットワークインターフェース）を検出するアイテムと、(2) そのアイテムの値に基づいて作成されるアイテム、トリガー、グラフのプロトタイプで構成されます。

必要なエンティティを検出するアイテムは、他で見られる通常のアイテムと同様です。サーバーは Zabbix エージェント（または、そのアイテムに設定されたタイプに応じたもの）にそのアイテムの値を要求し、エージェントはテキスト値を返します。違いは、エージェントが返す値に、検出されたエンティティの一覧が JSON 形式で含まれていなければならない点です。この形式の詳細は、カスタムディスカバリチェックの実装者にとってのみ重要ですが、返される値にはマクロ → 値のペアの一覧が含まれることを知っておく必要があります。たとえば、アイテム”net.if.discovery” は、”{#IFNAME}” → ”lo” および”{#IFNAME}” → ”eth0” という 2 つのペアを返す場合があります。

これらのマクロは、名前、キー、その他のプロトタイプフィールドで使用され、受信した値に置き換えられて、検出された各エンティティに対する実際のアイテム、トリガー、グラフ、さらにはホストの作成に利用されます。LLD マクロの使用方法については、オプションの完全な一覧を参照してください。

サーバーがディスカバリアイテムの値を受信すると、マクロ → 値のペアを確認し、各ペアについて、そのプロトタイプに基づいて実際のアイテム、トリガー、グラフを生成します。上記の”net.if.discovery” の例では、サーバーはループバックインターフェース”lo” 用に 1 セットのアイテム、トリガー、グラフを生成し、インターフェース”eth0” 用に別の 1 セットを生成します。

Zabbix 4.2 以降、ローレベルディスカバリルールが返す JSON の形式が変更されている点に注意してください。JSON に”data” オブジェクトが含まれていることは、もはや想定されていません。ローレベルディスカバリは、アイテム値の前処理や JSON ドキュメント内のローレベルディスカバリマクロ値へのカスタムパスといった新機能をサポートするため、配列を含む通常の JSON を受け付けるようになりました。

組み込みのディスカバリキーは、JSON ドキュメントのルートで LLD 行の配列を返すよう更新されました。配列フィールドがキーとして {#MACRO} 構文を使用している場合、Zabbix はマクロと値を自動的に抽出します。今後の新しいネイティブディスカバリチェックでは、”data” 要素を含まない新しい構文が使用されます。ローレベルディスカバリ値を処理する際には、まずルート（`$.` または `$.data` の配列）が特定されます）。

”data” 要素はディスカバリ関連のすべてのネイティブアイテムから削除されましたが、後方互換性のため、Zabbix は引き続き”data” 要素を含む JSON 表記も受け付けます。ただし、その使用は推奨されません。JSON に”data” 配列要素のみを 1 つ持つオブジェクトが含まれている場合、JSONPath `$.data` を使用してその要素の内容が自動的に抽出されます。ローレベルディスカバリは、JSONPath 構文で指定されたカスタムパスを持つ、ユーザー定義の任意の LLD マクロも受け付けるようになりました。

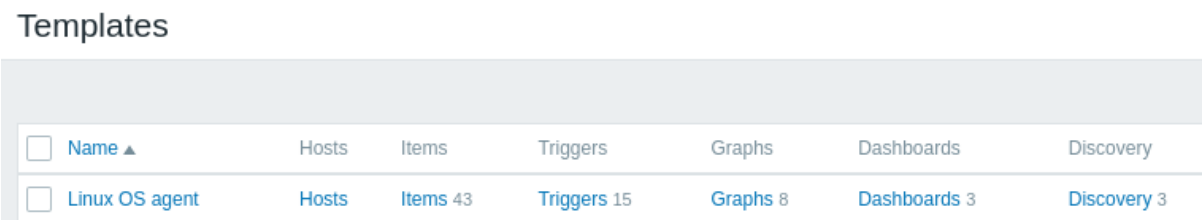
Warning:
上記の変更の結果、新しいエージェントは古い Zabbix サーバーでは動作しなくなります。

参照: 検出されたエンティティ

ローレベルディスカバリの設定　ファイルシステムディスカバリの例に基づいて、ローレベルディスカバリを説明します。

ディスカバリを設定するには、次の手順を実行します。

- データ収集 > テンプレートまたは ホストに移動します。
- 適切なテンプレート/ホストの行にある ディスカバリをクリックします。



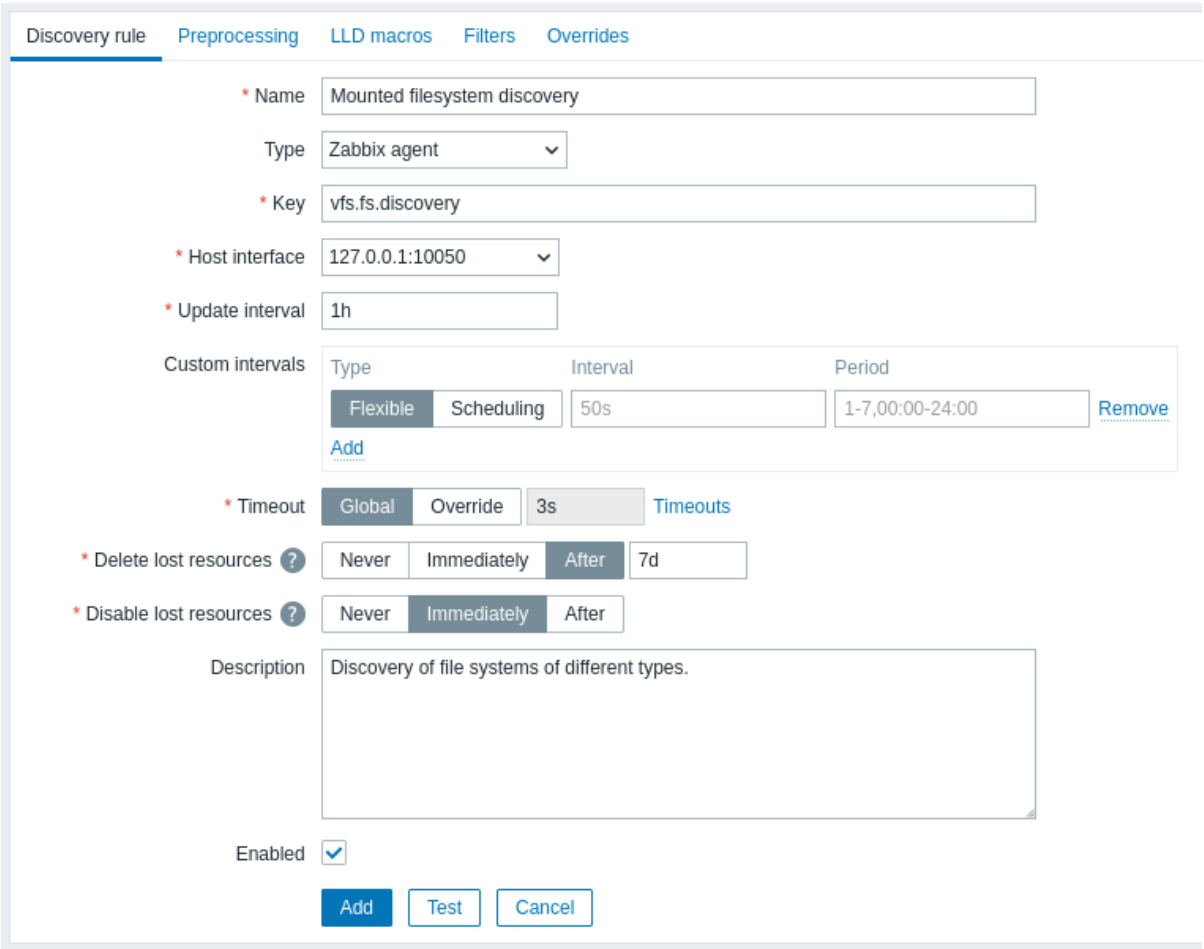
- 画面右上の ディスカバリルールの作成をクリックします。
- 必要な詳細を入力して、ディスカバリルールのフォームを記入します。

ディスカバリルール

ディスカバリルールのフォームには 5 つのタブがあり、左から右へ、ディスカバリ中のデータフローを表しています。

- ディスカバリルール - ディスカバリデータを取得するための組み込みアイテムまたはカスタムスクリプトを主に指定します。
- 前処理 - 検出されたデータに対して前処理を適用します。
- *LLD* マクロ - 検出されたアイテム、トリガーなどで使用するマクロ値を抽出できます。
- フィルター - 検出された値をフィルタリングできます。
- オーバーライド - 特定の検出オブジェクトに適用する際に、アイテム、トリガー、グラフ、またはホストプロトタイプを変更できます。

ディスカバリルールタブには、ディスカバリに使用するアイテムキー（および一般的なディスカバリルール属性）が含まれます。



必須入力フィールドにはすべて赤いアスタリスクが付いています。

Parameter	Description
Name	ディスカバリルールの名前。
Type	ディスカバリを実行するチェックのタイプ。 この例では、*Zabbix* エージェントアイテムタイプを使用しています。 ディスカバリルールは、通常のアイテムに依存するdependent item にすることもできます。別のディスカバリルールに依存させることはできません。依存アイテムの場合は、該当するタイプ（*Dependent item*）を選択し、'Master item' フィールドでマスターアイテムを指定します。マスターアイテムは存在している必要があります。
Key	ディスカバリアイテムキーを入力します（最大 2048 文字）。 たとえば、組み込みの"vfs.fs.discovery" アイテムキーを使用すると、コンピューター上に存在するファイルシステムの一覧、そのタイプ、およびマウントオプションを含む JSON 文字列を返すことができます。 ファイルシステムディスカバリの別の方法として、"vfs.fs.get" エージェントキーによるディスカバリ結果を使用することもできます（example を参照）。
Update interval	このフィールドでは、Zabbix がどのくらいの頻度でディスカバリを実行するかを指定します。ファイルシステムディスカバリを設定し始めたばかりの段階では短い間隔に設定したい場合がありますが、動作することが確認できたら、通常ファイルシステムはそれほど頻繁には変化しないため、30 分以上に設定できます。 Time suffixes がサポートされています（例: 30s、1m、2h、1d）。 User macros がサポートされています。 *Note*: 更新間隔を'0' に設定できるのは、0 以外の値を持つカスタム間隔が存在する場合のみです。'0' に設定され、0 以外の値を持つカスタム間隔（柔軟またはスケジュール）が存在する場合、アイテムはそのカスタム間隔の期間中にポーリングされます。 新しいディスカバリルールは、Scheduling または Flexible update interval があり、かつ *Update interval* が 0 に設定されている場合を除き、作成後 60 秒以内にチェックされます。 既存のディスカバリルールについては、*Execute now* button を押すことで、直ちにディスカバリを実行できることに注意してください。
Custom intervals	アイテムをチェックするためのカスタムルールを作成できます。 **Flexible** - *Update interval* の例外を作成します（異なる頻度の間隔） **Scheduling** - カスタムのポーリングスケジュールを作成します。 詳細はCustom intervals を参照してください。
Timeout	ディスカバリチェックのタイムアウトを設定します。タイムアウトオプションを選択します。 **Global** - プロキシ/グローバルタイムアウトが使用されます（グレー表示された *Timeout* フィールドに表示されます）。 **Override** - カスタムタイムアウトが使用されます（*Timeout* フィールドで設定。許容範囲: 1 - 600s）。Time suffixes（例: 30s、1m）およびuser macros がサポートされています。 *Timeouts* リンクをクリックすると、プロキシ のタイムアウト、または（プロキシを使用しない場合は）global タイムアウトを設定できます。*Timeouts* リンクは、*Administration* > *General* または *Administration* > *Proxies* の Web インターフェースセクションへの権限を持つ *Super admin* タイプのユーザーにのみ表示されることに注意してください。
Delete lost resources	検出状態が"Not discovered anymore" になった後、検出されたエンティティをどのくらいで削除するかを指定します。 *Never* - 削除されません。 *Immediately* - 直ちに削除されます。 *After* - 指定した期間後に削除されます。この値は *Disable lost resources* の値より大きくなければなりません。 Time suffixes がサポートされています（例: 2h、1d）。 User macros がサポートされています。 *Note*: "Immediately" の使用は推奨されません。フィルターを誤って編集しただけで、すべての履歴データとともにエンティティが削除される可能性があるためです。 手動で無効化されたリソースは、ローレベルディスカバリによって削除されないことに注意してください。

Parameter	Description
Disable lost resources	検出状態が”Not discovered anymore” になった後、検出されたエンティティをどのくらいで無効化するかを指定します。 *Never* - 無効化されません。 *Immediately* - 直ちに無効化されます。 *After* - 指定した期間後に無効化されます。この値はディスカバリルールの更新間隔より大きい必要があります。 自動的に無効化されたリソースは、ローレベルディスカバリによって再検出されると再び有効になることに注意してください。手動で無効化されたリソースは、再検出されても再び有効にはなりません。 *Delete lost resources* が”Immediately” に設定されている場合、このフィールドは表示されません。 Time suffixes がサポートされています（例: 2h、1d）。 User macros がサポートされています。
Description	説明を入力します。
Enabled	チェックすると、ルールが処理されます。

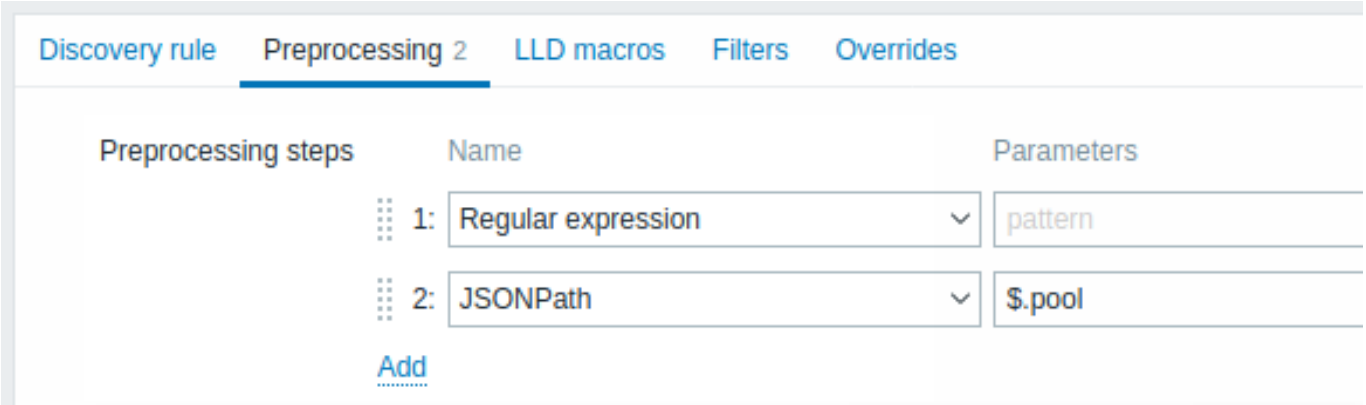
Note:
ディスカバリルールの履歴は保持されません。

前処理

前処理タブでは、ディスカバリの結果に適用する変換ルールを定義できます。
このステップでは、1 つまたは複数の変換を設定できます。
変換は、定義された順序で実行されます。
すべての前処理は Zabbix サーバーによって実行されます。

以下も参照してください:

- 前処理の詳細
- 前処理のテスト



Type		
	変換	説明
Text		
	正規表現	受信した値を <pattern> 正規表現に照合し、抽出した <output> で値を置き換えます。正規表現では、\N シーケンスを使用して最大 10 個のキャプチャグループを抽出できます。 パラメータ: **pattern** - 正規表現 **output** - 出力書式テンプレート。\N（N=1...9）エスケープシーケンスは N 番目に一致したグループに置き換えられます。\0 エスケープシーケンスは一致したテキストに置き換えられます。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。

Type	
置換	検索文字列を見つけて別の文字列（または何もない文字列）に置き換えます。検索文字列のすべての出現箇所が置き換えられます。 パラメータ: **search string** - 検索および置換する文字列。大文字と小文字は区別されます（必須） **replacement** - search string を置き換える文字列。置換文字列は空にすることもでき、その場合、見つかった search string を削除できます。 改行、復帰、タブ、スペースを検索または置換するために、エスケープシーケンス”\n \r \t \s” を使用できます。バックスラッシュは”\\” としてエスケープでき、エスケープシーケンス自体は”\\n” のようにエスケープできます。改行、復帰、タブのエスケープは、ローレベルディスカバリ中に自動的に行われます。
Structured data	
JSONPath	JSONPath 機能を使用して、JSON データから値または断片を抽出します。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
XML XPath	XPath 機能を使用して、XML データから値または断片を抽出します。 このオプションを機能させるには、Zabbix サーバーが libxml サポート付きでコンパイルされている必要があります。 例: `number(/document/item/value)` は `<document><item><value>10</value></item></document>` から 10 を抽出します `number(/document/item/@attribute)` は `<document><item attribute="10"></item></document>` から 10 を抽出します `/document/item` は `<document><item><value>10</value></item>` を `<document><item><value>10</value></item></document>` から抽出します 名前空間はサポートされていないことに注意してください。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
CSV to JSON	CSV ファイルデータを JSON 形式に変換します。 詳細はCSV to JSON preprocessing を参照してください。
XML to JSON	XML 形式のデータを JSON に変換します。 詳細はSerialization rules を参照してください。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
SNMP	
SNMP walk value	指定した OID/MIB 名で値を抽出し、書式設定オプションを適用します: **Unchanged** - Hex-STRING をエスケープされていない 16 進文字列として返します（注意: 表示ヒントは引き続き適用されます）。 **UTF-8 from Hex-STRING** - Hex-STRING を UTF-8 文字列に変換します。 **MAC from Hex-STRING** - Hex-STRING を MAC アドレス文字列に変換します（このとき `' '` は `':'` に置き換えられます）。 **Integer from BITS** - 16 進文字の並びで表現されたビット文字列（例: ”1A 2B 3C 4D”）の先頭 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、それ以降のバイトは無視されます。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
SNMP walk to JSON	SNMP 値を JSON に変換します。JSON 内のフィールド名と、それに対応する SNMP OID パスを指定します。フィールド値には、指定した SNMP OID パスの値が設定されます。 この前処理ステップは、SNMP OID discovery に使用できます。 *SNMP walk value* ステップと同様の値書式設定オプションを利用できます。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。

Type	
SNMP get value	SNMP get 値に書式設定オプションを適用します: **UTF-8 from Hex-STRING** - Hex-STRING を UTF-8 文字列に変換します。 **MAC from Hex-STRING** - Hex-STRING を MAC アドレス文字列に変換します（このとき ' ' は ':' に置き換えられます）。 **Integer from BITS** - 16 進文字の並びで表現されたビット文字列（例: "1A 2B 3C 4D"）の先頭 8 バイトを、64 ビット符号なし整数に変換します。8 バイトを超えるビット文字列では、それ以降のバイトは無視されます。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
Custom scripts	
JavaScript	パラメータフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディタに JavaScript コードを入力します。 使用可能な JavaScript の長さは、使用しているデータベース に依存することに注意してください。 詳細はJavascript preprocessing を参照してください。
Validation	
正規表現に一致しない	値が一致してはならない正規表現を指定します。 例: `Error:(.*?)\.` *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
JSON 内のエラーを確認	JSONPath にあるアプリケーションレベルのエラーメッセージを確認します。成功し、かつメッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、前処理ステップ情報を追加せず、そのままユーザーに報告されることに注意してください。 例: `$.errors`。`{"errors":"e1"}` のような JSON を受信した場合、次の前処理ステップは実行されません。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
XML 内のエラーを確認	Xpath にあるアプリケーションレベルのエラーメッセージを確認します。成功し、かつメッセージが空でない場合は処理を停止します。それ以外の場合は、この前処理ステップの前の値で処理を続行します。これらの外部サービスエラーは、前処理ステップ情報を追加せず、そのままユーザーに報告されることに注意してください。 無効な XML の解析に失敗した場合、エラーは報告されません。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
正規表現に一致する	値が一致しなければならない正規表現を指定します。 *Custom on fail* チェックボックスをオンにすると、カスタムのエラー処理オプションを指定できます。値を破棄する、指定した値を設定する、または指定したエラーメッセージを設定することが可能です。
Throttling	
ハートビート付きで未変更を破棄	定義した期間（秒）内に値が変化していない場合、その値を破棄します。 秒数の指定には正の整数値を使用できます（最小 1 秒）。このフィールドでは時間サフィックスを使用できます（例: 30s、1m、2h、1d）。このフィールドではユーザーマクロおよびローレベルディスカバリマクロを使用できます。 ディスカバリアイテムには、スロットリングオプションを 1 つだけ指定できます。 例: `1m`。同一のテキストが 60 秒以内にこのルールへ 2 回渡された場合、その値は破棄されます。 注意: アイテムのプロトタイプを変更してもスロットリングはリセットされません。スロットリングがリセットされるのは、前処理ステップが変更された場合のみです。
Prometheus	
Prometheus to JSON	必要な Prometheus メトリクスを JSON に変換します。 詳細はPrometheus checks を参照してください。

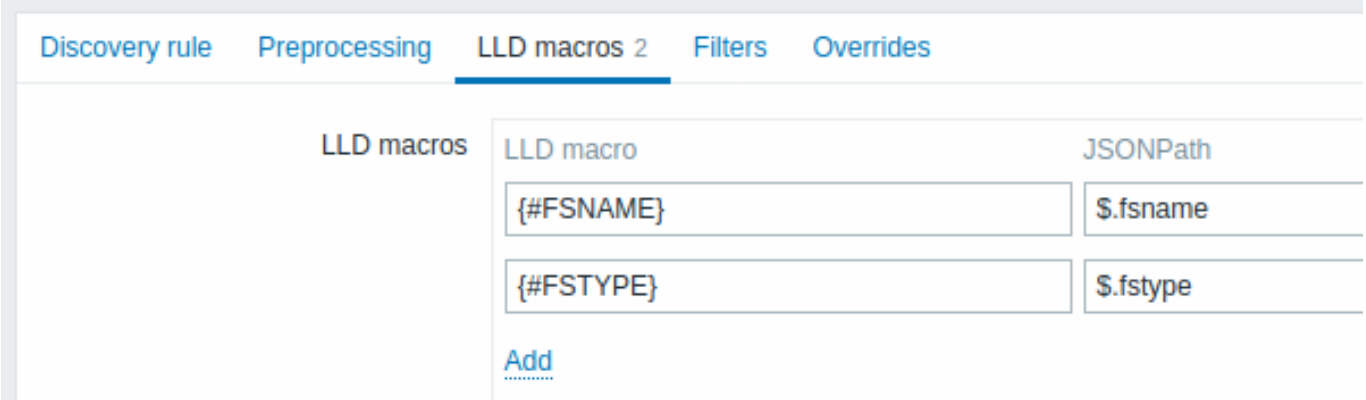
ディスカバリルールがテンプレート経由でホストに適用されている場合、このタブの内容は読み取り専用になることに注意してください。

カスタムマクロ

LLD マクロタブでは、カスタムのローレベルディスカバリマクロを指定できます。

カスタムマクロは、返された JSON に必要なマクロがあらかじめ定義されていない場合に役立ちます。例えば、次のような場合です。

- ファイルシステムディスカバリ用のネイティブな `vfs.fs.discovery` キーは、{#FSNAME}、{#FSTYPE} などの事前定義された LLD マクロを含む JSON を返します。これらのマクロは、アイテムやトリガーのプロトタイプ（ページの後続セクションを参照）で直接使用できるため、カスタムマクロを定義する必要はありません。
- `vfs.fs.get` エージェントアイテムも、ファイルシステムデータを含む JSON を返しますが、事前定義された LLD マクロは含まれていません。この場合は、自分でマクロを定義し、JSONPath を使用して JSON 内の値にマッピングできます。

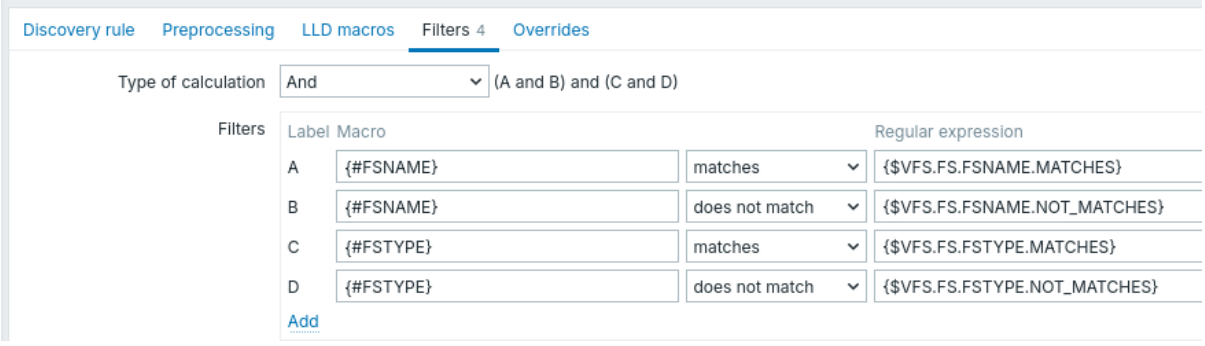


抽出された値は、ディスカバリされたアイテム、トリガーなどで使用できます。値は、ディスカバリの結果およびそれまでに適用されたすべての前処理ステップの結果から抽出されることに注意してください。

Parameter	Description
LLD macro	ローレベルディスカバリマクロの名前。構文は次のとおりです: {#MACRO}。
JSONPath	JSONPath 構文を使用して、LLD 行から LLD マクロ値を抽出するためのパスです。 返された JSON から抽出された値は、アイテム、トリガーなどのプロトタイプフィールド内の LLD マクロを置き換えるために使用されます。 JSONPath はドット記法またはブラケット記法で指定できます。特殊文字や Unicode を含む場合は、`$['unicode + special chars #1']['unicode + special chars #2']` のようにブラケット記法を使用する必要があります。 例えば、`$.foo` はこの JSON から"bar" と"baz" を抽出します: `[{"foo":"bar"}, {"foo":"baz"}]` `$.foo` はこの JSON からも"bar" と"baz" を抽出することに注意してください: `{"data":[{"foo":"bar"}, {"foo":"baz"}]}`。これは、単一の"data" オブジェクトが自動的に処理されるためです（4.2 より前の Zabbix バージョンにおけるローレベルディスカバリ実装との後方互換性のため）。

フィルター

フィルターを使用すると、条件に一致するエンティティに対してのみ、実際のアイテム、トリガー、グラフを生成できます。
フィルタータブには、ディスカバリ値を絞り込むためのディスカバリルールのフィルター定義が含まれています。



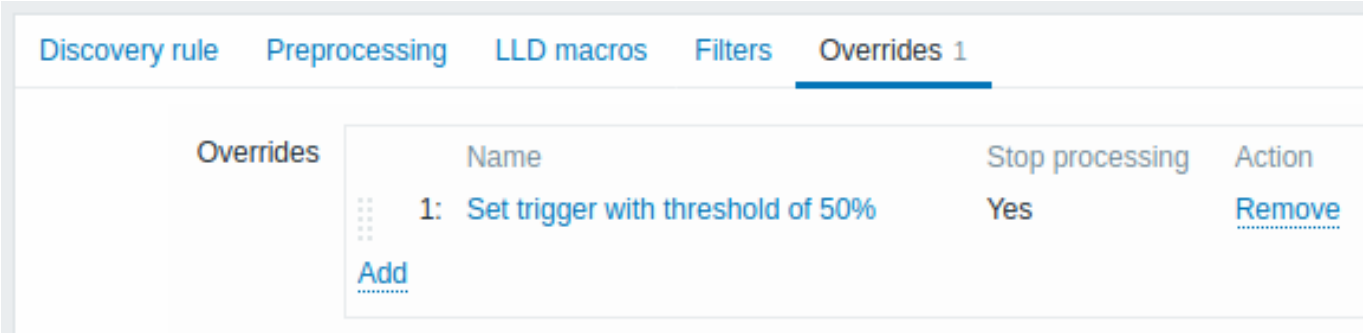
Parameter	Description
Type of calculation	フィルターの計算方法には、次のオプションがあります。 **And** - すべてのフィルター条件を満たす必要があります。 **Or** - いずれか 1 つのフィルター条件を満たせば十分です。 **And/Or** - 異なるマクロ名には *And*、同じマクロ名には *Or* を使用します。 **Custom expression** - フィルターのカスタム計算式を定義できます。式には、リスト内のすべてのフィルターを含める必要があります。255 文字までに制限されます。
Filters	使用可能なフィルター条件演算子は次のとおりです: *matches*、*does not match*、*exists*、*does not exist*。 *matches* および *does not match* 演算子では、Perl Compatible Regular Expression (PCRE) を指定します。たとえば、C:、D:、E: のファイルシステムのみに関心がある場合は、「Macro」に {#FSNAME} を、「Regular expression」のテキストフィールドに”^C|^D|^E” 正規表現を入力できます。ファイルシステムタイプによるフィルタリングも {#FSTYPE} マクロ（例: ”^ext|^reiserfs”）を使用して可能です。また、ドライブタイプによるフィルタリングも {#FSDRIVETYPE} マクロ（例: ”fixed”）を使用して可能です（Windows エージェントでのみサポート）。 「Regular expression」フィールドには、正規表現を直接入力することも、グローバルな正規表現を参照することもできます。 正規表現をテストするには、たとえば”grep -E” を使用できます: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` *exists* および *does not exist* 演算子を使用すると、レスポンス内に指定した LLD マクロが存在するかどうかに基づいてエンティティをフィルタリングできます。 フィルター内のマクロがレスポンスに存在しない場合、そのマクロに対して”does not exist” 条件が指定されていない限り、検出されたエンティティは無視されることに注意してください。 マクロの欠落が式の結果に影響する場合は、警告が表示されます。たとえば、次の場合に {#B} が欠落しているとします。 {#A} matches 1 and {#B} matches 2 - 警告が表示されます {#A} matches 1 or {#B} matches 2 - 警告は表示されません

Warning:
LLD ルールで使用される正規表現に誤りやタイプミスがあると（たとえば、不正な「File systems for discovery」正規表現）、多数のホストに対して数千もの設定要素、履歴値、イベントが削除される可能性があります。

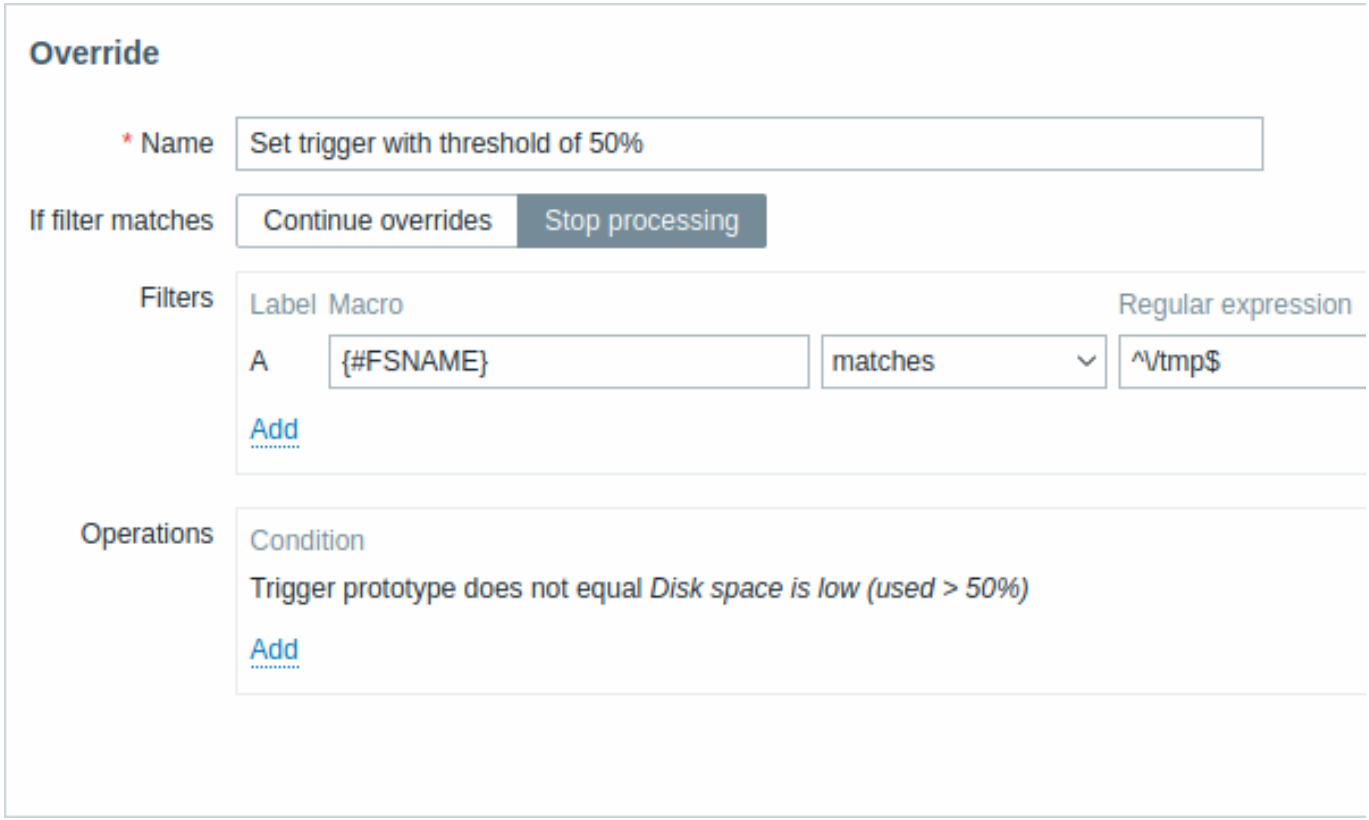
Attention:
大文字と小文字の違いだけが異なるファイルシステム名を正しくディスカバリする必要がある場合、MySQL の Zabbix データベースは大文字と小文字を区別する設定で作成する必要があります。

オーバーライド

オーバーライドタブでは、指定した条件に一致する検出済みオブジェクトに対して、アイテム、トリガー、グラフ、ホスト、ディスカバリの各プロトタイプ、またはそれらの属性の一覧を変更するルールを設定できます。



オーバーライド（存在する場合）は、ドラッグ＆ドロップで並べ替え可能な一覧として表示され、定義された順序で実行されます。新しいオーバーライドの詳細を設定するには、*Overrides* ブロック内の Add をクリックします。
既存のオーバーライドを編集するには、オーバーライド名をクリックします。
オーバーライドルールの詳細を編集できるポップアップウィンドウが開きます。

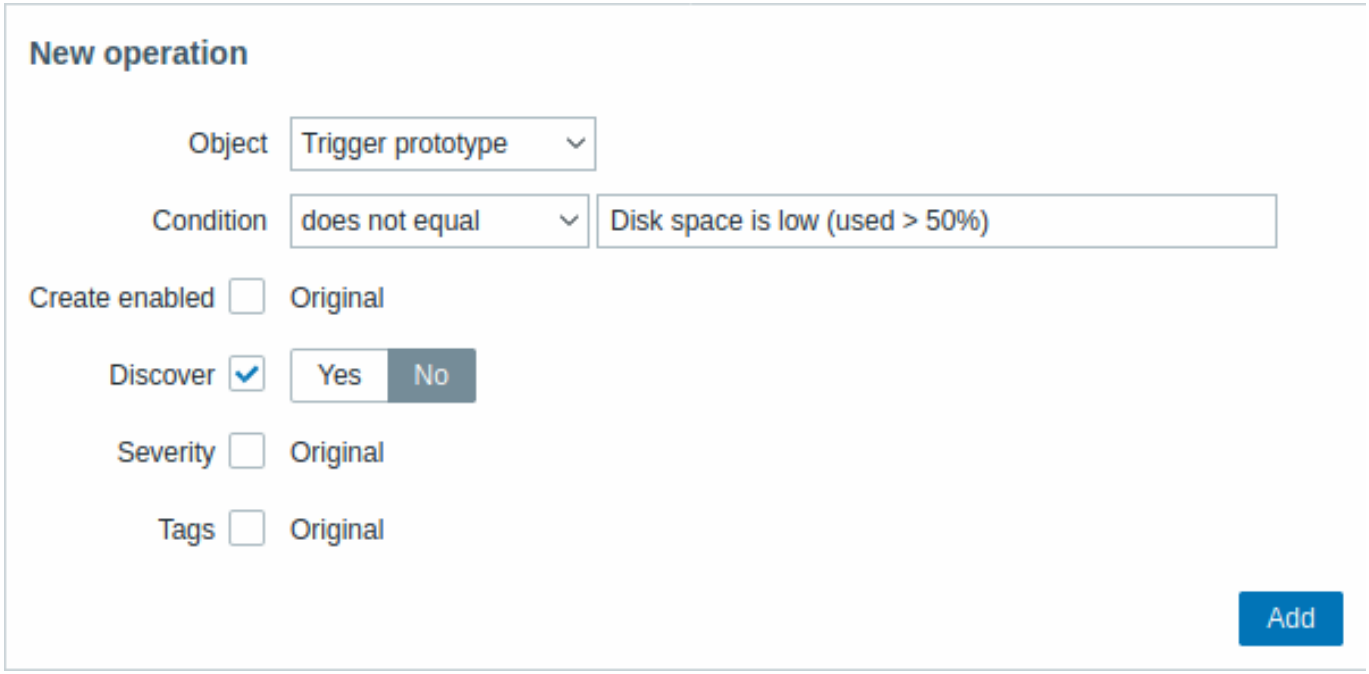


必須パラメータには、赤いアスタリスクが付いています。

Parameter	Description
Name	一意の（LLD ルールごとの）オーバーライド名。
If filter matches	フィルター条件に一致した場合に、次のオーバーライドの処理を続行するかどうかを定義します： **Continue overrides** - 後続のオーバーライドを処理します。 **Stop processing** - 先行するオーバーライド（存在する場合）およびこのオーバーライドの操作を実行し、一致した LLD 行については後続のオーバーライドを無視します。
Filters	どの検出済みエンティティにオーバーライドを適用するかを決定します。オーバーライドフィルターはディスカバリルールのフィルターの後に処理され、同じ機能を持ちます。
Operations	オーバーライド操作は、次の詳細とともに表示されます： **Condition** - オブジェクトタイプと、オブジェクト名に対して満たす必要がある条件。例：トリガープロトタイプが *Disk space is low (used > 50%)* と等しくない。 **Actions** - 操作を編集および削除するためのリンクが表示されます。

操作の設定

新しい操作の詳細を設定するには、Operations ブロック内の Add をクリックします。
既存の操作を編集するには、操作の横にある Edit をクリックします。
操作の詳細を編集できるポップアップウィンドウが開きます。



Parameter	Description
Object	使用可能なオブジェクトタイプは 5 種類あります： アイテムプロトタイプ トリガープロトタイプ グラフプロトタイプ ホストプロトタイプ ディスカバリプロトタイプ
Condition	この操作を適用するエンティティをフィルタリングできます。
Operator	サポートされる演算子： **equals** - このプロトタイプに適用 **does not equal** - これ以外のすべてのプロトタイプに適用 **contains** - プロトタイプ名にこの文字列が含まれる場合に適用 **does not contain** - プロトタイプ名にこの文字列が含まれない場合に適用 **matches** - プロトタイプ名が正規表現に一致する場合に適用 **does not match** - プロトタイプ名が正規表現に一致しない場合に適用
Pattern	検索対象の文字列または正規表現。
Object: *Item pro-to-type*	
Create enabled	チェックボックスをオンにするとボタンが表示され、元のアイテムプロトタイプ設定を上書きできます： *Yes* - アイテムは有効な状態で追加されます。 *No* - アイテムは検出済みエンティティに追加されますが、無効な状態になります。
Discover	チェックボックスをオンにするとボタンが表示され、元のアイテムプロトタイプ設定を上書きできます： *Yes* - アイテムが追加されます。 *No* - アイテムは追加されません。
Update interval	チェックボックスをオンにすると 2 つのオプションが表示され、アイテムに別の間隔を設定できます： *Delay* - アイテムの更新間隔。ユーザーマクロおよび時間サフィックス（例：30s、1m、2h、1d）をサポートします。*Custom interval* を使用する場合は 0 に設定する必要があります。 *Custom interval* - Add をクリックして、柔軟な間隔/スケジュール間隔を指定します。詳細はカスタム間隔を参照してください。

Parameter		Description
	History	チェックボックスをオンにするとボタンが表示され、アイテムに別の履歴保存期間を設定できます： *Do not store* - 選択すると、履歴は保存されません。 *Store up to* - 選択すると、保存期間を指定する入力フィールドが右側に表示されます。ユーザーマクロおよびLLD マクロをサポートします。
	Trends	チェックボックスをオンにするとボタンが表示され、アイテムに別のトレンド保存期間を設定できます： *Do not store* - 選択すると、トレンドは保存されません。 *Store up to* - 選択すると、保存期間を指定する入力フィールドが右側に表示されます。ユーザーマクロおよびLLD マクロをサポートします。
	Tags	チェックボックスをオンにすると新しいブロックが表示され、タグと値のペアを指定できます。 これらのタグは、タグ名が一致する場合でも、アイテムプロトタイプで指定されたタグに追加されます。
Object: *Trigger prototype*		
	Create enabled	チェックボックスをオンにするとボタンが表示され、元のトリガープロトタイプ設定を上書きできます： *Yes* - トリガーは有効な状態で追加されます。 *No* - トリガーは検出済みエンティティに追加されますが、無効な状態になります。
	Discover	チェックボックスをオンにするとボタンが表示され、元のトリガープロトタイプ設定を上書きできます： *Yes* - トリガーが追加されます。 *No* - トリガーは追加されません。
	Severity	チェックボックスをオンにするとトリガー深刻度ボタンが表示され、トリガー深刻度を変更できます。
	Tags	チェックボックスをオンにすると新しいブロックが表示され、タグと値のペアを指定できます。 これらのタグは、タグ名が一致する場合でも、トリガープロトタイプで指定されたタグに追加されます。
Object: *Graph prototype*		
	Discover	チェックボックスをオンにするとボタンが表示され、元のグラフプロトタイプ設定を上書きできます： *Yes* - グラフが追加されます。 *No* - グラフは追加されません。
Object: *Host prototype*		
	Create enabled	チェックボックスをオンにするとボタンが表示され、元のホストプロトタイプ設定を上書きできます： *Yes* - ホストは有効な状態で作成されます。 *No* - ホストは無効な状態で作成されます。
	Discover	チェックボックスをオンにするとボタンが表示され、元のホストプロトタイプ設定を上書きできます： *Yes* - ホストがディスカバリされます。 *No* - ホストはディスカバリされません。
	Link templates	チェックボックスをオンにすると、テンプレートを指定する入力フィールドが表示されます。テンプレート名の入力を開始するか、フィールドの横にある *Select* をクリックして、ポップアップウィンドウの一覧からテンプレートを選択します。 このオーバーライドのテンプレートは、すでにホストプロトタイプにリンクされているすべてのテンプレートに追加されます。

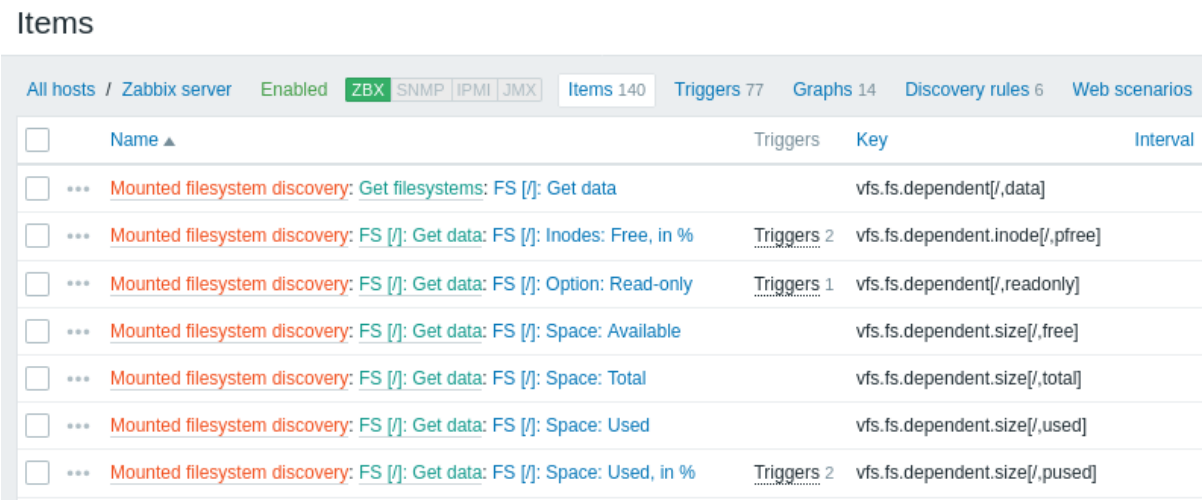
Parameter	Description
Tags	チェックボックスをオンにすると新しいブロックが表示され、タグと値のペアを指定できます。 これらのタグは、タグ名が一致する場合でも、ホストプロトタイプで指定されたタグに追加されます。
Host inventory	チェックボックスをオンにするとボタンが表示され、ホストプロトタイプに対して別のインベントリモードを選択できます： *Disabled* - ホストインベントリを設定しない *Manual* - 詳細を手動で入力 *Automated* - 収集したメトリクスに基づいてホストインベントリデータを自動入力

フォームボタン

フォームの下部にあるボタンでは、いくつかの操作を実行できます。

Add	ディスカバリルールを追加します。このボタンは、新しいディスカバリルールでのみ使用できます。
Update	ディスカバリルールのプロパティを更新します。このボタンは、既存のディスカバリルールでのみ使用できます。
Clone	現在のディスカバリルールのプロパティに基づいて、別のディスカバリルールを作成します。
Execute now	ディスカバリルールに基づいて、ただちにディスカバリを実行します。ディスカバリルールは、すでに存在している必要があります。詳細はこちらを参照してください。 ディスカバリをただちに実行する場合、設定キャッシュは更新されないため、ディスカバリルール設定に対するごく最近の変更は結果に反映されません。
Test	ディスカバリルールの設定をテストします。このボタンを使用すると、変更を恒久的に適用することなく、設定内容（接続性やパラメータの正確性など）を確認できます。
Delete	ディスカバリルールを削除します。
Cancel	ディスカバリルールのプロパティの編集をキャンセルします。

検出されたエンティティ　以下のスクリーンショットは、検出されたアイテム、トリガー、グラフがホストの設定でどのように表示されるかを示しています。
検出されたエンティティには、それらの元となったディスカバリルールへのオレンジ色のリンクが先頭に付きます。



同じ一意性条件を持つ既存のエンティティがすでに存在する場合、検出されたエンティティは作成されません。たとえば、同じキーを持つアイテムや同じ名前を持つグラフが該当します。
この場合、Web インターフェースには、ローレベルディスカバリルールが特定のエンティティを作成できなかったことを示すエラーメッセージが表示されます。

ただし、エンティティを作成できずスキップされたことを理由に、ディスカバリルール自体が未サポートになることはありません。ディスカバリルールは、他のエンティティの作成や更新を継続します。

検出されたエンティティ（ホスト、ファイルシステム、インターフェースなど）が検出されなくなった場合（またはフィルターを通過しなくなった場合）、それに基づいて作成されたエンティティは自動的に無効化され、最終的に削除されることがあります。

失われたリソースは、*Disable lost resources* パラメータの値に基づいて自動的に無効化される場合があります。
これは、失われたホスト、アイテム、トリガーに影響します。

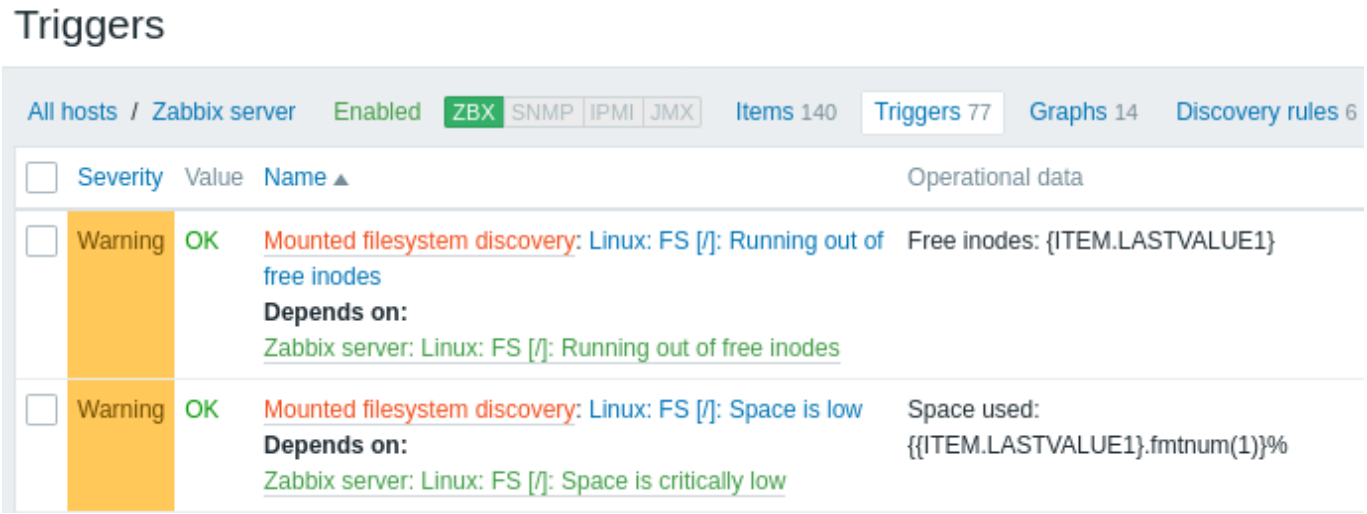
失われたリソースは、*Delete lost resources* パラメータの値に基づいて自動的に削除される場合があります。
これは、失われたホスト、ホストグループ、アイテム、トリガー、グラフに影響します。

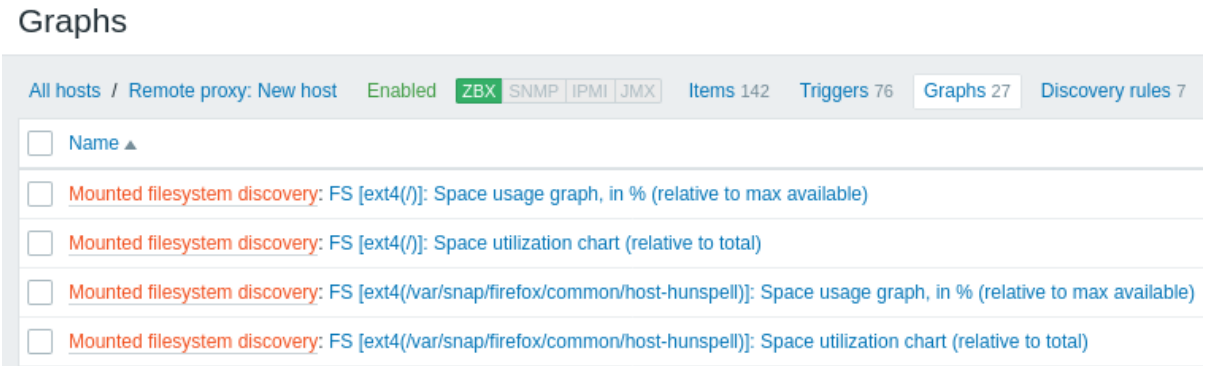
検出されたエンティティが「Not discovered anymore」になると、エンティティ一覧に有効期間インジケーターが表示されます。その上にマウスポインターを移動すると、状態の詳細を示すメッセージが表示されます。



エンティティが削除対象としてマークされていても、想定された時刻に削除されなかった場合（ディスカバリルールまたはアイテムのホストが無効な場合）、次回そのディスカバリルールが処理されたときに削除されます。

削除対象としてマークされた他のエンティティを含むエンティティは、ディスカバリルールレベルで変更されても更新されません。たとえば、LLD ベースのトリガーは、削除対象としてマークされたアイテムを含んでいる場合、更新されません。





その他のディスカバリの種類　その他の標準搭載のディスカバリの詳細および手順については、以下のセクションを参照してください。

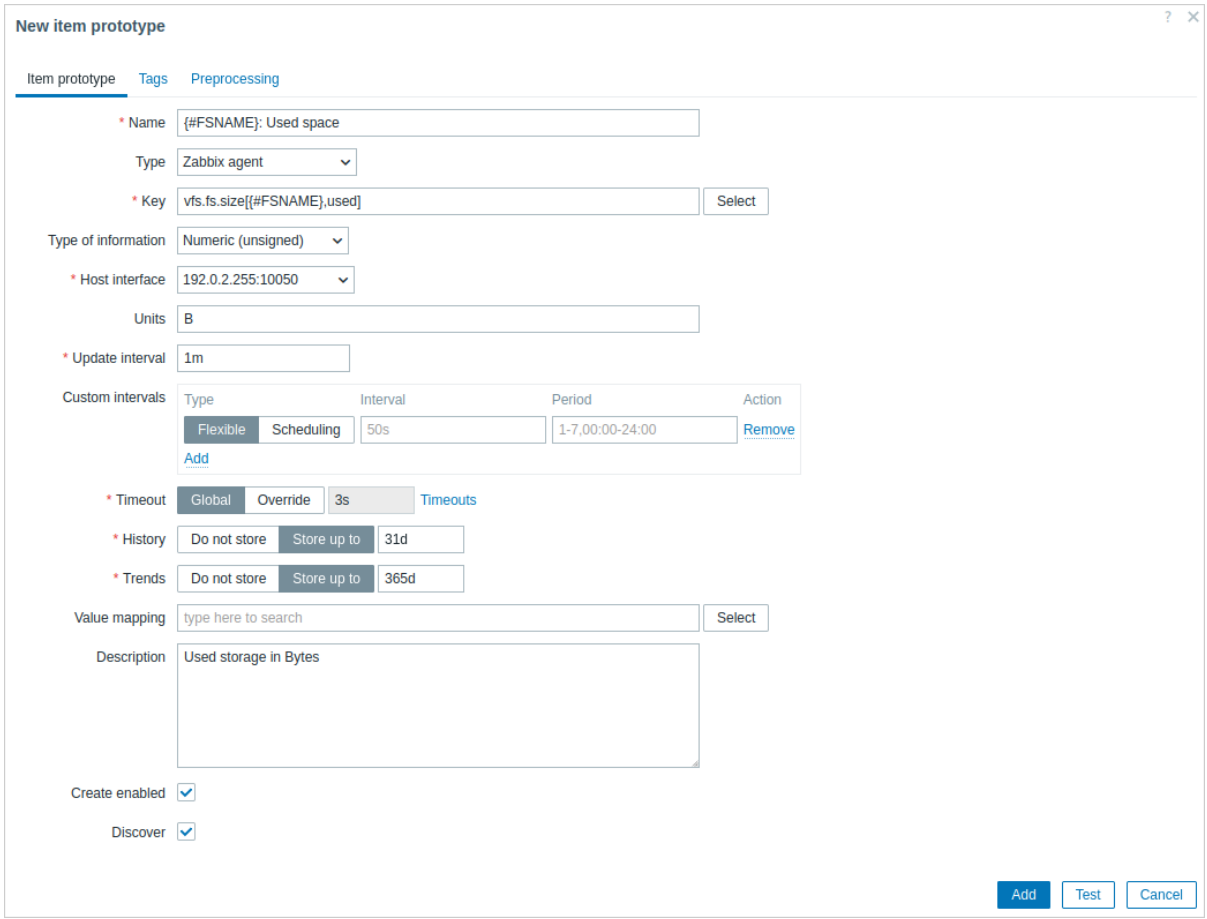
- ネットワークインターフェースのディスカバリ
- CPU および CPU コアのディスカバリ
- SNMP OIDのディスカバリ
- JMX オブジェクトのディスカバリ
- ODBC SQL クエリを使用したディスカバリ
- Windows サービスのディスカバリ
- Zabbix におけるホストインターフェースのディスカバリ

ディスカバリアイテムの JSON 形式の詳細、および独自のファイルシステムディスカバラを Perl スクリプトとして実装する方法の例については、カスタム LLD ルールの作成を参照してください。

1 アイテムのプロトタイプ

ルールを作成したら、そのルールのアイテムに移動し、「Create item prototype」を押してアイテムのプロトタイプを作成します。

ファイルシステム名が必要な箇所で、{#FSNAME} マクロがどのように使用されているかに注目してください。ディスカバリが正しく処理されるようにするため、アイテムキーでは低レベルディスカバリマクロの使用が必須です。ディスカバリルールが処理されると、このマクロは検出されたファイルシステムに置き換えられます。



Note:
item key パラメータが使用されていない場合は、LLD マクロをアイテムキーの角括弧 [...] 内に配置してください（例: v_[{#MACRO}]）。

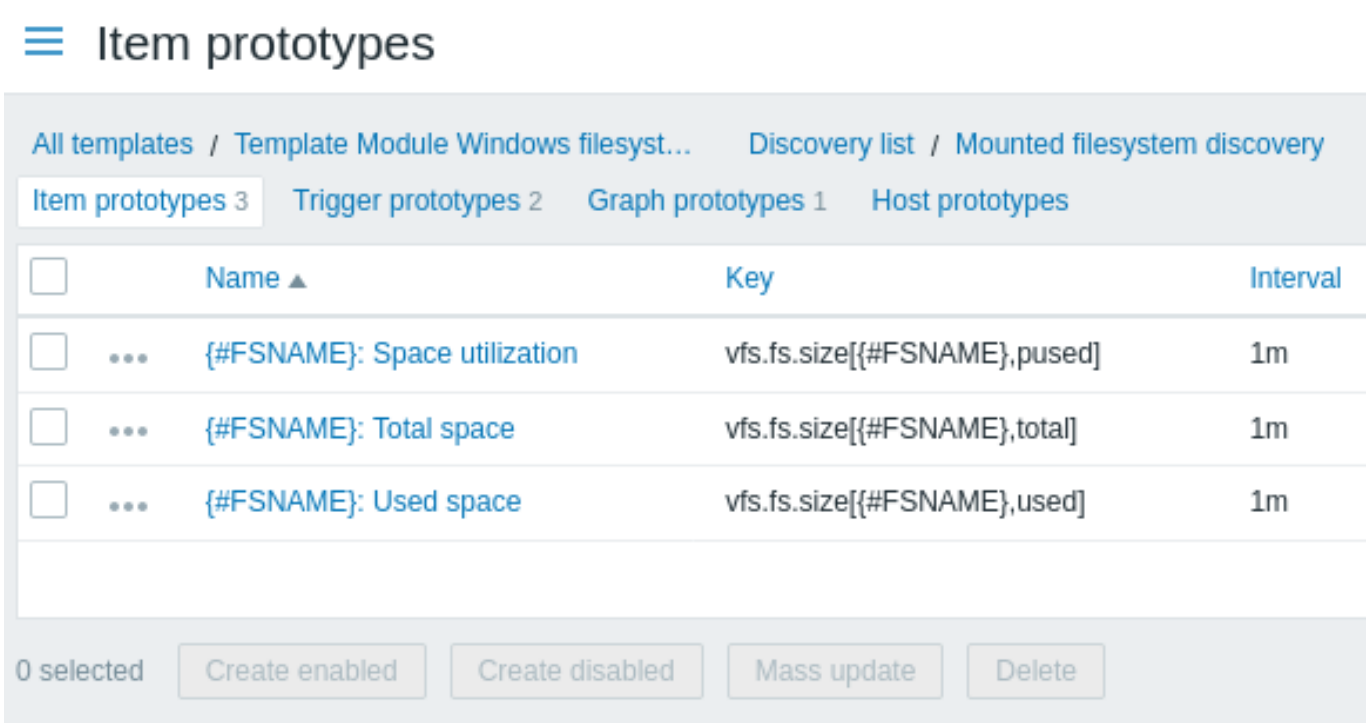
低レベルディスカバリマクロおよびユーザーマクロは、アイテムのプロトタイプ設定とアイテム値の前処理パラメータでサポートされています。更新間隔で使用する場合は、単一のマクロがフィールド全体を満たしている必要がある点に注意してください。1 つのフィールド内で複数のマクロを使用したり、マクロをテキストと混在させたりすることはサポートされていません。

Note:
正規表現および XPath 前処理パラメータで安全に使用できるよう、低レベルディスカバリマクロに対してコンテキスト固有のエスケープ処理が実行されます。

アイテムのプロトタイプ固有の属性:

Parameter	Description
Create enabled	チェックされている場合、アイテムは有効な状態で追加されます。 チェックされていない場合、アイテムは検出されたエンティティに追加されますが、無効な状態になります。
Discover	チェックされている場合（デフォルト）、アイテムは検出されたエンティティに追加されます。 チェックされていない場合、この設定がディスカバリルールで上書きされない限り、アイテムは検出されたエンティティに追加されません。

関心のある各ファイルシステムメトリクスについて、複数のアイテムのプロトタイプを作成できます。



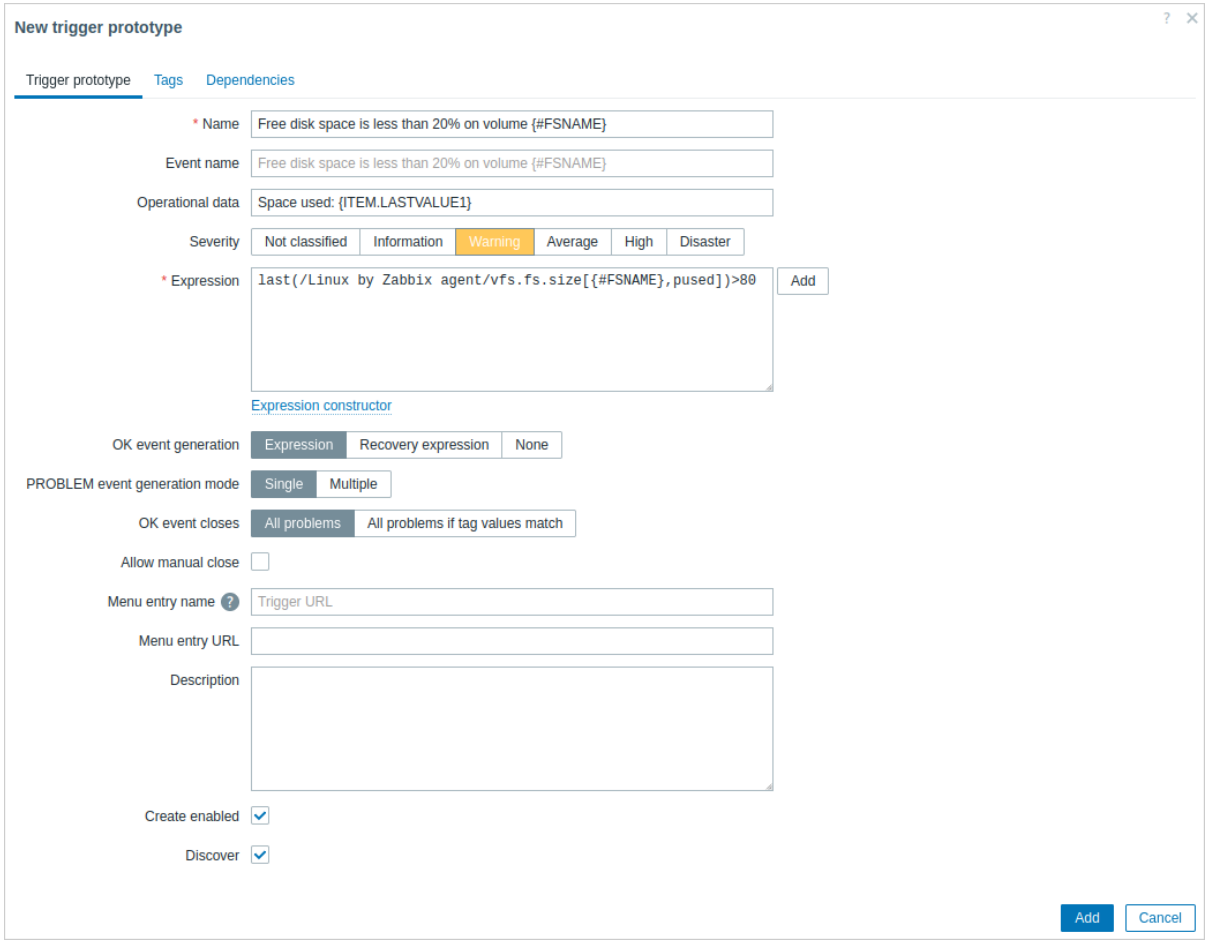
三点アイコンをクリックすると、特定のアイテムのプロトタイプに対するメニューが開き、次のオプションを利用できます。

- *Create trigger prototype* - このアイテムのプロトタイプに基づいてトリガーのプロトタイプを作成します
- *Trigger prototypes* - クリックすると、このアイテムのプロトタイプに対してすでに設定されているトリガーのプロトタイプへのリンク一覧が表示されます
- *Create dependent item* - このアイテムのプロトタイプ用の依存アイテムを作成します

複数のアイテムのプロトタイプのプロパティを一度に更新したい場合は、*Mass update* オプションを利用できます。

2 トリガープロトタイプ

トリガープロトタイプは、アイテムプロトタイプと同様の方法で作成します。

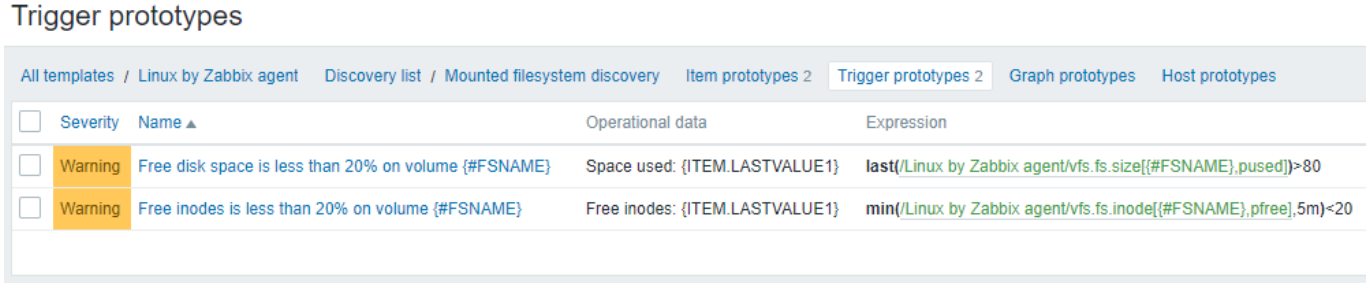


トリガープロトタイプ固有の属性:

Parameter	Description
Create enabled	チェックすると、トリガーは有効な状態で追加されます。 チェックしない場合、トリガーはディスカバリされたエンティティに追加されますが、無効な状態になります。
Discover	チェックすると（デフォルト）、トリガーはディスカバリされたエンティティに追加されます。 チェックしない場合、この設定がディスカバリルールで上書きされない限り、トリガーはディスカバリされたエンティティに追加されません。

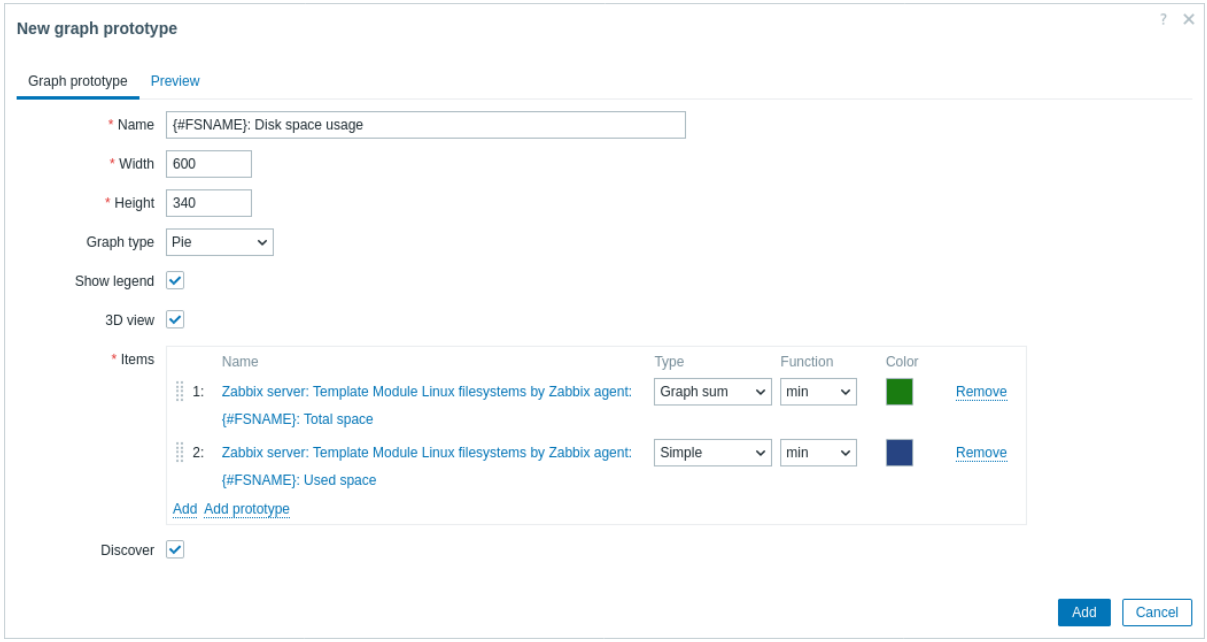
実際のトリガーがプロトタイプから作成される際、式内で比較に使用する定数（この例では「20」）について、柔軟に設定できる必要がある場合があります。そのような柔軟性を実現するには、コンテキスト付きユーザーマクロが役立ちます。

トリガープロトタイプ間には、依存関係を定義できます。これを行うには、*Dependencies* タブを開きます。トリガープロトタイプは、同じローレベルディスカバリ（LLD）ルール内の別のトリガープロトタイプ、または通常のトリガーに依存できます。トリガープロトタイプは、別の LLD ルールのトリガープロトタイプや、トリガープロトタイプから作成されたトリガーには依存できません。ホストのトリガープロトタイプは、テンプレートのトリガーに依存できません。



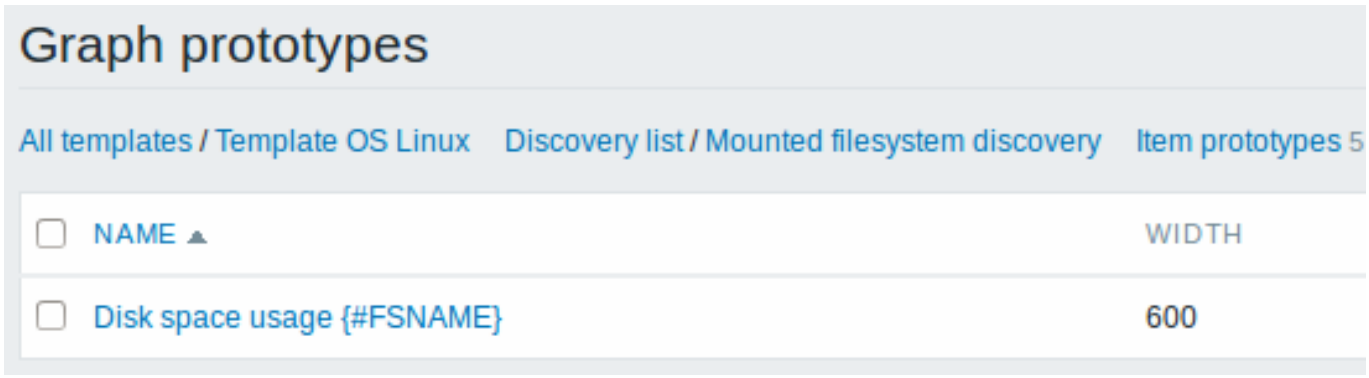
3 グラフプロトタイプ

グラフプロトタイプも作成できます。

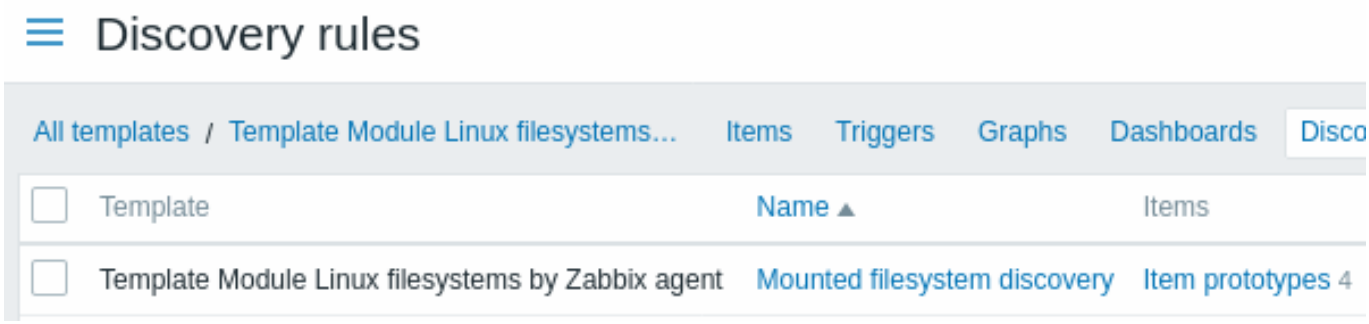


グラフプロトタイプ固有の属性:

Parameter	Description
Discover	チェックされている場合（デフォルト）、グラフはディスカバリされたエンティティに追加されます。 チェックされていない場合、この設定がディスカバリルールで上書きされない限り、グラフはディスカバリされたエンティティに追加されません。



最後に、以下のようなディスカバリルールを作成しました。これには 5 つのアイテムプロトタイプ、2 つのトリガープロトタイプ、および 1 つのグラフプロトタイプがあります。



4 ホストプロトタイプ

ホストプロトタイプは、ローレベルディスカバリルールを通じてホストを作成するための設計図です。
ホストとしてディスカバリされる前は、これらのプロトタイプには、テンプレートからリンクされたものを除き、アイテムおよびトリガーを設定できません。

設定

ホストプロトタイプは、ローレベルディスカバリルールで設定します。

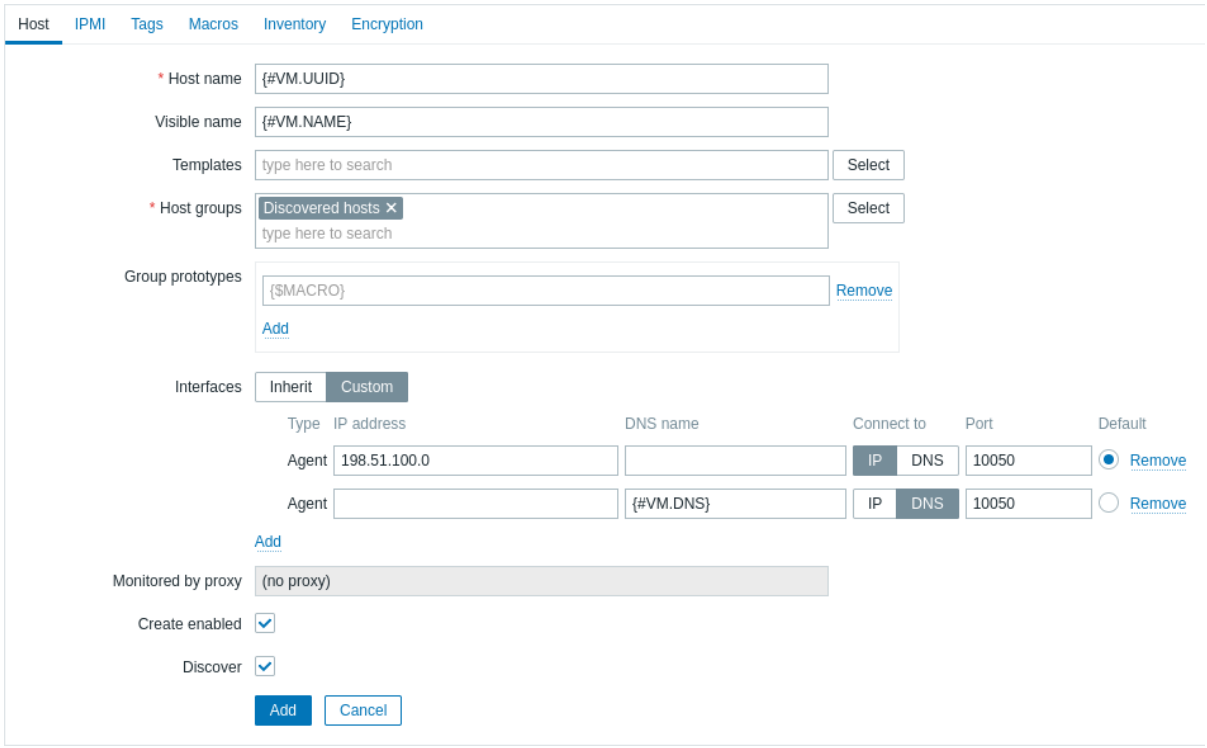
ホストプロトタイプを作成するには、次の手順を実行します。

1. データ収集 → ホストに移動します。

2. 対象のホストの ディスカバリをクリックし、そのホストに設定されているローレベルディスカバリルールの一覧を開きます。

3. 対象のディスカバリルールの ホストプロトタイプをクリックします。

4. 右上隅にある ホストプロトタイプの作成ボタンをクリックします。



ホストプロトタイプには通常のホストと同じパラメータがありますが、以下のパラメータでは異なる、または追加の設定が可能です。

パラメータ	説明
ホスト名	作成されるホスト名の一意性を確保するため、このパラメータには少なくとも 1 つのローレベルディスカバリマクロを含める必要があります。 ローレベルディスカバリマクロを使用できます。
表示名	
グループプロトタイプ	ローレベルディスカバリマクロを使用してホストグループプロトタイプを指定できます。 指定されたグループプロトタイプに基づいて、ホストグループがディスカバリされ、作成され、作成されたホストにリンクされます。また、他のローレベルディスカバリルールによってすでに作成されているディスカバリ済みグループも、作成されたホストにリンクされます。ただし、手動で作成されたホストグループと一致するディスカバリ済みホストグループは、作成されたホストにリンクされません。
インターフェース	ディスカバリされたホストが、そのディスカバリルールが属するホストから IP を継承するか（デフォルト）、またはカスタムインターフェースを取得するかを設定します。 ローレベルディスカバリマクロおよびユーザーマクロを使用できます。
有効にして作成	ディスカバリされたホストのステータスを設定します。チェックを外すと、ホストは無効な状態で作成されます。
ディスカバリ	ホストプロトタイプからホストを作成するかどうかを設定します。チェックを外すと、ホストプロトタイプからホストは作成されません（ただし、ローレベルディスカバリルールでこの設定が上書きされている場合を除きます）。

Note:
ローレベルディスカバリマクロは、タグ値およびホストプロトタイプのユーザーマクロ値でも使用できます。
 ホストプロトタイプでは、値のマッピングはサポートされていません。

ホストプロトタイプの設定例については、仮想マシンの監視を参照してください。

ホストインターフェース

カスタムインターフェースを追加するには、*Interfaces* セレクターを「Inherit」から「Custom」に切り替えます。Add をクリックし、インターフェースタイプ（Zabbix エージェント、SNMP、JMX、IPMI）を選択します。

Note:
Custom が選択されていてもインターフェースが設定されていない場合、ホストはインターフェースなしで作成されます。

 Inherit が選択され、かつホストプロトタイプがテンプレートに属している場合、検出されたすべてのホストは、そのテンプレートがリンクされているホストからホストインターフェースを継承します。

複数のカスタムインターフェースが指定されている場合、プライマリインターフェースを *Default* 列で設定できます。

カスタムホストインターフェースの設定例については、*VMware monitoring setup example* を参照してください。

Warning:
ホストインターフェースに正しいデータが含まれている場合にのみ、ホストが作成されます。

検出されたホスト

ホスト一覧では、検出されたホストには、それを作成したディスカバリルール名が接頭辞として付けられます。

検出されたホストは、ホストプロトタイプからほとんどのパラメータを読み取り専用として継承します。
設定できる検出されたホストのパラメータは、次のものだけです。

- テンプレート - 新しいテンプレートをリンクするか、手動で追加したもののリンクを解除します。ホストプロトタイプから継承されたテンプレートはリンク解除できません。
- 説明 - ホストの説明を追加または編集します。
- ステータス - ホストを有効化または無効化します。
- タグ - 新しいタグを追加するか、手動で追加したものを削除します。ホストプロトタイプから継承されたタグは削除できません。なお、タグは重複できません（同じ名前と値を持つタグ）。ホストプロトタイプに新しいタグが追加され、それが検出されたホスト上で手動追加されたタグと一致する場合、ディスカバリ時に手動追加されたタグは継承タグに置き換えられます。
- マクロ - 新しいホストマクロを追加するか、手動で追加したもの、およびホストプロトタイプから継承されたものを削除します。また、マクロ値やタイプを変更できます。

Note:
検出されたホストは、ディスカバリルールが設定されているホストからもuser macroを継承します。
これらの継承されたマクロは削除でき、値とタイプも変更できます。

検出されたホストは、さらに別のホストを検出することもできます。
たとえば、ホストプロトタイプ（ハイパーバイザーの検出に使用）に、それ自体のホストプロトタイプを持つローレベルディスカバリルール（仮想マシンの検出に使用）を含むテンプレートがリンクされている場合、Zabbix はハイパーバイザーとその仮想マシンの両方を検出します。
それらの VM 上のコンテナも追加で検出するには、ローレベルディスカバリルールを作成するか、ホストプロトタイプを含む新しいテンプレートを検出された VM にリンクできます。あるいは、ホストプロトタイプ自体を含むテンプレートをリンクして、VM ホストプロトタイプを事前設定することもできます。

検出されたホストは手動で削除できます。ただし、それらに対してディスカバリが有効になっている場合は、再び検出されます。

検出されなくなったホストは、次のように処理される場合があります。

- 自動的に無効化される（ディスカバリルールの失われたリソースを無効化の値に基づく）
- 自動的に削除される（ディスカバリルールの失われたリソースを削除の値に基づく）

5 ディスカバリのプロトタイプ

概要

ディスカバリプロトタイプは、「親」ディスカバリルール内にネストされたローレベルディスカバリルールであり、独自のアイテム、トリガーなどを持つオブジェクトの多段階ディスカバリを作成できます。たとえば、データベースサーバー上のすべてのデータベースインスタンスをディスカバリし、次に各インスタンスの表領域をディスカバリし、その後各表領域のテーブルをディスカバリしたい場合があります。

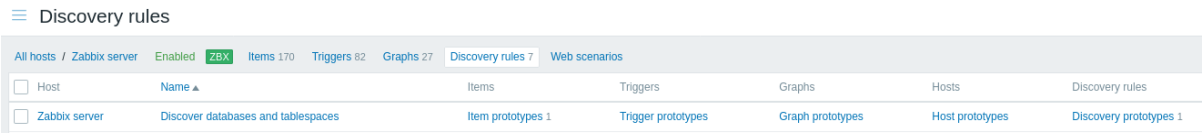
ディスカバリプロトタイプには、独自のアイテム、トリガー、グラフ、ホスト、およびディスカバリプロトタイプがあります。*Nested* タイプを指定すると、ネストされたディスカバリプロトタイプは親ルールと同じ JSON 値を使用します。

ディスカバリプロトタイプのネストレベルに制限はありません。

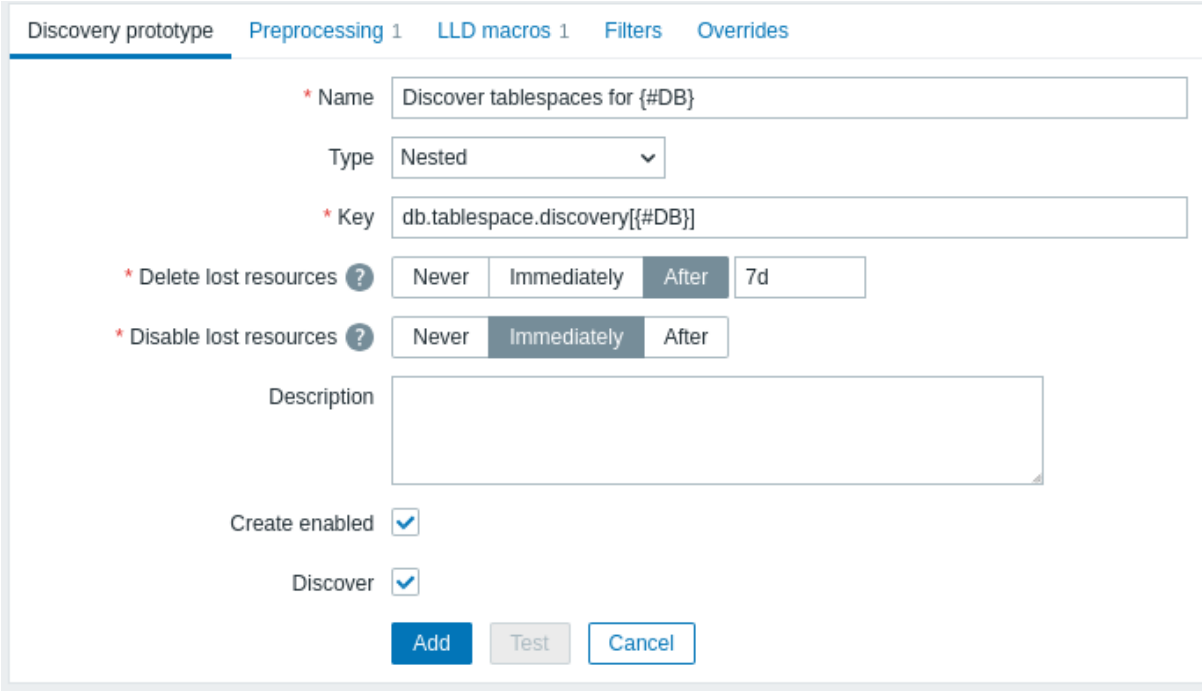
設定

ディスカバリプロトタイプを作成するには:

- 既存のディスカバリルールの行で、*Discovery prototypes* をクリックします



- *Create discovery prototype* をクリックします



このフォームの設定フィールドは、通常のローレベルディスカバリと共通です。

開いたディスカバリプロトタイプフォームで、*Type* として「Nested」を選択すると、親ディスカバリルールと同じ JSON 値内の JSON オブジェクトに基づいて、ディスカバリルール（ディスカバリプロトタイプから）が生成されます。たとえば、元の JSON が [<object A>, <object B>] で、ネストされたディスカバリルールプロトタイプが 1 つある場合、それぞれオブジェクト A とオブジェクト B のデータに基づいて 2 つのディスカバリルールが生成されます。

この場合、ディスカバリプロトタイプは親ルールと同時に有効化されます。そのため、ネストされたルールでは前処理を使用して、親によってすでに取得された同じデータの別の「スライス」を処理できます。

親 LLD ルールの LLD マクロは、ネストされたディスカバリルールでも利用できます。

検出されたホスト上のネストされた LLD ルール

Nested 低レベルディスカバリルールは、ホストプロトタイプに割り当てられたホストテンプレートで使用できます。検出されたホスト上に *Nested* ディスカバリルールが存在する場合、そのホストの検出に使用された JSON オブジェクトは、このホスト上のネスト型のすべての LLD ルールにも送信されます。詳細については、example を参照してください。

ホストを作成したディスカバリルールの LLD マクロは、ネストされたディスカバリルールで利用できます。

例

次のマルチレベル JSON の例を受信することを基に、ディスカバリプロトタイプの考えられる適用例を示します。

```
[
  {
    "database": "db1",
    "created_at": "2024-02-01T12:30:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "10GB" },
      { "name": "ts2", "max_size": "20GB" },
      { "name": "ts3", "max_size": "15GB" }
    ]
  },
  {
```

```
    "database": "db2",
    "created_at": "2023-11-15T08:45:00Z",
    "encoding": "UTF16",
    "tablespaces": [
      { "name": "ts1", "max_size": "5GB" },
      { "name": "ts2", "max_size": "25GB" },
      { "name": "ts3", "max_size": "30GB" }
    ]
  },
  {
    "database": "db3",
    "created_at": "2024-01-05T15:10:00Z",
    "encoding": "UTF8",
    "tablespaces": [
      { "name": "ts1", "max_size": "12GB" },
      { "name": "ts2", "max_size": "18GB" },
      { "name": "ts3", "max_size": "22GB" }
    ]
  }
]
```

ケース 1

データベースサーバー上のデータベースインスタンスをディスカバリし、その後、各インスタンスの表領域をディスカバリします。

1. データベースサーバーのディスカバリに関連するホストが少なくとも 1 つ必要です。
2. このホストに対して、*Discover databases and tablespaces* という名前のLLD ルールを作成します。
3. このルールの *LLD* マクロタブに切り替え、マクロ {#DB}=$.database を追加します。
4. このルールに対して、*Active connections to {#DB}* という名前のアイテムのプロトタイプを追加します（タイプ: エージェント、キー: db.connections[{#DB}]）。
5. 各データベースに関連するアイテムがディスカバリされます。

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. このルールに対して、*Discover tablespaces for {#DB}* という名前のディスカバリのプロトタイプを作成します（タイプ: ネスト、キー: db.tablespace.discovery[{#DB}]）。
7. このディスカバリのプロトタイプの 前処理タブに切り替え、ステップ JSONPath=$.tablespaces を追加します。
8. このディスカバリのプロトタイプの *LLD* マクロタブに切り替え、マクロ {#TSNAME}=$.name を追加します。
9. このディスカバリのプロトタイプに対して、*Size of tablespace {#TSNAME} for {#DB}* という名前のアイテムのプロトタイプを作成します（タイプ: エージェント、キー: db.ts.size[{#DB}, {#TSNAME}]）。
10. 各データベースの各表領域に関連するアイテムがディスカバリされます。

```
Size of tablespace ts1 for db1, Size of tablespace ts2 for db1, Size of tablespace ts3 for db1,
Size of tablespace ts1 for db2, Size of tablespace ts2 for db2, Size of tablespace ts3 for db2,
Size of tablespace ts1 for db3, Size of tablespace ts2 for db3, Size of tablespace ts3 for db3.
```

キーは db.ts.size[db1,ts1]、db.ts.size[db1,ts2]、... db.ts.size[db3,ts3] です。

ケース 2

データベースサーバー上のデータベースインスタンスをディスカバリされたホストとして表現して検出し、その後、各インスタンスの表領域を検出します。

1. データベースサーバーのディスカバリに関連するホスト（ルートホスト）が少なくとも 1 つあることを確認します。
2. 各データベースの表領域を検出するためのテンプレートを作成します。
3. このテンプレートに、*{#DB}* へのアクティブ接続数という名前のアイテムを作成します（タイプ: エージェント、キー: db.connections[{#DB}]）。
4. このテンプレートに、表領域の検出という名前のLLD ルール を作成します（タイプ: ネスト）。
5. このルールの *Preprocessing* タブに切り替え、ステップ JSONPath=$.tablespaces を追加します。
6. このルールの *LLD Macros* タブに切り替え、マクロ {#TSNAME}=$.name を追加します。

7. このルールに、*{#DB}* の表領域 *{#TSNAME}* のサイズという名前のアイテムプロトタイプを作成します（タイプ: エージェント、キー: `db.ts.size[{#DB}, {#TSNAME}]`）。
8. ルートホストに戻り、このホストに データベースと表領域の検出という名前のLLD ルール を作成します。
9. このルールの *LLD Macros* タブに切り替え、マクロ `{#DB}=$.database` を追加します。
10. このルールに、データベース *{#DB}* 用ホストという名前のホストプロトタイプを追加します。
11. このホストプロトタイプの *Macros* タブに切り替え、マクロ `{$DB}={#DB}` を追加します（手順 3 のアイテム名とキーのため）。
12. 手順 2 で作成したテンプレートをこのホストプロトタイプにリンクします。
13. ディスカバリされたホストには、各データベースおよびその表領域に関連するディスカバリ済みアイテムが含まれます。

ホスト	アイテム
データベース *db1* 用ホスト	db1 へのアクティブ接続数
	db1 の表領域 ts1 のサイズ
	db1 の表領域 ts2 のサイズ
	db1 の表領域 ts3 のサイズ
データベース *db2* 用ホスト	db2 へのアクティブ接続数
	db2 の表領域 ts1 のサイズ
	db2 の表領域 ts2 のサイズ
	db2 の表領域 ts3 のサイズ
データベース *db3* 用ホスト	db3 へのアクティブ接続数
	db3 の表領域 ts1 のサイズ
	db3 の表領域 ts2 のサイズ
	db3 の表領域 ts3 のサイズ

6 ローレベルディスカバリに関する注意事項

ユーザーマクロコンテキストでの LLD マクロの使用

LLD マクロは、トリガープロトタイプなど、ユーザーマクロコンテキスト内で使用できます。

同じアイテムに対する複数の LLD ルール

同じディスカバリアイテムを使用して、複数の低レベルディスカバリルールを定義できます。

これを行うには、Alias エージェントparameter を定義する必要があります。これにより、異なるディスカバリルールで変更したディスカバリアイテムキーを使用できます。たとえば、`vfs.fs.discovery[foo]`、`vfs.fs.discovery[bar]` などです。

戻り値のデータ制限

低レベルディスカバリルールの JSON データが Zabbix server で直接受信される場合、制限はありません。これは、戻り値がデータベースに保存されることなく処理されるためです。

カスタム低レベルディスカバリルールについても制限はありません。ただし、カスタム低レベルディスカバリルールのデータを user parameter を使用して取得する場合は、user parameter の戻り値の制限が適用されます。

データが Zabbix proxy を経由する必要がある場合、このデータをデータベースに保存する必要があります。この場合は、データベースの制限が適用されます。

7 ディスカバリルール

さまざまなケースにおけるディスカバリルール設定例については、サイドバーを参照してください。

1 ファイルシステムマウントの検出

概要

マウントされているファイルシステムとそのプロパティをディスカバリすることができます。

- マウントポイント名
- ファイルシステムの種類
- ファイルシステムのサイズ
- inode 統計

- マウントオプション

これを行うには、以下を組み合わせて使用できます。

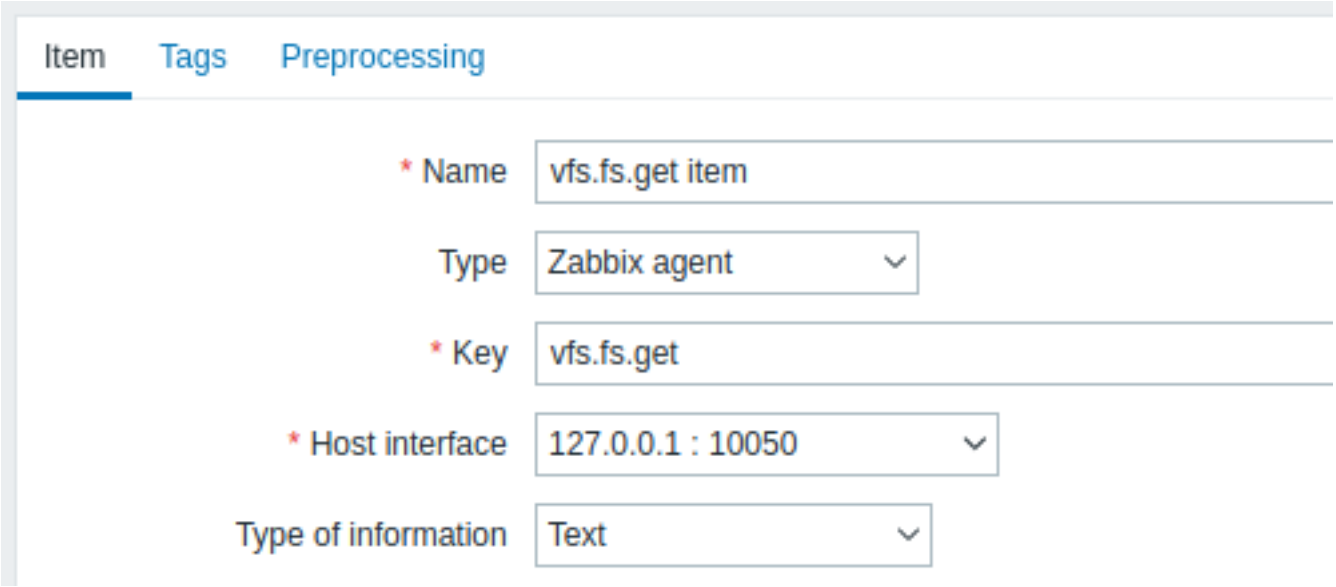
- マスターアイテムとしてvfs.fs.get エージェントアイテム
- 依存するローレベルディスカバリルールとアイテムのプロトタイプ

設定

マスターアイテム

以下のキーを使用して、Zabbix エージェントのアイテムを作成します。

vfs.fs.get



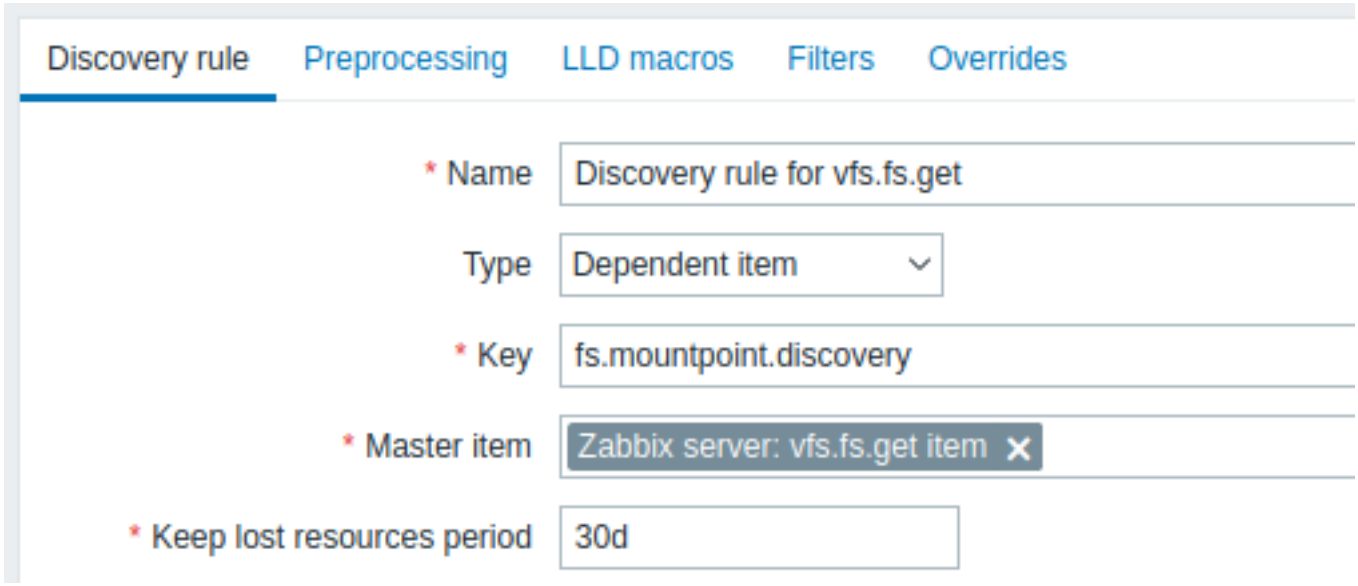
大きな JSON データになる可能性があるため、情報の型を「テキスト」に設定します。

このアイテムによって返されるデータには、マウントされたファイルシステムについて、次のような内容が含まれます。

```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
  }
]
```

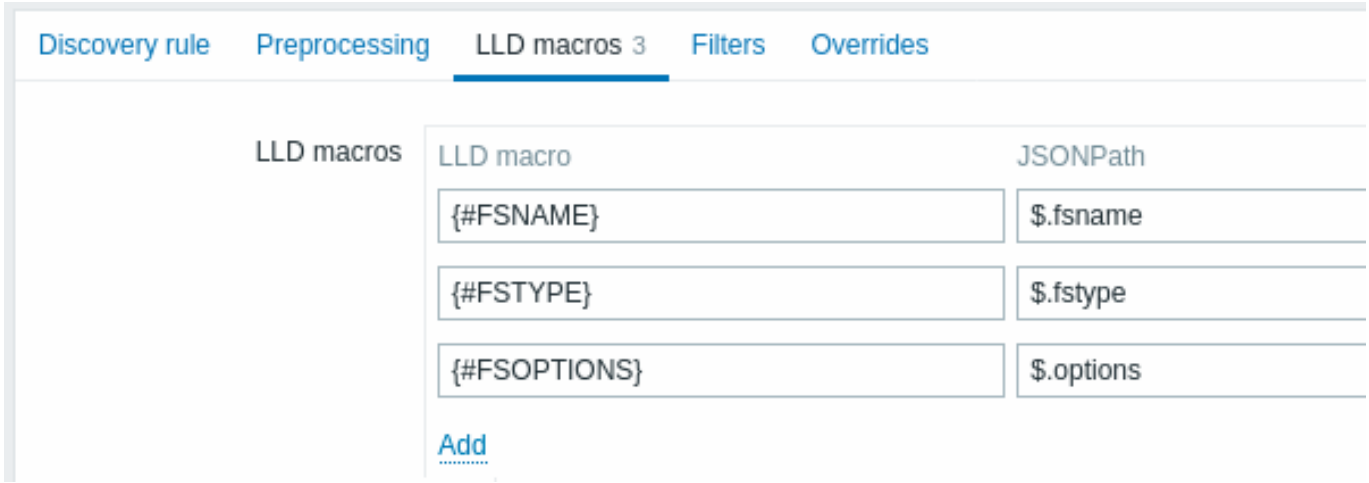
依存 LLD ルール

「依存アイテム」タイプとしてローレベルディスカバリルールを作成します。

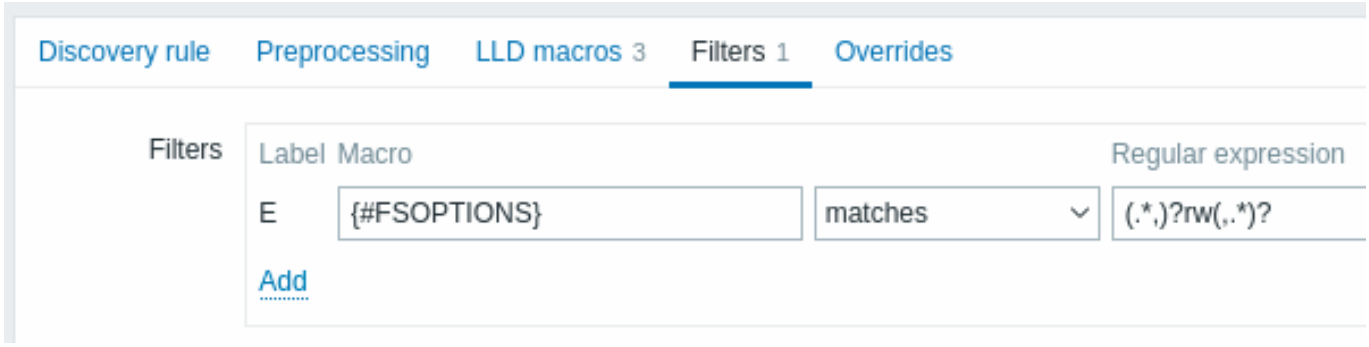


マスターアイテムとして、作成した `vfs.fs.get` アイテムを選択します。

「LLD マクロ」タブで、対応する JSONPath を使用してカスタムマクロを定義します。

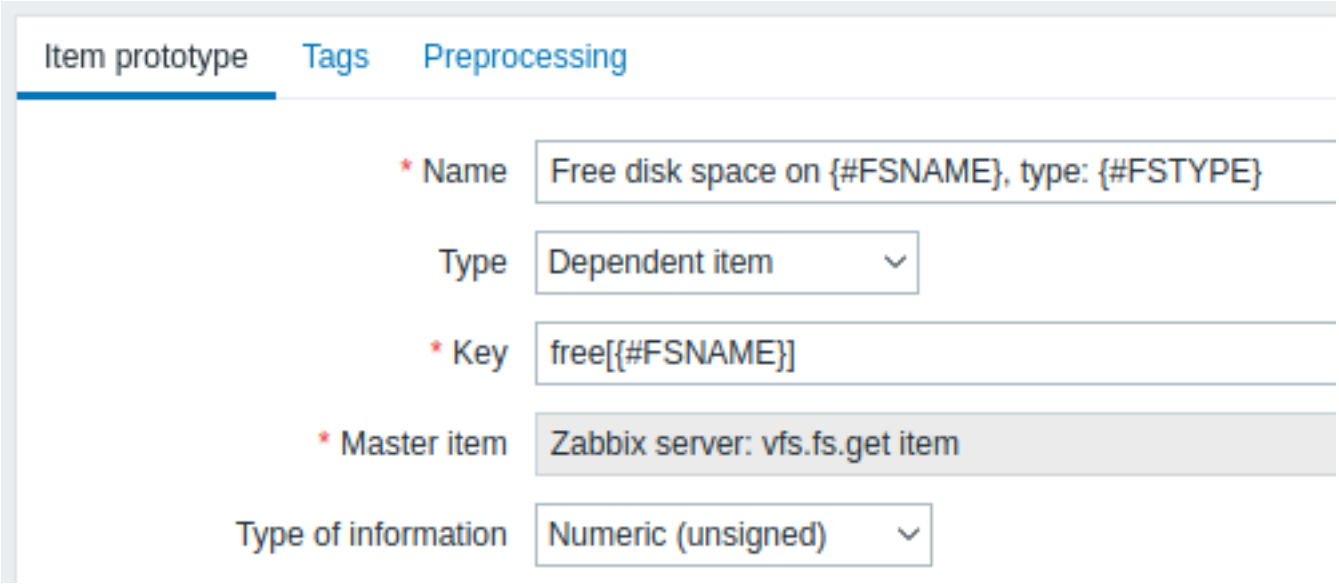


「フィルター」タブでは、読み書き可能なファイルシステムのみをフィルタリングする正規表現を追加できます。



依存アイテムのプロトタイプ

この LLD ルールで" 依存アイテム" タイプのアイテムプロトタイプを作成します。このプロトタイプのマスターアイテムとして、作成した `vfs.fs.get` アイテムを選択します。



アイテムのプロトタイプ名とキーでカスタムマクロが使用されていることに注意してください。

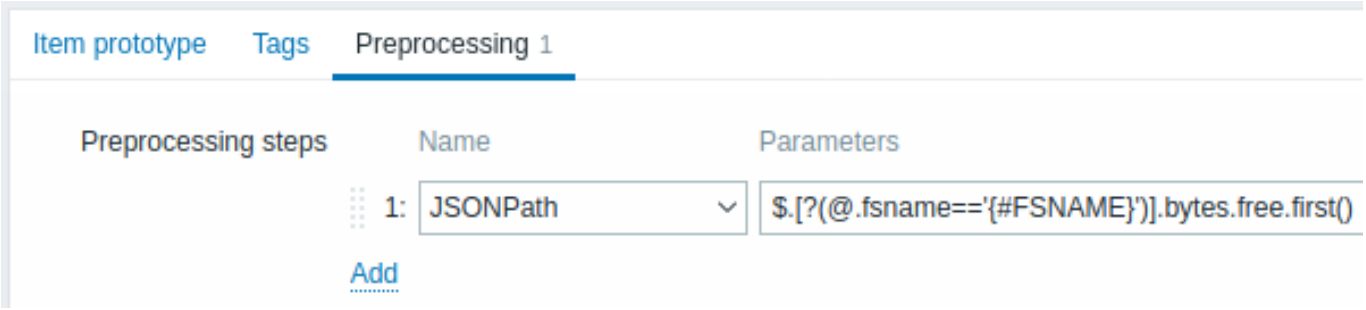
- *Name*: Free disk space on {#FSNAME}, type: {#FSTYPE}
- *Key*: Free[{#FSNAME}]

情報の種類として、次を使用します。

- 数値 *(整数)* 次のメトリックの場合'free', 'total', 'used'
- 数値 *(浮動小数)* 次のメトリックの場合'pfree', 'pused' (percentage)

アイテムのプロトタイプの" 保存前処理" タブで、JSONPath を選択し、次の JSONPath 式をパラメーターとして使用します。

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



検出されるとマウントポイントごとに 1 つのアイテムが作成されます。この項目は、指定されたマウントポイントの空きバイト数を返します。

2 ネットワークインターフェースの検出

ファイルシステムが検出されるのと同様の方法で、ネットワークインターフェイスも検出できます。

アイテムキー

ディスカバリルールで使用するアイテムキーは次のとおりです。

```
net.if.discovery
```

サポートされているマクロ

ディスカバリルールフィルターで {#IFNAME} マクロを使用したり、アイテム、トリガー、グラフのプロトタイプを使用したりできます。

"net.if.discovery" に基づいて作成する可能性のあるアイテムのプロトタイプの例：

- "net.if.in[{#IFNAME},bytes]",
- "net.if.out[{#IFNAME},bytes]".

Windows では {#IFGUID} も返されることに注意してください。

3 CPU と CPU コアの検出

ファイルシステムが検出されるのと同様の方法で、CPU と CPU コアも検出できます。

アイテムキー

ディスカバリールールで使用するアイテムキーは次のとおりです。

`system.cpu.discovery`

サポートされているマクロ

このディスカバリキーは、CPU の順序番号とステータスをそれぞれ識別する 2 つのマクロ - {#CPU.NUMBER} と {#CPU.STATUS} - を返します。プロセッサーは 0 から N - 1 まで列挙されます。ここで、N はプロセッサーの総数です。

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" is only on Windows */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

実際の物理プロセッサー、コア、ハイパースレッドを明確に区別することはできない点に注意してください。Linux、UNIX、BSD システム上の {#CPU.STATUS} はプロセッサーのステータスを返し、その値は”online” または”offline” のいずれかです。Windows システムでは、この同じマクロが 3 つ目の値 - ”unknown” - を表す場合があります。これは、プロセッサーは検出されているものの、まだその情報が収集されていないことを示します。

CPU ディスカバリは、collector によって提供されるデータとの一貫性を保ち、データ取得時のリソースを節約するために、エージェントの collector プロセスに依存しています。そのため、このアイテムキーはエージェントバイナリのテスト用コマンドラインフラグ (-t) では動作せず、NOT_SUPPORTED ステータスと、collector プロセスが開始されていないことを示すメッセージを返します。

CPU ディスカバリに基づいて作成できるアイテムのプロトタイプには、たとえば次のものがあります。

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

アイテムキーの詳細な説明については、Zabbix agent item keys を参照してください。

NUMA システムで検出されたパフォーマンスカウンターアイテムに関する注意事項

一部のWindows パフォーマンスカウンターは、NUMA システム上の一部の論理プロセッサーでは利用できない場合があります。

例えば、以下のキーを持つアイテムプロトタイプで検出されたアイテムは、最初の NUMA ノードでのみ動作する可能性があります。その他の NUMA ノードのアイテムは未サポート状態になる場合があります。

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

また、キー”system.cpu.discovery” を持つディスカバリルールは、次のようなパフォーマンスカウンターに置換するためのローレベルディスカバリマクロ値を提供しません。

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

一方、system.cpu.util アイテムは、プロセッサーグループ数およびプロセッサー数に応じて、CPU 使用率を監視するために異なるパフォーマンスカウンターに依存します。また、Windows のプロセッサーグループに関連するまれなバグもあります。

4 SNMP OIDs の検出

概要

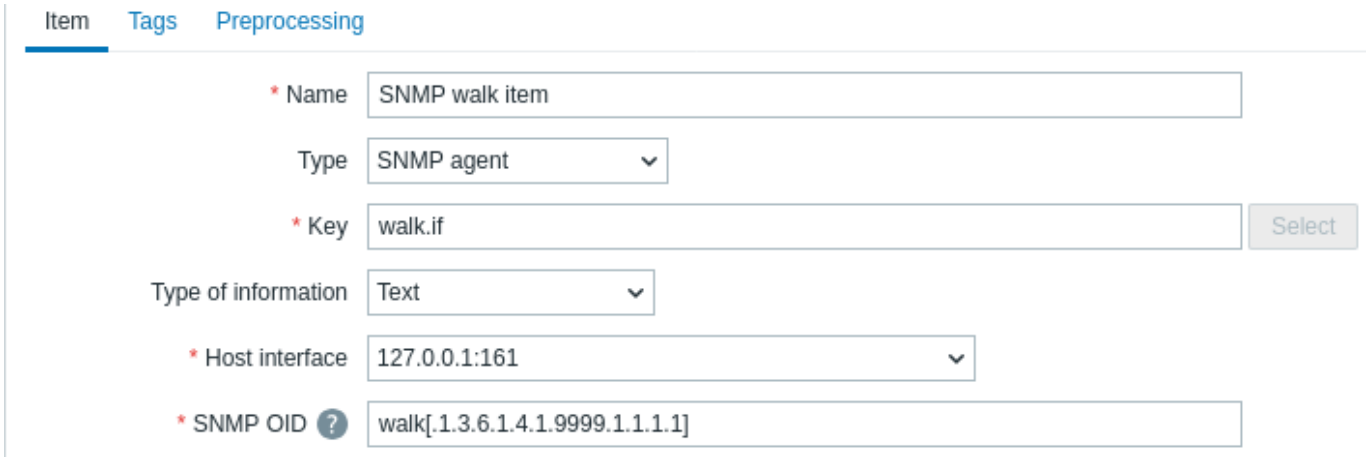
このセクションでは、SNMP デバイスに対してローレベルディスカバリを実行します。

この SNMP OID のディスカバリ方法は、Zabbix サーバー/プロキシ 6.4 以降でサポートされています。

設定例

1. 次のようなキーを持つ SNMP エージェントアイテムを作成します。

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



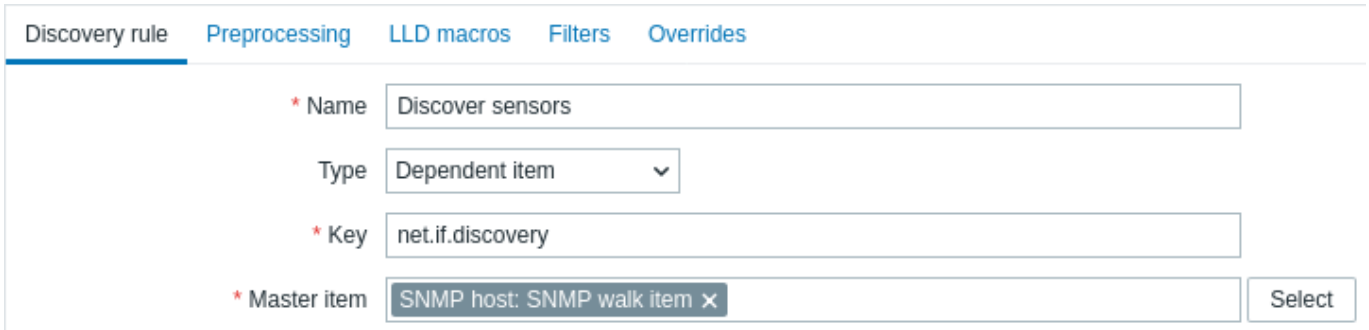
このアイテムは、SNMP テーブル全体に対して 1 回の walk を実行し、すべてのテーブルエントリを 1 回のリクエストで返します。返される形式は、`snmpwalk` ユーティリティの書式オプション `-Oe -Ot -On` を指定した出力に対応しています。

このアイテムは、次のような複数行のテキスト値を返します。

```
.1.3.6.1.4.1.9999.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.3.2 = 200
```

2. ディスカバリルールを作成します。

- *Name* フィールドに、わかりやすいディスカバリルール名を入力します（例: ”Discover sensors”）。
- *Type* フィールドで、”Dependent item” を選択します。
- *Key* フィールドに、説明的なキーを入力します（例: ”net.if.discovery”）。
- *Master item* フィールドで、”SNMP walk item” を選択します。



3. *Preprocessing* タブで、*Name* ドロップダウンから”SNMP walk to JSON” を選択した前処理ステップを追加し、3 つのパラメータを設定します。

- *Field name*: ”{#SENSORNAME}”; *OID prefix*: ”.1.3.6.1.4.1.9999.1.1.1.1.1”; *Format*: ”Unchanged”。
- *Field name*: ”{#SENSORTYPE}”; *OID prefix*: ”.1.3.6.1.4.1.9999.1.1.1.1.2”; *Format*: ”Unchanged”。
- *Field name*: ”{#SENSORVALUE}”; *OID prefix*: ”.1.3.6.1.4.1.9999.1.1.1.1.3”; *Format*: ”Unchanged”。

前処理後、ディスカバリルールはマクロセットの JSON 配列を返します。

例:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
```

```
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

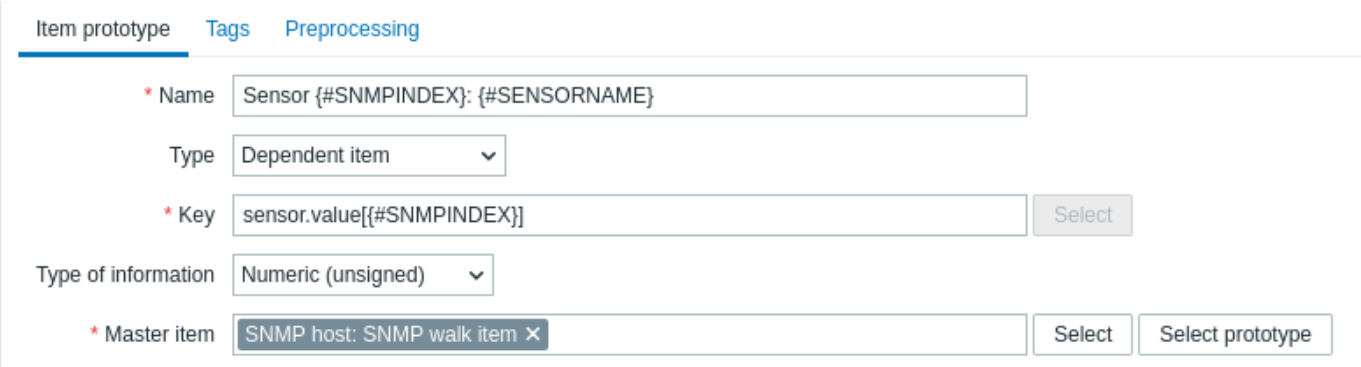
各オブジェクトは 1 つの検出されたセンサーを表し、{#SNMPINDEX}、{#SENSORNAME}、{#SENSORTYPE}、{#SENSORVALUE} などのマクロを提供します。

これらは SNMP インデックスごとにグループ化されます。SNMP インデックスは各 OID の末尾にある数値サフィックス（例: .1、.2）で、SNMP テーブル内の各行を一意に識別します。このインデックスは自動的に {#SNMPINDEX} として抽出されます。

4. ディスカバリルールの配下に、1 つ以上のアイテムプロトタイプを作成します（ディスカバリルールをマスターアイテムとして使用）。

例として、センサー値の依存アイテム:

- *Name* フィールドに、”Sensor {#SNMPINDEX}: {#SENSORNAME}” を入力します。
- *Type* フィールドで、”Dependent item” を選択します。
- *Key* フィールドに、”sensor.value[{#SNMPINDEX}]” を入力します。
- *Master item* フィールドで、”SNMP walk item” を選択します。



Preprocessing タブで、*Name* に”SNMP walk value” を指定した前処理ステップを追加し、*Parameter* フィールドに OID ”.1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}” を設定します。
Format: ”Unchanged”。

次のアイテムが検出されます。

Name	Key	値の抽出元 OID	アイテム値
Sensor 1: Temperature Sensor	sensor.value[1]	1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	1.3.6.1.4.1.9999.1.1.1.1.3.2	200

ディスカバリルールが実行されると、sensor.value[1]、sensor.value[2] のようなアイテムが作成されます。

各依存アイテムは、個別に SNMP リクエストを実行することなく、前処理を使用してマスターアイテムの SNMP walk 結果から値を抽出します。

5. ディスカバリルールと同じマクロを使用して、トリガープロトタイプ内で依存アイテムプロトタイプを参照します。
例:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

これにより、検出された各センサー（たとえば sensor.value[1]、sensor.value[2]）に対して 1 つのトリガーが作成され、最新の値（温度または湿度）が 75 を超えると発火します。

6. 検出された各エンティティに対して依存アイテムを含めます。
グラフアイテムキーの例:

```
sensor.value[{#SNMPINDEX}]
```

{#SNMPINDEX} ごとに 1 つのグラフが作成され、温度と湿度の推移が時系列でプロットされます。

この設定では、検出されたアイテム数に関係なく、ポーリングサイクルごとに実行される SNMP walk リクエストは 1 回だけです。すべての依存アイテムは、前処理を使用してマスター SNMP walk 結果から値を抽出するため、SNMP トラフィックと負荷を大幅に削減できます。

walk[] を使用した動的インデックス

動的インデックス（たとえば、インターフェースのインデックス）は、ハードウェアの再構成時に変化することがあります。この動作に対応するため、次のようなキーを持つマスター SNMP walk ディスカバリルールを作成します。

```
walk[1.3.6.1.2.1.2.2.1.10]
```

SNMP walk to JSON の前処理後、結果は次のようになります。

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

依存アイテムのプロトタイプでは、{#SNMPINDEX} マクロを使用してキーを構築します。

```
net.if.in[{#SNMPINDEX}]
```

このプロトタイプの前処理には、*Parameter* フィールドに OID 1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX} を指定した「SNMP walk value」という名前が含まれます。*Format*: 「Unchanged」。

実行時には、net.if.in[2] や net.if.in[3] のような実際のアイテムが作成されます。特定のインターフェースインデックスが変更された場合（たとえば、SNMP テーブル内でインデックス 2 が 5 に置き換えられた場合）、次回のディスカバリルール実行時に次のようになります。

- 古い依存アイテム net.if.in[2] は「lost」としてマークされるか削除され、そのアイテムでは新しいデータは収集されなくなります。
- 新しい依存アイテム net.if.in[5] が作成され、空の履歴から開始されます。
- net.if.in[2] の履歴データは、自動的に net.if.in[5] へ移動されません。

トリガーのプロトタイプの例:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

グラフのプロトタイプの例には、次のアイテムが含まれます。

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

この設定により、動的インデックスを持つテーブルを確実に監視しつつ、SNMP トラフィックを最小限に抑えることができます。ポーリングサイクルごとに必要な SNMP walk は 1 回だけで、必要な値は依存アイテムのプロトタイプによって抽出されます。

検出されたエンティティ

サーバーの実行時に、SNMP ディスカバリルールが返す値に基づいて、実際の依存アイテム、トリガー、およびグラフが作成されます。

5 SNMP OIDs の検出 (legacy)

概要

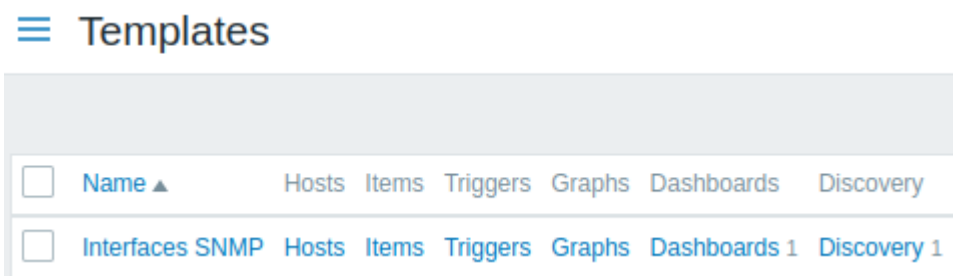
このセクションでは、スイッチで SNMP の検出を実行します。

アイテムキー

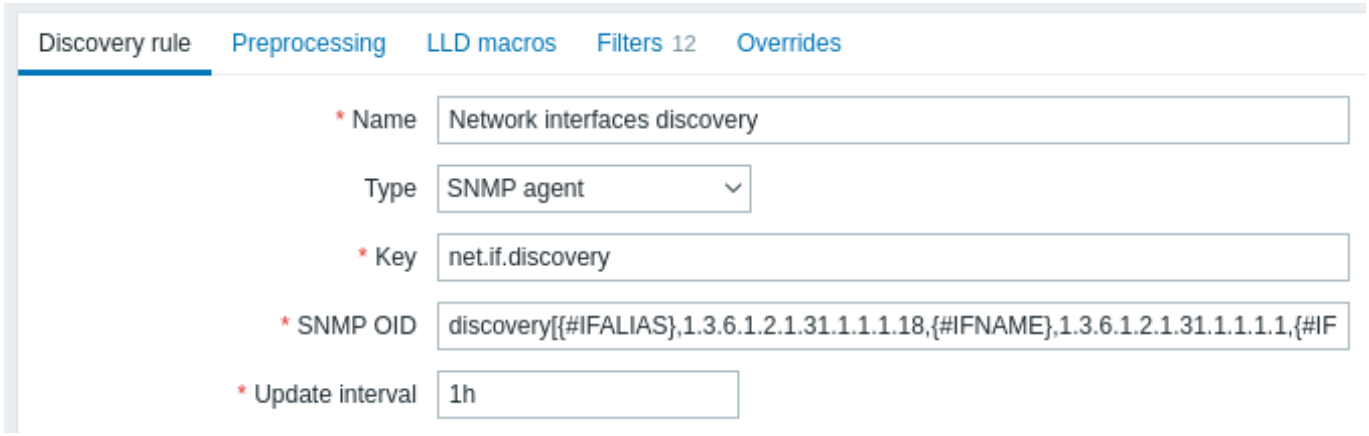
ファイルシステムやネットワークインターフェースのディスカバリとは異なり、このアイテムでは必ずしも "snmp.discovery" キーを使用する必要はありません。SNMP エージェントのアイテムタイプであれば十分です。

ディスカバリルールを設定するには、次の手順を実行します。

- *Data collection* → *Templates* に移動します
- 適切なテンプレートの行にある *Discovery* をクリックします



- 画面右上の *Create discovery rule* をクリックします
- 以下のスクリーンショットのように、必要な詳細をディスカバリルールフォームに入力します



必須の入力フィールドには、赤いアスタリスクが付いています。

検出する OID は、SNMP OID フィールドで次の形式で定義します: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

ここで、*{#MACRO1}*、*{#MACRO2}* ... は有効な lld マクロ名であり、*oid1*、*oid2*... はそれらのマクロに対して意味のある値を生成できる OID です。検出された OID のインデックスを含む組み込みマクロ *{#SNMPINDEX}* が、検出されたエンティティに適用されます。検出されたエンティティは、*{#SNMPINDEX}* マクロの値によってグループ化されます。

以下は、スイッチに対していくつかの snmpwalk を実行した例です。

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

そして、SNMP OID は次のように設定されます: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

このルールにより、{#IFDESCR} マクロが **WAN**、**LAN1**、**LAN2** に設定され、{#IFPHYSADDRESS} マクロが **8:0:27:90:7a:75**、**8:0:27:90:7a:76**、**8:0:27:2b:af:9e** に設定され、{#SNMPINDEX} マクロが検出された OID のインデックス **1**、**2**、**3** に設定されたエンティティが検出されます。

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
    },
    {
        "{#SNMPINDEX}": "2",
```

```
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
    }
]
```

あるエンティティに指定された OID が存在しない場合、そのエンティティでは対応するマクロは省略されます。たとえば、次のデータがあるとします。

```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

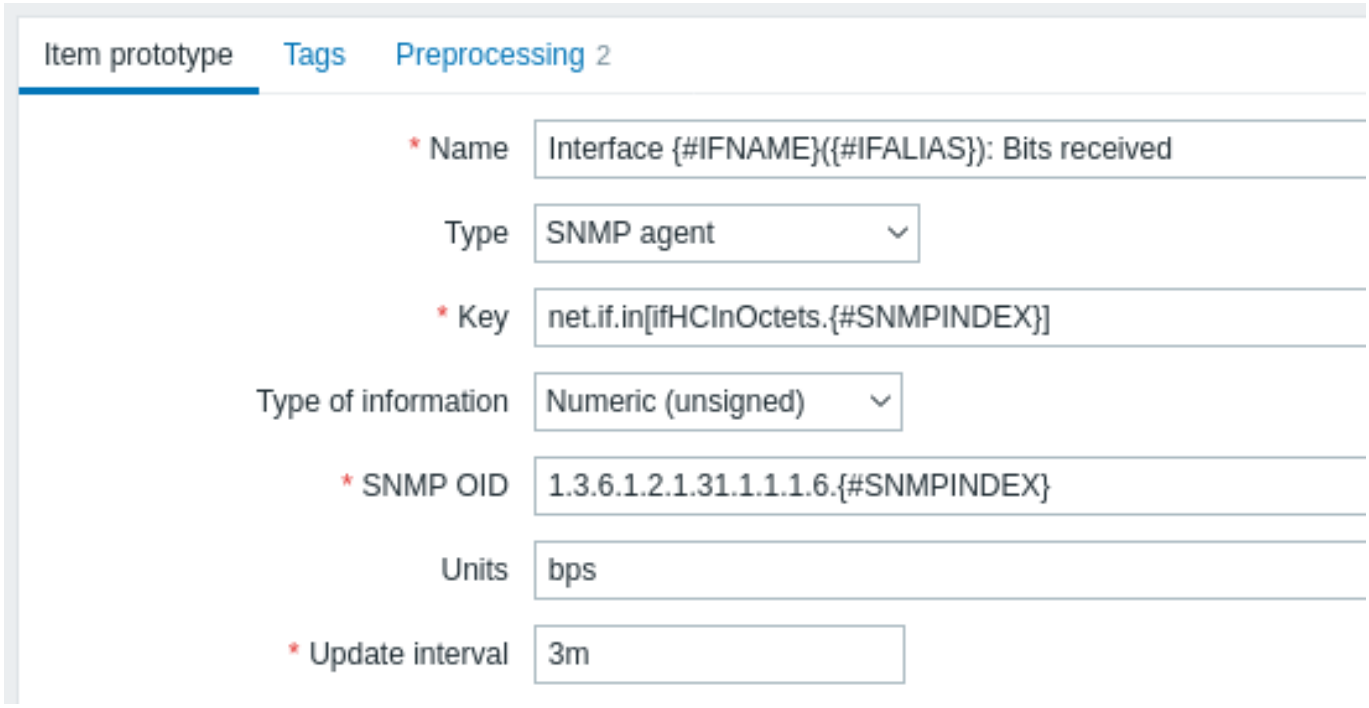
ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

この場合、SNMP ディスカバリ `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` は次の構造を返します。

```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```

アイテムのプロトタイプ

次のスクリーンショットは、これらのマクロをアイテムのプロトタイプでどのように使用できるかを示しています。



必要に応じて、アイテムのプロトタイプをいくつでも作成できます。

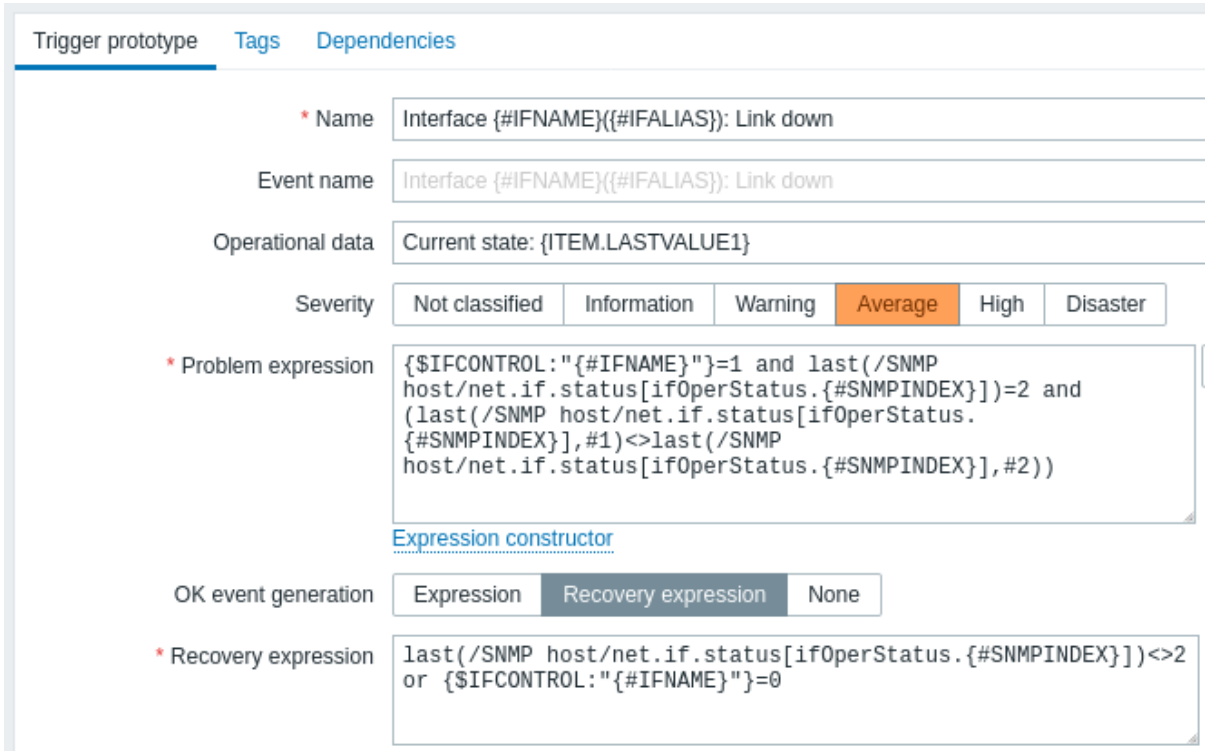
Item prototypes

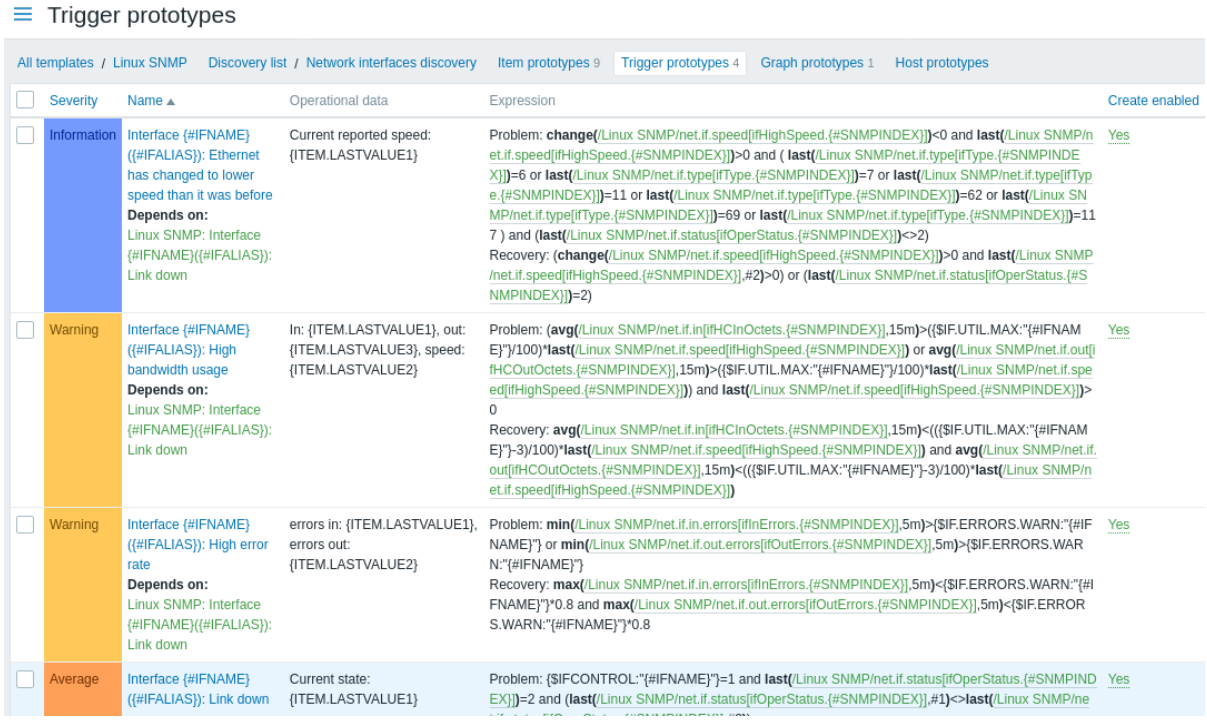
All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 Graph prototypes 1 Host prototypes

Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Interface type	net.if.type[ifType.{#SNMPINDEX}]	1h	7d	0d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]	1m	7d	0	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]	5m	7d	0d	SNMP agent	Yes

トリガープロトタイプ

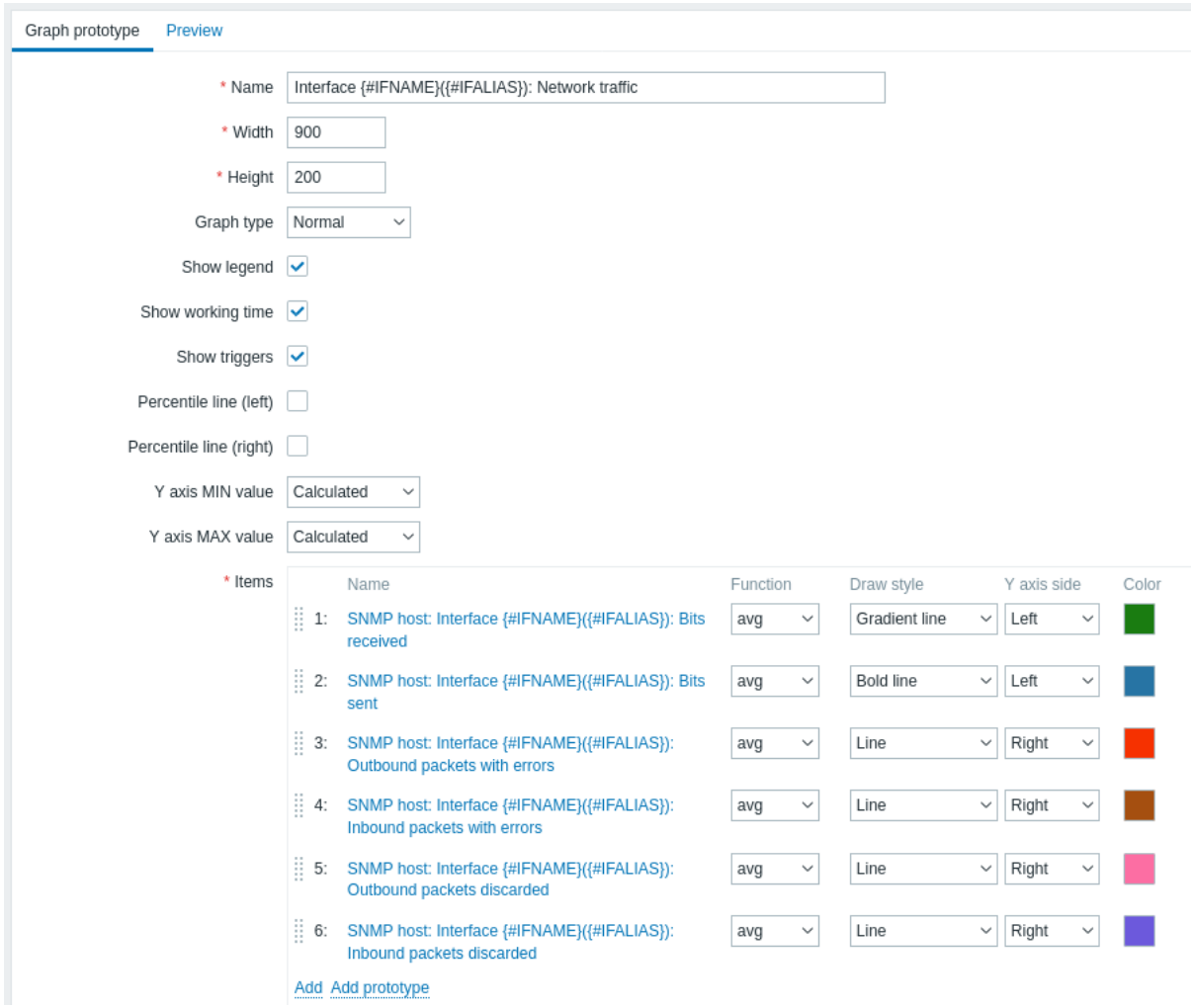
次のスクリーンショットは、これらのマクロをトリガープロトタイプでどのように使用できるかを示しています。

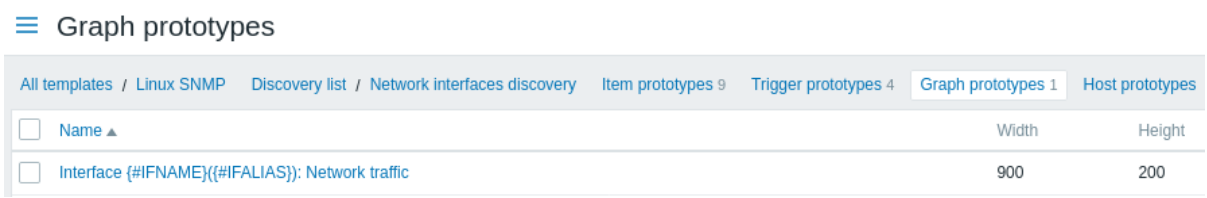




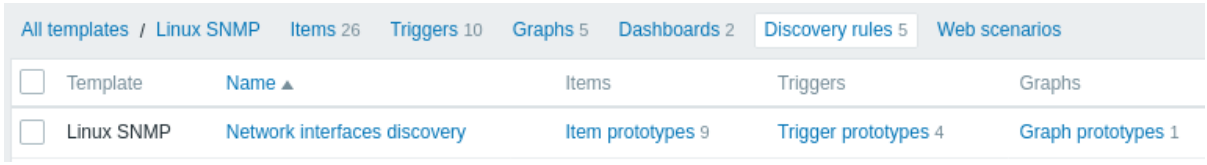
グラフのプロトタイプ

次のスクリーンショットは、グラフのプロトタイプでこれらのマクロをどのように使用できるかを示しています。





ディスカバリルールの概要:



検出されたエンティティ

サーバーが実行されると、SNMP ディスカバリルールが返す値に基づいて、実際のアイテム、トリガー、グラフが作成されます。ホスト設定では、それらには生成元のディスカバリルールへのオレンジ色のリンクが接頭表示されます。

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Severity	Value	Name ▲	Operational data	Expression
Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before **Depends on:** SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: **change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**<0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and (**last(**/SNMP host/net.if.type[ifType.2]**)**=6 or **last(**/SNMP host/net.if.type[ifType.2]**)**=7 or **last(**/SNMP host/net.if.type[ifType.2]**)**=11 or **last(**/SNMP host/net.if.type[ifType.2]**)**=62 or **last(**/SNMP host/net.if.type[ifType.2]**)**=69 or **last(**/SNMP host/net.if.type[ifType.2]**)**=117) and (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2) Recovery: (**change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2],#2**)**>0) or (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2)
Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage **Depends on:** SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (**avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) or **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 Recovery: **avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** and **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**
Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate **Depends on:** SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: **min(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} or **min(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} Recovery: **max(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8 and **max(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8
Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {$IFCONTROL:"enp4s0"}=1 and **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2 and (**last(**/SNMP host/net.if.status[ifOperStatus.2],#1**)**<>**last(**/SNMP host/net.if.status[ifOperStatus.2],#2**)**) Recovery: **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2 or {$IFCONTROL:"enp4s0"}=0

Graphs

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲
Mounted filesystem discovery: /: Disk space usage
Linux SNMP: CPU jumps
CPU discovery: CPU usage
CPU discovery: CPU utilization
Network interfaces discovery: Interface enp4s0(): Network traffic

6 JMX オブジェクトの検出

概要

すべての JMX MBean または MBean 属性を検出 したり、これらのオブジェクトの検出パターンを指定したりできます。

ディスカバリルールを設定するには、MBean と MBean 属性の違いを理解しておく必要があります。MBean は、管理が必要なデバイス、アプリケーション、または任意のリソースを表すことができるオブジェクトです。

たとえば、Web サーバーを表す MBean があります。その属性は、接続数、スレッド数、リクエストタイムアウト、http ファイルキャッシュ、メモリ使用量などです。この考えを人間の包括的な言語で表現すると、コーヒーマシンを豆と水の補充時間などを監視対象の属性として持つ MBean として定義できます。

アイテムキー

ディスカバリルール の設定で、タイプフィールドに **JMX agent** を選択します。

JMX オブジェクトのディスカバリでは、2 つのアイテムキー `jmx.discovery[]` および `jmx.get[]` がサポートされています。

アイテムキー	戻り値	パラメータ	コメント
jmx.discovery[<discovery mode>,<object name>,<unique short description>]	このアイテムは、MBean オブジェクトまたはその属性を記述する LLD マクロを含む JSON 配列を返します。	**discovery mode** - 次のいずれか: *attributes*（JMX MBean 属性を取得、デフォルト）または *beans*（JMX MBeans を取得） **object name** - 取得する MBean 名を識別するオブジェクト名パターン（documentationを参照）（デフォルトでは空で、登録されているすべての bean を取得） **unique short description** - 同じディスカバリモードおよびオブジェクト名を持つ複数の JMX アイテムをホスト上で使用できるようにする一意の説明（任意）	例: → jmx.discovery - すべての JMX MBean 属性を取得 → jmx.discovery[beans] - すべての JMX MBeans を取得 → jmx.discovery[attributes,"*:type=GarbageCollector,name - すべてのガベージコレクター属性を取得 → jmx.discovery[beans,"*:type=GarbageCollector,name=* - すべてのガベージコレクターを取得 マクロ名生成でサポートされる文字が限られているため、このアイテムが返せる MBean プロパティにはいくつかの制限があります（サポートされる文字は次の正規表現で表せます: `A-Z0-9_\.`）。そのため、たとえばハイフンを含む語や非 ASCII 文字を含む MBean プロパティをディスカバリするには、`jmx.get[]` を使用する必要があります。
jmx.get[<discovery mode>,<object name>,<unique short description>]			

アイテムキー			
	このアイテムは、MBean オブジェクトまたはその属性を含む JSON 配列を返します。 `jmx.discovery[]` と比較すると、LLD マクロは定義しません。	**discovery mode** - 次のいずれか: *attributes*（JMX MBean 属性を取得、デフォルト）または *beans*（JMX MBeans を取得） **object name** - 取得する MBean 名を識別するオブジェクト名パターン（documentationを参照）（デフォルトでは空で、登録されているすべての bean を取得） **unique short description** - 同じディスカバリモードおよびオブジェクト名を持つ複数の JMX アイテムをホスト上で使用できるようにする一意の説明（任意）	このアイテムを使用する場合は、返された JSON から JSONPath を使用して抽出した値を指すカスタムのローレベルディスカバリマクロを定義する必要があります。

Attention:
パラメータが渡されない場合、JMX からすべての MBean 属性が要求されます。JMX ディスカバリでパラメータを指定しない場合や、`*:type=*,name=*` のような広い範囲ですべての属性を取得しようとすると、パフォーマンス上の問題が発生する可能性があります。

jmx.discovery の使用

このアイテムは、MBean オブジェクトまたは属性を記述するローレベルディスカバリマクロを含む JSON オブジェクトを返します。たとえば、MBean 属性のディスカバリでは、次のようになります（見やすさのために整形）：


```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
```

```
        "{#JMXTYPE}":"javax.management.ObjectName",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

MBean のディスカバリでは、次のようになります（見やすさのために整形）：

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

サポートされているマクロ

次のマクロはディスカバリルールフィルター およびアイテム、トリガー、グラフのプロトタイプでの使用がサポートされています。

マクロ	説明
MBean 属性の検出	
{#JMXVALUE}	属性値
{#JMXTYPE}	属性タイプ
{#JMXOBJ}	オブジェクト名
{#JMXDESC}	属性名を含むオブジェクト名
{#JMXATTR}	属性名
MBean の検出	
{#JMXDOMAIN}	MBean ドメイン. (*Zabbix* 予約名)
{#JMXOBJ}	オブジェクト名. (*Zabbix* 予約名)
{#JMX<key property>}	MBean プロパティ (like {#JMXTYPE}, {#JMXNAME}) (参照Limitations below).

制限事項

MBean のプロパティ名から LLD マクロ名を作成するアルゴリズムには、いくつかの制限があります。

- 属性名は大文字に変換されます
- 属性名に LLD マクロ名でサポートされていない文字が含まれている場合、その属性名は無視されます（LLD マクロは生成されません）。サポートされる文字は、次の正規表現で表せます: `A-Z0-9_\.`
- 属性名が"obj" または"domain" の場合、予約済みの Zabbix プロパティ {#JMXOBJ} および {#JMXDOMAIN} の値と重複するため、無視されます

次の jmx.discovery（"beans" モード）の例を考えてみてください。MBean には以下のプロパティが定義されています（一部は無視されます。以下を参照してください）。

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

JMX ディスカバリの結果、次の LLD マクロが生成されます。

- {#JMXDOMAIN} - MBean のドメインを表す Zabbix 内部用
- {#JMXOBJ} - MBean オブジェクトを表す Zabbix 内部用
- {#JMXNAME} - "name" プロパティから作成

無視されるプロパティは次のとおりです。

- тип：名前にサポートされていない文字（非 ASCII 文字）が含まれているため
- attributes[]：名前にサポートされていない文字が含まれているため（角括弧はサポートされていません）
- Name：すでに定義されているため（name=test）
- domAin：Zabbix の予約名であるため

例

MBean を使用した LLD ルール作成の、より実践的な例をさらに 2 つ見ていきましょう。
MBean を収集する LLD ルールと、MBean 属性を収集する LLD ルールの違いをよりよく理解するために、次の表を参照してください。

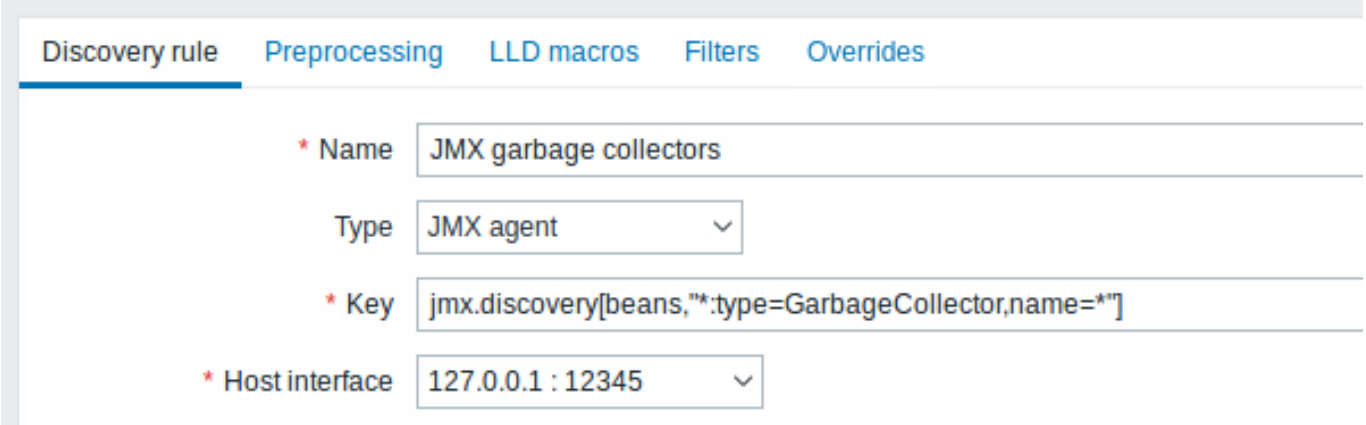
MBean1	MBean2	MBean3
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

例 1: MBean の検出

このルールは 3 つのオブジェクト、すなわち列の先頭行である MBean1、MBean2、MBean3 を返します。

オブジェクトの詳細については、supported macros テーブルの *MBean* の検出セクションを参照してください。

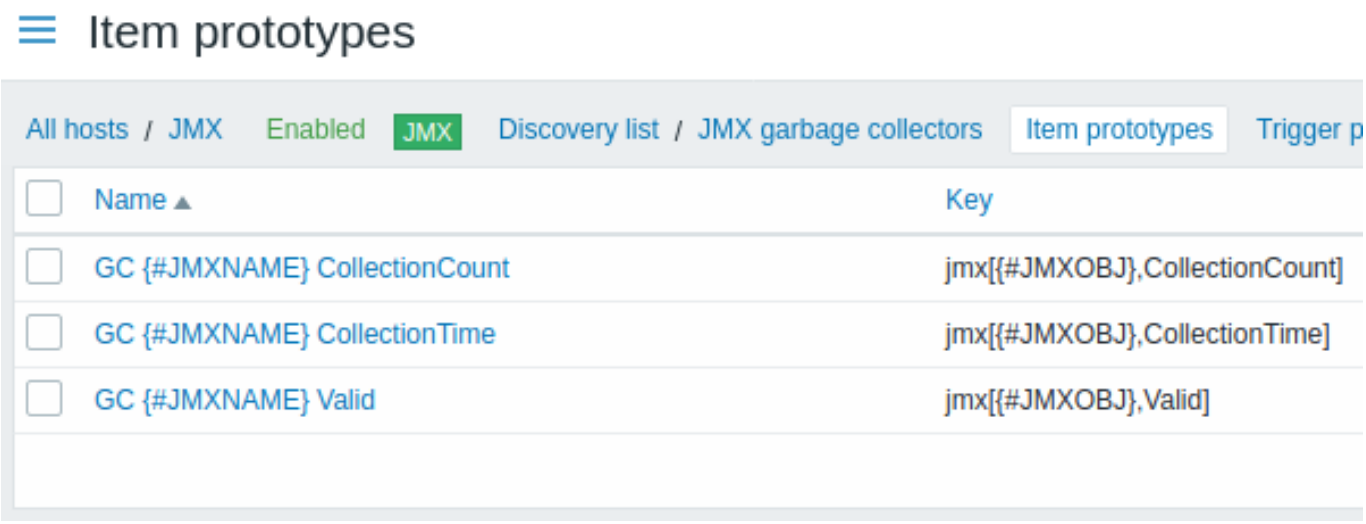
属性を含まない MBean を収集するディスカバリルールの設定は、次のようになります。



ここで使用されるキー:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

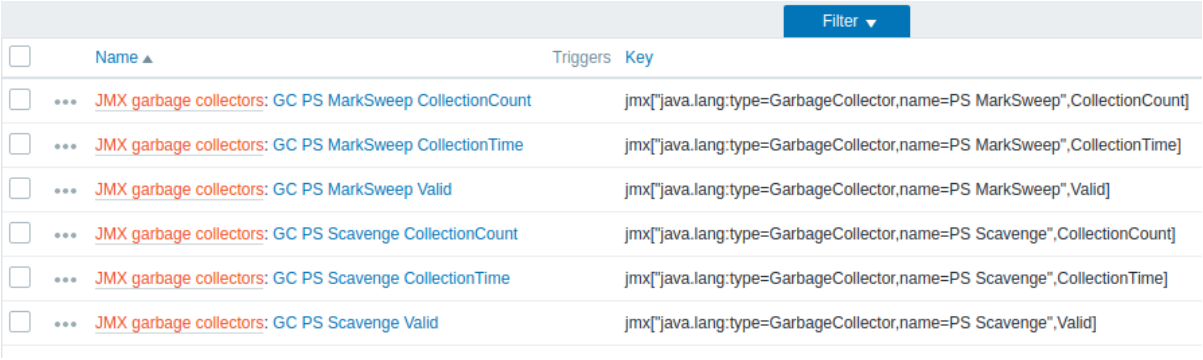
属性を持たないすべてのガベージコレクターが検出されます。ガベージコレクターは同じ属性セットを持つため、アイテムのプロトタイプで必要な属性を次のように使用できます。



ここで使用されるキー:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

LLD ディスカバリルールの結果は、次のようになります（2 つのガベージコレクターに対してアイテムが検出されています）。

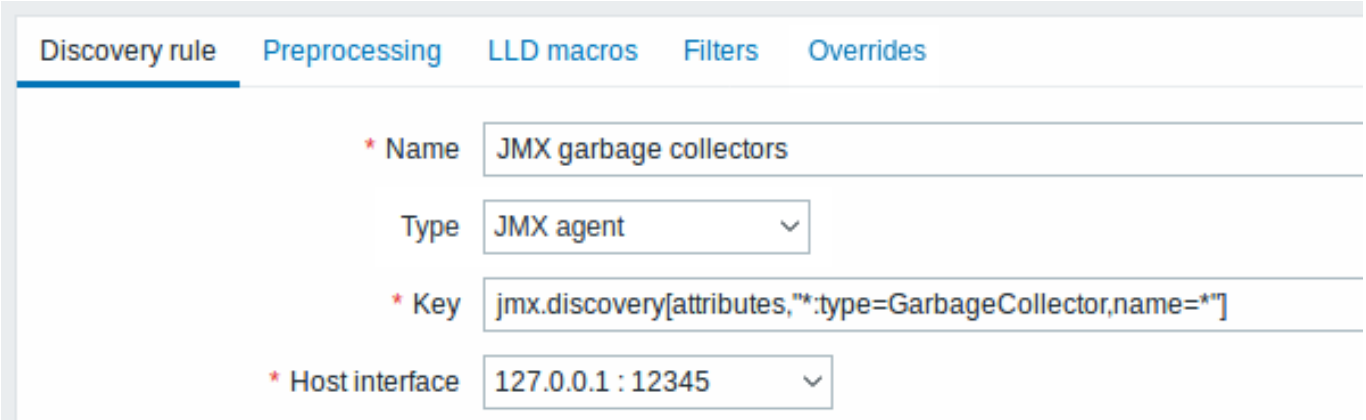


例 2: MBean 属性のディスカバリ

このルールは､以下のフィールドを持つ 9 個のオブジェクトを返します: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attrib MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

オブジェクトの詳細については、supported macros 表の *Discovery of MBean attributes* セクションを参照してください。

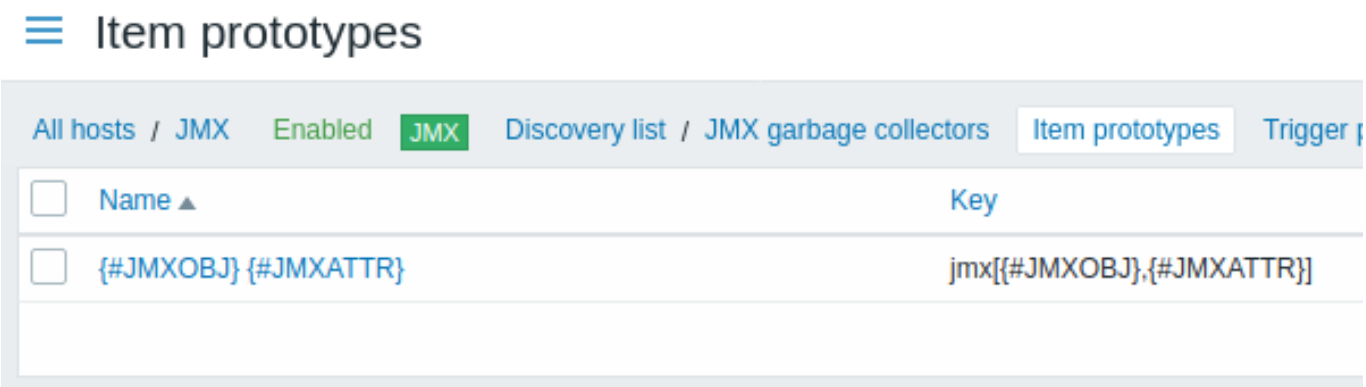
MBean 属性を収集するディスカバリルールの設定は、次のようになります:



ここで使用されるキー:

`jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]`

単一のアイテム属性を持つすべてのガベージコレクターがディスカバリされます。



この特定のケースでは、各 MBean 属性ごとにプロトタイプからアイテムが作成されます。この設定の主な欠点は、すべての属性に対してアイテムプロトタイプが 1 つしかないため、トリガープロトタイプからトリガーを作成できないことです。そのため、この設定はデータ収集には使用できますが、自動監視には推奨されません。

jmx.get の使用

`jmx.get[]` は `jmx.discovery[]` アイテムに似ていますが、Java オブジェクトのプロパティをローレベルディスカバリのマクロ名に変換しないため、ハイフンや非 ASCII 文字など、LLD マクロ名の生成に関連する制限を受けずに値を返すことができます。

ディスカバリに `jmx.get[]` を使用する場合、ローレベルディスカバリのマクロは、ディスカバリルール設定のカスタムLLD マクロタブで別途定義でき、JSONPath を使用して必要な値を指定できます。

MBeans の検出

検出アイテム: `jmx.get[beans,"com.example:type=*,*"]`

レスポンス:


```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```


MBean 属性の検出

検出アイテム: `jmx.get[attributes,"com.example:type=*,*"]`

レスポンス:


```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```


7 IPMI センサーの検出

概要

IPMI センサーを自動的に検出することが可能です。

これを行うには次の組み合わせを使用します。

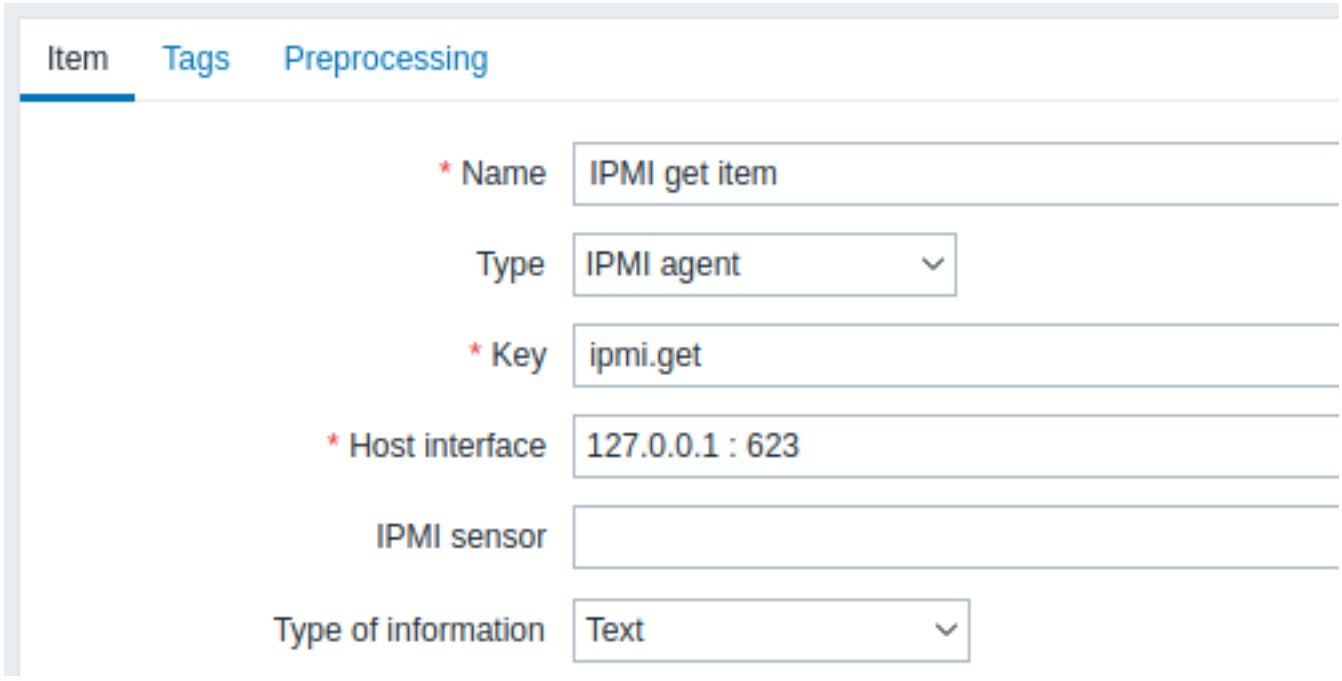
- マスターアイテムとしての `ipmi.get`IPMI アイテム
- 依存するローレベルディスカバリルールとアイテムのプロトタイプ

設定

マスターアイテム

次のキーを使用して IPMI アイテムを作成します。

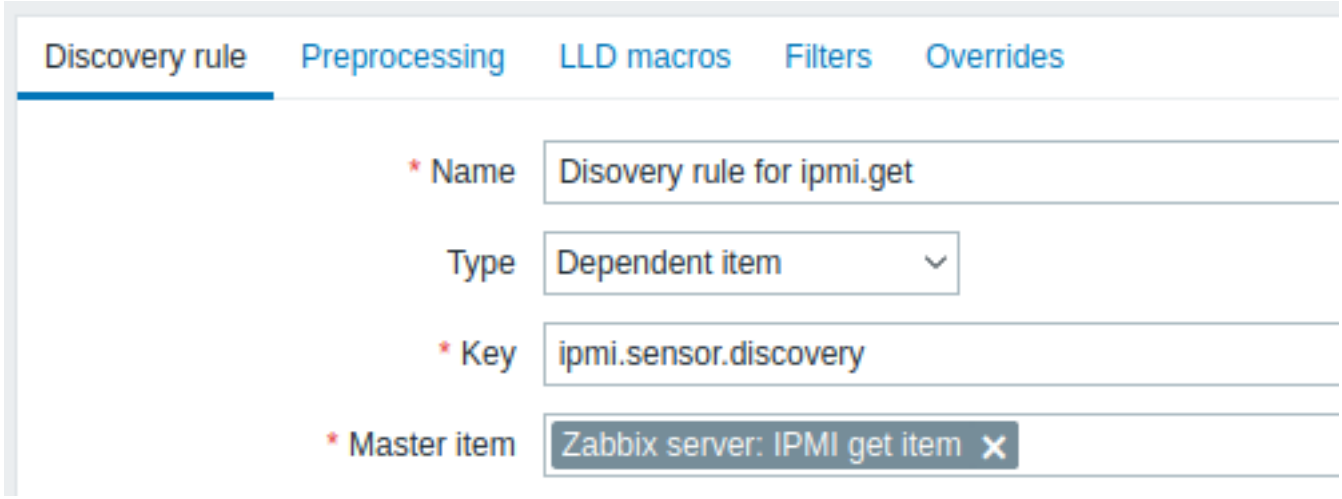
`ipmi.get`



JSON データが大きくなる可能性がある場合は、データ型を” テキスト ” に設定します。

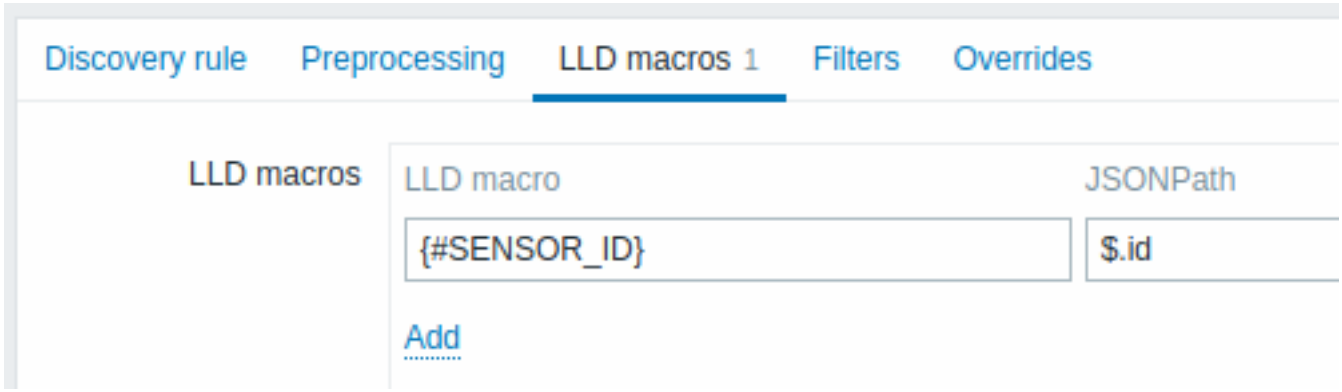
依存 LLD ルール

” 依存アイテム ” タイプとしてローレベルディスカバリルールを作成します。



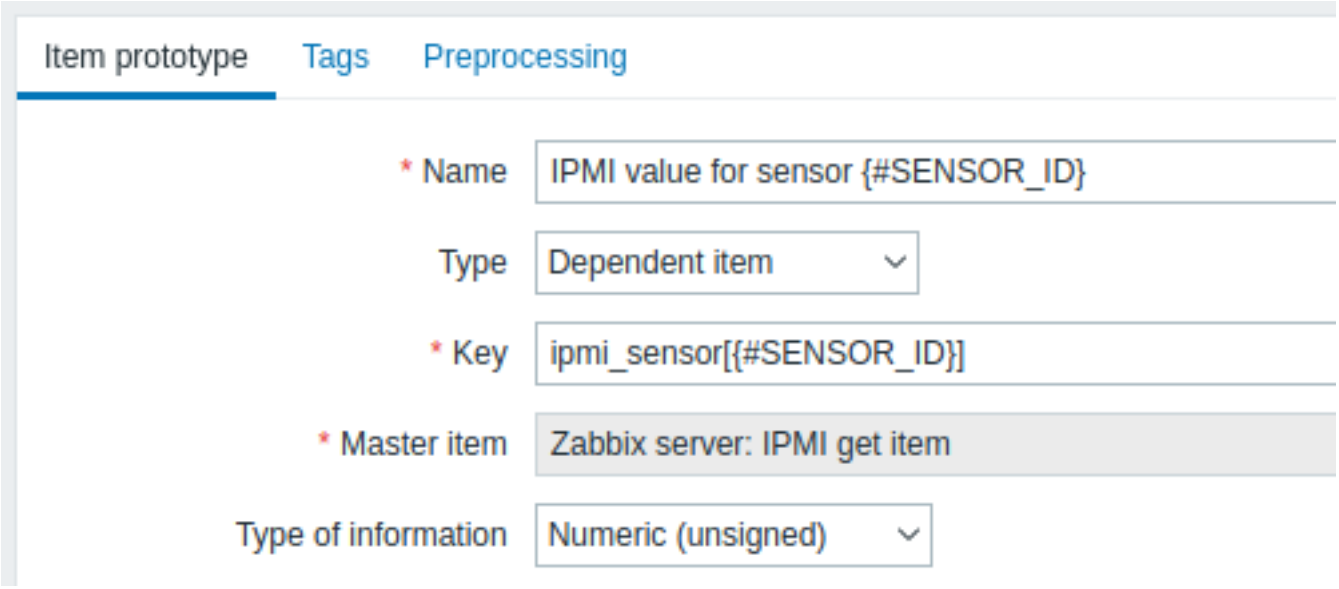
マスターアイテムとして作成した `ipmi.get` アイテムを選択します。

”LLD マクロ ” タブで対応する JSONPath を使用してカスタムマクロを定義します。



依存アイテムのプロトタイプ

この LLD ルールで” 依存アイテム ” タイプのアイテムのプロトタイプを作成します。このプロトタイプのマスターアイテムとして、作成した `ipmi.get` アイテムを選択します。



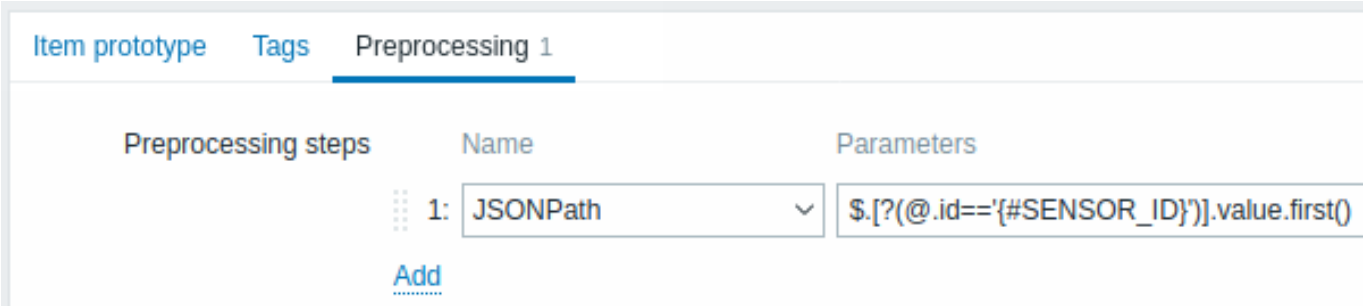
アイテムのプロトタイプ名とキーで {#SENSOR_ID} マクロが使用されていることに注意してください。

- 名前: IPMI value for sensor {#SENSOR_ID}
- キー: ipmi_sensor[{#SENSOR_ID}]

データ型として数値（整数）

アイテムのプロトタイプの” 保存前処理” タブで JSONPath を選択し、次の JSONPath 式をパラメーターとして使用します。

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



ディスカバリが開始されると、IPMI センサーごとに 1 つのアイテムが作成されます。このアイテムは、指定されたセンサーの整数値を返します。

8 systemd サービスの検出

概要

Zabbix を使用して systemd ユニット（デフォルトではサービス）を検出することができます。

アイテムキー

ディスカバリルールで使用するアイテムは、次のとおりです。

```
systemd.unit.discovery
```

> **Attention:**
> このアイテムキーは、Zabbix エージェント 2 でのみサポートされています。

このアイテムは、systemd ユニットに関する情報を含む JSON を返します。例:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "MySQL Server",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
```

```
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

無効化された systemd ユニットのディスカバリ

無効化された systemd ユニットをディスカバリすることも可能です。この場合、結果の JSON では 3 つのマクロが返されます。

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:
無効化された systemd ユニット用のプロトタイプからアイテムとトリガーを作成するには、{#UNIT.ACTIVESTATE} および {#UNIT.UNITFILESTATE} に対する禁止 LLD フィルターを調整する（または削除する）ようにしてください。

サポートされているマクロ

以下のマクロは、ディスカバリルールのfilter およびアイテム、トリガー、グラフのプロトタイプで使用できます。

Macro	Description
{#UNIT.NAME}	プライマリユニット名。
{#UNIT.DESCRIPTION}	人が読みやすい説明。
{#UNIT.LOADSTATE}	ロード状態（つまり、ユニットファイルが正常に読み込まれたかどうか）。
{#UNIT.ACTIVESTATE}	アクティブ状態（つまり、ユニットが現在開始されているかどうか）。
{#UNIT.SUBSTATE}	サブ状態（アクティブ状態をより細かくしたもので、ユニットタイプ固有の状態）。
{#UNIT.FOLLOWED}	このユニットが状態を追従しているユニット。存在しない場合は空文字列。
{#UNIT.PATH}	ユニットオブジェクトパス。
{#UNIT.JOBID}	ジョブユニットにキューされたジョブがある場合は数値のジョブ ID、それ以外は 0。
{#UNIT.JOBTYPE}	ジョブタイプ。
{#UNIT.JOBPATH}	ジョブオブジェクトパス。
{#UNIT.UNITFILESTATE}	ユニットファイルのインストール状態。
{#UNIT.SERVICETYPE}	サービスユニットのタイプ（例: `simple`、`forking`、`oneshot`、`idle` など）。このマクロは、ユニットがサービスである場合にのみ返されます。

アイテムのプロトタイプ

systemd サービスディスカバリに基づいて作成できるアイテムのプロトタイプには、例えば次のものがあります。

- アイテム名: {#UNIT.DESCRIPTION} active state info、アイテムキー: systemd.unit.info["{#UNIT.NAME}"]
- アイテム名: {#UNIT.DESCRIPTION} load state info、アイテムキー: systemd.unit.info["{#UNIT.NAME}",LoadState]

9 Windows サービスの検出

概要

ファイルシステムが検出されるのと同様の方法で、Windows サービスも検出できます。

アイテムキー

ディスカバリールールで使用するアイテムは次のとおりです。

`service.discovery`

サポートされているマクロ

以下のマクロは、ディスカバリルールのfilter およびアイテム、トリガー、グラフのプロトタイプで使用できます。

Macro	Description
{#SERVICE.NAME}	サービス名。
{#SERVICE.DISPLAYNAME}	表示されるサービス名。
{#SERVICE.DESCRIPTION}	サービスの説明。
{#SERVICE.STATE}	サービス状態の数値。 詳細はservice.info アイテムを参照してください。
{#SERVICE.STATENAME}	サービス状態の名前。 詳細はservice.info アイテムを参照してください。
{#SERVICE.PATH}	サービスパス。
{#SERVICE.USER}	サービスユーザー。
{#SERVICE.STARTUP}	サービス起動タイプの数値。 詳細はservice.info アイテムを参照してください。
{#SERVICE.STARTUPNAME}	サービス起動タイプの名前。 詳細はservice.info アイテムを参照してください。
{#SERVICE.STARTUPTRIGGER}	サービス起動タイプが以下のいずれかであることを示す数値です。 0 - 起動トリガーなし 1 - 起動トリガーあり *Automatic (trigger start)*、*Automatic delayed (trigger start)*、*Manual (trigger start)* のようなサービス起動タイプをディスカバリする際に便利です。

Windows サービスディスカバリに基づいて、次のようなアイテムプロトタイプを作成できます。

`service.info[{#SERVICE.NAME},<param>]`

ここで param には次の値を指定できます: *state*、*displayname*、*path*、*user*、*startup*、または *description*。

たとえば、サービスの表示名を取得するには、"service.info[{#SERVICE.NAME},displayname]" アイテムを使用できます。param の値が指定されていない場合 ("service.info[{#SERVICE.NAME}]") は、デフォルトの *state* パラメータが使用されます。

10 Windows パフォーマンスカウンターインスタンスの検出

概要

Windows パフォーマンスカウンターのオブジェクトインスタンスをディスカバリすることができます。
これは、複数インスタンスのパフォーマンスカウンターで役立ちます。

アイテムキー

ディスカバリルールを設定するには、次のアイテムを使用します。

- `perf_instance.discovery[object]`

オブジェクト名はローカライズされる場合があることに注意してください。例えば:

```
perf_instance.discovery[Processor] # オブジェクト名は英語です。
perf_instance.discovery[Processador] # オブジェクト名はポルトガル語です。
```

また、OS のローカライズに関係なく、オブジェクト名を英語で指定するには、次のアイテムを使用します。

- `perf_instance_en.discovery[object]`

例えば:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

サポートされているマクロ

ディスカバリープロセスは、指定されたオブジェクトのすべてのインスタンスを {#INSTANCE} マクロで返します。

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

このマクロは、perf_counter[] および perf_counter_en[] アイテムのプロトタイプで使用できます。

たとえば、ディスカバリールールで使用されるアイテムキーが perf_instance.discovery[Processor] の場合、次のアイテムプロトタイプを作成できます。

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

注意:

- 指定されたオブジェクトが見つからない場合、または可変インスタンスをサポートしていない場合、ディスカバリーアイテムは NOTSUPPORTED になります。
- 指定されたオブジェクトが可変インスタンスをサポートしていても、現在インスタンスが存在しない場合は、空の JSON 配列が返されます。
- 重複するインスタンスはスキップされます。

11 WMI クエリを使用した検出

概要

WMIは Windows の強力なインターフェイスであり、インストールされている Windows コンポーネント、サービス、状態、およびソフトウェアに関するさまざまな情報を取得するために使用できます。

物理ディスクの検出とそのパフォーマンスデータの収集、ネットワークインターフェイスの検出、Hyper-V ゲストの検出、Windows サービスの監視など WindowsOS の他の多くのことに使用できます。

このタイプのローレベルディスカバリは、結果がローレベルディスカバリに適した JSON オブジェクトに自動的に変換される WQL クエリを使用して実行されます。

アイテムキー

ディスカバリルールで使用するアイテムは次のとおりです。

```
wmi.getall[<namespace>,<query>]
```

このアイテムは、クエリ結果を JSON 配列に変換します。例えば、次のようなクエリです。

```
select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'
```

は、次のような結果を返す場合があります。

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
    },
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
```

```
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```


ローレベルディスカバリマクロ

返された JSON にローレベルディスカバリマクロは作成されませんが、これらのマクロは、返された JSON で検出された値を指す JSONPath でカスタム LLD マクロ機能を使用して、追加の手順としてユーザーが定義できます。

その後、マクロを使用して、アイテム、トリガーなどのプロトタイプを作成できます。

12 ODBC SQL クエリを使用した検出

概要

このタイプのローレベルディスカバリは SQL クエリで実行され、その結果はローレベルディスカバリに適した JSON オブジェクトに自動的に変換されます。

アイテムキー

SQL クエリは” データベースモニター” アイテムタイプを使用して実行されます。したがってODBC 監視ページのほとんどの手順は、機能する” データベースモニター” ディスカバリルールを取得するために適用されます。

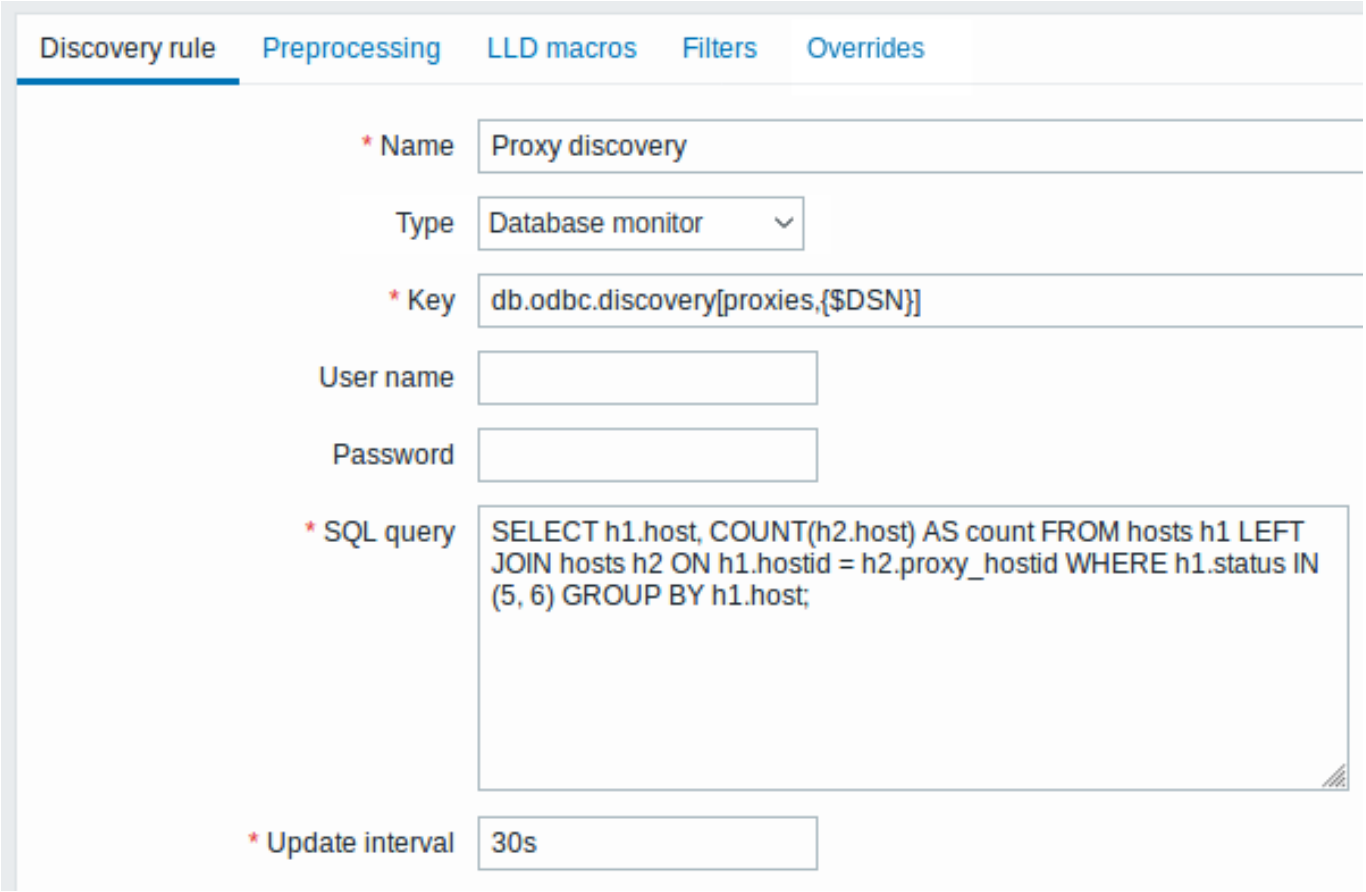
” データベースモニター” ディスカバリルールでは、次の 2 つのアイテムキーを使用できます。

- **db.odbc.discovery**[<unique short description>,<dsn>,<connection string>] - このアイテムは、SQL クエリ結果を JSON 配列に変換し、クエリ結果の列名を検出されたフィールド値とペアになったローレベルディスカバリマクロ名に変換します。これらのマクロは、アイテム、トリガーなどのプロトタイプの作成に使用できます。参照: db.odbc.discovery の使用。
- **db.odbc.get**[<unique short description>,<dsn>,<connection string>] - このアイテムは、SQL クエリ結果を JSON 配列に変換し、クエリ結果の元の列名を、検出された値とペアになった JSON のフィールド名として保持します。`db.odbc.discovery[]` と比較すると、このアイテムは返された JSON にローレベルディスカバリマクロを作成しないため、列名が有効なマクロ名であるかどうかを確認する必要はありません。ローレベルディスカバリマクロは、必要に応じて追加のステップとして定義できます。カスタム LLD マクロ機能を使用して、返される JSON で検出された値を指す JSONPath を使用します。参照: using db.odbc.get の使用。

db.odbc.discovery の使用

次の例では、Zabbix データベースに対する ODBC クエリに基づき、Zabbix プロキシのローレベルディスカバリを使用して、SQL クエリがどのように JSON に変換されるかを示します。これは、どのプロキシが稼働しているかを監視するための”zabbix[proxy,<name>,lastaccess]” 内部アイテム を自動作成するのに役立ちます。

まず、ディスカバリルールの設定から始めます。



必須の入力フィールドには、赤いアスタリスクが付いています。

ここでは、Zabbix データベースに対する次の直接クエリを使用して、すべての Zabbix プロキシと、それぞれが監視しているホスト数を選択しています。ホスト数は、たとえば空のプロキシを除外するためのフィルタリングに使用できます。

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

”db.odbc.discovery[,{$DSN}]” アイテムの内部動作により、このクエリの結果は自動的に次の JSON に変換されます。

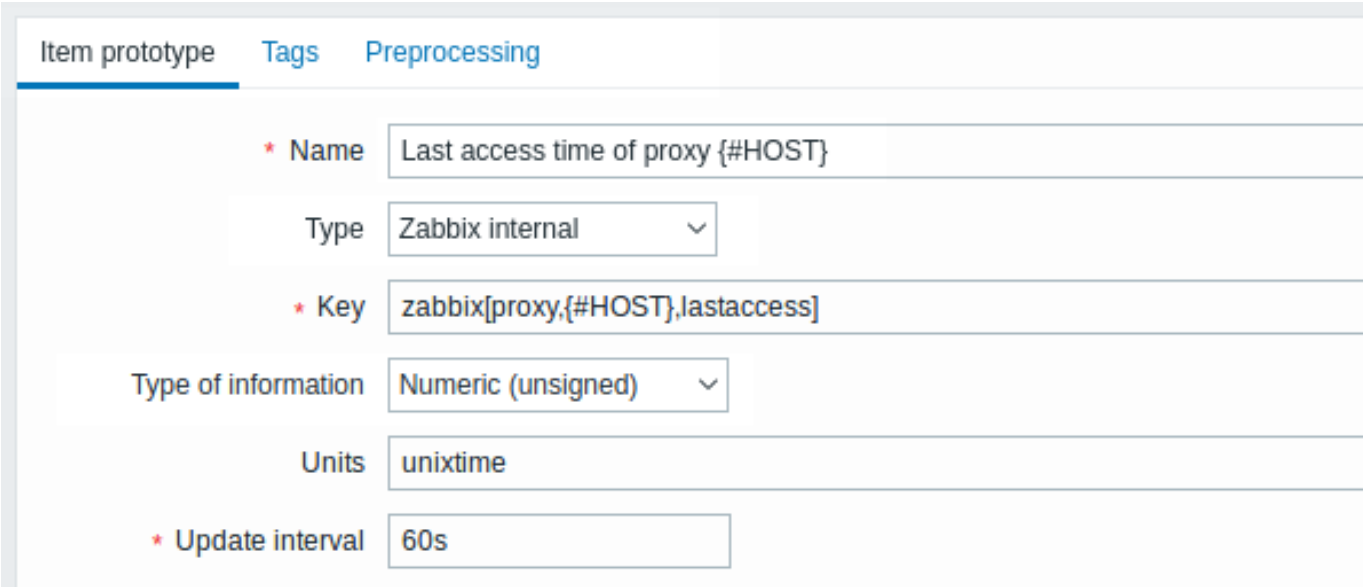
```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
]
```

列名がマクロ名になり、選択された行がそれらのマクロの値になることがわかります。

Note:
列名がどのようにマクロ名へ変換されるかが明確でない場合は、上記の例の"COUNT(h2.host) AS count" のように列エイリアスを使用することを推奨します。
列名を有効なマクロ名に変換できない場合、ディスカバリルールは未サポートとなり、エラーメッセージに問題のある列番号が表示されます。さらに情報が必要な場合は、取得された列名が Zabbix サーバーのログファイルに DebugLevel=4 で出力されます。

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 L
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

これで SQL クエリがどのように JSON オブジェクトへ変換されるかを理解できたので、アイテムのプロトタイプで {#HOST} マクロを使用できます。



ディスカバリが実行されると、各プロキシに対してアイテムが作成されます。



db.odbc.get を使用する

db.odbc.get[,{$DSN}] と次の SQL の例を使用します。

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxy_ho
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

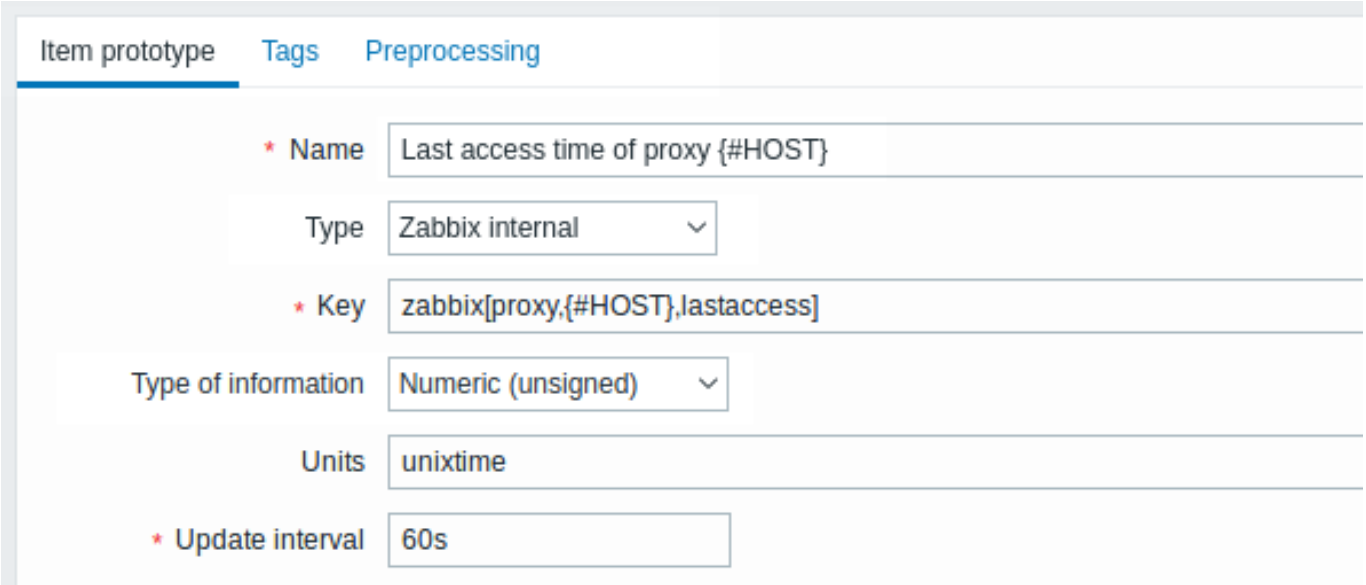
この JSON が返されます：

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

ご覧のとおりそこにはローレベルディスカバリマクロはありません。ただしカスタムのローレベルディスカバリマクロは、JSONPathを使用してディスカバリルールのLLD マクロタブで作成できます。次に例を示します。

`{#HOST}` → `$.host`

これで、この {#HOST} マクロをアイテムのプロトタイプで使用できるようになりました。



13 Prometheus データを使用した検出

概要

Prometheus ラインフォーマットで提供されるデータは、ローレベルディスカバリに使用できます。

Zabbix での Prometheus データクエリの実装方法の詳細についてはPrometheus チェックを参照してください。

設定

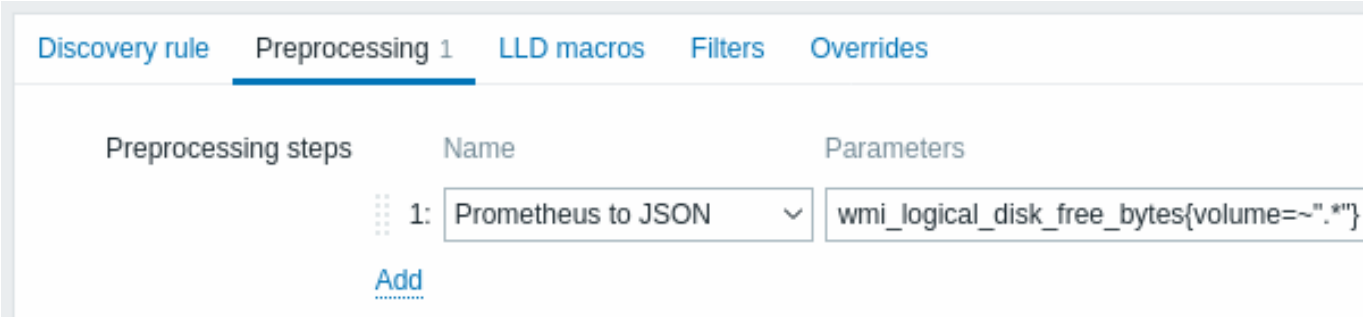
ローレベルディスカバリルールは、Prometheus データを収集する HTTP マスターアイテムへの依存アイテムとして作成する必要があります。

プロメテウスから JSON

ディスカバリルールで [保存前処理] タブに移動し、*Prometheus* から *JSON* オプションを選択します。ディスカバリには JSON 形式のデータが必要であり、*Prometheus* から *JSON* オプションは、次の属性を使用してそれを返します。

- metric name
- metric value
- help (if present)
- type (if present)
- labels (if present)
- raw line

例：`wmi_logical_disk_free_bytes` をクエリします。



これらのプロメテウスラインから：

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

以下のように返します

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

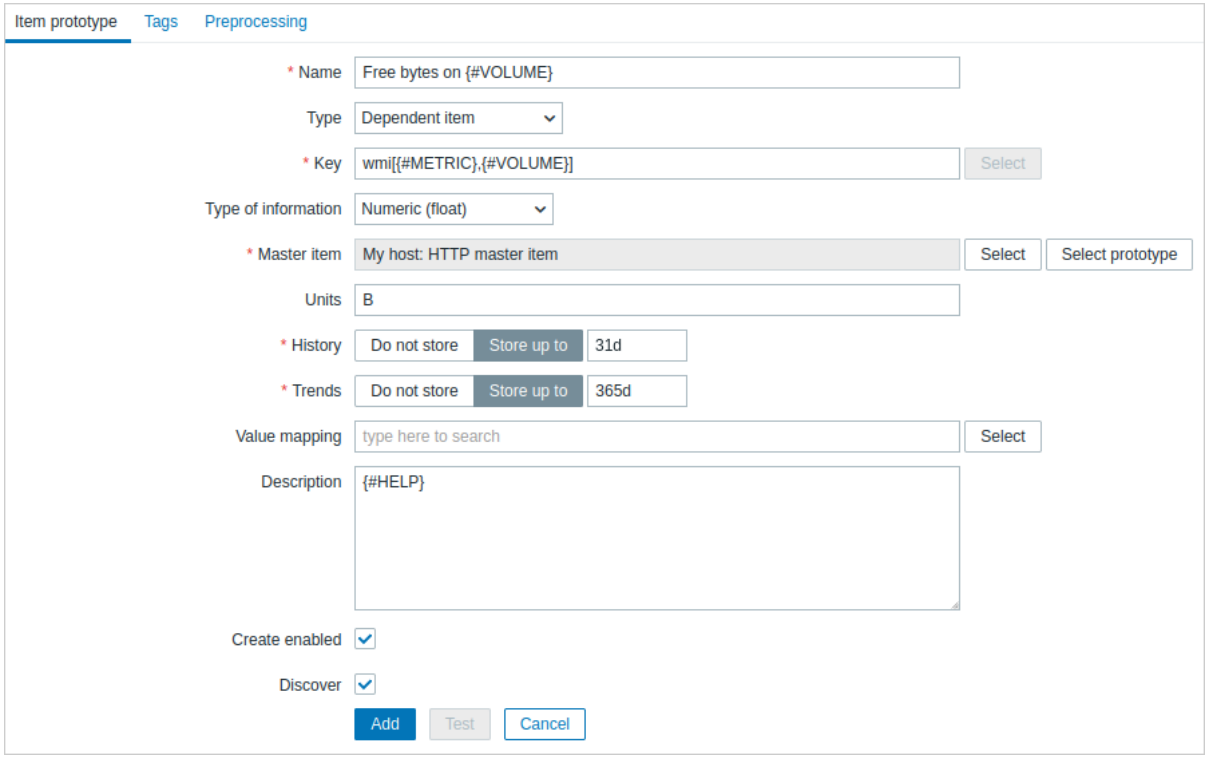
LLD マクロマッピング

次に LLD マクロタブに移動して次のマッピングを行います。

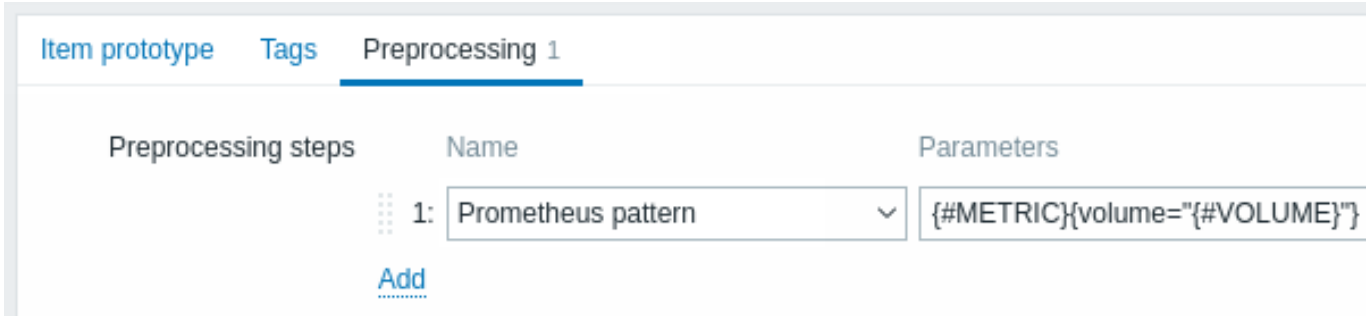
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

アイテムのプロトタイプ

次のようなアイテムのプロトタイプを作成することをお勧めします。



保存前処理オプション:



14 ブロックデバイスの検出

ファイルシステムが検出されるのと同様の方法で、ブロックデバイスとそのタイプも検出できます。

アイテムキー

ディスカバリルールで使用するアイテムキーは次のとおりです。

`vfs.dev.discovery`

このアイテムは Linux プラットフォームでのみサポートされています。

このディスカバリアイテムを使用して、次のようなディスカバリルールを作成できます。

- フィルター: **{#DEVNAME}** が `sd[\D]$` に一致 - 「sd0」、「sd1」、「sd2」、... という名前のデバイスをディスカバリします
- フィルター: **{#DEVTYPE}** が `disk` に一致 **AND {#DEVNAME}** が `^loop.*` に一致しない - 名前が「loop」で始まらない disk タイプのデバイスをディスカバリします

サポートされているマクロ

このディスカバリキーは、ブロックデバイスの名前とタイプをそれぞれ識別する {#DEVNAME} と {#DEVTYPE} の 2 つのマクロを返します。次に例を示します。

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"dm-0",
```

```
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partition"
    }
]
```

ブロックデバイスの検出では、vfs.dev.read[] および vfs.dev.write[] アイテムと {#DEVNAME} マクロを使用してアイテムのプロトタイプを作成できます。次に例を示します。

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} は、デバイスのフィルタリングを目的としています。

15 Zabbix でのホストインターフェース検出

概要

ホストの Zabbix フロントエンドで設定されているすべてのインターフェースを検出することができます。

アイテムキー

ディスカバリルールで使用するアイテムは、Zabbix 内部アイテムです。

zabbix[host,discovery,interfaces]

このアイテムは、インターフェースの説明を含む JSON を返します。内容は以下のとおりです。

- IP アドレス/DNS ホスト名（ホストの「接続先」設定に応じて決まります）
- ポート番号
- インターフェースタイプ（Zabbix エージェント、SNMP、JMX、IPMI）
- デフォルトのインターフェースかどうか
- 一括リクエスト機能が有効かどうか - SNMP インターフェースのみ

例:

```
[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG
```

複数のインターフェースがある場合、JSON 内のレコードは次の順序で並びます。

- インターフェースタイプ
- デフォルト - デフォルトのインターフェースは、デフォルトでないインターフェースより前に配置されます
- インターフェース ID（昇順）

サポートされているマクロ

以下のマクロは、ディスカバリルールのフィルター およびアイテム、トリガー、グラフのプロトタイプで使用できます。

Macro	Description
{#IF.CONN}	インターフェースの IP アドレスまたは DNS ホスト名。
{#IF.IP}	インターフェースの IP アドレス。
{#IF.DNS}	インターフェースの DNS ホスト名。
{#IF.PORT}	インターフェースのポート番号。
{#IF.TYPE}	インターフェースのタイプ（"AGENT"、"SNMP"、"JMX"、または"IPMI"）。
{#IF.DEFAULT}	インターフェースのデフォルト状態： 0 - デフォルトインターフェースではない 1 - デフォルトインターフェース
{#IF.SNMP.BULK}	インターフェースの SNMP バルク処理状態： 0 - 無効 1 - 有効 このマクロは、インターフェースタイプが "SNMP" の場合にのみ返されます。

概要

完全にカスタムの LLD ルールを作成し、あらゆる種類のエンティティをディスカバリすることも可能です。例えば、データベースサーバー上のデータベースなどです。

これを行うには、JSON 文字列を返すカスタムアイテムを作成する必要があります。この JSON 文字列では、検出されたオブジェクトと、必要に応じてそれらのいくつかのプロパティを指定します。エンティティごとのマクロ数に制限はありません。組み込みのディスカバリルールでは 1 つまたは 2 つのマクロ（例えば、ファイルシステムのディスカバリでは 2 つ）が返されますが、それより多くを返すことも可能です。

例

必要な JSON 文字列形式は、例を見るのが最もわかりやすいでしょう。たとえば、古い Zabbix 1.8 エージェント（vfs.fs.discovery キーをサポートしていないもの）を実行しているが、それでもファイルシステムをディスカバリする必要があるとします。
以下は、マウントされたファイルシステムを検出し、ファイルシステム名とタイプの両方を含む JSON を出力する Linux 用のシンプルな Perl スクリプトです。これを使用する方法の 1 つは、キー”vfs.fs.discovery_perl” を持つ UserParameter として使用することです。

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"{#FSNAME}\":\"$fsname\",\n";
    print "\t\t\"{#FSTYPE}\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

Attention:
LLD マクロ名に使用できる記号は **0-9**、**A-Z**、_、**.** です。
名前で小文字はサポートされていません。

その出力例（わかりやすさのために整形済み）を以下に示します。カスタムディスカバリチェック用の JSON も同じ形式に従う必要があります。

```
[
    { "{#FSNAME}":"/",                          "{#FSTYPE}":"rootfs"   },
    { "{#FSNAME}":"/sys",                       "{#FSTYPE}":"sysfs"    },
    { "{#FSNAME}":"/proc",                      "{#FSTYPE}":"proc"     },
    { "{#FSNAME}":"/dev",                       "{#FSTYPE}":"devtmpfs" },
    { "{#FSNAME}":"/dev/pts",                   "{#FSTYPE}":"devpts"   },
    { "{#FSNAME}":"/lib/init/rw",               "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/dev/shm",                   "{#FSTYPE}":"tmpfs"    },
    { "{#FSNAME}":"/home",                      "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/tmp",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/usr",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/var",                       "{#FSTYPE}":"ext3"     },
    { "{#FSNAME}":"/sys/fs/fuse/connections",   "{#FSTYPE}":"fusectl"  }
]
```

前の例では、キーがプロトタイプで使用される LLD マクロ名と一致している必要があります。別の方法として、JSONPath を使用して LLD マクロ値を {#FSNAME} → $.fsname および {#FSTYPE} → $.fstype のように抽出することもでき、その場合は次のよう

なスクリプトが可能です。

```
####!/usr/bin/perl

$first = 1;

print "[\n";

for (`cat /proc/mounts`)
{
    ($fsname, $fstype) = m/\S+ (\S+) (\S+)/;

    print "\t,\n" if not $first;
    $first = 0;

    print "\t{\n";
    print "\t\t\"fsname\":\"$fsname\",\n";
    print "\t\t\"fstype\":\"$fstype\"\n";
    print "\t}\n";
}

print "]\n";
```

その出力例（わかりやすさのために整形済み）を以下に示します。

```
[
    { "fsname":"/",                          "fstype":"rootfs"   },
    { "fsname":"/sys",                       "fstype":"sysfs"    },
    { "fsname":"/proc",                      "fstype":"proc"     },
    { "fsname":"/dev",                       "fstype":"devtmpfs" },
    { "fsname":"/dev/pts",                   "fstype":"devpts"   },
    { "fsname":"/lib/init/rw",               "fstype":"tmpfs"    },
    { "fsname":"/dev/shm",                   "fstype":"tmpfs"    },
    { "fsname":"/home",                      "fstype":"ext3"     },
    { "fsname":"/tmp",                       "fstype":"ext3"     },
    { "fsname":"/usr",                       "fstype":"ext3"     },
    { "fsname":"/var",                       "fstype":"ext3"     },
    { "fsname":"/sys/fs/fuse/connections",   "fstype":"fusectl"  }
]
```

その後、ディスカバリルールの *Filter* フィールドで、マクロとして”{#FSTYPE}”、正規表現として”rootfs|ext3” を指定できます。

Note:
カスタム LLD ルールでは、FSNAME/FSTYPE のようなマクロ名を使用する必要はありません。任意の名前を自由に使用できます。JSONPath を使用する場合、LLD 行はオブジェクトである配列要素になりますが、別の配列や値にすることもできます。

ユーザーパラメータを使用する場合、戻り値は 16MB に制限されることに注意してください。詳細は、LLD 戻り値のデータ制限 を参照してください。

16. 分散監視

概要　Zabbix は、Zabbixプロキシを使用して分散 IT インフラストラクチャを監視する効果的で信頼性の高い方法を提供します。

プロキシは一元化された Zabbix サーバーに代わってローカルでデータを収集し、サーバーにデータを報告することができます。

プロキシの機能

プロキシを使用するかどうかを選択する際には、いくつかの考慮事項を考慮する必要があります。

	プロキシ
軽量	はい
GUI	いいえ

	プロキシ
独立して動作	はい
保守が容易	はい
DB の自動作成	はい[1]
ローカル管理	いいえ
組み込みハードウェア対応	はい
一方向の *TCP* 接続	はい
設定の一元管理	はい
通知を生成	いいえ

[1] DB の自動作成機能は SQLite でのみ動作します。その他のサポートされているデータベースでは、手動セットアップが必要です。

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

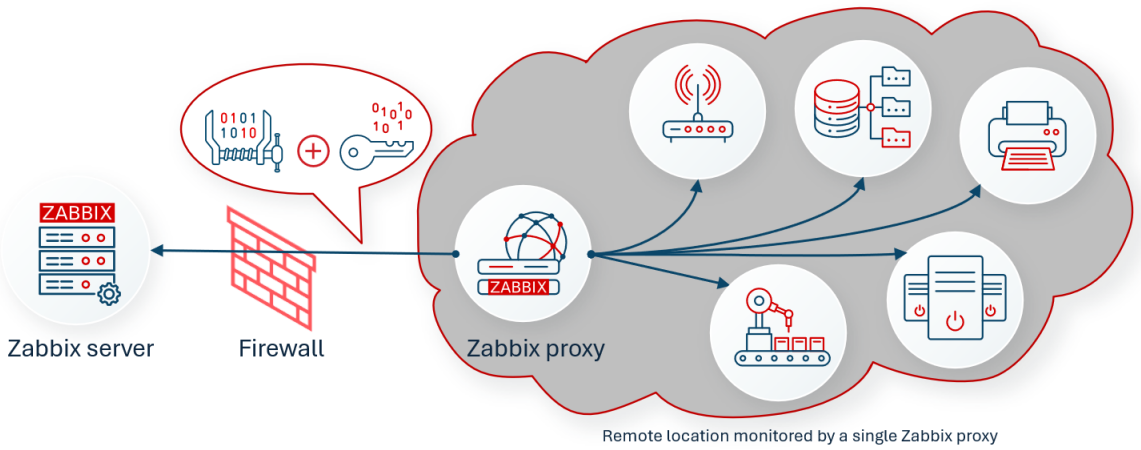
1 プロキシ

概要　Zabbix プロキシは、Zabbix サーバーに代わってパフォーマンスデータおよび可用性データを収集できます。これにより、プロキシがデータ収集の負荷の一部を担い、Zabbix サーバーの負荷を軽減できます。

また、プロキシを使用することは、集中監視および分散監視を実装する最も簡単な方法です。この場合、すべてのエージェントとプロキシが 1 台の Zabbix サーバーに報告し、すべてのデータが一元的に収集されます。

Zabbix プロキシは、次の用途に使用できます。

- リモート拠点の監視
- 通信が不安定な拠点の監視
- 数千台のデバイスを監視する際の Zabbix サーバーの負荷軽減
- 分散監視の保守の簡素化



プロキシが必要とする Zabbix サーバーへの TCP 接続は 1 つだけです。そのため、設定が必要なファイアウォールルールは 1 つだけで済み、ファイアウォールを通過させやすくなります。

Attention:
Zabbix プロキシは、必ず専用のデータベースを使用する必要があります。Zabbix サーバーのデータベースを指定すると、設定が破損します。

プロキシによって収集されたすべてのデータは、サーバーへ送信される前にローカルに保存されます。これにより、サーバーとの一時的な通信障害が発生しても、データが失われることはありません。プロキシ設定ファイルの *ProxyLocalBuffer* および *ProxyOfflineBuffer* パラメータで、データをローカルに保持する期間を制御します。

Attention:
Zabbix サーバーのデータベースから最新の設定変更を直接受け取るプロキシは、CacheUpdateFrequency の値により設定の更新がそれほど速く行われない Zabbix サーバーよりも、新しい設定を保持している場合があります。その結果、プロキシがデータ収集を開始して Zabbix サーバーへ送信しても、Zabbix サーバーがそのデータを無視することがあります。

Zabbix プロキシはデータコレクターです。トリガーの計算、イベントの処理、アラートの送信は行いません。プロキシの機能概要については、次の表を参照してください。

機能		プロキシでサポート
アイテム		
	Zabbix エージェントチェック	はい
	Zabbix エージェントチェック（アクティブ）	はい [1]
	シンプルチェック	はい
	トラッパーアイテム	はい
	SNMP チェック	はい
	SNMP トラップ	はい
	IPMI チェック	はい
	JMX チェック	はい
	ログファイル監視	はい
	内部チェック	はい
	SSH チェック	はい
	Telnet チェック	はい
	外部チェック	はい
	依存アイテム	はい
	スクリプトアイテム	はい
	ブラウザアイテム	はい
組み込み Web 監視		はい
アイテム値の前処理		はい
ネットワークディスカバリ		はい
アクティブエージェントの自動登録		はい
ローレベルディスカバリ		はい [2]
リモートコマンド		はい
トリガーの計算		いいえ
イベントの処理		いいえ
イベント相関		いいえ
アラートの送信		いいえ

Note:
[1] エージェントがアクティブチェックを取得する際に、サーバーではなくプロキシへ問い合わせるようにするには、エージェント設定ファイルの **ServerActive** パラメータにプロキシを指定する必要があります。
 [2] LLD については、Zabbix プロキシはデータの収集と前処理のみを行い、その後の処理のために Zabbix サーバーへ送信します。

過負荷からの保護

Zabbix サーバーがしばらく停止していて、プロキシが大量のデータを収集していた場合、
その後サーバーが起動すると、過負荷になる可能性があります（履歴キャッシュ使用率がしばらく 95～100% のままになります）。
この過負荷により、本来よりもチェックの処理が遅くなるなど、パフォーマンス低下が発生する可能性があります。
このようなシナリオに対する保護機能は、履歴キャッシュの過負荷によって発生する問題を回避するために実装されました。

Zabbix サーバーの履歴キャッシュがいっぱいになると、履歴キャッシュへの書き込みアクセスが
スロットリングされ、サーバーのデータ収集プロセスが停止します。履歴キャッシュの過負荷が
最もよく発生するのは、サーバー停止後にプロキシが収集済みデータをアップロードする場合です。
これを回避するため、プロキシのスロットリングが追加されました（現在は無効化できません）。

Zabbix サーバーは、履歴キャッシュ使用率が 80% に達すると、プロキシからのデータ受信を停止します。
代わりに、それらのプロキシはスロットリングリストに追加されます。これは、キャッシュ使用率が
60% まで低下するまで継続されます。その後、サーバーはスロットリングリストに従って、プロキシからの
データ受信を 1 台ずつ再開します。つまり、スロットリング期間中に最初にデータのアップロードを試みた
プロキシが最初に処理され、その処理が完了するまで、サーバーは他のプロキシからのデータを受信しません。

このスロットリングモードは、キャッシュ使用率が再び 80% に達するか、20% まで低下するか、
またはスロットリングリストが空になるまで継続されます。最初のケースでは、サーバーは再び

プロキシデータの受信を停止します。後者の 2 つのケースでは、サーバーは通常動作に戻り、
すべてのプロキシからのデータを受信するようになります。

上記の情報は、次の表で示すことができます。

履歴書き込みキャッシュ使用率	Zabbix サーバーのモード	Zabbix サーバーの動作
80% に達する	待機	プロキシデータの受信を停止しますが、スロットリングリスト（後で接続するプロキシの優先
60% まで低下する	スロットリング中	スロットリングリストの処理を開始しますが、引き続きプロキシデータは受信しません。
20% まで低下する	通常	スロットリングリストを破棄し、通常どおりプロキシデータの受信を開始します。

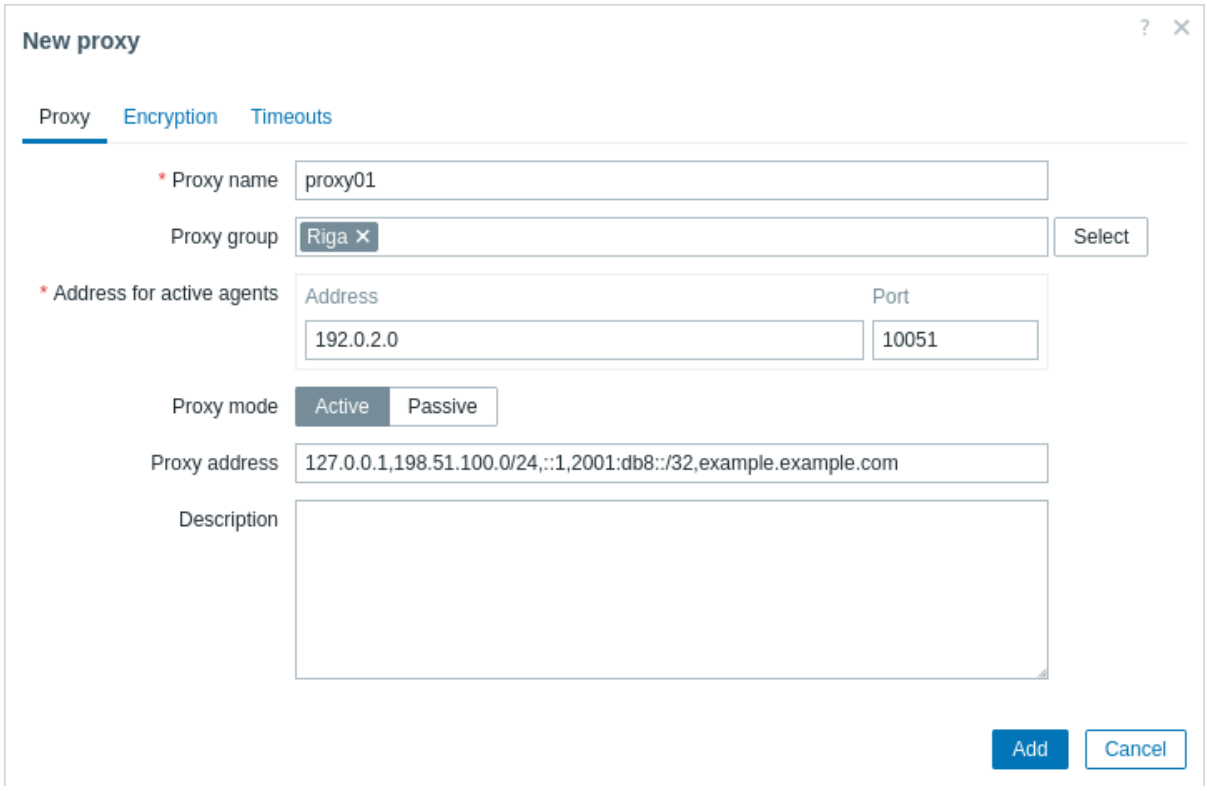
この Zabbix サーバーの動作は、zabbix[wcache,history,pused] 内部アイテムを使用してメトリクスと関連付けることができます。

設定　Zabbix プロキシサーバーをインストールして設定したのち、Zabbix フロントエンドでプロキシの設定をします。

プロキシの追加

Zabbix フロントエンドでプロキシを設定するには:

- 管理 → プロキシに移動してください。
- プロキシの作成をクリックしてください。



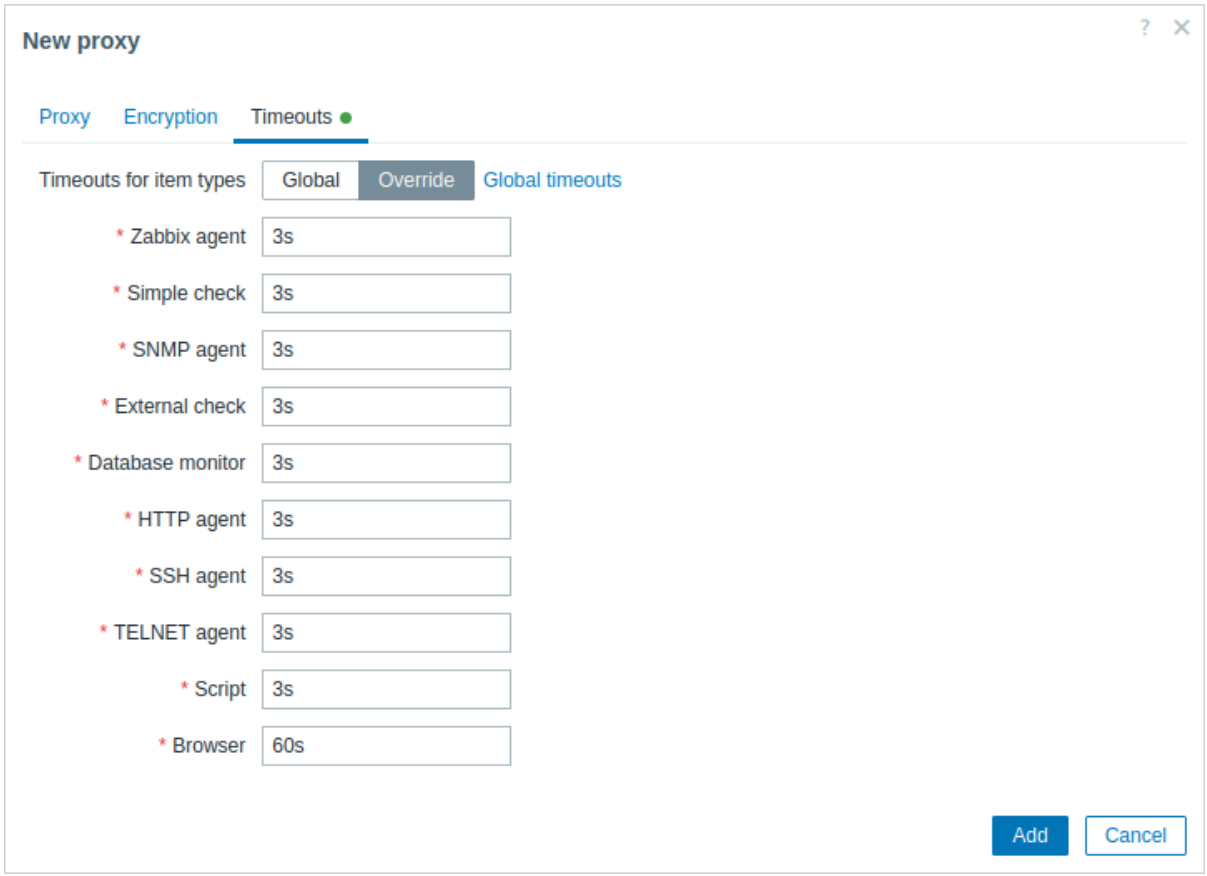
パラメーター	説明
プロキシグループ	ロードバランシング/高可用性のプロキシのプロキシグループを選択します。 選択できるグループは 1 つだけです。

パラメーター		説明
アクティブエージェントのアドレス		監視されているアクティブなエージェントまたは送信者が接続する必要があるアドレスを入力します。Zabbix 7.0 エージェント以降のみサポート。 このアドレスは、アクティブプロキシとパッシブプロキシの両方に接続するために使用されます。このフィールドは、プロキシグループフィールドでプロキシグループが選択されている場合にのみ使用できます。
プロキシ名	*IP* アドレス	接続する IP アドレス/DNS 名。
	ポート	接続する TCP ポート番号 (デフォルト: 10051)。ユーザーマクロがサポートされています。
プロキシモード		プロキシ名を入力します。プロキシ設定ファイルの *Hostname* パラメーターと同じ名前である必要があります。 プロキシモードを選択します。 アクティブ - プロキシが Zabbix サーバーに接続し、設定データを要求します。 パッシブ - Zabbix サーバーがプロキシに接続します。 注意: 暗号化されていない (機密性の高い) プロキシ設定データは、アクティブなプロキシを使用しているときに Zabbix サーバーのトラッパーポートにアクセスできる関係者が利用できるようになる可能性があります。これが可能なのは、認証が行われない場合、またはプロキシアドレスがプロキシのアドレスフィールドで制限されていない場合、誰でもアクティブなプロキシのふりをして設定データを要求する可能性があるためです。
プロキシのアドレス		指定した場合、アクティブなプロキシリクエストは、このコンマ区切りの IP アドレスのリスト (オプションで CIDR 表記またはアクティブな Zabbix プロキシの DNS 名) からのみ受け入れられます。 このフィールドはプロキシモードフィールドでアクティブが選択されている場合にのみ使用できます。マクロはサポートされていません。
インターフェース		パッシブプロキシのインターフェース詳細を入力します。 このフィールドはプロキシモードフィールドでパッシブが選択されている場合にのみ使用できます。
	IP アドレス	パッシブプロキシの IP アドレス/DNS 名。
	ポート	パッシブプロキシの TCP ポート番号 (デフォルト: 10051)。ユーザーマクロがサポートされています。
説明		プロキシの説明を入力します。

暗号化タブではプロキシとの暗号化された接続を設定できます。

パラメーター	説明
プロキシへの接続	サーバーがパッシブプロキシに接続する方法: 暗号化なし (デフォルト)、PSK(事前共有キー)、証明書。
プロキシからの接続	アクティブなプロキシから許可される接続のタイプを選択します。複数の接続タイプを同時に選択できます (テストおよび他の接続タイプへの切り替えに役立ちます)。デフォルトは” 暗号なし” です。
発行者	証明書発行者。証明書は最初に CA(認証局) で検証されます。CA によって署名された有効な証明書は、発行者フィールドを使用して許可される CA をさらに制限できます。このフィールドはオプションであり Zabbix インストールで複数の CA からの証明書を使用する場合に使用することを目的としています。
サブジェクト	許可された証明書名。証明書は最初に CA で検証されます。CA によって署名された有効な証明書はサブジェクトフィールドを使用して、サブジェクト文字列の値を 1 つだけ許可できます。このフィールドが空の場合、構成された CA によって署名された有効な証明書が受け入れられます。
PSK アイデンティティ	事前共有キー ID 文字列 機密情報を PSKID に入れないでください。機密情報は暗号化されずにネットワーク経由で送信され、使用する PSK を受信者に通知します。
PSK	事前共有キー (16 進文字列)。最大長: Zabbix が GnuTLS または OpenSSL ライブラリを使用する場合は 512 hex-digits (256 バイト PSK)、Zabbix が mbed TLS (PolarSSL) ライブラリを使用する場合は 64 hex-digits (32 バイト PSK)。例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

タイムアウトタブを使用すると、サポートするアイテムタイプのグローバルタイムアウトをオーバーライドできます。



パラメーター	説明
アイテムタイプのタイムアウト	Select the timeout option: **Global** - global timeout is used (displayed in the grayed out *Timeout* field for each item type); **Override** - custom timeout is used (set in the *Timeout* field for each item type). Allowed range: 1 - 600s (default: inherited from global timeouts). Time suffixes, e.g. 30s, 1m, and user macros are supported. Clicking the *Global timeouts* link allows you to configure global timeouts. Note that the *Global timeouts* link is visible only to users of *Super admin* type with permissions to the *Administration* → *General* frontend section. Supported item types: - Zabbix agent (both passive and active checks) - Simple check (except `icmpping*`, `vmware.*` items) - SNMP agent (only for SNMP `walk[OID]` and `get[OID]` items) - External check - Database monitor - HTTP agent - SSH agent - TELNET agent - Script - Browser Note that the timeouts set under **Override** will prevail over the global ones but will be overridden by individual item timeouts if those are set under item configuration.

Note:

If proxy major version does not match server major version, icon will be displayednext to *Timeouts for item types*, with the hover message"Timeouts disabled because the proxy and server versions do not match".In such cases, the proxy will use the `Timeout` parameter from the proxy configuration file.

既存のプロキシの編集フォームには、次の追加ボタンがあります。

- *Refresh configuration* - プロキシの設定を更新します。
- *Clone* - 既存のプロキシのプロパティに基づいて新しいプロキシを作成します。
- *Delete* - プロキシを削除します。

ホスト設定

個々のホストをプロキシまたはプロキシグループで監視するように指定するには、ホスト設定フォームの 監視方法フィールドを使用します。



ホストの一括更新は、ホストをプロキシまたはプロキシグループで監視するように指定するもう 1 つの方法です。

1 監視設定の同期

概要

このページでは、プロキシの監視設定更新、つまりサーバー上で監視設定に加えられた変更がどのようにプロキシへ同期されるかについて詳しく説明します。

増分更新

プロキシの設定更新は増分方式です。設定同期中は、変更されたエンティティのみが更新されます（したがって、変更されたエンティティがない場合は何も送信されません）。このアプローチにより、リソースを節約し、プロキシ設定更新の間隔をより短く（ほぼ即時に）設定できます。

プロキシ設定の変更は、リビジョン番号を使用して追跡されます。プロキシ設定のリビジョンより大きいリビジョンを持つエンティティのみが、プロキシに送信される設定データに含まれます。

設定同期の対象となるエンティティは次のとおりです。

Entity	Details
自動登録 *TLS* データ	すべての自動登録 TLS データ。
式	すべての式（正規表現、式テーブル）。
グローバル設定	'config' テーブルで定義されたグローバル設定
ホスト	ホストのすべてのプロパティ、インターフェース、インベントリ、アイテム、アイテムの前処理、アイテムパラ
ホストマクロ	ホスト上で定義されたすべてのマクロと、それにリンクされたすべてのテンプレート ID。
プロキシディスカバリルール	プロキシに割り当てられたディスカバリルールとチェック。

つまり、次のことを意味します。

- ホスト上のアイテムが変更された場合、そのホストのすべての設定が同期されます。
- 正規表現が変更された場合、すべての正規表現が同期されます。

例外として、ホストマクロは、ホスト上の何かが変更された場合にも送信されます。

プロキシ上で `-R config_cache_reload` コマンドを実行した場合も、増分更新が開始されます。

なお、プロキシの起動/再起動時、HA フェイルオーバー時、セッショントークンが変更された場合、またはプロキシで設定更新に失敗した場合（たとえば、設定データの受信中に接続が切断された場合）には、完全な設定同期が実行されます。

設定パラメータ

ProxyConfigFrequency パラメータは、プロキシ設定をサーバーと同期する頻度を決定します（デフォルトでは 10 秒）。

ProxyConfigFrequency は次のとおりであることに注意してください。

- パッシブプロキシではサーバーのパラメータ
- アクティブプロキシではプロキシのパラメータ

アクティブプロキシでは、ProxyConfigFrequency は Zabbix 6.4 以降の新しいパラメータであり、現在は非推奨となっている ConfigFrequency の代わりに使用する必要があります。

Attention:
ProxyConfigFrequency と ConfigFrequency の両方が使用されている場合、プロキシはエラーをログに記録して終了します。

メンテナンス中のキューの計算

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

2 プロキシの負荷分散と高可用性

概要

Zabbix プロキシは、プロキシの負荷分散と高可用性を実現するために、プロキシグループに編成できます。

プロキシの負荷分散と高可用性とは、プロキシグループ内のプロキシ間でホストを自動的に再分配することです。

- プロキシがオフラインになると、そのホストは他のプロキシに移動され、プロキシの高可用性が維持されます。
- あるプロキシのホスト数が他のプロキシと比べて大幅に多い、または少ない場合、そのホストは他のプロキシに移動され、プロキシの負荷が分散されます。

ホストの再分配は、グループ内のプロキシ間で、次の条件を満たす場合にのみ機能します。

- プロキシが Zabbix 7.0 以降で稼働していること。
- プロキシのバージョン が Zabbix サーバーのバージョンと一致していること。Zabbix エージェント（パッシブ）を使用する場合、プロキシのバージョンはエージェントのバージョンと一致している必要があります。アクティブエージェントでは、Zabbix 7.0 以降のみが必要です。
- プロキシグループがオンライン状態 であること。
- ホストが、個別のプロキシではなく、プロキシグループによって監視されるように設定されていること。

Note:
プロキシグループの健全性は、プロキシグループに割り当てられた任意のホストによる内部チェックで監視できます。
ただし、グループ内の単一のプロキシの健全性を監視するには、そのホストをそのプロキシに割り当ててください。そうしないと、結果に一貫性がなくなる可能性があります。

ホストの再分配

プロキシの負荷分散と高可用性は、proxy group manager を介して Zabbix サーバーによって管理されます。これは、各プロキシグループ内のすべてのプロキシの状態と、それらのホスト分散を継続的に監視します。

グループ内のプロキシ高可用性は、プロキシフェイルオーバーによって確保されます。プロキシがオフラインになると、そのホストは直ちに他のプロキシへ再分配されます。
プロキシの負荷分散も行われ、割り当てられているホスト数が最も少ないプロキシにホストが再割り当てされます。

さらに、プロキシのホスト数がグループ平均と比べて少なくとも 10 ホスト、かつ 2 倍の差（ホスト過多またはホスト不足）がある場合にも、プロキシの負荷分散がトリガーされます。
猶予期間（10 x failover delay）の後も不均衡が継続する場合、そのプロキシグループはホスト再分配の対象としてキューに入れられます。

proxy group manager は、次のロジックを使用してホストを再分配します。

1. プロキシごとの平均ホスト数を計算します。
2. ホスト過多のプロキシについては、超過分のホストを未割り当てプロキシのプールに移動します。
3. ホスト不足のプロキシについては、均衡に達するために必要なホスト数を計算します。
4. 最も多くのホストを持つプロキシから、必要数のホストを取り外します。
5. 未割り当てのホストを、ホスト数が最も少ないプロキシへ移動します。

ホスト再分配の例:

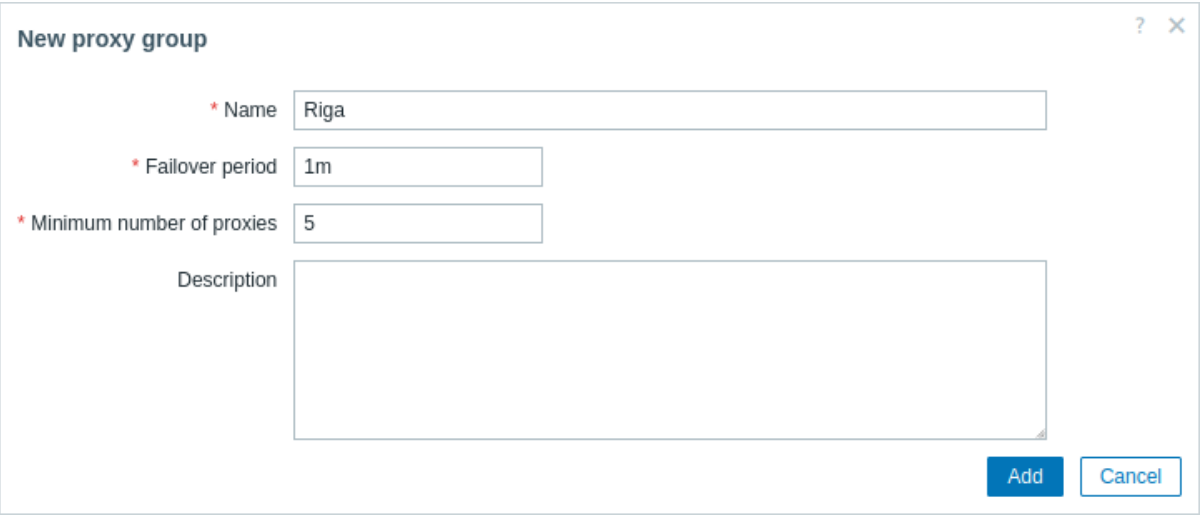
プロキシ上のホスト数	グループ平均	ホスト再割り当て
100	50	はい
60	50	いいえ
40	50	いいえ
25	50	はい
15	5	はい
10	5	いいえ

Attention:
プロキシグループによって監視されるホスト数が 10 未満の場合、グループ内のプロキシ間でホスト分散が不均一になる可能性があります。

プロキシグループの設定

Zabbix Web インターフェースでプロキシグループを設定するには、次の手順を実行します。

1. *Administration* > *Proxy groups* に移動します
2. *Create proxy group* をクリックします



Parameter	Description
Name	プロキシグループの名前。

Parameter	Description
Failover period	プロキシグループ内のプロキシがオンラインと見なされるために、Zabbix サーバーと通信している必要がある期間（秒単位）（デフォルト: 1m、範囲: 10s–15m）。この期間内にプロキシが通信しない場合、プロキシの状態は *Offline* に変更され、そのホストは直ちに他のプロキシに再分散されます。プロキシの負荷分散は、この期間の 10 倍経過後に開始されます。 時間の接尾辞（例: 30s、1m）およびユーザーマクロをサポートします。
Minimum number of proxies	プロキシグループをオンライン に維持するために必要な最小のオンラインプロキシ数（デフォルト: 1、範囲: 1–1000）。 ユーザーマクロをサポートします。 この値は、グループ内のプロキシ総数より少なくする必要があります。例えば、10 台のプロキシからなるグループで最小値を 10 に設定すると、いずれか 1 台のプロキシが障害を起こした時点でグループはオフラインになります。オフラインのグループ内にあるオンラインプロキシは通常どおり動作し続けますが、負荷分散/高可用性は実行されないことに注意してください。
Description	プロキシグループの説明。
Proxies	プロキシが含まれるグループを編集する際に、最大 5 つのプロキシの一覧を表示します（プロキシに対するユーザー権限に応じて、リンクまたはプレーンテキストで表示）。

プロキシの負荷分散の設定

プロキシの負荷分散を使用するには、Zabbix Web インターフェースでプロキシグループを設定し（上記参照）、ホストが個別のプロキシではなくプロキシグループによって監視されるようにしてください（ホストをプロキシからプロキシグループへ移動するには、ホストの一括更新を使用できます）。

Zabbix エージェントを使用する場合は、以下のようにも設定してください。

- パッシブチェックの場合、Serverパラメータに、プロキシグループ内のすべてのプロキシを列挙します。
- アクティブチェックの場合、ServerActiveパラメータに、プロキシグループ内のすべてのプロキシ、または Zabbix サーバーを列挙することを推奨します。プロキシグループをアクティブモードで使用できるのは、Zabbix エージェント 7.0 以降のみであることに注意してください。

`ServerActive` パラメータにプロキシグループ内の 1 つのプロキシ（または Zabbix サーバー）しか含まれていない場合でも、エージェントは正しいプロキシに接続できます。エージェントサービスの起動時に指定されたプロキシへ接続すると、エージェントはグループ内のプロキシ IP の完全な一覧と、それぞれの現在の負荷を受信してキャッシュします。その後、アクティブチェックは、プロキシグループ内における現在のプロキシ-ホスト割り当てに基づいて、そのホストに対応する正しいオンラインプロキシへリダイレクトされます。

Warning:
Zabbix エージェントの `ServerActive` パラメータに 1 つのプロキシしか指定していない場合、指定したプロキシがオフラインのときにエージェントが起動または再起動されると、監視データが失われる可能性があります。

Note:
Zabbix sender を使用する場合も、データ要求は、プロキシグループ内における現在のプロキシ-ホスト割り当てに基づいて、そのホストに対応する正しいオンラインプロキシへリダイレクトされます。ただし、入力ファイルから複数ホストの値を送信する場合は、誤ったプロキシにデータが送信されるのを防ぐため、`-g` オプションを使用してください。

また、Zabbix エージェントは、ファイアウォールを通じてプロキシグループ内のすべてのプロキシに接続できる必要があります。そうでない場合、アクティブチェックはリダイレクトまたはフェイルオーバー中にハングしたり失敗したりする可能性があります。例:

- アクティブチェック中に、あるプロキシがエージェントを別のプロキシへリダイレクトすることがあります。そのプロキシがファイアウォールでブロックされている場合、応答待ちの間、通信がハングします。
- 最近リバランスが行われていない安定した高可用性構成では、エージェントがバックアッププロキシに一度も接続しないことがあります。ファイアウォールルールが変更されていて、かつテストされていない場合、フェイルオーバーに失敗する可能性があります。

プロキシの負荷分散のテスト

プロキシの負荷分散をテストするには、次の手順を実行します。

1. プロキシグループを設定します。
2. プロキシグループがオンライン状態であることを確認します。
3. ホストが個別のプロキシではなく、プロキシグループによって監視されていることを確認します（ホストをプロキシからプロキシグループへ移動するには、ホストの一括更新を使用できます）。
4. 設定の更新と、プロキシグループ内のプロキシ間でのホストの分散が完了するまで数秒待ちます。*Administration* > *Proxies* でホスト一覧を更新して、変更を確認します。

重要な注意事項

- プロキシグループ内のプロキシでは、SNMP トラップはサポートされていません。
- 外部設定に依存するチェック（例: 外部チェック用のスクリプトや、データベースチェック用の ODBC 設定）では、プロキシグループ内のすべてのプロキシで同一の設定が必要です。
- データベースチェックでは、データベースオブジェクト/サーバーに対する拡張権限が必要です。
- プロキシグループによって監視されるVMware ホストは、グループ内のプロキシ間でランダムに分散されます。そのため、各プロキシがすべての VMware データをキャッシュすることになり、vCenter に追加の負荷が発生します。
- プロキシグループ内のプロキシからの自動登録データに基づいて作成されたホストは、そのプロキシグループによって監視されるように設定されます。一方、プロキシグループ内のプロキシからのネットワークディスカバリデータに基づいて作成されたホストは、そのプロキシによって監視されるように設定されます。

17. 暗号化

概要　Zabbix は、Transport Layer Security（TLS）プロトコル v.1.2 および 1.3（暗号ライブラリに依存）を使用して、Zabbix コンポーネント間の暗号化通信をサポートしています。証明書ベースおよび事前共有鍵ベースの暗号化がサポートされています。

暗号化は、次の接続に対して設定できます。

- Zabbix サーバー、Zabbix プロキシ、Zabbix エージェント、Zabbix Web サービス、zabbix_sender および zabbix_get ユーティリティ間
- Zabbix データベースへの接続Zabbix Web インターフェースおよびサーバー/プロキシから
- Zabbix Web インターフェースと Zabbix サーバー間

暗号化はオプションであり、個々のコンポーネントごとに設定できます。

- 一部のプロキシおよびエージェントは、サーバーとの通信に証明書ベースの暗号化を使用するよう設定でき、他のものは事前共有鍵ベースの暗号化を使用でき、さらに別のものは暗号化されていない通信（従来どおり）を継続できます。
- サーバー（プロキシ）は、異なるホストに対して異なる暗号化設定を使用できます。

Zabbix デーモンプログラムは、暗号化された受信接続と暗号化されていない受信接続の両方に対して、1 つの待受ポートを使用します。暗号化を追加しても、ファイアウォールで新しいポートを開放する必要はありません。

制限事項

- 秘密鍵は、起動時に Zabbix コンポーネントが読み取り可能なファイルに平文で保存されます。
- 事前共有鍵は Zabbix Web インターフェースで入力され、Zabbix データベースに平文で保存されます。
- 組み込みの暗号化では、Zabbix Web インターフェースを実行している Web サーバーとユーザーの Web ブラウザー間の通信は保護されません。
- 現在、暗号化された各接続は完全な TLS ハンドシェイクで開始され、セッションキャッシュおよびチケットは実装されていません。
- 暗号化を追加すると、ネットワーク遅延に応じてアイテムのチェックおよびアクションにかかる時間が増加します。
  - たとえば、パケット遅延が 100ms の場合、TCP 接続を開いて暗号化されていないリクエストを送信するには約 200ms かかります。暗号化を使用すると、TLS 接続の確立に約 1000ms が追加されます。
  - タイムアウト値を増やす必要がある場合があります。そうしないと、エージェント上でリモートスクリプトを実行する一部のアイテムやアクションは、暗号化されていない接続では動作しても、暗号化された接続ではタイムアウトして失敗する可能性があります。
- 暗号化はネットワークディスカバリではサポートされていません。ネットワークディスカバリによって実行される Zabbix エージェントチェックは暗号化されず、Zabbix エージェントが暗号化されていない接続を拒否するよう設定されている場合、そのようなチェックは成功しません。

暗号化サポート付きでの **Zabbix** のコンパイル　暗号化をサポートするには、Zabbix をサポート対象の暗号ライブラリのいずれかとともにコンパイルおよびリンクする必要があります。

- GnuTLS - バージョン 3.1.18 以降
- OpenSSL - バージョン 1.0.1、1.0.2、1.1.0、1.1.1、3.0.x
- LibreSSL - バージョン 2.7.4、2.8.2 でテスト済み:
  - LibreSSL 2.6.x はサポートされていません
  - LibreSSL は OpenSSL の互換置換としてサポートされています; 新しい `tls_*()` の LibreSSL 固有 API 関数は使用されません。LibreSSL でコンパイルされた Zabbix コンポーネントでは PSK を使用できず、証明書のみ使用できます。

Note:
Zabbix Web インターフェース向けの SSL 設定については、こちらのベストプラクティスを参照してください。

ライブラリは、"configure" スクリプトに対応するオプションを指定して選択します:

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (LibreSSL にも使用)

たとえば、*OpenSSL* を使用してサーバーおよびエージェント向けにソースを設定するには、次のようにします:

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

異なる Zabbix コンポーネントを異なる暗号ライブラリでコンパイルすることもできます（例: サーバーは *OpenSSL*、エージェントは *GnuTLS*）。

Attention:
事前共有鍵（PSK）を使用する予定がある場合は、PSK を使用する Zabbix コンポーネントでは *GnuTLS* または *OpenSSL 1.1.0*（またはそれ以降）のライブラリの使用を検討してください。*GnuTLS* および *OpenSSL 1.1.0* ライブラリは、Perfect Forward Secrecy を備えた PSK 暗号スイートをサポートしています。古いバージョンの *OpenSSL* ライブラリ（1.0.1、1.0.2c）も PSK をサポートしていますが、利用可能な PSK 暗号スイートでは Perfect Forward Secrecy は提供されません。

接続の暗号化管理　Zabbix の接続では、以下を使用できます。

- 暗号化なし（デフォルト）
- RSA 証明書ベースの暗号化
- PSK ベースの暗号化

Zabbix コンポーネント間の暗号化を指定するために使用される重要なパラメータは 2 つあります。

- TLSConnect - 送信接続に使用する暗号化方式を指定します（暗号化なし、PSK、または証明書）
- TLSAccept - 受信接続で許可する接続タイプを指定します（暗号化なし、PSK、または証明書）。1 つ以上の値を指定できます。

`TLSConnect` は、Zabbix プロキシの設定ファイル（アクティブモードでは、サーバーへの接続のみを指定）および Zabbix エージェントの設定ファイル（アクティブチェック用）で使用されます。Zabbix Web インターフェースでは、TLSConnect に相当するものは、データ収集 → ホスト → *<some host>* → 暗号化タブの ホストへの接続フィールド、および 管理 → プロキシ → *<some proxy>* → 暗号化タブの プロキシへの接続フィールドです。設定された接続の暗号化タイプで接続に失敗した場合、他の暗号化タイプは試行されません。

`TLSAccept` は、Zabbix プロキシの設定ファイル（パッシブモードでは、サーバーからの接続のみを指定）および Zabbix エージェントの設定ファイル（パッシブチェック用）で使用されます。Zabbix Web インターフェースでは、TLSAccept に相当するものは、データ収集 → ホスト → *<some host>* → 暗号化タブの ホストからの接続フィールド、および 管理 → プロキシ → *<some proxy>* → 暗号化タブの プロキシからの接続フィールドです。

通常、受信接続に対して設定する暗号化タイプは 1 つだけです。ただし、たとえば停止時間を最小限に抑え、ロールバックの可能性を確保しながら、暗号化なしから証明書ベースへ暗号化タイプを切り替えたい場合があります。これを実現するには、次のようにします。

- エージェント設定ファイルで `TLSAccept=unencrypted,cert` を設定し、Zabbix エージェントを再起動します
- 証明書を使用して zabbix_get でエージェントへの接続をテストします。動作する場合は、Zabbix Web インターフェースの データ収集 → ホスト → *<some host>* → 暗号化タブで、そのエージェントの暗号化設定を再構成し、ホストへの接続を「証明書」に設定できます。
- サーバーの設定キャッシュが更新されると（そのホストがプロキシによって監視されている場合は、プロキシ設定も更新されると）、そのエージェントへの接続は暗号化されます
- すべてが期待どおりに動作する場合は、エージェント設定ファイルで `TLSAccept=cert` を設定し、Zabbix エージェントを再起動します。これでエージェントは、証明書ベースで暗号化された接続のみを受け付けます。暗号化なしおよび PSK ベースの接続は拒否されます。

同様に、サーバーおよびプロキシでも機能します。Zabbix Web インターフェースのホスト設定で ホストからの接続が「証明書」に設定されている場合、エージェント（アクティブチェック）および zabbix_sender（トラッパーアイテム）からは、証明書ベースで暗号化された接続のみが受け付けられます。

多くの場合、受信接続と送信接続には同じ暗号化タイプ、または暗号化なしを設定することになるでしょう。ただし技術的には、たとえば受信接続には証明書ベースの暗号化、送信接続には PSK ベースの暗号化というように、非対称に設定することも可能です。

各ホストの暗号化設定は、Zabbix Web インターフェースの データ収集 → ホストにある エージェントの暗号化列に表示されます。例:

例	ホストへの接続	ホストからの許可される接続	ホストからの拒否
NONE	暗号化なし	暗号化なし	暗号化された接続、
CERT NONE PSK CERT	暗号化、証明書ベース	暗号化、証明書ベース	暗号化なし、およ
PSK NONE PSK CERT	暗号化、PSK ベース	暗号化、PSK ベース	暗号化なし、およ

例	ホストへの接続	ホストからの許可される接続	ホストからの拒否
PSK NONE PSK CERT	暗号化、PSK ベース	暗号化なし、および PSK ベースの暗号化	証明書ベースの暗号
CERT NONE PSK CERT	暗号化、証明書ベース	暗号化なし、PSK ベース、または証明書ベースの暗号化	-

Attention:
接続はデフォルトで暗号化されません。暗号化は各ホストおよびプロキシごとに個別に設定する必要があります。

zabbix_get と **zabbix_sender** の暗号化機能　zabbix_get とzabbix_sender manpages のマニュアルを参照してください。

暗号スイート　暗号スイートは、デフォルトでは Zabbix の起動時に内部的に設定されます。

GnuTLS および OpenSSL では、ユーザー設定の暗号スイートもサポートされています。ユーザーは、各自のセキュリティポリシーに従って暗号スイートを設定できます。この機能の使用は任意です（組み込みのデフォルト暗号スイートも引き続き使用できます）。

デフォルト設定でコンパイルされた暗号ライブラリでは、Zabbix の組み込みルールにより、通常は次の暗号スイートになります（優先度の高い順）：

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
OpenSSL 1.1.1d	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

ユーザー設定の暗号スイート　組み込みの暗号スイート選択基準は、ユーザー設定の暗号スイートで上書きできます。

Attention:
ユーザー設定の暗号スイートは、TLS 暗号スイート、その安全性、設定ミスの影響を理解し、TLS のトラブルシューティングに慣れている上級ユーザー向けの機能です。

組み込みの暗号スイート選択基準は、以下のパラメータを使用して上書きできます。

上書きの範囲	パラメータ	値	説明
証明書用の暗号スイート選択	TLSCipherCert13	TLS 1.3 プロトコル用の有効な OpenSSL 1.1.1 のcipher strings（その値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 用の証明書ベースの暗号スイート選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherCert	TLS 1.2 用の有効な OpenSSL のcipher strings、または有効な GnuTLS のpriority strings。それぞれの値は `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）用の証明書ベースの暗号スイート選択基準
PSK 用の暗号スイート選択	TLSCipherPSK13	TLS 1.3 プロトコル用の有効な OpenSSL 1.1.1 のcipher strings（その値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 用の PSK ベースの暗号スイート選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherPSK	TLS 1.2 用の有効な OpenSSL のcipher strings、または有効な GnuTLS のpriority strings。それぞれの値は `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）用の PSK ベースの暗号スイート選択基準
証明書と PSK の統合暗号スイートリスト	TLSCipherAll13	TLS 1.3 プロトコル用の有効な OpenSSL 1.1.1 のcipher strings（その値は OpenSSL 関数 `SSL_CTX_set_ciphersuites()` に渡されます）。	TLS 1.3 用の暗号スイート選択基準 OpenSSL 1.1.1 以降のみ。
	TLSCipherAll	TLS 1.2 用の有効な OpenSSL のcipher strings、または有効な GnuTLS のpriority strings。それぞれの値は `SSL_CTX_set_cipher_list()` または `gnutls_priority_init()` 関数に渡されます。	TLS 1.2/1.3（GnuTLS）、TLS 1.2（OpenSSL）用の暗号スイート選択基準

zabbix_get およびzabbix_sender ユーティリティで暗号スイート選択を上書きするには、以下のコマンドラインパラメータを使用します。

- `--tls-cipher13`
- `--tls-cipher`

新しいパラメータは任意です。パラメータが指定されていない場合は、内部のデフォルト値が使用されます。パラメータが定義されている場合、空にすることはできません。

暗号ライブラリで TLSCipher* の値の設定に失敗した場合、サーバー、プロキシ、またはエージェントは起動せず、エラーがログに記録されます。

各パラメータがどのような場合に適用されるかを理解することが重要です。

発信接続

最も単純なケースは発信接続です。

- 証明書を使用する発信接続では、TLSCipherCert13 または TLSCipherCert を使用します
- PSK を使用する発信接続では、TLSCipherPSK13 または TLSCipherPSK を使用します
- zabbix_get および zabbix_sender ユーティリティの場合、コマンドラインパラメータ `--tls-cipher13` または `--tls-cipher` を使用できます（暗号化は `--tls-connect` パラメータで一意に指定されます）

受信接続

受信接続については、ルールがコンポーネントおよび設定ごとに異なるため、少し複雑になります。

Zabbix エージェントの場合:

Agent connection setup	Cipher configuration
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13

Agent connection setup	Cipher configuration
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Zabbix サーバーおよび プロキシの場合:

Connection setup	Cipher configuration
PSK を使用する送信接続	TLSCipherPSK, TLSCipherPSK13
証明書を使用する受信接続	TLSCipherAll, TLSCipherAll13
サーバーに証明書がない場合の、PSK を使用する受信接続	TLSCipherPSK, TLSCipherPSK13
サーバーに証明書がある場合の、PSK を使用する受信接続	TLSCipherAll, TLSCipherAll13

上記 2 つの表から、いくつかのパターンが見て取れます:

- TLSCipherAll および TLSCipherAll13 は、証明書ベース および PSK ベースの暗号スイートを組み合わせたリストを使用する場合にのみ指定できます。これが該当するケースは 2 つあります: 証明書が設定されたサーバー（プロキシ）(暗号ライブラリが PSK をサポートしている場合、サーバーおよびプロキシでは常に PSK 暗号スイートも設定されます)、証明書ベースと PSK ベースの両方の受信接続を受け入れるよう設定されたエージェント
- それ以外の場合は、TLSCipherCert* および/または TLSCipherPSK* で十分です

以下の表は、`TLSCipher*` の組み込みデフォルト値を示しています。これらは、独自のカスタム値を設定する際の良い出発点になるでしょう。

Parameter	GnuTLS 3.6.12
TLSCipherCert	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509
TLSCipherPSK	NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL
TLSCipherAll	NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

Parameter	OpenSSL 1.1.1d [1]
TLSCipherCert13	
TLSCipherCert	EECDH+aRSA+AES128:RSA+aRSA+AES128
TLSCipherPSK13	TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
TLSCipherPSK	kECDHEPSK+AES128:kPSK+AES128
TLSCipherAll13	
TLSCipherAll	EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

[1] 古い OpenSSL バージョン (1.0.1、1.0.2、1.1.0)、LibreSSL、および OpenSSL が PSK サポートなしでコンパイルされている場合は、デフォルト値が異なります。

ユーザー設定の暗号スイートの例

以下に、ユーザー設定の暗号スイートの例を示します:

- 暗号文字列をテストし、PFS 暗号スイートのみを許可する
- AES128 から AES256 への切り替え

暗号スイート文字列のテストと、PFS 暗号スイートのみの許可

どの暗号スイートが選択されたかを確認するには、設定ファイルで'DebugLevel=4' を設定するか、zabbix_sender に `-vv` オプションを使用する必要があります。

目的の暗号スイートを得るには、`TLSCipher*` パラメータをいくつか試す必要がある場合があります。`TLSCipher*` パラメータを調整するたびに Zabbix サーバー、プロキシ、またはエージェントを何度も再起動するのは不便です。より便利な方法として、zabbix_sender または `openssl` コマンドを使用できます。ここでは両方を示します。

1. zabbix_sender を使用する方法。

たとえば `/home/zabbix/test.conf` というテスト用設定ファイルを、`zabbix_agentd.conf` ファイルの構文で作成します:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agent.psk
```

この例では、有効な CA 証明書、エージェント証明書、および PSK が必要です。証明書および PSK ファイルのパスと名前は、環境に合わせて調整してください。

証明書を使用せず、PSK のみを使用している場合は、より簡単なテストファイルを作成できます:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

選択された暗号スイートは、zabbix_sender を実行することで確認できます（OpenSSL 1.1.d でコンパイルされた例）:

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [41271]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

ここでは、デフォルトで選択される暗号スイートを確認できます。これらのデフォルト値は、古い OpenSSL バージョン（1.0.1 以降）を使用するシステム上で動作する Zabbix エージェントとの相互運用性を確保するために選ばれています。

より新しいシステムでは、許可する暗号スイートを少数に絞ることで、セキュリティを強化できます。たとえば、PFS（Perfect Forward Secrecy）を持つ暗号スイートのみを許可できます。TLSCipher* パラメータを使用して、PFS を持つ暗号スイートのみを許可してみましょう。

> **Attention:**
> PSK を使用する場合、結果は OpenSSL 1.0.1 および 1.0.2 を使用するシステムとは相互運用できません。証明書ベースの暗号化は動作するはずです。

test.conf 設定ファイルに 2 行追加します:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

そして再度テストします:

```
$ zabbix_sender -vv -c /home/zabbix/test.conf -k nonexisting_item -o 1 2>&1 | grep ciphersuites
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate ciphersuites: TLS_AES_256_GCM_SHA384 TLS_
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() PSK ciphersuites: TLS_CHACHA20_POLY1305_SHA256 TLS_AE
zabbix_sender [42892]: DEBUG: zbx_tls_init_child() certificate and PSK ciphersuites: TLS_AES_256_GCM_SHA
```

「certificate ciphersuites」と「PSK ciphersuites」のリストは変更されており、以前より短くなっています。想定どおり、TLS 1.3 の暗号スイートと TLS 1.2 の ECDHE-* 暗号スイートのみが含まれています。

2. TLSCipherAll および TLSCipherAll13 は zabbix_sender ではテストできません。これらは、上記の例で表示されている「certificate and PSK ciphersuites」の値には影響しません。TLSCipherAll および TLSCipherAll13 を調整するには、エージェント、プロキシ、またはサーバーで試す必要があります。

したがって、PFS を持つ暗号スイートのみを許可するには、最大で 3 つのパラメータを

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

zabbix_agentd.conf、zabbix_proxy.conf、および zabbix_server_conf に追加する必要がある場合があります。これは、それぞれに証明書が設定されており、さらにエージェントでも PSK が設定されている場合です。

Zabbix 環境で証明書を使用せず、PSK ベースの暗号化のみを使用している場合は、必要なのは次の 1 つだけです:

```
TLSCipherPSK=kECDHEPSK+AES128
```

仕組みを理解したところで、openssl コマンドを使えば Zabbix の外でも暗号スイートの選択をテストできます。3 つの TLSCipher* パラメータ値をすべてテストしてみましょう:

```
$ openssl ciphers EECDH+aRSA+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
$ openssl ciphers kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-PSK-AES128-CBC-SHA256 E
$ openssl ciphers EECDH+aRSA+AES128:kECDHEPSK+AES128 | sed 's/:/ /g'
TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256 E
```

より詳細な出力が必要な場合は、openssl ciphers に -V オプションを付けて使用することもできます:

```
$ openssl ciphers -V EECDH+aRSA+AES128:kECDHEPSK+AES128
          0x13,0x02 - TLS_AES_256_GCM_SHA384  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(256) Mac=AEAD
          0x13,0x03 - TLS_CHACHA20_POLY1305_SHA256 TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256
          0x13,0x01 - TLS_AES_128_GCM_SHA256  TLSv1.3 Kx=any      Au=any  Enc=AESGCM(128) Mac=AEAD
          0xC0,0x2F - ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
          0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA256
          0xC0,0x13 - ECDHE-RSA-AES128-SHA    TLSv1 Kx=ECDH     Au=RSA  Enc=AES(128)  Mac=SHA1
          0xC0,0x37 - ECDHE-PSK-AES128-CBC-SHA256 TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA256
          0xC0,0x35 - ECDHE-PSK-AES128-CBC-SHA TLSv1 Kx=ECDHEPSK Au=PSK  Enc=AES(128)  Mac=SHA1
```

同様に、GnuTLS の優先順位文字列もテストできます:

```
$ gnutls-cli -l --priority=NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-A
Cipher suites for NONE:+VERS-TLS1.2:+ECDHE-RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+CURVE-ALL:+COMP-
TLS_ECDHE_RSA_AES_128_GCM_SHA256                        0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                        0xc0, 0x27      TLS1.2

Protocols: VERS-TLS1.2
Ciphers: AES-128-GCM, AES-128-CBC
MACs: AEAD, SHA256
Key Exchange Algorithms: ECDHE-RSA
Groups: GROUP-SECP256R1, GROUP-SECP384R1, GROUP-SECP521R1, GROUP-X25519, GROUP-X448, GROUP-FFDHE2048, GR
PK-signatures: SIGN-RSA-SHA256, SIGN-RSA-PSS-SHA256, SIGN-RSA-PSS-RSAE-SHA256, SIGN-ECDSA-SHA256, SIGN-E
```

AES128 から AES256 への切り替え

Zabbix はデータ用のビルトインデフォルトとして AES128 を使用しています。証明書を使用していて、OpenSSL 1.1.1 の
 AES256 に切り替えたい場合を考えてみましょう。

これは、zabbix_server.conf に各パラメータを追加することで実現可能です:

```
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/server.crt
TLSKeyFile=/home/zabbix/server.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
TLSCipherPSK13=TLS_CHACHA20_POLY1305_SHA256
TLSCipherPSK=kECDHEPSK+AES256:-SHA1
TLSCipherAll13=TLS_AES_256_GCM_SHA384
TLSCipherAll=EECDH+aRSA+AES256:-SHA1:-SHA384
```

Attention:
証明書関連の暗号スイートのみが使用されますが、より広い相互運用性のために、安全性の低い暗号を含むデフォルト値を避けるために、TLSCipherPSK* パラメータも定義されています。PSK 暗号スイートは、server / proxy で完全に無効にすることはできません。

zabbix_agentd.conf でも:

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/ca.crt
TLSCertFile=/home/zabbix/agent.crt
TLSKeyFile=/home/zabbix/agent.key
TLSCipherCert13=TLS_AES_256_GCM_SHA384
TLSCipherCert=EECDH+aRSA+AES256:-SHA1:-SHA384
```

1 証明書の利用

概要

Zabbix では、公開または社内の認証局（CA）によって署名された、PEM 形式の RSA 証明書を使用できます。

証明書の検証は、事前に設定された CA 証明書に対して実行されます。必要に応じて、証明書失効リスト（CRL）を使用できます。

各 Zabbix コンポーネントには、設定できる証明書は 1 つだけです。

内部 CA の設定と運用、証明書要求の生成と署名、証明書の失効についての詳細は、OpenSSL PKI Tutorial v2.0 などのチュートリアルを参照してください。

証明書拡張は慎重に検討し、十分にテストしてください。詳細は、X.509 v3 証明書拡張の使用上の制限を参照してください。

証明書の設定パラメータ

以下の設定パラメータは、Zabbix コンポーネントで証明書を設定するためにサポートされています。

パラメータ	必須	説明
TLSCAFile	はい	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。 複数のメンバーからなる証明書チェーンを使用する場合は、下位レベルの CA 証明書を先に、その後に上位レベルの CA 証明書を配置してください。 複数の CA の証明書を 1 つのファイルに含めることができます。
TLSCRLFile	いいえ	証明書失効リスト (CRL) を含むファイルのフルパス名。
TLSCertFile	はい	証明書を含むファイルのフルパス名。 複数のメンバーからなる証明書チェーンを使用する場合は、サーバー、プロキシ、またはエージェントの証明書を先頭に配置し、その後に下位レベルの CA 証明書、最後に上位レベルの CA 証明書を配置してください。
TLSKeyFile	はい	秘密鍵を含むファイルのフルパス名。 適切なアクセス権を設定し、このファイルがZabbix user のみ読み取り可能であることを確認してください。
TLSServerCertIssuer	いいえ	許可されるサーバー証明書の発行者。
TLSServerCertSubject	いいえ	許可されるサーバー証明書のサブジェクト。

設定例

必要な証明書を設定した後、証明書ベースの暗号化を使用するように Zabbix コンポーネントを設定します。

以下に、次の設定手順の詳細を示します。

- Zabbix サーバー
- Zabbix プロキシ
- Zabbix エージェント

Zabbix サーバー

1. CA 証明書ファイルを準備します。

ピア証明書を検証するために、Zabbix サーバーは最上位の自己署名ルート CA 証明書を含むファイルにアクセスできる必要があります。
例えば、2 つの独立したルート CA の証明書が必要な場合は、それらを `/home/zabbix/zabbix_ca_file.crt` のファイルに配置します。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
```

```
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ...
-----BEGIN CERTIFICATE-----
MIID2jCCAsKgAwIBAgIBATANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Zabbix サーバー証明書/証明書チェーンを、例えば /home/zabbix/zabbix_server.crt のファイルに配置します。
最初の証明書は Zabbix サーバー証明書で、その後に中間 CA 証明書が続きます。

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
```

```
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJM0Hw==
-----END CERTIFICATE-----
```

Note:
証明書の検証プロセスに影響を与えないように、クライアント証明書とサーバー証明書の両方で、上記の属性のみを使用してください。
例えば、*X509v3 Subject Alternative Name* または *Netscape Cert Type* 拡張を使用すると、OpenSSL が暗号化接続の確立に失敗する場合があります。
詳細については、X.509 v3 証明書拡張の使用に関する制限を参照してください。

3. Zabbix サーバー秘密鍵を、例えば /home/zabbix/zabbix_server.key のファイルに配置します。

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Zabbix サーバー設定ファイルで TLS 設定パラメータを編集します。

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix プロキシ

1. Zabbix サーバー セクションで説明されているとおりに、トップレベル CA 証明書、Zabbix プロキシ証明書/証明書チェーン、および秘密鍵を含むファイルを準備します。
次に、Zabbix プロキシ設定ファイル で TLSCAFile、TLSCertFile、TLSKeyFile パラメータを適切に編集します。

2. Zabbix プロキシ設定ファイル で追加の TLS パラメータを編集します。

- アクティブプロキシの場合: TLSConnect=cert
- パッシブプロキシの場合: TLSAccept=cert

Note:
プロキシのセキュリティを向上させるために、TLSServerCertIssuer および TLSServerCertSubject パラメータを設定することもできます。
詳細については、許可される証明書の発行者およびサブジェクトの制限 を参照してください。

最終的なプロキシ設定ファイルの TLS パラメータは、以下のようになります。

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

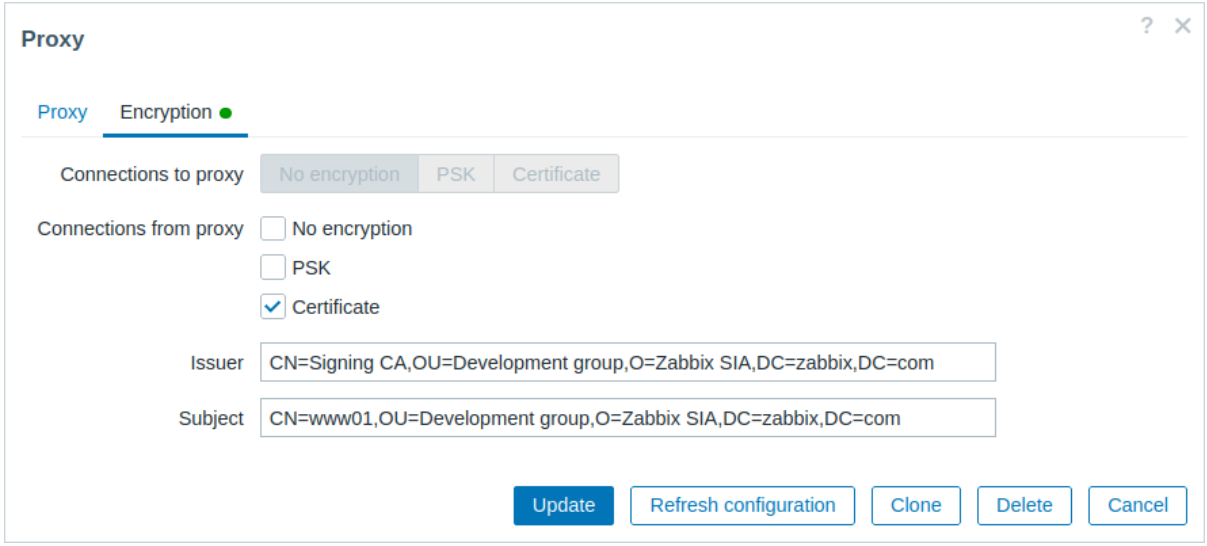
```
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Zabbix Web インターフェースでこのプロキシの暗号化を設定します。

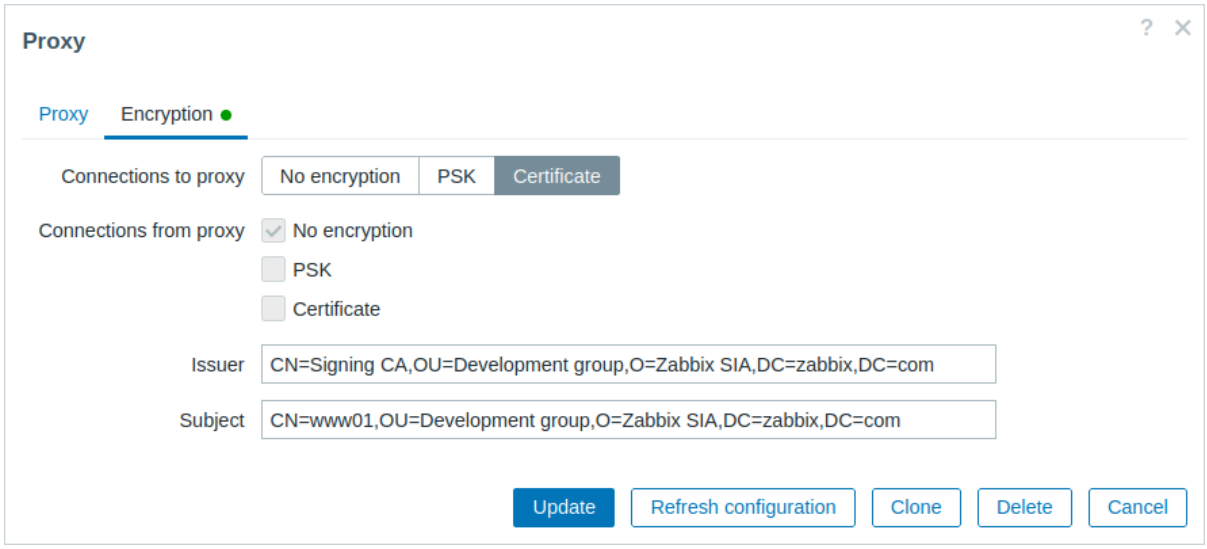
- *Administration* → *Proxies* に移動します。
- プロキシを選択し、*Encryption* タブをクリックします。

以下の例では、*Issuer* フィールドと *Subject* フィールドが入力されています。
これらのフィールドを使用する理由と使用方法の詳細については、許可される証明書の発行者およびサブジェクトの制限 を参照してください。

アクティブプロキシの場合:



パッシブプロキシの場合:



Zabbix エージェント

1. Zabbix サーバー セクションで説明されているように、トップレベル CA 証明書、Zabbix エージェント証明書/証明書チェーン、および秘密鍵のファイルを準備します。次に、Zabbix エージェント設定ファイル の TLSCAFile、TLSCertFile、TLSKeyFile パラメータをそれに応じて編集します。

2. Zabbix エージェント設定ファイル で追加の TLS パラメータを編集します。

- アクティブエージェントの場合: `TLSConnect=cert`
- パッシブエージェントの場合: `TLSAccept=cert`

Note:
エージェントのセキュリティを向上させるため、TLSServerCertIssuer および TLSServerCertSubject パラメータを設定できます。詳細については、許可される証明書の発行者およびサブジェクトの制限 を参照してください。

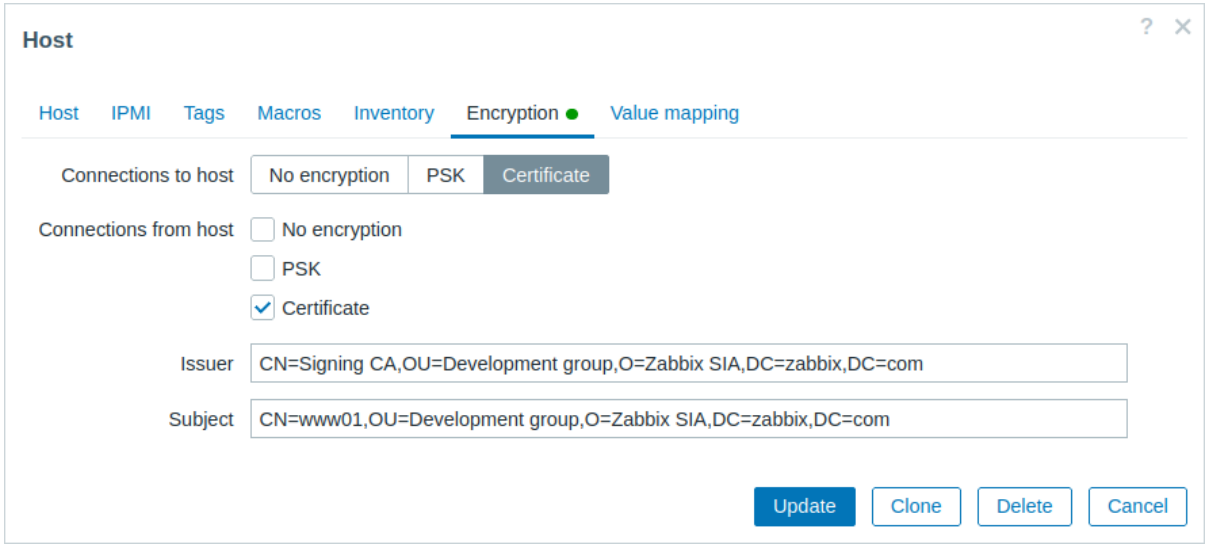
最終的なエージェント設定ファイル内の TLS パラメータは、以下のようになります。この例では、ホストがプロキシによって監視されていることを前提としているため、証明書の Subject としてプロキシが指定されている点に注意してください。

```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. このエージェントによって監視されるホストについて、Zabbix Web インターフェースで暗号化を設定します。

- 次へ移動: データ収集 → ホスト。
- ホストを選択し、暗号化タブをクリックします。

以下の例では、*Issuer* フィールドと *Subject* フィールドが入力されています。これらのフィールドを使用する理由と使用方法の詳細については、許可される証明書の発行者およびサブジェクトの制限 を参照してください。



Zabbix web service

1. Zabbix server セクションで説明されているとおりに、最上位 CA 証明書、Zabbix web service の証明書/証明書チェーン、および秘密鍵を含むファイルを準備します。
次に、Zabbix web service 設定ファイル の TLSCAFile、TLSCertFile、TLSKeyFile パラメータをそれに応じて編集します。

2. Zabbix web service 設定ファイル で追加の TLS パラメータ TLSAccept=cert を編集します。

最終的な web service 設定ファイルの TLS パラメータは、次のようになります。

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Zabbix server 設定ファイル の WebServiceURL パラメータを編集し、TLS が設定された Zabbix web service に接続するように Zabbix server を設定します。

```
WebServiceURL=https://example.com:443/report
```

許可される証明書の発行者およびサブジェクトの制限

2 つの Zabbix コンポーネント（たとえば、サーバーとエージェント）が TLS 接続を確立する際、それぞれ相手の証明書を検証します。ピア証明書が信頼された CA によって署名されており（TLSCAFile に事前設定された最上位証明書を使用）、有効で、期限切れでなく、その他のチェックにも合格した場合、コンポーネント間の通信を続行できます。
この最も単純なケースでは、証明書の発行者およびサブジェクトは検証されません。

ただし、これにはリスクがあります。有効な証明書を持つ者であれば誰でも、他者になりすますことができるためです（たとえば、ホスト証明書を使用してサーバーになりすますことが可能です）。
これは、証明書が専用の社内 CA によって署名されており、なりすましのリスクが低い小規模な環境では許容できる場合がありますが、より大規模な環境や、セキュリティ要件の高い環境では十分ではない可能性があります。

最上位 CA が、Zabbix で受け入れるべきではない証明書を発行する可能性がある場合、またはなりすましのリスクを低減したい場合は、発行者およびサブジェクトを指定して、許可される証明書を制限できます。

たとえば、Zabbix プロキシの設定ファイルでは、次のように指定できます。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

これらの設定により、アクティブプロキシは、証明書の発行者またはサブジェクトが異なる Zabbix サーバーとは通信しません。
同様に、パッシブプロキシは、そのようなサーバーからのリクエストを受け付けません。

Issuer および Subject 文字列の照合ルール

Issuer および Subject 文字列の照合ルールは以下のとおりです。

- Issuer 文字列と Subject 文字列は、それぞれ独立してチェックされます。
  どちらも省略可能です。
- 文字列が指定されていない場合は、任意の文字列が受け入れられます。
- 文字列はそのまま比較され、完全に一致する必要があります。
- UTF-8 文字がサポートされています。
  ただし、ワイルドカード（*）や正規表現はサポートされていません。
- 以下の RFC 4514 の要件が実装されています - エスケープが必要な文字（'\' バックスラッシュ、U+005C を使用）：
  - 文字列内の任意の位置: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - 文字列の先頭: スペース (' ', U+0020) または番号記号 ('#', U+0023);
  - 文字列の末尾: スペース (' ', U+0020)。
- Null 文字（U+0000）はサポートされていません。
  Null 文字が検出された場合、照合は失敗します。
- RFC 4517 および RFC 4518 の標準はサポートされていません。

たとえば、Issuer および Subject の organization (O) 文字列に末尾スペースが含まれ、Subject の organizational unit (OU) 文字列に二重引用符が含まれている場合、これらの文字はエスケープする必要があります。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development head,O=\ Example SIA\ ,DC=example,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group \"5\",O=\ Example SIA\ ,DC=example,DC=com
```

フィールドの順序と書式

Zabbix は RFC 4514 の推奨事項に従っており、これらのフィールドには「逆順」の並びを指定しています。これは、最下位レベルのフィールド (CN) から始まり、中間レベルのフィールド (OU, O) へ進み、最後に最上位レベルのフィールド (DC) で終わる順序です。

```
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

一方、OpenSSL はデフォルトで Issuer および Subject の文字列を、上位レベルから下位レベルへの順序で表示します。
次の例では、Issuer フィールドと Subject フィールドは最上位レベル (DC) で始まり、最下位レベル (CN) フィールドで終わっています。
また、スペースやフィールド区切りの書式も使用するオプションによって異なるため、Zabbix で必要とされる形式とは一致しません。

```
$ openssl x509 -noout -in /home/zabbix/zabbix_proxy.crt -issuer -subject
issuer= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Signing CA
subject= /DC=com/DC=zabbix/O=Zabbix SIA/OU=Development group/CN=Zabbix proxy

$ openssl x509 -noout -text -in /home/zabbix/zabbix_proxy.crt
Certificate:
    ...
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix proxy
```

Zabbix 用に *Issuer* および *Subject* の文字列を正しく整形するには、次のオプションを付けて OpenSSL を実行します。

```
$ openssl x509 -noout -issuer -subject \
    -nameopt esc_2253,esc_ctrl,utf8,dump_nostr,dump_unknown,dump_der,sep_comma_plus,dn_rev,sname\
    -in /home/zabbix/zabbix_proxy.crt
```

すると、出力は逆順でカンマ区切りとなり、Zabbix の設定ファイルおよび Web インターフェースで使用できる形式になります。

```
issuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
subject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
```

X.509 v3 証明書拡張の使用に関する制限事項

Zabbix で X.509 v3 証明書を実装する場合、一部の拡張機能は完全にはサポートされていないか、一貫しない動作を引き起こす可能性があります。

Subject Alternative Name 拡張

Zabbix は、IP アドレスやメールアドレスなどの代替 DNS 名を指定するために使用される *Subject Alternative Name* 拡張をサポートしていません。
Zabbix が検証できるのは、証明書の *Subject* フィールドの値のみです（許可された証明書の発行者および Subject の制限を参照）。
証明書に `subjectAltName` フィールドが含まれている場合、証明書検証の結果は、Zabbix コンポーネントのコンパイルに使用された暗号ツールキットによって異なることがあります。
その結果、これらの組み合わせに応じて、Zabbix は証明書を受け入れる場合もあれば拒否する場合もあります。

Extended Key Usage 拡張

Zabbix は *Extended Key Usage* 拡張をサポートしています。
ただし、これを使用する場合は、通常、*clientAuth*（TLS WWW クライアント認証用）と *serverAuth*（TLS WWW サーバー認証用）の両方の属性を指定する必要があります。
例えば、次のとおりです。

- Zabbix エージェントが TLS サーバーとして動作するパッシブチェックでは、エージェントの証明書に *serverAuth* 属性を含める必要があります。
- エージェントが TLS クライアントとして動作するアクティブチェックでは、エージェントの証明書に *clientAuth* 属性を含める必要があります。

GnuTLS はキー使用法違反に対して警告を出す場合がありますが、通常はこれらの警告があっても通信の継続を許可します。

Name Constraints 拡張

Name Constraints 拡張のサポートは、暗号ツールキットによって異なります。
選択したツールキットがこの拡張をサポートしていることを確認してください。
使用しているツールキットによっては、このセクションが critical としてマークされている場合、この拡張により Zabbix が CA 証明書を読み込めなくなることがあります。

証明書失効リスト（CRL）

証明書が侵害された場合、認証局（CA）はその証明書を証明書失効リスト（CRL）に含めることで失効させることができます。
CRL は設定ファイルを通じて管理され、サーバー、プロキシ、およびエージェントの設定ファイルで `TLSCRLFile` パラメータを使用して指定できます。
例えば、次のようになります。

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

この場合、`zabbix_crl_file.crt` には複数の CA の CRL を含めることができ、次のようになります。

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

CRL ファイルは Zabbix の起動時にのみ読み込まれます。
CRL を更新するには、Zabbix を再起動してください。

Attention:
Zabbix コンポーネントが OpenSSL でコンパイルされており、CRL を使用する場合は、証明書チェーン内の各ルート CA および中間 CA に対応する CRL（空であっても）を `TLSCRLFile` に含めるようにしてください。

2 事前共有鍵の使用

概要

Zabbix の各事前共有キー (PSK) は、次のペアです。

- 非秘匿 PSK ID 文字列、
- 秘匿 PSK 文字列値。

PSK ID 文字列は、空ではない UTF-8 の文字列です。たとえば”PSK ID 001 Zabbix agentd” などです。この特定の PSK が Zabbix コンポーネントによって参照される一意の名前です。暗号化されずにネットワーク経由で送信されますので、機密情報を PSK ID 文字列に含めないでください。

PSK 文字列値は推測しにくい 16 進数の文字列です。たとえば、”e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9” などです。

サイズ制限

Zabbix には PSK ID と値のサイズ制限があります。場合によっては、暗号化ライブラリの制限がより低くなることがあります。

コンポーネント	PSK ID 最大サイズ	PSK 値最小サイズ	PSK 値最大サイズ
Zabbix	128 UTF-8 文字列	128-bit (16-bytes PSK, 16 進数 32 桁)	2048-bit (256-bytes PSK, 16 進数 512 桁)
GnuTLS	128 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)
OpenSSL 1.0.x, 1.1.0	127 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)
OpenSSL 1.1.1	127 bytes (UTF-8 文字列を含む)	-	512-bit (64-byte PSK, 16 進数 128 桁)
OpenSSL 1.1.1a 以降	127 bytes (UTF-8 文字列を含む)	-	2048-bit (256-byte PSK, 16 進数 512 桁)

Attention:
Zabbix フロントエンドでは使用する暗号化ライブラリに関係なく、最大 128 文字の PSK ID 文字列と 2048 ビットの PSK を設定できます。\一部の Zabbix コンポーネントが下限をサポートしている場合、これらのコンポーネントに許可された長さで PSK ID 文字列と PSK 値を設定するのはユーザーの責任です。\長さの制限を超えると、Zabbix コンポーネント間で通信障害が発生します。

Zabbix サーバーが PSK を使用してエージェントに接続する前に、サーバーはそのエージェント用に構成された PSK ID と PSK 値をデータベース (実際には構成キャッシュ) で検索します。接続を受信すると、エージェントは設定ファイルから PSK ID と PSK 値を使用します。双方が同じ PSK ID 文字列と PSK 値を持っている場合、接続が成功します。

Attention:
各 PSK ID は、1 つの値のみとペアにする必要があります。ID 文字列が同じで値が異なる 2 つの PSK が存在しないことを確認するのは、ユーザーの責任です。そうしないと、予測できないエラーが発生したり、この PSK ID 文字列を持つ PSK を使用する Zabbix コンポーネント間の通信が中断したりする可能性があります。

PSK の生成

たとえば、256 ビット (32 バイト) の PSK は、次のコマンドを使用して生成できます。

- with *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- with *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Generating a random key for user 'psk_identity'
Key stored to database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

上記の「psktool」は、PSK ID とそれに関連付けられた PSK を含むデータベースファイルを生成することに注意してください。Zabbix は PSK ファイルに PSK のみを想定しているため、識別文字列とコロン (':') をファイルから削除する必要があります。

サーバーエージェント通信用の PSK 設定 (例)

エージェントホストで PSK 値をファイルに書き込みます (例: `/home/zabbix/zabbix_agentd.psk`)。ファイルには、最初のテキスト文字列に PSK が含まれている必要があります。例えば、

1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

PSK ファイルへのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

エージェント設定ファイル zabbix_agentd.conf で TLS パラメーターを編集します。たとえば次のように設定します。

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

エージェントはサーバーに接続し、PSK を使用したサーバー (アクティブチェック) と zabbix_get からの接続のみを受けつけます。PSK ID は”PSK 001” です。

エージェント再起動後、zabbix_get を使用して接続をテストできます。例えば、

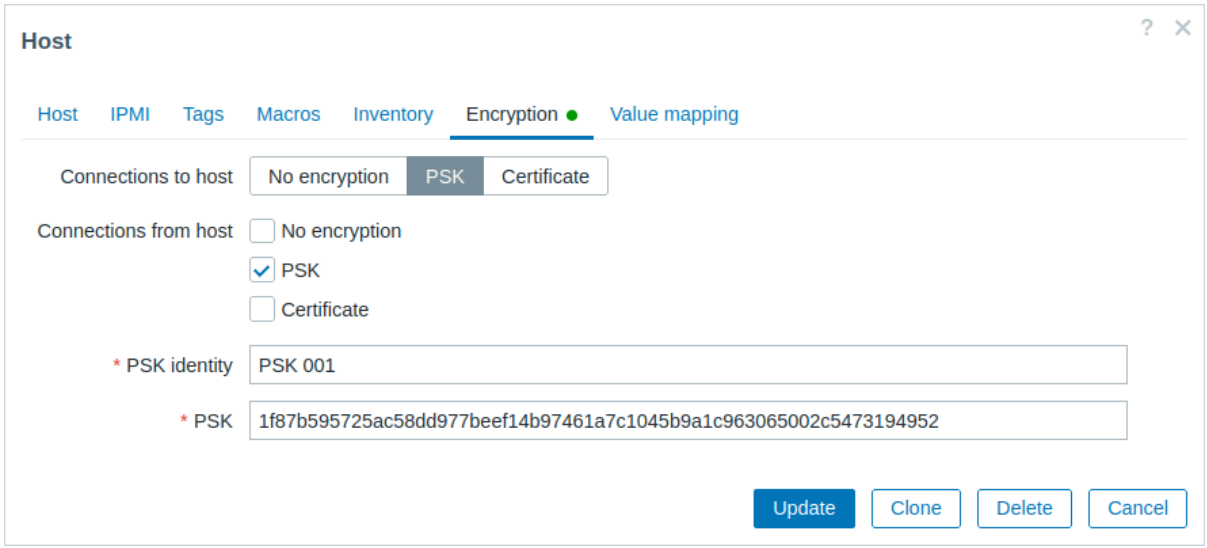
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(ダウンタイムを最小限に抑えるには、接続暗号化の管理の接続タイプを変更する方法を参照してください。)

Zabbix フロントエンドでこのエージェントの PSK 暗号化を設定します。

- データ収集 → ホストに移動
- ホストを選択して暗号化タブをクリック

例:



すべての必須入力フィールドには赤いアスタリスクが付いています。

設定キャッシュがデータベースと同期されると、新しい接続は PSK を使用します。サーバーとエージェントのログファイルでエラーメッセージを確認してください。

サーバーの PSK の構成 - アクティブなプロキシ通信 (例)

プロキシで PSK 値をファイルに書き込みます (例: /home/zabbix/zabbix_proxy.psk)。ファイルの最初のテキスト文字列に PSK が含まれている必要があります。次に例を示します。

e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9

PSK ファイルへのアクセス権を設定します。Zabbix ユーザーだけが読み取れるようにする必要があります。

プロキシ構成ファイル zabbix_proxy.conf で TLS パラメータを編集します。たとえば、次のように設定します。

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

プロキシは PSK を使用してサーバーに接続します。PSK ID は”PSK 002” になります。

(ダウンタイムを最小限に抑えるには、接続暗号化の管理 で接続タイプを変更する方法を参照してください)。

Zabbix フロントエンドでこのプロキシの PSK を構成します。*[*管理*]*→プロキシ に移動し、プロキシを選択して、暗号化 タブに移動します。” プロキシからの接続” で PSK にチェックします。”PSK アイデンティティ” フィールドに”PSK 002” を貼り付け、”e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9” を”PSK” フィールドに入力して、” 更新” をクリックします。

プロキシを再起動して、サーバーへの PSK ベースの暗号化された接続の使用を開始します。サーバーとプロキシのログファイルでエラーメッセージを確認してください。

パッシブプロキシの場合、手順は非常に似ています。唯一の違いは、プロキシ構成ファイルで `TLSAccept=psk` を設定し、Zabbix フロントエンドで" プロキシへの接続" を `PSK` に設定することです。

3 トラブルシューティング

一般的な推奨事項

- 問題が発生した場合、どのコンポーネントが TLS クライアントとして機能し、どのコンポーネントが TLS サーバーとして機能するかを理解することから始めます。\Zabbix サーバー、プロキシ、およびエージェントは、それらの間の相互作用に応じて、すべて TLS サーバーおよびクライアントとして機能します。\たとえば、Zabbix パッシブチェックのためにエージェントに接続するサーバーは、TLS クライアントとして機能します。エージェントは TLS サーバーの役割を果たします。\Zabbix エージェントは、プロキシからアクティブなチェックのリストを要求し、TLS クライアントとして機能します。プロキシは TLS サーバーの役割を果たします。'zabbix_get および zabbix_sender' ユーティリティは常に TLS クライアントとして機能します。
- Zabbix は相互認証を使用します。 それぞれの側がピアを検証し、接続を拒否する場合があります。 たとえば、エージェントに接続している Zabbix サーバーは、エージェントの証明書が無効な場合、すぐに接続を閉じることができます。逆もまた同様です - サーバーがエージェントによって信頼されていない場合、サーバーからの接続を受け入れる Zabbix エージェントは接続を閉じることができます。
- TLS クライアントと TLS サーバーの両方でログファイルを調べます。\接続を拒否する側は、拒否された正確な理由をログに記録する場合があります。反対側は、かなり一般的なエラーを報告することがよくあります (たとえば、「ピアによって接続が閉じられました」、「接続が正しく終了されませんでした」など)。
- 暗号化の設定を誤ると、実際の原因を示すものではなく、混乱を招くエラーメッセージが表示されることがあります。 以下のサブセクションでは、トラブルシューティングに役立つ可能性のあるメッセージと考えられる原因の (完全ではない) コレクションを提供しようとしています. さまざまな暗号化ツールキット (OpenSSL、GnuTLS) は、同じ問題の状況で異なるエラーメッセージを生成することがよくあります。 場合によっては、エラーメッセージが両側の暗号化ツールキットの特定の組み合わせに依存することもあります。

1 接続タイプまたは権限の問題

サーバーは PSK を使用してエージェントに接続するように設定されているが、エージェントは暗号化されていない接続のみを受け入れる場合

サーバーまたはプロキシログ (*GnuTLS* 3.3.16 を使用)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

サーバーまたはプロキシログ (*OpenSSL* 1.0.2c を使用)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

片側は証明書で接続するが、反対側は PSK のみを受け入れる、またはその逆

任意のログ (*GnuTLS* を使用):

```
failed to accept an incoming connection: from 127.0.0.1: zbx_tls_accept(): gnutls_handshake() failed:\
    -21 Could not negotiate a supported cipher suite.
```

任意のログ (*OpenSSL* 1.0.2c を使用):

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake returned error code 1:\
    file .\ssl\s3_srvr.c line 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

TLS サポートでコンパイルされた Zabbix センダーを使用して、TLS なしでコンパイルされた Zabbix サーバー/プロキシにデータを送信した場合

接続元ログ:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL library (version OpenSSL 1.1.1  11 Sep 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
```

```
...End of zbx_tls_connect():FAIL error:'connection closed by peer'
...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by pee
```

Windows:

```
...OpenSSL library (version OpenSSL 1.1.1a  20 Nov 2018) initialized
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

受付側ログ:

```
...failed to accept an incoming connection: from 127.0.0.1: support for TLS was not compiled in
```

一方は PSK で接続するが、もう一方は LibreSSL を使用しているか、暗号化サポートなしでコンパイルされている場合

LibreSSL は PSK をサポートしていません。

接続元ログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

受付側ログ :

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Zabbix フロントエンド:

```
Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

一方は PSK で接続しますが、もう一方は PSK サポートを無効にして OpenSSL を使用している場合

接続元ログ:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERR
```

受付側ログ:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 証明書の問題

CRL で使用される OpenSSL と証明書チェーンの一部の CA では、CRL が TLSCRLFile に含まれていない

OpenSSL ピアの場合の TLS サーバーログ:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file s3_srvr.c line 3251: error:14089086: SSL routines:ssl3_get_client_certificate:certificate verify
    TLS write fatal alert "unknown CA"
```

GnuTLS ピアの場合の TLS サーバーログ:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL が期限切れになった、またはサーバー操作中に期限切れになった

OpenSSL サーバーログ:

- 期限切れ前:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- 期限切れ後:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
    SSL_connect() returned SSL_ERROR_SSL: file s3_clnt.c line 1253: error:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

ここでのポイントは、有効な CRL を使用すると、取り消された証明書が「証明書が取り消された」として報告されることです。CRL の有効期限が切れると、エラーメッセージが「証明書の有効期限が切れました」に変わります。これは誤解を招く可能性があります。

GnuTLS サーバーログ:

- 有効期限の前後で同じ:

```
cannot connect to proxy "proxy-openssl-1.0.1e": TCP successful, cannot establish TLS to [[127.0.0.1]:20004
      invalid peer certificate: The certificate is NOT trusted. The certificate chain is revoked.
```

自己署名証明書や不明な CA

OpenSSL ログ:

```
error:'self signed certificate: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/statem/statem_
      line 1924: error:1416F086:SSL routines:tls_process_server_certificate:certificate verify failed:\
      TLS write fatal alert "unknown CA"'
```

これはサーバー証明書が CA によって署名されているにもかかわらず、間違って発行者とサブジェクトの文字列が同じである場合に発生します。最上位の CA 証明書では発行者とサブジェクトは同じですが、サーバー証明書では同じにすることはできません。(プロキシ証明書とエージェント証明書も同様です。)

証明書に同じ発行者とサブジェクトのエントリが含まれているかどうかを確認するには、次を実行します。

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

ルート (トップレベル) 証明書が発行者とサブジェクトの同一の値を持つことは許容されます。

3 PSK の問題

PSK に奇数個の 16 進数が含まれている

プロキシまたはエージェントが起動しない、プロキシまたはエージェントログのメッセージ:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

128 バイトを超える PSK ID 文字列が GnuTLS に渡された

TLS クライアント側のログ:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

TLS サーバー側のログ:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

OpenSSL 1.1.1 で使用される PSK 値が長すぎる

接続側ログ: ...OpenSSL library (version OpenSSL 1.1.1 11 Sep 2018) initialized ... ...In zbx_tls_connect(): psk_identity:"PSK 1" ...zbx_psk_client_cb() requested PSK identity "PSK 1" ...End of zbx_tls_connect():FAIL error:'SSL_connect() set result code to SSL_ERROR_SSL: file ssl\statem\extensions_clnt.c line 801: error:14212044:SSL routines:tls_construct_ctos_early_data:internal error: TLS write fatal alert "internal error"'

受付側ログ:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

この問題は通常、OpenSSL を 1.0.x または 1.1.0 から 1.1.1 にアップグレードするときに、PSK 値が 512 ビット (64 バイト PSK、128 の 16 進数として入力) よりも長い場合に発生します。

参照: 値のサイズ制限

18. Web インターフェース

概要　どこからでも、どのプラットフォームからでも Zabbix に簡単にアクセスできるように、Web ベースのインターフェースが提供されています。

> **Note:**
> 複数のフロントエンドインスタンスを使用する場合は、言語や地域の設定とライブラリ (LDAP、SAML など) がすべてのフロントエンドにインストールされ、同じように構成されていることを確認してください。

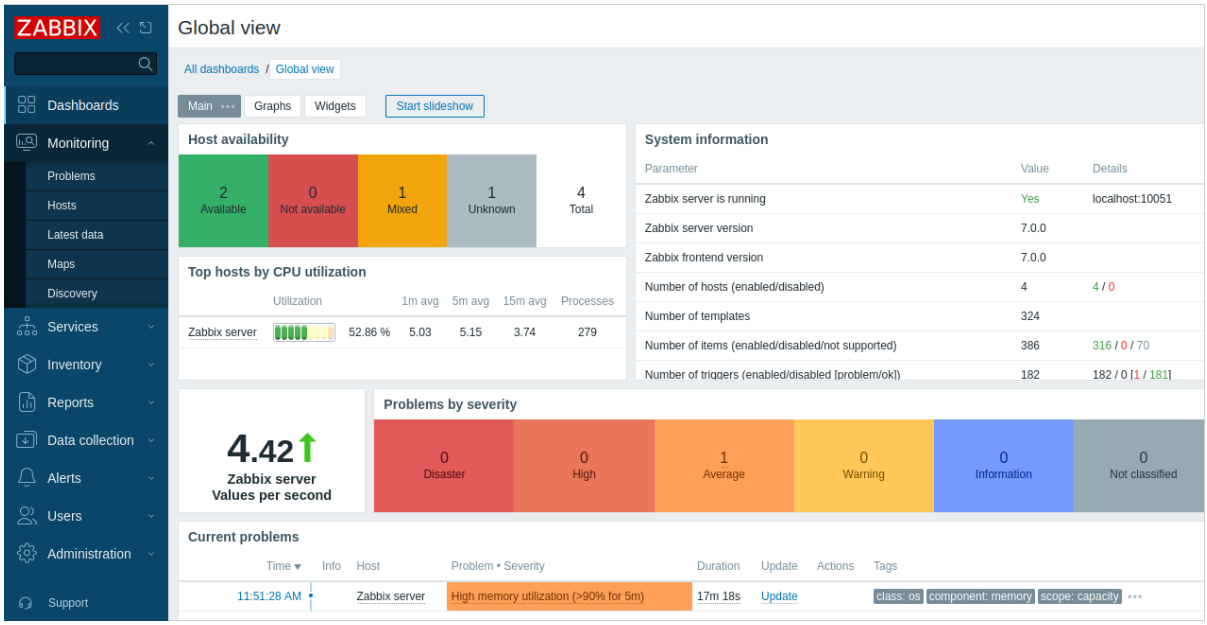
フロントエンドヘルプ　Zabbix フロントエンドフォームにはヘルプリンク があり、ドキュメントの対応する部分への直接リンクが提供されます。

1 メニュー

概要

サイドバーのメニューから、さまざまな Zabbix フロントエンドセクションにアクセスできます。

デフォルトのテーマでは、メニューは紺色です。

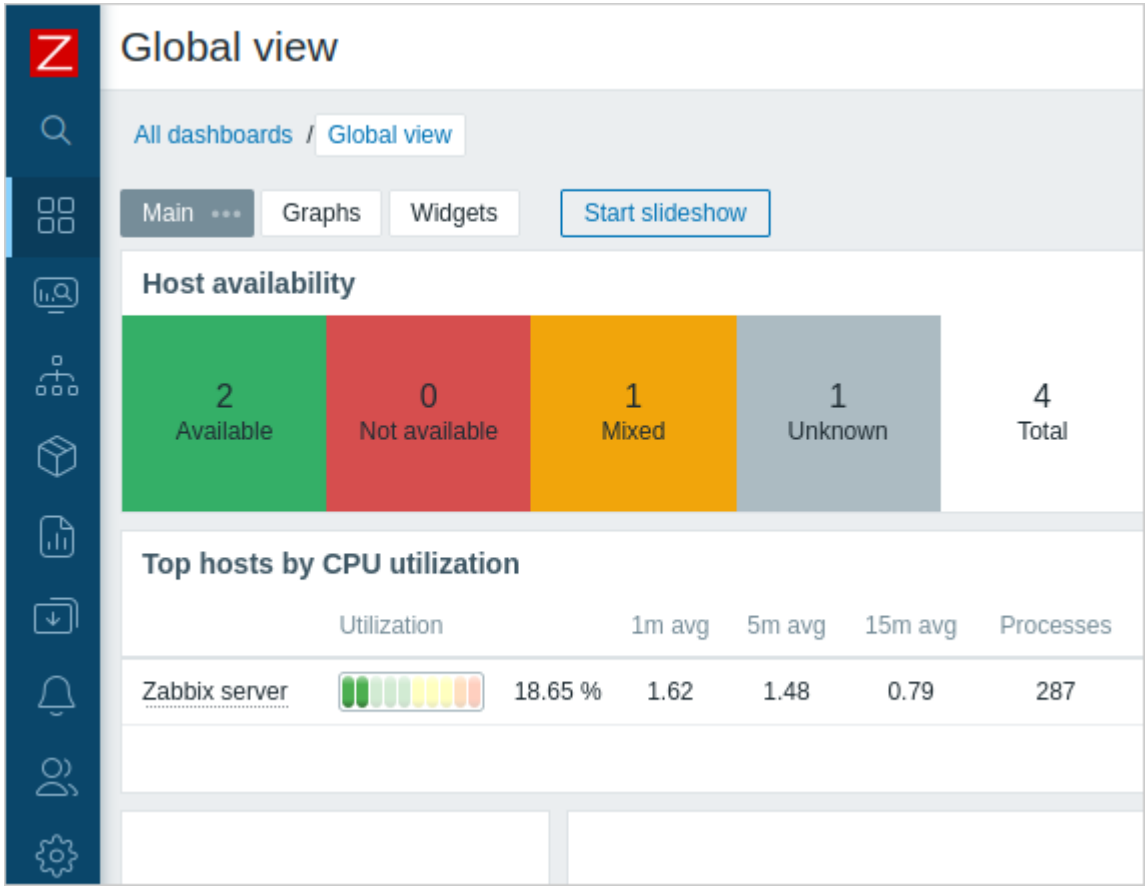


メニューの操作

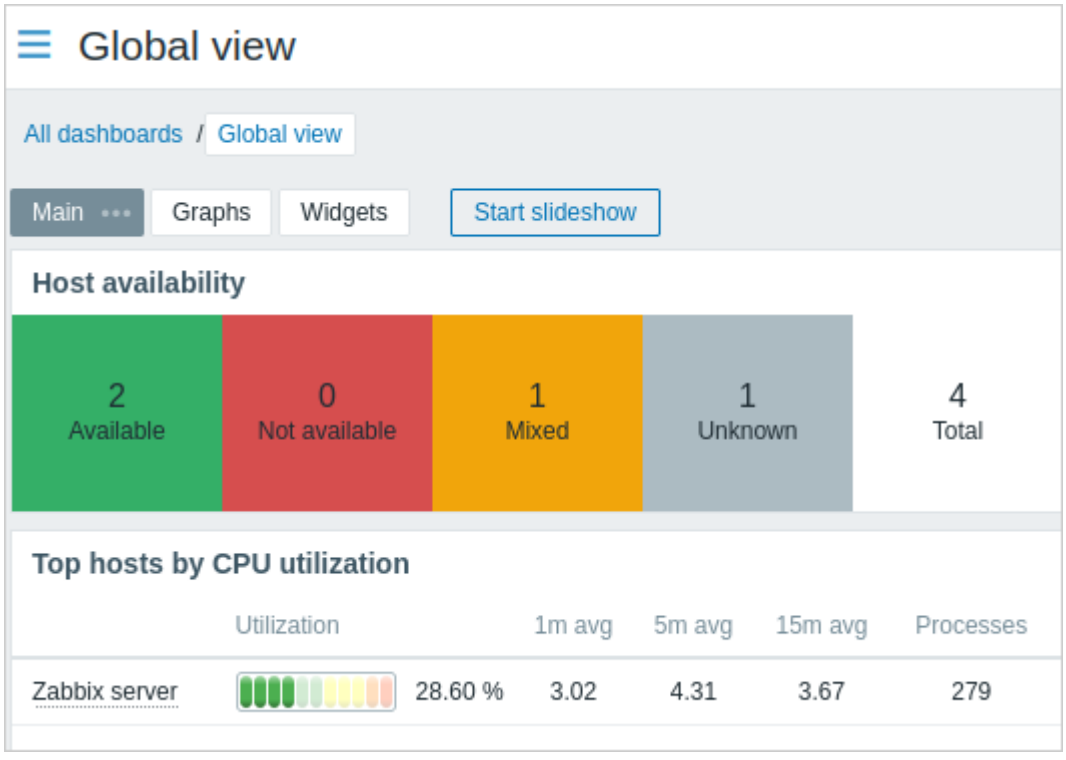
Zabbix ロゴの下にグローバル検索ボックスがあります

メニューは折りたたんだり、完全に非表示にしたりできます。

- 折りたたむには、Zabbix ロゴの横にある をクリックします。折りたたまれたメニューでは、アイコンのみが表示されます。



- 非表示にするには、Zabbix ロゴの横にある  をクリックします。非表示メニューではすべてが非表示になります。



折りたたまれたメニュー

メニューがアイコンのみの表示に折りたたまれている場合、マウスカーソルをその上に置くと、すぐに完全なメニューが再表示されます。再表示されたメニューはページコンテンツの上に重なって表示される点に注意してください。ページコンテンツを右に移動するには、展開ボタンをクリックする必要があります。マウスカーソルを再び完全なメニューの外に移動すると、2 秒後にメニューは再度折りたたまれます。

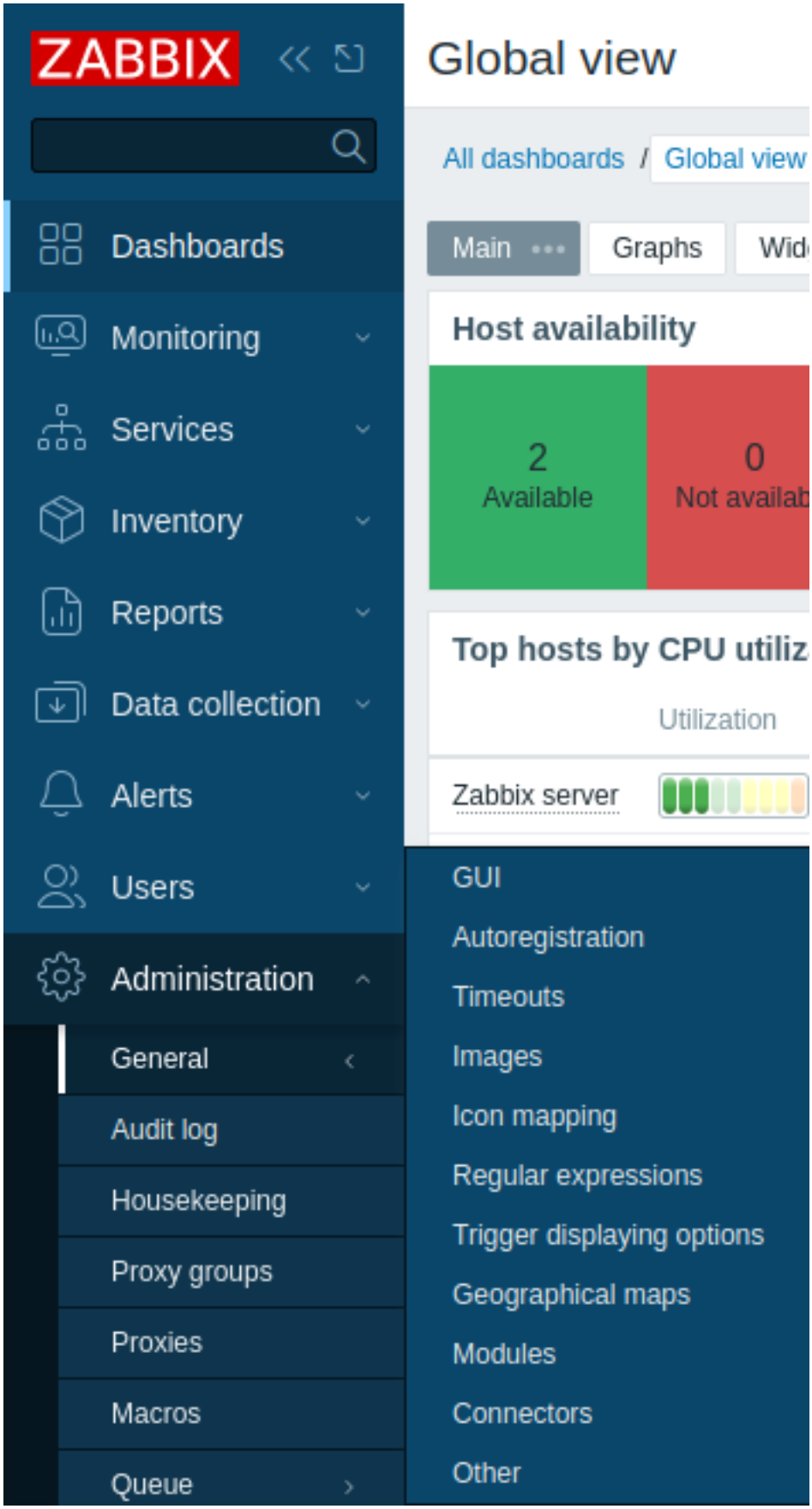
また、Tab キーを押すことで、折りたたまれたメニューを完全表示で再表示することもできます。Tab キーを繰り返し押すと、次のメニュー要素にフォーカスを移動できます。

メニュー非表示

メニューが完全に非表示になっている場合でも、ハンバーガーアイコン  を一回クリックするだけで完全なメニューを表示できます。ページコンテンツの上に再表示されることに注意してください。ページコンテンツを右に移動するには、サイドバーの表示ボタンをクリックしてメニューを再表示する必要があります。

メニューの階層

メニューには最大 3 つの階層があります。



コンテキストメニュー

メインメニューに加えて、Zabbix はホスト、アイテム、イベントのコンテキストメニューを提供し、最新の値、シンプルグラフ、設定フォーム、関連スクリプト、外部リンクなど、頻繁に使用されるエンティティにすばやくアクセスできます。

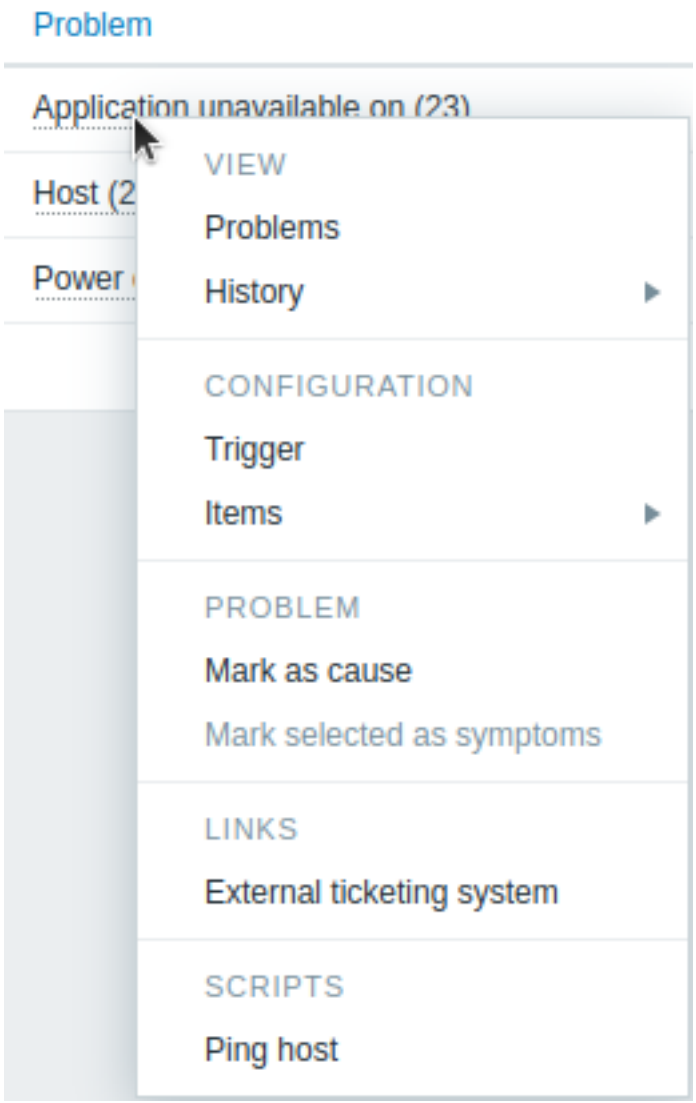
コンテキストメニューには、サポートされている場所でホスト、アイテム、または障害/トリガー名をクリックするとアクセスできます。

1 イベントメニュー

概要

イベントメニューには、イベントで頻繁に必要となるアクションやフロントエンドセクションへのショートカットが含まれています。

イベントメニューは、イベント名をクリックすると開きます。



コンテンツ

イベントコンテキストメニューには、表示、アクション、設定、障害、リンク、スクリプトの 6 つのセクションがあります。設定されていないエンティティの場合、リンクは無効になり、灰色で表示されます。スクリプトセクションとリンクセクションは、エンティティが設定されている場合に表示されます。

表示セクションには、次へのリンクが含まれています。

- 障害 - 基礎となるトリガーの未解決の障害のリストを開きます。
- ヒストリ - l 基礎となるアイテムの最新データグラフ/アイテムヒストリにつながります。トリガーが複数のアイテムを使用する場合、各アイテムのリンクが使用可能になります。

アクションセクションは、トリガー概要ウィジェットでのみ使用できます。以下へのリンクが含まれています。

- 障害更新 - 障害更新画面を開きます。

設定セクションには、次の設定へのリンクが含まれています。

- 障害を引き起こしたトリガー。
- トリガー式で使用されるアイテム。

Note:
設定セクションは、admin および Super admin ユーザーのみが使用できることに注意してください。

障害セクションには、次のオプションが含まれています。

- 副次的な障害から削除 - 障害を原因としてマークします。
- 選択したものを副次的な障害として追加 - 選択した障害をこの障害の症状としてマークします。

リンクセクションには、次のリンクが含まれています。

- 設定されたトリガー URLにアクセスします
- グローバルスクリプトで設定されたカスタムリンクにアクセスします (' 手動イベント処理' スコープに'URL' タイプで)
- 障害に対して設定された外部チケットにアクセスします (Webhookを設定するときにイベントメニュー項目を含めるオプションを参照してください)。

スクリプトセクションには、グローバルスクリプトを実行するためのリンクが含まれています (手動イベントアクションスコープで)。この機能は、外部システムで障害チケットを管理するために使用されるスクリプトを実行する場合に便利です。

サポートされている場所

イベントコンテキストメニューには、さまざまなフロントエンドセクションで障害またはイベント名をクリックするとアクセスできます。次に例を示します。

- ダッシュボードウィジェット、障害ウィジェット、トリガー概要ウィジェットなど。
- 監視 → 障害
- 監視 → 障害 → イベント詳細
- レポート → 障害発生数上位 100 項目 (グローバルスクリプトと外部チケットへのアクセスは、この場所ではサポートされていません)

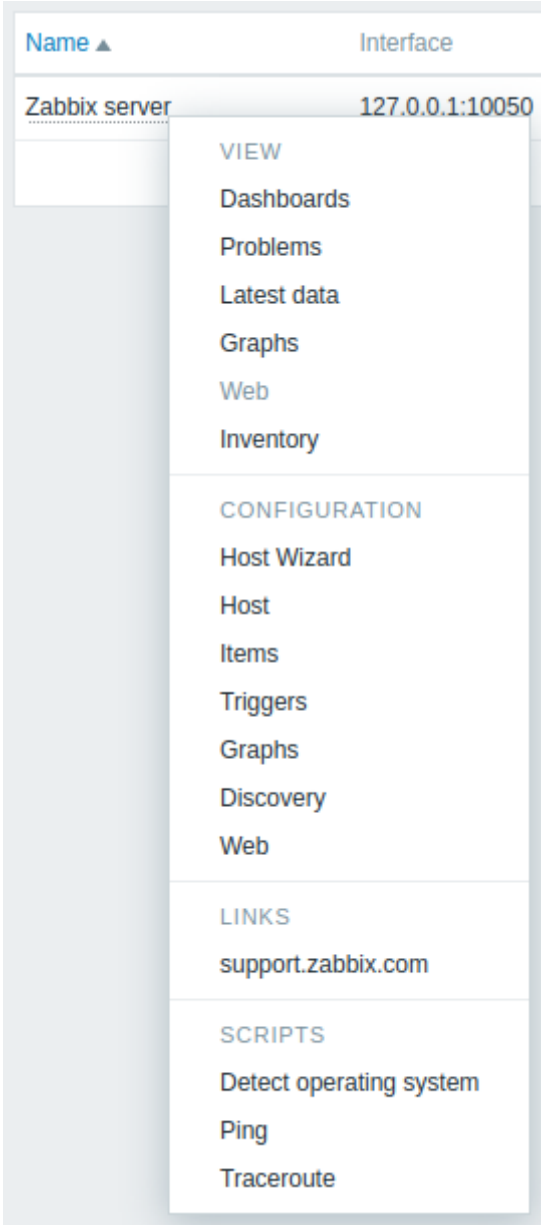
2 ホストメニュー

概要

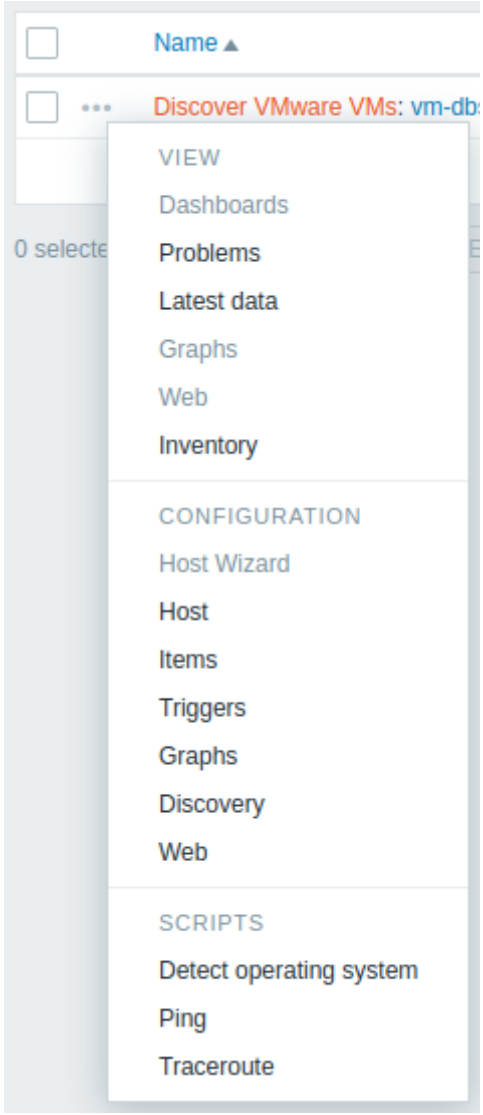
ホストメニューには、ホストに対して頻繁に必要となるアクションや、Web インターフェースのセクションへのショートカットが含まれています。

ホストメニューは、Web インターフェースのセクションに応じて、ホスト名または三点アイコンをクリックすると開くことができます。例えば、次のとおりです。

Monitoring → Latest data



Data collection → Hosts



内容

ホストのコンテキストメニューには、表示、設定、リンク、スクリプトの 4 つのセクションがあります。
設定されていないエンティティについては、リンクは無効化され、灰色で表示されます。
スクリプトおよび リンクセクションは、それらのエンティティが設定されている場合に表示されます。

表示セクションには、次へのリンクがあります。

- ダッシュボード - ウィジェットとグラフを開きます。
- 障害 - 基になるトリガーの未解決の障害一覧を表示した 障害セクションを開きます。
- 最新データ - 現在のホストのすべての最新データ一覧を表示した 最新データセクションを開きます。
- グラフ - 現在のホストのシンプルグラフを開きます。
- **Web** - 設定済みの Web シナリオへのリンクを開きます。
- インベントリ - 現在のホストのインベントリへのリンクを開きます。

設定セクションには、次へのリンクがあります。

- ホストウィザード - 現在のホストのHost Wizard を開きます（ディスカバリされたホストでは無効）。
- ホスト - 現在のホストの設定フォームです。
- アイテム - 現在のホストのアイテム一覧です。
- トリガー - 現在のホストのトリガー一覧です。
- グラフ - 現在のホストのシンプルグラフです。
- ディスカバリ - 現在のホストのローレベルディスカバリルール一覧です。

- **Web** - 現在のホストの Web シナリオ一覧です。

Note:
設定セクションは、Admin および Super admin ユーザーのみが利用可能です。

リンクセクションには、次へのリンクがあります。

- 設定されたtrigger URL にアクセスします。
- Global scripts で設定されたカスタムリンクにアクセスします
  (スコープが *Manual host action*、タイプが'URL' のもの)。

スクリプトセクションでは、現在のホスト用に設定されたglobal scripts を実行できます。
これらのスクリプトは、ホストメニューで利用可能にするために、スコープが *Manual host action* として定義されている必要があります。

サポートされている場所

ホストメニューは、さまざまな Web インターフェースのセクションでホスト名をクリックすると表示されます。例えば、以下の場所です。

- ダッシュボードのウィジェット（障害、上位アイテム、トリガーの概要など）
- 監視 → 障害
- 監視 → 障害 → イベントの詳細
- 監視 → ホスト
- 監視 → ホスト → Web 監視
- 監視 → 最新データ
- 監視 → マップ
- インベントリ → ホスト
- レポート → 上位 100 トリガー

ホストメニューは、データ収集 → ホスト で三点アイコンをクリックしても表示できます。

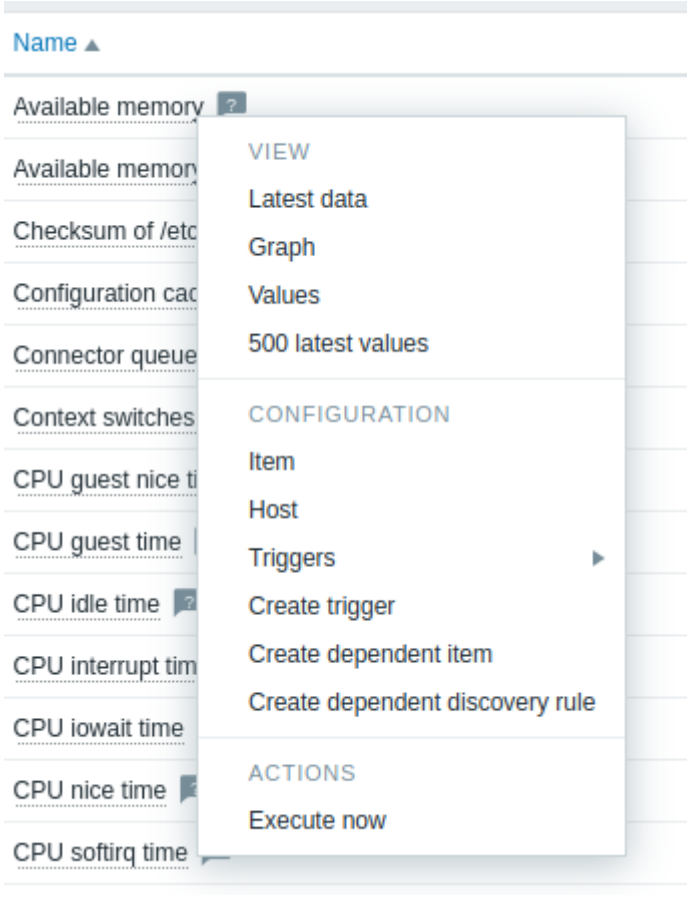
3 アイテムメニュー

概要

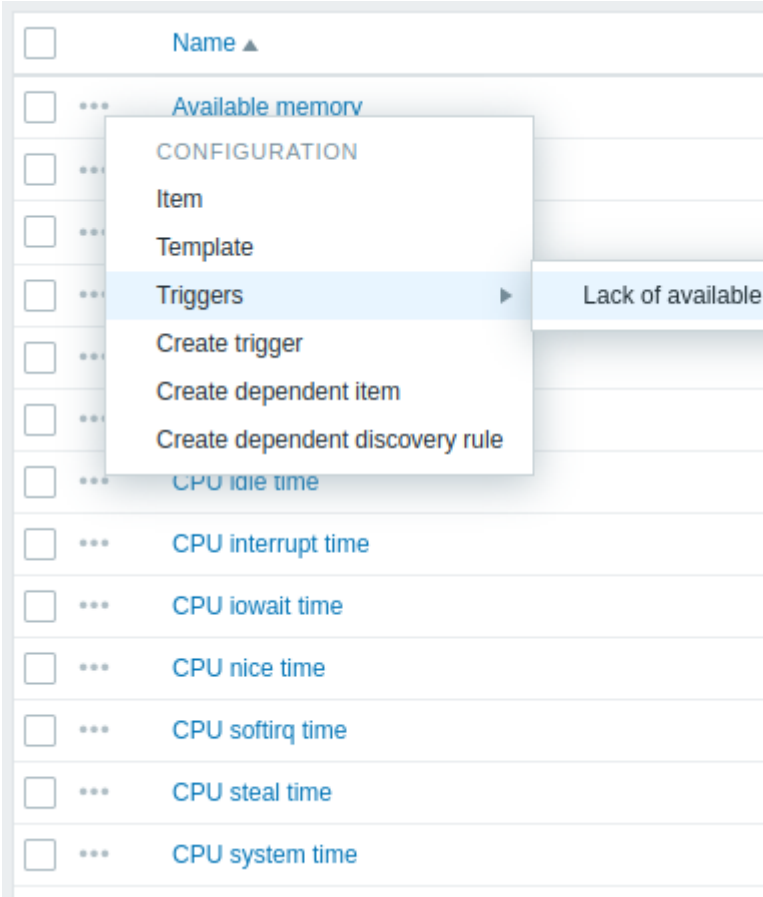
アイテムメニューには、アイテムで頻繁に必要となるアクションまたはフロントエンドセクションへのショートカットが含まれています。

アイテムメニューは、フロントエンドセクションに応じて、アイテム名または 3 つのドットのアイコンをクリックして開くことができます。例:

監視データ → 最新データ



データ収集 → テンプレート → アイテム



内容

アイテムメニューには、表示、設定、アクションの 3 つのセクションがあります。

表示セクションには、次のオプションがあります。

- 最新データ - 現在のホストとアイテムでフィルタリングされた最新データ セクションを開きます。
- グラフ - 現在のアイテムのシンプルグラフ を開きます。
- 値 - 過去 60 分以内に現在のアイテムで受信したすべての値 の一覧を開きます。
- 最新 **500** 件の値 - 現在のアイテムの最新 500 件の値 の一覧を開きます。

設定セクション（*Admin* および *Super admin* タイプのユーザーのみ利用可能）には、次のオプションがあります。

- アイテム - 現在のアイテムのアイテム設定フォーム を開きます。
- テンプレート - アイテムが属するテンプレートのテンプレート設定フォーム を開きます（アイテムメニューに データ収集 → テンプレート → アイテム からアクセスした場合のみ利用可能）。
- ホスト - アイテムが属するホストのホスト設定フォーム を開きます。
- トリガー - マウスオーバーすると、そのアイテムに関連付けられたトリガーの一覧が表示されます（存在する場合）。トリガーをクリックすると、そのトリガー設定フォーム を開きます。
- トリガーの作成 - このアイテムのトリガーを作成するためのトリガー設定フォーム を開きます。
- 依存アイテムの作成 - 現在のアイテムをマスターアイテムとして、依存アイテムを作成するためのアイテム設定フォーム を開きます。
- 依存ディスカバリルールの作成 - 現在のアイテムをマスターアイテムとして、依存ディスカバリルールを作成するためのディスカバリルール設定フォーム を開きます。

アクションセクションには、次のオプションがあります。

- 今すぐ実行 - 新しいアイテムの値を直ちに取得するために、チェックを実行 します。

サポートされる場所

アイテムメニューは、たとえば次のようなさまざまな Web インターフェースのセクションでアイテム名をクリックすると表示できます。

- 監視 → 最新データ
- データ収集 → ホスト → アイテム

- データ収集 → ホスト → ディスカバリルール → アイテムのプロトタイプ

アイテムメニューは、Top items ダッシュボードウィジェットでアイテムの値をクリックしても表示できます。

2 Web インターフェースセクション

メニュー構成　Zabbix Web インターフェースのメニューは、次の構成になっています。

- ダッシュボード
- 監視
  - 障害
  - ホスト
  - 最新データ
  - マップ
  - ディスカバリ
- サービス
  - サービス
  - SLA
  - SLA レポート
- インベントリ
  - 概要
  - ホスト
- レポート
  - システム情報
  - 定期レポート
  - 可用性レポート
  - トップ 100 トリガー
  - 監査ログ
  - アクションログ
  - 通知
- データ収集
  - テンプレートグループ
  - ホストグループ
  - テンプレート
  - ホスト
  - メンテナンス
  - イベント相関
  - ディスカバリ
- アラート
  - アクション
    - トリガーアクション
    - サービスアクション
    - ディスカバリアクション
    - 自動登録アクション
    - 内部アクション
  - メディアタイプ
  - スクリプト
- ユーザー
  - ユーザーグループ
  - ユーザーロール
  - ユーザー
  - API トークン
  - 認証
- 管理
  - 一般
    - GUI
    - 自動登録
    - タイムアウト
    - 画像
    - アイコンマッピング
    - 正規表現

        * トリガー表示オプション
        * 地理マップ
        * モジュール
        * コネクタ
        * その他
    - 監査ログ
    - ハウスキーピング
    - プロキシグループ
    - プロキシ
    - マクロ
    - キュー
        * キュー概要
        * プロキシごとのキュー概要
        * キュー詳細

1 ダッシュボード

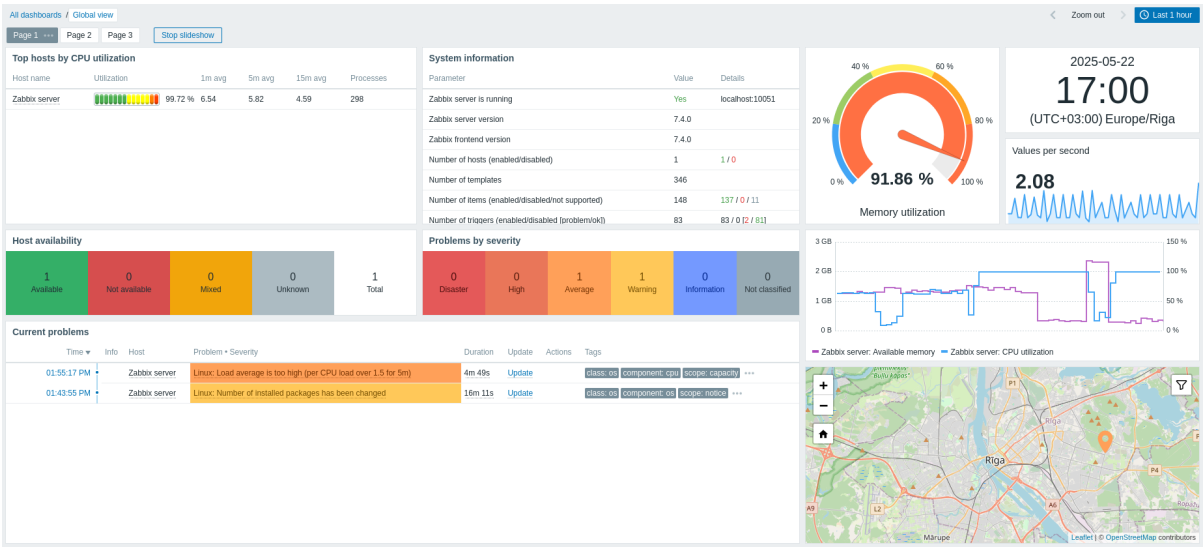
概要

ダッシュボードセクションは、ダッシュボード内のすべての重要な情報の概要を表示するために設計されています。

一度に表示できるダッシュボードは 1 つだけですが、複数のダッシュボードを設定することが可能です。各ダッシュボードには、スライドショーとして切り替え表示できる 1 つまたは複数のページを含めることができます。

ダッシュボードページはウィジェットで構成されており、各ウィジェットは概要、マップ、グラフ、時計など、特定の種類およびソースの情報を表示するように設計されています。

ウィジェット内のホストへのアクセスは、ホストの権限に依存します。



ページとウィジェットはダッシュボードに追加し、ダッシュボードの編集モードで編集します。ページの表示および切り替えは、ダッシュボードの表示モードで行えます。

グラフウィジェットに表示される期間は、ウィジェットの上部にある期間セレクターによって制御されます。右側にある 期間セレクターのラベルには、現在選択されている期間が表示されます。タブラベルをクリックすると、期間セレクターを展開または折りたたむことができます。

ダッシュボードのサイズ

ダッシュボードの最小幅は 1200 ピクセルです。ダッシュボードはこの幅より小さくはならず、代わりにブラウザーウィンドウがそれより小さい場合は水平方向のスクロールバーが表示されます。

ダッシュボードの最大幅はブラウザーウィンドウの幅です。ダッシュボードウィジェットは、ウィンドウに合わせて水平方向に伸縮します。同時に、ダッシュボードウィジェットをウィンドウの範囲を超えて水平方向に拡張することはできません。

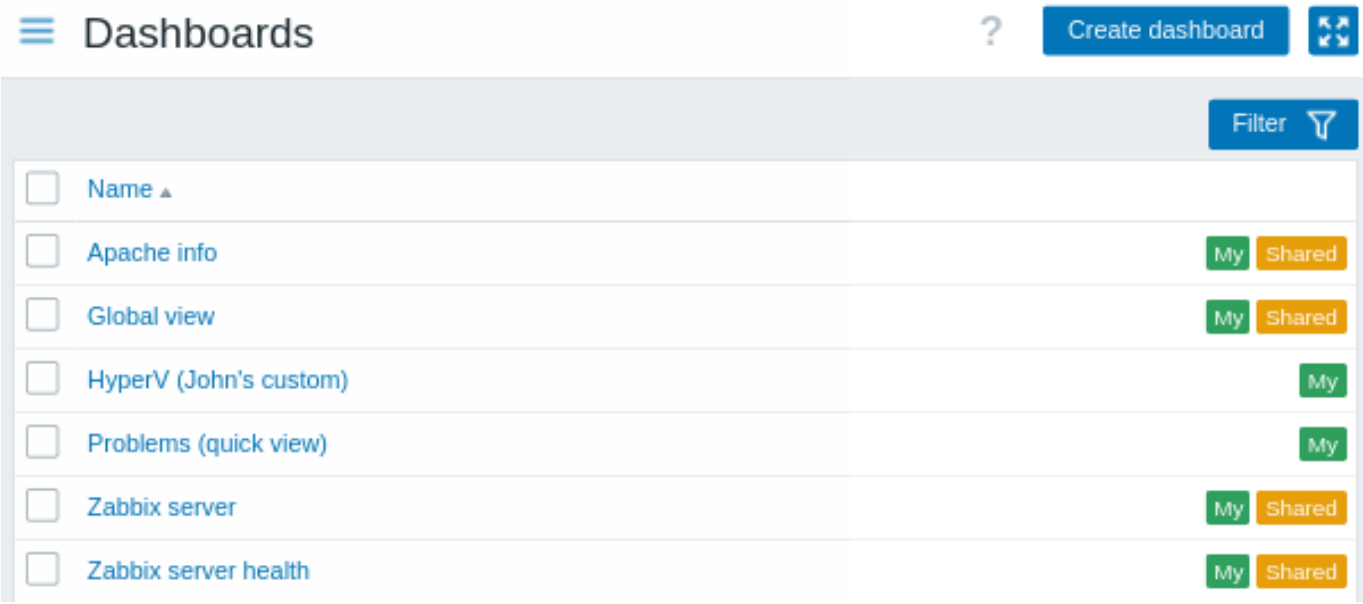
水平方向では、ダッシュボードは常に同じ幅の 72 列で構成されており、これらの列は動的に拡大・縮小します（ただし、合計幅が 1200 ピクセル未満にはなりません）。

垂直方向では、ダッシュボードには最大 64 行を含めることができ、各行の高さは固定で 70 ピクセルです。

したがって、ウィジェットは最大で幅 72 列、高さ 64 行まで設定できます。

ダッシュボードの表示

設定済みのすべてのダッシュボードを表示するには、セクションタイトルのすぐ下にある すべてのダッシュボードをクリックします。



ダッシュボードには、共有 タグが表示されます。

- *My* - 現在のユーザーが所有するダッシュボードを示します
- *Shared* - 公開ダッシュボード、または任意のユーザーもしくはユーザーグループと共有されているプライベートダッシュボードを示します

共有タグは、現在のユーザーが所有するダッシュボードにのみ表示されることに注意してください。他のユーザーが所有するダッシュボードは、共有タグなしで一覧表示されます。

一覧の右上にあるフィルターでは、名前および現在のユーザーが作成したダッシュボードで絞り込むことができます。

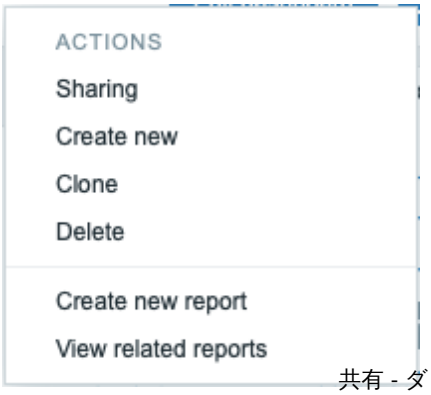
1 つまたは複数のダッシュボードを削除するには、対象のダッシュボードのチェックボックスを選択し、一覧の下にある 削除をクリックします。

ダッシュボードの表示

単一のダッシュボードを表示するには、ダッシュボードの一覧でその名前をクリックします。

ダッシュボードを表示中は、次のオプションを利用できます。

Edit dashboard	ダッシュボードの編集モードに切り替えます。 編集モードは、新しいダッシュボードの作成時や、ウィジェットの 編集ボタンをクリックしたときにも開きます。
	アクションメニューを開きます（アクションの説明は以下を参照）。



共有 - ダッシュボードの共有設定を編集します。
新規作成 - 新しいダッシュボードを作成します。
複製 - 既存のダッシュボードのプロパティをコピーして、新しいダッシュボードを作成します。最初にダッシュボードのパラメータを入力するよう求められます。その後、新しいダッシュボードが元のダッシュボードのすべてのウィジェットを含んだ編集モードで開きます。
削除 - ダッシュボードを削除します。
新しいレポートの作成 - レポートの設定フォームを含むポップアップウィンドウを開きます。ユーザーに定期レポートを管理する権限がない場合は無効です。
関連レポートの表示 - 現在のダッシュボードに基づく既存レポートの一覧を含むポップアップウィンドウを開きます。関連レポートが存在しない場合、またはユーザーに定期レポートを表示する権限がない場合は無効です。

ページの内容のみを表示します（キオスクモード）。
キオスクモードには、次の URL パラメータでもアクセスできます:
`/zabbix.php?action=dashboard.view&kiosk=1`。
通常モードに戻るには: `/zabbix.php?action=dashboard.view&kiosk=0`。

共有

ダッシュボードは公開または非公開にできます。

公開ダッシュボードは、すべてのユーザーに表示されます。非公開ダッシュボードは、その所有者と、共有設定で追加されたユーザー/ユーザーグループにのみ表示されます。

ダッシュボードの共有ステータスを編集するには、単一のダッシュボードを表示しているときに、アクションメニューの 共有オプションをクリックします。

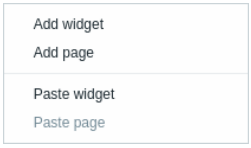
Parameter	Description
Type	ダッシュボードのタイプを選択します。 非公開 - ダッシュボードは、選択したユーザーグループとユーザーにのみ表示されます。 公開 - ダッシュボードは、すべてのユーザーに表示されます。
List of user group shares	ダッシュボードへのアクセスを許可するユーザーグループを選択します。 読み取り専用または読み書き可能のアクセスを許可できます。
List of user shares	ダッシュボードへのアクセスを許可するユーザーを選択します。 読み取り専用または読み書き可能のアクセスを許可できます。

共有オプションによってユーザーが実行できる操作がどのように制限されるかについては、ダッシュボードに対する権限を参照してください。

ダッシュボードの編集

ダッシュボードを編集中は、以下のオプションを利用できます。

ダッシュボードの一般パラメータを編集します。
新しいウィジェットを追加します。
矢印ボタンをクリックすると、アクションメニューが開きます（アクションの説明は以下を参照）。

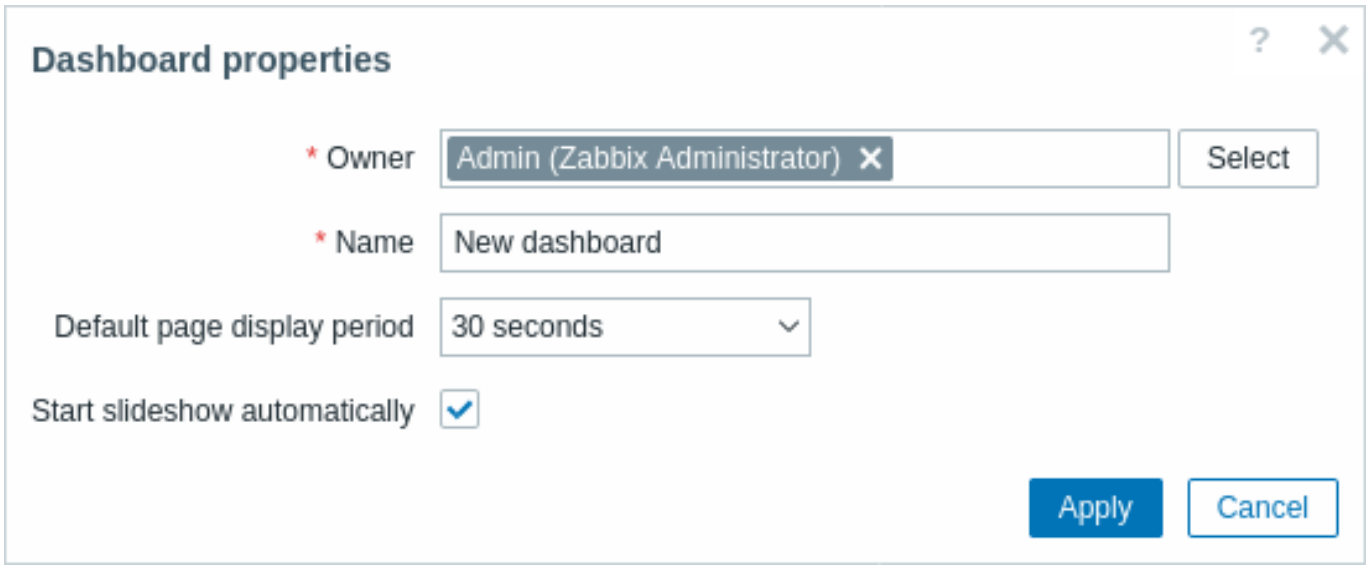
	ウィジェットの追加 - 新しいウィジェットを追加します。 ページの追加 - 新しいページを追加します。 ウィジェットの貼り付け - コピーしたウィジェットを貼り付けます。ウィジェットがコピーされていない場合、このオプションはグレーアウトされます。一度にコピーできるエンティティは 1 つ（ウィジェットまたはページ）のみです。 ページの貼り付け - コピーしたページを貼り付けます。ページがコピーされていない場合、このオプションはグレーアウトされます。
	ダッシュボードの変更を保存します。
	ダッシュボードの変更をキャンセルします。

ダッシュボードの作成

新しいダッシュボードは、次の 2 つの方法で作成できます。

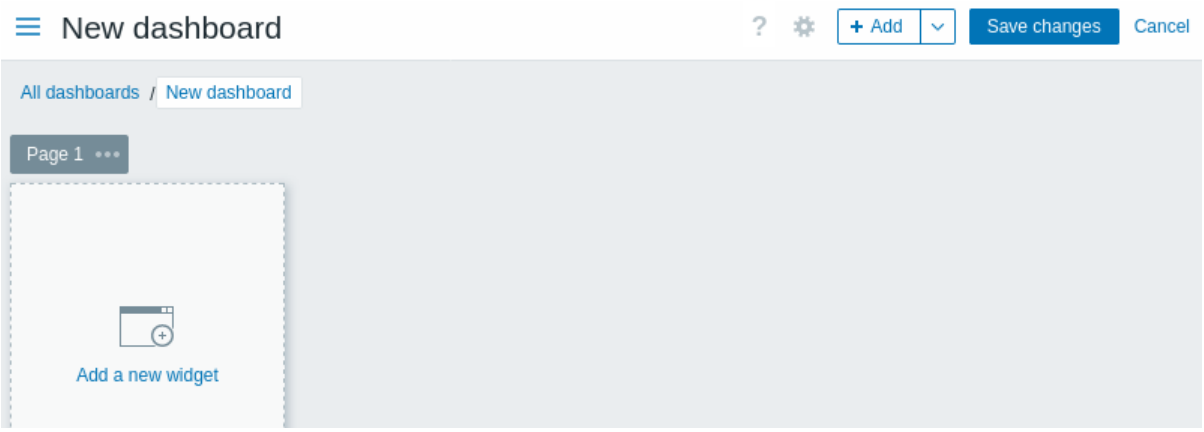
- すべてのダッシュボードを表示しているときに、*Create dashboard* をクリックします
- 単一のダッシュボードを表示しているときに、アクションメニューから *Create new* を選択します

最初に、ダッシュボードの一般的なパラメータを入力するよう求められます。



Parameter	Description
Owner	ダッシュボードの所有者となるシステムユーザーを選択します。
Name	ダッシュボード名を入力します。
Default page display period	スライドショーで次のページに切り替わるまで、ダッシュボードページを表示する期間を選択します。
Start slideshow automatically	ダッシュボードページが複数存在する場合に、スライドショーを自動的に実行するには、このチェックボックスをオンにします。

Apply をクリックすると、空のダッシュボードが開きます。



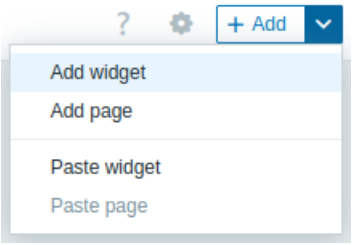
ダッシュボードに内容を追加するには、ウィジェットとページを追加できます。

ダッシュボードを保存するには、*Save changes* ボタンをクリックします。*Cancel* をクリックすると、ダッシュボードは作成されません。

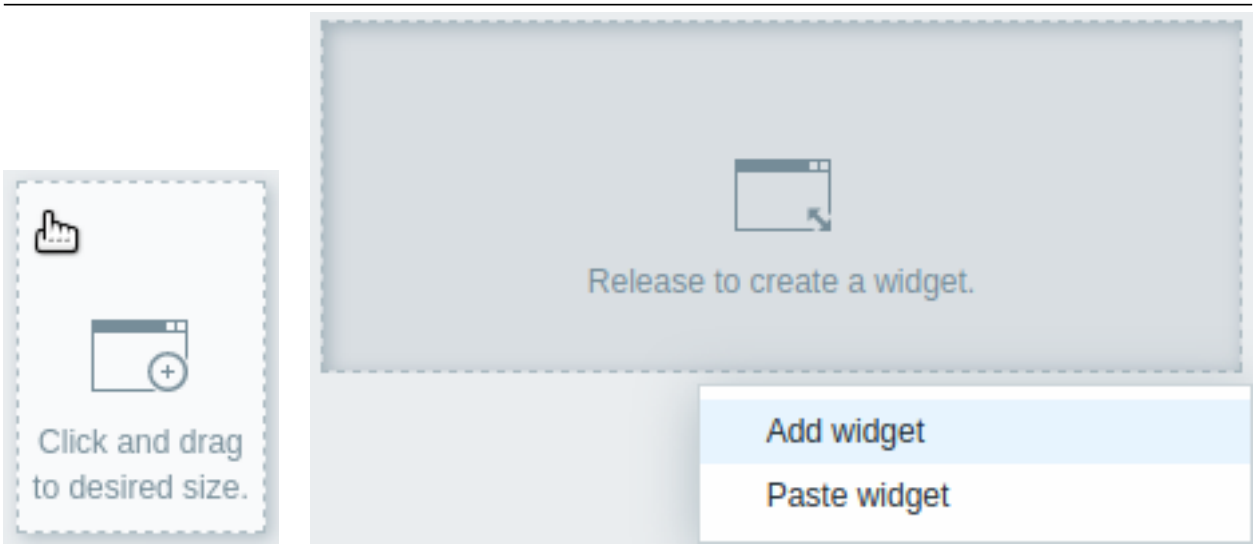
ウィジェットの追加

ダッシュボードには、さまざまなウィジェット（アクションログ、時計、ディスカバリのステータスなど）を追加できます。ウィジェットは、次の 2 つの方法で追加できます。

- 追加ボタン、またはその横の矢印をクリックし、アクションメニューからウィジェットの追加を選択します。ウィジェットを設定すると、デフォルトサイズで追加され、既存のウィジェットの後ろに配置されます。

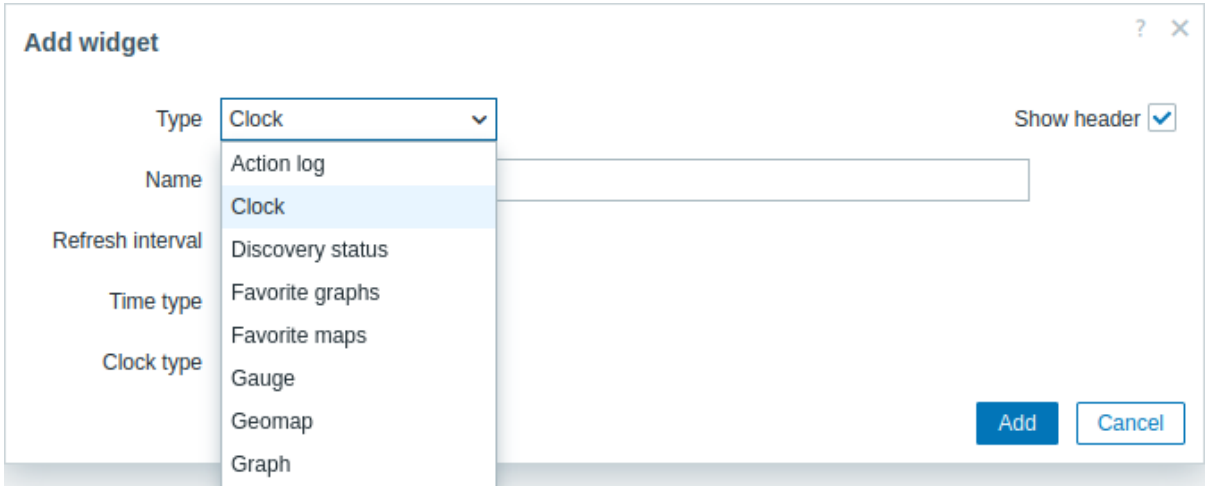


- ダッシュボード上の空いている場所にマウスカーソルを移動します。プレースホルダーが表示されたら、それをクリックしてウィジェット設定フォームを開きます。ウィジェットを設定すると、デフォルトサイズで追加されるか、利用可能なスペースに合わせてサイズ変更されます。または、プレースホルダーをドラッグしてウィジェットのサイズを決め、マウスボタンを離してウィジェット設定フォームを開くこともできます（ウィジェットがコピーされてクリップボードにある場合は、最初にウィジェットの追加またはウィジェットの貼り付けのどちらかを選択するよう求められます）。



ウィジェット設定フォームでは、次の操作を行います。

1. ウィジェットの種類を選択します。
2. ウィジェットのパラメータを設定します。
3. 追加をクリックして、ウィジェットをダッシュボードに追加します。



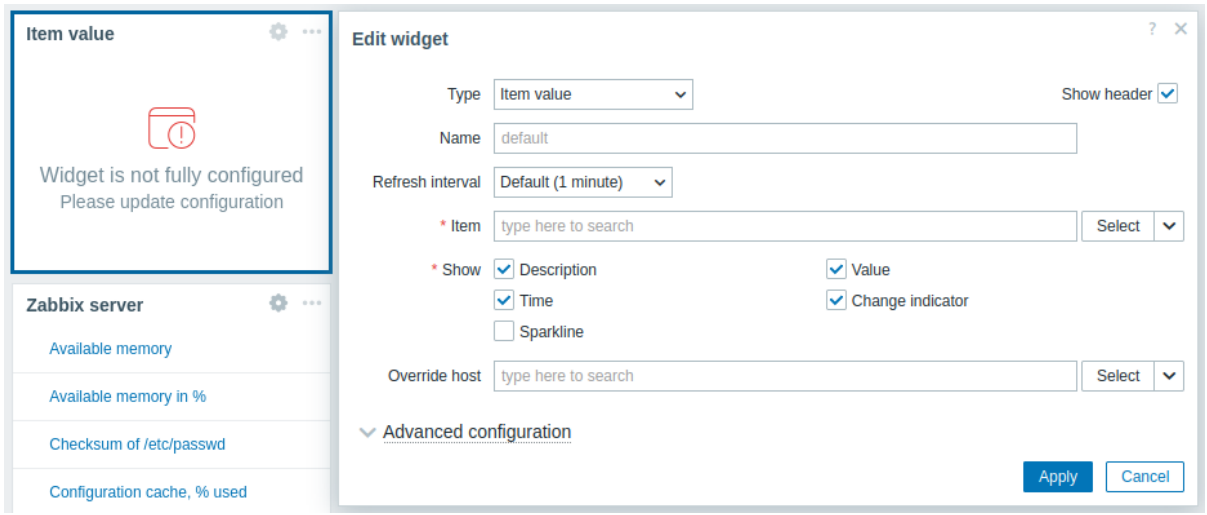
ウィジェットの編集

ウィジェットは、ダッシュボードの編集モードでタイトルバーをドラッグして、サイズ変更や再配置ができます。また、タイトルバーをドラッグしてウィジェット設定フォームの位置を変更することもできます。

各ウィジェットの右上には、次のコントロールがあります。

- - ウィジェットを編集します
- - ウィジェットメニューを開きます

編集中は、ウィジェットが強調表示され、変更内容がリアルタイムでプレビューされます。必須のウィジェットパラメータが設定されていない場合、ウィジェットはウィジェットが完全に設定されていません状態になります。



編集後、変更を反映するには、ウィジェット内の適用をクリックし、その後ダッシュボード右上の変更を保存をクリックします。

ウィジェットのコピー/貼り付け

ダッシュボードウィジェットをコピーして貼り付けることで、同じ設定を持つ新しいウィジェットをすばやく作成できます。ウィジェットは、同じダッシュボード内でも、異なるタブで開いているダッシュボード間でもコピー/貼り付けできます。

ウィジェットをコピーするには、ウィジェットメニューを使用します。コピーしたウィジェットを貼り付けるには（ダッシュボードの編集モードで使用可能）：

- 追加ボタンの横にある矢印をクリックし、ウィジェットを貼り付けを選択します。
- または、ダッシュボード内の空いている領域をクリックし、ウィジェットを貼り付けを選択します。

また、既存のウィジェットのウィジェットメニューにある貼り付けオプションを使用して、そのウィジェットの上にコピーしたウィジェットを貼り付けることもできます。

スライドショーの作成

ダッシュボードに 2 つ以上のページが含まれている場合（ページの追加を参照）で、かつ次のいずれかの条件を満たしている場合、スライドショーは自動的に実行されます。

- ダッシュボードのプロパティで *Start slideshow automatically* オプションが選択されている
- ダッシュボード URL に `slideshow=1` パラメータが含まれている

ページは、ダッシュボードおよび各ページのプロパティで指定された間隔に従って切り替わります。次をクリックします。

- *Stop slideshow* - スライドショーを停止します
- *Start slideshow* - スライドショーを開始します



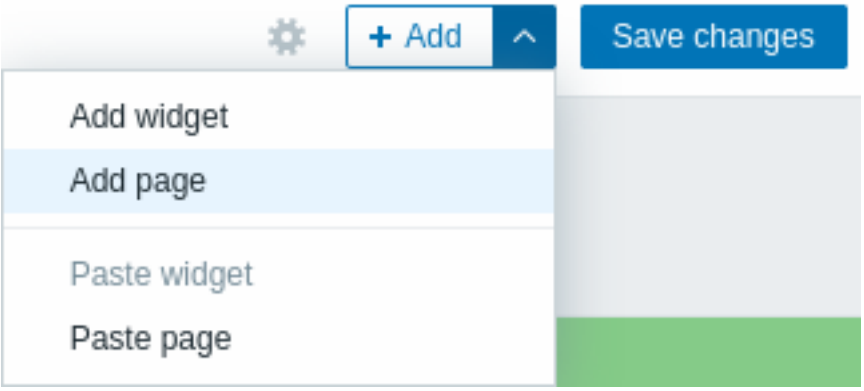
スライドショー関連のコントロールは、キオスクモード（ページの内容のみが表示されるモード）でも利用できます。

-  - スライドショーを停止
-  - スライドショーを開始
-  - 1 ページ戻る
-  - 次のページへ移動

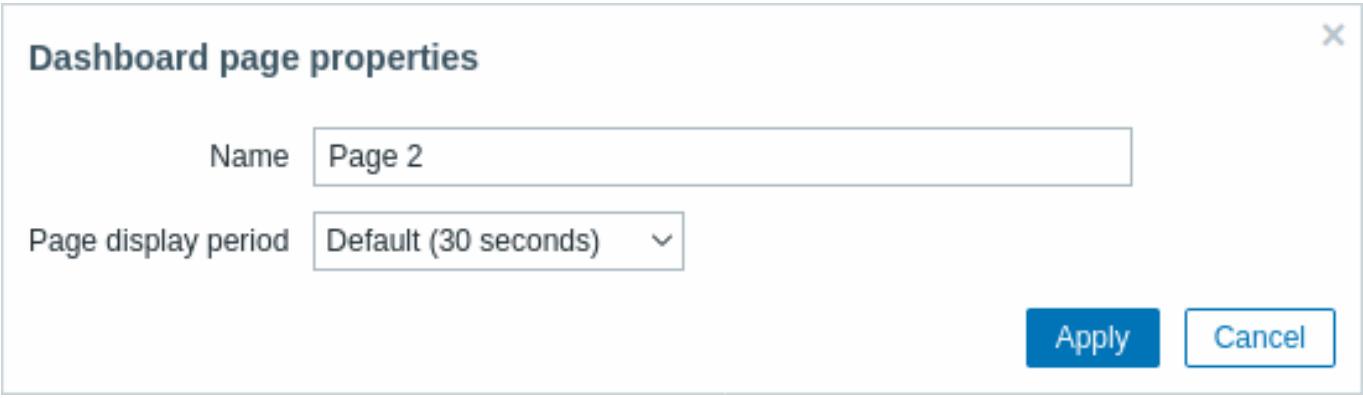
ページの追加

ダッシュボードに新しいページを追加するには、次のようにします。

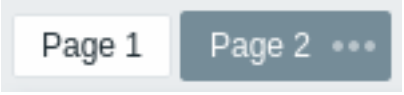
- ダッシュボードが編集モードになっていることを確認します
- 追加ボタンの横にある矢印をクリックし、ページの追加オプションを選択します



- ページの一般パラメータを入力し、適用をクリックします。名前を空のままにすると、ページは Page N という名前で追加されます。ここで、'N' はページの連番です。ページの表示期間では、スライドショーでページを表示する時間をカスタマイズできます。



新しいページが追加され、新しいタブ（*Page 2*）で示されます。



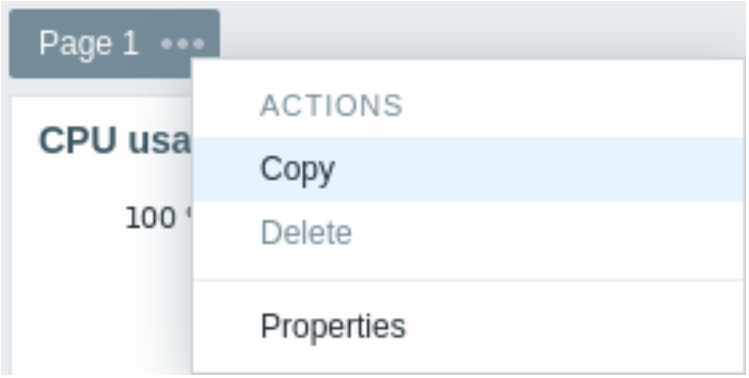
ページは、ページタブをドラッグ＆ドロップして並べ替えることができます。並べ替えても、元のページ名は維持されます。各ページには、そのタブをクリックすることでいつでも移動できます。

新しいページを追加した時点では、そのページは空です。上記の説明に従って、そこにウィジェットを追加できます。

ページのコピー/貼り付け

ダッシュボードのページはコピーして貼り付けることができ、既存のページのプロパティを持つ新しいページを作成できます。貼り付け元は、同じダッシュボードでも別のダッシュボードでもかまいません。

既存のページをダッシュボードに貼り付けるには、まずページメニューを使用してそのページをコピーします。

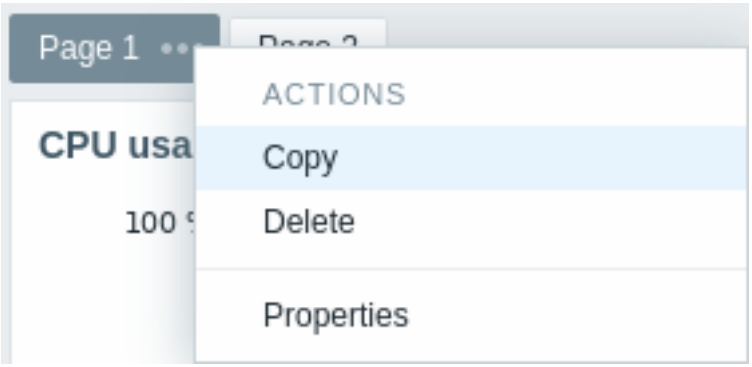


コピーしたページを貼り付けるには、次のようにします。

- ダッシュボードが編集モードになっていることを確認します
- 追加ボタンの横にある矢印をクリックし、ページを貼り付けオプションを選択します

ページメニュー

ページメニューは、ページ名の横にある  3 つのドットをクリックすると開くことができます。

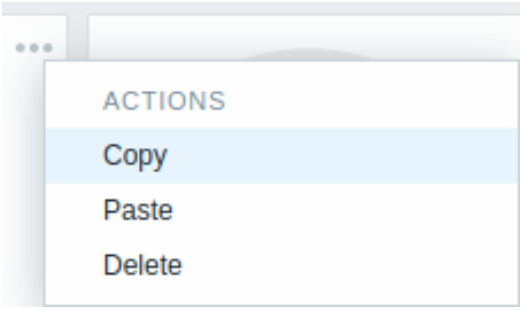


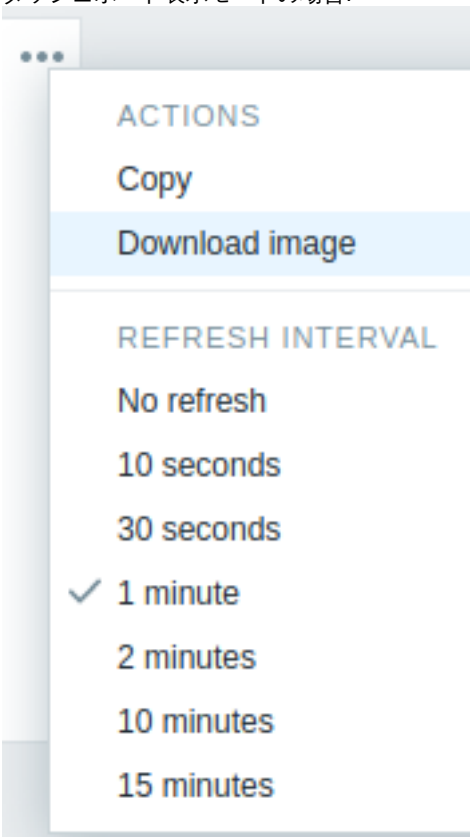
このメニューには、次のオプションが含まれます。

- コピー - ページをコピーします
- 削除 - ページを削除します（ページはダッシュボードの編集モードでのみ削除できます）
- プロパティ - ページのパラメータ（名前およびスライドショーでのページ表示期間）をカスタマイズします

ウィジェットメニュー

ウィジェットメニューには、ダッシュボードが編集モードか表示モードかに応じて、異なるオプションが含まれます。

ウィジェットメニュー	オプション
ダッシュボード編集モードの場合:  	コピー - ウィジェットをコピーします。 貼り付け - コピーしたウィジェットをこのウィジェットの上に貼り付けます。 ウィジェットがコピーされていない場合、このオプションはグレー表示されます。 削除 - ウィジェットを削除します。

ウィジェットメニュー	オプション
ダッシュボード表示モードの場合: 	コピー - ウィジェットをコピーします。 画像をダウンロード - ウィジェットを PNG 画像としてダウンロードします。 このオプションは、この機能をサポートするウィジェットで使用できます (各ウィジェットのページを参照してください)。 ブラウザーの設定によっては、画像は自動的にダウンロードフォルダーに保存されるか、保存ダイアログが開いて保存先を選択できます。 更新間隔 - ウィジェットの内容を更新する頻度を選択します。

ダッシュボードに対する権限

通常ユーザーおよび「Admin」タイプのユーザーに対するダッシュボードの権限は、割り当てられているユーザーロールでダッシュボードへのアクセスとダッシュボードの作成と編集オプションが有効になっている場合でも、以下のように制限されます。

- ダッシュボードの共有設定にユーザーまたはユーザーグループが追加されていない場合でも、公開ダッシュボードを表示および複製できます。
- 共有設定によって少なくとも *Read* 権限が設定されている場合、非公開ダッシュボードを表示および複製できます。
- 共有設定によって *Read-write* 権限が設定されている場合にのみ、ダッシュボードを編集および削除できます。
- ダッシュボードの所有者を変更することはできません。

1 ダッシュボードウィジェット

概要

このセクションでは、ダッシュボードウィジェットに共通するパラメータの詳細を説明します。

共通パラメータ

以下のパラメータは、すべてのウィジェットに共通です。

名前	ウィジェット名を入力します。

更新間隔	デフォルトの更新間隔を設定します。 ウィジェットのデフォルトの更新間隔は、ウィジェットの種類に応じて 更新なしから *15* 分までの範囲で設定されています。例えば、 - *URL* ウィジェットでは 更新なし、 - アクションログウィジェットでは *1* 分、 - 時計ウィジェットでは *15* 分です。 更新間隔は、すべてのユーザーに対してデフォルト値を設定できます。ダッシュボードを編集モードに切り替え、ウィジェットの編集ボタンをクリックして、ドロップダウンリストから希望する更新間隔を選択します。 各ユーザーは、自分用のウィジェット更新間隔を設定することもできます。ダッシュボードの表示モードで、ウィジェット上の三点リーダー ボタンをクリックし、ドロップダウンリストから希望する更新間隔を選択します。ユーザー固有の更新間隔はウィジェット設定より優先され、ウィジェットの設定が変更されても保持されることに注意してください。
ヘッダーを表示	チェックボックスをオンにすると、ウィジェットのヘッダーを常に表示します。 オフの場合、スペースを節約するためにヘッダーは非表示になり、ウィジェットにマウスオーバーしたときのみ表示されます（表示モードと編集モードの両方）。また、ウィジェットを新しい位置にドラッグしている間は、ヘッダーが半透明で表示されます。

個別パラメータ

各ウィジェット固有のパラメータについては、各ウィジェットのページを参照してください。

- アクションログ
- 時計
- ディスカバリステータス
- お気に入りグラフ
- お気に入りマップ
- ゲージ
- ジオマップ
- グラフ
- グラフ（クラシック）
- グラフプロトタイプ
- ハニカム
- ホストの可用性
- ホストカード
- ホストナビゲーター
- アイテムカード
- アイテム履歴
- アイテムナビゲーター
- アイテム値
- マップ
- マップナビゲーションツリー
- 円グラフ
- 障害ホスト
- 障害
- 深刻度別の障害
- SLA レポート
- システム情報
- トップホスト
- トップアイテム
- トップトリガー
- トリガー概要
- URL
- Web 監視

動的パラメータ

複数のウィジェットには、他のウィジェットまたはダッシュボードとの間で設定データを共有できるようにするパラメータがあります。

ホストグループ、ホスト、アイテム、アイテムリスト

ホストグループ、ホスト、アイテム、およびアイテムリストパラメータでは、対応するエンティティ、またはウィジェットがデータを表示できるホストグループ、ホスト、またはアイテムを含むデータソースのいずれかを選択できます。

ホストグループ、アイテム、およびアイテムリストパラメータでは、データソースとして同じダッシュボード内の互換性のあるウィジェットを指定できます。

ホストパラメータでは、データソースとして同じダッシュボード内の互換性のあるウィジェット、またはダッシュボード自体を指定できます。

Note:
マップウィジェットも、ホストグループおよびホストのデータを互換性のあるウィジェットにブロードキャストできます。詳細については、ウィジェットの動作を参照してください。

ホストの上書き

ホストの上書きパラメータでは、ウィジェットがデータを表示する対象のホストを含むデータソースを選択できます。データソースには、同じダッシュボード上の互換性のあるウィジェット、またはダッシュボード自体を指定できます。

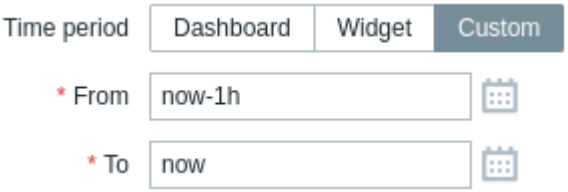


- 互換性のあるウィジェットを指定するには、その名前を入力して選択します。または、選択ボタン（またはドロップダウンボタンをクリックしてから「ウィジェット」）をクリックすると、利用可能なウィジェットのポップアップが開きます。
- ダッシュボードを指定するには、ドロップダウンボタンをクリックしてから「ダッシュボード」を選択します。ダッシュボードを保存すると、ダッシュボードの上部にホストフィールド（ホストを選択するためのフィールド）が表示されます。



期間

期間パラメータでは、ウィジェットがデータを表示する期間を含むデータソースを選択できます。データソースには、同じダッシュボード上の互換性のあるウィジェット、ダッシュボード自体、またはウィジェット自体に設定された期間を指定できます。



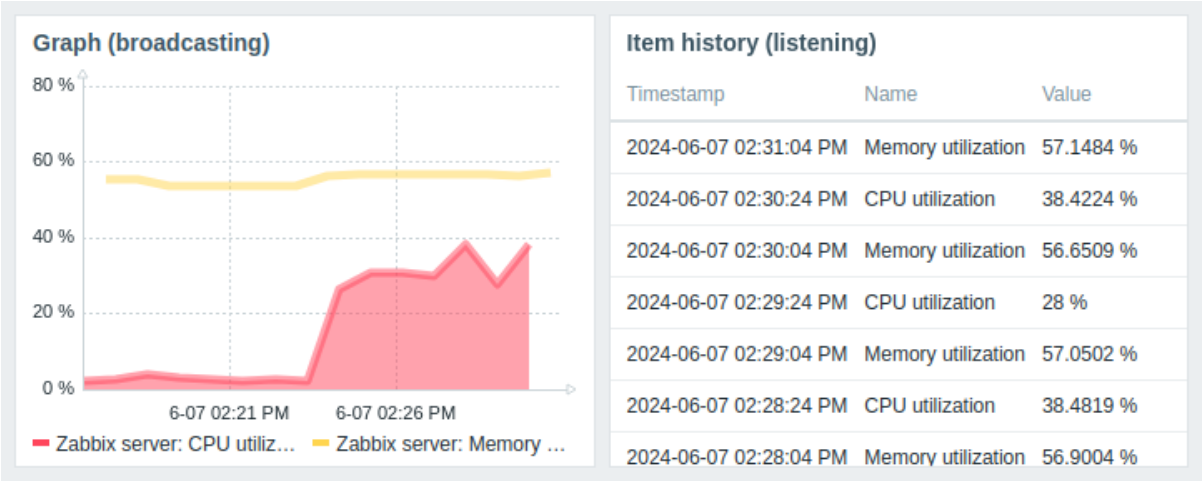
- 互換性のあるウィジェットを指定するには、期間を「ウィジェット」に設定し、その名前を入力して選択します。または、選択ボタンをクリックして、利用可能なウィジェットのポップアップを開きます。
- ダッシュボードを指定するには、期間を「ダッシュボード」に設定します。ダッシュボードを保存すると、ダッシュボードの上部に期間セレクターが表示されます。
- ウィジェット自体で期間を設定するには、期間を「カスタム」に設定し、期間の開始時刻と終了時刻を入力または選択します。

Note:
ウィジェットの期間設定に関係なく、互換性のあるウィジェットはそのウィジェットを期間のデータソースとして使用できます。

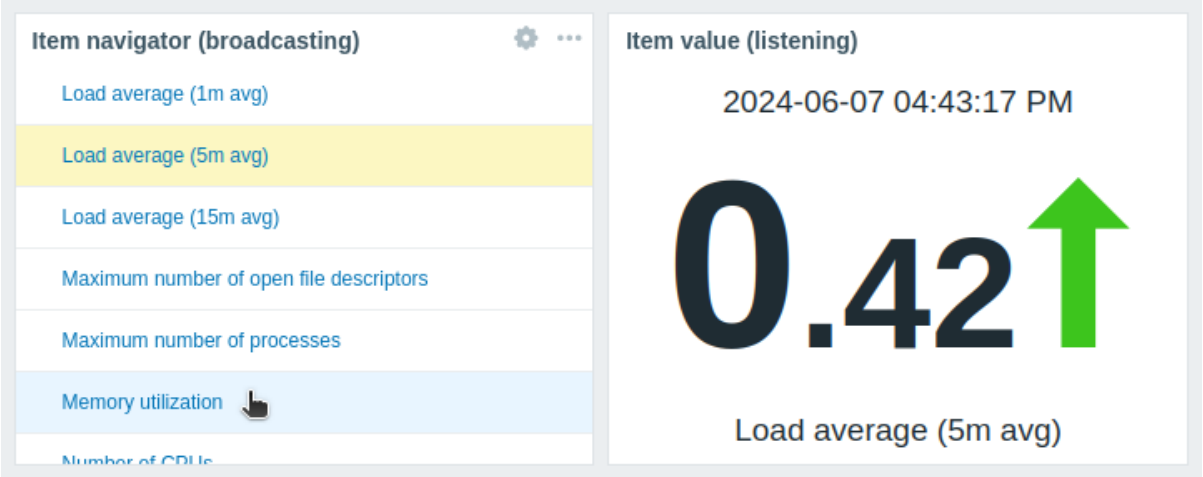
ウィジェットの動作

ウィジェットは、データを他のウィジェットにブロードキャストする方法がそれぞれ異なります。

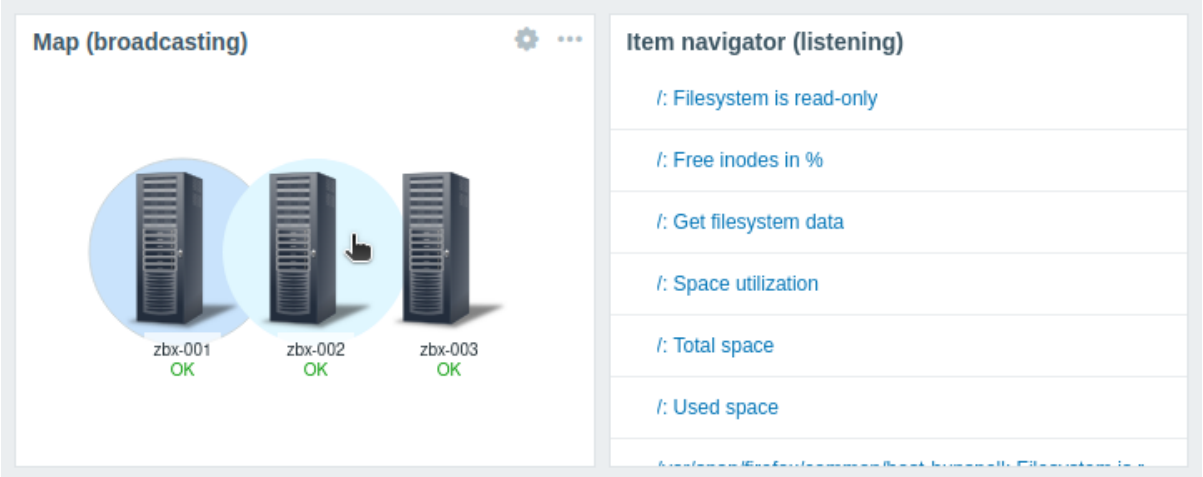
データをブロードキャストできるすべてのウィジェットは、作成されると自動的にブロードキャストを開始します。
例えば、グラフウィジェットは、リッスンしているウィジェットに対して期間データを即座にブロードキャストします。



エンティティ選択に対応したウィジェット（例: アイテムナビゲーター）は、最初に利用可能なエンティティのデータを自動的にブロードキャストします。
例えば、アイテムナビゲーターウィジェットは、アイテム一覧の最初のアイテムのデータをブロードキャストします。
別のアイテムを選択すると、そのアイテムのデータをブロードキャストします。
マウスオーバー時には、アイテムは薄い青色で強調表示され、選択時には黄色で強調表示されます。



マップウィジェットも、最初に利用可能な要素（例: ホスト）のデータを自動的にブロードキャストします。この要素は、ウィジェットのビューポートの左上隅に最も近いものです（ジオマップウィジェットの場合は、ウィジェットのビューポートの中央に最も近いもの）。
別の要素を選択すると、その要素のデータをブロードキャストします。
マウスオーバー時には、要素は薄い青色で強調表示され、選択時には濃い青色で強調表示されます。

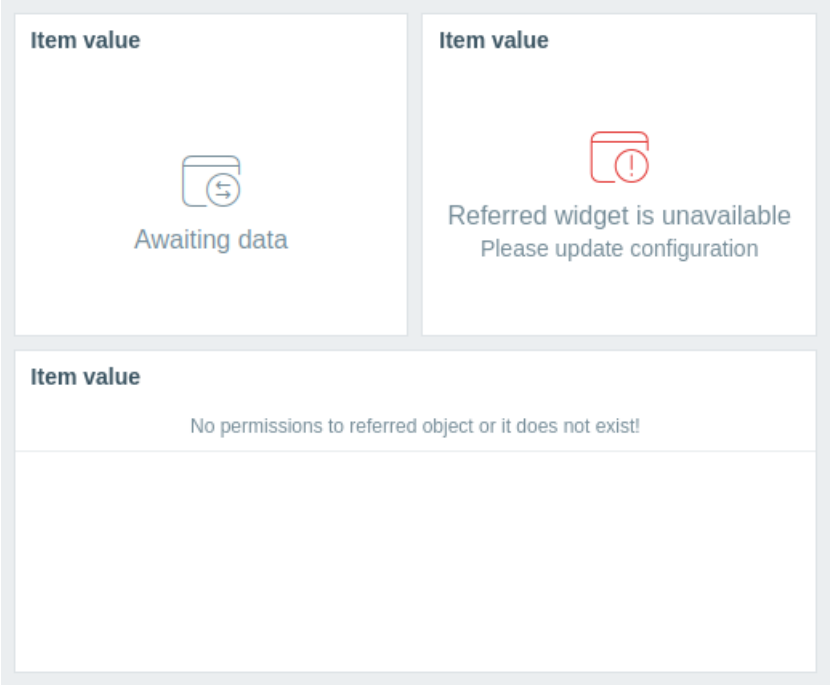


また、ウィジェットには、他のウィジェットからのデータをリッスンする際の固有の動作もあります。

- データソースウィジェットがデータをブロードキャストしていない場合、リッスン側のウィジェットはデータ待機中状態になります。
- データソースウィジェットが削除された場合、互換性のないウィジェットに置き換えられた場合、または別のダッシュボード

ページに移動された場合、リッスン側のウィジェットは参照先ウィジェットは利用できません状態になります。

- データソース（ウィジェットまたはダッシュボード）で指定されたホストに、リッスン側のウィジェットで設定されたエンティティ（アイテム、グラフ、マップなど）が存在しない場合、またはユーザーにそのホストへアクセスする権限がない場合、リッスン側のウィジェットには次のメッセージが表示されます: ”参照先オブジェクトへの権限がないか、存在しません!”



各ウィジェットのブロードキャスト機能およびリッスン機能については、ウィジェットの互換性を参照してください。

ウィジェットの互換性

一部のウィジェットは設定データを他のウィジェットにブロードキャストでき、一部はデータを受信でき、またその両方が可能なものもあります。例えば、次のとおりです。

- *Action log* ウィジェットは、*Graph*、*Graph (classic)*、および *Graph prototype* ウィジェットから時間範囲データのみを取得できます。
- *Geomap* ウィジェットは、ホストデータをそれを受信するウィジェット（*Honeycomb*、*Top items* など）にブロードキャストでき、また、それをブロードキャストするウィジェット（*Honeycomb*、*Problem hosts* など）からホストグループおよびホストデータを受信することもできます。
- *Clock* ウィジェットは、いかなるデータもブロードキャストまたは受信できません。

次の表は、各ウィジェットのブロードキャスト機能と受信機能を示しています。

ウィジェット	ブロードキャスト	受信
Action log	-	時間範囲
Clock	-	-
Discovery status	-	-
Favorite graphs	-	-
Favorite maps	-	-
Gauge	-	ホスト、アイテム
Geomap	ホスト	ホストグループ、ホスト
Graph	時間範囲	ホスト、アイテム、時間範囲
Graph (classic)	時間範囲	ホスト、アイテム、グラフ、時間範囲
Graph prototype	時間範囲	ホスト、時間範囲
Honeycomb	ホスト、アイテム	ホストグループ、ホスト
Host availability	-	ホストグループ
Host card	-	ホスト
Host navigator	ホスト	ホストグループ
Item card	-	ホスト、アイテム、時間範囲
Item history	アイテム	ホスト、時間範囲
Item navigator	アイテム	ホストグループ、ホスト
Item value	-	ホスト、アイテム、時間範囲
Map	ホストグループ、ホスト	マップ
Map navigation tree	マップ	-
Pie chart	-	アイテム、時間範囲

ウィジェット	ブロードキャスト	受信
Problem hosts	ホストグループ	ホストグループ、ホスト
Problems	イベント	ホストグループ、ホスト
Problems by severity	ホストグループ	ホストグループ、ホスト
SLA report	-	-
System information	-	-
Top hosts	ホスト	ホストグループ、ホスト
Top items	-	ホストグループ、ホスト
Top triggers	-	時間範囲
Trigger overview	-	ホストグループ、ホスト
URL	-	ホスト
Web monitoring	ホストグループ	ホストグループ、ホスト

1 アクションログ

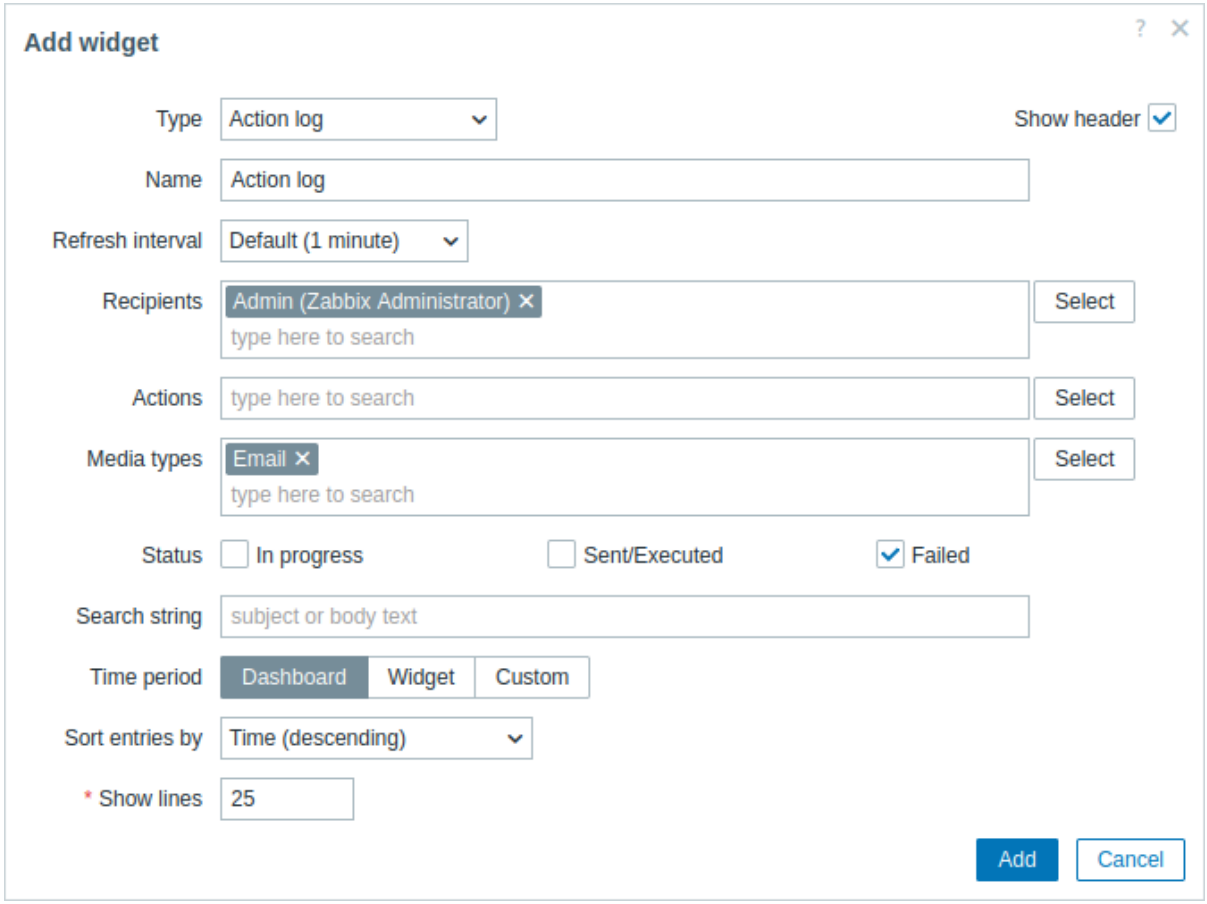
概要

アクションログウィジェットでは、アクション操作 (通知、リモートコマンド) の詳細を表示できます。レポート → アクションログの情報を複製します。

最大 1000 のレコードを表示できます。

設定

設定するには、タイプとして アクションログを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

Recipients	受信者でエントリを絞り込みます。このフィールドはオートコンプリートに対応しているため、受信者名の入力を開始すると、一致する受信者のドロップダウンが表示されます。受信者が選択されていない場合は、すべての受信者に対するアクション実行の詳細が表示されます。

Actions	アクションでエントリを絞り込みます。このフィールドはオートコンプリートに対応しているため、アクション名の入力を開始すると、一致するアクションのドロップダウンが表示されます。アクションが選択されていない場合は、すべてのアクションに対するアクション実行の詳細が表示されます。
Media types	メディアタイプでエントリを絞り込みます。このフィールドはオートコンプリートに対応しているため、メディアタイプ名の入力を開始すると、一致するメディアタイプのドロップダウンが表示されます。メディアタイプが選択されていない場合は、すべてのメディアタイプに対するアクション実行の詳細が表示されます。
Status	対応するステータスでエントリを絞り込むには、チェックボックスをオンにします。 進行中 - 進行中のアクション実行が表示されます。 送信済み/実行済み - 通知を送信した、または実行済みのアクション実行が表示されます。 失敗 - 失敗したアクション実行が表示されます。
Search string	メッセージ/リモートコマンドの内容でエントリを絞り込みます。ここに文字列を入力すると、入力した文字列を含むメッセージ/リモートコマンドを持つアクション実行のみが表示されます。マクロは展開されません。
Time period	期間でエントリを絞り込みます。期間のdata sourceを選択します。 **Dashboard** - *Time period* セレクターをデータソースとして設定します。 **Widget** - *Widget* パラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 **Custom** - *From* および *To* パラメータで指定した期間をデータソースとして設定します。これを設定すると、ウィジェット右上に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。
Widget	期間のデータソースとして、互換性のあるウィジェットを入力または選択します。 このパラメータは、*Time period* が”Widget” に設定されている場合に使用できます。
From	期間の開始を入力または選択します。 相対時間構文 (`now`, `now/d`, `now/w-1w` など) に対応しています。 このパラメータは、*Time period* が”Custom” に設定されている場合に使用できます。
To	期間の終了を入力または選択します。 相対時間構文 (`now`, `now/d`, `now/w-1w` など) に対応しています。 このパラメータは、*Time period* が”Custom” に設定されている場合に使用できます。
Sort entries by	エントリの並び替え条件を指定します。 **Time**（降順または昇順）； **Type**（降順または昇順）； **Status**（降順または昇順）； **Recipient**（降順または昇順）。
Show lines	ウィジェットに表示するアクションログの行数を設定します。

2 時計

概要

時計ウィジェットでは、ローカル、サーバー、または指定されたホスト時間を表示できます。

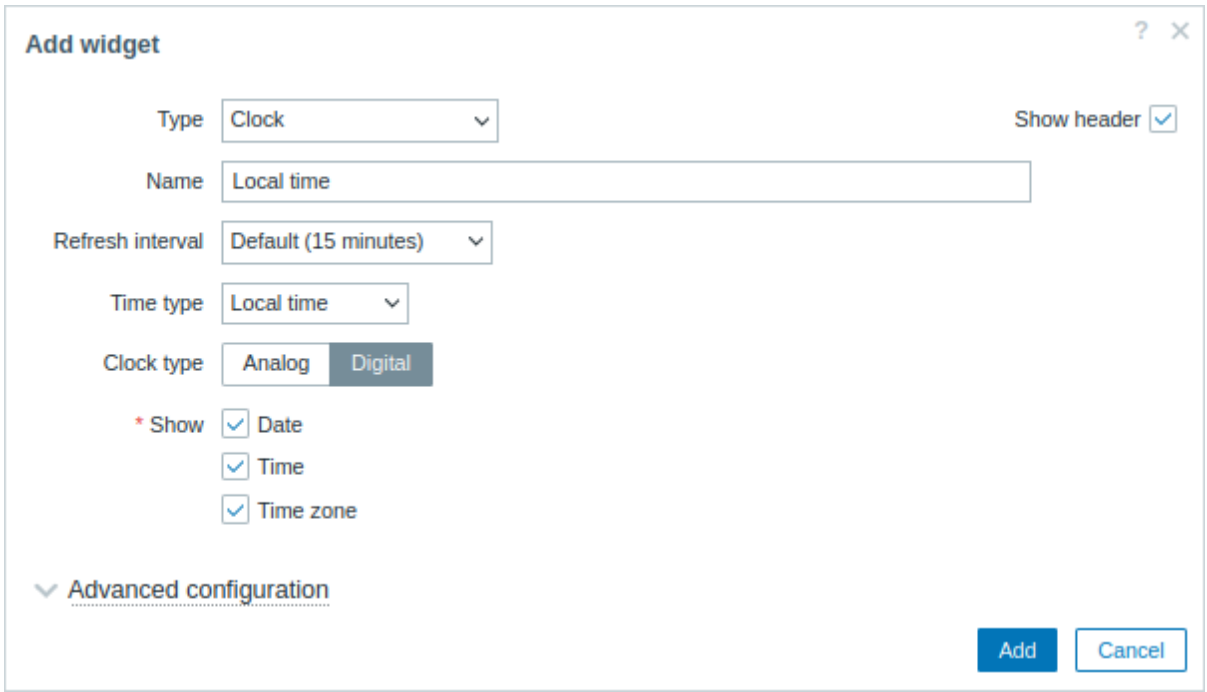
アナログ時計とデジタル時計の表示ができます。





設定

設定するには、タイプとして *Clock* を選択します。

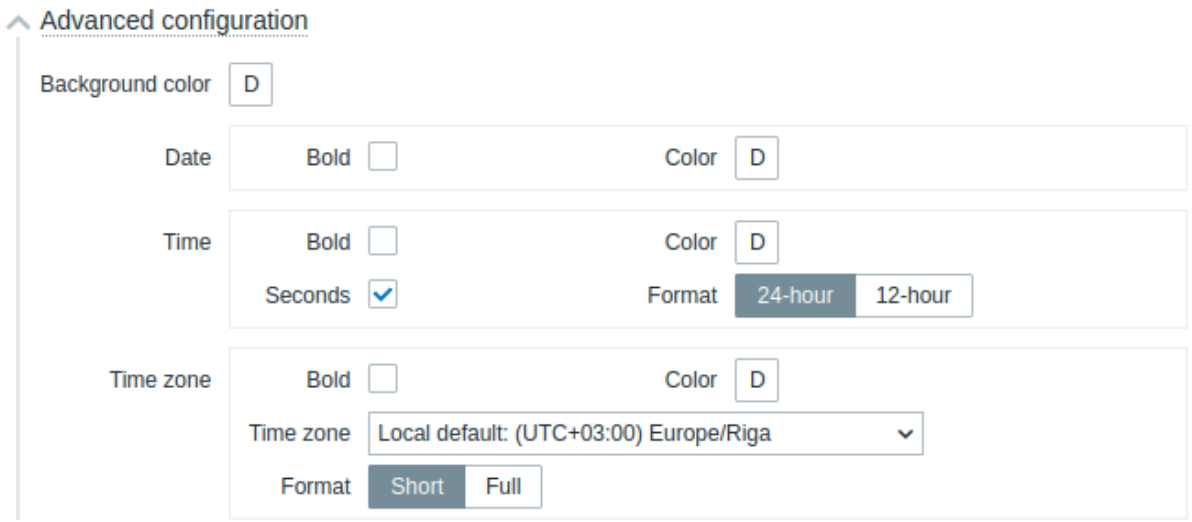


すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

Time type	ローカル時刻、サーバー時刻、または指定したホストの時刻を選択します。 サーバー時刻は、グローバルに設定されたタイムゾーン、または Zabbix ユーザーに設定されたタイムゾーンと同一になります。
Item	時刻を表示するためのアイテムを選択します。ホスト時刻を表示するには、`system.localtime[local]` アイテムを使用します。このアイテムはホスト上に存在している必要があります。 このフィールドは、*Host time* が選択されている場合にのみ使用できます。
Clock type	時計の種類を選択します。 **Analog** - アナログ時計 **Digital** - デジタル時計
Show	デジタル時計に表示する情報の単位（日付、時刻、タイムゾーン）を選択します。 単位のサイズは動的で、表示する単位、ウィジェットのサイズ、表示スケーリングに基づいて、利用可能なウィジェット領域全体を埋めるように調整されます。 このフィールドは、*Clock type* フィールドで”Digital” が選択されている場合にのみ使用できます。 また、少なくとも 1 つの情報単位タイプを選択する必要があります。
Advanced configuration	*Advanced configuration* ラベルをクリックすると、デジタル時計の高度な設定オプションが表示されます。 このセクションは、*Clock type* フィールドで”Digital” が選択されている場合にのみ使用できます。

詳細設定

詳細設定オプションは、折りたたみ可能な「詳細設定」セクションで利用可能です。ただし、これらは「表示」フィールドで選択された要素（上記参照）に対してのみ有効です。さらに、詳細設定ではウィジェット全体の背景色を変更できます。



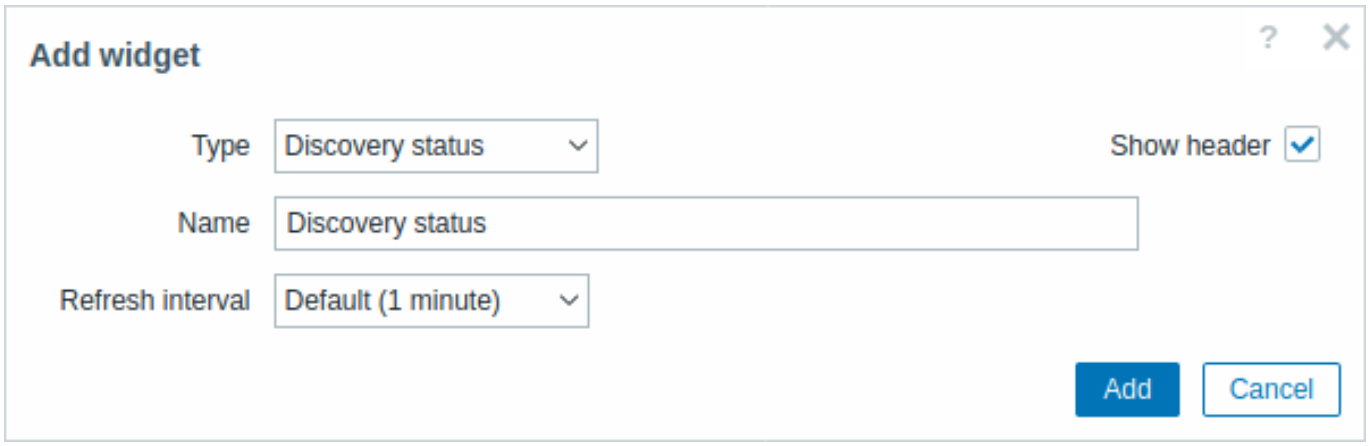
背景色	カラーピッカーから背景色を選択します。 D はデフォルト色（フロントエンドテーマに依存）を表します。デフォルト値に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
日付	
太字	日付を太字で表示するにはチェックボックスをオンにします。
色	カラーピッカーから日付の色を選択します。 D はデフォルト色（フロントエンドテーマに依存）を示します。デフォルト値に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
時刻	
太字	時刻を太字で表示するにはチェックボックスをオンにします。
色	カラーピッカーから時刻の色を選択します。 D はデフォルト色（フロントエンドテーマに依存）を示します。デフォルト値に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
秒	秒を表示するにはチェックボックスをオンにしてください。オフの場合、時と分のみ表示されます。
形式	24 時間制または 12 時間制の時間表示を選択します。
タイムゾーン	
太字	チェックボックスをオンにするとタイムゾーンが太字で表示されます。
色	カラーピッカーからタイムゾーンの色を選択します。 D はデフォルト色（フロントエンドテーマに依存）を表します。デフォルト値に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
タイムゾーン	タイムゾーンを選択してください。

形式	タイムゾーンを短縮形式（例：ニューヨーク）または完全形式（例：(UTC-04:00) America/New York）で表示するか選択してください。

3 ディスカバリのステータス

概要

このウィジェットは、アクティブなネットワークディスカバリルールのステータス概要を表示します。



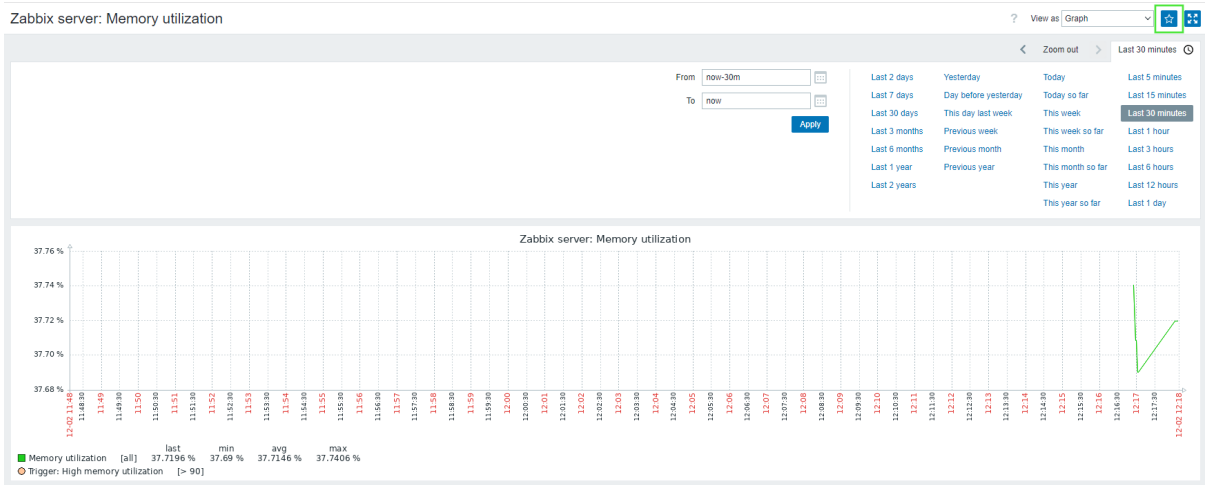
すべての構成パラメーターは全てのウィジェットと共通です。

4 お気に入りのグラフ

概要

このウィジェットには、最も必要なグラフへのショートカットがアルファベット順に含まれています。

監視 -> 最新データ -> グラフでグラフを表示し、  お気に入りに追加ボタンをクリックすると、ショートカットのリストが表示されます。



すべての構成パラメータはすべてのウィジェットと共通です。

5 お気に入りのマップ

概要

このウィジェットには、最も必要なグラフへのショートカットがアルファベット順に含まれています。

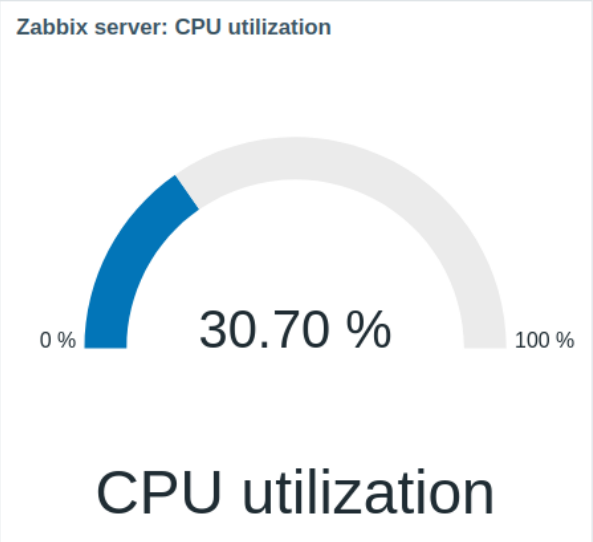
マップを表示し、  お気に入りに追加ボタンをクリックすると、ショートカットのリストが表示されます。

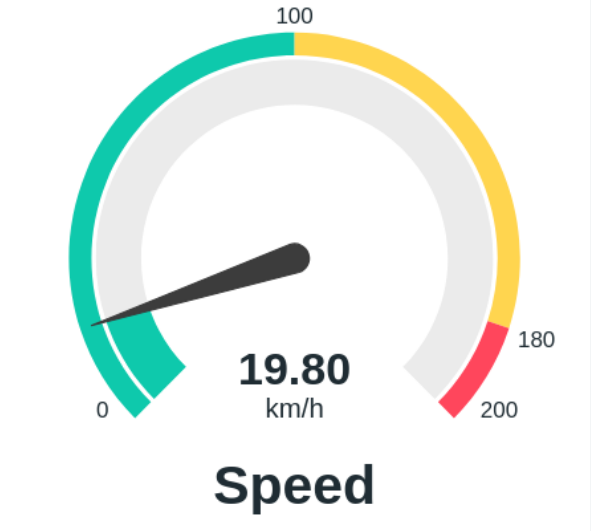
すべての構成パラメータはすべてのウィジェットと共通です。

6 ゲージ

概要

ゲージウィジェットは、単一のアイテムの値をゲージとして表示します。

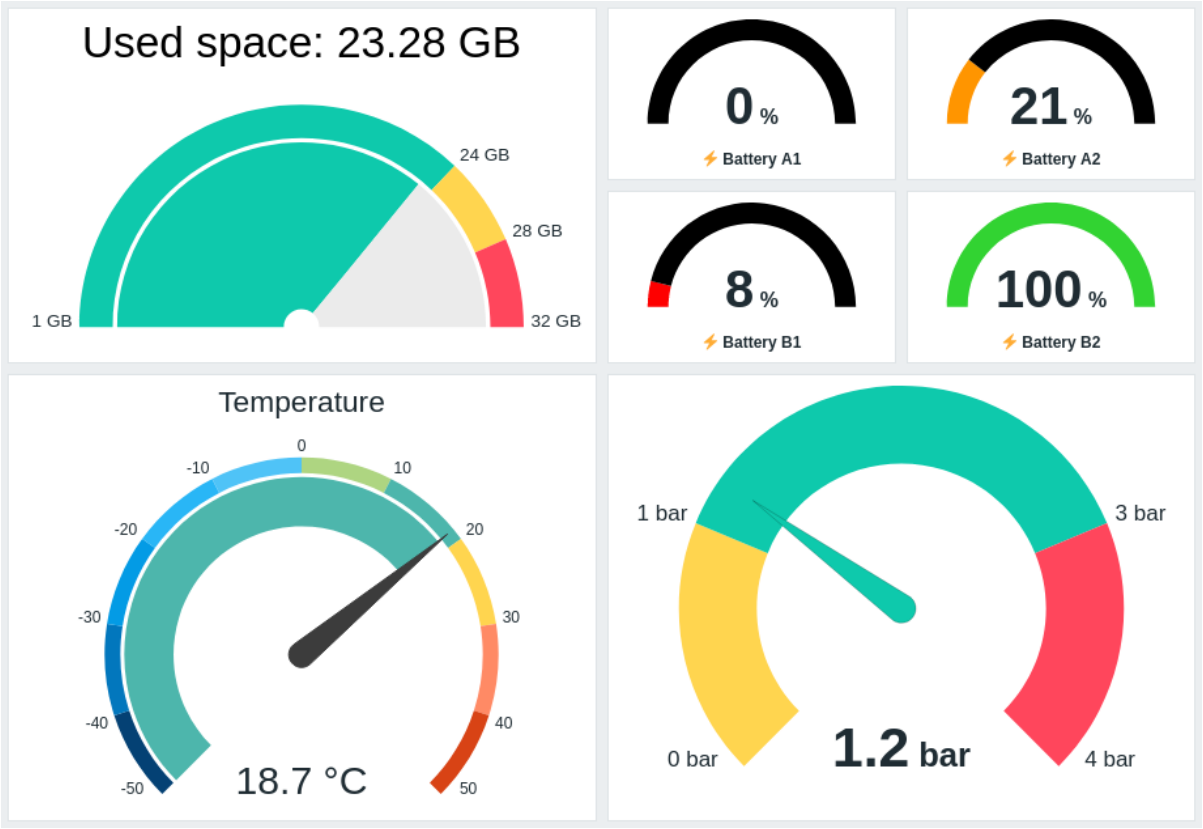




設定すると、このウィジェットは次の要素で構成できます。

- アイテムの説明（例: 「CPU 使用率」、「速度」）
- アイテムの値（例: 「30.70」、「19.80」）
- アイテムの値の単位（例: 「%」、「km/h」）
- ゲージの目盛り（例: 「0%/100%」、「0/100/180/200」）
- ゲージの弧（ゲージ値の弧およびゲージしきい値の弧）
- ゲージの針

このウィジェットは、高度な設定オプションを使用して外観を細かく調整でき、多様なビジュアルスタイルを作成できます。



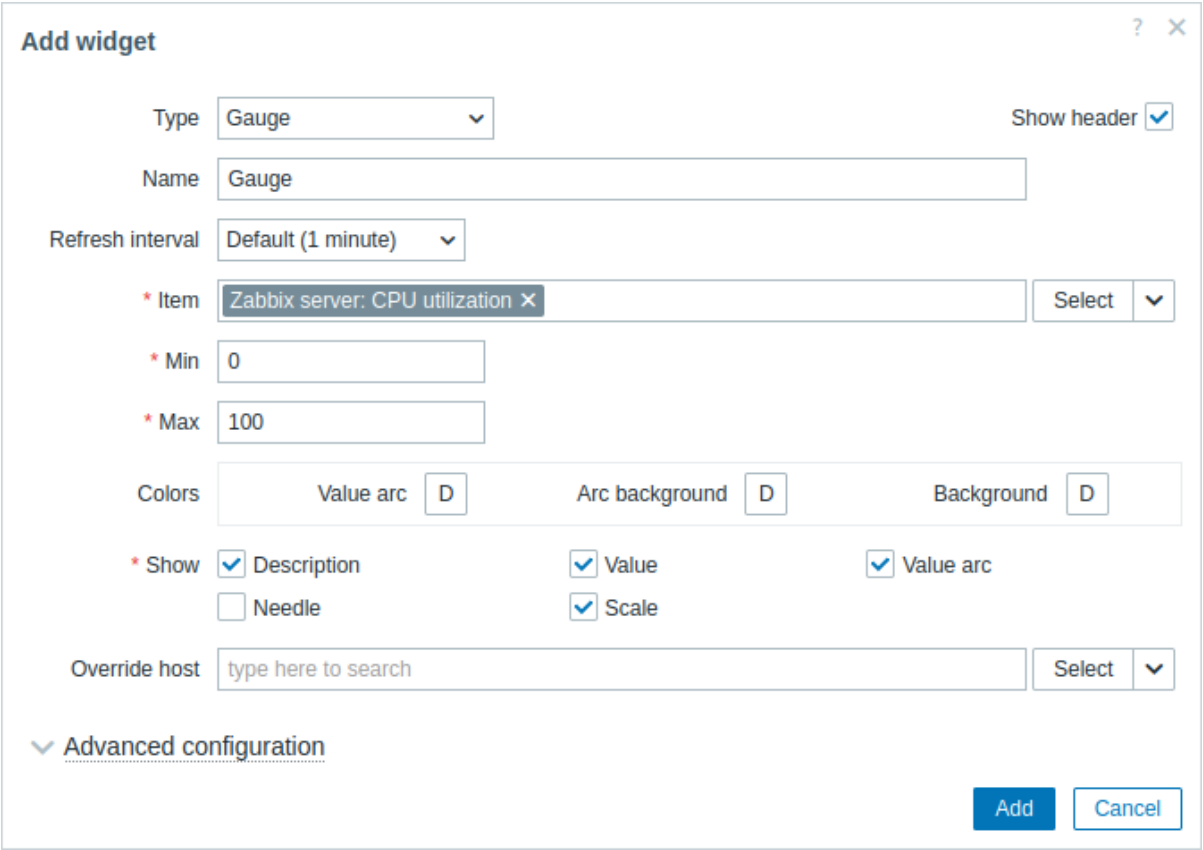
ゲージウィジェットで表示できるのは数値のみです。バイナリ値の表示はサポートされていません。

ゲージウィジェットをクリックすると、そのアイテムのグラフが表示されます。

ゲージウィジェットに表示される情報は、ウィジェットメニューで *Download image* オプションを選択することで、PNG 画像としてダウンロードできます。

設定

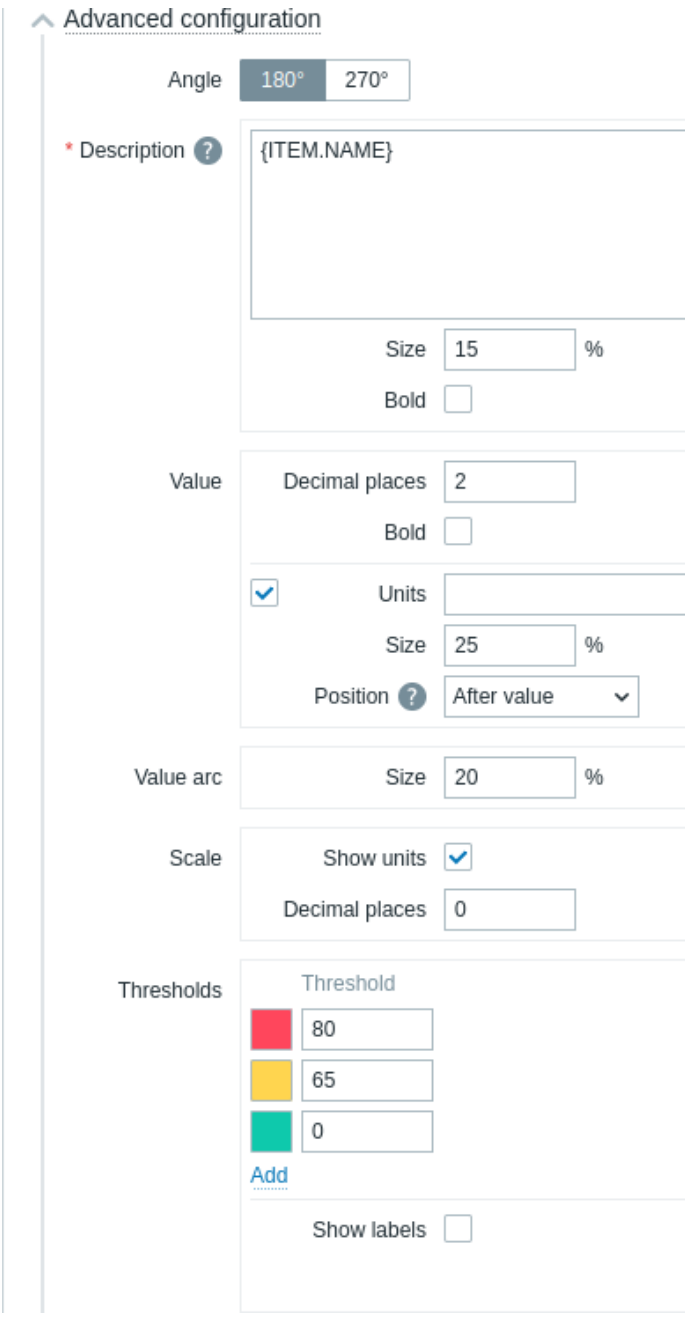
設定するには、タイプとして ゲージを選択します:



すべてのウィジェットに共通のパラメータに加えて、以下の特定のオプションを設定できます：

項目	項目を選択してください。 または、項目のデータソースとして互換性のあるウィジェットを選択することもできます。 このフィールドは自動補完機能を備えており、項目名を入力し始めると一致する項目がドロップダウンで表示されます。 数値データを返す項目のみ選択可能である点に注意してください。
最小値	ゲージの最小値を入力します。 接尾辞（例: 「1d」、「2w」、『4K』、「8G」）がサポートされています。値マッピングもサポートされています。
最大値	ゲージの最大値を入力してください。 接尾辞（例: 「1d」、「2w」、『4K』、「8G」）がサポートされています。値マッピングがサポートされています。
Colors	カラーピッカーから色を選択: **Value arc** - ゲージ値の弧の色を選択; **Arc background** - ゲージ値の弧と閾値の弧の背景色を選択; **Background** - ウィジェットの背景色を選択. 「D」はデフォルト色を表し、フロントエンドテーマに依存します。しきい値が設定されている場合、値の弧のデフォルト色はしきい値の色に依存します。デフォルト色に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
表示	チェックボックスをオンにすると、対応するゲージ要素（説明、値、値弧、針、目盛（ゲージ弧の始点と終点におけるゲージの最小値と最大値））が表示されます。チェックを外すと非表示になります。少なくとも 1 つの要素を選択する必要があります。 ゲージの針と目盛りは、ゲージの値アークまたはゲージのしきい値アーク（詳細設定オプションを参照）が表示されている場合に表示可能です。また、ゲージの針が表示されている場合、値は針の下に配置されます。針が非表示の場合、値はゲージアークの下端に揃えられます。
ホストの上書き	互換性のあるウィジェットまたはダッシュボードをホストのデータソースとして選択します。 このパラメータは、テンプレートダッシュボード上でウィジェットを設定する際には利用できません。
詳細設定	詳細設定ラベルをクリックすると、詳細設定オプションが表示されます。表示フィールドで選択したゲージ要素の調整もここで行えます。

詳細設定



詳細設定オプションは折りたたみ可能な詳細設定セクションで利用可能です:

角度	ゲージの角度を選択してください（180° または 270°）。
説明	
説明	アイテムの説明を入力します。この説明はデフォルトのアイテム名を上書きする場合があります。 複数行の説明をサポートしています。テキストとサポートされているマクロの組み合わせが可能です。 {HOST.*}, {ITEM.*}, {INVENTORY.*} およびユーザー定義マクロがサポートされています。
サイズ	アイテム説明のフォントサイズの高さ（ウィジェット全体の高さに対するパーセント単位）を入力してください。
垂直位置	アイテム説明の垂直位置（ゲージアークに対する上部または下部）を選択してください。
太字	チェックボックスをオンにすると、アイテム説明を太字で表示します。
色	カラーピッカーからアイテム説明文の色を選択してください。 「D」はフロントエンドテーマに依存するデフォルト色を表します。デフォルト色に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
値	
小数点以下の桁数	値と共に表示する小数点以下の桁数を入力してください。 このオプションは数値（浮動小数点）データを返すアイテムにのみ影響します。
サイズ	値のフォントサイズの高さ（ゲージアークの高さに対するパーセント単位）を入力してください。
太字	値を太字で表示するにはチェックボックスをオンにしてください。
色	カラーピッカーから値の色を選択してください。 「D」はデフォルト色を表し、フロントエンドテーマに依存します。デフォルト色に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。

単位	
単位	項目値と共に単位を表示するにはチェックボックスをオンにしてください。 単位名を入力すると、項目設定で設定された単位が上書きされます。
サイズ	項目単位のフォントサイズの高さ（ゲージの弧の高さに対するパーセント単位）を入力してください。
太字	チェックボックスをオンにすると、アイテム単位を太字で表示します。
位置	アイテム単位の位置を選択します（値の上、下、前、後）。 以下の時間関連単位ではこのオプションは無視されます：unixtime、uptime、s。
色	カラーピッカーから項目単位の色を選択します。 「D」はフロントエンドテーマに依存するデフォルト色を表します。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックしてください。
値アーク	
アークサイズ	ゲージ値アークの高さサイズを入力します（ゲージアーク半径に対するパーセント）。
針	
色	カラーピッカーからゲージ針の色を選択してください。 「D」はフロントエンドテーマに依存するデフォルト色を表します。しきい値が設定されている場合、針のデフォルト色はしきい値の色に依存します。デフォルト色に戻すには、カラーピッカー内のデフォルトを使用ボタンをクリックしてください。
スケール	
単位を表示	ゲージの最小値と最大値に単位を表示するにはチェックボックスをオンにします。
サイズ	ゲージの最小値と最大値のフォントサイズの高さを入力します（ゲージの半径に対するパーセント）。
スケール	
単位を表示	ゲージの最小値と最大値に単位を表示するにはチェックボックスをオンにします。
サイズ	ゲージの最小値と最大値のフォントサイズの高さを入力します（ゲージの弧の高さに対するパーセント）。
小数点以下の桁数	ゲージの最小値と最大値に表示する小数点以下の桁数を入力します。 このオプションは、数値（浮動小数点） データを返す項目にのみ影響します。
しきい値	
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択し、数値を指定します。 保存時にしきい値リストは昇順で並べ替えられます。 数値項目でのみ、しきい値として設定した色が正しく表示される点に注意してください。 接尾辞（例: 「1d」、「2w」、『4K』、「8G」）がサポートされています。値のマッピングがサポートされています。
ラベルを表示	ゲージ目盛上にしきい値をラベルとして表示するにはチェックボックスをオンにします。
弧を表示	ゲージしきい値の弧を表示するにはチェックボックスをオンにします。
弧のサイズ	ゲージしきい値の弧の高さサイズを入力します（ゲージの弧半径に対するパーセント）。

7 ジオマップ

概要

地理マップウィジェットは、オープンソースの JavaScript インタラクティブマップライブラリ Leaflet を使用して、ホストを地理マップ上のマーカーとして表示します。

> **Note:**
> Zabbix は複数の事前定義されたマップタイルサービスプロバイダーとカスタムタイルサービスプロバイダーまたはホストタイル自体を追加するオプションを提供します（管理 → 一般設定 → 地理マップメニューセクションで設定可能）

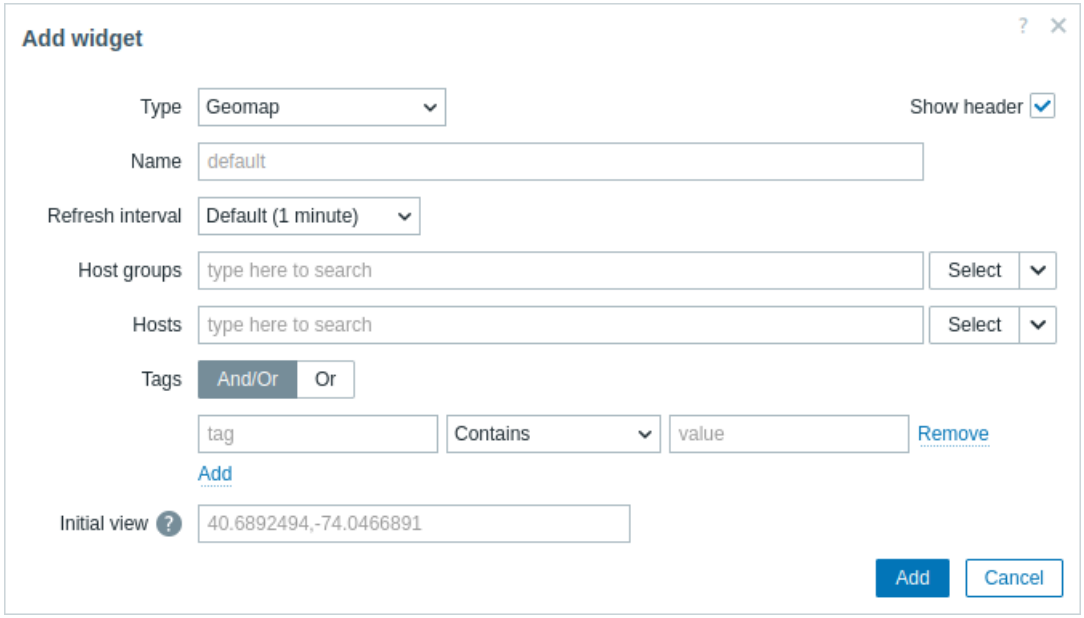
デフォルトでは、このウィジェットには、インベントリ で有効な座標が定義されているすべての有効なホストが表示されます。ウィジェットのパラメータでは、ホストのフィルタリングを設定できます。

有効なホスト座標は次のとおりです。

- 緯度: -90 から 90 まで（整数または浮動小数点数）
- 経度: -180 から 180 まで（整数または浮動小数点数）

設定

ウィジェットを追加するにはタイプとして地理マップを選択します。



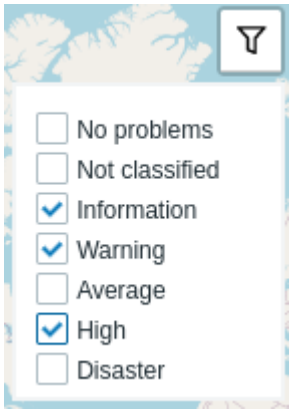
すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

ホストグループ	マップに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 ホストグループフィールドとホストフィールドの両方で何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	マップに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストグループフィールドとホストフィールドの両方で何も選択されていない場合、有効な座標を持つすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
タグ	ウィジェットに表示されるホスト数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は、*Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

初期表示	ウィジェットの初回読み込み時に表示する中心座標と、必要に応じてズームレベルを、<latitude>,<longitude>,<zoom> 形式でカンマ区切りで指定します。 初期ズームが指定されている場合、Geomap ウィジェットは指定したズームレベルで読み込まれます。指定されていない場合、初期ズームは、そのタイルプロバイダーに対するmax zoomの半分として計算されます。 デフォルト表示が設定されている場合、初期表示は無視されます（以下を参照）。 例: 40.6892494,-74.0466891 40.6892494,-122.0466891

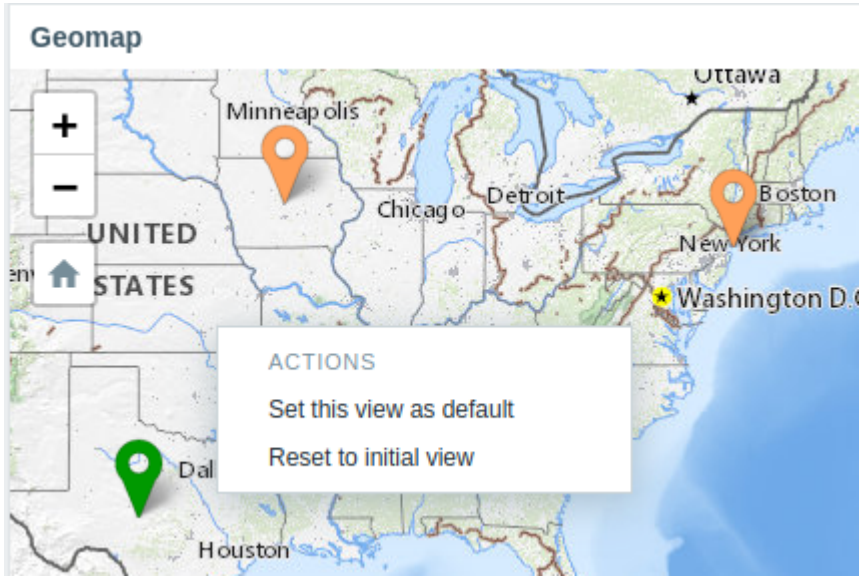
マップに表示されるホストマーカーは、そのホストの最も深刻な問題の色で表示され、ホストに問題がない場合は緑色で表示されます。
ホストマーカーをクリックすると、ホストの表示名と、深刻度ごとにグループ化された未解決の問題数を確認できます。
表示名をクリックすると、ホストメニューが開きます。

マップに表示されるホストは、問題の深刻度でフィルタリングできます。
ウィジェット右上のフィルターアイコンを押し、必要な深刻度を選択してください。



ウィジェットの左上隅にあるプラスおよびマイナスのボタン、またはマウスのスクロールホイールやタッチパッドを使用して、マップを拡大・縮小できます。
現在の表示をデフォルトとして設定するには、マップ上の任意の場所を右クリックし、*Set this view as default* を選択します。
この設定は、現在のユーザーに対して *Initial view* ウィジェットパラメータを上書きします。
この操作を元に戻すには、再度マップ上の任意の場所を右クリックし、*Reset to initial view* を選択します。

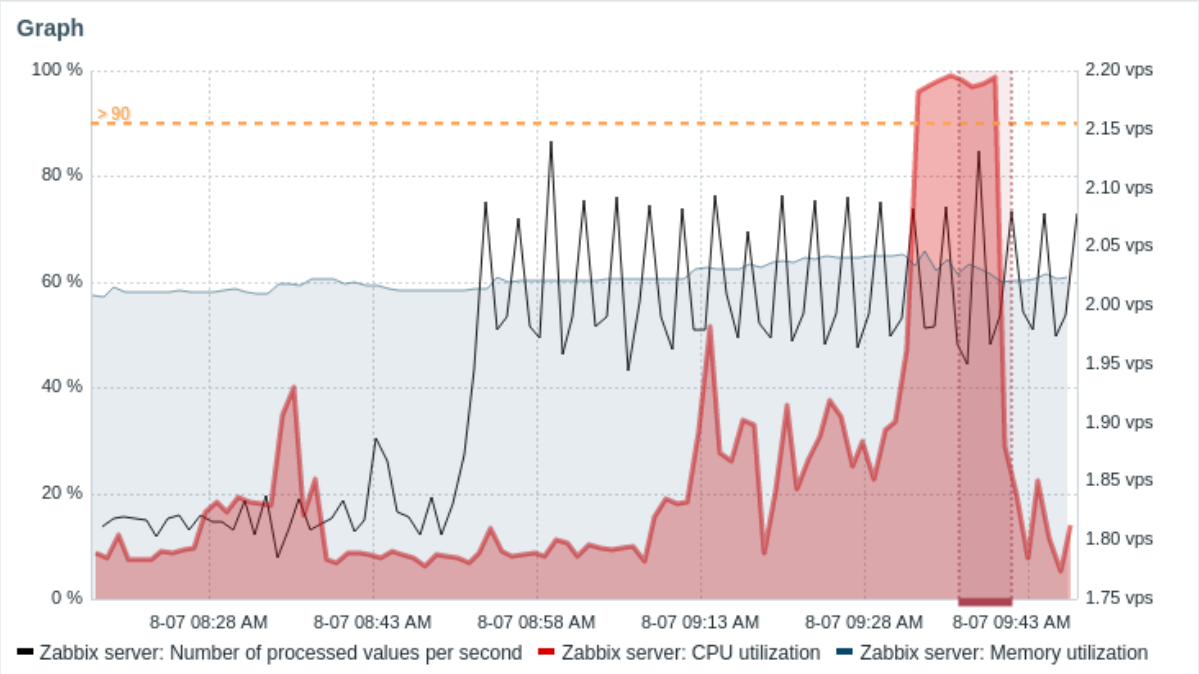
Initial view または *Default view* が設定されている場合は、左側のホームアイコンを押すことで、いつでもこの表示に戻ることができます。



8 グラフ

概要

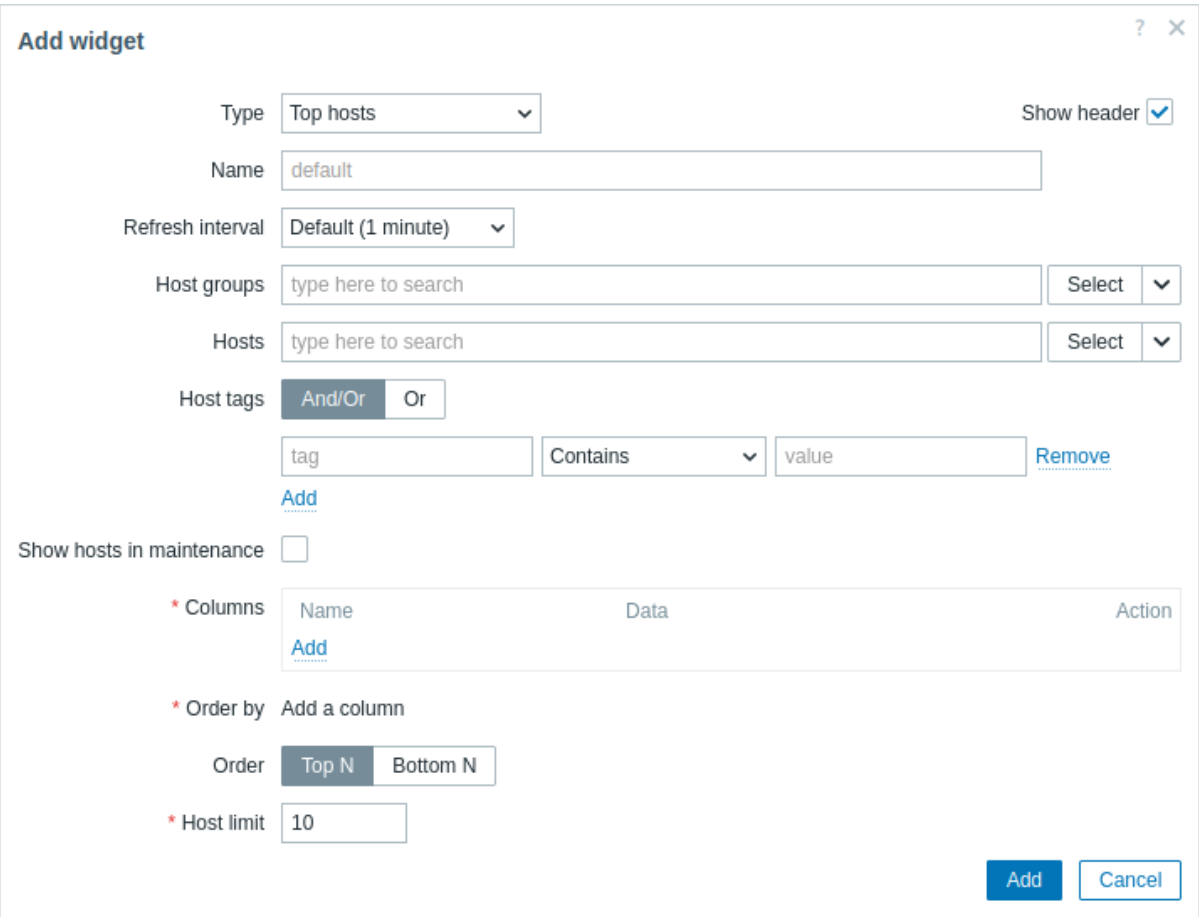
グラフウィジェットは、ベクター画像描画技術を使用して Zabbix が収集したデータを可視化するための、モダンで汎用性の高い方法を提供します。このグラフウィジェットは、Zabbix 4.0 以降でサポートされています。Zabbix 4.0 より前にサポートされていたグラフウィジェットも、グラフ (クラシック) として引き続き使用できることに注意してください。詳細については、ダッシュボードページのウィジェットの追加 セクションも参照してください。



グラフウィジェットに表示される情報は、ウィジェットメニュー で 画像をダウンロードオプションを選択することで、PNG 画像としてダウンロードできます。

設定

設定するにはタイプとしてグラフを選択します。



データセット

データセットタブでは、データセットを追加してグラフ用のデータを選択できます。追加できるデータセットには 2 種類あります。

- アイテムパターン - 一致するアイテムのデータが表示されます。単一のベースカラーを選択するか、パレットの行を選択して、一致した各アイテムに異なる色を割り当てることができます。
- アイテムリスト - 選択したアイテムのデータが表示されます。各アイテムの色をピッカーから個別に選択できます。

デフォルトでは、アイテムパターンデータセットが追加されます。

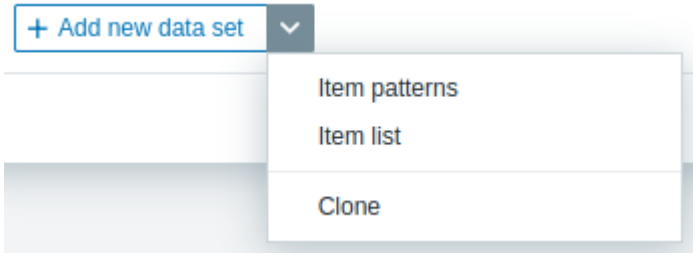
データセット	アイテムパターンデータセットの場合: ホストパターンとアイテムパターンを選択または入力します。これらのパターンに一致するアイテムのデータがグラフに表示されます。最大 50 個のアイテムを表示できます。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。 ワイルドカード記号は常に解釈されるため、たとえば *item** という名前のアイテムがあり、他にも一致するアイテム（たとえば *item2*、*item3*）が存在する場合、そのアイテムだけを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストパターンとアイテムパターンの指定は必須です。 あわせて参照: データセット設定の詳細。 アイテムリストデータセットの場合: アイテムの追加ボタンをクリックして、グラフ用のアイテムを選択します。 ウィジェットの追加ボタンをクリックして、アイテムのdata sourceとして互換性のあるウィジェットを選択することもできます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットの指定は必須です。 あわせて参照: データセット設定の詳細。 数値型のアイテムのみ使用できます。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するためのパラメータは使用できず、アイテムリストを指定するためのパラメータではテンプレート上で設定されたアイテムのみ選択できます。
描画	メトリクスの描画タイプを選択します。 選択可能な描画タイプ: 線（デフォルト）、ポイント、階段、棒。 線/階段グラフにデータポイントが 1 つしかない場合は、描画タイプに関係なくポイントとして描画されます。ポイントサイズは線幅から計算されますが、線幅がそれ未満であっても 3 ピクセル未満にはなりません。
積み上げ	チェックボックスをオンにすると、データを積み上げ表示します（塗りつぶし領域を表示）。 このオプションは、ポイント描画タイプが選択されている場合は無効です。
幅	線幅を設定します。 このオプションは、線または 階段描画タイプが選択されている場合に使用できます。
ポイントサイズ	ポイントサイズを設定します。 このオプションは、ポイント描画タイプが選択されている場合に使用できます。
透明度	透明度レベルを設定します。
塗りつぶし	塗りつぶしレベルを設定します。 このオプションは、線または 階段描画タイプが選択されている場合に使用できます。
欠損データ	欠損データの表示方法を選択します。 なし - ギャップは空白のままになります。 接続 - 両端の 2 つの値を接続します。 **0** として扱う - 欠損データを 0 値として表示します。 直前の値 - 欠損データを最後に判明している値と同じ値で表示します。ポイントおよび 棒描画タイプには適用されません。
ホストを上書き	ホストの data source として、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Y 軸	Y 軸を表示するグラフの側を選択します。
時間シフト	必要に応じて時間シフトを指定します。 このフィールドではtime suffixesを使用できます。負の値も使用できます。

集計関数	使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の件数を表示します。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します。 **none** - すべての値を表示します（集計なし）。 集計を使用すると、すべての値の代わりに、選択した間隔（5 分、1 時間、1 日）の集計値を表示できます。あわせて参照: グラフでの集計。
集計間隔	値を集計する間隔を指定します。 このフィールドではtime suffixesを使用できます。サフィックスのない数値は秒として扱われます。 ウィジェットがトレンドに基づく履歴データを表示するよう設定されている場合（履歴データの選択が トレンドまたは 自動に設定されている場合）、1 時間の倍数となる集計間隔（例: 3600、60m、1h、3h など）を使用することを推奨します。トレンドには 1 時間単位で集計された値が保存されるため、1 時間の倍数でない集計間隔（例: 100s、7min、15min、90min など）を使用すると、解釈しにくい結果になる可能性があります。
集計	集計方法を指定します。 各アイテム - データセット内の各アイテムを個別に集計して表示します。 データセット - データセット内のすべてのアイテムを集計し、1 つの値として表示します。
近似	グラフの縦 1 ピクセルあたりに複数の値が存在する場合に表示する値を指定します。 **all** - 最小値、最大値、平均値を表示します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 この設定は、更新間隔が短い長期間のグラフ（たとえば、10 分ごとに収集された 1 年分の値）を表示する場合に便利です。
データセットラベル	グラフの データセット設定およびグラフの 凡例（集計されたデータセットの場合）に表示されるデータセットラベルを指定します。 データセットラベルが指定されているものも含め、すべてのデータセットには番号が付けられます。ラベルが指定されていない場合、データセットには番号に応じて自動的にラベルが付けられます（例: 「データセット #2」、「データセット #3」など）。データセットの番号は、データセットの並べ替え/ドラッグ後に再計算されます。 長すぎるデータセットラベルは、表示領域に収まるように短縮されます（例:「プロセス数...」）。

データセット設定の詳細

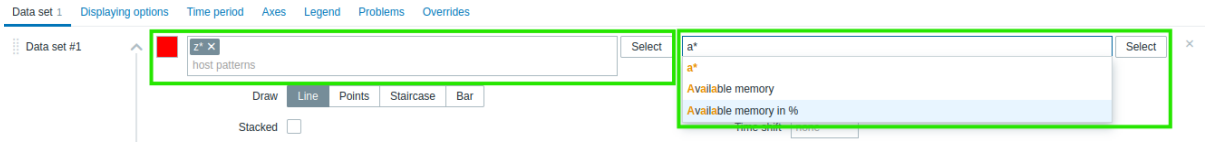
既存のデータセットは一覧で表示されます。次の操作を行えます。

-  移動アイコンをクリックし、データセットをドラッグして一覧内の別の位置に移動します。
- 展開アイコンをクリックして、データセットの詳細を展開します。展開すると、このアイコンは 折りたたみアイコンに変わります。
- 色アイコンをクリックして、カラーピッカーを開きます。16 進コードを入力するか、単色のスウォッチを選択するか、*Palette* タブに切り替えて定義済みの色の行を選択できます。選択した色は、*Item list* データセットには直接適用され、*Item patterns* では生成される色合いのベースとして使用されます。Tab キーでダイアログ内のコントロール間を移動し、矢印キーでスウォッチまたはパレットの行を移動し、Enter キーで選択し、Esc キーでキャンセルします。
- *Add new data set* ボタンをクリックすると、ホストおよびアイテムパターンを選択できる空のデータセットが追加されます。*Add new data set* ボタンの横にある下向きアイコンをクリックすると、ドロップダウンメニューが表示され、新しい *Item patterns* または *Item list* データセットを追加したり、現在開いているデータセットを *Clone* したりできます。すべてのデータセットが折りたたまれている場合、*Clone* オプションは使用できません。

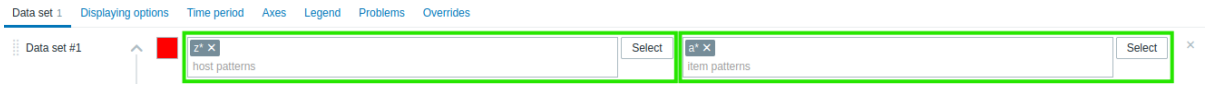


アイテムパターンデータセットには、ホストパターンフィールドと アイテムパターンフィールドがあり、どちらも完全名またはワイルドカード記号（*）を含むパターンを認識します。
この機能により、選択したパターンを含むすべてのホスト名とアイテム名を選択できます。
アイテムパターンフィールドにアイテム名またはアイテムパターンを入力している間、ドロップダウンリストには、選択したホスト名に属するアイテムのみが表示されます。

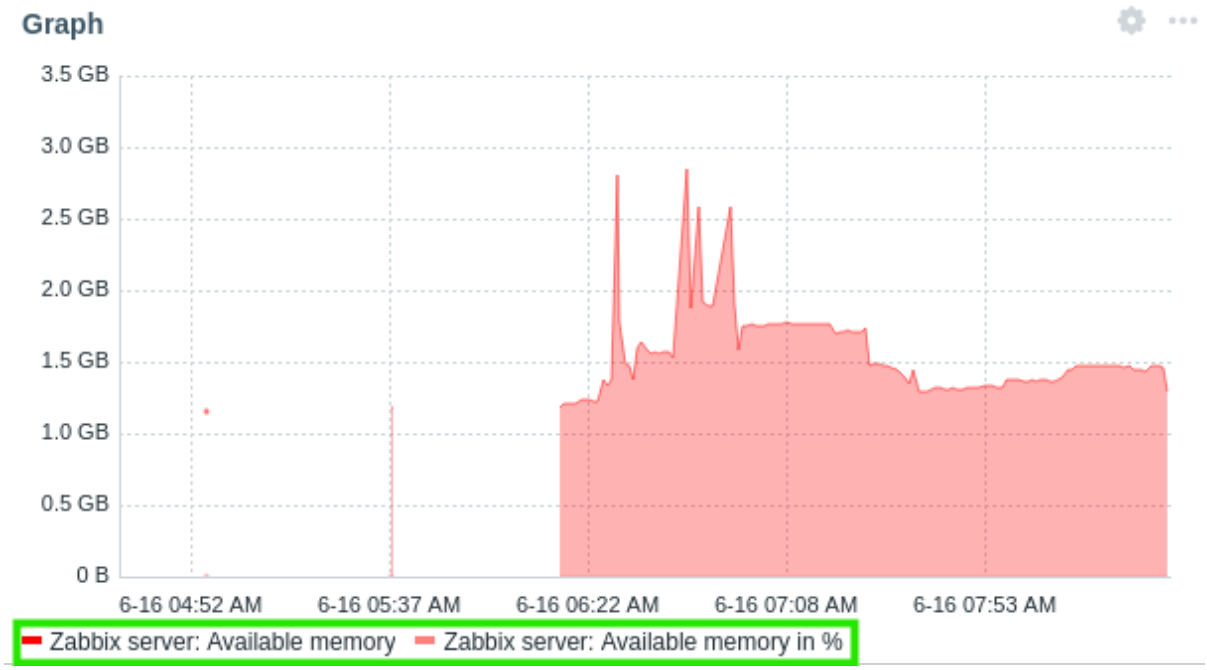
例えば、ホストパターンフィールドにパターン **z*** を入力すると、ドロップダウンリストにはこのパターンを含むすべてのホスト名、すなわち **z***、**Zabbix server**、**Zabbix proxy** が表示されます。
Enter を押すと、このパターンが受け入れられ、**z*** として表示されます。
同様に、アイテムパターンフィールドにパターン **a*** を入力すると、ドロップダウンリストにはこのパターンを含むすべてのアイテム名、すなわち **a***、**Available memory**、**Available memory in %** が表示されます。



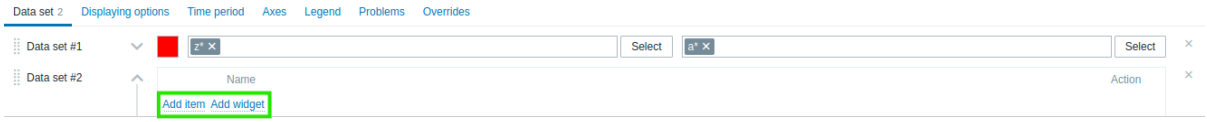
Enter を押すと、パターンが受け入れられ、**a*** として表示されます。



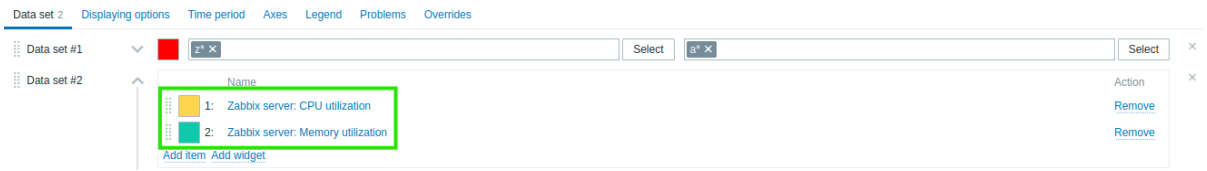
その後、グラフには選択したホスト名に属するすべてのアイテムが表示されます。



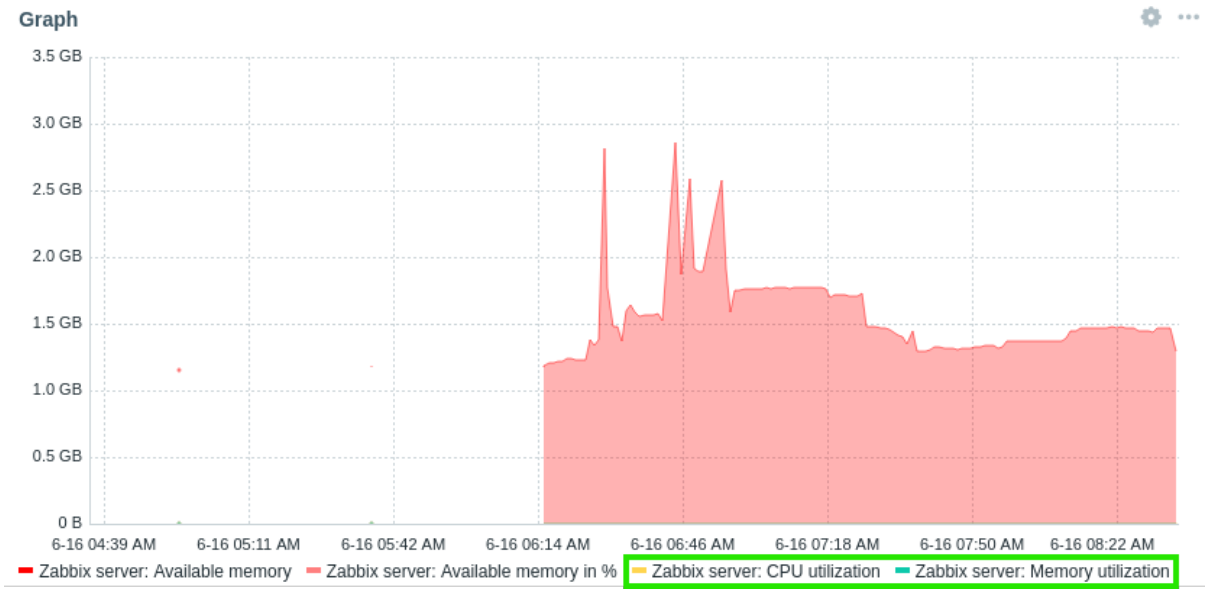
アイテムリストデータセットには、グラフに表示するアイテムを追加できる アイテムの追加ボタンがあります。
また、ウィジェットの追加ボタンをクリックすると、アイテムのデータソースとして互換性のあるウィジェットを追加することもできます。



たとえば、アイテムの追加ボタンをクリックすると、ホストパラメータを含むポップアップウィンドウが開きます。
ホストを選択すると、選択可能なそのホストのすべてのアイテムが一覧に表示されます。

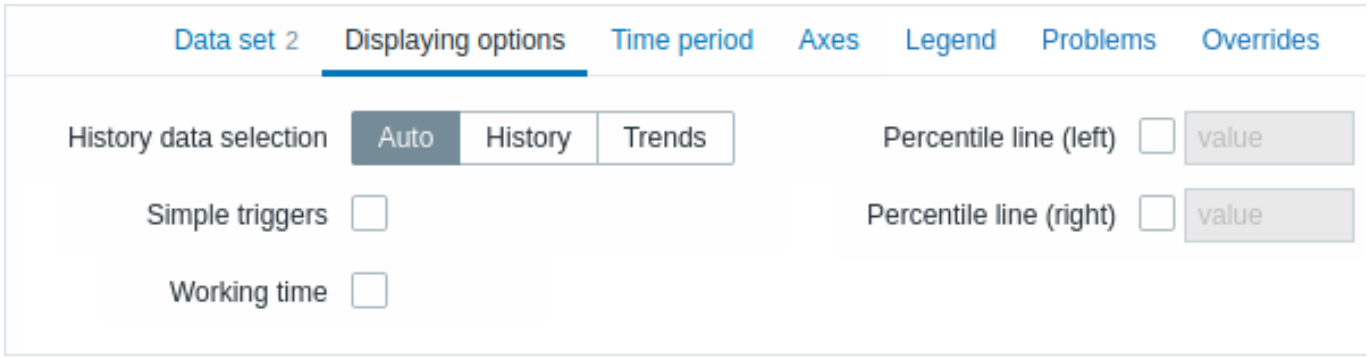


1 つ以上のアイテムを選択すると、それらはデータセットのアイテム一覧とグラフに表示されます。



表示オプション

表示オプションタブでは、履歴データの選択を定義できます。

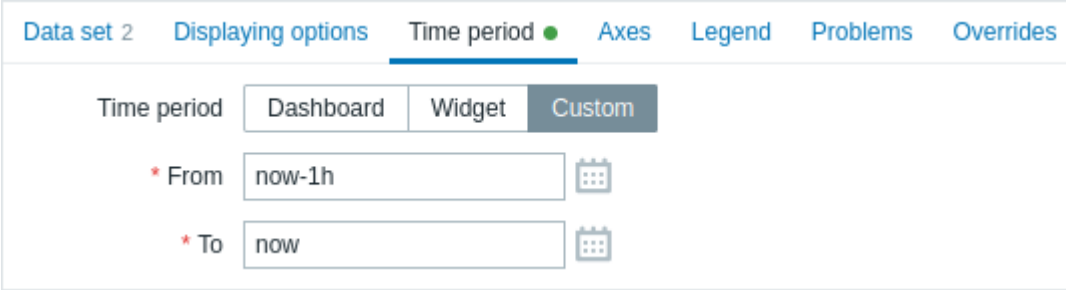


履歴データの選択	グラフデータのソースを設定します。 自動 - クラシックグラフのアルゴリズムに従ってデータを取得します（デフォルト）。 履歴 - 履歴からのデータ。 トレンド - トレンドからのデータ。
シンプルトリガー	チェックボックスをオンにすると、シンプルトリガーのトリガーしきい値を表示します。しきい値は、トリガーの深刻度の色を使用した破線として描画されます。 シンプルトリガーとは、式内の 1 つのアイテムに対して 1 つの関数（`last`、`max`、`min`、`avg` のみ）を持つトリガーです。 描画できるトリガーは最大 3 つです。表示するには、トリガーが描画範囲内にある必要があることに注意してください。
稼働時間	チェックボックスをオンにすると、グラフ上に稼働時間を表示します。稼働時間（稼働日）はグラフでは白い背景で表示され、非稼働時間は灰色で表示されます（デフォルトの Web インターフェーステーマ *Original blue* の場合）。
パーセンタイル線（左）	チェックボックスをオンにしてパーセンタイル値を入力すると、指定したパーセンタイルをグラフの左 Y 軸上の線として表示します。 たとえば 95% パーセンタイルを設定した場合、パーセンタイル線は値の 95 パーセントがその下に入るレベルに表示されます。

パーセンタイル線（右）	チェックボックスをオンにしてパーセンタイル値を入力すると、指定したパーセンタイルをグラフの右 Y 軸上の線として表示します。 たとえば 95% パーセンタイルを設定した場合、パーセンタイル線は値の 95 パーセントがその下に入るレベルに表示されます。

期間

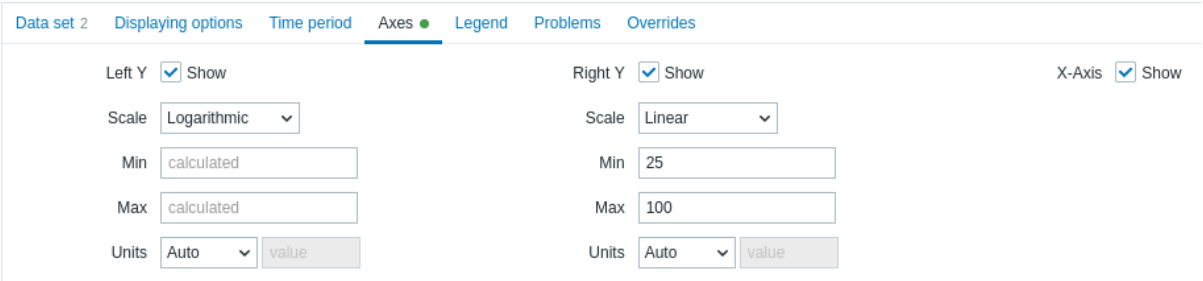
期間タブでは、グラフに表示するデータの期間を設定できます。



期間	期間のデータソースを選択します。詳細はdata sourceを参照してください。 ダッシュボード - 期間セレクターをデータソースとして設定します。 ウィジェット - ウィジェットパラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 カスタム - 開始および終了パラメータで指定した期間をデータソースとして設定します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。
ウィジェット	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に使用できます。
開始	期間の開始を入力または選択します。 Relative time syntax（now、now/d、now/w-1w など）をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。
終了	期間の終了を入力または選択します。 Relative time syntax（now、now/d、now/w-1w など）をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。

軸

軸タブでは、軸の表示方法をカスタマイズできます。

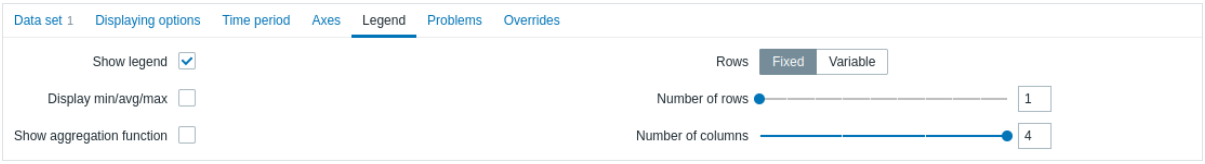


左 *Y*	このチェックボックスをオンにすると、左 Y 軸が表示されます。 データセットタブまたはオーバーライドタブのいずれかで未選択の場合、このチェックボックスは無効になることがあります。
右 *Y*	このチェックボックスをオンにすると、右 Y 軸が表示されます。 データセットタブまたはオーバーライドタブのいずれかで未選択の場合、このチェックボックスは無効になることがあります。
X 軸	このチェックボックスをオフにすると、X 軸を非表示にします（デフォルトでオン）。
スケール	ドロップダウンからグラフ軸の値のスケールを選択します。 線形 - 軸の値は一定量ずつ増加します（例: 10、20、30）。一定の変化をするデータや、小～中程度の範囲を扱うデータに適しています。 対数 - 軸の値は指数関数的に増加します（例: 10、100、1000）。急激に変化するデータや、大きな範囲を扱うデータに適しています。

最小	対応する軸の最小値を設定します。 Y 軸の表示範囲の最小値を指定します。
最大	対応する軸の最大値を設定します。 Y 軸の表示範囲の最大値を指定します。
単位	ドロップダウンからグラフ軸の値の単位を選択します。 自動 - 軸の値は、データセット内の最初のアイテムの単位を使用して表示されます。 固定 - 軸の値は、*value* 入力フィールドで指定した単位を使用して表示されます。フィールドを空欄のままにすると、数値のみが表示されます。

凡例

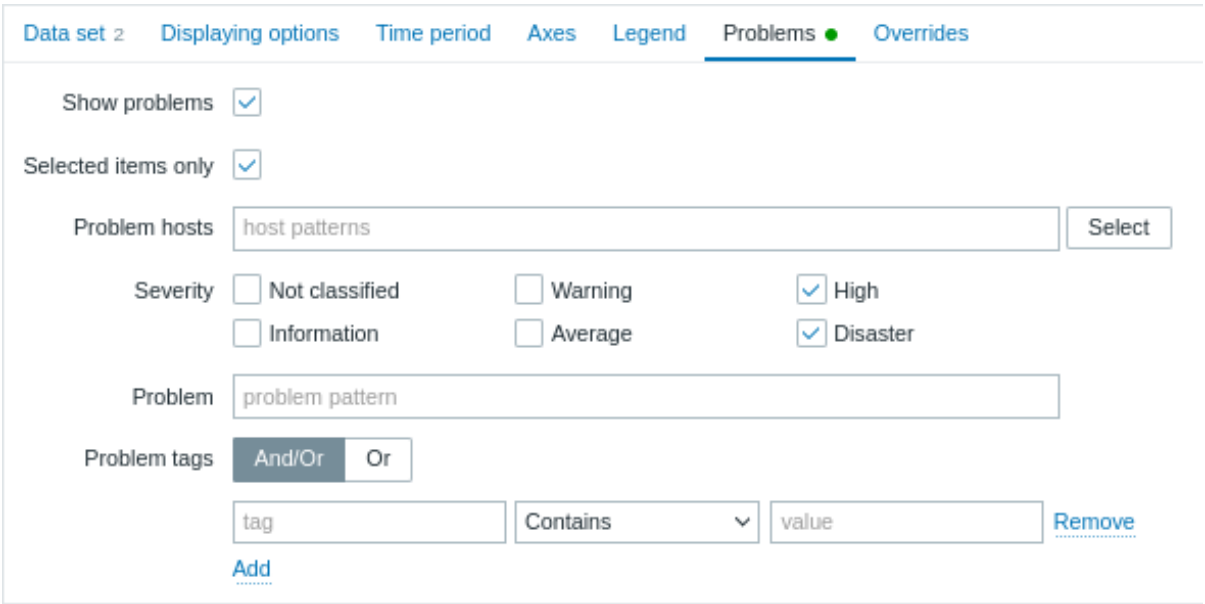
凡例タブでは、グラフの凡例をカスタマイズできます。



凡例を表示	このチェックボックスをオフにすると、グラフ上の凡例が非表示になります（デフォルトでオン）。
最小/平均/最大を表示	このチェックボックスをオンにすると、凡例にアイテムの最小値、平均値、最大値が表示されます。
集計関数を表示	このチェックボックスをオンにすると、凡例に集計関数が表示されます。
行	凡例の行の表示モードを選択します。 固定 - 表示される行数は、行数パラメータの値によって決まります。 可変 - 表示される行数は、設定されたアイテム数によって決まり、最大行数パラメータの値を超えません。
行数/ 最大行数	行が「固定」に設定されている場合、表示する凡例の行数を設定します（1～10）。 行が「可変」に設定されている場合、表示する凡例の最大行数を設定します（1～10）。
列数	表示する凡例の列数を設定します（1～4）。 このパラメータは、最小/平均/最大を表示がオフの場合に使用できます。

障害

障害タブでは、障害の表示をカスタマイズできます。

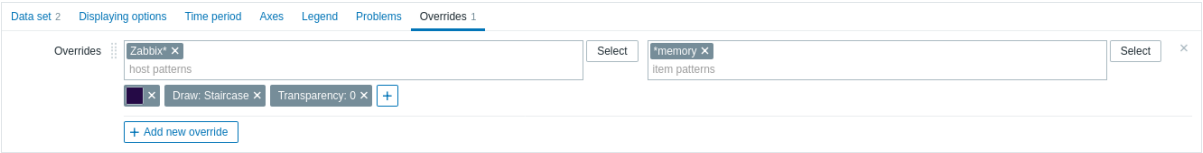


障害を表示	このチェックボックスをオンにすると、グラフ上で障害の表示が有効になります（デフォルトではオフ、つまり無効）。
選択したアイテムのみ	このチェックボックスをオンにすると、グラフには選択したアイテムの障害のみが表示されます。

障害ホスト	グラフに表示する障害ホストを選択します。 ワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。 入力中は、一致するすべてのホストがドロップダウンに表示されることに注意してください。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
深刻度	グラフに表示する障害を絞り込むため、障害の深刻度を選択します。 深刻度が 1 つも選択されていない場合は、すべての障害が表示されます。
障害	グラフに表示する障害名を指定します。
障害タグ	ウィジェットに表示する障害数を制限するため、障害タグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

オーバーライド

オーバーライドタブでは、データセットに対してカスタムオーバーライドを追加できます。



オーバーライドは、* ワイルドカードを使用して 1 つのデータセットに複数のアイテムを選択しており、それらのアイテムのデフォルトの表示方法（例えば、デフォルトの基本色やその他のプロパティ）を変更したい場合に便利です。

既存のオーバーライド（存在する場合）は一覧に表示されます。新しいオーバーライドを追加するには、次の手順を実行します。

- 次の 

ボタンをクリックします
- オーバーライドの対象となるホストとアイテムを選択します。あるいは、ホストパターンとアイテムパターンを入力することもできます。ワイルドカードパターンを使用できます（例えば、* は 0 文字以上に一致する結果を返します）。ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力中は、一致するすべてのホストがドロップダウンに表示されることに注意してください。ワイルドカード記号は常に解釈されるため、例えば”item*” という名前のアイテムがあっても、他に一致するアイテム（例: item2、item3）が存在する場合、それだけを個別に追加することはできません。ホストパターンおよびアイテムパターンのパラメータは必須です。テンプレートダッシュボード でウィジェットを設定する場合、ホストパターンを指定するためのパラメータは使用できません。アイテム一覧を指定するためのパラメータでは、テンプレートダッシュボード でウィジェットを設定する場合、テンプレート上で設定されたアイテム のみ選択できます。
- 次の 

をクリックして、オーバーライドパラメータを選択します。少なくとも 1 つのオーバーライドパラメータを選択する必要があります。各パラメータの説明については、上記の データセットタブを参照してください。

9 グラフ（クラシック）

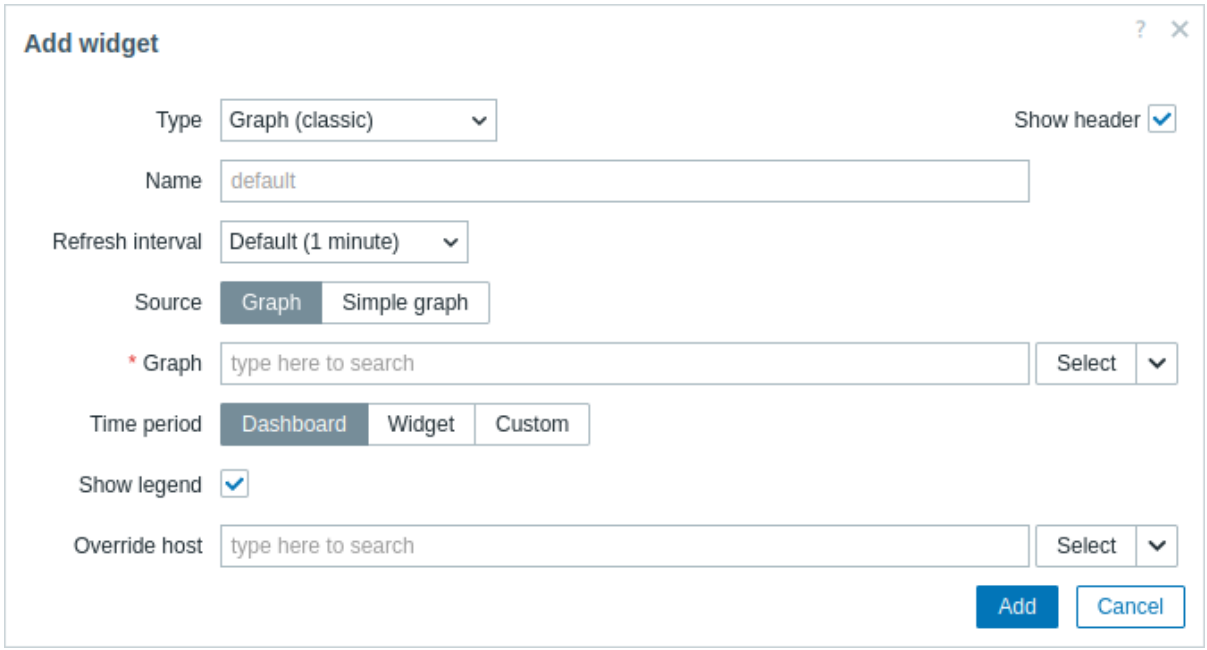
概要

クラシックグラフウィジェットでは、単一のカスタムグラフまたはシンプルグラフを表示できます。

グラフ（クラシック）ウィジェットに表示される情報は、ウィジェットメニューで画像をダウンロードオプションを選択することで、PNG 画像としてダウンロードできます。

設定

設定するには、タイプとして *Graph (classic)* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

Source	グラフタイプを選択します: **Graph** - カスタムグラフ; **Simple graph** - シンプルグラフ。
Graph	表示するカスタムグラフを選択します。 または、グラフのdata sourceとして互換性のあるウィジェットを選択します。 このパラメータは、*Source* が"Graph" に設定されている場合に使用できます。
Item	シンプルグラフに表示するアイテムを選択します。 または、アイテムのdata sourceとして互換性のあるウィジェットを選択します。 このパラメータは、*Source* が"Simple graph" に設定されている場合に使用できます。
Time period	グラフにデータを表示する期間を設定します。期間のdata sourceを選択します: **Dashboard** - *Time period* セレクターをデータソースとして設定します; **Widget** - *Widget* パラメータで指定した互換性のあるウィジェットをデータソースとして設定します; **Custom** - *From* および *To* パラメータで指定した期間をデータソースとして設定します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。
Widget	期間のデータソースとして、互換性のあるウィジェットを入力または選択します。 このパラメータは、*Time period* が"Widget" に設定されている場合に使用できます。
From	期間の開始を入力または選択します。 相対時刻構文 (`now`, `now/d`, `now/w-1w` など) をサポートしています。 このパラメータは、*Time period* が"Custom" に設定されている場合に使用できます。
To	期間の終了を入力または選択します。 相対時刻構文 (`now`, `now/d`, `now/w-1w` など) をサポートしています。 このパラメータは、*Time period* が"Custom" に設定されている場合に使用できます。
Show legend	このチェックボックスをオフにすると、グラフ上の凡例を非表示にします（デフォルトではオン）。
Override host	ホストのdata sourceとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

グラフの凡例は、次の 3 つのセクションで構成されます。

- アイテムとその集計値
- パーセンタイル（設定されている場合）

- トリガー（表示されているアイテムに関連付けられている場合）

ウィジェット内のグラフの高さが不十分な場合、凡例は表示されないか、一部のみ表示されることがあります。
最初にトリガーとパーセンタイルが非表示になり、その後アイテムの凡例が非表示になります。
凡例全体を表示するには、ウィジェットの縦サイズを大きくしてください。

Note:
表示できるトリガーラインは最大 3 本です。
トリガーがそれより多い場合は、重要度の低いトリガーが優先して表示されます。

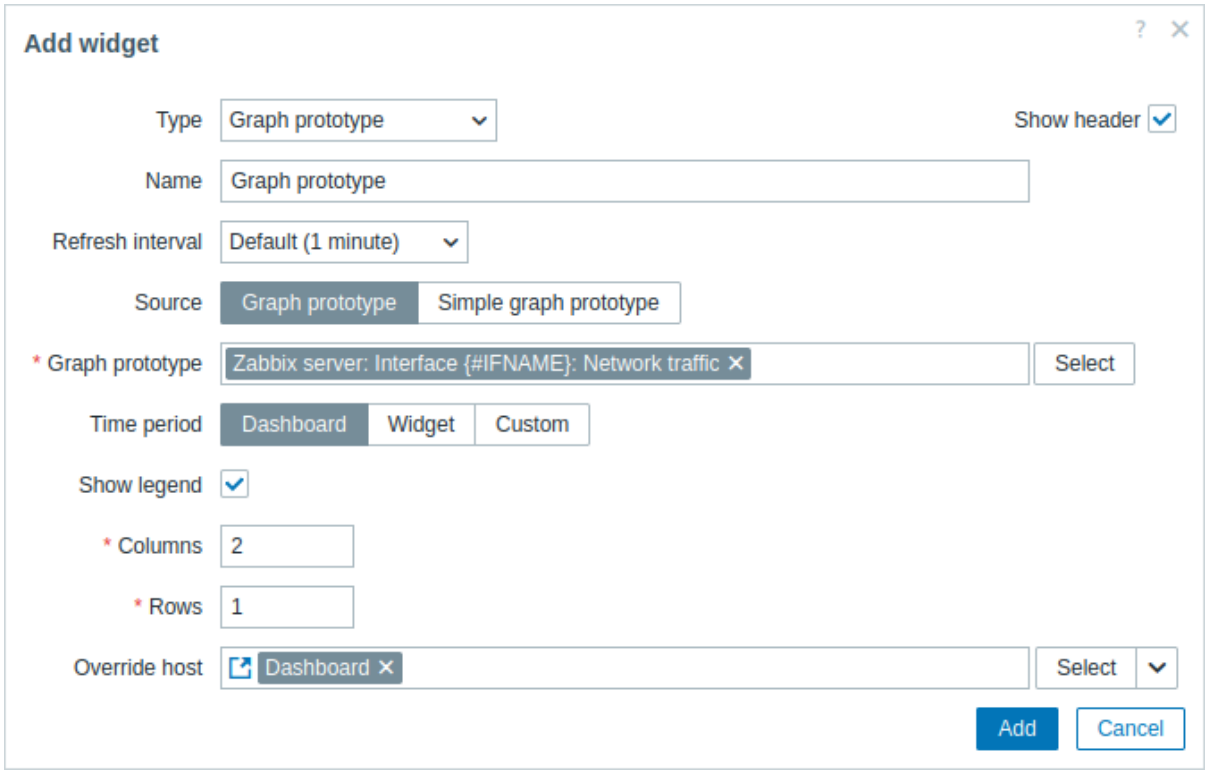
10 グラフのプロトタイプ

概要

グラフのプロトタイプウィジェットでは、低レベルディスカバリによってプロトタイプグラフまたはプロトタイプアイテムのいずれかから作成されたグラフのグリッドを表示できます。

設定

設定するには、ウィジェットタイプとして *Graph prototype* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

Source	グラフのソースを選択します: **Graph prototype** または **Simple graph prototype**。
Graph prototype	グラフプロトタイプによってディスカバリされたグラフを表示するためのグラフプロトタイプを選択します。 このパラメータは、*Source* が「Graph prototype」に設定されている場合に使用できます。
Item prototype	アイテムプロトタイプによってディスカバリされたアイテムのシンプルグラフを表示するためのアイテムプロトタイプを選択します。 このパラメータは、*Source* が「Simple graph prototype」に設定されている場合に使用できます。
Time period	グラフにデータを表示する期間を設定します。期間のデータソースとしてdata sourceを選択します: **Dashboard** - *Time period* セレクターをデータソースとして設定します。 **Widget** - *Widget* パラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 **Custom** - *From* および *To* パラメータで指定した期間をデータソースとして設定します。これを設定すると、ウィジェット右上に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。

Widget	期間のデータソースとして使用する互換性のあるウィジェットを入力または選択します。 このパラメータは、*Time period* が「Widget」に設定されている場合に使用できます。
From	期間の開始を入力または選択します。 Relative time syntax (`now`, `now/d`, `now/w-1w` など) をサポートしています。 このパラメータは、*Time period* が「Custom」に設定されている場合に使用できます。
To	期間の終了を入力または選択します。 Relative time syntax (`now`, `now/d`, `now/w-1w` など) をサポートしています。 このパラメータは、*Time period* が「Custom」に設定されている場合に使用できます。
Show legend	このチェックボックスをオフにすると、グラフ上の凡例を非表示にします（デフォルトでオン）。
Override host	ホストのdata sourceとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、template dashboard 上でウィジェットを設定する場合は使用できません。
Columns	グラフプロトタイプウィジェット内に表示するグラフの列数を入力します。
Rows	グラフプロトタイプウィジェット内に表示するグラフの行数を入力します。

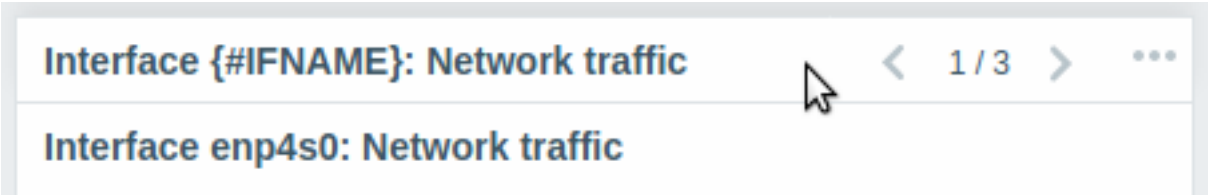
グラフの凡例は、次の 3 つのセクションで構成されます。

- アイテムとその集計値
- パーセンタイル（設定されている場合）
- トリガー（表示されているアイテムに関連付けられている場合）

ウィジェット内のグラフの高さが不十分な場合、凡例は表示されないか、一部のみ表示されることがあります。
最初にトリガーとパーセンタイルが非表示になり、その後にアイテムの凡例が非表示になります。
凡例全体を表示するには、ウィジェットの縦方向のサイズを大きくしてください。

Note:
表示できるトリガー行は最大 3 行です。
トリガーがそれより多い場合は、深刻度の低いトリガーが優先して表示されます。

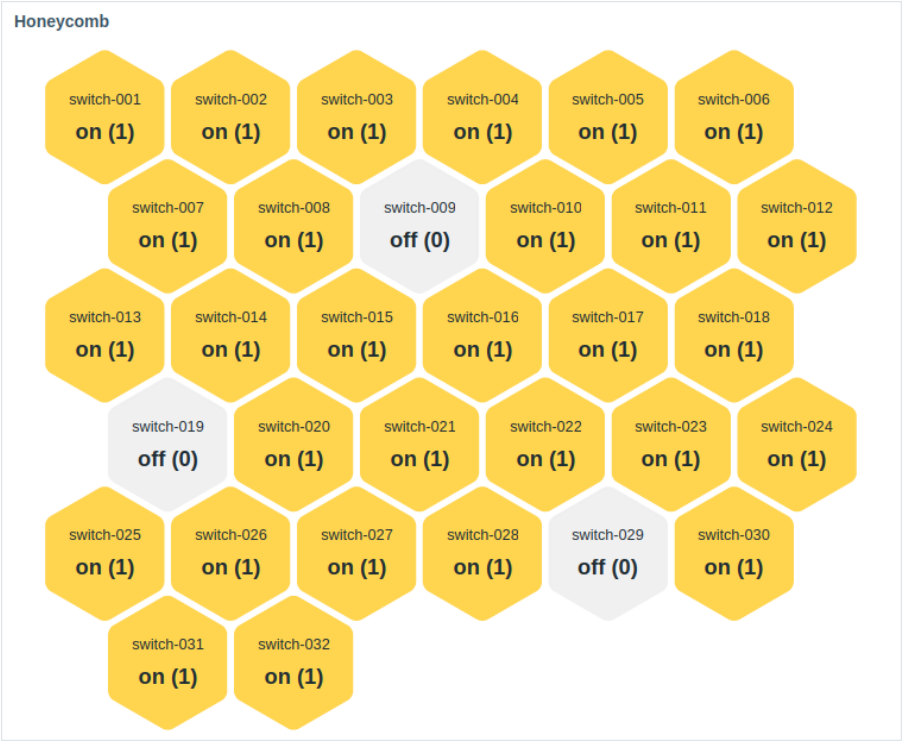
Columns および *Rows* パラメータを使用すると、ウィジェット内に複数のグラフを収めることができますが、ディスカバリされたグラフ数がウィジェット内の列数/行数を上回る場合があります。
この場合、ウィジェットでページ切り替えが利用可能になり、スライドアップするヘッダーから左右の矢印を使ってページを切り替えられます。



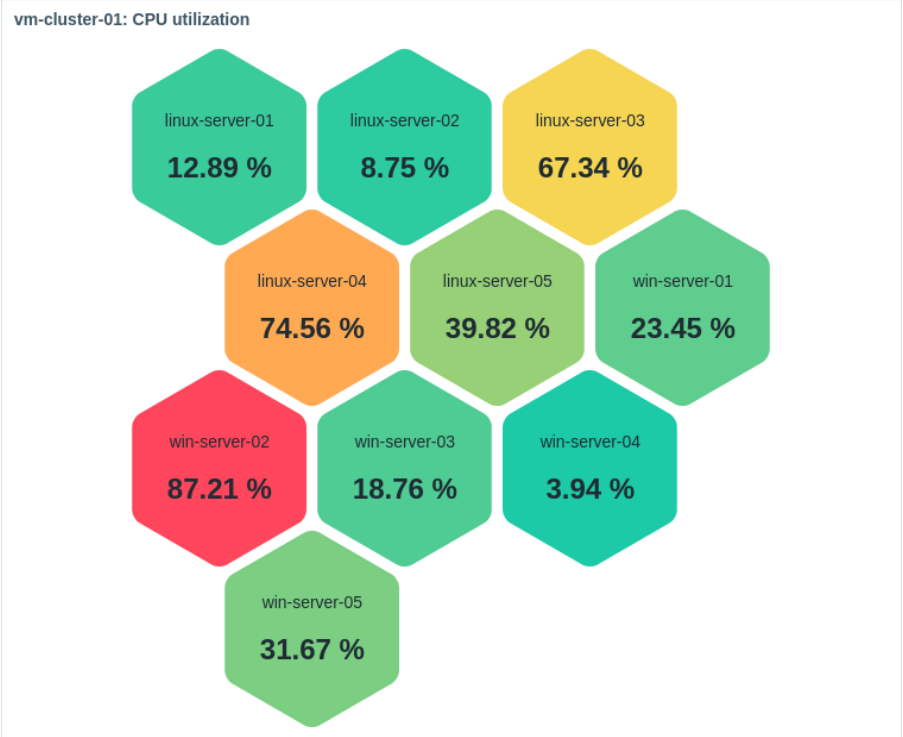
11 ハニカム

概要

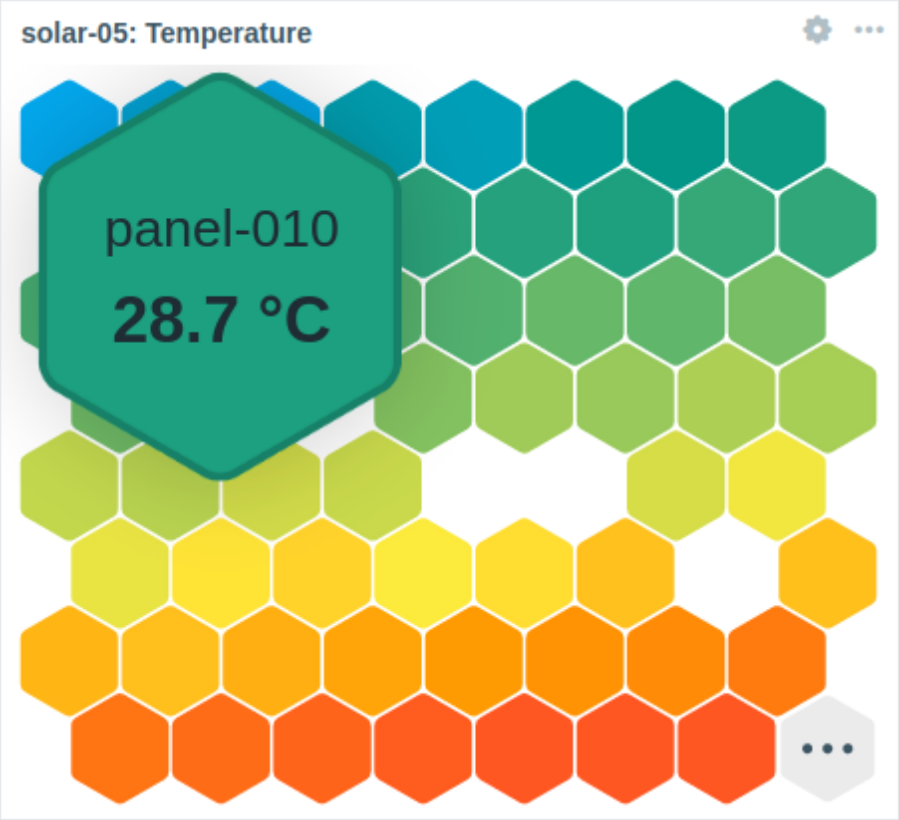
ハニカムウィジェットは、監視対象のネットワークインフラストラクチャとリソースの動的な概要を提供します。仮想マシンやネットワークデバイスなどのホストグループと、それぞれに対応するアイテムが、対話可能な六角形のセルとして視覚的に表示されます。



このウィジェットは、高度な設定オプションを使用して外観を細かく調整でき、多様なビジュアルスタイルを作成できます。



マウスオーバーすると、フォーカスされたハニカムセルが拡大表示され、視認性が向上します。
セルをクリックすると、別のセルが選択されるまでその境界線が強調表示されます。



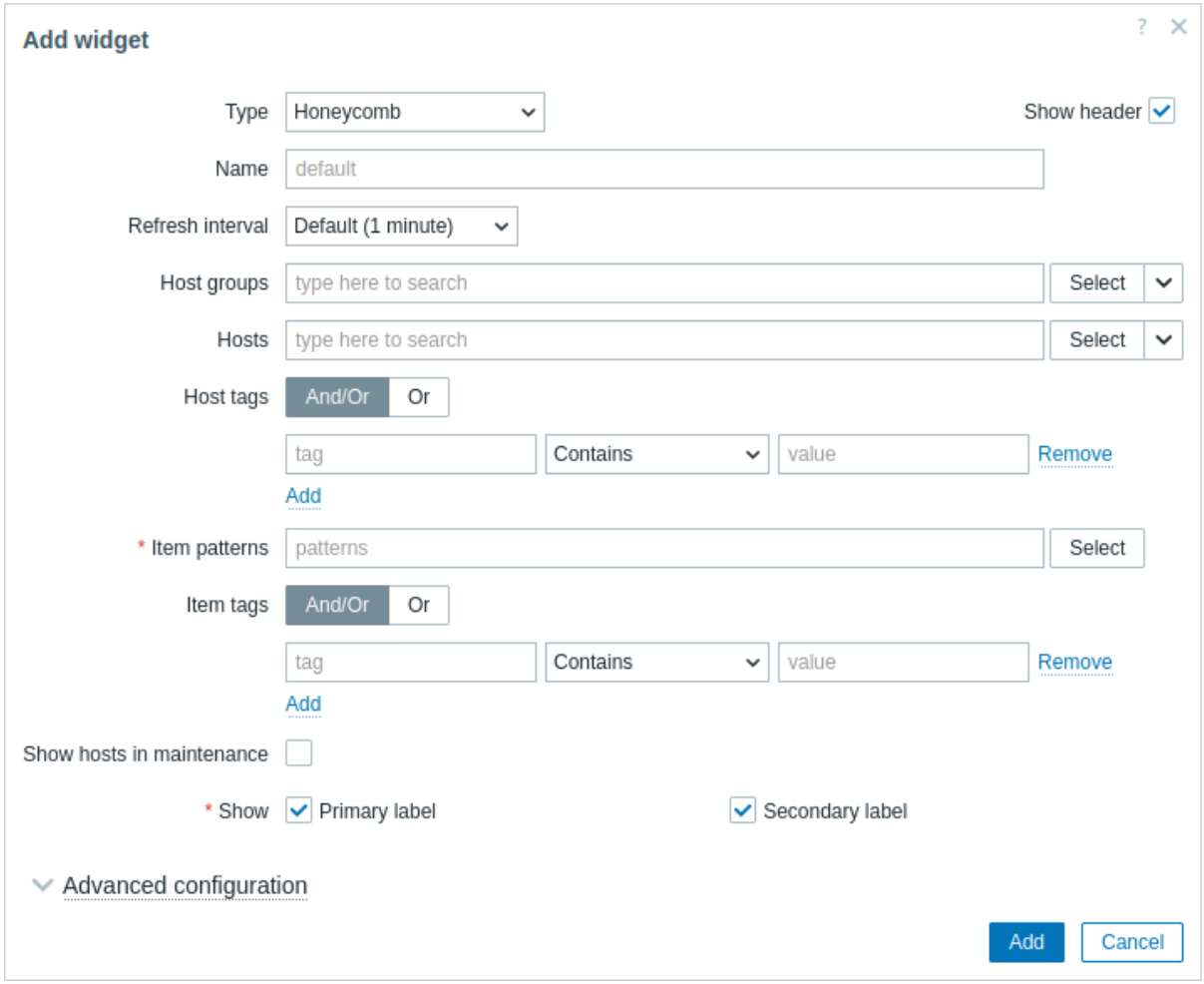
表示されるハニカムセルの数は、ウィジェットのサイズと最小セルサイズ（32 ピクセル）によって制限されます。
すべてのセルをウィジェット内に収められない場合、最後のセルとして省略記号が表示されます。

より多くのセルを表示できるように、ウィジェットのサイズを変更できます。
サイズ変更時には、ハニカムセルのサイズと配置が動的に調整されます。
ハニカムの各行では、総セル数がその行のセル数で割り切れない場合の最終行を除き、各行のセル数は同じになります。

Honeycomb ウィジェットに表示されている情報は、ウィジェットメニューの *Download image* オプションを選択することで、PNG 画像としてダウンロードできます。

設定

設定するには、タイプとして *Honeycomb* を選択します。



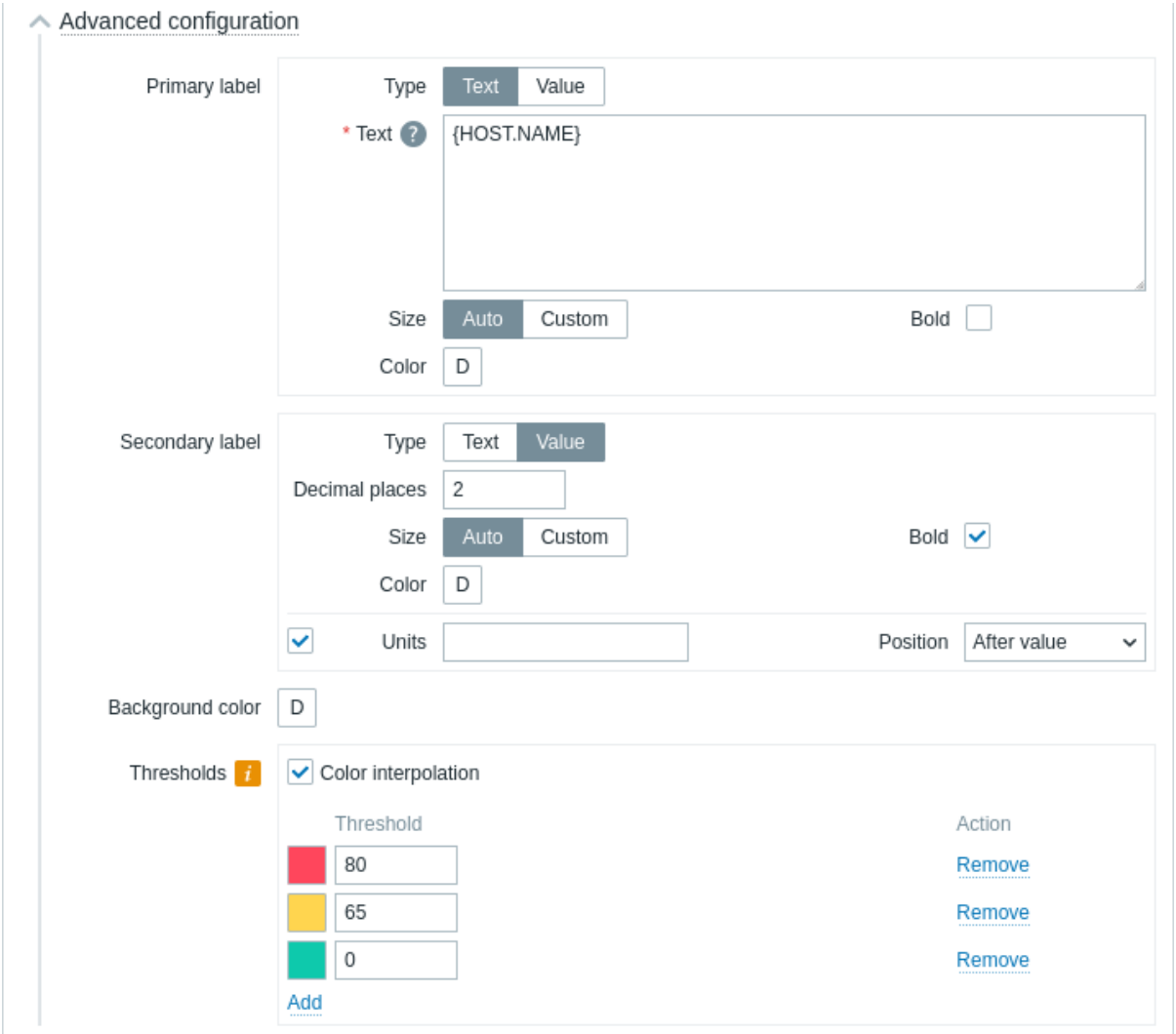
すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

Host groups	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、ネストされたすべてのホストグループも暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットには、選択した *Item pattern* に一致するアイテムを持つホストを含むすべてのホストグループが表示されます（以下を参照）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Hosts	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットには、選択した *Item pattern* に一致するアイテムを持つすべてのホストが表示されます（以下を参照）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

Host tags	ウィジェットに表示するホスト数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。 タグ名の一致では、常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列を含む指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Item patterns	アイテムパターンを入力するか、既存のアイテムをアイテムパターンとして選択します。入力または選択したパターンに一致するアイテムのデータが honeycomb に表示されます。*Item patterns* パラメータは必須です。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致するアイテムを返し、Zabbix* は”Zabbix” で始まるアイテムを返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力を開始すると、選択した *Hosts*、または選択した *Host groups* 内のホストに属する一致アイテムに限定されたドロップダウンリストが表示されます（該当する場合）。ワイルドカード記号は常に解釈されるため、たとえば *item** という名前のアイテムがあっても、他に一致するアイテム（例: item2、item3）が存在する場合は、それだけを個別に追加することはできません。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータではテンプレート上で設定されたアイテムのみ選択できます。
Item tags	ウィジェットに表示するアイテム数を制限するためのタグを指定します。詳細については、上記の *Host tags* を参照してください。
Show hosts in maintenance	このチェックボックスをオンにすると、メンテナンス中のホストを表示します（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータのラベルは *Show data in maintenance* になります。
Show	このチェックボックスをオンにすると、対応する honeycomb セル要素（主ラベル、副ラベル）を表示します。 少なくとも 1 つの要素を選択する必要があります。
Advanced configuration	*Advanced configuration* ラベルをクリックすると、高度な設定オプションが表示されます。

高度な設定

高度な設定オプションは、折りたたみ可能な高度な設定セクションで利用できます。また、表示フィールドで選択した要素（上記参照）、およびハニカムセルの背景色またはしきい値に対してのみ使用できます。



主/副ラベル

タイプ	ラベルのタイプを選択します： テキスト - ラベルにはテキストパラメータで指定した文字列が表示されます。 値 - ラベルには、小数点以下桁数パラメータで指定した小数点以下桁数でアイテムの値が表示されます。
テキスト	ラベルのテキストを入力します。このテキストはデフォルトのアイテム名を上書きする場合があります。 複数行のテキストをサポートしています。テキストとsupported macrosの組み合わせも可能です。 {HOST.*}、{ITEM.*}、{INVENTORY.*}、およびuser macrosをサポートしています。 ハニカムセルはホスト名のアルファベット順に並び、各ホスト内ではアイテム名順に並びます。 このパラメータは、タイプが「テキスト」に設定されている場合に使用できます。
小数点以下桁数	値とともに表示する小数点以下の桁数を入力します。 このパラメータは、タイプが「値」に設定されている場合に使用でき、numeric (float)データを返すアイテムにのみ影響します。
サイズ	ラベルサイズを選択します： 自動 - 自動調整されたラベルサイズを使用します。 カスタム - カスタムのラベルサイズを入力します（ハニカムセルサイズに対する割合）。 ハニカムセルサイズに収まらないラベルは切り詰められることに注意してください。
太字	アイテムの単位を太字で表示するには、チェックボックスをオンにします。
色	カラーピッカーからアイテムの単位の色を選択します。 「D」はデフォルト色を表し、これは Web インターフェースのテーマに依存します。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。

単位

単位	アイテムの値とともに単位を表示するには、チェックボックスをオンにします。 単位名を入力すると、アイテム設定で設定された単位を上書きします。 このパラメータは、タイプが「テキスト」に設定されている場合に使用できます。
位置	アイテムの単位の位置（アイテムの値の前または後）を選択します。 このパラメータは、次の時間関連の単位では無視されます：unixtime、uptime、s。 このパラメータは、タイプが「テキスト」に設定されている場合に使用できます。

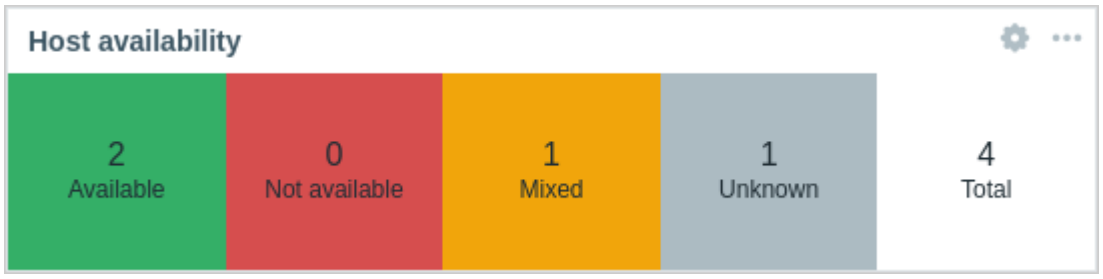
背景色	
背景色	カラーピッカーからハニカムセルの背景色を選択します。 「D」はデフォルト色を表し、これは Web インターフェースのテーマに依存します。デフォルト色に戻すには、カラーピッカーのデフォルトを使用ボタンをクリックします。
しきい値	
色補間	ハニカムセルのしきい値色の間を滑らかに遷移させるには、チェックボックスをオンにします。 このパラメータは、2 つ以上のしきい値が設定されている場合に使用できます。
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択して、数値を指定します。 しきい値の一覧は保存時に昇順で並べ替えられます。 しきい値として設定した色は、数値アイテムに対してのみ正しく表示されることに注意してください。 接尾辞（たとえば、”1d”、”2w”、”4K”、”8G”）をサポートしています。値のマッピングをサポートしています。

12 ホスト稼働状況

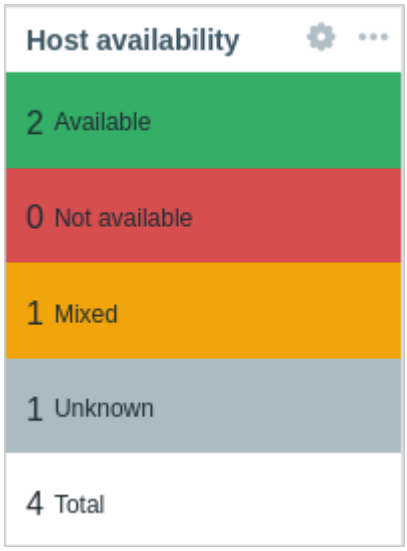
概要

ホストの可用性ウィジェットでは、選択したレイアウトに応じて、ホストの可用性に関する高レベルの統計情報が色付きの列/行で表示されます。

横表示（列）：



縦表示（行）：



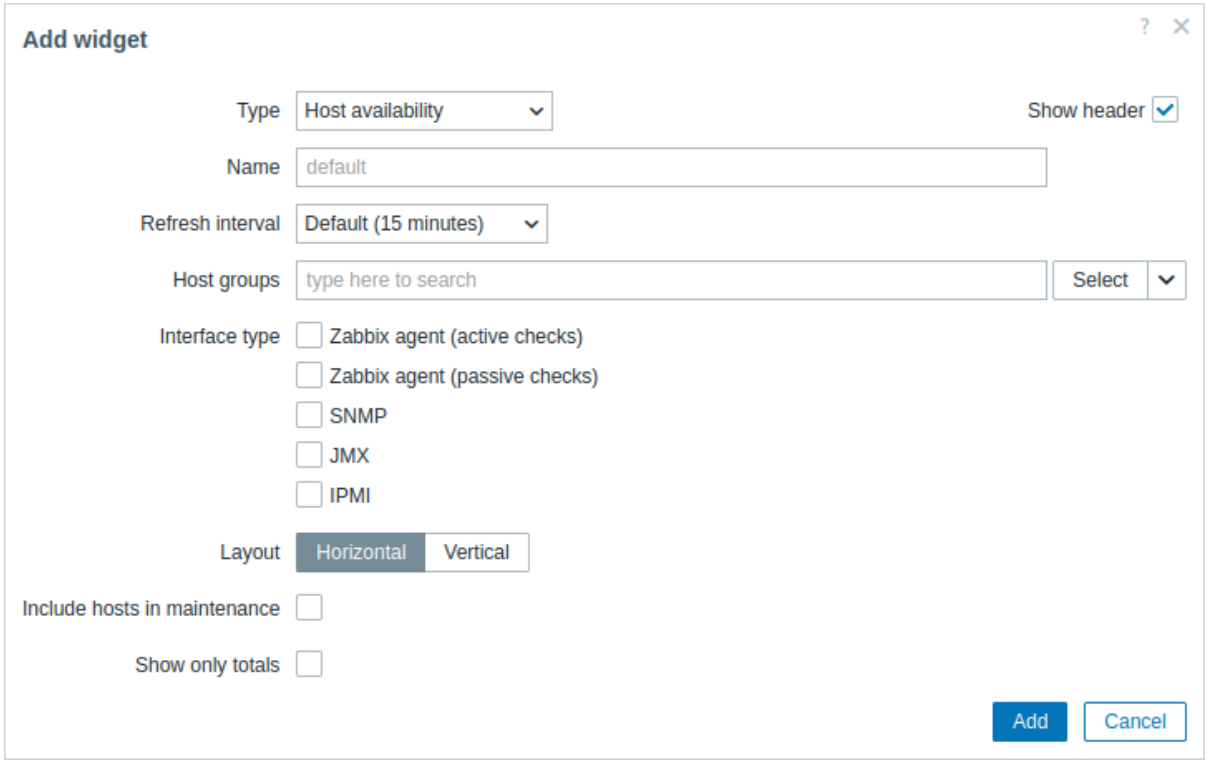
各列/行のホストの可用性は、次のようにカウントされます。

- 利用可能 - すべてのインターフェースが利用可能なホスト
- 利用不可 - すべてのインターフェースが利用不可なホスト
- 混在 - 少なくとも 1 つのインターフェースが利用不可で、かつ少なくとも 1 つのインターフェースが利用可能または不明であるホスト。その他のインターフェースは、「不明」を含む任意の値を取ることができます
- 不明 - 少なくとも 1 つのインターフェースが不明であるホスト（利用不可のインターフェースはない）
- 合計 - すべてのホストの合計

Note:
Zabbix エージェント（アクティブチェック）の場合、このタイプのアイテムは複数のインターフェースを持てないため、混在のセルは常に空になります。

設定

設定するには、タイプとしてホストの可用性を選択します。



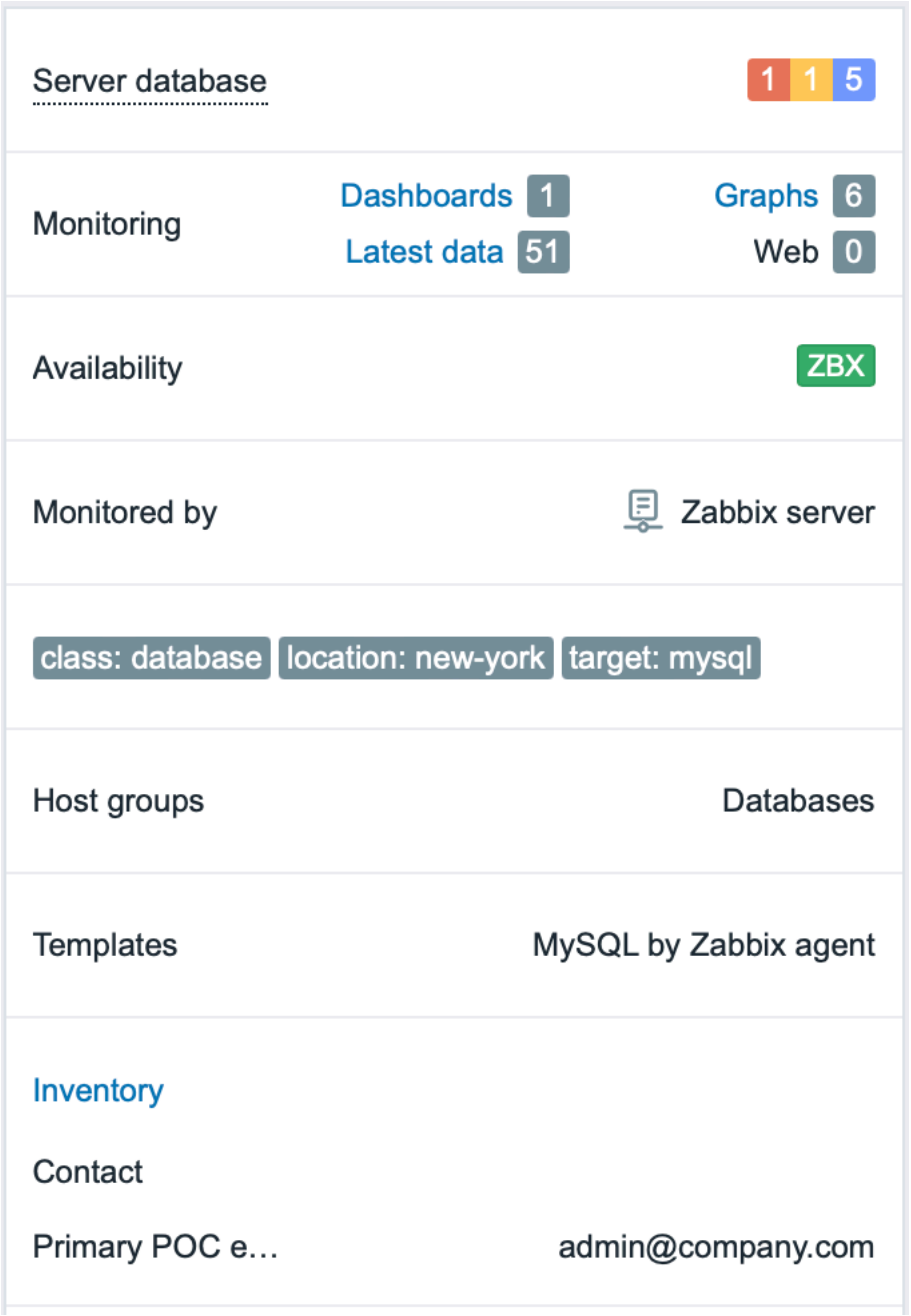
すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
インターフェースタイプ	可用性データを表示するホストインターフェースを選択します。 何も選択しない場合、デフォルトですべてのインターフェースの可用性が表示されます。
レイアウト	横方向表示（列）または縦方向表示（行）を選択します。
メンテナンス中のホストを含める	メンテナンス中のホストを統計に含めます。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータのラベルはメンテナンス中のデータを表示になります。
合計のみ表示	チェックすると、インターフェースごとの内訳なしで、ホストの合計が表示されます。インターフェースが 1 つだけ選択されている場合、このオプションは無効になります。

13 ホストカード

概要

ホストカードウィジェットは、単一のホストに関する包括的でカスタマイズ可能な情報を提供します。ホストは、ウィジェット設定で指定するか、別の互換性のあるウィジェットから選択するか、ダッシュボードレベルのホストセレクターを使用して特定できます。

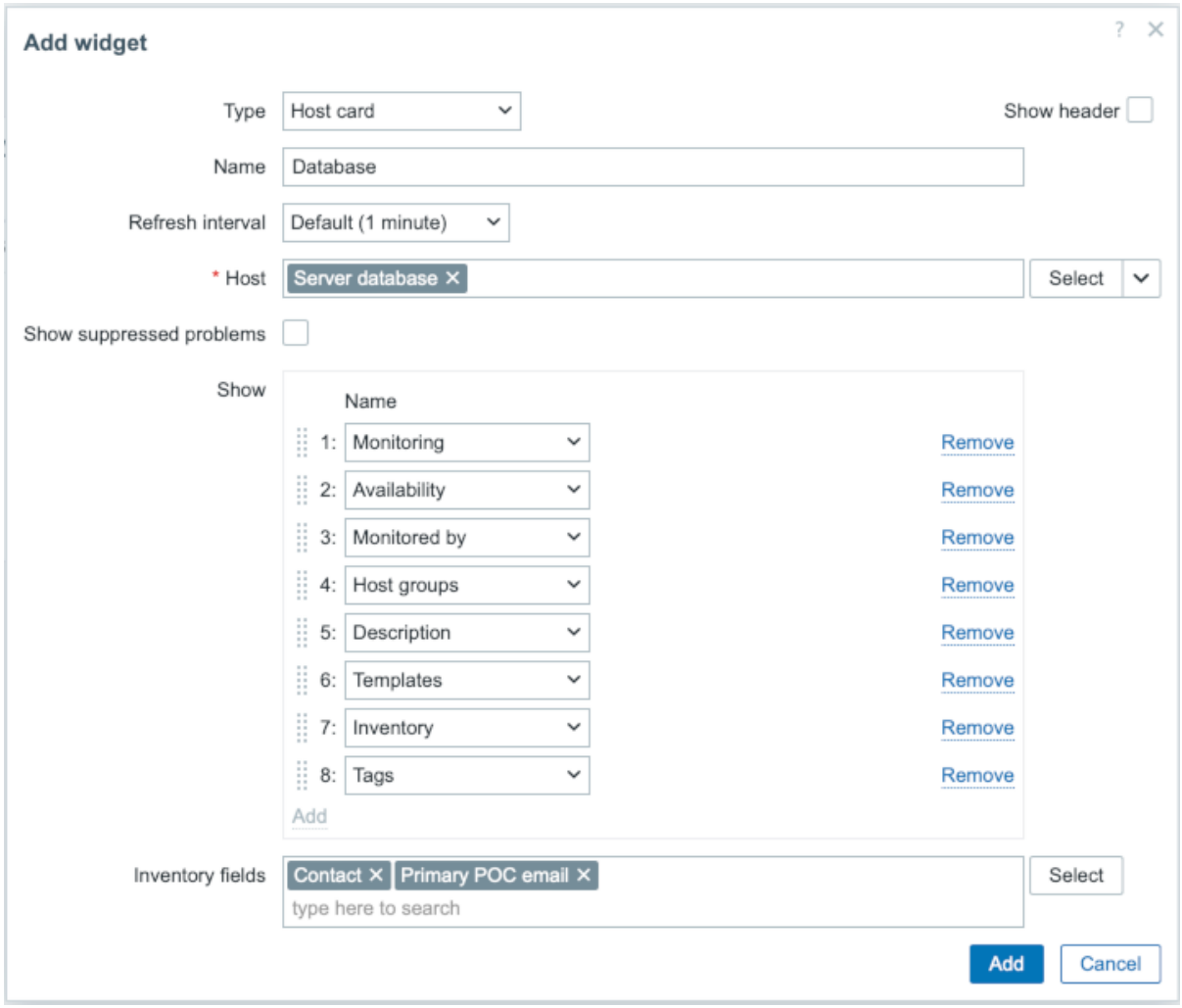


ウィジェットの上部には、ホスト名と、深刻度ごとにグループ化されたホストの障害数が表示されます。ホストが無効になっている場合、ホスト名の横に *Disabled* という文字が表示されます。ホストがメンテナンス中の場合、ホスト名の横にメンテナンスアイコンが表示されます。ウィジェットのその他のセクションは、必要に応じてカスタマイズできます。

このウィジェットは、幅に応じてレイアウトを自動的に調整します。ウィジェットのセクションを複数列に再配置するには、ウィジェットを水平方向に広げてください。

設定

設定するには、タイプとしてホストカードを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のパラメータを設定できます。

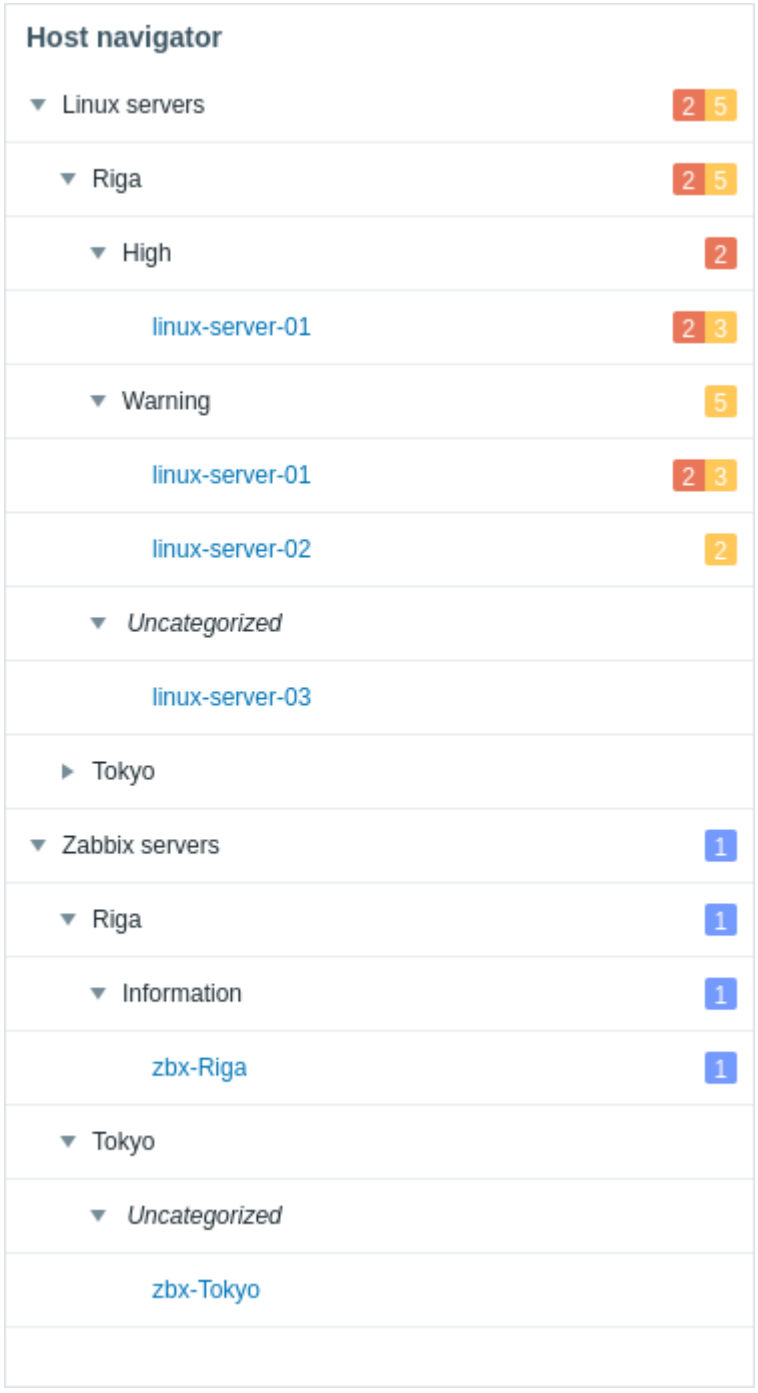
ホスト		ホストを選択します。 または、互換性のあるウィジェットまたはダッシュボードをデータソースとして選択します。 このフィールドでは、名前の入力を開始すると一致する候補が表示されます。Select ボタンを押して手動で選択することもできます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
抑制された障害を表示		このチェックボックスをオンにすると、障害の計算に抑制された障害を含めます。
表示		ウィジェットに表示するセクションを最大 8 つまで設定します。セクションはドラッグ＆ドロップで並べ替えることができます。
	ホストグループ	ホストが属するホストグループ。
	説明	ホストの説明。
	監視	ホストのダッシュボード、最新データ、グラフ、Web シナリオに移動するためのクイックリンク。各リンクの横の数字はエンティティ数を示します。
	可用性	インターフェースごとのホストの可用性。
	監視元	ホストを監視している Zabbix サーバー、プロキシ、またはプロキシグループ。
	テンプレート	ホストにリンクされたテンプレート。
	インベントリ	ホストインベントリのフィールドを表示します。選択すると、表示するフィールドを指定できるインベントリフィールドパラメータが表示されます。

	タグ	ホストタグ。
インベントリフィールド		表示するインベントリフィールドを指定します。 このフィールドでは、名前の入力を開始すると一致する候補が表示されます。Select ボタンを押して手動で選択することもできます。 インベントリフィールドが指定されていない場合は、値が設定されているすべてのインベントリフィールドが表示されます。 このパラメータは、表示セクションでインベントリが選択されている場合にのみ使用可能になります。

14 ホストナビゲーター

概要

ホストナビゲーターウィジェットは、さまざまなフィルタリングおよびグループ化オプションに基づいてホストを表示します。



このウィジェットでは、選択したホストに基づいて、他のウィジェットに表示される情報を制御することもできます。

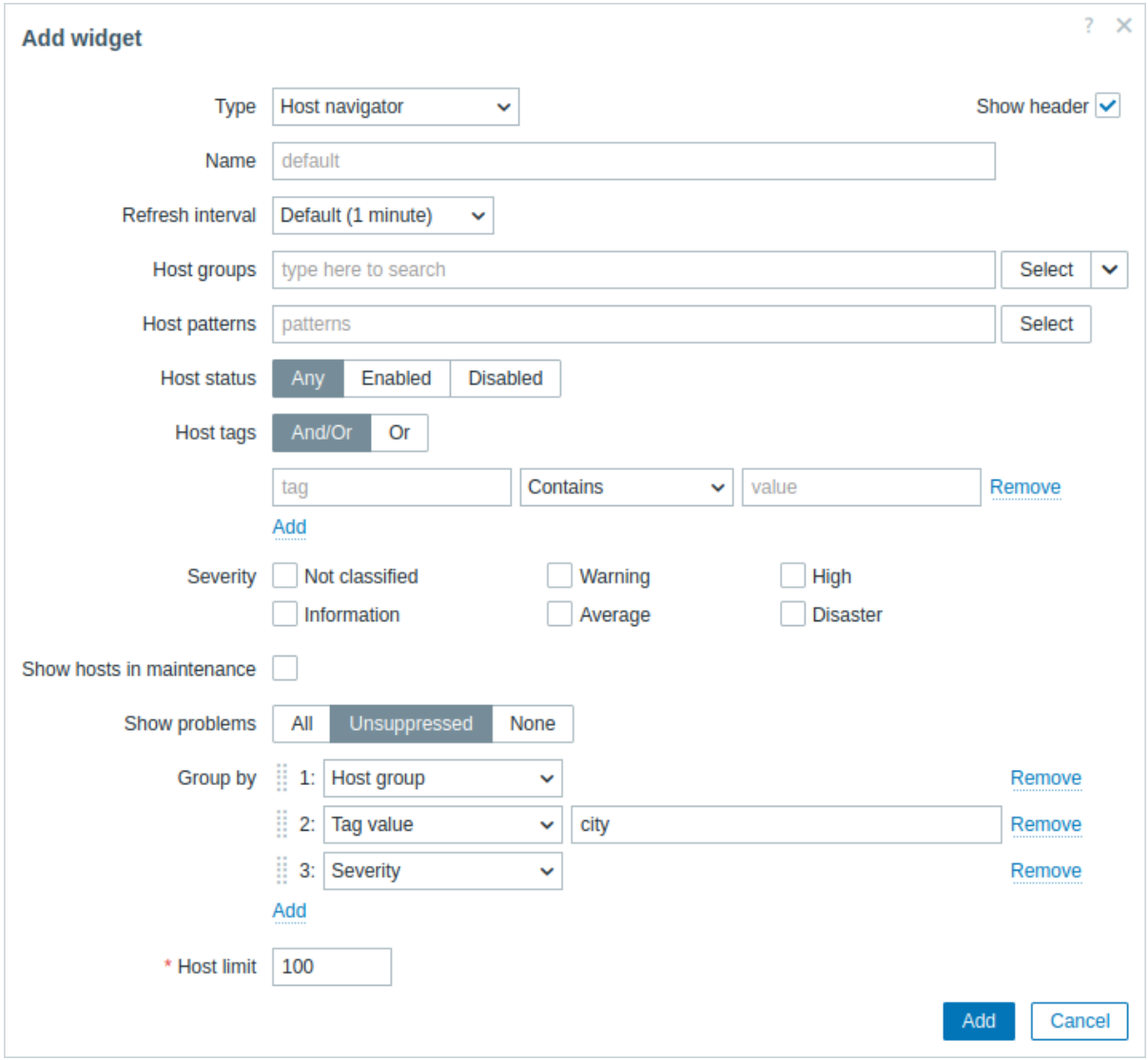


ホストが整理されるグループは、展開または折りたたみできます。

グループ、障害、およびメンテナンス中のホストについては、マウスオーバーヒントで追加の詳細を確認できます。

設定

設定するには、タイプとして ホストナビゲーターを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリート対応のため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、ネストされたすべてのホストグループも暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットにはすべてのホストグループのすべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホストパターン	ホストパターンを入力するか、既存のホストをホストパターンとして選択します。指定したパターンに一致するホストがホストナビゲーターに表示されます。 このフィールドはオートコンプリート対応のため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットにはすべてのホストが表示されます。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致するホストを返し、Zabbix* は"Zabbix" で始まるホストを返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力を開始すると、選択されている ホストグループに属するホストに限定された一致ホストのドロップダウンリストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

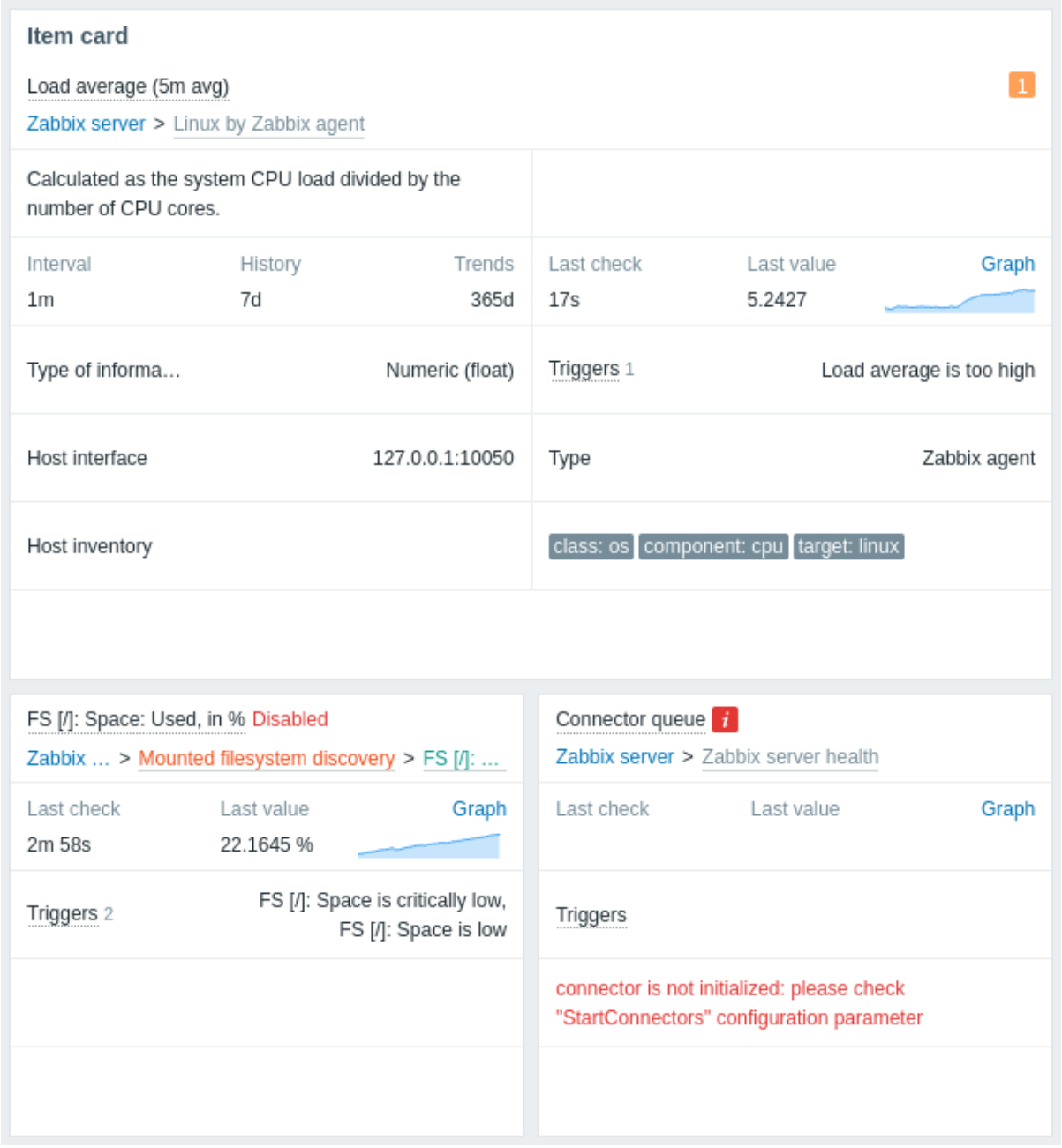
ホストステータス	ステータス（任意、有効、無効）に基づいて、表示するホストを絞り込みます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホストタグ	ウィジェットに表示するホストを絞り込むためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。 タグ名の照合では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列を含む指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
重要度	問題のあるホストをウィジェットに表示するために、問題の重要度をマークします。 重要度が 1 つもマークされていない場合、すべての問題を持つすべてのホストが表示されます。
メンテナンス中のホストを表示	このチェックボックスをオンにすると、メンテナンス中のホストを表示します（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータのラベルは メンテナンス中のデータを表示になります。
問題を表示	ステータス（すべて、抑制されていないもの、なし）に基づいて、ウィジェット内でホストとともに表示する問題を絞り込みます。
グループ化	選択したホストをグループ化するための属性を追加します。 ホストグループ - ホストをホストグループごとにグループ化します。 タグ値 - タグ名を入力して、そのタグの値ごとにホストをグループ化します（たとえば、”city” を入力すると、”Riga”、”Tokyo” などの値ごとにホストをグループ化します）。 重要度 - 問題の重要度ごとにホストをグループ化します。 問題を表示が問題を表示するよう設定されている場合、問題は以下のように表示されます。 - 各重要度グループでは、対応する問題数のみが表示されます。 - 各ホストでは、そのホストのすべての問題数が表示されます。 重要度パラメータでマークされた重要度に対してのみホストがグループ化される点に注意してください。重要度が 1 つもマークされていない場合、すべてのホストがすべての重要度でグループ化されます。 グループ化属性は、グループ名の前にあるハンドルを上下にドラッグして並べ替えることができます。グループ化属性の順序によって、グループのネスト順が決まる点に注意してください。たとえば、複数のタグ名（1: color、2: city）を指定すると、ホストはまず color（red、blue など）で、その後 city（Riga、Tokyo など）でグループ化されます。 設定されたグループ化属性によっては、1 つのホストが複数のグループに表示される場合があります（たとえば、ホストグループでグループ化し、そのホストが複数のホストグループに属している場合）。そのようなホストをクリックすると、すべてのグループで選択および強調表示されます。 設定されたグループ化属性に一致しないホストは、未分類グループに表示されます。 最大 10 個のグループ化属性を指定でき、すべて一意である必要があります。 グループ化属性が指定されていない場合、ホストはグループ化されません。

ホスト上限	表示するホストの最大数を入力します。指定可能な値は 1～9999 です。 表示可能なホスト数が設定した上限を超える場合、表示されたホストの下に対応するメッセージが表示されます（たとえば、”100 of 100+ hosts are shown”）。 設定したホスト上限は、設定済みグループの表示にも影響する点に注意してください。たとえば、ホスト上限を 100 に設定し、タグ値（200 超）でホストをグループ化した場合、対応するホストを含む最初の 100 個のタグ値のみがウィジェットに表示されます。 このパラメータは、*Administration* → *General* → *GUI* の *Limit for search and filter results* パラメータの影響を受けません。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

15 アイテムカード

概要

アイテムカードウィジェットは、単一のアイテムに関する包括的でカスタマイズ可能な情報を提供します。

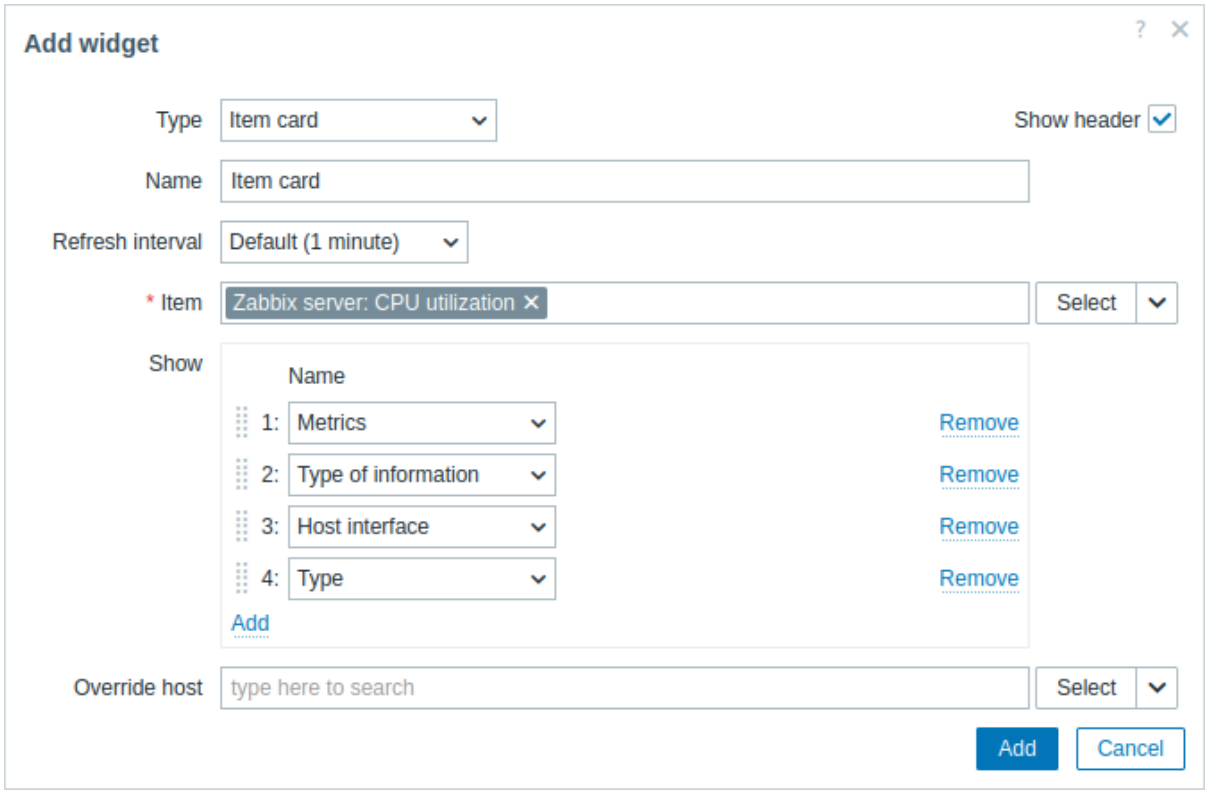


アイテムは、ウィジェット設定で直接指定することも、ダッシュボード上の互換性のあるウィジェットから選択することもできます。

このウィジェットでは複数のセクションを表示でき、それぞれに異なる種類の情報（アイテムの説明、最新データなど）が表示されます。
セクションのレイアウトは、ウィジェットの幅に応じて自動的に調整されます。
水平方向に拡張すると、セクションは複数の列に再配置され、各行で左から右の順に並べられます。

設定

設定するには、タイプとして アイテムカードを選択します。



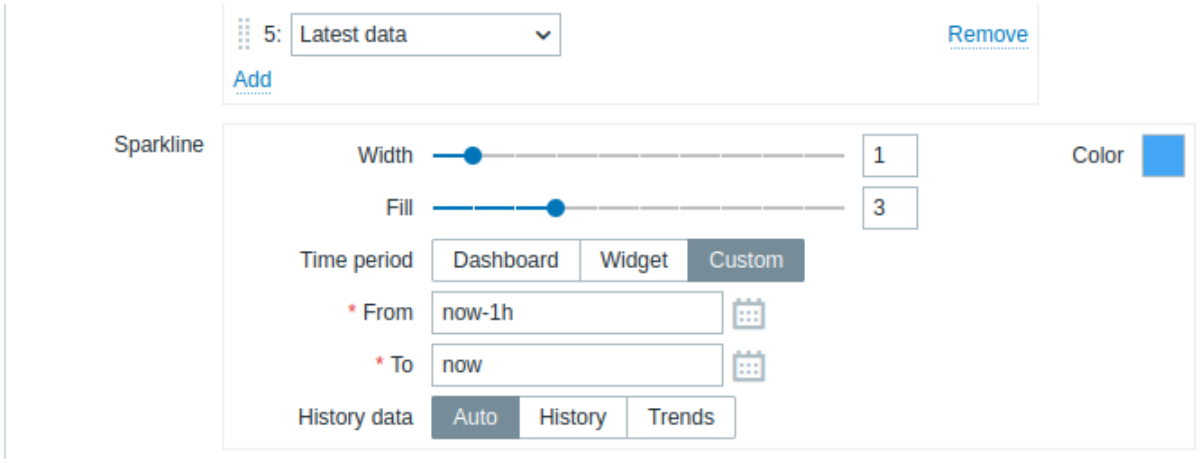
すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

アイテム		アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。 入力すると、一致する候補が表示されます。 ウィジェットの上部セクションには常に アイテム名が表示されます（クリックするとアイテムメニューが開きます）。 - アイテムが無効な場合、アイテム名の横に赤い *Disabled* ラベルが表示されます。 - エラーがある場合は、文字「i」の付いた四角いアイコンが表示されます。障害がある場合は、障害の深刻度ごとに障害件数を示す四角いアイコンが表示されます。アイコンにマウスオーバーすると、詳細を示すポップアップが開きます。 - テンプレートダッシュボードでは、アイテムの表示名ではなく技術名が表示されます。 ウィジェットの上部セクションには アイテムパスも表示されます。これは、そのホストと、該当する場合はテンプレート、ローレベルディスカバリ（LLD）ルール、またはマスターアイテムを示します（エンティティをクリックするとその設定が開きます）。 アイテムパスは、それぞれのエンティティに対する権限を持つユーザーにのみ表示されます。
表示		ウィジェットに表示するセクションを追加します。セクションはドラッグ＆ドロップで並べ替えることができます。
	説明	アイテムの説明を表示します。マクロは展開されます。
	エラーテキスト	アイテムのエラー（存在する場合）を表示します。

メトリクス	以下の情報を表示します。 - アイテムの更新間隔（トラッパーアイテムを除く）。 - アイテムの履歴設定（存在する場合）。 - アイテムのトレンド設定（存在する場合）。 マクロは展開されます。 アイテムがカスタム間隔を使用している場合、アイテムの更新間隔の横にヘルプアイコンが表示されます。アイコンにマウスオーバーすると、カスタム間隔の詳細を示すポップアップが開きます。
最新データ	以下の情報を表示します。 - 前回のアイテムチェックからの経過時間。 - アイテムの最新値。 - アイテム値のシンプルグラフ/履歴へのリンク（履歴またはトレンドが保存されている場合）。 - 設定可能なsparklineチャート（アイテムが数値型の場合）。 バイナリ型のアイテムでは、最新値の代わりにサムネイルまたは *Show* オプションが表示されます。サムネイルにマウスオーバーすると、画像を示すポップアップが開きます。*Show* にマウスオーバーすると、アイテム値（Base64 文字列）を示すポップアップが開きます。 テンプレートダッシュボードでは、アイテムの最新値の代わりに *No data* が表示されます。
情報の型	アイテムの情報の型を表示します。
トリガー	以下の情報を表示します。 - トリガー数を含む *Triggers* ラベル（*Triggers* にマウスオーバーすると、トリガーの詳細を示すポップアップが開きます）。 - トリガー名の一覧（マクロは展開されます）。
ホストインターフェース	アイテムで使用されるホストインターフェースを表示します。アイテムにホストインターフェースがない場合（例: 依存アイテム、シンプルチェック）は *No data* を表示します。 テンプレートダッシュボードでは、*No data* が表示されます。
タイプ	アイテムタイプを表示します。
ホストインベントリ	アイテム値が格納されるホストインベントリフィールドを表示します。
タグ	アイテムタグを表示します。
ホストを上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定している場合は使用できません。

スパークライン

スパークラインのグラフ設定オプションは、最新データセクションがウィジェットに追加されている場合にのみ使用できます。



幅	スライダーを使用するか、0～10 の範囲で値を手動入力して、グラフ線の太さを設定します。
色	線と塗りつぶしの色を選択します。
塗りつぶし	スライダーを使用するか、0～10 の範囲で値を手動入力して、塗りつぶし色の透明度を設定します。

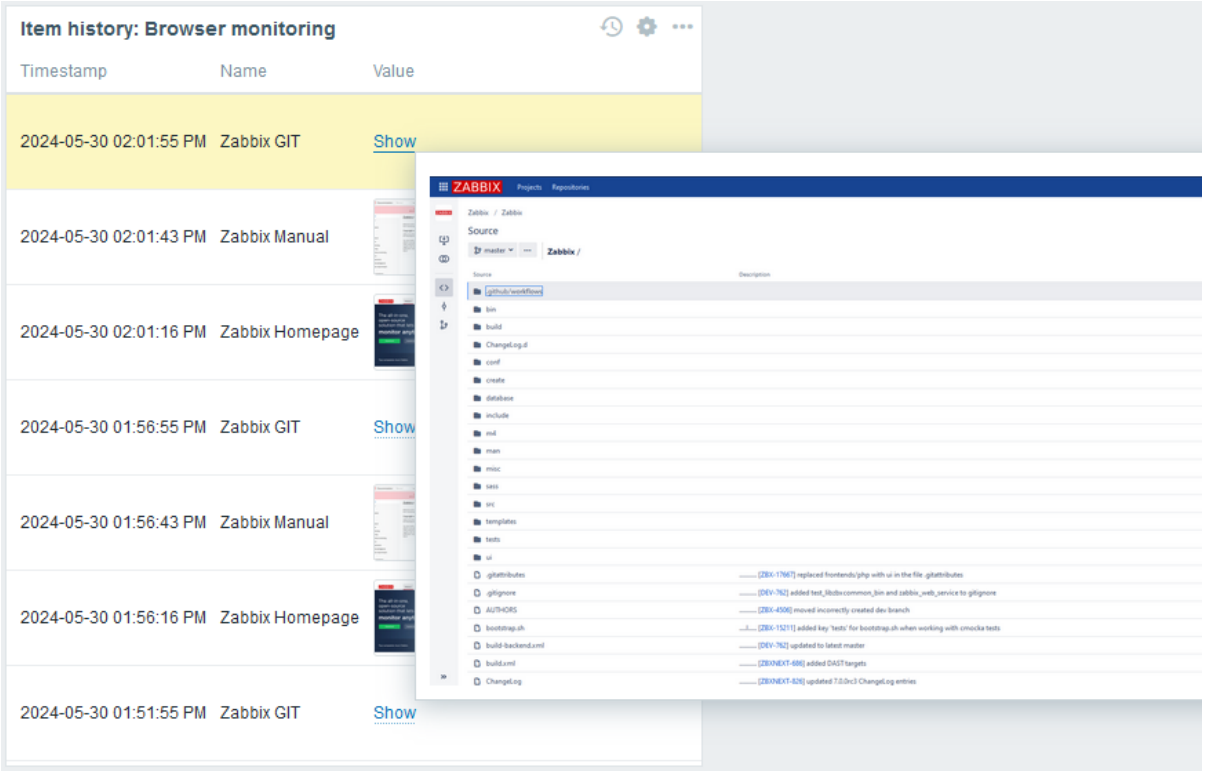
期間	期間のdata sourceを選択します: ダッシュボード - 期間セレクターをデータソースとして設定します。 ウィジェット - ウィジェットパラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 カスタム - 開始および 終了パラメータで指定した期間をデータソースとして設定します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。
ウィジェット	期間のデータソースとして、互換性のあるウィジェットを入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に使用できます。
開始	期間の開始時刻を入力または選択します。 相対時間構文 (now, now/d, now/w-1w など) をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。
終了	期間の終了時刻を入力または選択します。 相対時間構文 (now, now/d, now/w-1w など) をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。
履歴データ	履歴またはトレンドからデータを取得します: 自動 - 自動選択。 履歴 - 履歴データを取得します。 トレンド - トレンドデータを取得します。 このパラメータは数値データにのみ適用されます。非数値データは常に履歴から取得されます。

16 アイテムの履歴

概要

アイテム履歴ウィジェットは、さまざまなアイテムタイプ（数値、文字、ログ、テキスト、バイナリ）の最新データを表形式で表示します。また、進行状況バー、バイナリデータ型の画像（ブラウザアイテムで便利）、および値の強調表示（ログファイル監視で便利）も表示できます。

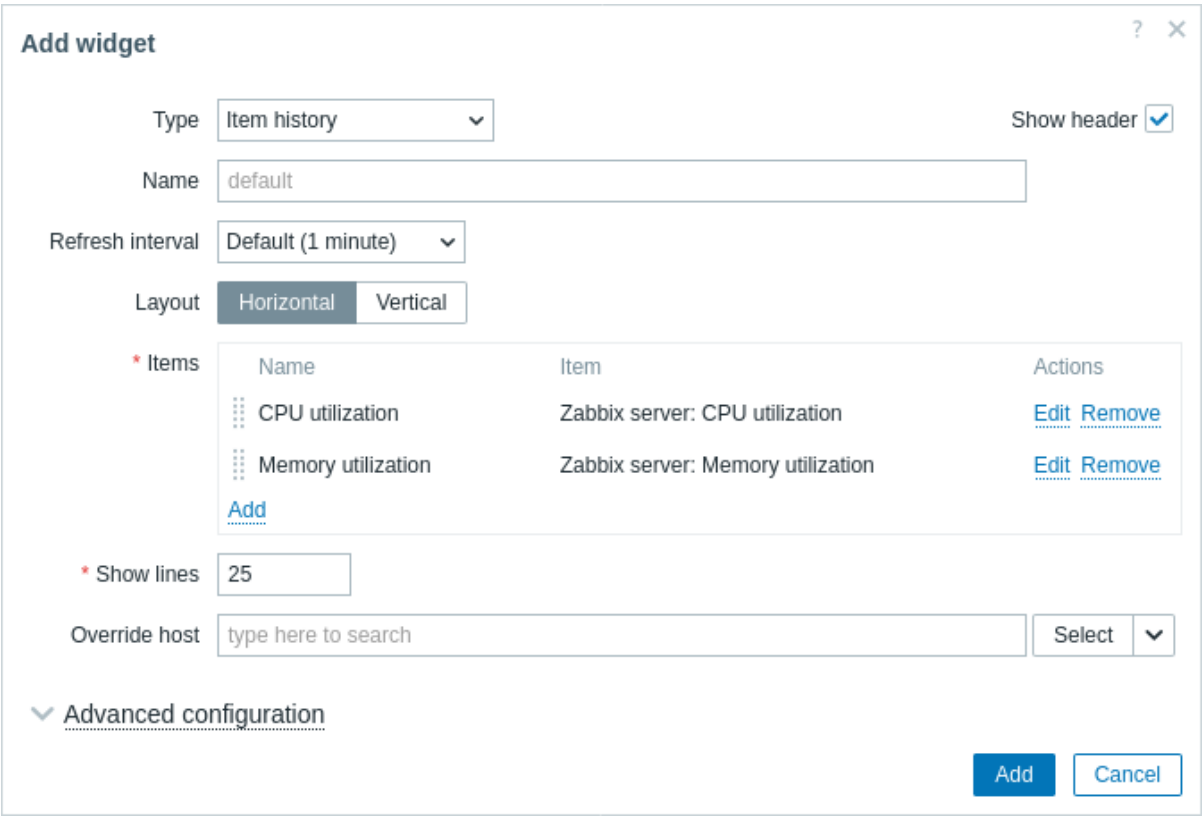




最大 1000 件のレコードを表示できます。

設定

設定するには、タイプとして アイテム履歴を選択します。



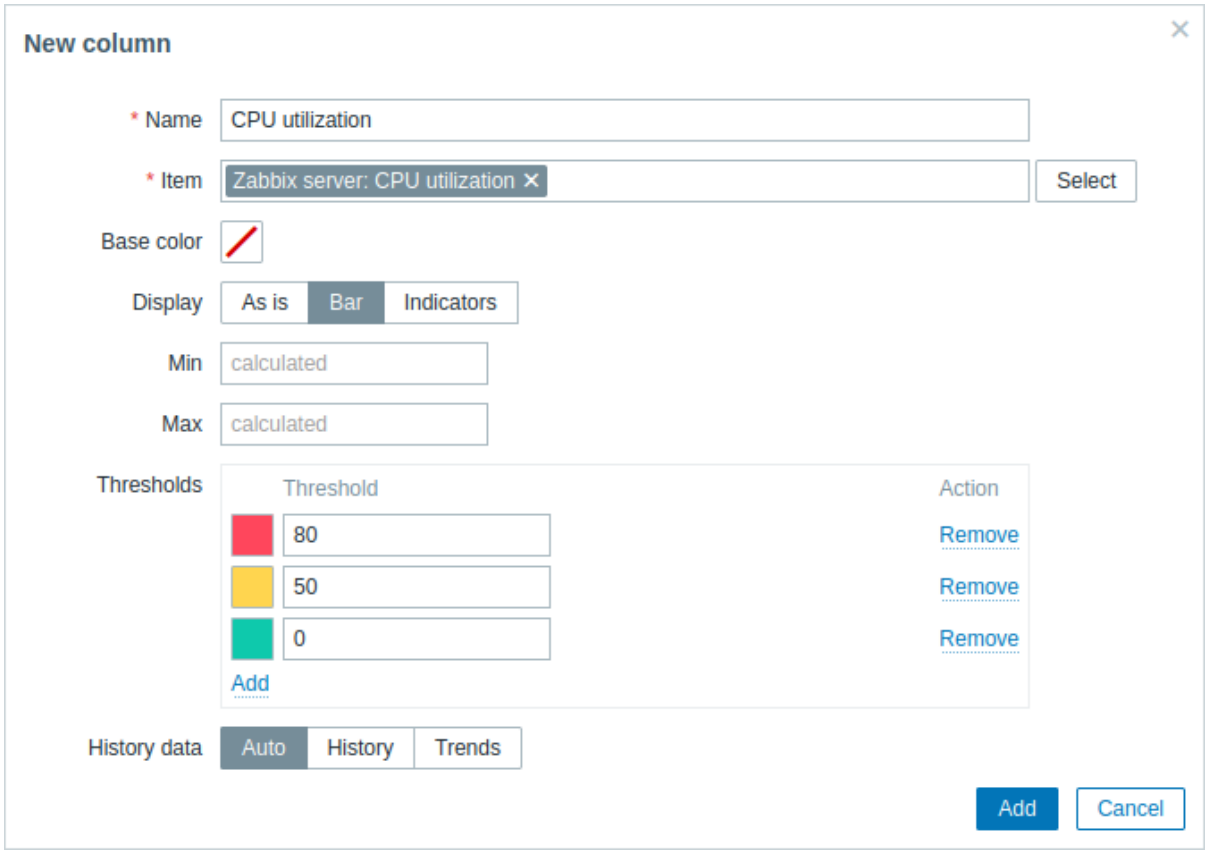
すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

レイアウト	アイテム列のレイアウトオプションを選択します。 横方向 - アイテムは横方向に、値は縦方向に表示されます。 縦方向 - アイテムは縦方向に、値は横方向に表示されます。

アイテム	表示するアイテムの列を追加します。 アイテムの順序によって表示順が決まります。 アイテム名の前にあるハンドルを使って上下にドラッグすることで、アイテムの順序を変更できます。
行数を表示	表示するアイテム値の行数を指定します。
ホストを上書き	ホストのdata sourceとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
高度な設定	高度な設定オプションを表示するには、高度な設定ラベルをクリックします。

列の設定

アイテム列を設定するには、アイテムパラメータで 追加をクリックします。



共通の列パラメータ:

名前	列の名前を入力します。 空欄のままにすると、アイテムパラメータのアイテム名が使用されます。
アイテム	アイテムを選択します。 選択したアイテムの情報タイプに応じて列の設定パラメータは異なることに注意してください。詳細は以下の各パラメータを参照してください。 ウィジェットをテンプレートダッシュボードで設定する場合、選択できるのはテンプレート上で設定されたアイテムのみです。
ベース色	列の背景色を選択します。表示が「バー」または「インジケーター」に設定されている場合は、塗りつぶし色を選択します。 ベース色はしきい値または強調表示の色で上書きされる場合があることに注意してください。

数値型アイテムに固有の列パラメータ:

表示	アイテムの値の表示方法を選択します: そのまま - 通常のテキストとして表示します。 バー - 単色で塗りつぶされたバーとして表示します。 インジケーター - 区切られた色付きバーとして表示します。

最小	バー/インジケーターの最小値を入力します。 空のままにすると、ウィジェットはアイテムの最小値を使用します。 このパラメータは、表示が「バー」または「インジケーター」に設定されている場合にのみ使用できます。
最大	バー/インジケーターの最大値を入力します。 空のままにすると、ウィジェットはアイテムの最大値を使用します。 このパラメータは、表示が「バー」または「インジケーター」に設定されている場合にのみ使用できます。
しきい値	追加をクリックしてしきい値を追加し、カラーピッカーからしきい値の色を選択して、数値を指定します。 しきい値の一覧は、保存時に昇順で並べ替えられます。 接尾辞（例: 「1d」、「2w」、「4K」、「8G」）をサポートしています。値のマッピングをサポートしています。
履歴データ	履歴データとトレンドデータのどちらを使用するかを選択します: 自動 - 自動選択。 履歴 - 履歴データを使用します。 トレンド - トレンドデータを使用します。

文字、テキスト、ログタイプのアイテムに固有の列パラメータ:

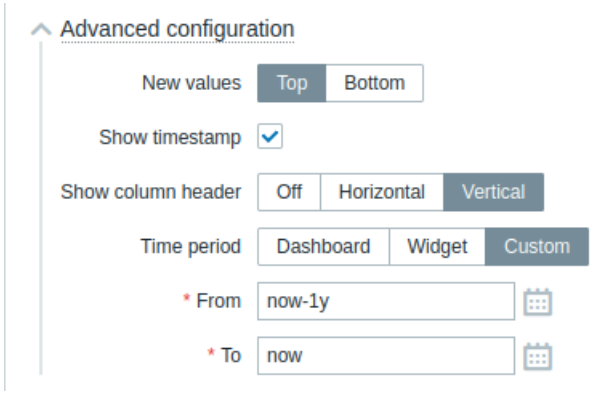
ハイライト	追加をクリックしてハイライトを追加し、カラーピッカーからハイライト色を選択して、正規表現を指定します。 選択した色は、指定した正規表現がテキストに一致するアイテム値の背景色として使用されます。
表示	アイテム値の表示方法を選択します: そのまま - 受信したとおりにそのまま表示します。値に改行が含まれている場合は、改行が保持されます。レイアウトが「縦」に設定され、複数の列が設定されている場合は、折り返して表示されます。 **HTML** - HTML 形式のテキストとして表示します。 単一行 - 単一行として表示し、指定した長さ（1～500 文字）で切り詰めます。 切り詰められた値にマウスオーバーするかクリックすると、完全な値を表示するポップアップが開きます。
等幅フォントを使用	このチェックボックスをオンにすると、アイテム値を等幅フォントで表示します（デフォルトではオフ）。
ローカル時刻を表示	このチェックボックスをオンにすると、タイムスタンプ列にタイムスタンプの代わりにローカル時刻を表示します。 詳細設定の高度な設定にあるタイムスタンプを表示チェックボックスもオンにする必要があることに注意してください。 このパラメータは、ログタイプのアイテムでのみ使用できます。

バイナリタイプのアイテムに固有の列パラメータ:

サムネイルを表示	このチェックボックスをオンにすると、画像バイナリにはサムネイルが表示され、画像以外のバイナリには「表示」オプションが表示されます。 このチェックボックスをオフにすると、すべてのバイナリアイテム値に「表示」オプションが表示されます。 「表示」オプションにマウスオーバーするかクリックすると、アイテム値（画像または Base64 文字列）を含むポップアップウィンドウが開きます。 アイテム値が空文字列の場合は、「表示」オプションが表示され、それにマウスオーバーするかクリックすると、「Empty string」を含むポップアップが開きます。

詳細設定

折りたたみ可能な詳細設定セクションでは、詳細な設定オプションを利用できます。

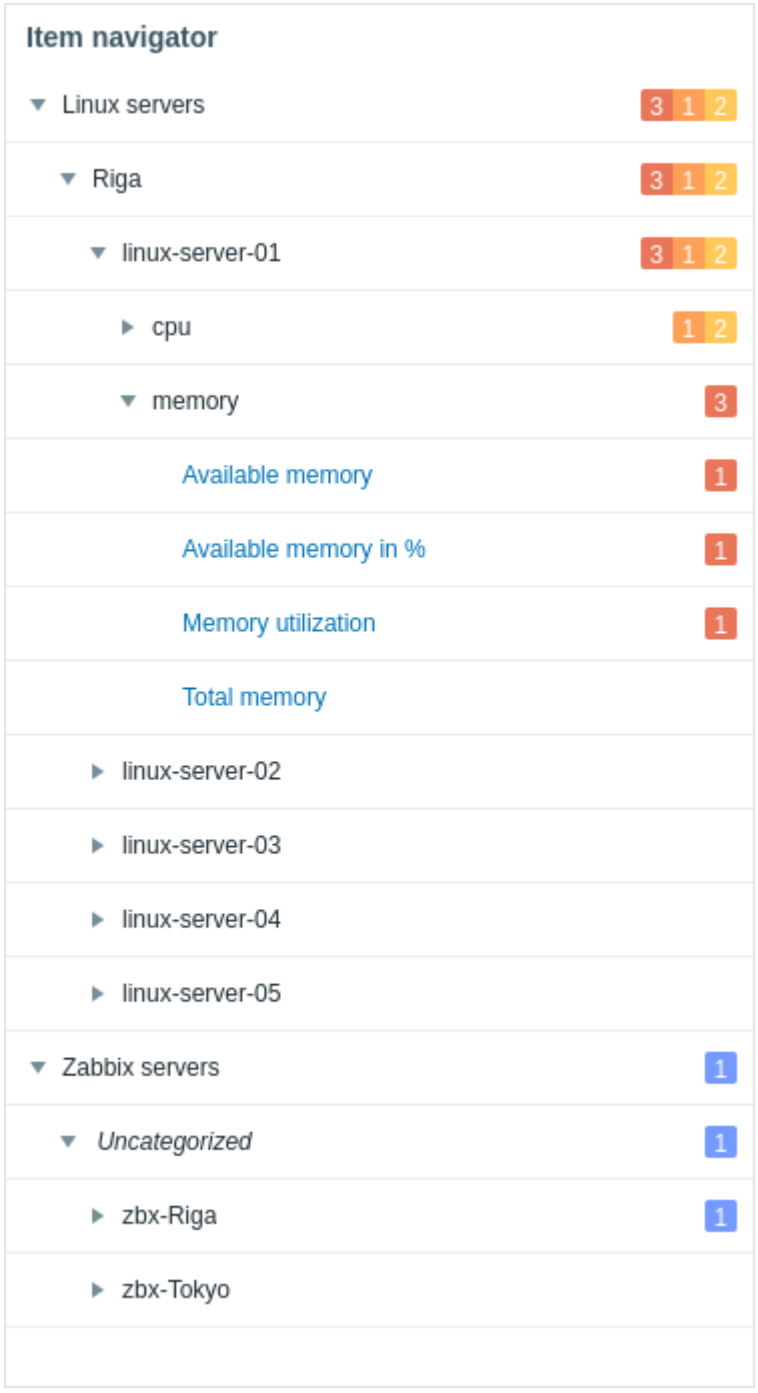


新しい値	新しいアイテムの値を追加する位置を選択します: 上 - 列の上部に追加します。 下 - 列の下部に追加します。
タイムスタンプを表示	このチェックボックスをオンにすると、タイムスタンプ列を表示します（デフォルトではオフ）。
列ヘッダーを表示	列ヘッダーの向きを選択します: オフ - ヘッダーを非表示にします。 横 - ヘッダーを横向きに表示します。 縦 - ヘッダーを縦向きに表示します。
期間	期間のデータソースを選択します: ダッシュボード - 期間セレクターをデータソースとして設定します。 ウィジェット - ウィジェットパラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 カスタム - 開始および終了パラメータで指定した期間をデータソースとして設定します。これを設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。
ウィジェット	期間のデータソースとして、互換性のあるウィジェット（グラフ、グラフ（クラシック）、グラフプロトタイプ）を入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に使用できます。
開始	期間の開始を入力または選択します。 相対時間の構文（now、now/d、now/w-1w など）をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。
終了	期間の終了を入力または選択します。 相対時間の構文（now、now/d、now/w-1w など）をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。

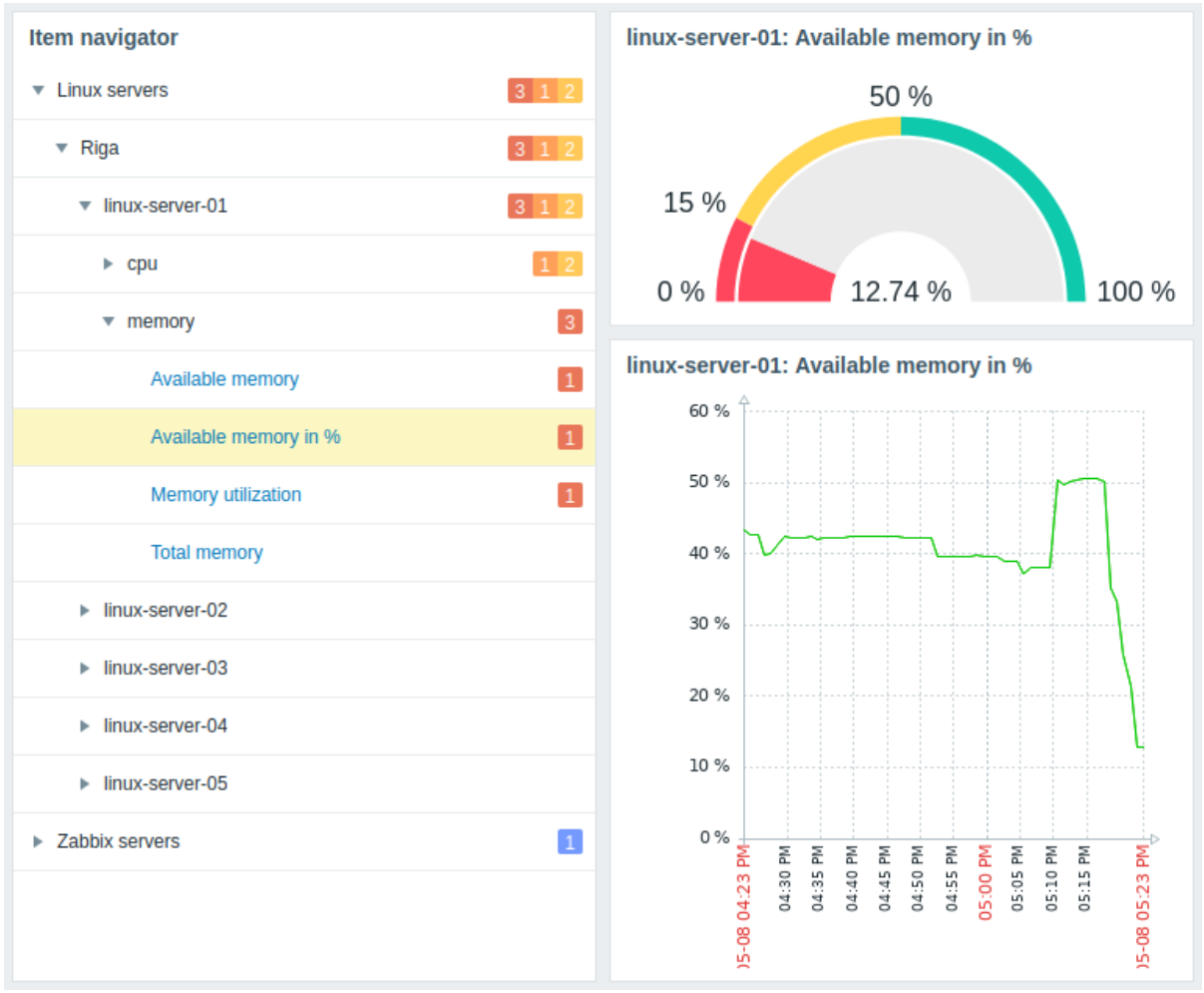
17 アイテムナビゲーター

概要

アイテムナビゲーターウィジェットは、さまざまなフィルタリングおよびグループ化オプションに基づいてアイテムを表示します。



このウィジェットでは、選択したアイテムに基づいて、他のウィジェットに表示される情報を制御することもできます。

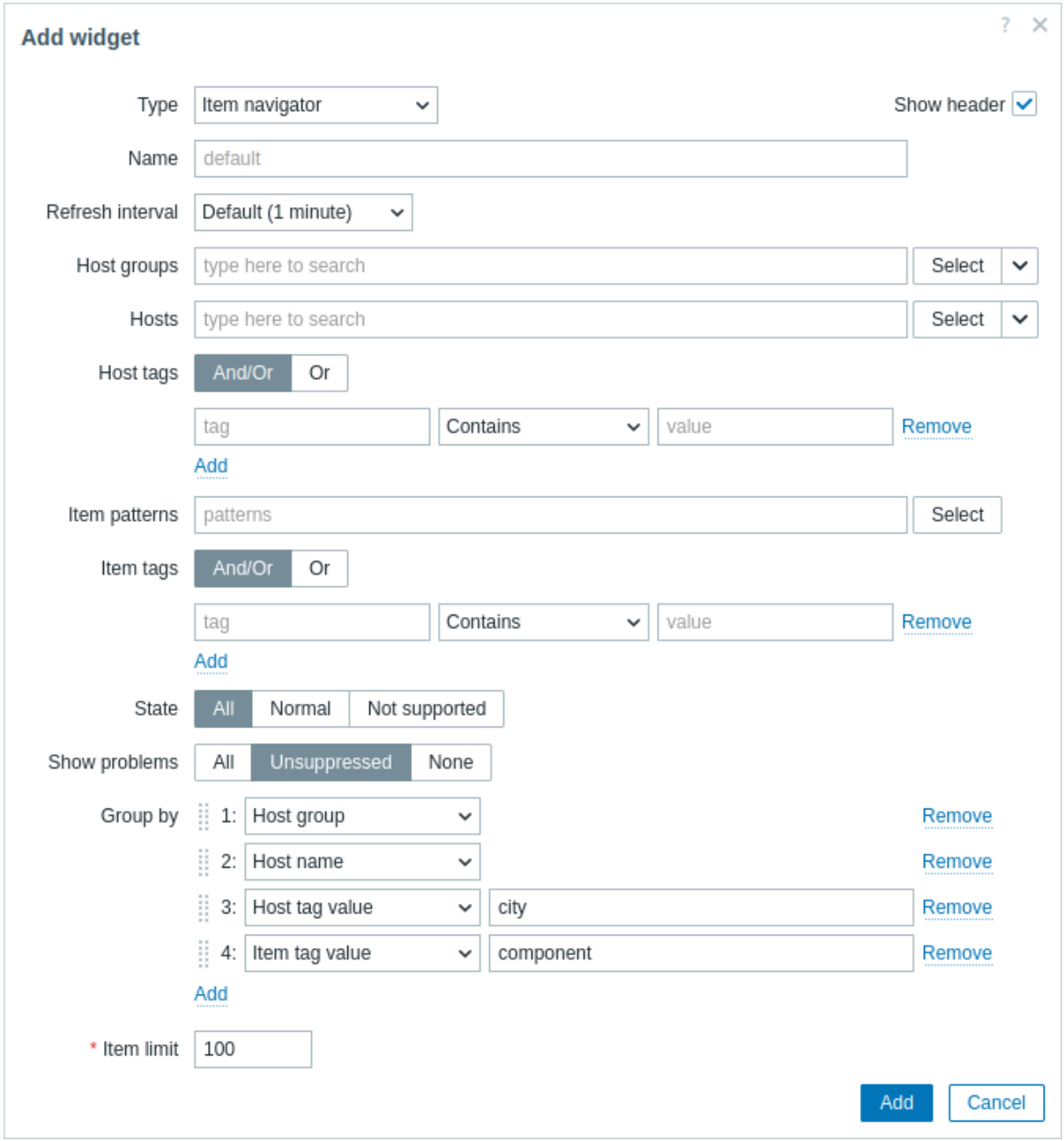


アイテムが整理されるグループは、展開または折りたたみできます。

グループおよび障害については、マウスオーバーヒントで追加の詳細を確認できます。

設定

設定するには、タイプとして アイテムナビゲーターを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを選択すると、ネストされたすべてのホストグループも暗黙的に選択されます。 ホストグループが選択されていない場合、ウィジェットにはすべてのホストグループ内のすべてのホストに属するアイテムが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストが選択されていない場合、ウィジェットにはすべてのホストに属するアイテムが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

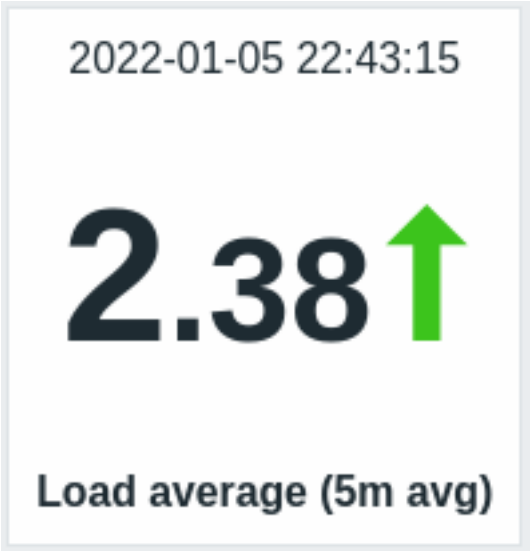
ホストタグ	ウィジェットに表示するアイテムを絞り込むためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。 タグ名の一致は常に大文字と小文字を区別します。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列を含む指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
アイテムパターン	アイテムパターンを入力するか、既存のアイテムをアイテムパターンとして選択します。指定したパターンに一致するアイテムがアイテムナビゲーターに表示されます。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致するアイテムを返し、Zabbix* は”Zabbix” で始まるアイテムを返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。入力を開始すると、選択した ホスト、または選択した ホストグループ内のホストに属するものに限定された一致アイテムのドロップダウンリストが表示されます。ワイルドカード記号は常に解釈されるため、たとえば *item** という名前のアイテムを、他に一致するアイテム（例: item2、item3）が存在する場合に個別に追加することはできません。 テンプレートダッシュボードでウィジェットを設定する場合、このパラメータではテンプレート上に設定されたアイテムのみ選択できます。
アイテムタグ	ウィジェットに表示するアイテムを絞り込むためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。 タグ名の一致は常に大文字と小文字を区別します。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列を含む指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列を含む指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
状態	状態（すべて、正常、未サポート）に基づいて、表示するアイテムを絞り込みます。
障害の表示	ステータス（すべて、抑制されていないもの、なし）に基づいて、アイテムとともに表示する障害を絞り込みます。

グループ化	アイテムをグループ化するための属性を追加します。 ホストグループ - アイテムを、それらのホストが属するホストグループごとにグループ化します。 ホスト名 - アイテムをそれらのホストごとにグループ化します。 ホストタグ値 - タグ名を入力して、そのホストタグの値ごとにアイテムをグループ化します（たとえば、”city” と入力すると、”Riga”、”Tokyo” などの値ごとにグループ化されます）。 アイテムタグ値 - タグ名を入力して、そのアイテムタグの値ごとにアイテムをグループ化します（たとえば、”component” と入力すると、”cpu”、”memory” などの値ごとにグループ化されます）。 グループ化属性は、グループ名の前にあるハンドルを上下にドラッグして並べ替えることができます。グループ化属性の順序によって、グループのネスト順序が決まることに注意してください。たとえば、複数のホストタグ名（1: color、2: city）を指定すると、アイテムはまず color（red、blue など）で、その後 city（Riga、Tokyo など）でグループ化されます。 設定したグループ化属性によっては、1 つのアイテムが複数のグループに表示される場合があります（たとえば、ホストグループでグループ化していて、そのアイテムのホストが複数のホストグループに属している場合）。そのようなアイテムをクリックすると、すべてのグループで選択および強調表示されます。 設定したグループ化属性に一致しないアイテムは、未分類グループに表示されます。 障害の表示が障害を表示するよう設定されている場合、障害は以下のように表示されます。 - 各グループについて、配下のすべてのアイテムの障害数が表示されます。 - 各アイテムについては、そのアイテム自身の障害数のみが表示されます。 グループ化属性は最大 10 個まで指定でき、すべて一意である必要があります。 グループ化属性が指定されていない場合、アイテムはグループ化されません。
アイテム数の上限	表示するアイテムの最大数を入力します。指定可能な値は 1～9999 です。 表示可能なアイテム数が設定した上限を超える場合、表示されたアイテムの下に対応するメッセージが表示されます（たとえば、”100 of 100+ items are shown”）。 設定したアイテム数の上限は、設定済みグループの表示にも影響することに注意してください。たとえば、アイテム数の上限を 100 に設定し、アイテムをホストごとにグループ化していて、各ホストに 200 個のアイテムがある場合、ウィジェットには最初のホストとその 100 個のアイテムのみが表示されます。 このパラメータは、*Administration → General →* GUI *の Limit for search and filter results* パラメータの影響を受けません。

18 アイテムの値

概要

このウィジェットは、単一のアイテムの値を目立つ形で表示するのに便利です。
これは最新の値だけでなく、過去の一定期間における集計値を表示することもできます。



値そのものに加えて、必要に応じて次の要素も表示できます。

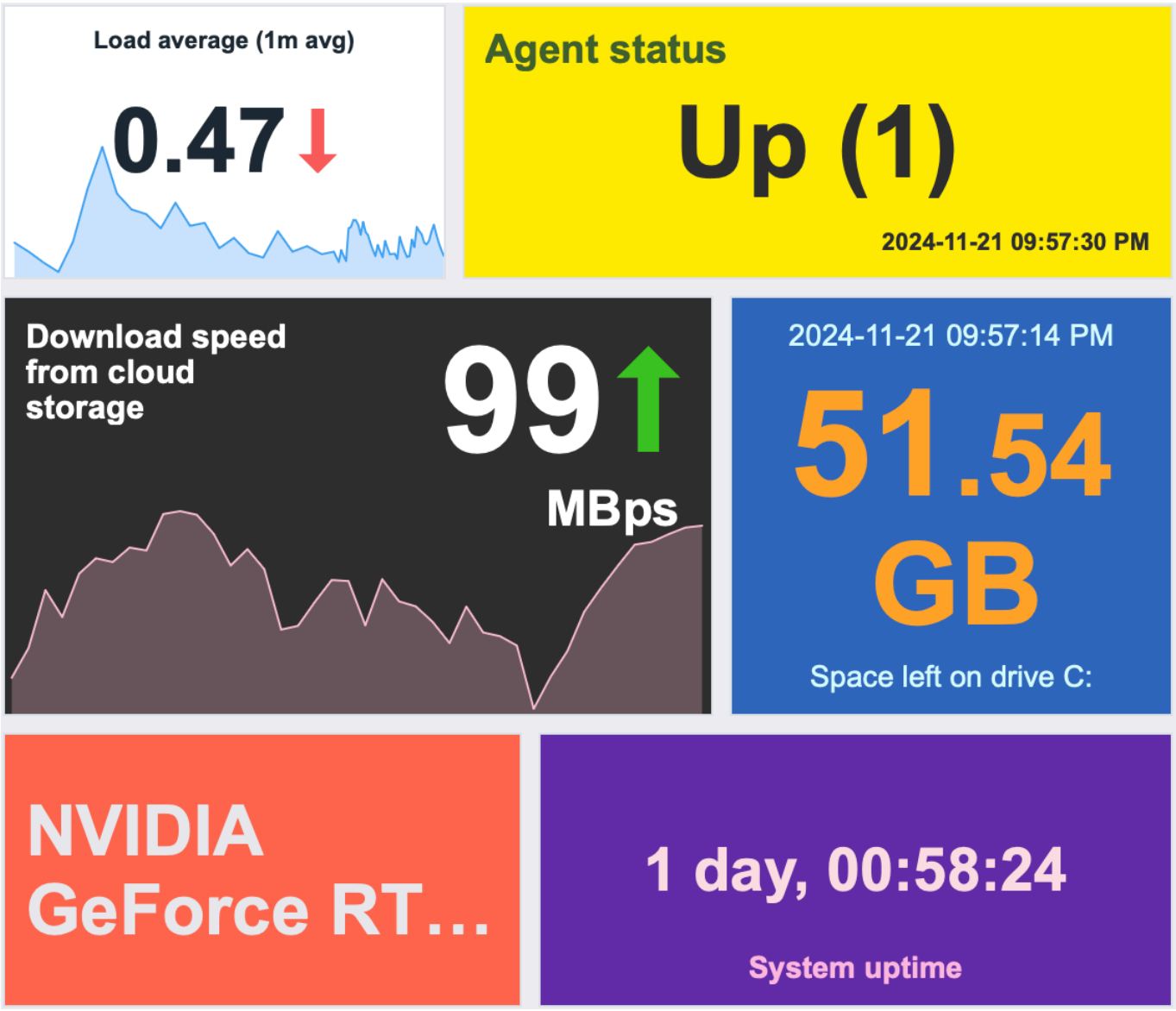
- メトリクスの時刻
- アイテムの説明
- 値の変化インジケーター
- 値の背景色
- アイテムの単位
- 指定した期間の値を示すスパークラインチャート

このウィジェットでは、数値および文字列の値を表示できます。バイナリ値の表示はサポートされていません。
文字列の値は 1 行で表示され、必要に応じて切り詰められます。アイテムに値がない場合は、「No data」と表示されます。

変化インジケーターは、常に過去の同じ期間と比較します。したがって、たとえば最新の値は直前の値と比較され、最新の月はその前の月と比較されます。集計における前の期間は、元の期間と同じ長さの期間で、元の期間の開始時刻の直前を終了時刻とする期間として計算されることに注意してください。

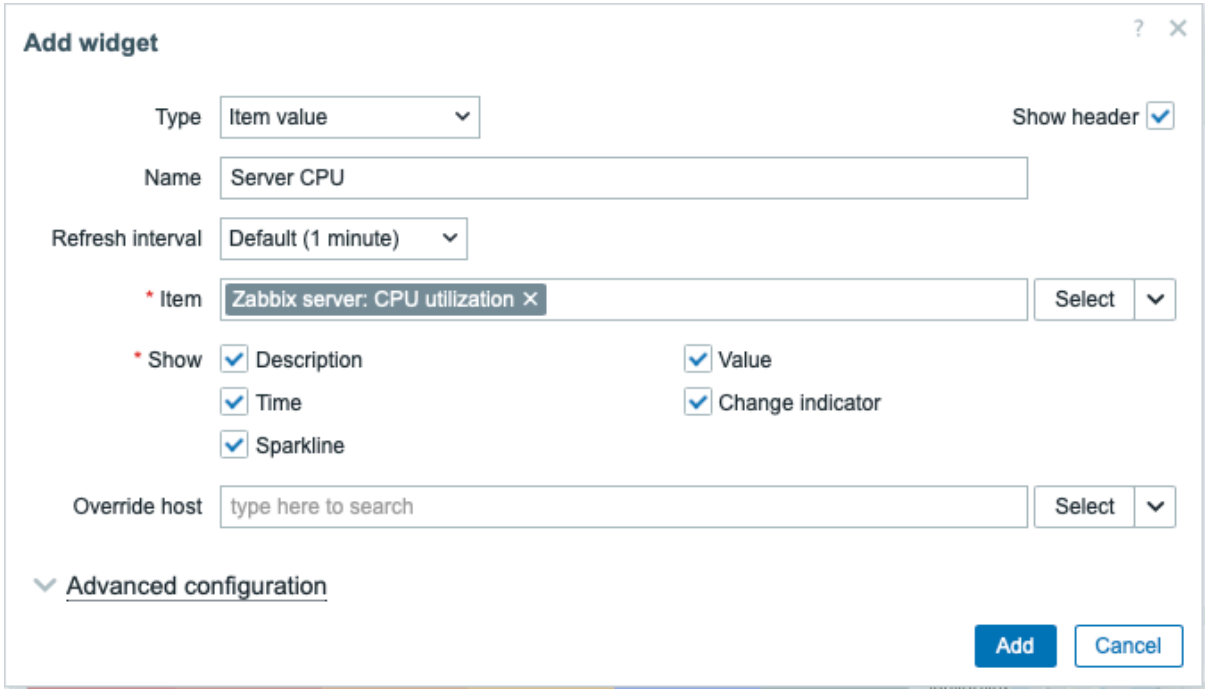
値をクリックすると、数値アイテムの場合はアドホックグラフ、文字列アイテムの場合は最新データが表示されます。

このウィジェットおよびその中のすべての要素は、高度な設定オプションを使用して視覚的に細かく調整でき、多様な表示スタイルを作成できます。



設定

設定するには、ウィジェットタイプとして アイテムの値を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

アイテム	アイテムを選択します。 または、アイテムのデータソースとして互換性のあるウィジェットを選択します。
表示	チェックボックスをオンにすると、対応する要素を表示します: 説明、値、時刻、変化インジケーター、スパークライン。オフにすると非表示になります。少なくとも 1 つの要素を選択する必要があります。
ホストの上書き	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
高度な設定	高度な設定ラベルをクリックすると、高度な設定オプションが表示されます。

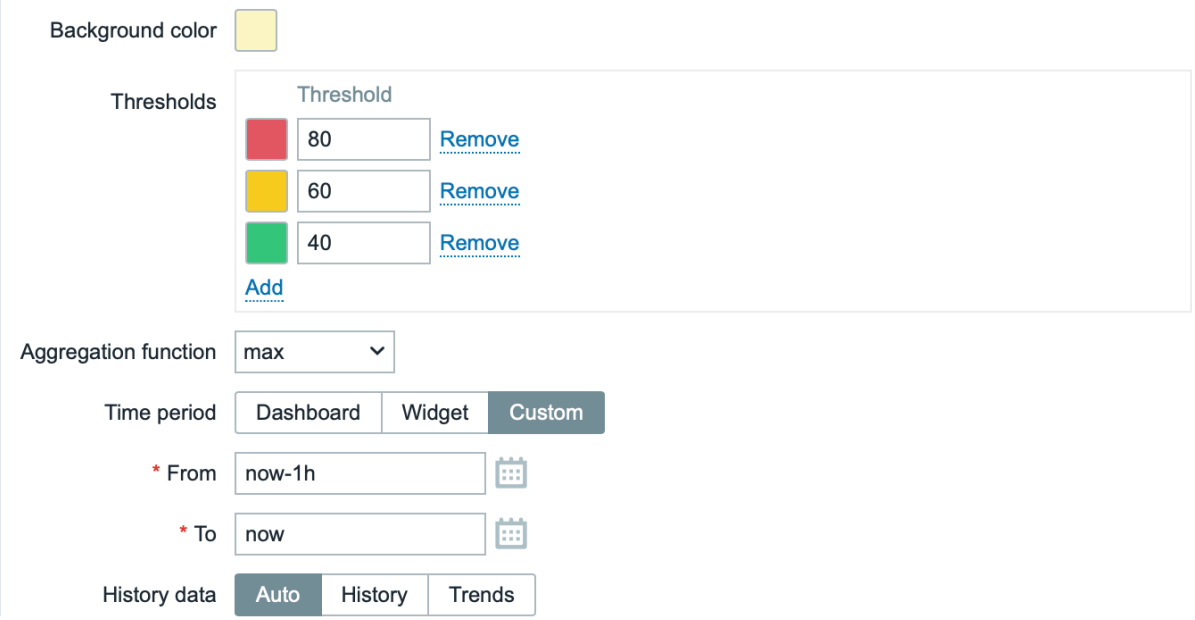
詳細設定

折りたたみ可能な詳細設定セクションでは詳細な設定オプションを利用できます。これらは、表示フィールドで選択された要素に対してのみ使用できます（上記参照）。

複数の要素を同じスペースに配置することはできません。同じスペースに配置した場合は、エラーメッセージが表示されます。

汎用ウィジェットパラメータ

これらのパラメータでは、ウィジェット全体の背景色（静的または動的）と、値の表示に使用する集計関数を設定します。

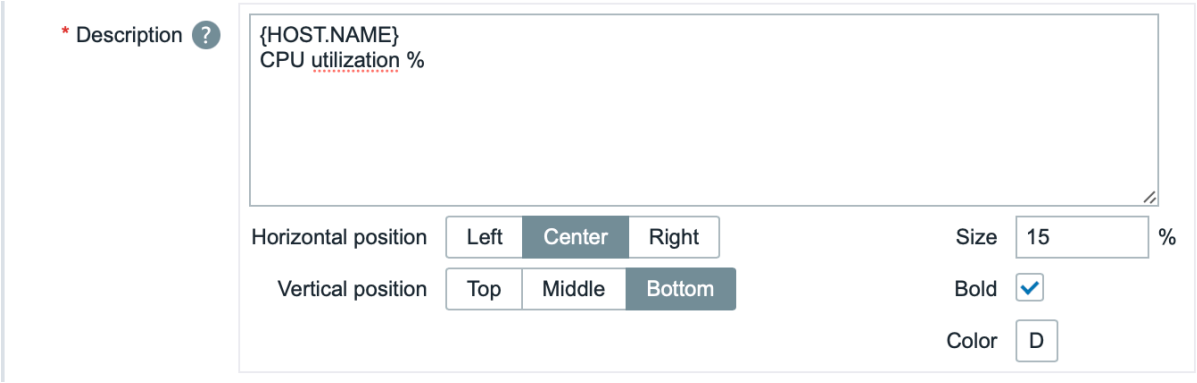


背景色	カラーピッカーから、ウィジェット全体の背景色を選択します。 D はデフォルト色を表します（Web インターフェースのテーマに依存します）。デフォルト値に戻すには、カラーピッカー内の *Use default* ボタンをクリックします。
しきい値	ウィジェット全体の動的な背景色を設定します。*Add* をクリックしてしきい値を追加し、カラーピッカーから背景色を選択して、数値を指定します。アイテムの値がしきい値以上になると、背景色が変わります。 リストは保存時に昇順でソートされます。 動的な背景色が正しく表示されるのは、数値アイテムの場合のみであることに注意してください。
集計関数	使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **count** - 値の件数を表示します。 **sum** - 値の合計を表示します。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します。 **not used** - 最新の値を表示します（集計なし）。 集計を使用すると、最新の値の代わりに、選択した期間（5 分、1 時間、1 日）の集計値を表示できます。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では、非数値データは数値に変換されます。

期間	値の集計に使用する期間を指定します。 **Dashboard** - ダッシュボードの期間を使用します。 **Widget** - 指定したウィジェットの期間を使用します。 **Custom** - カスタム期間を使用します。 <*Aggregation function* が”not used” に設定されている場合、このパラメータは表示されません。
ウィジェット	ウィジェットを選択します。 *Time period* が”Widget” に設定されている場合にのみ、このパラメータが表示されます。
開始	期間の開始時点を選択します（デフォルト値は `now-1h`）。相対時間の構文を参照してください。 *Time period* が”Custom” に設定されている場合にのみ、このパラメータが表示されます。
終了	期間の終了時点を選択します（デフォルト値は `now`）。相対時間の構文を参照してください。 *Time period* が”Custom” に設定されている場合にのみ、このパラメータが表示されます。
履歴データ	履歴またはトレンドからデータを取得します。 **Auto** - 自動選択。 **History** - 履歴データを取得します。 **Trends** - トレンドデータを取得します。 この設定は数値データにのみ適用されます。非数値データは常に履歴から取得されます。

説明

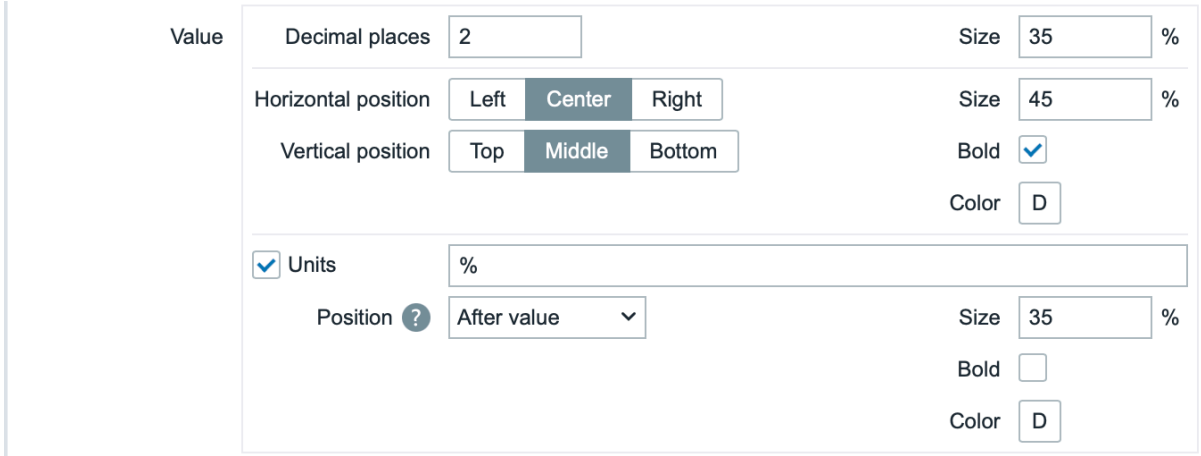
これらのパラメータは、アイテムの説明をどのように表示するかを決定します。



説明	アイテムの説明を入力します。この説明は、デフォルトのアイテム名を上書きする場合があります。 複数行の説明に対応しています。テキストとサポートされているマクロを組み合わせて使用できます。 {HOST.*}、{ITEM.*}、{INVENTORY.*} およびユーザーマクロがサポートされています。
水平方向の位置	アイテムの説明の水平方向の位置を選択します。左、右、中央から選べます。
垂直方向の位置	アイテムの説明の垂直方向の位置を選択します。上、下、中央から選べます。
サイズ	アイテムの説明のフォントサイズの高さを入力します（ウィジェット全体の高さに対する割合）。
太字	チェックボックスをオンにすると、アイテムの説明が太字で表示されます。
色	カラーピッカーからアイテムの説明の色を選択します。 D はデフォルトの色を表します（Web インターフェースのテーマに依存します）。デフォルト値に戻すには、カラーピッカーの デフォルトを使用ボタンをクリックします。

値

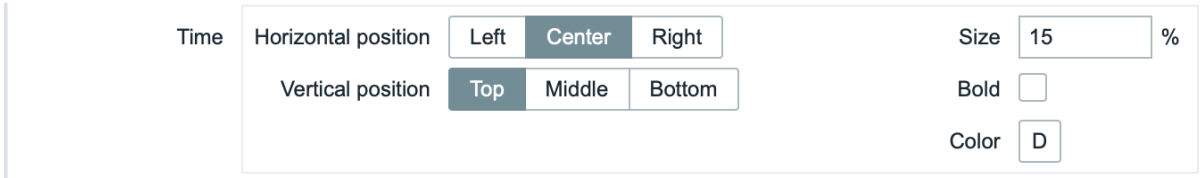
これらのパラメータは、アイテムの値をどのように表示するかを決定します。



小数点以下の桁数	値とともに表示する小数点以下の桁数を選択します。この値は float アイテムにのみ影響します。単位が「s」に設定されているアイテムでは、この値が 0 より大きい場合（デフォルトは 2）、ウィジェットは最上位の時間単位を丸めて数値として表示します（例: 「10.43m」）。値を完全な時間文字列に変換することはありません。0 に設定すると、秒は人が読みやすい時間文字列に変換されます（例: 「4h 56m 30s」）。
サイズ	小数点以下の桁数のフォントサイズの高さを入力します（ウィジェット全体の高さに対する割合）。
水平方向の位置	アイテムの値の水平方向の位置を、左、右、中央から選択します。
垂直方向の位置	アイテムの値の垂直方向の位置を、上、下、中央から選択します。
サイズ	アイテムの値のフォントサイズの高さを入力します（ウィジェット全体の高さに対する割合）。 アイテムの値のサイズが優先されるため、他の要素は値のためにスペースを譲る必要があることに注意してください。ただし、変化インジケーターについては、値が大きすぎる場合、変化インジケーターを表示するために切り詰められます。
太字	チェックボックスをオンにすると、アイテムの値を太字で表示します。
色	カラーピッカーからアイテムの値の色を選択します。 D はデフォルトの色を表します（Web インターフェースのテーマに依存します）。デフォルト値に戻すには、カラーピッカーの *Use default* ボタンをクリックします。
単位	
単位	チェックボックスをオンにすると、アイテムの値とともに単位を表示します。単位名を入力すると、アイテム設定の単位よりもそちらが優先されます。
位置	アイテムの単位の位置を、値の上、下、前、後から選択します。
サイズ	アイテムの単位のフォントサイズの高さを入力します（ウィジェット全体の高さに対する割合）。
太字	チェックボックスをオンにすると、アイテムの単位を太字で表示します。
色	カラーピッカーからアイテムの単位の色を選択します。 D はデフォルトの色を表します（Web インターフェースのテーマに依存します）。デフォルト値に戻すには、カラーピッカーの *Use default* ボタンをクリックします。

時刻

これらのパラメータは、時刻（アイテム履歴のクロック値）の表示方法を決定します。



水平方向の位置	時刻の水平方向の位置を選択します。左、右、中央から選択できます。
垂直方向の位置	時刻の垂直方向の位置を選択します。上、下、中央から選択できます。
サイズ	時刻のフォントサイズの高さを入力します（ウィジェット全体の高さに対する割合）。
太字	チェックボックスをオンにすると、時刻が太字で表示されます。
色	カラーピッカーから時刻の色を選択します。 D はデフォルトの色を表します（Web インターフェースのテーマによって異なります）。デフォルト値に戻すには、カラーピッカーの *Use default* ボタンをクリックします。

変化インジケーター

このセクションでは、カラーピッカーから変化インジケーターの色を選択できます。



変化インジケーターは次のとおりです。

- ↑ - アイテムの値が上昇している（数値アイテムの場合）
- ↓ - アイテムの値が下降している（数値アイテムの場合）
- ↕ - アイテムの値が変化している（文字列アイテムおよび値のマッピングがあるアイテムの場合）

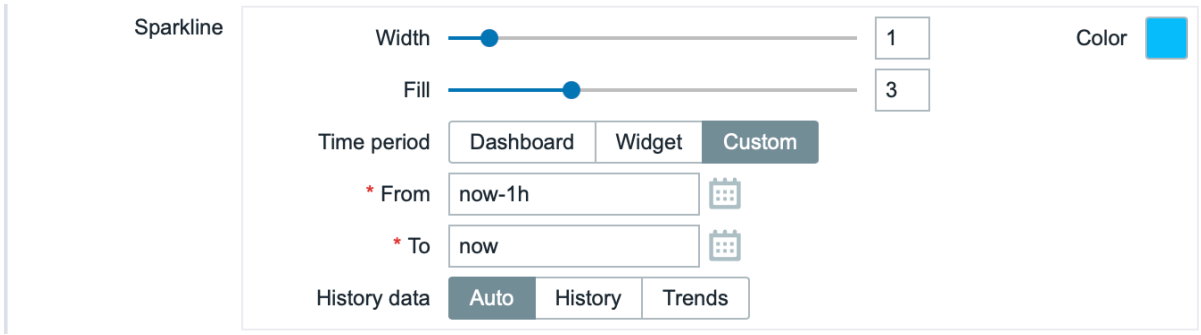
D はデフォルト色を表します（Web インターフェースのテーマに依存します）。デフォルト値に戻すには、カラーピッカー内の *Use default* ボタンをクリックしてください。

変化インジケーターの縦サイズは値のサイズに等しくなります（数値アイテムの場合は値の整数部）。

上向きおよび下向きのインジケーターは、値が 1 つだけの場合は表示されないことに注意してください。

スパークライン

これらのパラメータは、スパークラインチャートの表示方法を決定します。



幅	スライダーを使用するか、0～10 の範囲で値を手動入力して、グラフ線の太さを設定します。
色	線と塗りつぶしの色を選択します。
塗りつぶし	スライダーを使用するか、0～10 の範囲で値を手動入力して、塗りつぶし色の透明度を設定します。
期間	スパークラインチャートに含める値の期間を指定します。 ダッシュボード - ダッシュボードの期間を使用します。 ウィジェット - 指定したウィジェットの期間を使用します。 カスタム - カスタム期間を使用します。
ウィジェット	ウィジェットを選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合にのみ表示されます。
開始	期間の開始時点を選択します（デフォルト値は `now-1h`）。相対時間の構文を参照してください。 このパラメータは、期間が「カスタム」に設定されている場合にのみ表示されます。
終了	期間の終了時点を選択します（デフォルト値は `now`）。相対時間の構文を参照してください。 このパラメータは、期間が「カスタム」に設定されている場合にのみ表示されます。
履歴データ	履歴またはトレンドからデータを取得します。 自動 - 自動選択。 履歴 - 履歴データを取得します。 トレンド - トレンドデータを取得します。

19 マップ

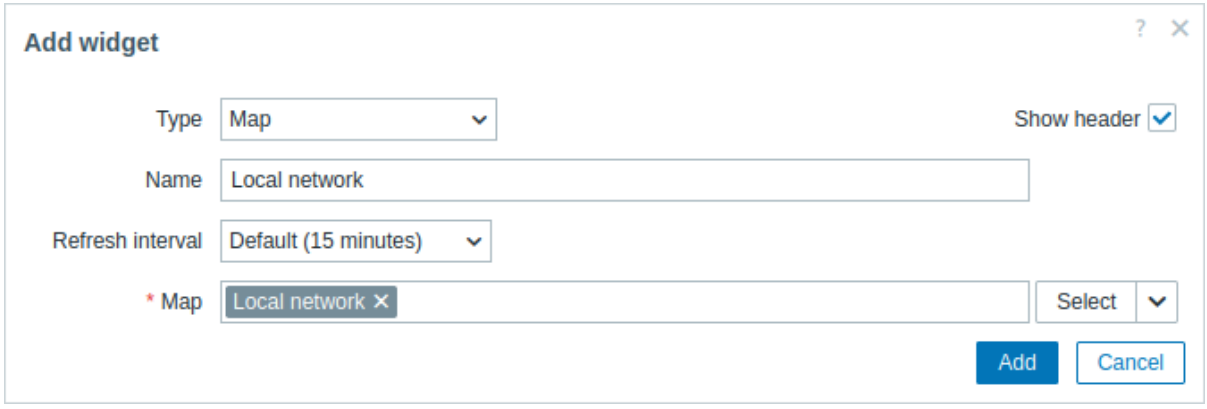
概要

マップウィジェットでは、次のいずれかを表示できます。

- 設定済みの単一のネットワークマップ
- マップナビゲーションツリー 内の設定済みネットワークマップの 1 つ（ツリー内のマップ名をクリックした場合）

設定

設定するには、タイプとして *Map* を選択します。



すべてのウィジェットに共通するパラメータに加えて、以下の固有のオプションを設定できます。

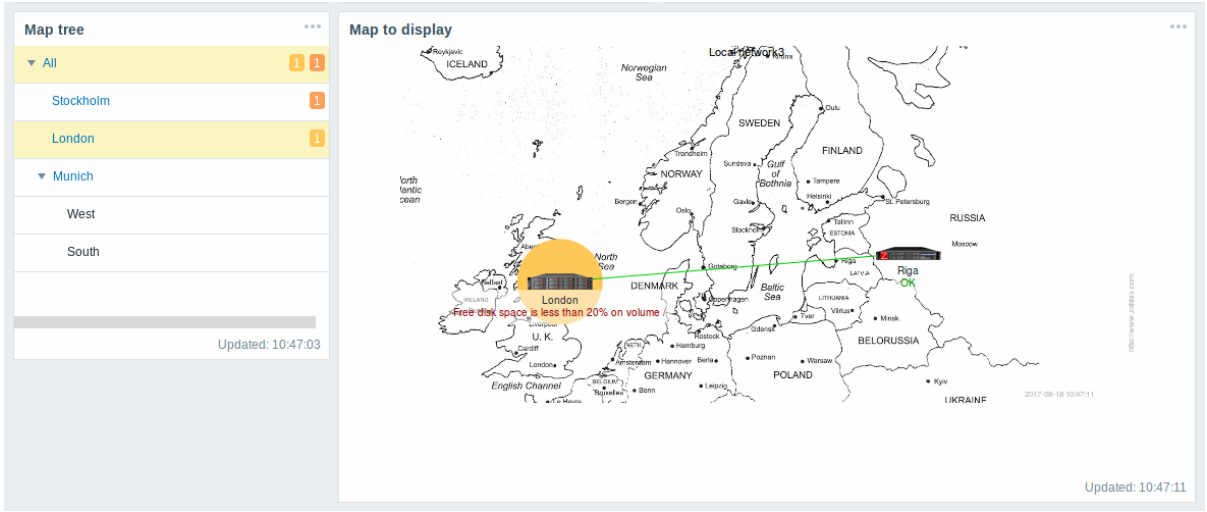
Map	表示するマップを設定します。 または、表示するマップのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、マップまたはウィジェットの名前を入力し始めると、一致するマップまたはウィジェットのドロップダウンが表示されます。

20 マップナビゲーションツリー

概要

このウィジェットでは、既存のマップの階層を構築でき、さらに含まれる各マップおよびマップグループの障害統計も表示できます。

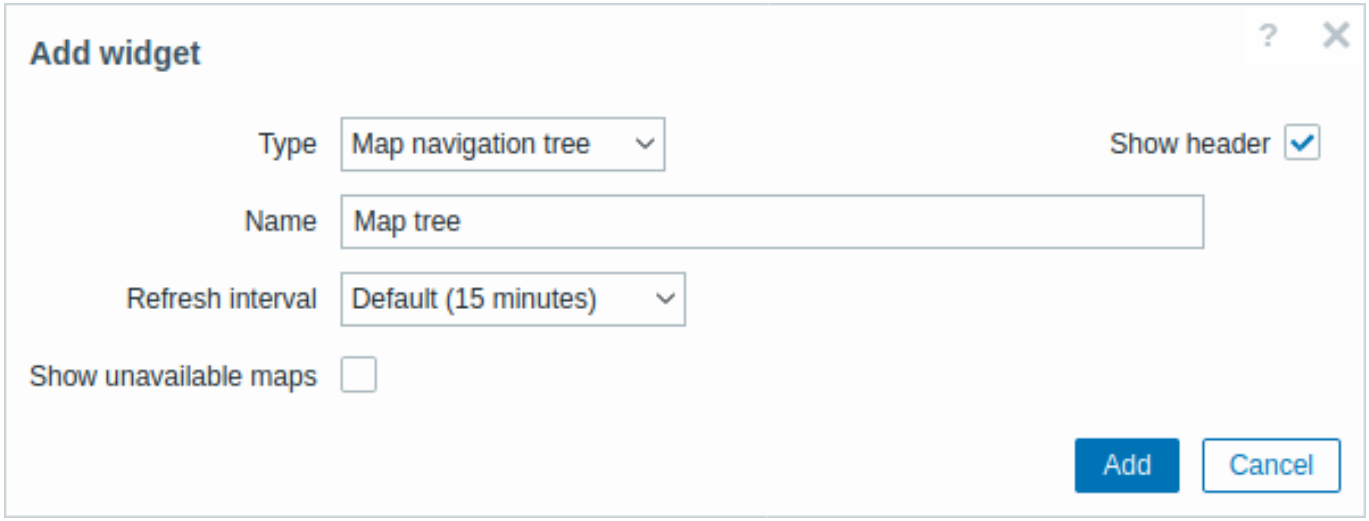
マップウィジェットをナビゲーションツリーにリンクすると、さらに便利になります。この場合、ナビゲーションツリー内のマップ名をクリックすると、そのマップがマップウィジェットに全体表示されます。



階層内の最上位マップの統計には、すべてのサブマップの障害数と、そのマップ自体の障害数の合計が表示されます。

設定

設定するには、タイプとして マップナビゲーションツリーを選択します。

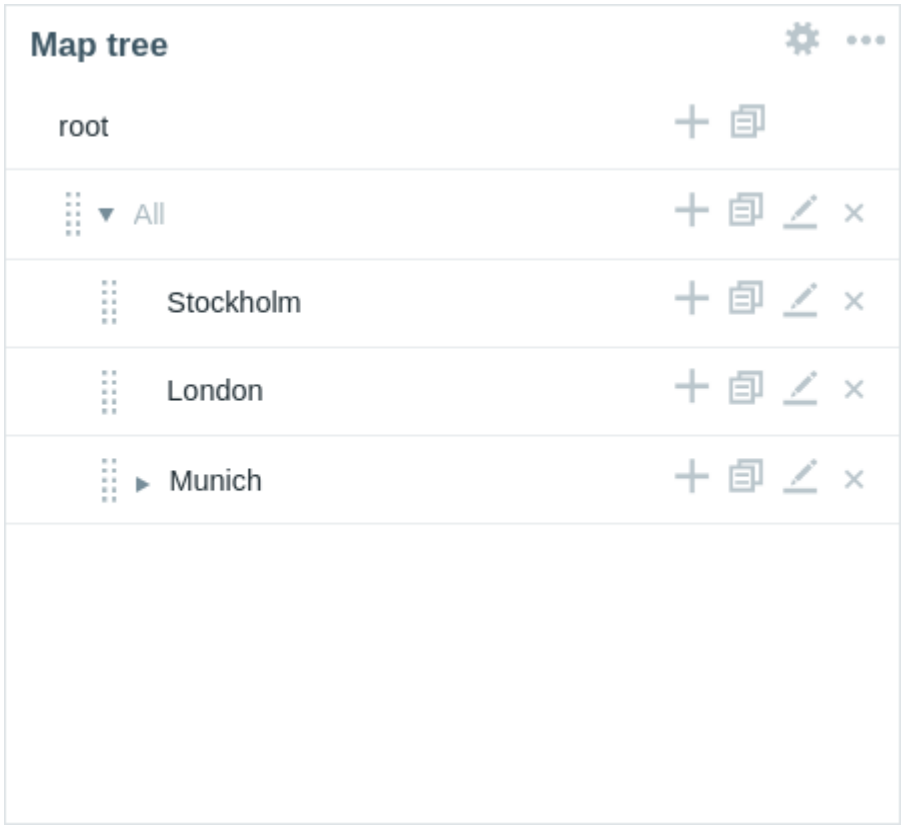


すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

利用不可のマップを表示	このチェックボックスをオンにすると、ユーザーに読み取り権限がないマップを表示します。 ナビゲーションツリー内の利用不可のマップは、グレーアウトしたアイコンで表示されます。 このチェックボックスがオンの場合、親レベルのマップが利用不可であっても、利用可能なsubmapが表示されることに注意してください。オフの場合、利用不可の親マップに属する利用可能な submap はまったく表示されません。 障害数は、利用可能なマップおよび利用可能なマップ要素に基づいて計算されます。

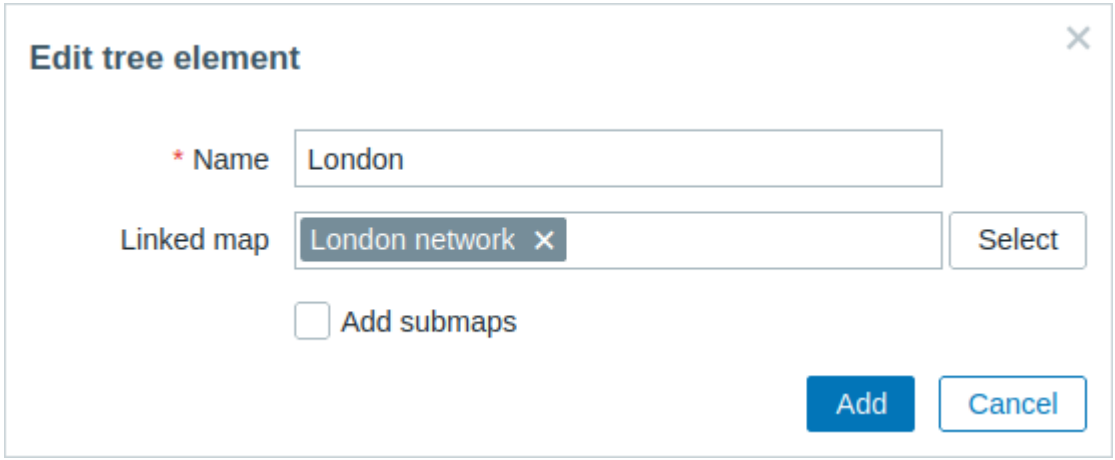
ナビゲーションツリーの要素はリストに表示されます。次のことができます。

- 要素（その子要素を含む）をリスト内の新しい位置へドラッグする。
- 要素を展開または折りたたんで、子要素を表示または非表示にする。
- 要素に子要素（リンクされたマップの有無を問わない）を追加する。
- 要素に複数の子要素（リンクされたマップ付き）を追加する。
- 要素を編集する。
- 要素（その子要素を含む）を削除する。



要素の設定

ナビゲーションツリー要素を設定するには、新しい要素を追加するか、既存の要素を編集します。



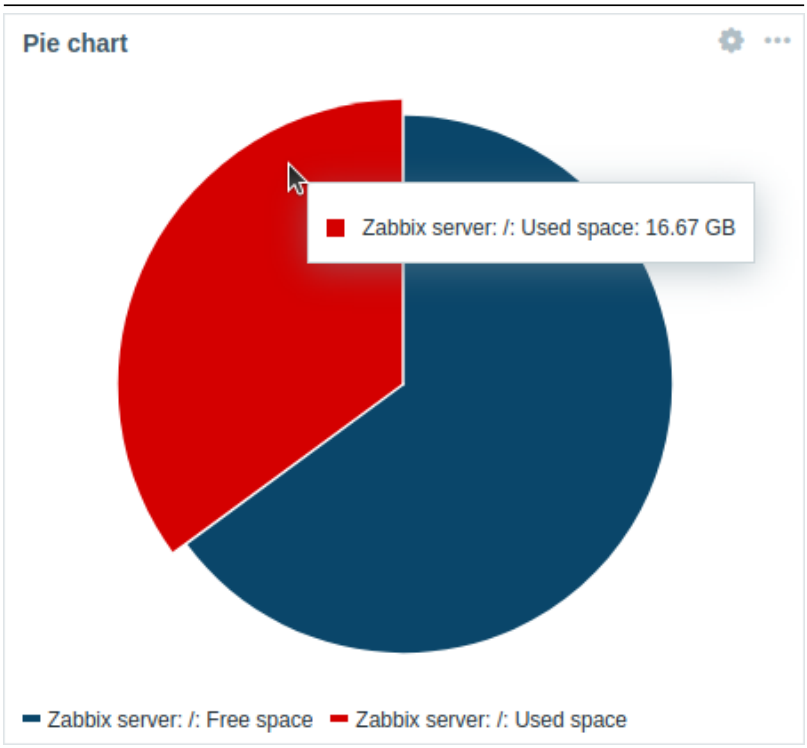
以下のナビゲーションツリー要素の設定パラメータを利用できます。

名前	ナビゲーションツリー要素の名前を入力します。
リンクされたマップ	ナビゲーションツリー要素にリンクするマップを選択します。 このフィールドはオートコンプリート対応のため、マップ名の入力を開始すると、一致するマップのドロップダウンが表示されます。
サブマップを追加	このチェックボックスをオンにすると、リンクされたマップのsubmapsがナビゲーションツリー要素の子要素として追加されます。

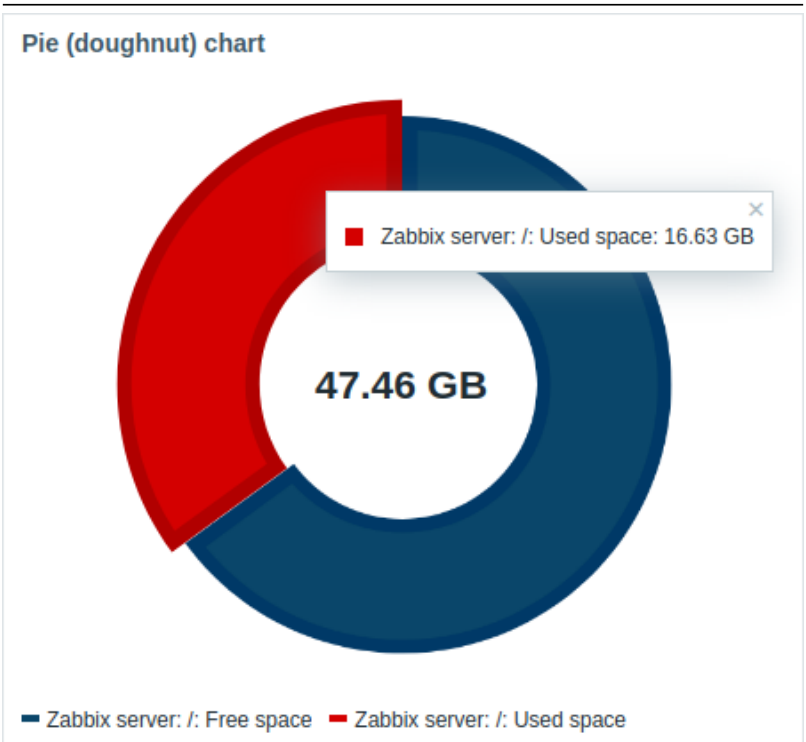
21 円グラフ

概要

円グラフウィジェットでは、選択したアイテムの値を円またはドーナツグラフとして表示できます。



円グラフ。



ドーナツグラフ。

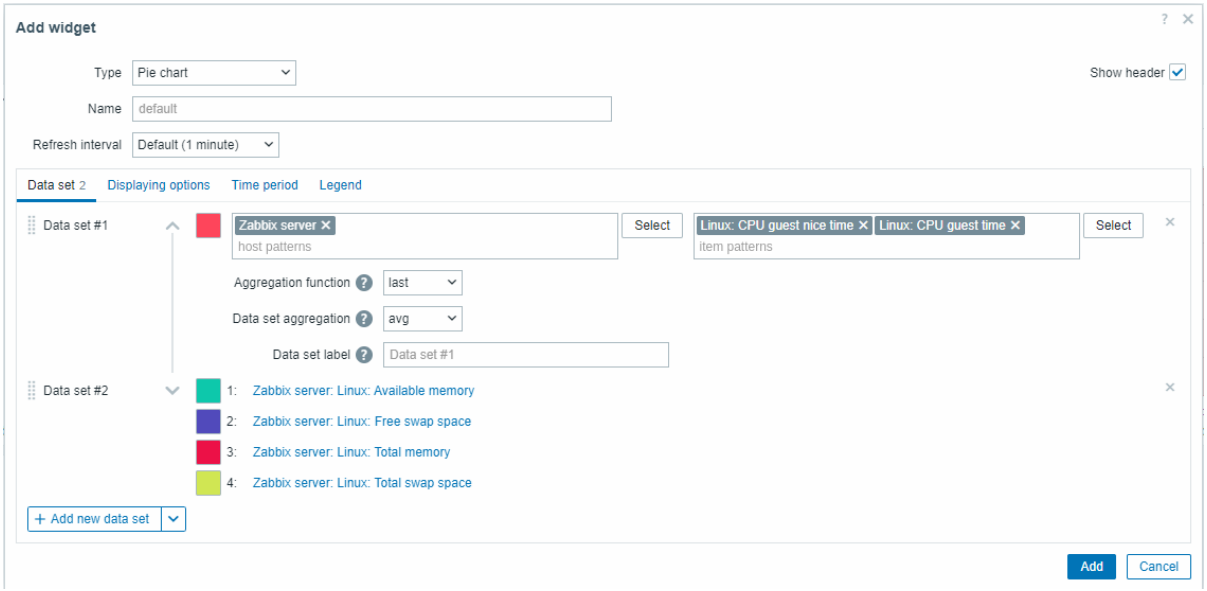
マウスオーバーすると、フォーカスされたセクターが外側に向かって拡大され、そのセクターの凡例が表示されます。フォーカスされたセクターをクリックすると、ポップアウト効果は「x」で閉じるまで維持されます。

これらのグラフは、ベクター画像描画技術を使用して描画されます。

円グラフウィジェットに表示される情報は、ウィジェットメニューで *Download image* オプションを選択することで、PNG 画像としてダウンロードできます。

設定

設定するには、タイプとして *Pie chart* を選択します。



データセット

データセットタブでは、データセットを追加して円グラフ用のデータを選択できます。追加できるデータセットは 2 種類あります。

- アイテムパターン - 一致するアイテムのデータが表示されます。単一のベースカラーを選択するか、パレットの行を選択して、一致した各アイテムに異なる色を割り当てることができます。
- アイテムリスト - 選択したアイテムのデータが表示されます。各アイテムの色をピッカーから個別に選択できます。

デフォルトでは、アイテムパターンデータセットが追加されます。

データセット	アイテムパターンデータセットの場合: ホストパターンとアイテムパターンを選択または入力します。入力したパターンに一致するアイテムのデータが円グラフに表示されます。表示できるアイテムは最大 50 件です。 選択にはワイルドカードパターンを使用できます（たとえば、* は 0 文字以上に一致する結果を返します）。 ワイルドカードパターンを指定するには、文字列を手動で入力して *Enter* を押します。 ワイルドカード記号は常に解釈されるため、たとえば *item** という名前のアイテムがあり、他にも一致するアイテム（たとえば *item2*、*item3*）が存在する場合、*item** のみを個別に追加することはできません。 「アイテムパターン」データセットでは、ホストパターンとアイテムパターンの指定が必須です。 あわせて参照: データセット設定の詳細。 アイテムリストデータセットの場合: アイテムを追加ボタンをクリックして、円グラフ用のアイテムを選択します。 アイテム名の後にある タイプドロップダウンでは、各アイテムの表示タイプを選択できます。 標準 - アイテムの値が円グラフ上で比例して表されます（デフォルト）。 合計 - アイテムの値が円グラフ全体を占めます。「合計」アイテムは円グラフごとに 1 つだけ設定でき、円グラフの凡例の先頭に配置されます。アイテムを「合計」に設定すると、データセット集計パラメータ（以下参照）は無効になり、「使用しない」に設定されます。 また、ウィジェットを追加ボタンをクリックして、アイテムのデータソースとして互換性のあるウィジェットを選択することもできます。ウィジェットには、個別のアイテムと同じオプションが適用されます。 「アイテムリスト」データセットでは、アイテムまたはウィジェットの指定が必須です。 あわせて参照: データセット設定の詳細。 数値型のアイテムのみ使用できます。 テンプレートダッシュボードでウィジェットを設定する場合、ホストパターンを指定するパラメータは使用できず、アイテムリストを指定するパラメータではテンプレート上で設定されたアイテムのみ選択できます。
集計関数	データセット内の各アイテムに使用する集計関数を指定します。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の件数を表示します。 **first** - 最初の値を表示します。 **last** - 最後の値を表示します（デフォルト）。
データセット集計	集計を使用すると、期間タブで選択した、またはダッシュボード全体で使用される期間（5 分、1 時間、1 日）に対する集計値を表示できます。 データセット全体に使用する集計関数を指定します。 **not used** - 集計しません。アイテムは個別に表示されます（デフォルト）。 **min** - 最小値を表示します。 **max** - 最大値を表示します。 **avg** - 平均値を表示します。 **sum** - 値の合計を表示します。 **count** - 値の件数を表示します。
データセットラベル	集計を使用すると、期間タブで選択した、またはダッシュボード全体で使用される期間（5 分、1 時間、1 日）に対する集計値を表示できます。 データセットのカスタムラベルを指定します。 このラベルは、データセット設定および円グラフの凡例（集計されたデータセットの場合）に表示されます。 すべてのデータセットには、データセットラベルが指定されているものも含めて番号が付けられます。ラベルが指定されていない場合、データセットには番号に応じて自動的にラベルが付けられます（例: 「データセット #2」、「データセット #3」など）。データセットの番号は、データセットの並べ替えやドラッグ後に再計算されます。 長すぎるデータセットラベルは、表示領域に収まるように短縮されます（例: 「プロセス数...」）。

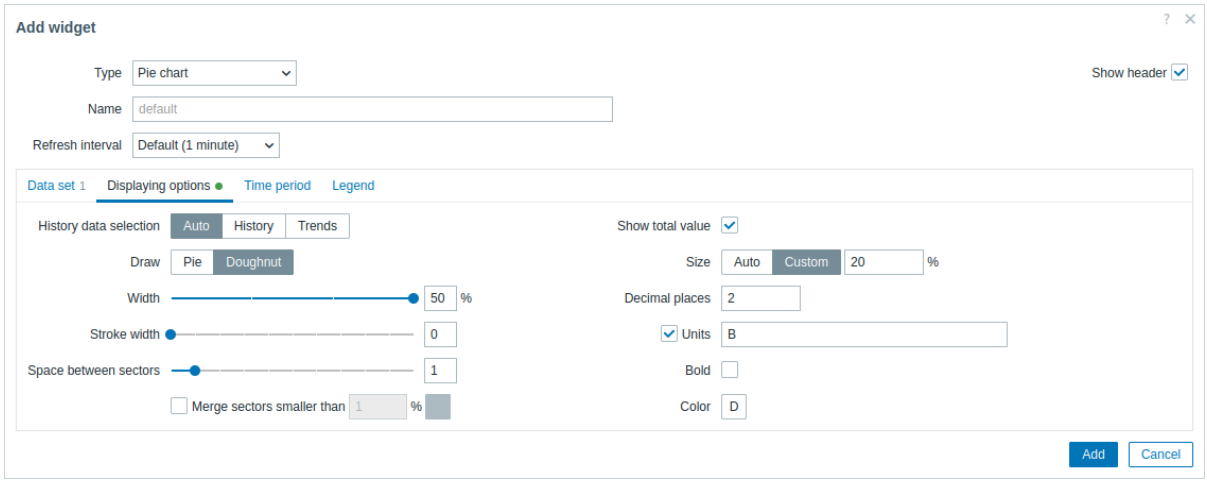
データセット設定の詳細

既存のデータセットは一覧に表示されます。これらのデータセットは、並べ替え、展開/折りたたみ、色の変更、複製が可能です。

詳細については、グラフウィジェットのデータセット設定の詳細を参照してください。これらの詳細は、円グラフウィジェットにも適用されます。

表示オプション

表示オプションタブでは、円グラフの履歴データの選択および可視化オプションを定義できます。

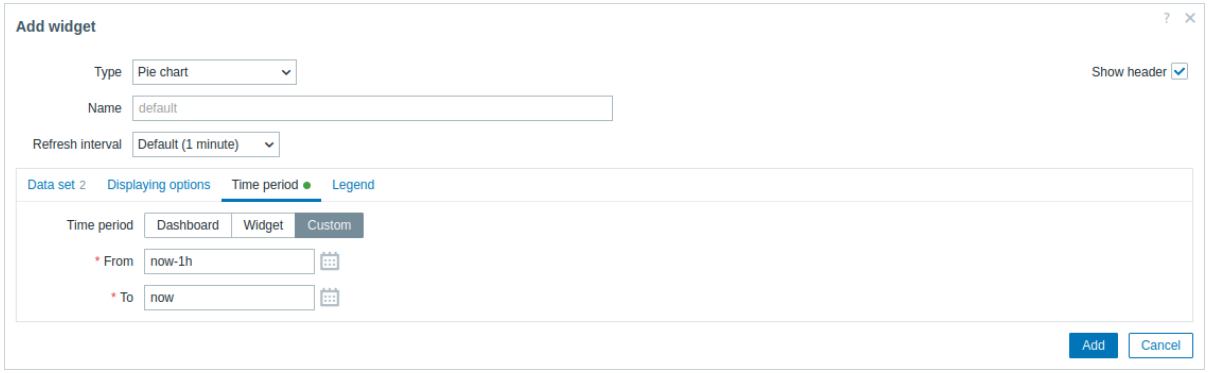


履歴データの選択	データソースを選択します: 自動 - 従来のアルゴリズムに従ってデータが取得されます（デフォルト）。 履歴 - 履歴のデータ。 トレンド - トレンドのデータ。
描画	円グラフの表示スタイルを選択します: 円グラフ - 完全な円グラフ（セクターは半径の 100% を使用）。 ドーナツ - 中央が空いた円グラフ（セクターは半径の最大 50% を使用）。
セクター間の間隔	セクター間の間隔サイズ（0～10 の単位）を選択します（デフォルトは「1」）。
N% 未満のセクターを結合	チェックボックスをオンにすると、N% 未満のセクターを結合します。 有効にした場合は、結合されたセクターの色と、小さいセクターを結合する割合のしきい値（N）を選択します。
描画スタイル: ドーナツ	

幅線の幅		ドーナツの幅を選択します: 半径の 20%、30%、40%、または 50%（デフォルト）。 ドーナツのセクター境界線の幅（0～10）を選択します。
合計値を表示		チェックボックスをオンにすると、ドーナツグラフの中央に合計値を表示します。
	サイズ	合計値のサイズオプションを選択します: 自動 - ドーナツ中央で読みやすく収まるようにテキストサイズが自動調整されます。 カスタム - ウィジェット全体の高さに対する割合でテキストサイズを指定します。
	小数点以下の桁数	合計値の小数点以下の桁数を指定します（0～6）。
	単位	合計値の単位を指定します。
	太字	チェックボックスをオンにすると、合計値を太字で表示します。
	色	合計値の色を選択します。

期間

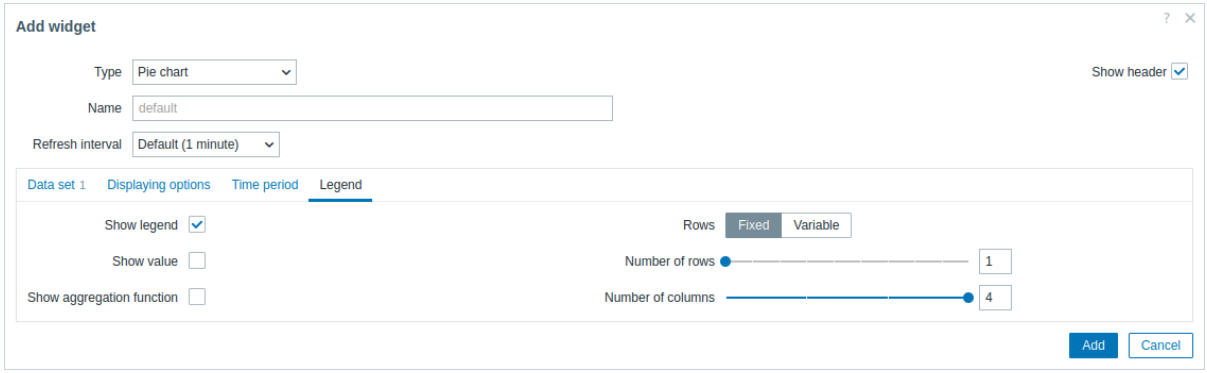
期間タブでは、円グラフの集計設定に対してカスタム期間を設定できます。



期間	期間のデータソースを選択します。 ダッシュボード - 期間セレクターをデータソースとして設定します。 ウィジェット - ウィジェットパラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 カスタム - 開始および終了パラメータで指定した期間をデータソースとして設定します。設定すると、ウィジェットの右上隅に時計アイコンが表示され、マウスオーバー時に設定された時刻が示されます。
ウィジェット	期間のデータソースとして、互換性のあるウィジェット（グラフ、グラフ（クラシック）、グラフプロトタイプ）を入力または選択します。 このパラメータは、期間が「ウィジェット」に設定されている場合に使用できます。
開始	期間の開始を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。
終了	期間の終了を入力または選択します。 相対時間構文（now、now/d、now/w-1w など）をサポートしています。 このパラメータは、期間が「カスタム」に設定されている場合に使用できます。

凡例

凡例タブでは、円グラフの凡例をカスタマイズできます。



凡例を表示	このチェックボックスをオフにすると、円グラフ上の凡例を非表示にします（デフォルトでオン）。
値を表示	このチェックボックスをオンにすると、凡例にアイテムの値を表示します。
集計関数を表示	このチェックボックスをオンにすると、凡例に集計関数を表示します。
行	凡例の行の表示モードを選択します： 固定 - 表示される行数は、行数パラメータの値によって決まります。 可変 - 表示される行数は、設定されたアイテム数によって決まりますが、最大行数パラメータの値を超えることはありません。
行数/ 最大行数	行が「固定」に設定されている場合、表示する凡例の行数を設定します（1～10）。 行が「可変」に設定されている場合、表示する凡例の最大行数を設定します（1～10）。
列数	表示する凡例の列数を設定します（1～4）。 このパラメータは、値を表示がオフの場合に使用できます。

22 障害ホスト

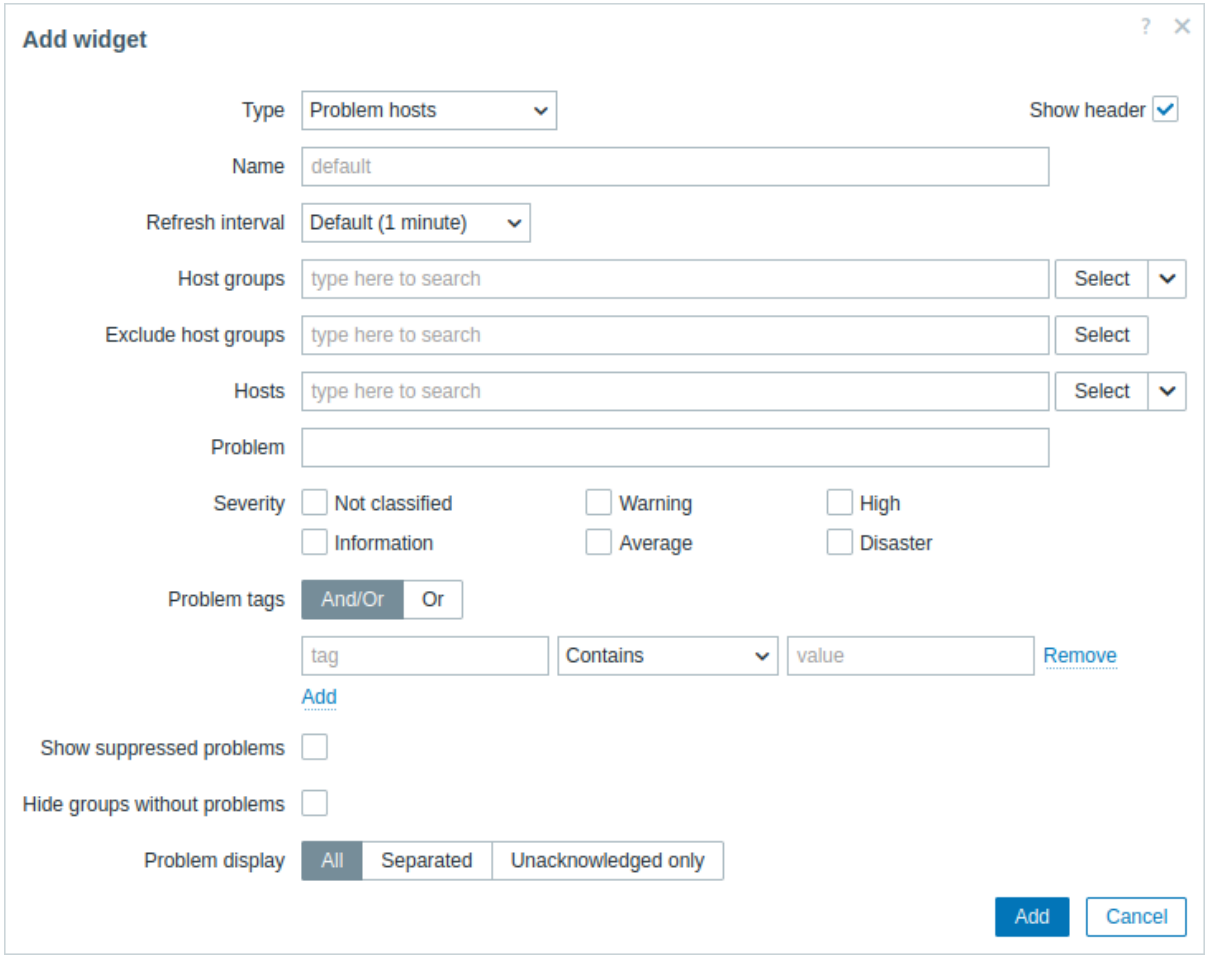
概要

障害ホストウィジェットでは、ホストグループごとの障害数と、グループ内で最も高い障害の深刻度を表示できます。

障害数は、原因障害についてのみ表示されます。

設定

設定するには、タイプとして 障害ホストを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
除外するホストグループ	ウィジェットで非表示にするホストグループを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループのホストデータはウィジェットに表示されません。たとえば、ホスト 001、002、003 がグループ A に属し、ホスト 002、003 がグループ B にも属しているとします。このとき、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害	障害名によって表示する障害ホストの数を制限できます。 ここに文字列を入力すると、名前にその文字列を含む障害を持つホストのみが表示されます。 マクロは展開されません。

深刻度	障害の深刻度を選択して、ウィジェットに表示する障害を絞り込みます。 深刻度が何も選択されていない場合は、すべての障害が表示されます。
障害タグ	ウィジェットに表示する障害数を制限するために、障害タグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
抑制された障害を表示	ホストのメンテナンスにより通常は抑制される（表示されない）障害を表示するには、このチェックボックスをオンにします。
障害のないグループを非表示	障害のないグループを非表示オプションを有効にすると、障害のないホストグループのデータをウィジェットで非表示にします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害表示	障害数の表示方法を指定します。 すべて - 障害の総数を表示します。 分離表示 - 未確認の障害数を、障害総数とは分けて表示します。 未確認のみ - 未確認の障害数のみを表示します。

23 件の障害

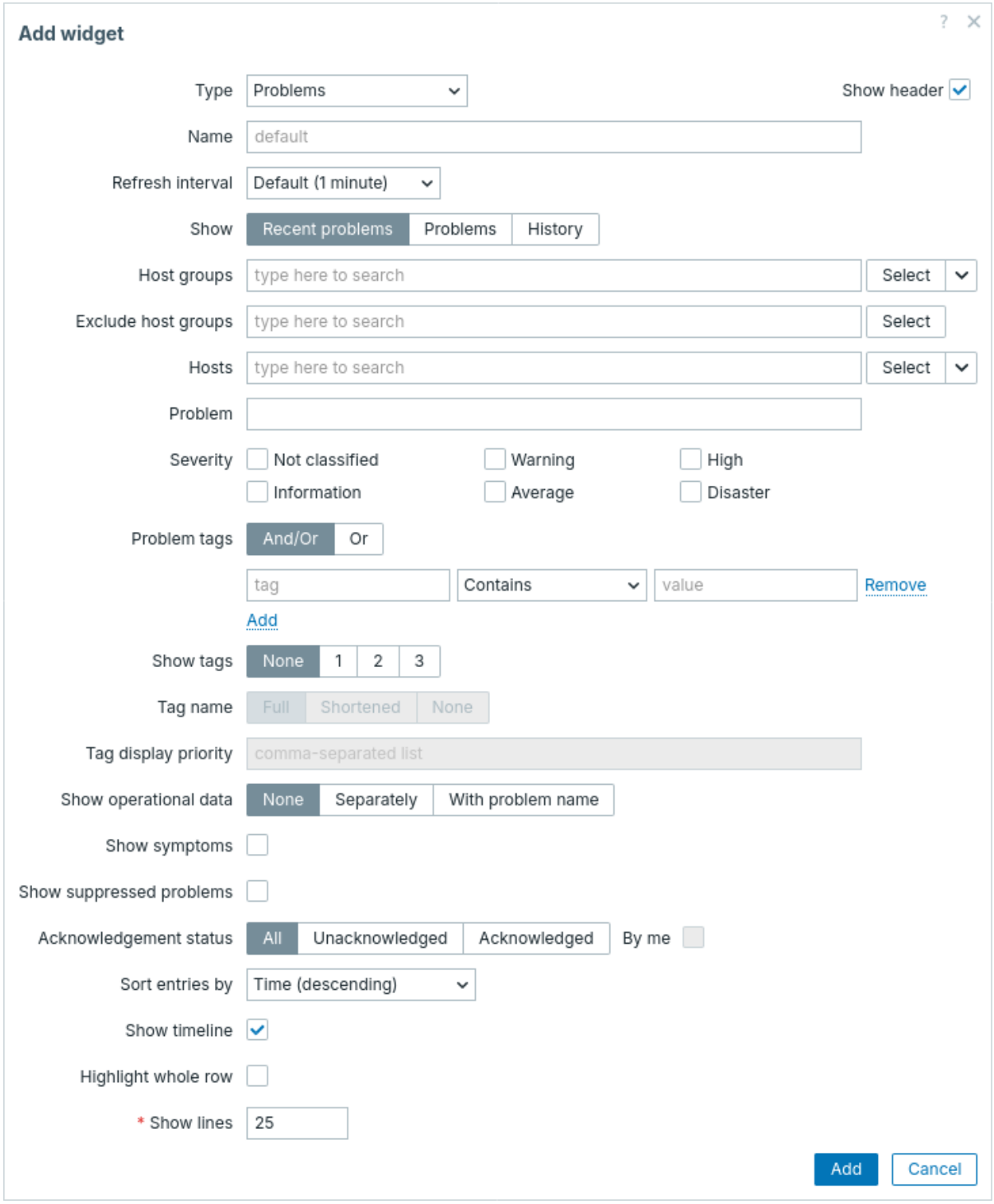
概要

このウィジェットでは、現在の障害を表示できます。このウィジェットの情報は、*Monitoring* > *Problems* と同様です。

最大 1000 件のレコードを表示できます。

設定

設定するには、タイプとして *Problems* を選択します。



ウィジェットに表示する障害の数は、障害ステータス、障害名、深刻度、ホストグループ、ホスト、イベントタグ、確認ステータスなど、さまざまな方法で制限できます。

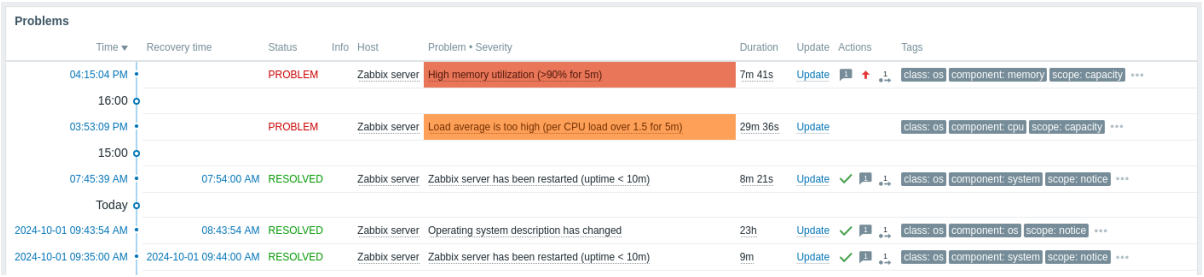
すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

Show	障害ステータスでフィルタリングします： **Recent problems** - 未解決および最近解決された障害を表示します（デフォルト）。 **Problems** - 未解決の障害を表示します。 **History** - すべてのイベントの履歴を表示します。

Host groups	ウィジェットに表示する障害のホストグループを選択します。 または、ホストグループのdata sourceとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。これらのホストグループの障害がウィジェットに表示されます。ホストグループが入力されていない場合は、すべてのホストグループの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Exclude host groups	ウィジェットから障害を非表示にするホストグループを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。これらのホストグループの障害はウィジェットに表示されません。たとえば、ホスト 001、002、003 がグループ A に属し、ホスト 002、003 がグループ B にも属している場合があります。このとき、同時にグループ A を *show* し、グループ B を *exclude* すると、ホスト 001 の障害のみがウィジェットに表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Hosts	ウィジェットに表示する障害のホストを選択します。 または、ホストのdata sourceとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストが入力されていない場合は、すべてのホストの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Problem	障害名によって表示する障害数を制限できます。 ここに文字列を入力すると、入力した文字列を名前に含む障害のみが表示されます。 マクロは展開されません。
Severity	ウィジェットに表示する障害をフィルタリングするために、障害の深刻度を選択します。 深刻度が選択されていない場合は、すべての障害が表示されます。
Problem tags	ウィジェットに表示する障害数を制限するために障害タグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件で使用できる演算子は次のとおりです。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 フィルタリング時には、ここで指定したタグが障害とともに最初に表示されます。ただし、*Tag display priority*（下記参照）リストで上書きされる場合を除きます。
Show tags	表示するタグ数を選択します。 **None** - *Tags* 列を表示しません。 **1** - *Tags* 列に 1 つのタグを表示します。 **2** - *Tags* 列に 2 つのタグを表示します。 **3** - *Tags* 列に 3 つのタグを表示します。 障害のすべてのタグを確認するには、三点アイコンにマウスカーソルを合わせてください。

Tag name	タグ名の表示モードを選択します。 **Full** - タグ名と値を完全な形で表示します。 **Shortened** - タグ名は 3 文字に短縮されますが、タグ値は完全な形で表示されます。 **None** - タグ値のみを表示し、名前は表示しません。
Tag display priority	障害のタグ表示優先順位を、タグのカンマ区切りリストとして入力します。 使用するのはタグ名のみで、値は指定しません。 例：`Services,Applications,Application` このリストのタグは、アルファベット順という自然な並び順より優先して、常に先に表示されます。
Show operational data	運用データの表示モードを選択します。 **None** - 運用データを表示しません。 **Separately** - 運用データを別の列に表示します。 **With problem name** - 運用データを障害名に追加し、運用データは括弧で囲んで表示します。
Show symptoms	症状として分類された障害を独立した行に表示するには、このチェックボックスをオンにします。
Show suppressed problems	ホストメンテナンスまたは単一のproblem suppressionによって本来は抑制されて表示されない障害を表示するには、このチェックボックスをオンにします。
Acknowledgement status	すべての障害、未確認の障害のみ、または確認済みの障害のみを表示するようにフィルタリングします。追加のチェックボックスをオンにすると、自分がこれまでに確認したことのある障害を除外できます。
Sort entries by	エントリの並び替え基準を選択します。 **Time**（降順または昇順） **Severity**（降順または昇順） **Problem name**（降順または昇順） **Host**（降順または昇順） **Host**（降順または昇順）による並び替えは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
Show timeline	視覚的なタイムラインを表示するには、このチェックボックスをオンにします。
Highlight whole row	未解決の障害について行全体を強調表示するには、このチェックボックスをオンにします。強調表示には障害の深刻度の色が使用されます。 *Highlight whole row* は高コントラストテーマでは使用できません。 このオプションは Zabbix 7.4.3 以降でサポートされています。
Show lines	表示する障害行数を指定します。

ウィジェットの使用

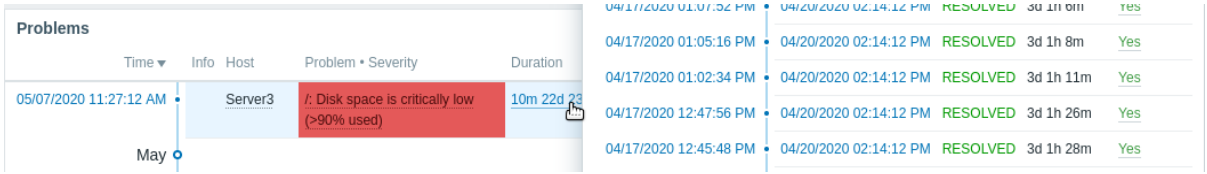


障害ウィジェットでは、追加情報にすばやくアクセスできます。

- 問題の発生日（*Time* 列）または復旧日時（*Recovery Time* 列）をクリックすると、イベントの詳細を表示できます。
- *Info* フィールドが空でない場合は、表示されているアイコンにマウスカーソルを合わせると追加の詳細を確認できます。
- ホスト名をクリックすると、ホストメニューが開きます。
- 問題名をクリックすると、イベントメニューが開きます。
- 問題の継続時間にマウスカーソルを合わせるかクリックすると、障害イベントポップアップを表示できます。
- *Update* を押すと、障害の更新ウィンドウが開きます。
- *Actions* 列の灰色の矢印アイコンにマウスカーソルを合わせるか押すと、実行されたアクションの一覧を表示できます。

障害イベントポップアップ

障害イベントポップアップには、このトリガーの障害イベントの一覧と、定義されている場合はトリガーの説明およびクリック可能な URL が含まれます。



障害イベントポップアップを表示するには、次のいずれかを行います。

- *Problems* ウィジェットの *Duration* 列にある問題の継続時間にマウスカーソルを合わせます。マウスカーソルを継続時間から外すと、ポップアップは消えます。
- *Problems* ウィジェットの *Duration* 列にある継続時間をクリックします。ポップアップは、継続時間をもう一度クリックした場合にのみ消えます。

24 深刻度別の障害

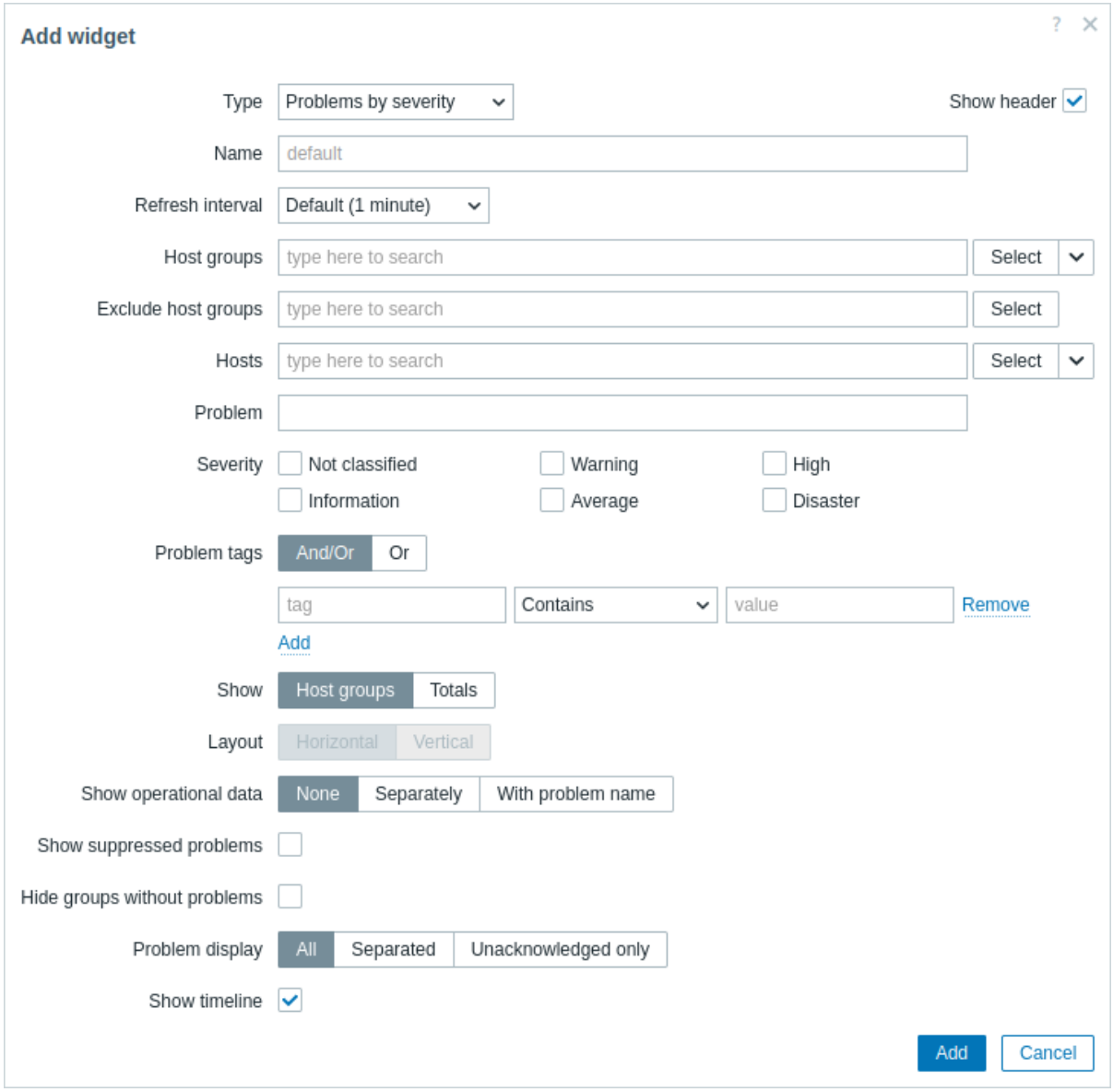
概要

このウィジェットでは、深刻度ごとの障害数を表示できます。ウィジェットに表示するホストとトリガーを制限し、障害数の表示方法を定義できます。

障害数は、原因障害についてのみ表示されます。

設定

設定するには、タイプとして 深刻度ごとの障害を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名を入力し始めると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
除外するホストグループ	ウィジェットに表示しないホストグループを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名を入力し始めると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループのホストデータはウィジェットに表示されません。たとえば、ホスト 001、002、003 がグループ A に属し、ホスト 002、003 がグループ B にも属しているとします。このとき、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名を入力し始めると、一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害	障害名によって表示される障害ホストの数を制限できます。 ここに文字列を入力すると、名前にその文字列を含む障害を持つホストのみが表示されます。 マクロは展開されません。
深刻度	ウィジェットに表示する障害を絞り込むため、障害の深刻度を選択します。 深刻度が 1 つも選択されていない場合は、すべての障害が表示されます。
障害タグ	ウィジェットに表示する障害数を制限するため、障害タグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 存在する - 指定したタグ名を含めます。 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別）。 含む - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 存在しない - 指定したタグ名を除外します。 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別）。 含まない - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
表示	表示オプションを選択します。 ホストグループ - ホストグループごとの障害を表示します。 合計 - 選択したすべてのホストグループの障害合計を、障害の深刻度に対応した色付きブロックで表示します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できず、障害合計のみが表示されます。
レイアウト	レイアウトオプションを選択します。 横 - 合計の色付きブロックを横方向に表示します。 縦 - 合計の色付きブロックを縦方向に表示します。 このパラメータは、表示が「合計」に設定されている場合に使用できます。

運用データを表示	運用データ（監視 > 障害の 運用データ の説明を参照）を次の形式で表示します。 なし - 運用データは表示されません 別々に - 運用データを別の行として表示します 障害名とともに - 運用データを障害名と一緒に表示します。
抑制された障害を表示	ホストのメンテナンスにより抑制（非表示）される障害も表示するには、このチェックボックスをオンにします。
障害のないグループを非表示	障害のないグループを非表示オプションをオンにすると、障害のないホストグループのデータをウィジェットで非表示にします。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害表示	障害数を次の形式で表示します。 すべて - 障害の総数を表示します。 分離 - 未確認の障害数を、障害総数とは分けて表示します。 未確認のみ - 未確認の障害数のみを表示します。
タイムラインを表示	視覚的なタイムラインを表示するには、このチェックボックスをオンにします。

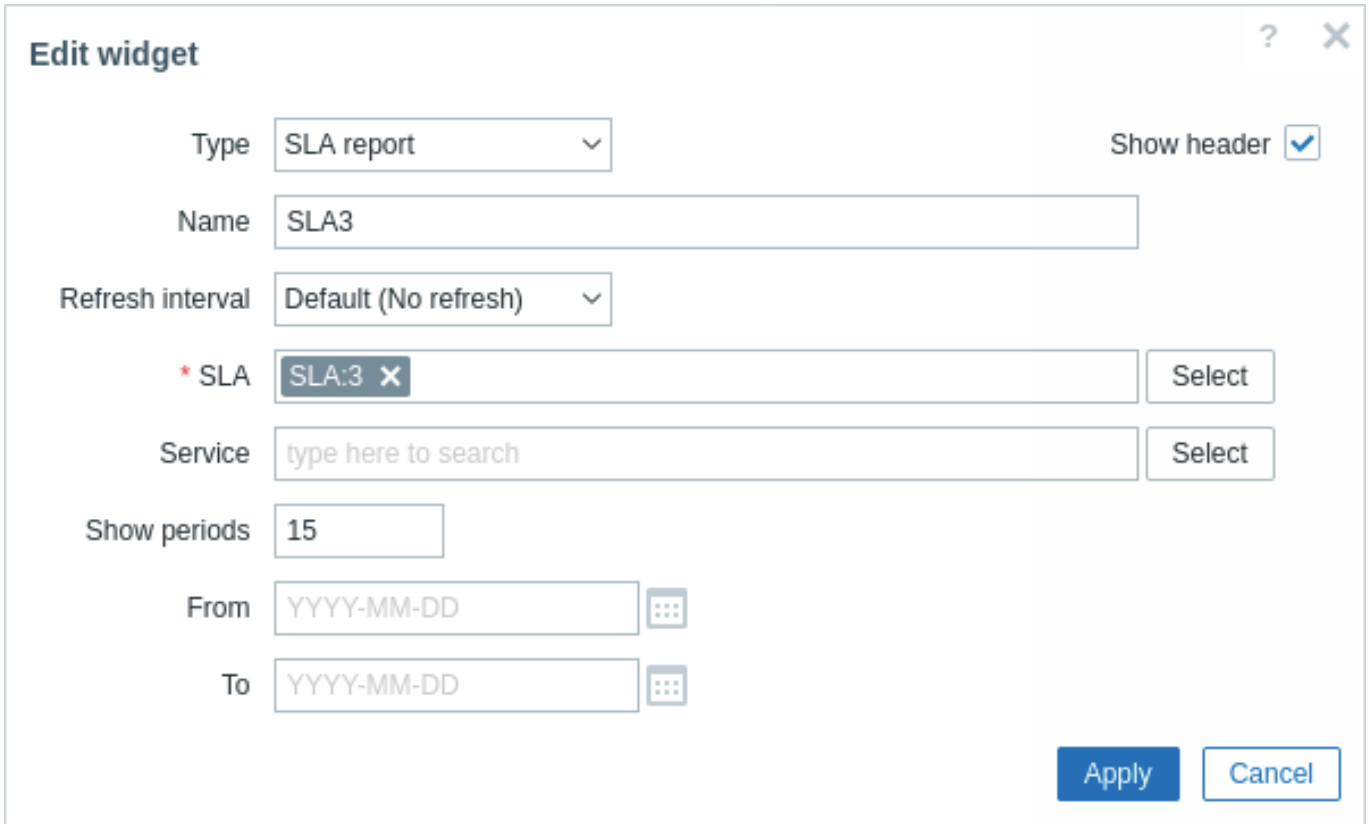
25 SLA レポート

概要

このウィジェットはSLA レポートを表示するのに役立ちます。機能的にはサービス -> *SLA* レポートセクションに似ています。

設定

設定するには、タイプとして *SLA* レポートを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

SLA	レポートの SLA を選択します。
Service	レポートのサービスを選択します。
Show periods	ウィジェットに表示する期間数を設定します（デフォルトは 20、最大 100）。
From	レポートの開始日を選択します。 相対日付をサポートしています: `now`, `now/d`, `now/w-1w` など。サポートされる日付修飾子: d, w, M, y。

To	レポートの終了日を選択します。 相対日付をサポートしています: `now`, `now/d`, `now/w-1w` など。サポートされる日付修飾子: d, w, M, y。

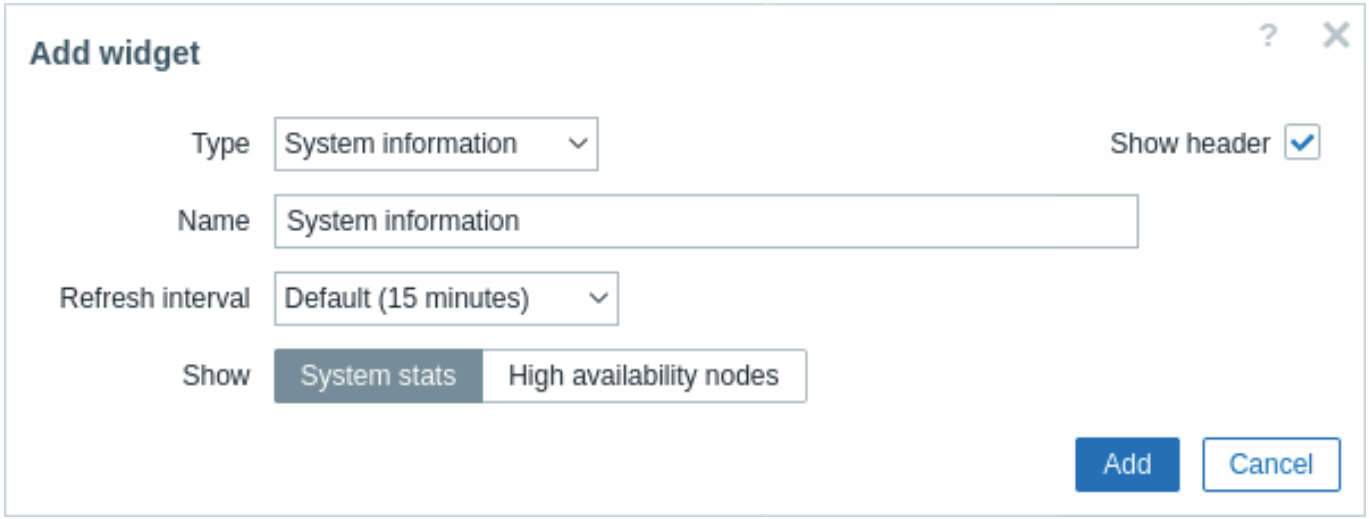
26 システム情報

概要

このウィジェットはレポート → システム情報と同じ情報を表示しますが、単一のダッシュボードウィジェットは一度にシステム統計または HA ノードのいずれかのみを表示できます。

設定

設定するにはタイプとしてシステム情報を選択します。



すべてのウィジェットの共通パラメーターに加えて、次の特定のオプションを設定できます。

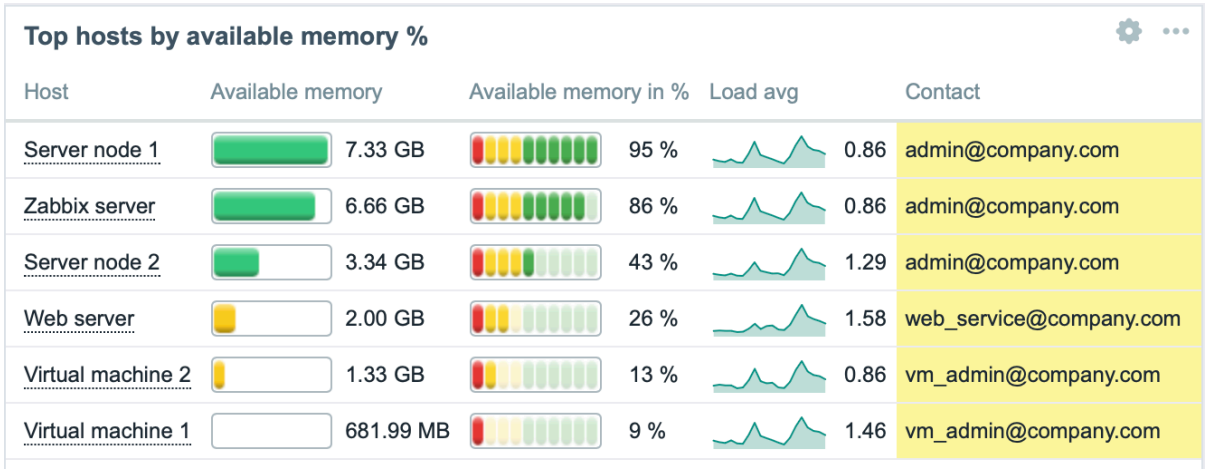
表示	表示する内容を選択します: システムステータス - 主要な Zabbix サーバーとシステムデータの概要を表示します。 **HA** ノード - 高可用性ノードのステータスを表示します (高可用性クラスターが有効な場合)。
ソフトウェアアップデート確認の詳細を表示	Zabbix ソフトウェアアップデート確認の詳細を表示するには、チェックボックスをオンにします。 このオプションは、Zabbix サーバー設定でソフトウェアアップデート確認が有効になっており、表示フィールドで” システム統計ステータス” が選択されている場合にのみ使用できます。

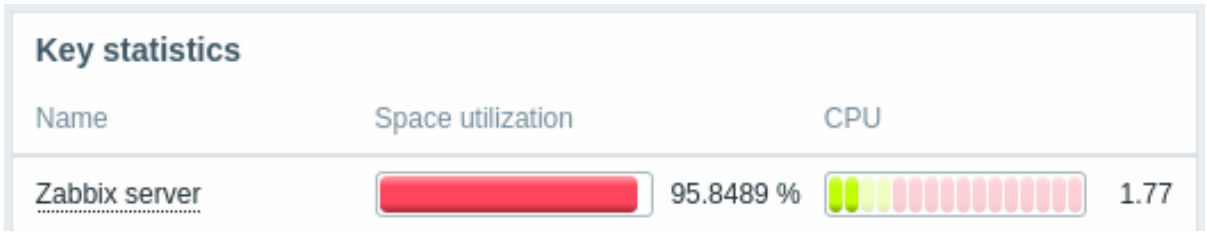
27 上位ホスト

概要

Top hosts ウィジェットでは、複数のホストのデータを表示するためのカスタムテーブルを作成でき、*Top N* のようなレポートや、キャパシティプランニングに役立つプログレスバー形式のレポートの作成に利用できます。表示するデータの種類はカスタマイズ可能で、CPU 使用率、メモリの空き容量、その他の収集された数値、テキスト、バイナリ形式の値に加え、静的なテキスト文字列も含めることができます。データは、バー、インジケーター、スパークラインチャートなど、さまざまな形式で表示できます。

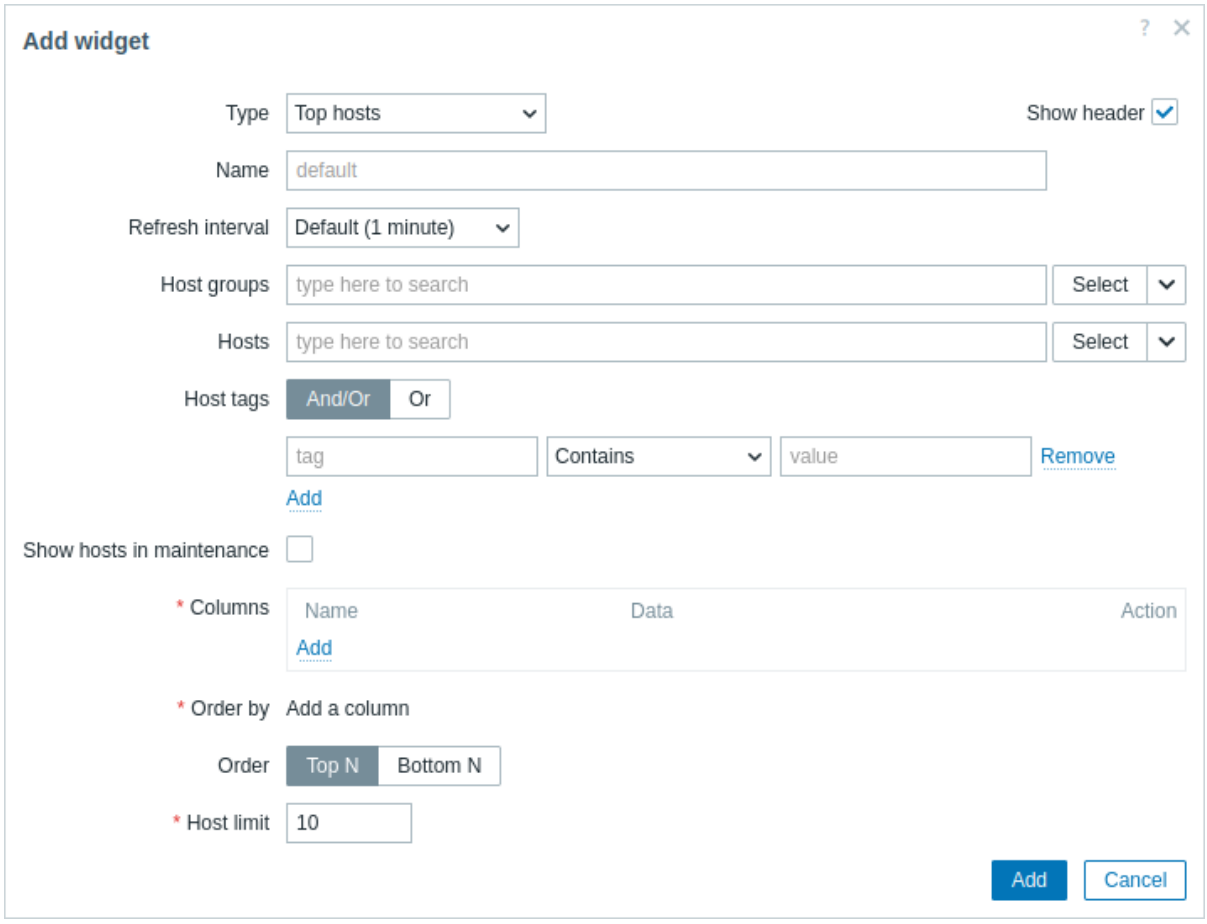
表示できるホストの最大数は 1000 です。





設定

設定するにはタイプとして上位ホストを選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

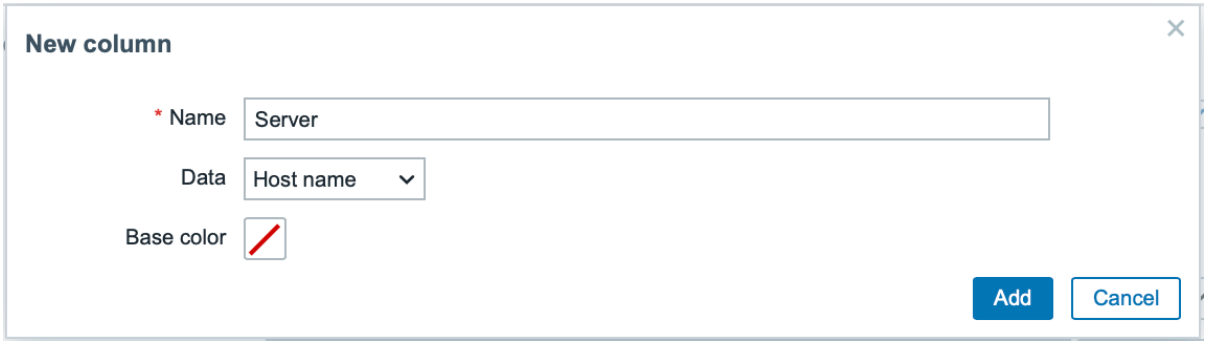
ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホストタグ	ウィジェットに表示されるホスト数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
メンテナンス中のホストを表示	このチェックボックスをオンにすると、メンテナンス中のホストも表示されます（この場合、ホスト名の横にメンテナンスアイコンが表示されます）。デフォルトではオフです。
列	表示するデータ列を追加します。 列の順序によって、左から右への表示順が決まります。 列名の前にあるハンドルを上下にドラッグして、列の順序を変更できます。
並べ替え基準	*Top N* または *Bottom N* の並べ替えに使用する列を、定義済みの *Columns* リストから指定します。
順序	行の並び順を指定します。 **Top N** - *Order by* の集計値に従って降順で表示します。 **Bottom N** - *Order by* の集計値に従って昇順で表示します。
ホスト数の上限	表示するホスト行数です（1～1000）。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

列の設定

3 種類のデータ型の列を追加できます: ホスト名、アイテム値、またはテキスト。利用可能な列パラメータの一覧は、列のデータ型、およびアイテム値タイプの場合は値の形式によって異なります。

ホスト名列

ホスト名列は、ホストの名前を表示するために使用されます。

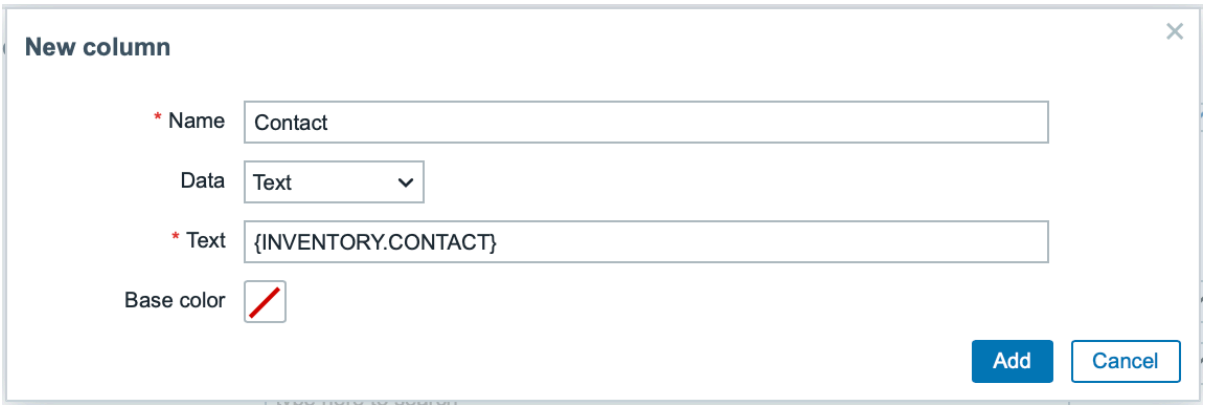


サポートされているパラメータ:

Name	列の名前。
Data	列に表示するデータ型。ホスト名を選択します。
Base color	列の背景色。

テキスト列

テキスト列は、指定した任意のテキスト文字列を表示するために使用します。

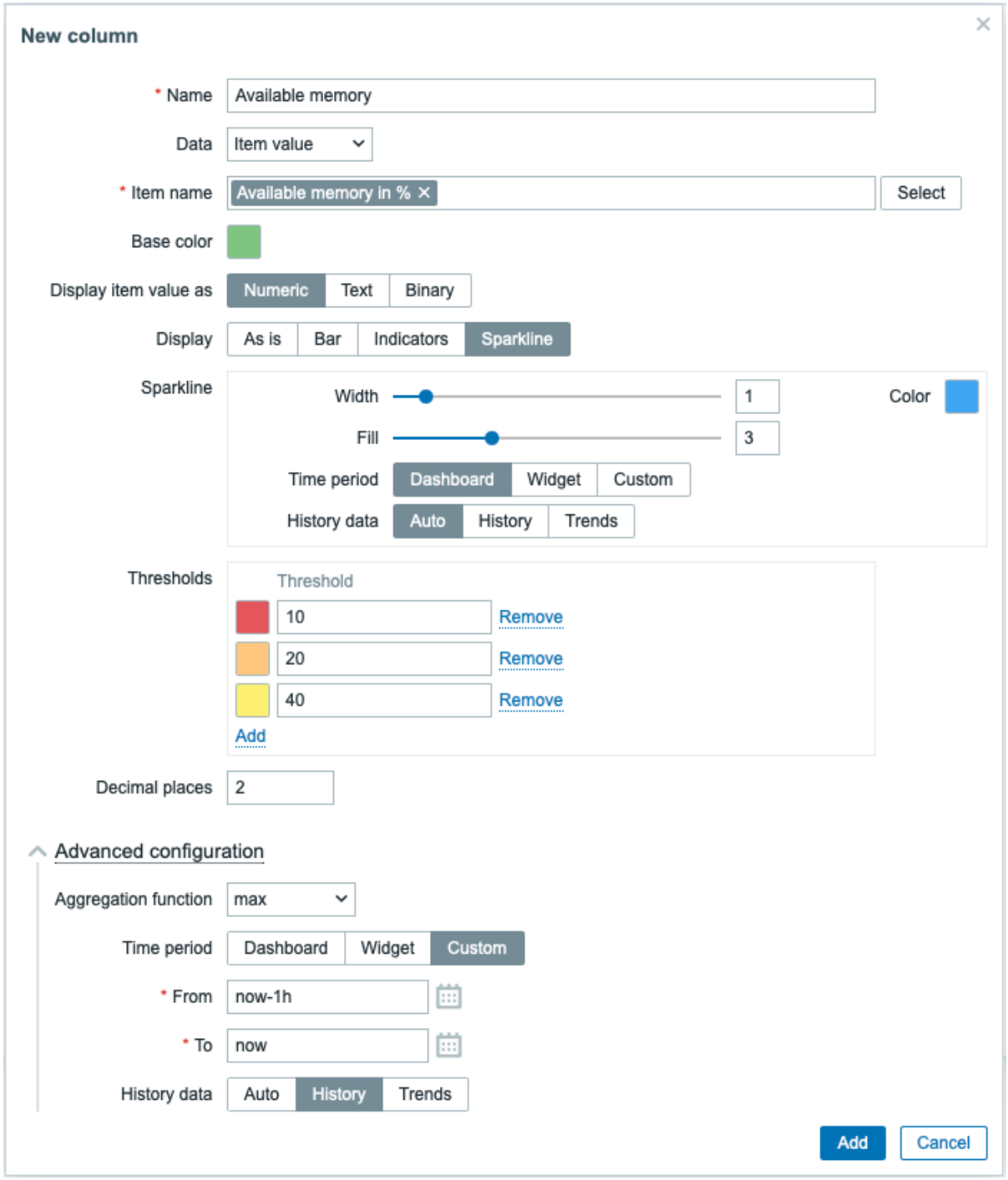


サポートされているパラメータ:

名前	列の名前。
データ	列に表示するデータ型です。テキストを選択します。
ベース色	列の背景色。
テキスト	表示する文字列を入力します。 ホストおよびインベントリのマクロを含めることができます。

アイテム値列

アイテム値列は、指定したアイテムの値を表示するために使用されます。複数の値表示形式とオプションをサポートしています。



サポートされているパラメータ:

Name	列の名前。
Data	列に表示するデータ型です。**Item value** を選択します。
Item name	アイテムを選択します。その名前は、選択したすべてのホスト上で同じ名前を持つアイテムを照合して表示するために使用されます。ホストにそのようなアイテムが複数ある場合、ウィジェットはキーがアルファベット順で最初のアイテムを表示します（例: キー `system.cpu.util` ではなく `proc.cpu.util` を持つ *CPU utilization*）。 ウィジェットをテンプレートダッシュボードで設定する場合、選択できるのはテンプレート上で設定されたアイテムのみです。
Base color	列の背景色です。*Item value* データがバー/インジケーターとして表示される場合は塗りつぶし色になります。アイテム値が指定したしきい値のいずれかを超えた場合、デフォルト色はカスタム色で上書きできます。
Display item value as	アイテム値の表示形式です: **Numeric**、**Text**、または **Binary**。 選択したオプションに応じて、利用可能な追加パラメータが決まります。以下の各形式のパラメーター一覧を参照してください。
Advanced configuration	

Aggregation function	使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の件数を表示; **sum** - 合計値を表示; **first** - 最初の値を表示; **last** - 最後の値を表示; **not used** - 最新の値を表示（集計なし）。 集計を使用すると、最新の値の代わりに、選択した期間（5 分、1 時間、1 日）の集計値を表示できます。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では、非数値データは数値に変換されます。
Time period	値の集計に使用する期間を指定します: **Dashboard** - ダッシュボードの期間を使用; **Widget** - 指定したウィジェットの期間を使用; **Custom** - カスタム期間を使用。 *Aggregation function* が”not used” に設定されている場合、このパラメータは表示されません。
Widget	ウィジェットを選択します。 このパラメータは、*Time period* が”Widget” に設定されている場合にのみ表示されます。
From	期間の開始を選択します（デフォルト値は `now-1h`）。相対時間構文を参照してください。 このパラメータは、*Time period* が”Custom” に設定されている場合にのみ表示されます。
To	期間の終了を選択します（デフォルト値は `now`）。相対時間構文を参照してください。 このパラメータは、*Time period* が”Custom” に設定されている場合にのみ表示されます。
History data	履歴またはトレンドからデータを取得します: **Auto** - 自動選択; **History** - 履歴データを取得; **Trends** - トレンドデータを取得。 このパラメータは数値アイテム値でのみ使用できます。テキスト値およびバイナリ値では、データは常に履歴から取得されます。

数値アイテム値のパラメータ:

Display	値の表示方法を定義します: **As is** - 通常のテキストとして表示; **Bar** - 単色の塗りつぶしバーとして表示; **Indicators** - 区切られた塗りつぶしバーとして表示; **Sparkline** - ミニ折れ線グラフ。
Thresholds	背景色/塗りつぶし色を変更するしきい値を指定します。 リストは保存時に昇順で並べ替えられます。 スパークラインでは、しきい値はアイテムの最新値にのみ適用されます。
Decimal places	値とともに表示する小数点以下の桁数を指定します。
Bar/Indicators	
Min	最小値。
Max	最大値。
Sparkline	
Width	スライダーを使用するか、0～10 の範囲の値を手動で入力して、グラフ線の太さを設定します。
Color	線と塗りつぶしの色を選択します。
Fill	スライダーを使用するか、0～10 の範囲の値を手動で入力して、塗りつぶし色の透明度を設定します。
Time period	スパークラインチャートに含める値の期間を指定します: **Dashboard** - ダッシュボードの期間を使用; **Widget** - 指定したウィジェットの期間を使用; **Custom** - カスタム期間を使用。 この期間はスパークラインチャートにのみ影響する点に注意してください。スパークラインの横に表示される値は、選択した期間内の最後の値ではなく、アイテムの最新値を表します。
Widget	ウィジェットを選択します。 このパラメータは、*Time period* が”Widget” に設定されている場合にのみ表示されます。

From	期間の開始を選択します（デフォルト値は `now-1h`）。相対時間構文を参照してください。 このパラメータは、*Time period* が”Custom” に設定されている場合にのみ表示されます。
To	期間の終了を選択します（デフォルト値は `now`）。相対時間構文を参照してください。 このパラメータは、*Time period* が”Custom” に設定されている場合にのみ表示されます。
History data	履歴またはトレンドからデータを取得します: **Auto** - 自動選択; **History** - 履歴データを取得; **Trends** - トレンドデータを取得。

テキストアイテム値のパラメータ:

Highlights	一致した場合に背景色/塗りつぶし色を変更する正規表現を指定します。

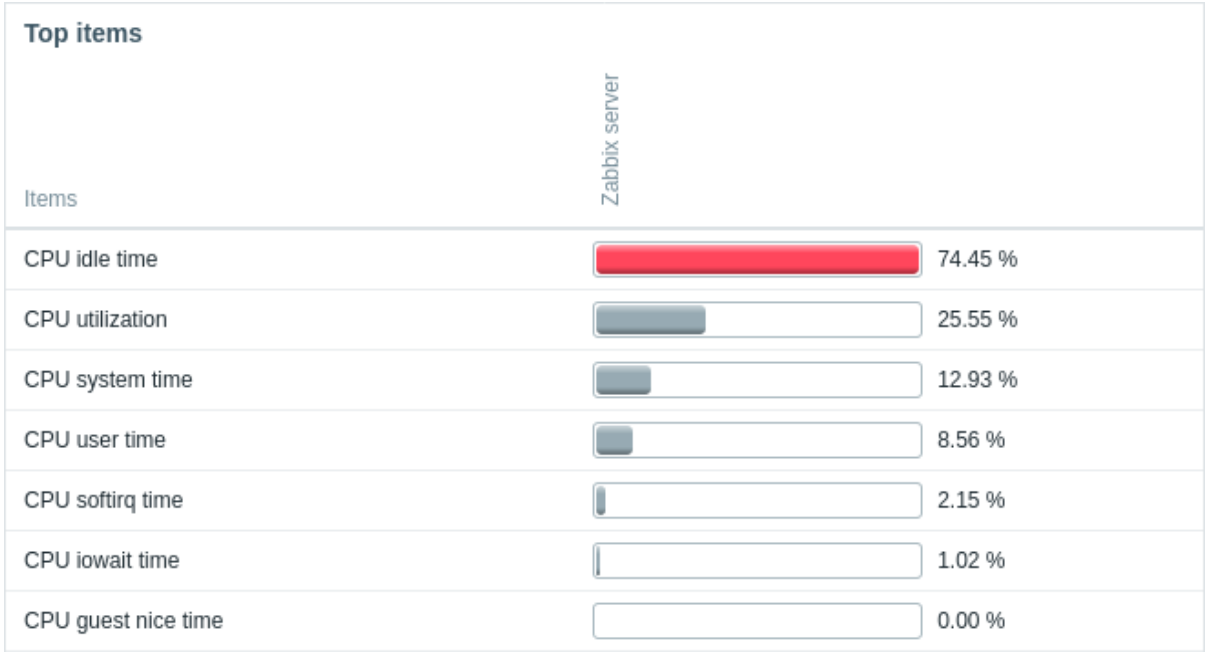
バイナリアイテム値のパラメータ:

Show thumbnail	バイナリデータを含む画像のサムネイルを作成して表示するか、または値列にフルサイズ画像へのリンク *Show* を表示するかを指定します。

28 上位アイテム

概要

Top items ウィジェットでは、アイテムのグループに対して上位（または下位）の値を表示できます。



アイテムを直接選択することも、パターンで選択することもでき、列の並び順や強調表示をカスタマイズできます。

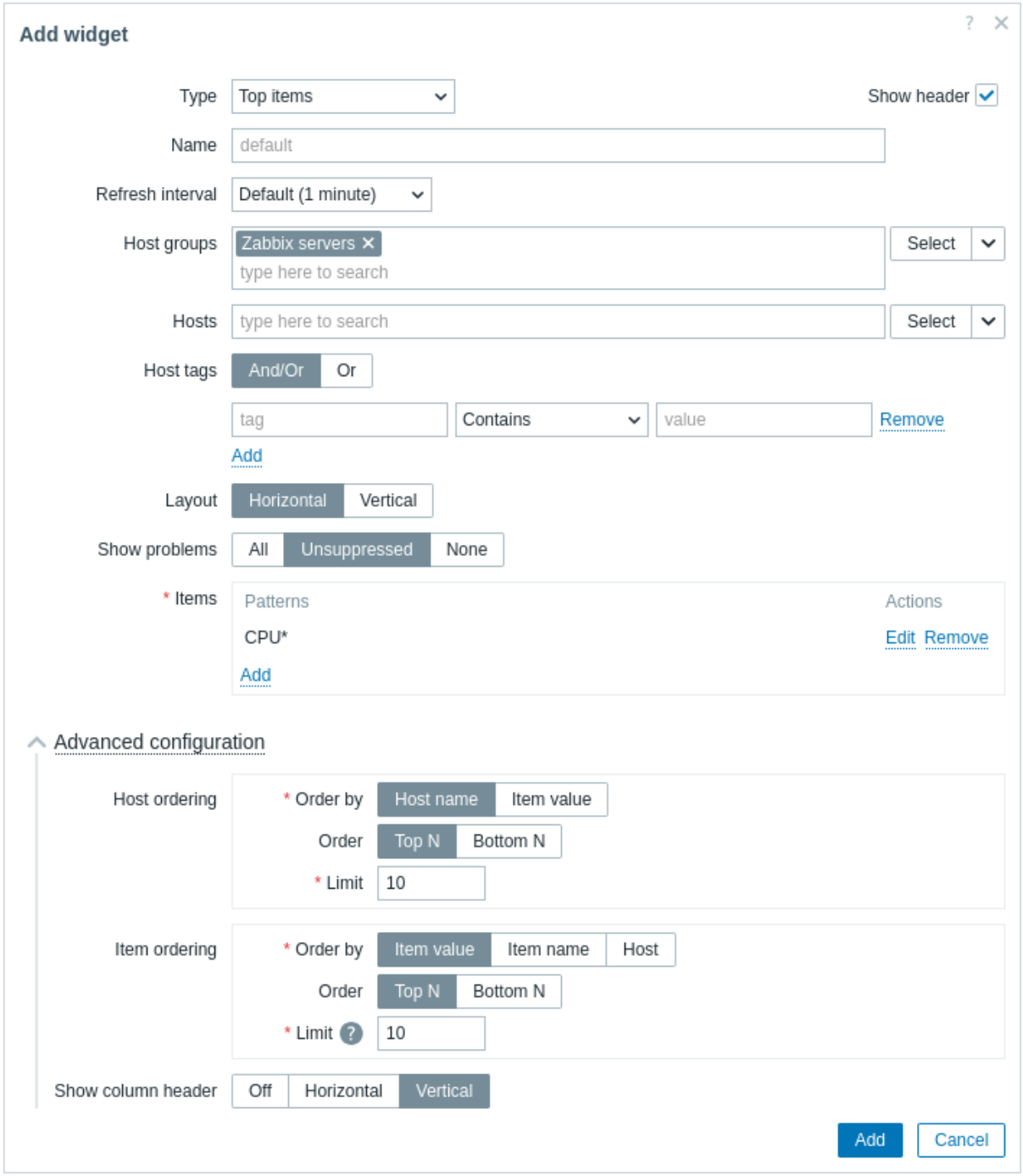
障害アイテムの色は障害の深刻度の色に基づいており、この色は障害の更新画面で調整できます。

デフォルトでは、直近 24 時間以内の値のみが表示されます。この制限は、最新データの大きなページにおける初期読み込み時間を改善することを目的として導入されました。この制限は、*Administration* → *General* → *GUI* の *Max history display period* オプションで設定できます。

アイテムの値をクリックすると、アイテムメニューが開きます。

設定

設定するには、タイプとして *Top items* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

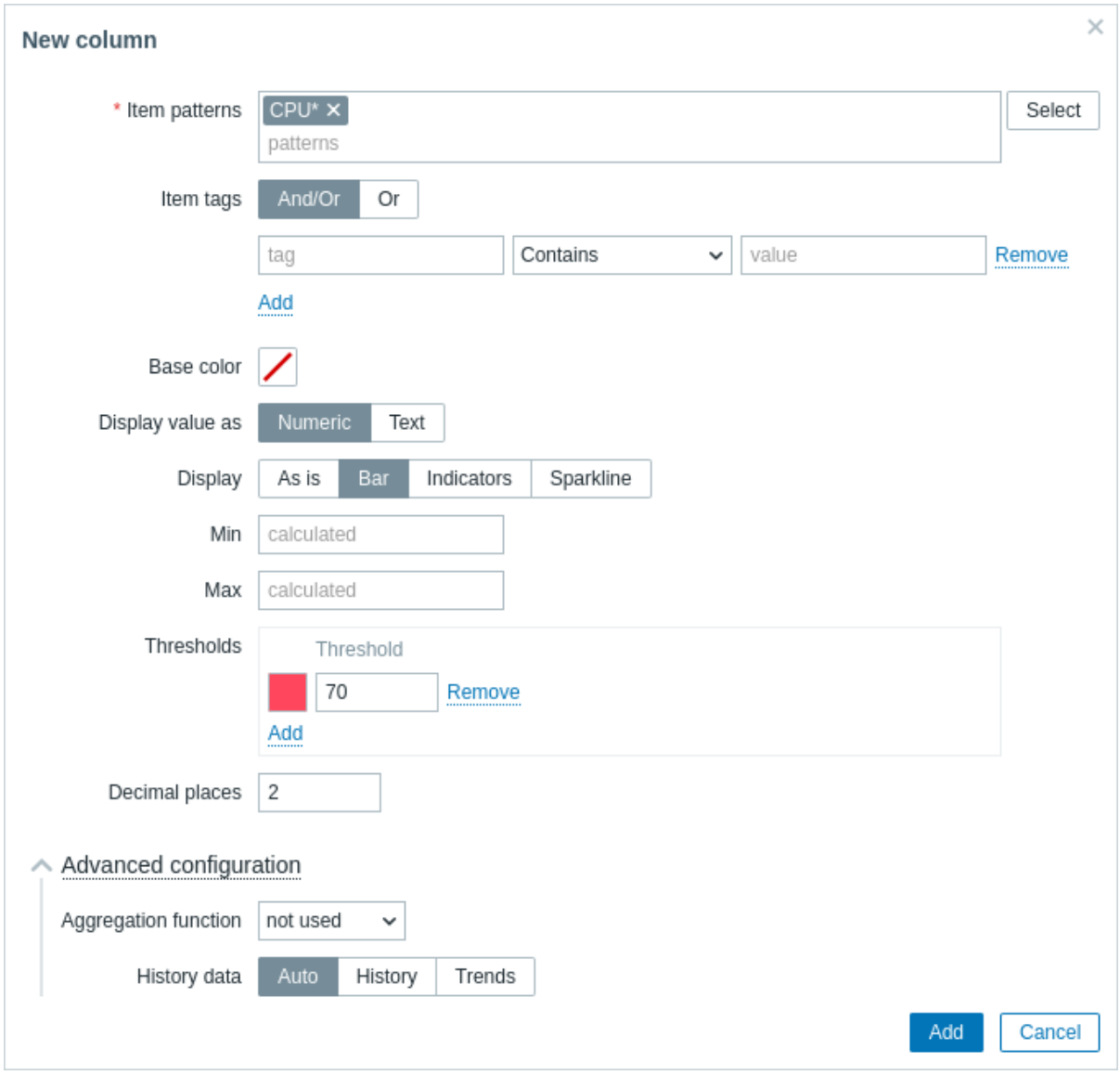
ホストグループ	ホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

ホストタグ		ウィジェットに表示するホストデータの数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
レイアウト		レイアウトオプションを選択します。 **Horizontal** - ホスト名は左側に表示されます。 **Vertical** - ホスト名は上部に表示されます。
障害の表示		障害の表示方法を選択します。 **All** - すべての障害を表示します。 **Unsuppressed** - ホストメンテナンスによって抑制されていない障害のみを表示します。 **None** - 障害をまったく表示しません。 問題状態が表示される場合、アイテム列設定の表示スタイル設定よりもそちらが優先される点に注意してください。
アイテム詳細設定		表示するアイテムパターンまたは特定のアイテムを追加します（列設定を参照）。
ホストの並び順		ホスト列/行の並び順オプションを選択します。
	Order by	ホストの並び順を選択します。 **Host name** - ホスト名で並べ替えます。 **Item value** - 選択したアイテムの値で並べ替えます。
	Order	最大値または最小値のどちらを表示するかを選択します。 **TopN** - 上位 N 件の値。 **BottomN** - 下位 N 件の値。 N の値は *Limit* フィールドで選択します。
	Limit	表示するホスト数の上限を入力します（範囲 1～1000、デフォルト 10）。 この値が *Order* フィールドの N の値になります。
アイテムの並び順		アイテム列/行の並び順オプションを選択します。

	Order by	アイテムの並び順を選択します。 **Item value** - アイテム値で並べ替えます。 **Item name** - アイテム名で並べ替えます。 **Host** - 選択したホストパターンで並べ替えます。
	Order	最大値または最小値のどちらを表示するかを選択します。 **TopN** - 上位 N 件の値。 **BottomN** - 下位 N 件の値。 N の値は *Limit* フィールドで選択します。
	Limit	表示するアイテム数の上限を入力します（範囲 1～1000、デフォルト 10）。 この値が *Order* フィールドの N の値になります。
列ヘッダーの表示		列ヘッダーの表示オプションを選択します。 **Off** - 列ヘッダーを表示しません。 **Horizontal** - ヘッダー内のテキストを横向きに表示します。 **Vertical** - ヘッダー内のテキストを縦向きに表示します。

列の設定

アイテム列を設定するには、アイテムパラメータで 追加をクリックします。



共通の列パラメータ:

アイテムパターン	ワイルドカード文字を使用して、1 つまたは複数のアイテムパターンを指定します。 または、アイテムを選択します。 ウィジェットをテンプレートダッシュボードで設定する場合、選択できるのはテンプレート上で設定されたアイテムのみです。
アイテムタグ	ウィジェットに表示されるアイテムデータの数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - 1 つの条件を満たせば十分です。
ベース色	列の背景色を選択します。表示が「Bar」または「Indicators」に設定されている場合は塗りつぶし色を選択します。 ベース色はしきい値の色または強調表示の色によって上書きされる場合があることに注意してください。
値の表示形式	アイテム値の表示形式: **Numeric** または **Text**。 選択したオプションによって、利用可能な追加パラメータが決まります。各形式のパラメータ一覧は以下を参照してください。

数値の値の列パラメータ:

表示	値の表示方法を定義します: **As is** - 通常のテキストとして表示; **Bar** - 単色で塗りつぶされたバーとして表示; **Indicators** - 区切られた単色のバーとして表示; **Sparkline** - ミニ折れ線グラフとして表示。
最小	バー/インジケーター表示の最小値。
最大	バー/インジケーター表示の最大値。
幅	スライダーを使用するか、0～10 の範囲の値を手動で入力して、グラフ線の太さを設定します。 このパラメータは Sparkline 表示でのみ使用されます。
塗りつぶし	スライダーを使用するか、0～10 の範囲の値を手動で入力して、塗りつぶし色の透明度を設定します。 このパラメータは Sparkline 表示でのみ使用されます。
色	線と塗りつぶしの色を選択します。 このパラメータは Sparkline 表示でのみ使用されます。
時間範囲および履歴データの選択に関連するフィールドの説明については、高度な設定を参照してください。	
しきい値	背景色/塗りつぶし色を変更するしきい値を指定します。 リストは保存時に昇順で並べ替えられます。 Sparkline では、しきい値はアイテムの最新の値にのみ適用されます。
小数点以下の桁数	値とともに表示する小数点以下の桁数を指定します。
高度な設定	

集計関数	使用する集計関数を指定します: **min** - 最小値を表示; **max** - 最大値を表示; **avg** - 平均値を表示; **count** - 値の件数を表示; **sum** - 値の合計を表示; **first** - 最初の値を表示; **last** - 最後の値を表示; **not used** - 最新の値を表示（集計なし）。 集計を使用すると、最新の値の代わりに、選択した間隔（5 分、1 時間、1 日）の集計値を表示できます。 *min*、*max*、*avg*、*sum* では数値データのみ表示できます。*count* では、非数値データは数値に変換されます。
時間範囲	値の集計に使用する時間範囲を指定します: **Dashboard** - ダッシュボードの時間範囲を使用; **Widget** - 指定したウィジェットの時間範囲を使用; **Custom** - カスタム時間範囲を使用。 集計関数が"not used" に設定されている場合、このパラメータは表示されません。
ウィジェット	ウィジェットを選択します。 このパラメータは、時間範囲が"Widget" に設定されている場合にのみ表示されます。
開始	開始する時間範囲を選択します（デフォルト値 `now-1h`）。相対時間の構文を参照してください。 このパラメータは、時間範囲が"Custom" に設定されている場合にのみ表示されます。
終了	終了する時間範囲を選択します（デフォルト値 `now`）。相対時間の構文を参照してください。 このパラメータは、時間範囲が"Custom" に設定されている場合にのみ表示されます。
履歴データ	履歴またはトレンドからデータを取得します: **Auto** - 自動選択; **History** - 履歴データを取得; **Trends** - トレンドデータを取得。 このパラメータは数値アイテムの値でのみ使用できます。テキスト値のデータは常に履歴から取得されます。

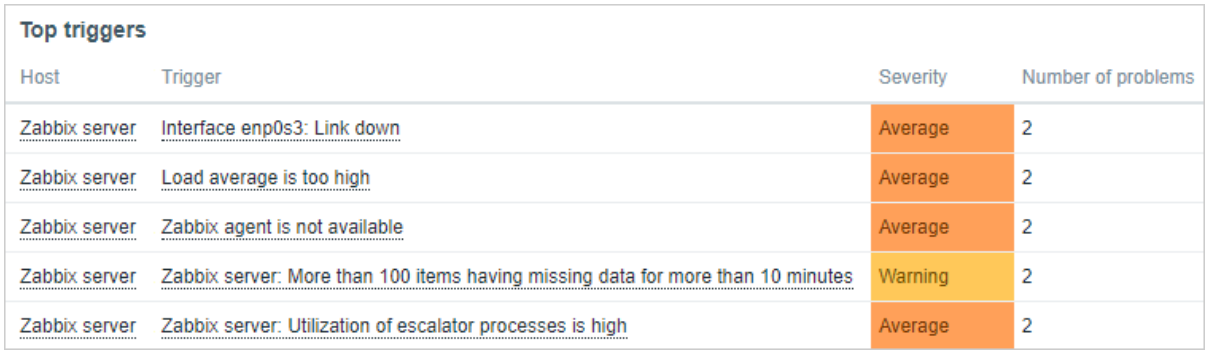
テキスト値の列パラメータ:

ハイライト	背景色/塗りつぶし色を変更する正規表現を指定します。

29 上位トリガー

概要

Top triggers ウィジェットでは、障害の発生数が最も多いトリガーを確認できます。

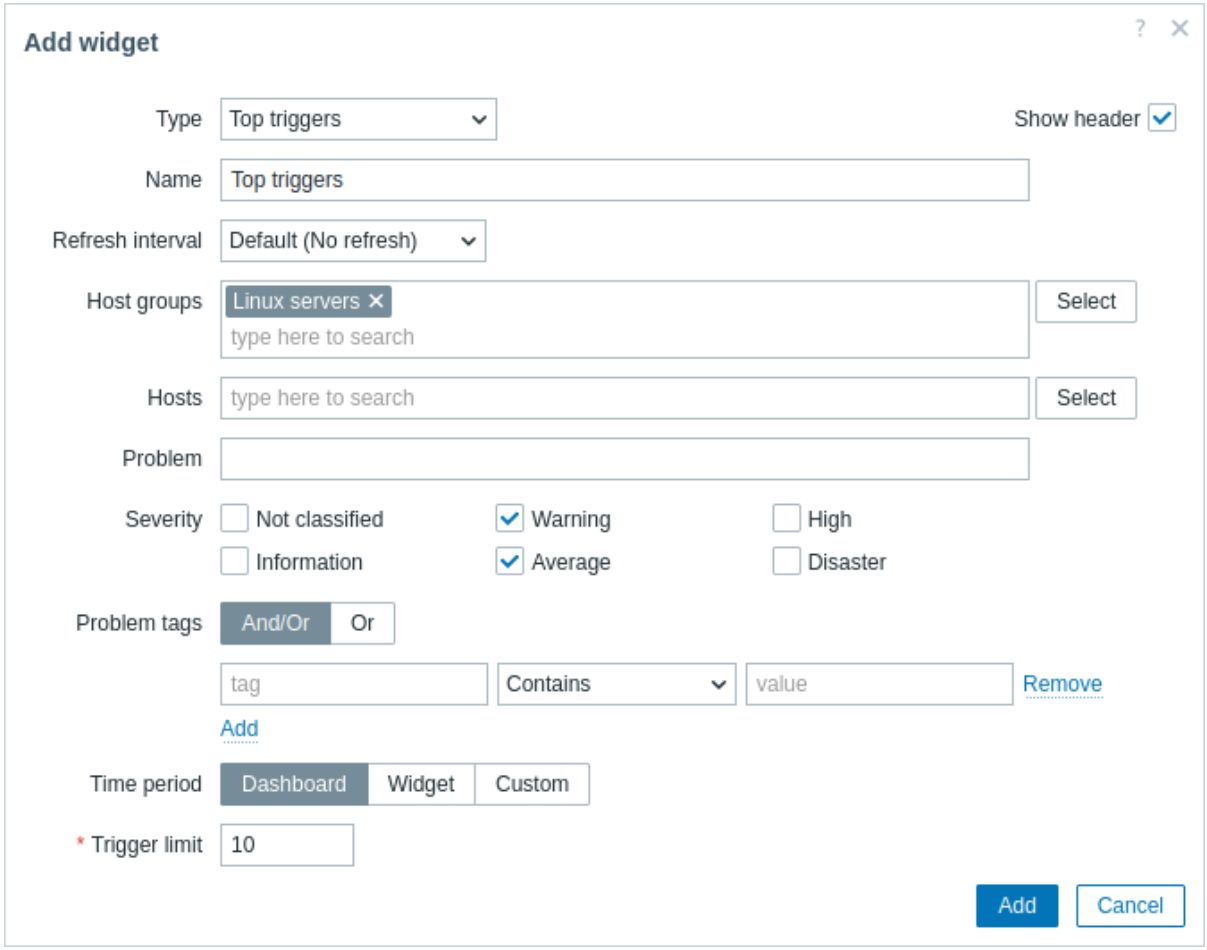


表示できるトリガーの最大数は 1000 です。ダッシュボード上でウィジェットを表示している場合は、データを表示する期間を選択できます。

上位トリガーの情報は、レポート → *Top 100 triggers* メニューセクションでも利用できます。

設定

設定するには、ウィジェットタイプとして *Top triggers* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有オプションを設定できます。

ホストグループ	ウィジェットに表示する障害のホストグループを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループの障害がウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ウィジェットに表示する障害のホストを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストの障害が表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
障害	特定の障害に対するトリガーのみを表示できます。これを行うには、障害名で一致させる文字列を入力します。 マクロは展開されません。
深刻度	ウィジェットに表示するトリガーを絞り込むため、トリガーの深刻度を選択します。 深刻度が 1 つも選択されていない場合は、すべてのトリガーが表示されます。

障害タグ	ウィジェットに表示する障害のタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
期間	期間のデータソースを選択します。 **Dashboard** - 期間セレクターをデータソースとして設定します。 **Widget** - *Widget* パラメータで指定した互換性のあるウィジェットをデータソースとして設定します。 **Custom** - *From* および *To* パラメータで指定した期間をデータソースとして設定します。これを設定すると、ウィジェット右上に時計アイコンが表示され、マウスオーバーで設定時刻が示されます。
Widget	期間のデータソースとして、互換性のあるウィジェット（*Graph*、*Graph (classic)*、*Graph prototype*）を入力または選択します。 このパラメータは、期間が”Widget” に設定されている場合に使用できます。
From	期間の開始を入力または選択します。 相対時間構文 (`now`, `now/d`, `now/w-1w` など) に対応しています。 このパラメータは、期間が”Custom” に設定されている場合に使用できます。
To	期間の終了を入力または選択します。 相対時間構文 (`now`, `now/d`, `now/w-1w` など) に対応しています。 このパラメータは、期間が”Custom” に設定されている場合に使用できます。
トリガー数上限	表示するトリガー数を設定します。指定可能な値の範囲は 1～1000 です。

30 トリガーの概要

概要

トリガー概要ウィジェットでは、ホストのグループに対するトリガーの状態を表示できます。

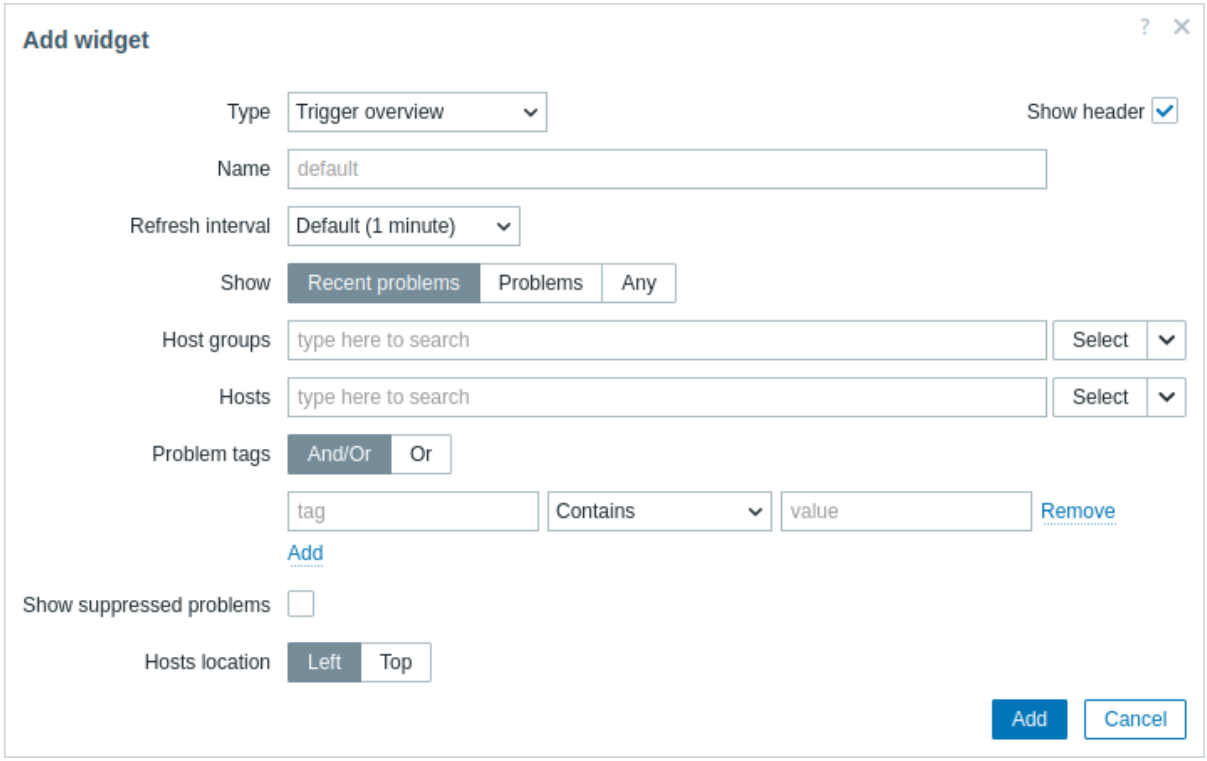
- トリガーの状態は色付きのブロックとして表示されます（PROBLEM トリガーのブロックの色は障害の深刻度の色に依存し、この色は障害の更新画面で調整できます）。なお、直近のトリガー状態の変更（過去 2 分以内）は点滅するブロックとして表示されます。
- 灰色の上下矢印は、依存関係を持つトリガーを示します。マウスオーバーすると、依存関係の詳細が表示されます。
- チェックボックスのアイコンは、確認済みの障害を示します。このアイコンを表示するには、トリガーのすべての障害、または解決済みの障害が確認済みである必要があります。

トリガーブロックをクリックすると、コンテキストに応じて、トリガーの障害イベント、障害確認画面、トリガーの設定、トリガーURL、またはシンプルグラフ/最新データ一覧へのリンクが表示されます。

デフォルトでは 50 件のレコードが表示されます（*Administration → General → GUI の Max number of columns and rows in overview tables* オプションで設定可能）。表示対象として設定された件数を超えるレコードが存在する場合、テーブルの下部に、より具体的なフィルタリング条件を指定するよう求めるメッセージが表示されます。ページネーションはありません。なお、この制限はタグなどによる追加のデータフィルタリングよりも前に、最初に適用されます。

設定

設定するには、タイプとして トリガー概要を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

表示	トリガーをトリガーの状態でフィルタリングします: 最近の障害 - *(デフォルト)* 最近 PROBLEM 状態であった、または現在も PROBLEM 状態であるトリガーを表示します（解決済みおよび未解決）。 障害 - PROBLEM 状態のトリガーを表示します（未解決）。 すべて - すべてのトリガーを表示します。
ホストグループ	ホストグループを選択します。 または、ホストグループのdata sourceとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ホストを選択します。 または、ホストのdata sourceとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

障害タグ	ウィジェットに表示されるトリガーをフィルタリングするタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 注意**:** パラメータ 表示が「すべて」に設定されている場合、タグが指定されていてもすべてのトリガーが表示されます。ただし、最近のトリガー状態の変化（点滅するブロックとして表示）はすべてのトリガーで更新されますが、トリガー状態の詳細（障害の深刻度の色、および障害が確認済みかどうか）は、指定したタグに一致するトリガーについてのみ更新されます。 各条件で使用できる演算子は次のとおりです: **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
抑制された障害を表示	ホストのメンテナンスにより抑制（非表示）される障害も表示するには、このチェックボックスをオンにします。
レイアウト	レイアウトオプションを選択します: **Horizontal** - ホスト名は左側に表示されます。 **Vertical** - ホスト名は上部に表示されます。

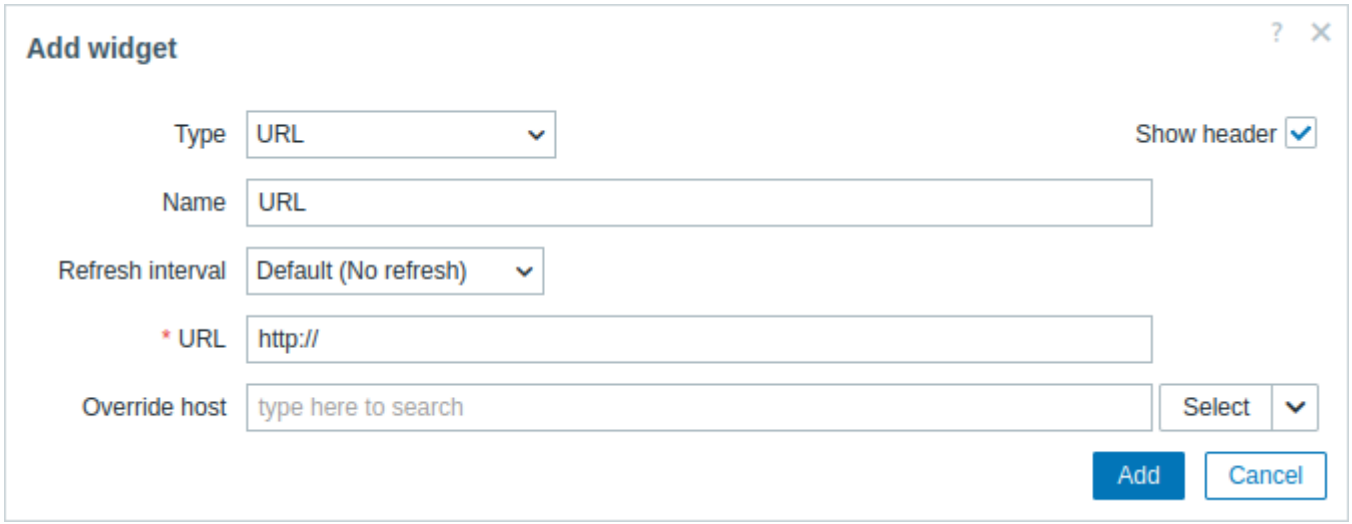
31 URL

概要

このウィジェットは、指定された URL から取得されたコンテンツを表示します。

設定

設定するには、タイプとして *URL* を選択します。



すべてのウィジェットに共通のパラメータに加えて、以下の固有のオプションを設定できます。

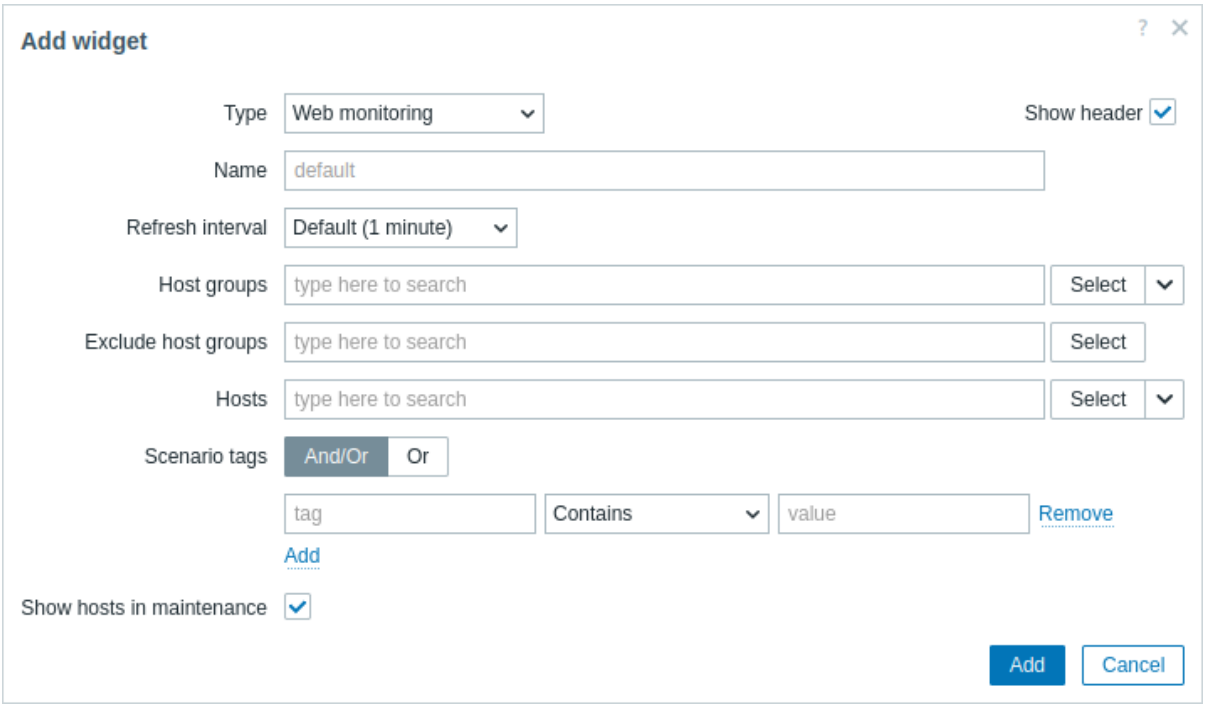
URL	表示する URL を入力します（最大 2048 文字）。 外部 URL は `http://` または `https://` で始まっている必要があります。 内部 URL では相対パスをサポートします（例: `zabbix.php?action=report.status`）。 {HOST.*} マクロをサポートしています。
Override host	ホストのデータソースとして、互換性のあるウィジェットまたはダッシュボードを選択します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。

Attention:
Zabbix Web インターフェースに HTTPS でアクセスしている場合、ブラウザはウィジェットで設定された HTTP ページを読み込めないことがあります。

概要

このウィジェットは、有効な Web 監視シナリオのステータス概要を表示します。詳細については、Web monitoring widget セクションを参照してください。

設定



Note:
ユーザーに特定のウィジェット要素へアクセスする権限がない場合、その要素名はウィジェットの設定中に *Inaccessible* と表示されます。その結果、要素の「実際の」名前の代わりに *Inaccessible Item*、*Inaccessible Host*、*Inaccessible Group*、*Inaccessible Map*、*Inaccessible Graph* が表示されます。

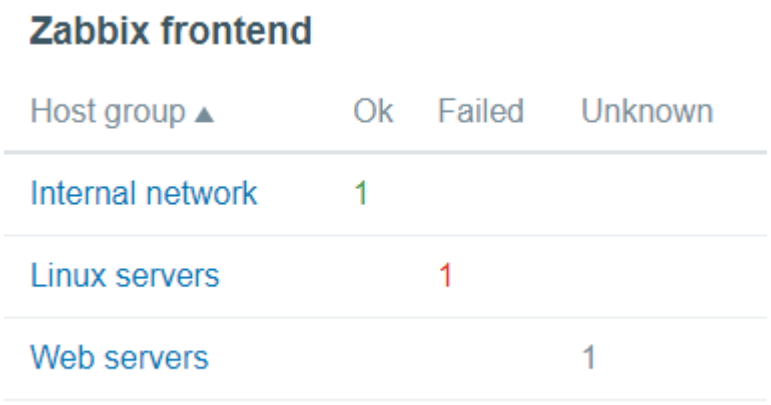
すべてのウィジェットに共通するパラメータに加えて、以下の固有オプションを設定できます。

ホストグループ	ウィジェットに表示するホストグループを選択します。 または、ホストグループのデータソースとして互換性のあるウィジェットを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループのホストデータがウィジェットに表示されます。ホストグループを入力しない場合は、すべてのホストグループが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
除外するホストグループ	ウィジェットで非表示にするホストグループを選択します。 このフィールドはオートコンプリートに対応しているため、グループ名の入力を開始すると、一致するグループのドロップダウンが表示されます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 これらのホストグループのホストデータはウィジェットに表示されません。たとえば、ホスト 001、002、003 がグループ A に属し、ホスト 002、003 がグループ B にも属しているとします。このとき、グループ A を表示し、同時にグループ B を除外すると、ダッシュボードにはホスト 001 のデータのみが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
ホスト	ウィジェットに表示するホストを選択します。 または、ホストのデータソースとして互換性のあるウィジェットまたはダッシュボードを選択します。 このフィールドはオートコンプリートに対応しているため、ホスト名の入力を開始すると、一致するホストのドロップダウンが表示されます。 ホストを入力しない場合は、すべてのホストが表示されます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合は使用できません。
シナリオタグ	ウィジェットに表示する Web シナリオ数を制限するためのタグを指定します。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 種類の計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は *Or* 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。
メンテナンス中のホストを表示	統計にメンテナンス中のホストを含めます。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合、メンテナンス中のデータを表示と表示されます。

Web 監視ウィジェット

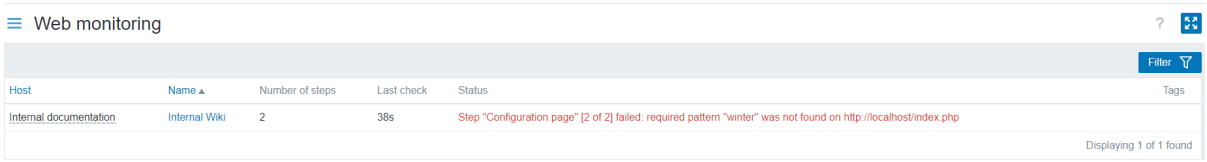
設定が完了したら、表示されるデータとともにウィジェットを確認したい場合があります。その場合は、ダッシュボードに移動し、ウィジェットを作成したダッシュボードの名前をクリックします。

この例では、”Zabbix frontend” という名前のウィジェットに、3 つのホストグループの Web 監視ステータスが表示されています。”Internal network”、”Linux servers”、”Web servers” です。

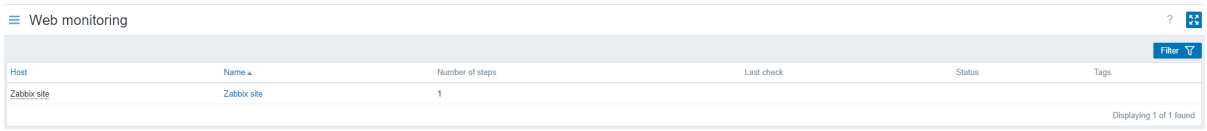


Web 監視ウィジェットには、次の情報が表示されます。

- ウィジェット名。その下に 4 つの列があります。
  - *Host group* - Web シナリオが設定されたホストを含むホストグループの一覧を表示します。
  - *Ok* - 次の 2 つの条件を満たす場合、Web シナリオの数を（緑色で）表示します。
    * Zabbix が 1 つ以上の Web シナリオの最新データを収集していること。
    * Web シナリオ で設定されたすべてのステップが、"Ok" の *Status* であること。
  - *Failed* - 失敗したステップがある Web シナリオの数を（赤色で）表示します。
    * ホスト名をクリックすると新しいウィンドウが開き、*Status* 列に、Zabbix がデータ収集に失敗したステップの詳細情報が（赤色で）表示されます。また、
    * 設定フォーム で修正が必要なパラメータについてのヒントも表示されます。



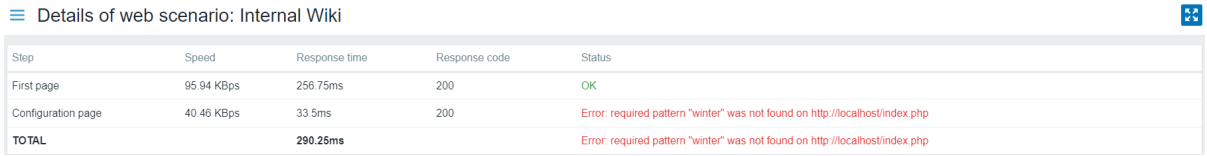
- *Unknown* - Zabbix がデータを収集しておらず、失敗したステップに関する情報もない Web シナリオの数を（灰色で）表示します。



ステータスとデータの表示

ウィジェット内のクリック可能なリンクを使用すると、各 Web シナリオの完全な情報へ簡単に移動し、すばやく確認できます。したがって、以下を表示するには:

- Web シナリオのステータスを表示するには、ホストグループ名をクリックします。
- より詳細な統計を表示するには、シナリオ名をクリックします。この例では「Zabbix frontend」です。
- *Failed* ステータスの場合の詳細を表示するには、ホストグループ名をクリックします。開いたウィンドウで、*Name* 列の Web シナリオ名をクリックします。すると、Zabbix がデータの収集に失敗した設定済みステップについて、より詳細な情報が表示されます。



これで、Web シナリオ設定フォームに戻って設定を修正できます。

Unknown ステータスの場合の詳細を表示するには、*Failed* の場合と同じ手順を繰り返します。

Attention:
最初の監視時には、Web シナリオは常に *Unknown* 状態で表示され、最初のチェックの直後に *Failed* または *Ok* 状態へ切り替わります。ホストがプロキシによって監視されている場合、ステータスの変更は、プロキシに設定されたデータ収集頻度に従って行われます。

2 監視データ

概要

監視データメニューは、すべてデータの表示に関するものです。Zabbix が収集、視覚化、操作するために設定されている情報が何であれ、それは監視データメニューのさまざまなセクションに表示されます。

表示モードボタン

右上隅にある以下のボタンは、すべてのセクションで共通です。

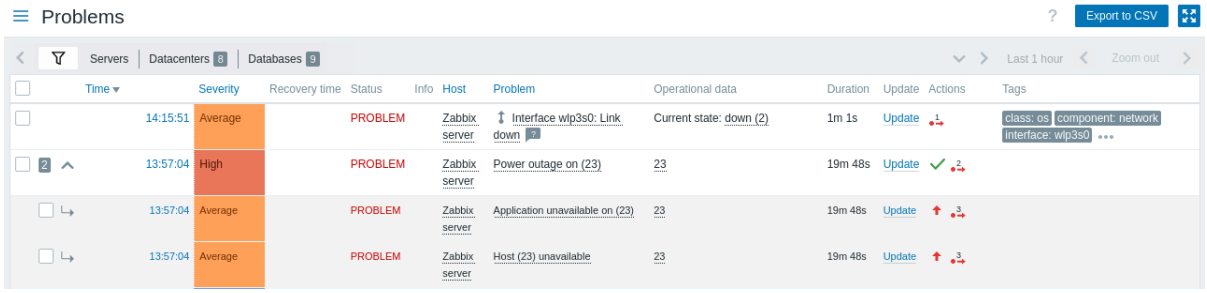
	ページをキオスクモードで表示します。このモードでは、ページの内容のみが表示されます。 キオスクモードを終了するには、  の終了ボタンが表示されるまでマウスカーソルを動かし、それをクリックします。通常モードに戻ります。

1 障害

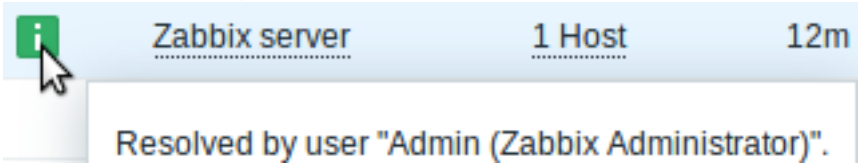
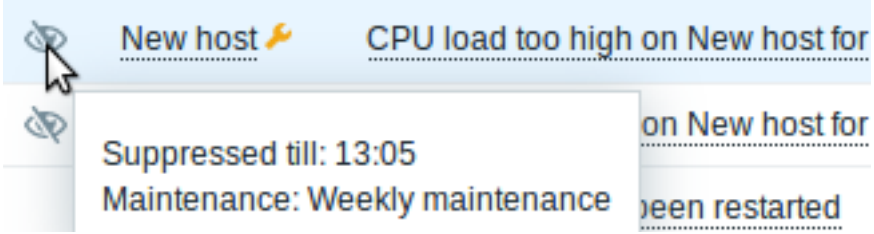
概要

監視 > 障害では、現在発生している障害を確認できます。障害とは、”Problem” 状態にあるトリガーです。

デフォルトでは、すべての新しい障害は原因障害として分類されます。特定の障害を、原因障害の症状障害として手動で再分類することも可能です。詳細については、原因イベントと症状イベントを参照してください。



列	説明
チェックボックス	障害を選択するためのチェックボックスが表示されます。 チェックボックスの横に表示されるアイコンには、次の意味があります。  - 原因障害に対する症状イベントの数。  - 展開して症状イベントを表示。  - 折りたたんで症状イベントを非表示。  - これは症状イベントです。
時刻	障害の開始時刻が表示されます。
深刻度	障害の深刻度が表示されます。 障害の深刻度は元々、基となる障害トリガーの深刻度に基づいていますが、イベント発生後に 障害の更新画面を使用して更新できます。障害発生中は、障害の深刻度の色がセルの背景色として使用されます。
復旧時刻	障害の解決時刻が表示されます。
ステータス	障害のステータスが表示されます。 **Problem** - 未解決の障害 **Resolved** - 最近解決された障害。フィルターを使用して最近解決された障害を非表示にできます。 新しい障害と最近解決された障害は 2 分間点滅します。解決済みの障害は合計 5 分間表示されます。 これらの値はどちらも 管理 > 一般 > トリガー表示オプション で設定できます。

列	説明
情報	障害がグローバル相関関係によって、または障害更新時に手動でクローズされた場合、緑色の情報アイコンが表示されます。アイコンにマウスカーソルを合わせると、詳細が表示されます。  抑制された障害が表示されている場合は、次のアイコンが表示されます（フィルターの 抑制された障害を表示オプションを参照）。アイコンにマウスカーソルを合わせると、詳細が表示されます。 
ホスト	障害が発生しているホストが表示されます。 ホスト名をクリックすると、ホストメニューが表示されます。
障害	障害名が表示されます。 障害名は、基となる障害トリガーの名前に基づいています。 トリガー名内のマクロは障害発生時に展開され、その後、展開済みの値は更新されません。 運用データを追加して、最新のアイテム値を障害名に表示することも可能です。 障害名をクリックすると、イベントメニューが表示されます。 障害名の後にある  アイコンにマウスカーソルを合わせると、トリガーの説明が表示されます(説明が設定されている障害のみ)。
運用データ	最新のアイテム値を含む運用データが表示されます。 運用データは、トリガーレベルで設定されている場合、テキストとアイテム値マクロの組み合わせにできます。トリガーレベルで運用データが設定されていない場合は、式に含まれるすべてのアイテムの最新値が表示されます。 この列は、フィルターの 運用データの表示で 個別が選択されている場合にのみ表示されます。
期間	障害の継続時間が表示されます。 参照: 負の障害継続時間
更新	更新リンクをクリックすると、障害更新画面に移動し、コメントの追加や障害の確認を含む、障害に対するさまざまな操作を実行できます。
アクション	障害に関するアクティビティの履歴が、記号アイコンを使用して表示されます。  - 障害が確認されています。このアイコンは常に最初に表示されます。  - コメントが追加されています。コメント数も表示されます。  - 障害の深刻度が上げられています（例: Information > Warning）。  - 障害の深刻度が下げられています（例: Warning > Information）。  - 障害の深刻度が変更された後、元のレベルに戻されています（例: Warning > Information > Warning）。  - 障害が抑制されています。  - 障害の抑制が解除されています。  - アクションが実行されています。アクション数も表示されます。  - アクションが実行されており、少なくとも 1 つが進行中です。アクション数も表示されます。  - アクションが実行されており、少なくとも 1 つが失敗しています。アクション数も表示されます。 アイコンにマウスカーソルを合わせると、アクティビティの詳細を示すポップアップが表示されます。実行されたアクションのポップアップで使用されるアイコンの詳細については、詳細の表示を参照してください。

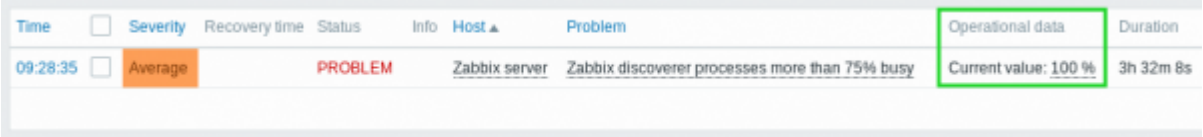
列	説明
タグ	タグが表示されます（存在する場合）。 さらに、外部チケットシステムのタグが表示されることもあります（webhook設定時の タグを処理オプションを参照）。

障害の運用データ

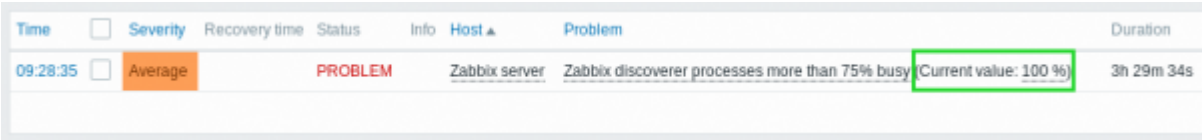
現在の障害について運用データ、つまり障害発生時点のアイテム値ではなく、最新のアイテム値を表示できます。

運用データの表示は、監視 > 障害のフィルター、または該当するダッシュボードウィジェットの設定で、次の 3 つのオプションから選択して設定できます。

- なし - 運用データは表示されません
- 別表示 - 運用データは別の列に表示されます



- 障害名とともに表示 - 運用データは障害名の末尾に括弧付きで追加されます。運用データは、トリガー設定の *Operational data* フィールドが空でない場合にのみ、障害名に追加されます。



運用データの内容は、各トリガーの *Operational data* フィールドで設定できます。このフィールドには、マクロを含む任意の文字列を指定できますが、特に重要なのは {ITEM.LASTVALUE<1-9>} マクロです。

このフィールド内の {ITEM.LASTVALUE<1-9>} は、常にトリガー式内のアイテムの最新値に展開されます。このフィールド内の {ITEM.VALUE<1-9>} は、トリガー状態が変化した時点のアイテム値（つまり、障害への変化、OK への変化、ユーザーによる手動クローズ、または相関関係によるクローズ）に展開されます。

問題を手動でクローズしても新しい値は生成されないため、{ITEM.LASTVALUE<1-9>} または {ITEM.VALUE<1-9>} の展開結果には、引き続き障害発生時点の値が表示されることに注意してください。

{ITEM.LASTVALUE<1-9>} または {ITEM.VALUE<1-9>} は、最新の履歴値が *Max history display period* より前に収集されたものである場合、*UNKNOWN* に展開されます（管理 > 一般 を参照）。

負の障害期間

実際には、一般的ないくつかの状況において、障害の解決時刻が障害の発生時刻よりも前になる場合、つまり負の障害期間が発生する可能性があります。例えば、次のような場合です。

- あるホストがプロキシによって監視されており、ネットワークエラーが発生して、しばらくの間プロキシからデータを受信できなくなると、サーバーによって nodata(/host/key) トリガーが発生します。接続が復旧すると、サーバーは過去の時刻を持つアイテムデータをプロキシから受信します。その後、nodata(/host/key) の障害は解決され、負の障害期間を持つことになります。
- 障害イベントを解決するアイテムデータが Zabbix sender によって送信され、そのタイムスタンプが障害の発生時刻よりも前である場合も、負の障害期間が表示されます。

Note:
負の障害期間は、特定のトリガーのSLA calculation やAvailability report にいかなる影響も与えません。障害時間を短縮することも延長することもありません。

一括編集オプション

リストの下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 一括更新 - 障害の更新画面に移動して、選択した障害を更新します

このオプションを使用するには、対象の障害の前にあるチェックボックスをオンにしてから、一括更新ボタンをクリックします。

ボタン

右側のボタンでは、次のオプションを利用できます。

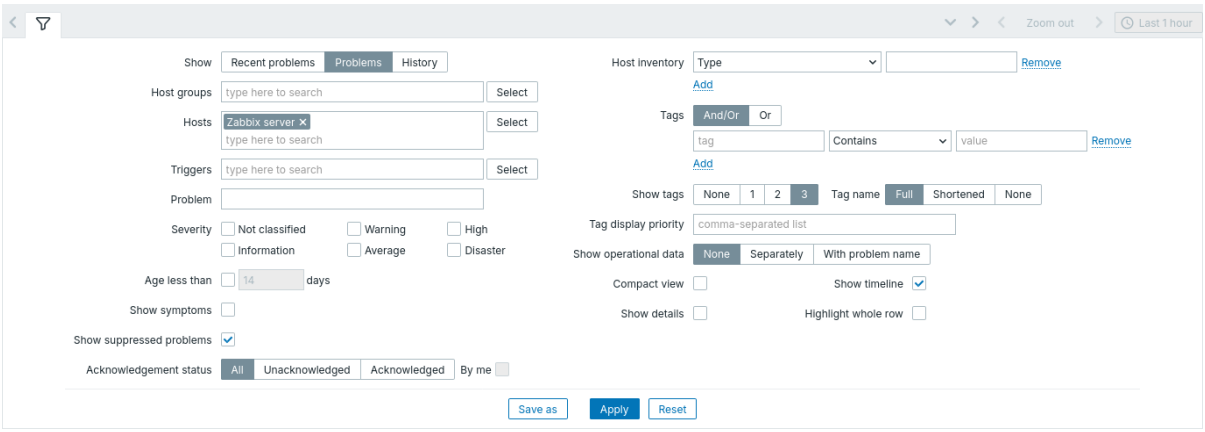
Export to CSV	すべてのページの内容を CSV ファイルにエクスポートします。

表示モードのボタンはすべてのセクションで共通のため、Monitoring ページで説明しています。

フィルターの使用

フィルターを使用すると、関心のある障害のみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルターはテーブルの上部にあります。よく使うフィルター設定はタブとして保存でき、フィルターの上にあるタブをクリックしてすばやくアクセスできます。



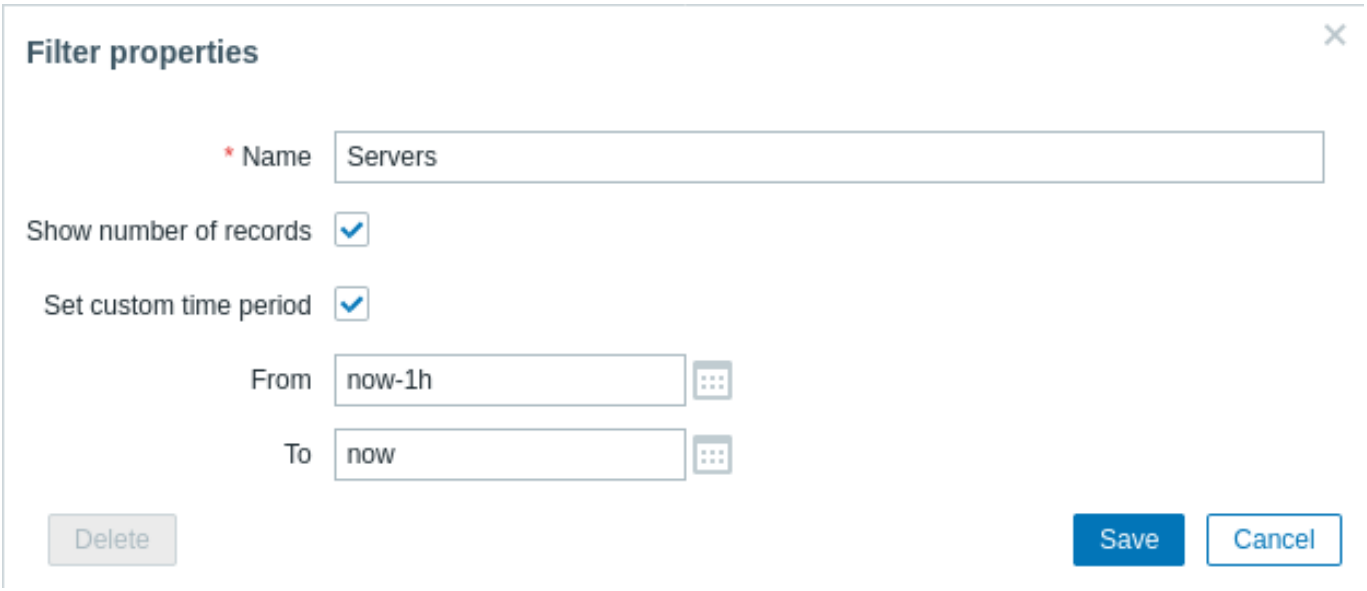
Parameter	Description
表示	障害ステータスでフィルタリングします: 最近の障害 - 未解決の障害と最近解決された障害を表示します（デフォルト） 障害 - 未解決の障害を表示します 履歴 - すべてのイベントの履歴を表示します
ホストグループ	1 つ以上のホストグループでフィルタリングします。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。
ホスト	1 つ以上のホストでフィルタリングします。
トリガー	1 つ以上のトリガーでフィルタリングします。
障害	障害名でフィルタリングします。
深刻度	トリガー（障害）の深刻度でフィルタリングします。
経過時間が次より短い	障害の発生からの経過時間でフィルタリングします。
症状を表示	チェックボックスをオンにすると、症状として分類された障害を独立した行に表示します。
抑制された障害を表示	チェックボックスをオンにすると、ホストのメンテナンスまたは個別の障害抑制によって通常は抑制されて表示されない障害を表示します。
確認ステータス	すべての障害、未確認の障害のみ、または確認済みの障害のみを表示するようフィルタリングします。追加のチェックボックスをオンにすると、自分が一度でも確認した障害を除外できます。
ホストインベントリ	インベントリのタイプと値でフィルタリングします。
タグ	イベントタグの名前と値でフィルタリングします。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件では複数の演算子を使用できます: 存在する - 指定したタグ名を含めます 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別） 含む - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない） 存在しない - 指定したタグ名を除外します 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別） 含まない - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない） 条件には 2 つの計算タイプがあります: **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です フィルタリング時には、ここで指定したタグが障害とともに最初に表示されます。ただし、タグ表示優先度（以下参照）リストで上書きされる場合を除きます。

Parameter	Description
タグを表示	表示するタグ数を選択します: なし - 監視 > 障害に タグ列を表示しません **1** - タグ列に 1 つのタグを表示します **2** - タグ列に 2 つのタグを表示します **3** - タグ列に 3 つのタグを表示します 障害のすべてのタグを確認するには、三点アイコンにマウスカーソルを合わせてください。
タグ名	タグ名の表示モードを選択します: 完全 - タグ名と値を完全な形で表示します 短縮 - タグ名は 3 文字に短縮され、タグ値は完全な形で表示されます なし - タグ値のみを表示し、名前は表示しません
タグ表示優先度	障害に対するタグ表示の優先度を、タグのカンマ区切りリストとして入力します（例: `customer,scope,component`）。使用できるのはタグ名のみで、値は指定しません。このリストのタグは、アルファベット順の自然な並びより優先して常に先頭に表示されます。
運用データを表示	運用データの表示モードを選択します: なし - 運用データを表示しません 別表示 - 運用データを別の列に表示します 障害名とともに表示 - 運用データを括弧付きで障害名に追加します
コンパクト表示	チェックボックスをオンにすると、コンパクト表示を有効にします。
詳細を表示	チェックボックスをオンにすると、障害の基になるトリガー式を表示します。コンパクト表示チェックボックスがオンの場合は無効です。
タイムラインを表示	チェックボックスをオンにすると、視覚的なタイムラインとグループ化を表示します。コンパクト表示チェックボックスがオンの場合は無効です。
行全体を強調表示	チェックボックスをオンにすると、未解決の障害について行全体を強調表示します。強調表示には障害の深刻度の色が使用されます。 行全体を強調表示は高コントラストテーマでは使用できません。 Zabbix 7.4.3 より前のバージョンでは、このオプションは標準の青テーマおよびダークテーマで コンパクト表示チェックボックスがオンの場合にのみサポートされます。

お気に入りフィルターのタブ

よく使用するフィルターパラメータのセットは、タブとして保存できます。

新しいフィルターパラメータのセットを保存するには、メインタブを開いてフィルター設定を構成し、*Save as* ボタンを押します。新しいポップアップウィンドウで、*Filter properties* を定義します。

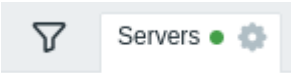


パラメータ	説明
Name	タブ一覧に表示するフィルター名。
Show number of records	タブ名の横に問題の件数を表示したい場合はチェックします。

パラメータ	説明
Override time period selector	このフィルターセットに対して特定のデフォルト期間を設定するにはチェックします。設定すると、このタブの期間はフィルター設定を更新することによってのみ変更できます。カスタム期間が設定されていないタブでは、右上隅の期間セレクターボタンを押して期間を変更できます（ボタン名は選択した期間に応じて異なります: This week、Last 30 minutes、Yesterday など）。 このオプションは、*Monitoring* > *Problems* のフィルターでのみ使用できます。
From/To	期間 の開始と終了を、絶対時間構文 (`Y-m-d H:i:s`) または相対時間構文 (`now-1d`) で指定します。 *Set custom time period* がチェックされている場合に使用できます。

保存すると、フィルターは名前付きのフィルタータブとして作成され、すぐに有効になります。

既存のフィルターのフィルタープロパティを編集するには、アクティブなタブ名の横にある歯車アイコンを押します。



注意:

- フィルター領域を非表示にするには、現在のタブ名をクリックします。アクティブなタブ名をもう一度クリックすると、フィルター領域が再び開きます。
- キーボード操作に対応しています: 矢印キーでタブを切り替え、*Enter* を押して開きます。
- フィルターの上にある左/右ボタンを使用して、保存済みフィルターを切り替えることができます。または、下向きボタンを押すと、保存済みフィルターがすべて表示されたドロップダウンメニューが開き、必要なものをクリックできます。
- フィルタータブは、ドラッグ＆ドロップで並べ替えることができます。
- 保存済みフィルターの設定が変更されているが、まだ保存されていない場合は、フィルター名の後ろに緑色の点が表示されます。新しい設定に従ってフィルターを更新するには、*Save as* ボタンの代わりに表示される *Update* ボタンをクリックします。
- 現在のフィルター設定はユーザープロファイルに記憶されます。ユーザーが再度ページを開いたときも、フィルター設定は同じ状態のままです。

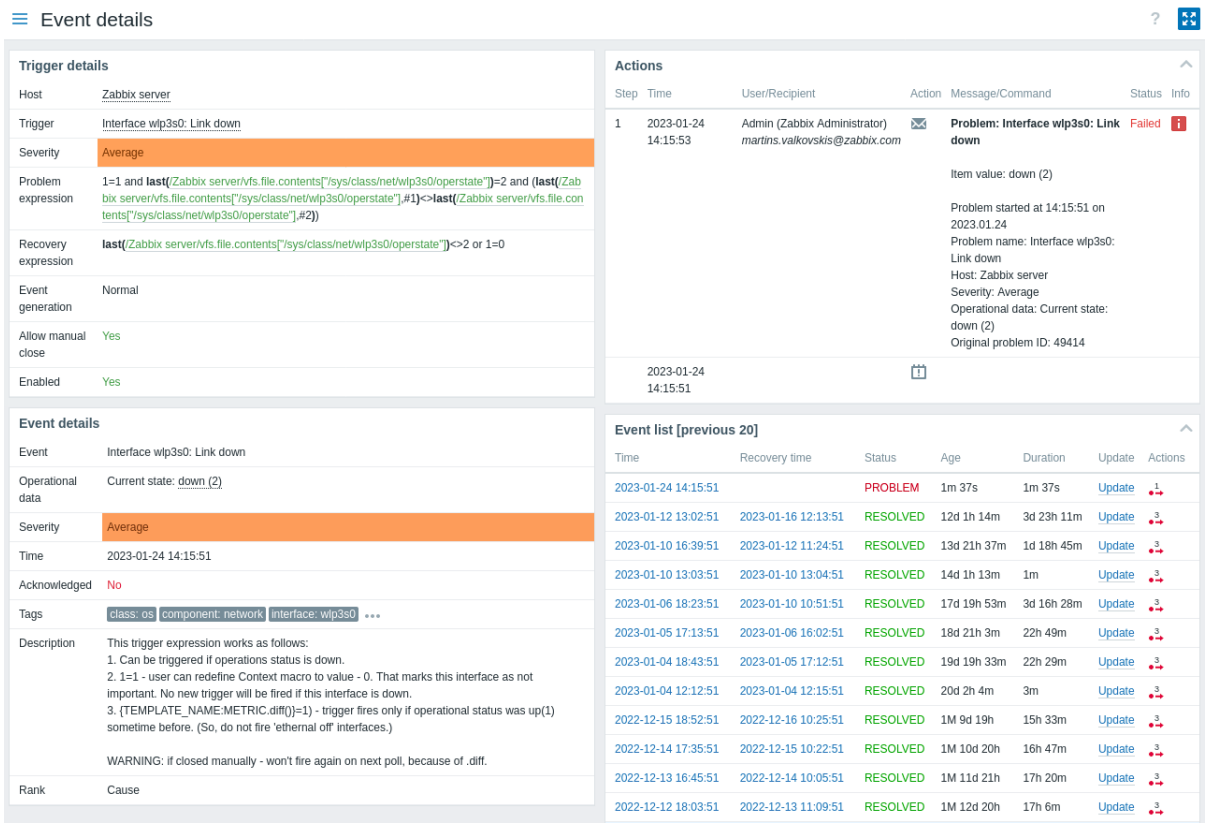
Note:
フィルターを共有するには、アクティブなフィルターの URL をコピーして他のユーザーに送信します。この URL を開くと、他のユーザーはこのパラメータセットを自分の Zabbix アカウントに永続的なフィルターとして保存できます。
あわせて参照: ページパラメータ。

フィルターのボタン

Apply	指定されたフィルタリングを（保存せずに）適用します
Reset	現在のフィルターをリセットし、現在のタブに保存されたパラメーターに戻ります。メインタブの場合はフィルターがクリアされます。
Save as	現在のフィルタリングパラメーターを新しいタブに保存します。メインタブでのみ使用できます。
Update	タブに保存されているパラメータを現在指定されているパラメータに置き換えます。メインタブでは使用できません。

詳細の表示

Monitoring > *Problems* の障害の開始時刻と復旧時刻はリンクになっています。これらをクリックすると、イベントの詳細情報が表示されます。



Update problem 画面 を使用して障害イベントの重大度が更新されている場合、トリガーと障害イベントで障害の重大度が異なることがある点に注意してください。

アクションリストでは、アクティビティの種類を示すために次のアイコンが使用されます。

- - 障害イベントが生成されたことを示します。
- - メッセージが送信されたことを示します。
- - 障害イベントが確認済みであることを示します。
- - 障害イベントが未確認であることを示します。
- - コメントが追加されたことを示します。
- - 障害の重大度が引き上げられたことを示します（例: Information > Warning）。
- - 障害の重大度が引き下げられたことを示します（例: Warning > Information）。
- - 障害の重大度が変更されたものの、元のレベルに戻ったことを示します（例: Warning > Information > Warning）。
- - リモートコマンドが実行されたことを示します。
- - 障害イベントが復旧したことを示します。
- - 障害が手動でクローズされたことを示します。
- - 障害が抑制されたことを示します。
- - 障害の抑制が解除されたことを示します。
- - 障害が症状障害に変換されたことを示します。
- - 障害が原因障害に変換されたことを示します。

1 原因と症状の障害

概要

デフォルトでは、すべての新しい障害は原因障害として分類されます。特定の障害を、原因障害の症状障害として手動で再分類する

ことができます。

例えば、停電が、あるホストに到達できない、またはあるサービスが停止している実際の根本原因である場合があります。この場合、「ホストに到達できない」および「サービスが停止している」障害は、原因障害である「停電」の症状障害として分類する必要があります。

原因と症状の階層は 2 レベルのみをサポートします。すでに症状である障害には、「下位」の症状障害を割り当てることはできません。症状障害に対して症状として割り当てられた障害は、すべて同じ原因障害の症状になります。

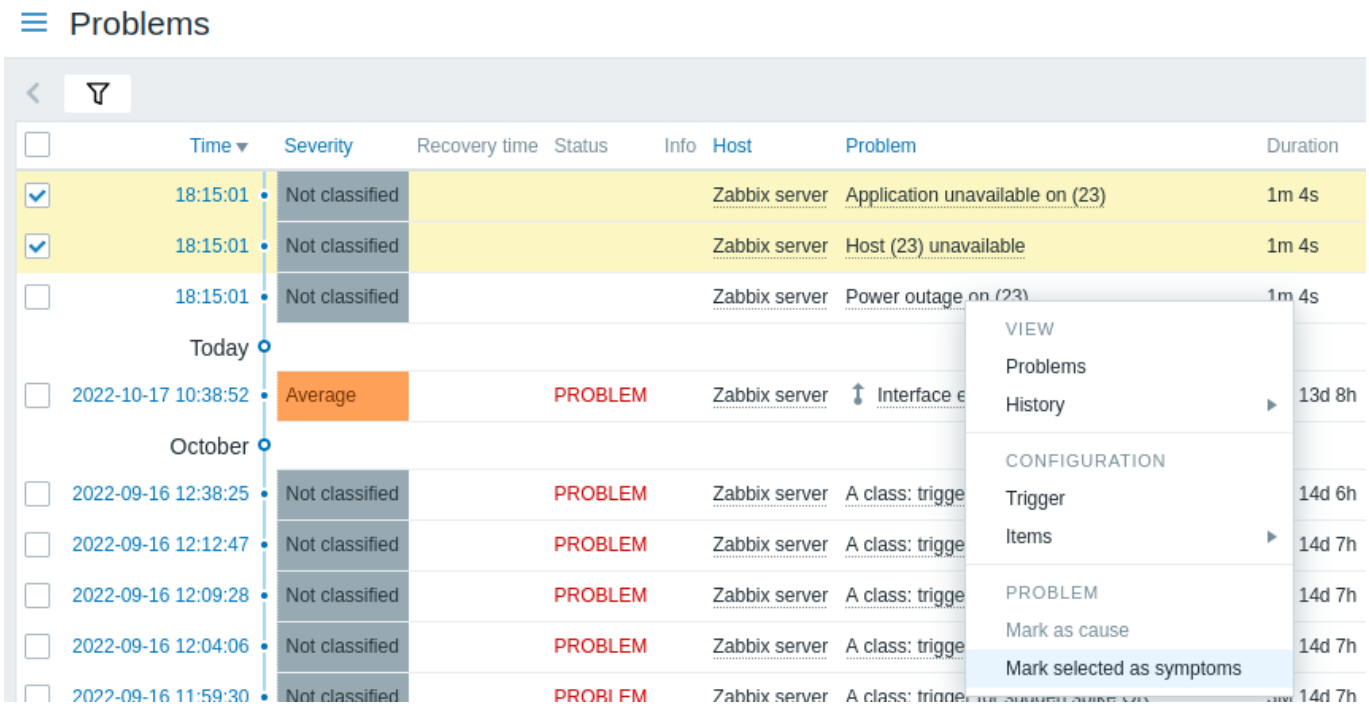
マップ、深刻度ごとの障害や 障害ホストなどのダッシュボードウィジェットなどでは、障害の合計数には原因障害のみがカウントされます。ただし、障害のランク付けはサービスには影響しません。

症状障害は、1 つの原因障害にのみリンクできます。原因障害が解決またはクローズされても、症状障害は自動的には解決されません。

設定

問題を症状問題として再分類するには、まず障害の一覧で対象を選択します。1 件または複数の問題を選択できます。

次に、原因問題に移動し、そのコンテキストメニューで 選択項目を症状としてマークオプションをクリックします。



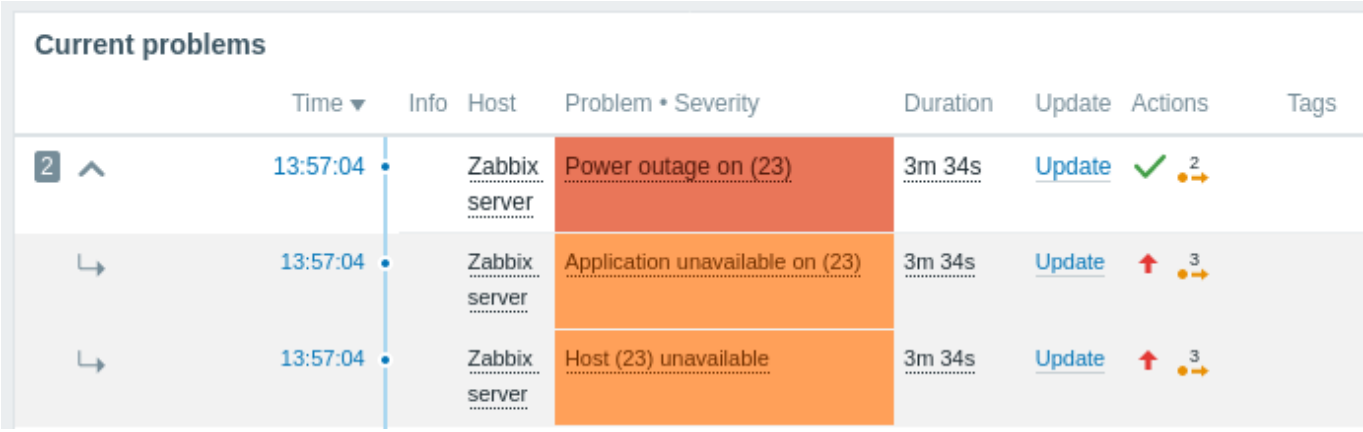
その後、選択した問題は、原因問題の症状問題としてサーバーによって更新されます。

問題のステータスが更新されている間は、次のいずれかの方法で表示されます。

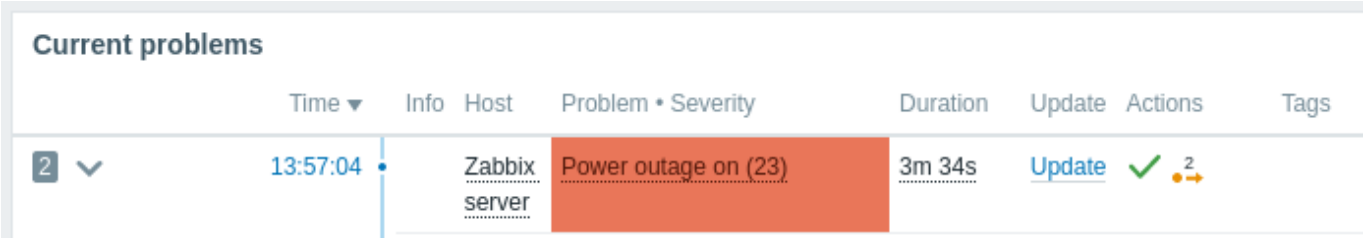
- Status 列に点滅する「UPDATING」ステータスが表示されます。
- Info 列に点滅する または アイコンが表示されます。(これは、フィルターで *Problems* のみが選択されており、Status 列が表示されていない場合に有効です。)

表示

症状の障害は、*Monitoring -> Problems*（および *Problems* ダッシュボードウィジェット）で原因障害の下に表示され、それに応じてマークされます。具体的には、アイコン、より小さいフォント、異なる背景で表示されます。



折りたたみ表示では、原因障害のみが表示されます。症状の障害が存在することは、行の先頭にある数字と、表示を展開するためのアイコンで示されます。



また、症状の障害を通常のフォントで個別の行に追加表示することもできます。その場合は、フィルター設定またはウィジェット設定で *Show symptoms* を選択します。

原因の障害への戻し

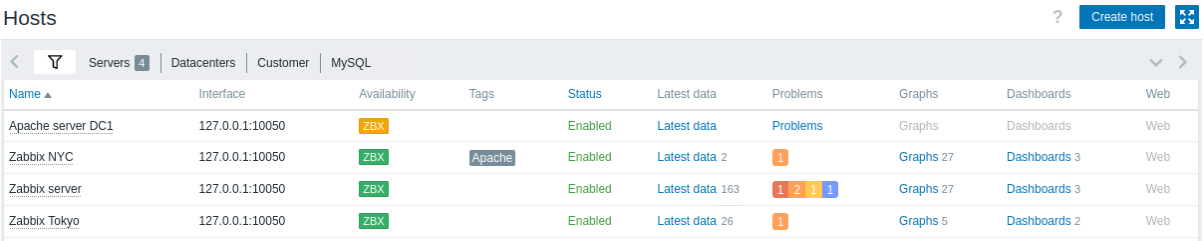
症状の障害は、原因の障害に戻すことができます。これを行うには、次のいずれかを実行します。

- 症状の障害のコンテキストメニューで、*Mark as cause* オプションをクリックします。
- 障害の更新画面で *Convert to cause* オプションを選択し、*Update* をクリックします（このオプションは、複数の障害が選択されている場合にも機能します）。

2 ホスト

概要

監視 → ホストセクションには、監視対象のホストの完全な一覧が表示されます。ここでは、ホストインターフェース、可用性、タグ、現在の障害、ステータス（有効/無効）に関する詳細情報に加え、ホストの最新データ、障害履歴、グラフ、ダッシュボード、Web シナリオへ簡単に移動するためのリンクを確認できます。



Column	Description
Name	表示用のホスト名です。名前をクリックすると、ホストメニューが表示されます。 名前の後ろにあるオレンジ色のレンチアイコン は、このホストがメンテナンス中であることを示します。 列ヘッダーをクリックすると、ホストを名前の昇順または降順で並べ替えることができます。
Interface	ホストのメインインターフェースが表示されます。

Column	Description
Availability	設定されているインターフェースごとのホストの可用性が表示されます。 可用性アイコンは、Zabbix サーバー上でのホストインターフェースの現在のステータスを表します。そのため、Web インターフェースでホストを無効にした場合、その可用性は Zabbix サーバーが設定変更を同期した後に更新されます。同様に、ホストを有効にした場合、その可用性は Zabbix サーバーが設定変更を同期し、ホストをポーリングした後に更新されます。 可用性アイコンは、設定されているインターフェースタイプ（Agent、SNMP、IPMI、JMX）のみを表します。 アイコンにマウスオーバーすると、同じタイプのすべてのインターフェースの一覧と、その詳細、ステータス、エラーを表示するポップアップが表示されます。Agent インターフェースでは、ポップアップにインターフェース（パッシブ）とアクティブチェックが表示されます。ホストにアクティブチェックしかない場合、ホストに Agent インターフェースが設定されていなくても、Agent インターフェースアイコンが表示されます。 インターフェースを持たないホストでは、この列は空になります。 単一のホストインターフェースのステータスは、そのインターフェースを使用する有効なアイテムとホストとの接続によって決まります。ステータスは次のいずれかになります。 利用可能 - ホストへの接続に成功しました。 利用不可 - ホストへの接続に失敗しました（タイムアウト、ファイアウォールの問題など）。 不明 - ホストへの接続がまだ試行されていないか、結果が不明です。 Zabbix サーバーがインターフェースのステータスをどのように判定するかの詳細については、不明なインターフェースステータス および到達不能/利用不可ホストインターフェースの設定 を参照してください。 単一タイプ（Agent、SNMP、IPMI、JMX）のすべてのホストインターフェースのステータスは、少なくとも 1 つの有効なアイテムで使用されているそれらのインターフェースによって決まります。ステータスはアイコンの色で示されます。 緑 - すべてのインターフェースが利用可能。 黄 - 少なくとも 1 つのインターフェースが利用不可で、かつ少なくとも 1 つが利用可能または不明。 赤 - すべてのインターフェースが利用不可。 灰色 - 少なくとも 1 つのインターフェースが不明で、利用不可のものはない。 アクティブチェックの可用性。ホスト上で少なくとも 1 つのアクティブチェックが有効になっている場合、アクティブチェックも上記の説明どおり Agent インターフェース全体の可用性に影響します。アクティブチェックの可用性を判定するために、ハートビートメッセージがエージェントのアクティブチェック用スレッドで送信されます。ハートビートメッセージの頻度は、Zabbix のエージェント またはagent 2 の設定にある HeartbeatFrequency パラメータで制御されます（デフォルトは 60 秒、範囲は 0～3600）。アクティブチェックのハートビートが 2 x HeartbeatFrequency 秒より古い場合、アクティブチェックは利用不可と見なされます。 **注:** バージョン 6.2.x より前の Zabbix エージェントはアクティブチェックのハートビートを送信しないため、それらのホストの可用性は不明のままになります。
Tags	ホストおよびリンクされたすべてのテンプレートのタグです。マクロは未解決のまま表示されます。
Status	ホストのステータスです。有効または 無効。 列ヘッダーをクリックすると、ホストをステータスの昇順または降順で並べ替えることができます。
Latest data	リンクをクリックすると、そのホストから収集されたすべての最新データを表示する 監視 - 最新データページが開きます。 最新データを持つアイテム数は灰色で表示されます。
Problems	重大度ごとに分類された、ホストで現在オープン中の障害数です。四角の色は障害の重大度を示します。四角内の数字は、その重大度における障害数を意味します。 アイコンをクリックすると、現在のホストに対する 監視 - 障害ページが開きます。 ホストに障害がない場合は、このホストの 障害セクションへのリンクがテキストとして表示されます。 抑制された障害を含めるかどうかはフィルターで選択できます（デフォルトでは含まれません）。
Graphs	リンクをクリックすると、そのホストに設定されているグラフが表示されます。グラフ数は灰色で表示されます。 ホストにグラフがない場合、リンクは無効化され（灰色のテキスト）、数も表示されません。

Column	Description
Dashboards	リンクをクリックすると、そのホストに設定されているダッシュボードが表示されます。ダッシュボード数は灰色で表示されます。 ホストにダッシュボードがない場合、リンクは無効化され（灰色のテキスト）、数も表示されません。
Web	リンクをクリックすると、そのホストに設定されている Web シナリオが表示されます。Web シナリオ数は灰色で表示されます。 ホストに Web シナリオがない場合、リンクは無効化され（灰色のテキスト）、数も表示されません。

ボタン

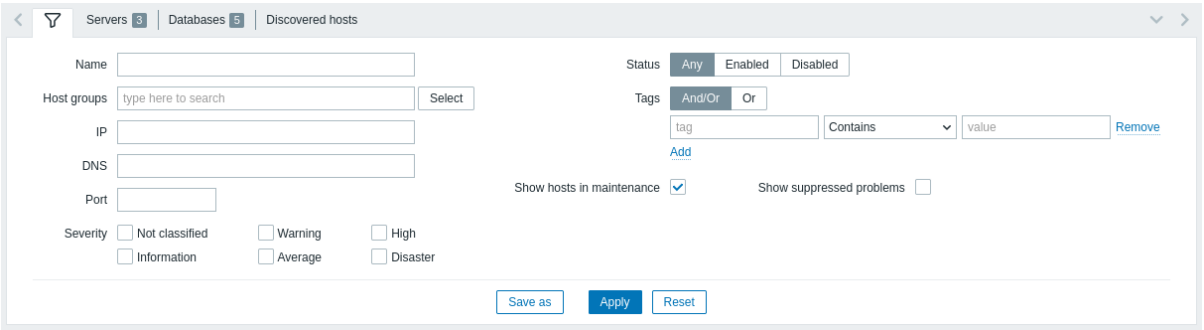
ホストの作成では、新しいホストを作成できます。このボタンは、管理者およびスーパー管理者ユーザーのみが使用できます。

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

フィルターの使用

フィルターを使用すると、関心のあるホストのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

フィルターはテーブルの上にあります。ホストは、名前、ホストグループ、IP または DNS、インターフェースのポート、タグ、障害の深刻度、ステータス（有効/無効/すべて）で絞り込むことができます。また、現在メンテナンス中のホストや、抑制された障害を表示するかどうかも選択できます。



パラメータ	説明
Name	表示用のホスト名で絞り込みます。
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。
IP	IP アドレスで絞り込みます。
DNS	DNS 名で絞り込みます。
Port	ポート番号で絞り込みます。
Severity	障害の深刻度で絞り込みます。デフォルトでは、すべての深刻度の障害が表示されます。障害は、抑制されていない場合に表示されます。
Status	ホストのステータスで絞り込みます。
Tags	ホストタグ名と値で絞り込みます。ホストは、ホストレベルのタグだけでなく、ネストされたテンプレートを含む、リンクされたすべてのテンプレートのタグでも絞り込めます。 特定のタグおよびタグ値を含めることも除外することも可能です。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです。 **Exists** - 指定したタグ名を含める **Equals** - 指定したタグ名と値を含める（大文字と小文字を区別） **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含める（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外する **Does not equal** - 指定したタグ名と値を除外する（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外する（部分一致、大文字と小文字を区別しない） 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
Show hosts in maintenance	チェックボックスをオンにすると、メンテナンス中のホストを表示します（デフォルトで表示）。

パラメータ	説明
Show suppressed problems	チェックボックスをオンにすると、ホストのメンテナンスまたは単一の障害抑制により、通常は抑制されて表示されない障害を表示します。

フィルターの保存

お気に入りのフィルター設定はタブとして保存でき、
フィルターの上にある対応するタブをクリックしてすばやくアクセスできます。

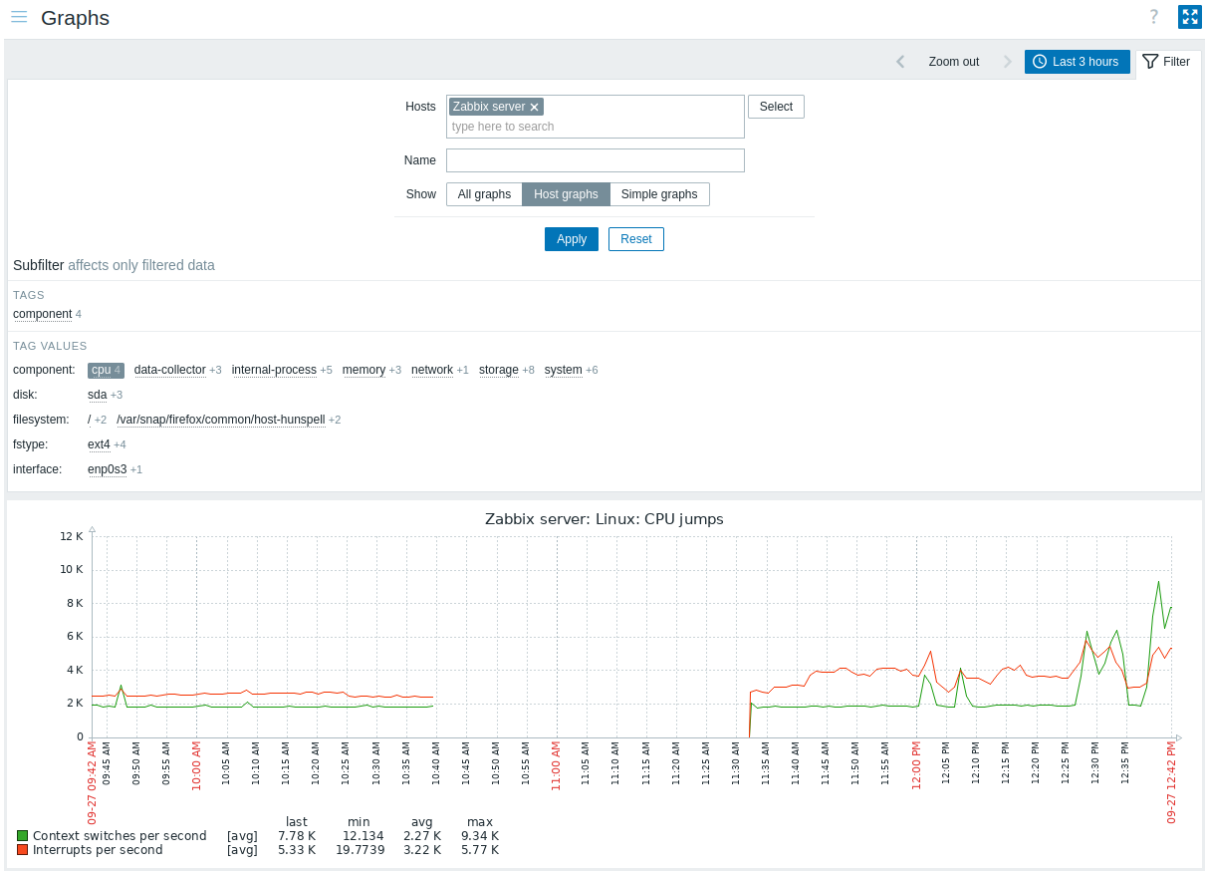
詳細は、フィルターの保存を参照してください。

1 グラフ

概要

ホストのグラフは、監視 → ホストから該当するホストの グラフをクリックしてアクセスできます。

ホストに設定されている任意のカスタムグラフと、任意のシンプルグラフを表示できます。



グラフは次の順に並べ替えられます。

- グラフ名（カスタムグラフ）
- アイテム名（シンプルグラフ）

無効化されたホストのグラフにもアクセスできます。

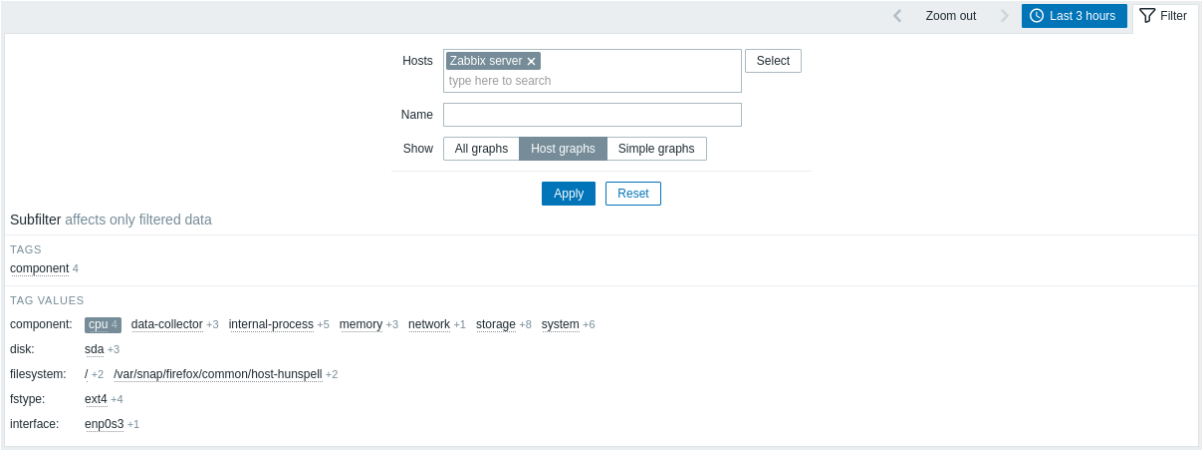
期間セレクター

グラフの上にある期間セレクターに注意してください。
これにより、よく必要とされる期間をマウスで 1 回クリックするだけで選択できます。

参照: 期間セレクター

フィルターの使用

特定のグラフを表示するには、フィルターでそのグラフを選択します。フィルターでは、ホスト、グラフ名、および 表示オプション（すべて／ホストのグラフ／シンプルグラフ）を指定できます。



フィルターでホストが選択されていない場合、グラフは表示されません。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターによる絞り込みをさらに調整できます。

サブフィルターには、関連するグラフにすばやくアクセスするためのクリック可能なリンクが含まれています。グラフは、タグ名または値という共通のエンティティによって関連付けられています。タグ名/値をクリックすると、灰色の背景で強調表示され、グラフは即座に絞り込まれます（メインフィルターで *Apply* をクリックする必要はありません）。別のタグ名/値をクリックすると、絞り込み結果に追加されます。同じタグ名/値をもう一度クリックすると、絞り込みが解除されます。

サブフィルターは、フィルタリングされたデータに基づいて生成されます。このデータは 1000 レコードに制限されています。サブフィルターにさらに多くのレコードを表示したい場合は、*Limit for search and filter results* パラメータの値を増やす必要があります (*Administration -> General -> GUI* 内)。

メインフィルターとは異なり、サブフィルターはテーブルの更新要求ごとに更新されるため、利用可能な絞り込みオプションとその件数について常に最新の情報を取得できます。

表示されるエンティティの数は、水平方向では 100 件までに制限されています。これを超える場合は、末尾に三点アイコンが表示されますが、クリックはできません。縦方向のリスト（タグとその値など）は 20 エントリまでに制限されています。これを超える場合は、三点アイコンが表示されますが、クリックはできません。

各クリック可能なエンティティの横に表示される数字は、メインフィルターの結果内でそのエンティティが持つグラフ数を示します。

1 つのエンティティが選択されると、他の利用可能なエンティティに表示される数字にはプラス記号が付き、現在の選択に追加できるグラフ数を示します。

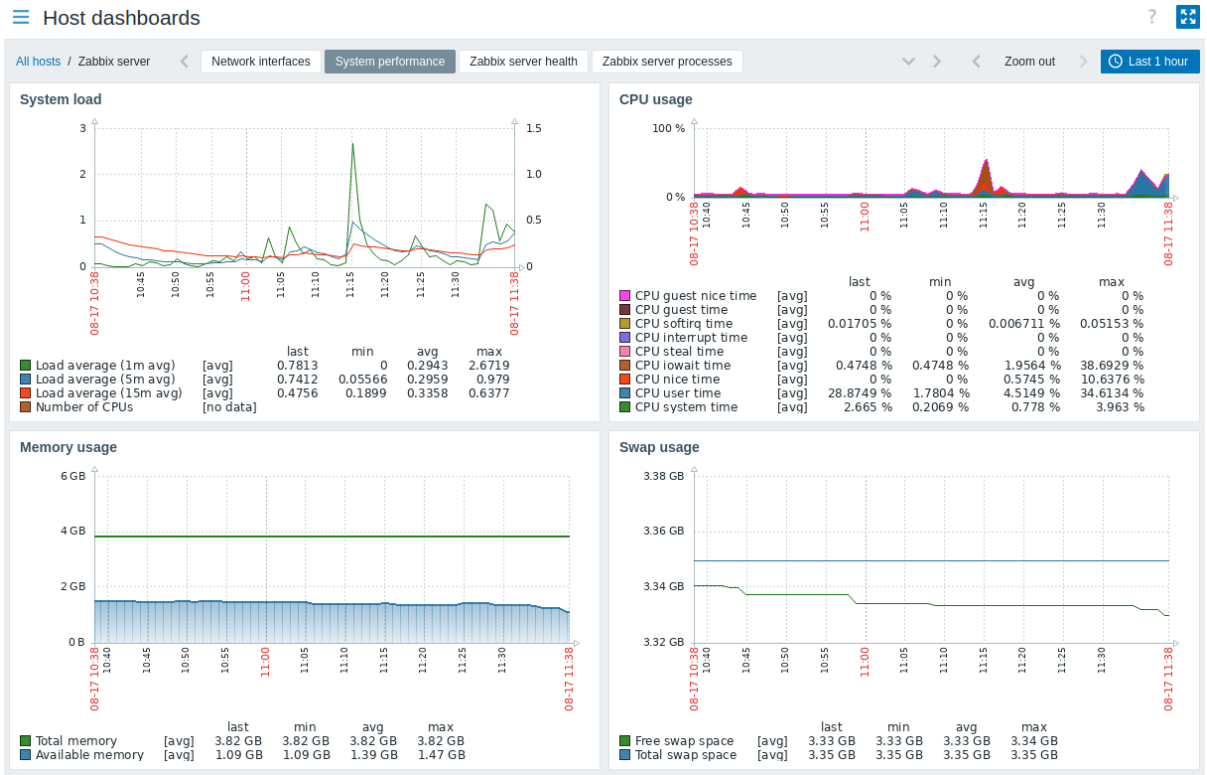
ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoring ページで説明しています。

2 ホストダッシュボード

概要

ホストダッシュボードはグローバルダッシュボードと似ていますが、ホストダッシュボードには所有者がなく、選択したホストのデータのみが表示されます。



ホストダッシュボードの表示中は、以下をクリックして設定済みのダッシュボードを切り替えることができます。

- ダッシュボードタブ
- ヘッダーの下にある  矢印ボタン
- ヘッダーの下にある  矢印ボタン。利用可能なホストダッシュボードの一覧全体が表示されます。

監視 → ホストセクションに切り替えるには、左上隅のヘッダーの下にある すべてのホストナビゲーションリンクをクリックします。

設定

ホストダッシュボードは、テンプレートレベルで設定します。テンプレートがホストにリンクされると、そのホスト用のホストダッシュボードが生成されます。ホストダッシュボードは、グローバルダッシュボード用に予約されているダッシュボードセクションでは設定できないことに注意してください。

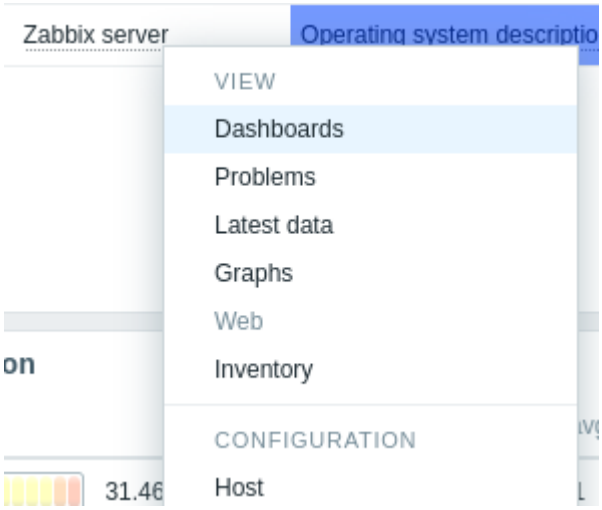
ホストダッシュボードのウィジェットも、更新間隔の変更を除き、テンプレートレベルでのみ設定できます。さらに、ホストダッシュボードのウィジェットは、同じテンプレート内の他のホストダッシュボードにのみコピーできます。グローバルダッシュボードのウィジェットはホストダッシュボードにコピーできないことに注意してください。

Note:
ホストダッシュボードは、Zabbix 5.2 より前はホストスクリーンと呼ばれていました。スクリーンを含む古いテンプレートをインポートする場合、スクリーンのインポートは無視されます。

アクセス

ホストダッシュボードには、次の方法でアクセスできます。

- グローバル検索でホスト名を検索した後（検索結果に表示される *Dashboards* リンクをクリックします）。
- インベントリ → ホストでホスト名をクリックした後（ホスト概要に表示される *Dashboards* リンクをクリックします）。
- ホストメニューから、*Dashboards* をクリックします。

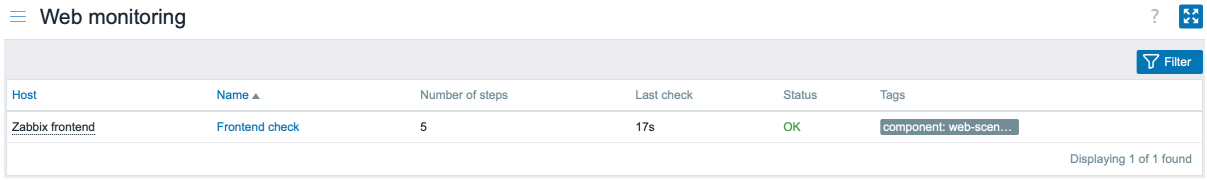


ホストダッシュボードは、グローバルダッシュボード用に予約されているダッシュボードセクションから直接アクセスすることはできない点に注意してください。

3 Web シナリオ

概要

ホストのWeb シナリオ情報には、監視データ → ホストでそれぞれのホストの Web をクリックしてアクセスできます。



ホスト名をクリックすると、ホストメニューが表示されます。無効になっているホストのデータにもアクセスできます。無効になっているホストの名前は赤で表示されます。

ページごとに表示されるシナリオの最大数は、ユーザープロファイル設定のページあたりの表示行数によって異なります。

デフォルトでは、過去 24 時間以内の値のみが表示されます。この制限は、最新データの大きなページの初期読み込み時間を改善することを目的として導入されました。この期間は、管理 → 一般設定 → *GUI*のヒストリの最大表示期間パラメーターの値を変更することで延長できます。

シナリオ名は、それに関するより詳細な統計へのリンクです。



フィルタの使用

このページには、選択したホストのすべての Web シナリオのリストが表示されます。監視データ → ホストページに戻らずに別のホストまたはホストグループの Web シナリオを表示するには、フィルターでそのホストまたはグループを選択します。タグに基づいてシナリオをフィルタリングすることもできます。

ボタン

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

3 最新データ

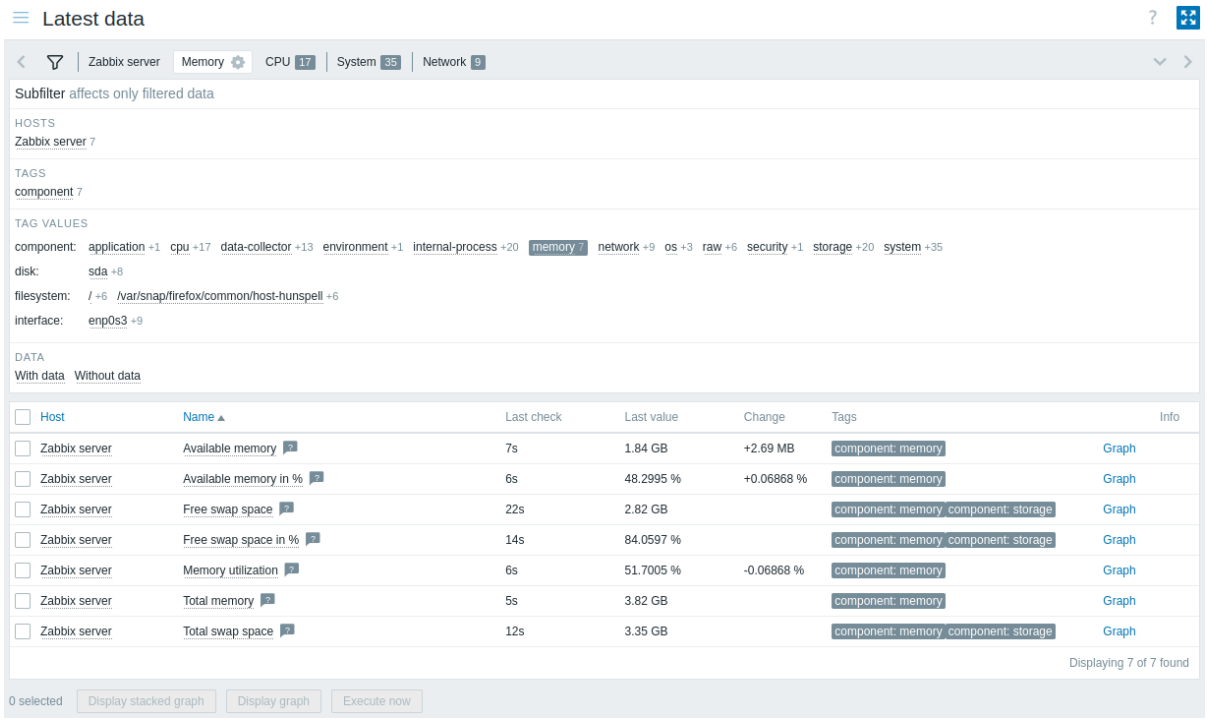
概要

監視 → 最新データセクションには、アイテムによって収集された最新の値が表示されます。

このセクションには、次の要素が含まれます。

- フィルター
- サブフィルター
- アイテム一覧

Note:
サブフィルターとアイテム一覧は、フィルターが設定されており、表示する結果がある場合にのみ表示されます。



列	説明
ホスト	アイテムが属するホストの名前です。 名前をクリックすると、ホストコンテキストメニューが表示されます。 ホストがメンテナンス中の場合、ホスト名の後にオレンジ色のレンチ アイコンが表示されます。 ホストが無効になっている場合、ホスト名は赤色で表示されます。無効なホストのデータ（グラフおよびアイテム値一覧を含む）にも、最新データセクションからアクセスできることに注意してください。
名前	アイテムの名前です。 名前をクリックすると、アイテムメニューが表示されます。 説明があるすべてのアイテムでは、アイテム名の横に疑問符 アイコンが表示されます。アイコンにマウスカーソルを合わせると、アイテムの説明を含むツールチップが表示されます。
最終チェック	最後にアイテムのチェックが行われてからの経過時間です。
最新の値	アイテムの最新の値です。 値は、単位変換および値のマッピングが適用された状態で表示されます。値にマウスカーソルを合わせると、生データ（8192 文字に切り詰められます）が表示されます。 バイナリ値は、実際の値の代わりにプレースホルダー（*binary value*）として表示されます。 デフォルトでは、過去 24 時間以内に受信した値のみが表示されます。この制限により、最新データの大きなページの初期読み込み時間が改善されます。期間を延長するには、管理 → 一般 → *GUI* にある *Max history display period* パラメータの値を更新してください。
変化	前回の値と最新の値との差分です。 更新間隔が 1 日以上のアイテムでは、変化量は（デフォルト設定では）表示されません。この場合、24 時間以上前に受信した最新の値もまったく表示されません。
タグ	アイテムに関連付けられたタグです。 アイテム一覧内のタグはクリック可能です。タグをクリックすると、サブフィルター でそのタグが有効になり、そのタグ（およびサブフィルターで以前に選択された他のタグ）を含むアイテムのみがアイテム一覧に表示されます。この方法でアイテムがフィルタリングされると、アイテム一覧内のタグはクリックできなくなることに注意してください。タグに基づく追加の変更（たとえば、タグの削除や他のフィルターの指定）は、サブフィルターで行う必要があります。
グラフ/履歴	アイテム値のシンプルグラフ/履歴へのリンクです。
情報	アイテムに関する追加情報です。 アイテムにエラーがある場合（たとえば、未サポートになった場合）、情報 アイコンが表示されます。詳細を確認するには、アイコンにマウスカーソルを合わせてください。

ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoringページで説明しています。

一括操作

リストの下にあるボタンでは、選択した 1 つまたは複数のアイテムに対して一括操作を実行できます。

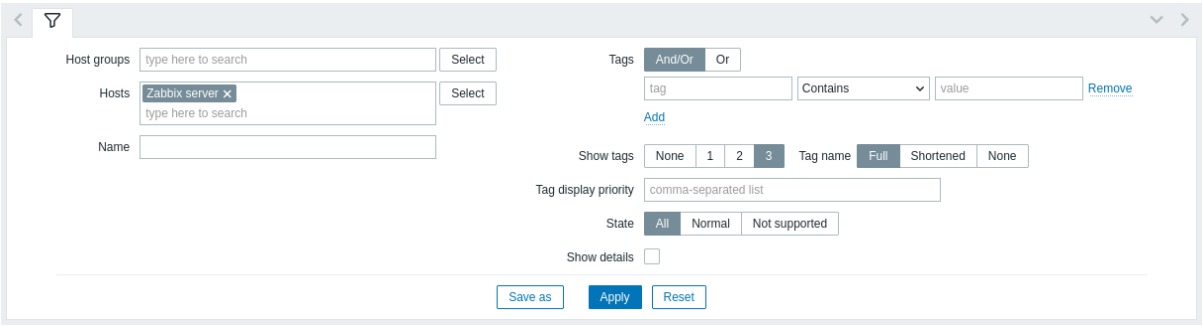
- 積み上げグラフを表示 - 積み上げアドホックグラフ を表示します。
- グラフを表示 - シンプルなアドホックグラフ を表示します。
- 今すぐ実行 - 新しいアイテム値のチェックを直ちに実行します。パッシブチェックでのみサポートされます（詳細はこちら を参照）。このオプションは、読み書きアクセス権を持つホストでのみ利用できます。読み取り専用権限を持つホストでこのオプションにアクセスできるかどうかは、読み取り専用ホストで「今すぐ実行」を呼び出すというユーザーの役割 オプションに依存します。

これらのオプションを使用するには、対象のアイテムの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるアイテムのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルター  アイコンは、アイテム一覧とサブフィルターの上部にあります。これをクリックすると、フィルターが展開されます。



フィルターを使用すると、ホストグループ、ホスト、アイテム名、タグ、状態、その他の設定によってアイテム一覧を絞り込むことができます。フィルターで親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。タグによるフィルタリングの詳細については、*Monitoring* → *Problems* を参照してください。

Show details フィルターオプションを使用すると、アイテムに表示される情報を拡張できます。これを有効にすると、アイテムの更新間隔、履歴およびトレンドの設定、アイテムタイプ、アイテムのエラー（正常/未サポート）などの詳細が表示されます。

フィルターの保存

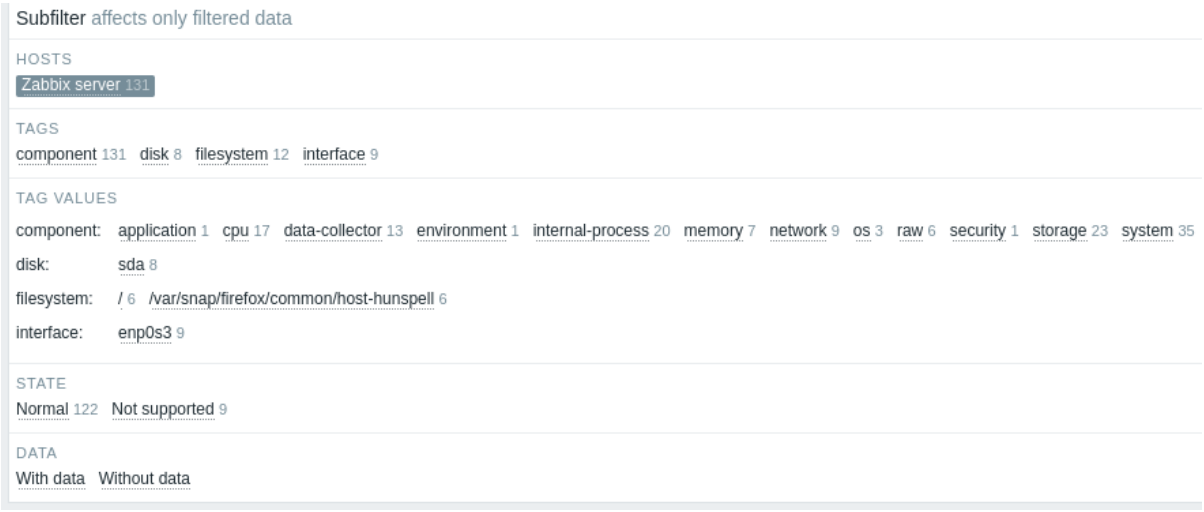
お気に入りのフィルター設定はタブとして保存でき、フィルターの上部にある対応するタブをクリックしてすばやくアクセスできます。

詳細については、フィルターの保存を参照してください。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターによる絞り込みをさらに調整できます。

サブフィルターには、関連するアイテムにすばやくアクセスするためのクリック可能なリンクが含まれています。アイテムは、共通のエンティティ（ホスト、タグ名または値、アイテムの状態、またはデータのステータス）によって関連付けられています。エンティティをクリックすると、そのエンティティは灰色の背景で強調表示され、アイテムは即座に絞り込まれます（メインフィルターで *Apply* をクリックする必要はありません）。別のエンティティをクリックすると、絞り込み結果に追加されます。同じエンティティをもう一度クリックすると、絞り込みが解除されます。



サブフィルターは、絞り込み済みデータに基づいて生成されます。このデータは 1000 レコードに制限されています。たとえば、20 台のホストがそれぞれ 100 個のアイテムを持っている場合（合計 2000 レコード）、サブフィルターにはホストの半分しか表示されません。サブフィルターでより多くのレコードを表示したい場合は、*Administration -> General -> GUI* にある *Limit for search and filter results* パラメータの値を増やす必要があります。

メインフィルターとは異なり、サブフィルターはテーブルの更新要求ごとに更新されるため、利用可能な絞り込みオプションとその件数の最新情報が常に反映されます。

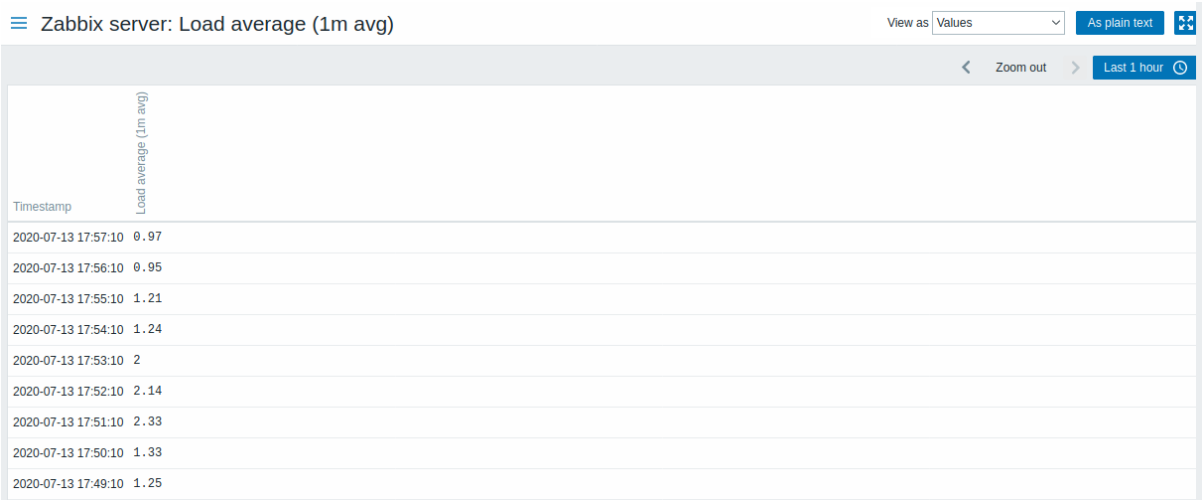
各エンティティグループ（ホスト、タグ、タグ値など）について、最大 10 行のエンティティが表示されます。エンティティがそれより多い場合は、リストの末尾にある三点アイコン をクリックすると、このリストを展開して最大 1000 件のエントリ（Web インターフェース definitions の SUBFILTER_VALUES_PER_GROUP の値）を表示できます。*Tag values* については、対応する値を含む最大 200 個のタグ名を表示するようにリストを展開できます。なお、一度完全に展開したリストは折りたたむことができません。

各クリック可能なエンティティの横に表示される数字は、そのエンティティにグループ化されたアイテム数を示します（メインフィルターの結果に基づきます）。エンティティをクリックすると、他の利用可能なエンティティの数字にはプラス記号が付いて表示され、現在の選択に追加できるアイテム数を示します。アイテムを持たないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

グラフと履歴

アイテム一覧の *Graph/History* 列には、次のリンクがあります。

- **History** - すべてのテキストアイテムで利用でき、以前のアイテム値の履歴を表示する一覧（*Values/500 latest values*）に移動します。
- **Graph** - すべての数値アイテムで利用でき、シンプルグラフに移動します。グラフの表示時には、右上のドロップダウンから *Values/500 latest values* に切り替えることもできます。



この一覧に表示される値は生の値であり、つまり、後処理は適用されません。

Note:
表示される値の総数は、*Administration → General → GUI* で設定する *Limit for search and filter results* パラメータの値によって定義されます。

4 マップ

概要

監視データ → マップセクションではネットワークマップを構成、管理、および表示できます。

このセクションを開くと最後にアクセスしたマップ、またはアクセスできるすべてのマップのリストが表示されます。

すべてのマップは、パブリックまたはプライベートのいずれかになります。パブリックマップはすべてのユーザーが利用できますが、プライベートマップは、所有者とマップが共有されているユーザーのみがアクセスできます。

マップ一覧



表示されるデータ:

列	説明
名前	マップの名前です。名前をクリックすると、マップを表示できます。
幅	マップの幅が表示されます。
高さ	マップの高さが表示されます。
アクション	2 つのアクションを利用できます: プロパティ - 一般的なマップのプロパティを設定します 編集 - マップ要素を追加するためのグリッドにアクセスします

新しいマップを設定するには、右上隅にある マップの作成ボタンをクリックします。YAML、XML、または JSON ファイルからマップをインポートするには、右上隅にある インポートボタンをクリックします。マップをインポートしたユーザーがその所有者として設定されます。

一覧の下にある 2 つのボタンでは、一括編集オプションを利用できます:

- エクスポート - マップを YAML、XML、または JSON ファイルにエクスポートします
- 削除 - マップを削除します

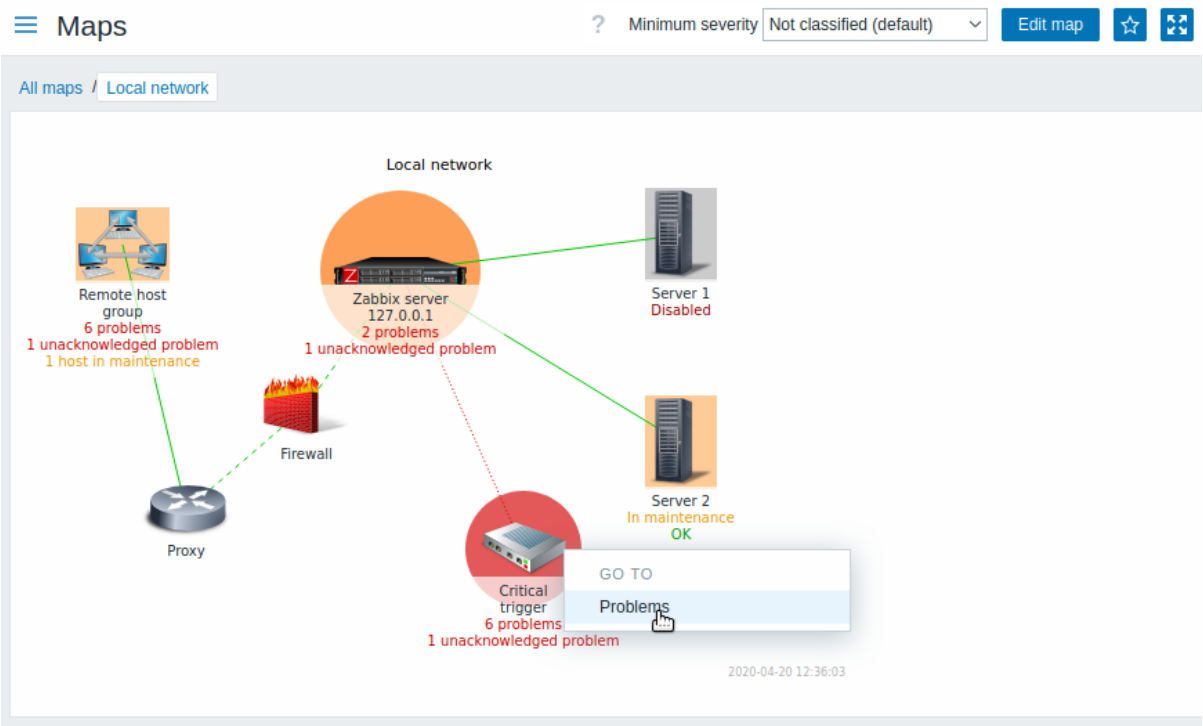
これらのオプションを使用するには、対象のマップの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、任意のマップのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

マップの表示

マップを表示するには、マップのリストで表示したいマップの名前をクリックします。



マップのタイトルバーのドロップダウンを使用して、表示する問題トリガーの最低の重大度レベルを選択できます。*default* としてマークされた重大度は、マップ設定で設定されたレベルです。マップにサブマップが含まれている場合は、サブマップに移動すると上位レベルのマップの深刻度が保持されます（ただし、分類されていない場合を除き、深刻度はサブマップに渡されません）。

アイコンの強調表示

マップ要素が障害ステータスの場合、丸い円で強調表示されます。円の塗りつぶし色は、障害の深刻度の色に対応します。選択した深刻度レベル以上の障害のみが、その要素とともに表示されます。すべての障害が確認済みの場合、円の周囲に太い緑色の枠線が表示されます。

さらに、以下のようになります。

- maintenance 中のホストは、オレンジ色の塗りつぶし四角形で強調表示されます。マップ要素がホストである場合、メンテナンスの強調表示は障害の深刻度の強調表示よりも優先されることに注意してください。
- 無効化された（監視されていない）ホストは、灰色の塗りつぶし四角形で強調表示されます。

強調表示は、マップの設定で *Icon highlighting* チェックボックスがオンになっている場合に表示されます。

最近の変更マーカー

要素の周りを内側に向けた赤い三角形は、最近のトリガーステータスの変更を示します。これは過去 30 分以内に発生したものです。これらの三角形は、マップ設定でトリガーステータス変更時に要素をマークするチェックボックスがマークされている場合に表示されます。

リンク

マップ要素をクリックすると、いくつかの利用可能なリンクが表示されたメニューが開きます。ホスト名をクリックすると、ホストメニューが表示されます。

ボタン

右側のボタンでは、以下の操作を実行できます。

Edit map	マップ内容の編集に移動します。
	マップをダッシュボードのお気に入りウィジェットに追加します。
	このマップはダッシュボードのお気に入りウィジェットに追加されています。クリックすると、お気に入りウィジェッ

すべてのセクションで共通の表示モードボタンについては、監視 ページを参照してください。

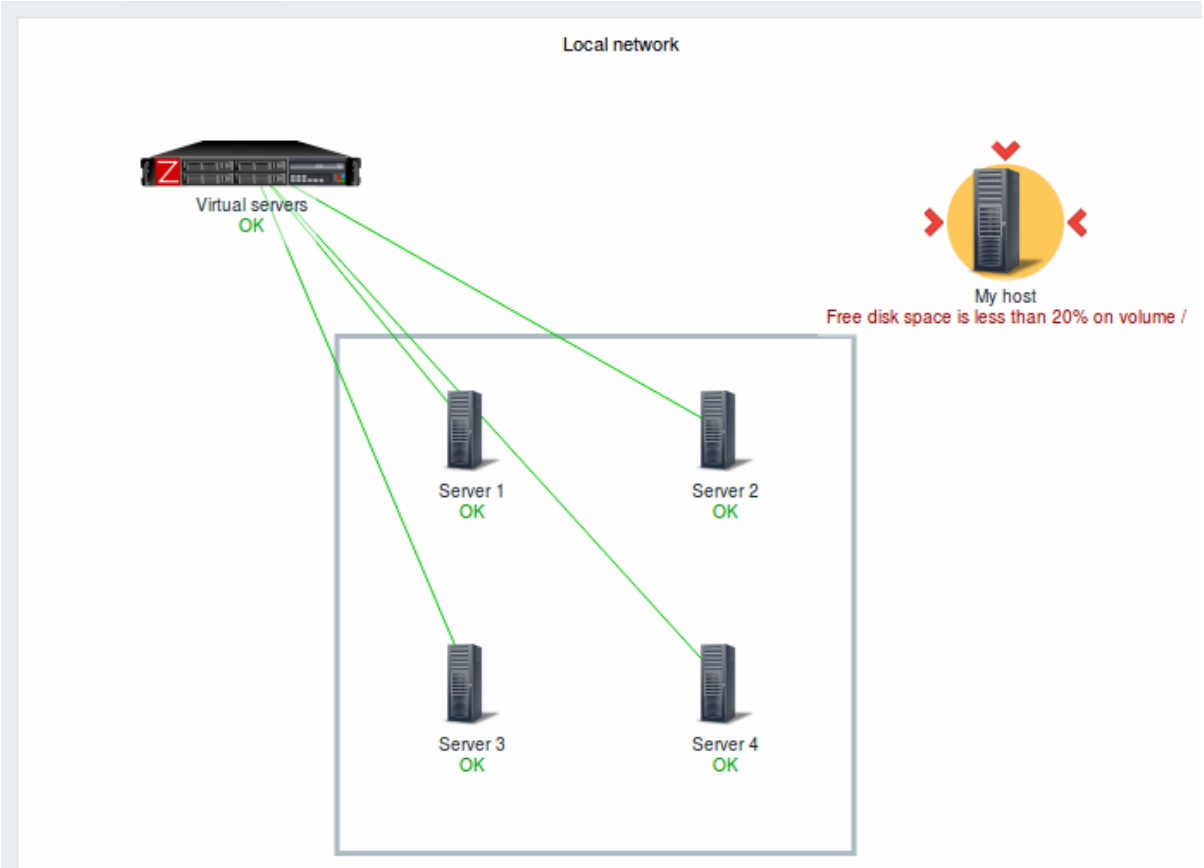
読みやすい要約地図

非表示の「aria-label」プロパティを使用すると、スクリーンリーダーで地図情報を読み取ることができます。一般的なマップの説明と個々の要素の説明の両方が、次の形式で利用できます。

- マップの説明：<Map name>, <* of * items in problem state>, <* problems in total>.
- 1 つの障害でで 1 つの要素を記述する：<Element type>, Status <Element status>, <Element name>, <Problem description>.
- 複数の障害がある 1 つの要素を記述する：<Element type>, Status <Element status>, <Element name>, <* problems>.
- 障害のない 1 つの要素を記述する：<Element type>, Status <Element status>, <Element name>.

たとえば、次のマップの場合以下のような説明が利用可能です。

'Local network, 1 of 6 elements in problem state, 1 problem in total. Host, Status problem, My host, Free



ネットワークマップの参照

ネットワークマップは sysmapid と mapname の両方の GET パラメータで参照できます。例えば、

http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network

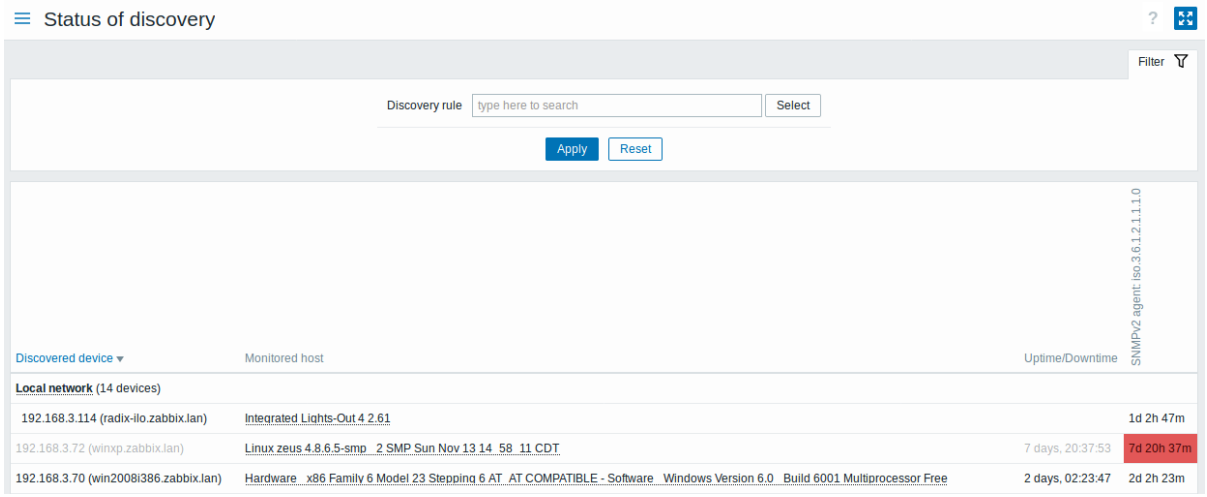
指定された名前のマップを開きます。(Local network)

sysmapid（マップ ID）と mapname（マップ名）の両方が指定されている場合、mapname の優先度が高くなります。

5 ディスカバリ

概要

監視 → ディスカバリセクションでは、ネットワークディスカバリ の結果が表示されます。検出されたデバイスは、ディスカバリルールごとに分類されます。



表示されるデータ:

Column	Description
検出されたデバイス	検出されたデバイスは、ディスカバリルールごとにグループ化されて一覧表示されます。ディスカバリルールをクリックすると、ディスカバリルールの設定フォームへのリンクを含むルールメニューが表示されます。
監視中のホスト	デバイスがすでに監視されている場合、この列にホスト名が表示されます。 ホスト名をクリックすると、ホストメニューが表示されます。
稼働時間/停止時間	前回のディスカバリ以降に、デバイスが検出されている、または見失われている期間がこの列に表示されます。
ディスカバリチェック	検出された各デバイスについて、個々のサービス（ディスカバリチェック）の状態が表示されます。赤いセルは、そのサービスが停止していることを示します。サービスの稼働時間または停止時間もセル内に含まれます。 この列は、少なくとも 1 つの検出されたデバイスでサービスが見つかった場合にのみ表示されます。

ボタン

すべてのセクションに共通の表示モードボタンについては監視データページで説明されています。

フィルタの使用

フィルタを使用して、任意のディスカバリルールのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルタで何も選択されていない場合、有効なすべてのディスカバリルールが表示されます。表示する特定のディスカバリルールを選択するには、フィルターにその名前を入力します。一致するすべての有効なディスカバリルールが選択のために一覧表示されます。複数のディスカバリルールを選択できます。

3 サービス

概要

サービスメニューは、Zabbix のサービスモニタリング機能です。

1 サービス

概要

このセクションでは、インフラストラクチャに基づいて Zabbix で設定されたサービス全体の大まかなステータスを確認できます。

サービスは複数レベルの他のサービスから成る階層構造を持つ場合があり、これらは「子」サービスと呼ばれ、サービス全体のステータスを構成する要素となります（サービス監視機能の概要も参照してください。）

サービスステータスの主なカテゴリは *OK* または *Problem* で、*Problem* ステータスは対応する障害の深刻度名と色で表されます。

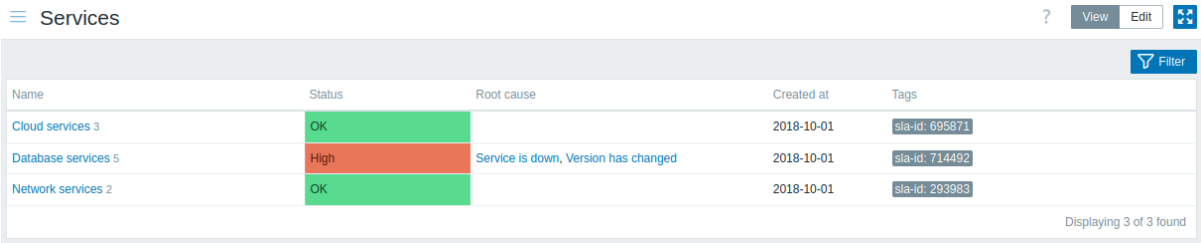
表示モードでは、サービスのステータスやその他の詳細を監視できますが、このセクションでは編集モードに切り替えることで、サービス階層を設定することもできます（サービスや子サービスの追加/編集）。

表示モードから編集モードへ（またはその逆へ）切り替えるには、右上隅にある該当のボタンをクリックします。

- View - サービスを表示
- Edit - サービスおよび子サービスを追加/編集

編集へのアクセスは、ユーザーの役割の設定に依存することに注意してください。

サービスの表示



既存のサービスの一覧が表示されます。

表示されるデータ:

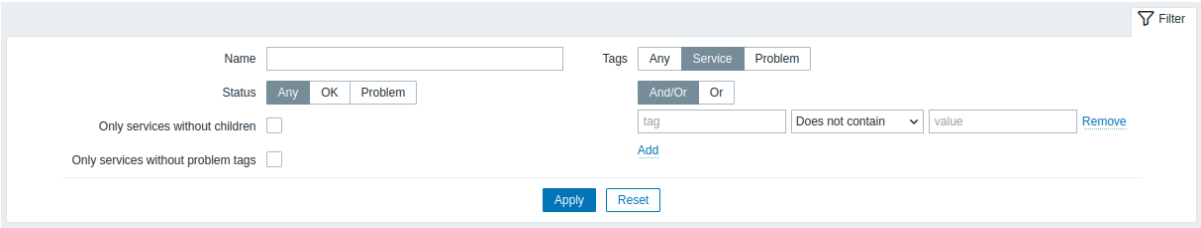
パラメータ	説明
名前	サービス名。 サービス名はサービスの詳細へのリンクです。 名前の後の数字は、そのサービスが持つ子サービスの数を示します。
ステータス	サービスのステータス: **OK** - 問題なし **<** 問題の色と深刻度 **>** - 問題とその深刻度を示します。複数の問題がある場合は、最も重大な問題の色と深刻度が表示されます。
根本原因	サービスのステータスに直接または間接的に影響する根本的な問題が一覧表示されます。 同じ問題は、{SERVICE.ROOTCAUSE} マクロが返す内容としても一覧表示されます。 問題名をクリックすると、監視 → 障害でその詳細を確認できます。 サービスのステータスに影響しない問題は一覧に表示されません。
作成日時	サービスが作成された時刻が表示されます。
タグ	サービスのタグが表示されます。タグは、サービスアクションおよびSLAでサービスを識別するために使用されます。

ボタン

すべてのセクションで共通の表示モードボタンについては、Monitoring ページで説明しています。

フィルターの使用

フィルターを使用して、関心のあるサービスのみを表示できます。



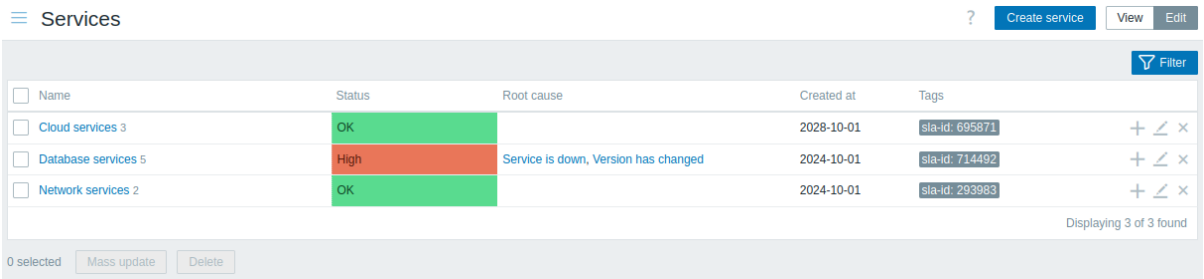
パラメータ	説明
名前	サービス名でフィルタリングします。
ステータス	サービスのステータスでフィルタリングします。
子サービスのないサービスのみ	チェックボックスをオンにすると、子サービスを持たないサービスのみを表示します。このパラメータは、サービスの編集モードでのみ使用できます。
障害タグのないサービスのみ	チェックボックスをオンにすると、障害タグを持たないサービスのみを表示します。このパラメータは、サービスの編集モードでのみ使用できます。

パラメータ	説明
タグ	サービスのタグ名と値、またはサービスの障害タグ名と値（サービスの編集モード）でフィルタリングします。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件で使用できる演算子は次のとおりです。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

サービスの編集

編集ボタンをクリックして編集モードにアクセスします。編集モードでは、一覧の各エントリの前にチェックボックスが追加され、さらに次の追加オプションが表示されます。

-  - このサービスに子サービスを追加
-  - このサービスを編集
-  - このサービスを削除

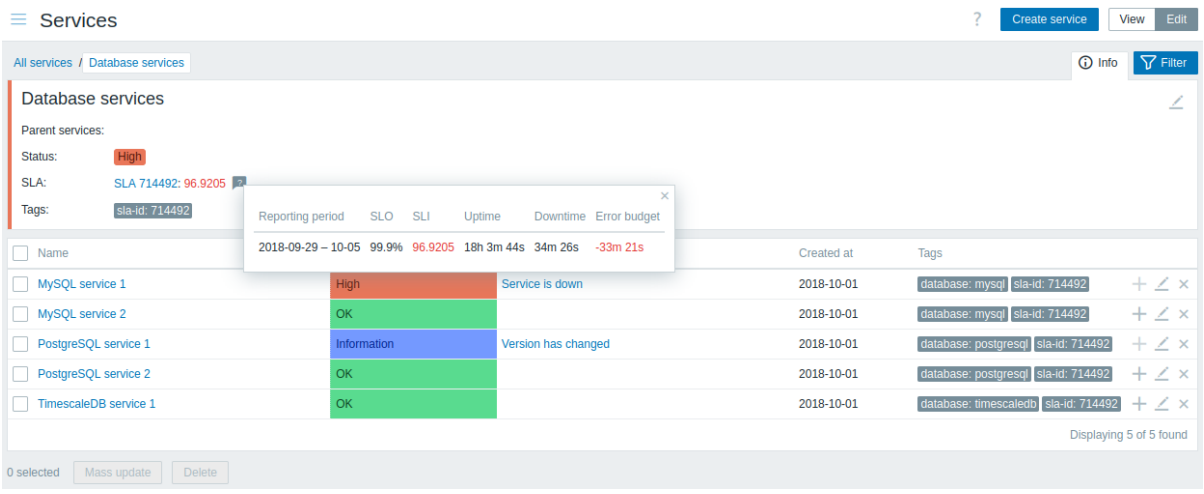


新しいサービスを設定するには、右上隅にあるサービスの作成ボタンをクリックします。

サービスの詳細

サービスの詳細にアクセスするには、サービス名をクリックします。すべてのサービスの一覧に戻るには、*All services* をクリックします。

サービスの詳細には、情報ボックスと子サービスの一覧が含まれます。



情報ボックスにアクセスするには、*Info* タブをクリックします。情報ボックスには、次の項目が含まれます。

- 親サービスの名前（存在する場合）
- このサービスの現在のステータス
- このサービスの現在の SLA（複数ある場合があります）。形式は `SLA name:service level indicator` です。`SLA name` は、このサービスの SLA レポートへのリンクにもなっています。情報ボックス内でサービスレベル指標（SLI）の横にマウスカーソルを置くと、SLI の詳細を示すポップアップ情報一覧が表示されます。サービスレベル指標には、現在のサービスレベルがパーセンテージで表示されます。
- サービスタグ

情報ボックスには、サービス設定 へのリンクも含まれます。

子サービスのフィルターを使用するには、*Filter* タブをクリックします。

編集モードでは、子サービス一覧に追加の編集オプションが表示されます。

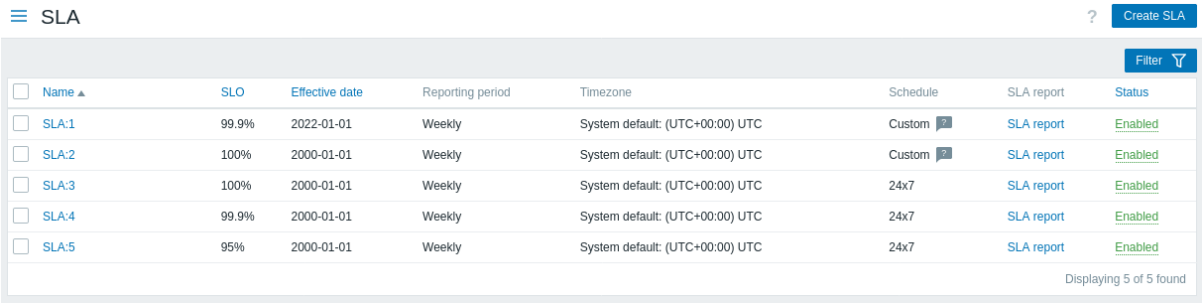
-  - このサービスに子サービスを追加
-  - このサービスを編集
-  - このサービスを削除

2 SLA

概要

このセクションでは SLA を表示および設定することができます。

SLAs



設定された SLAs のリストが表示されます。注ユーザーがアクセスできるサービスに関連する SLAs のみが表示されることに注意してください。(ユーザーロールに対して * SLA の管理 * が有効になっていない限り、読み取り専用として表示されます)

表示情報:

パラメータ	説明
名前	SLA 名が表示されます。 名前はSLA の設定へのリンクです。
SLO	サービスレベル目標（SLO）が表示されます。
発行日	SLA 計算の開始日が表示されます
レポート期間	SLA レポートで使用される期間が表示されます - 毎日、毎週、毎月、四半期、または毎年
タイムゾーン	SLA タイムゾーンが表示されます
スケジュール	SLA スケジュールが表示されます - 24 時間 365 日またはカスタム
SLA レポート	リンクをクリックして、この SLA の SLA レポートを表示します
ステータス	SLA ステータスが表示されます - 有効または無効

3 SLA レポート

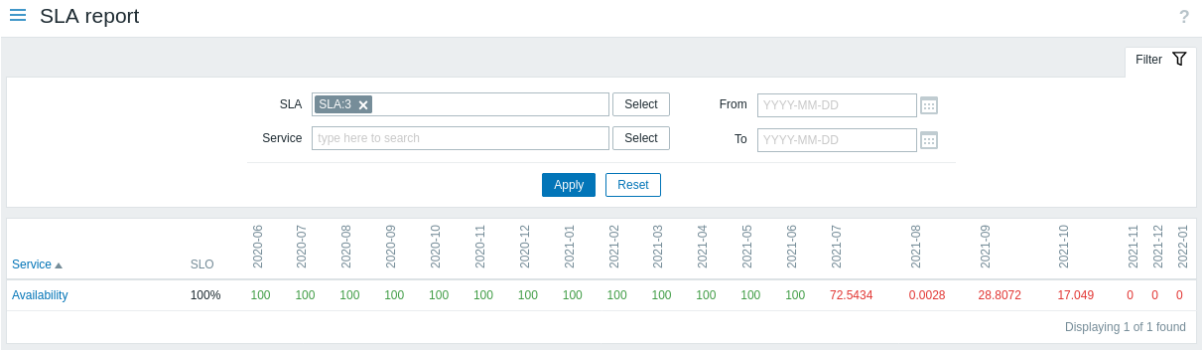
概要

このセクションでは、フィルターで選択した条件に基づくSLAレポートを表示できます。

SLA レポートは、ダッシュボードウィジェットとして表示することもできます。

レポート

フィルタを使用すると SLA 名とサービス名に基づいてレポートを選択できます。表示期間を絞り込む事も可能です。

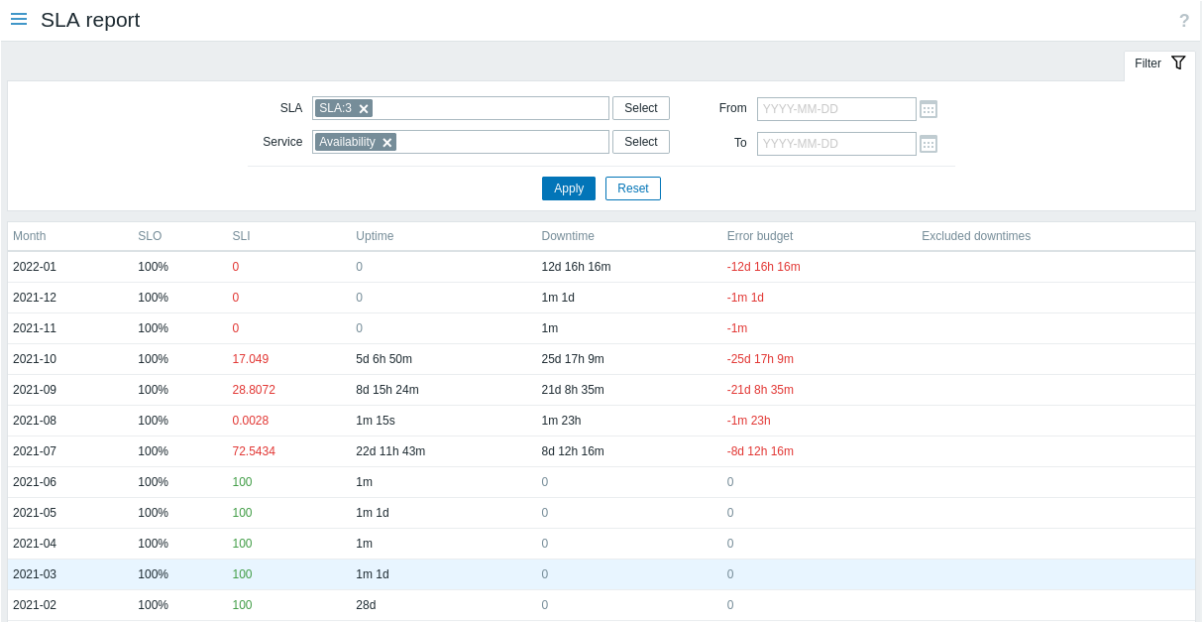


各列（期間）には、その期間の SLI が表示されます。セット SLO に違反している SLI は、赤で強調表示されます。

レポートには 20 の期間が表示されます。開始日と終了日の両方を指定した場合、最大 100 の期間を表示できます。

レポートの詳細

レポート内のサービス名をクリックすると、より詳細な表示を行う別のレポートにアクセスできます。



問題の継続時間が負になる場合は、SLA の計算やレポートには影響しないことに注意してください。

4 インベントリ

概要

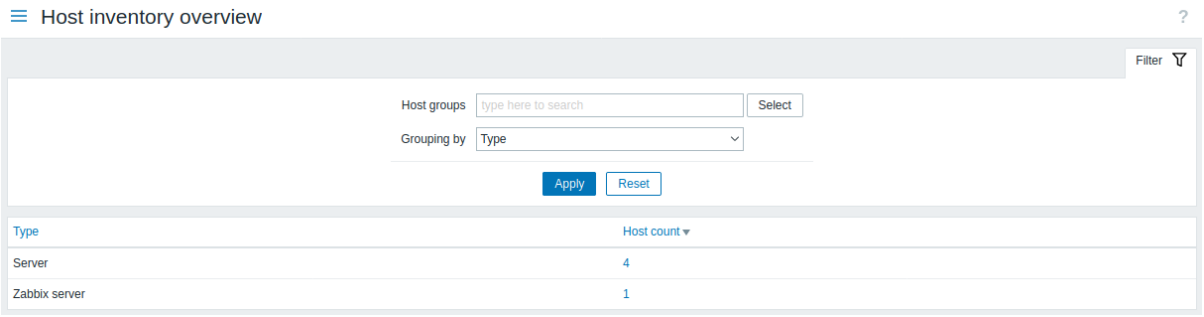
インベントリメニューには、選択したパラメータごとのホストインベントリデータの概要と、ホストインベントリの詳細を表示するセクションがあります。

1 概要

概要

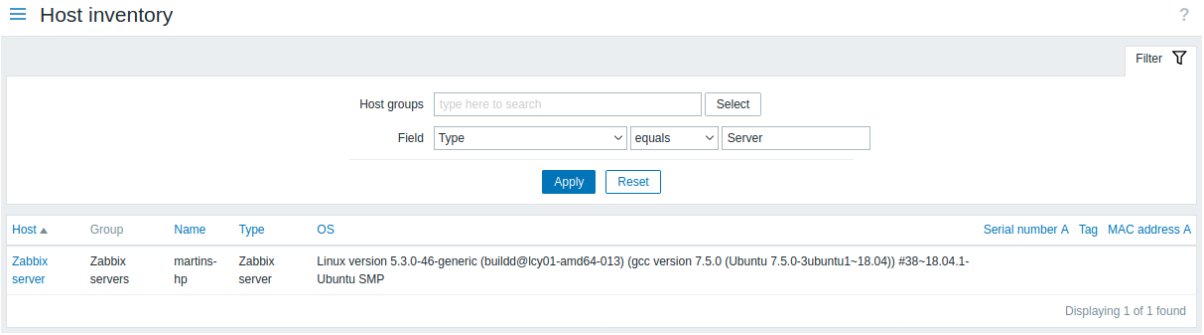
インベントリ → 概要セクションではホストインベントリデータの概要を確認する方法を提供します。

表示する概要については、ホストグループ（またはなし）とデータを表示するインベントリフィールドを選択します。選択したフィールドの各エントリに対応するホストの数が表示されます。



概要の完全性は、ホストで維持されるインベントリ情報の量によって異なります。

ホスト数列の数字はリンクです。これらのホストは *Host Inventory* テーブルで除外されます。

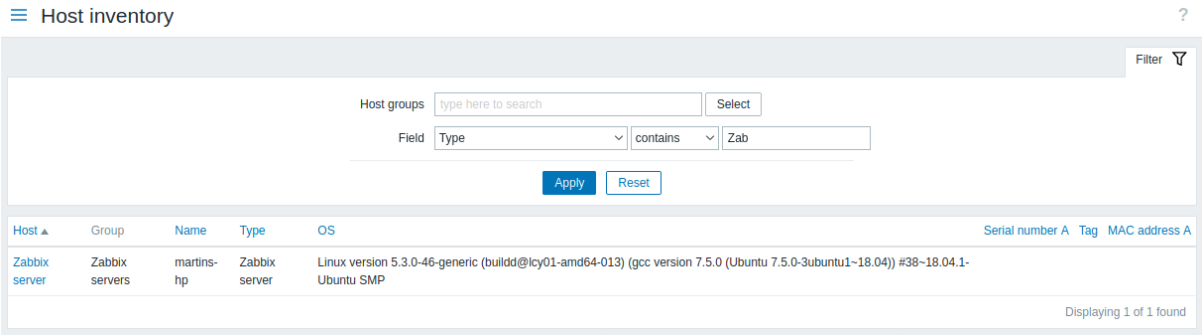


2 ホスト

概要

インベントリ → ホストセクションには、ホストのinventory dataが表示されます。

ホストグループおよび任意のインベントリフィールドでホストをフィルタリングし、必要なホストのみを表示できます。

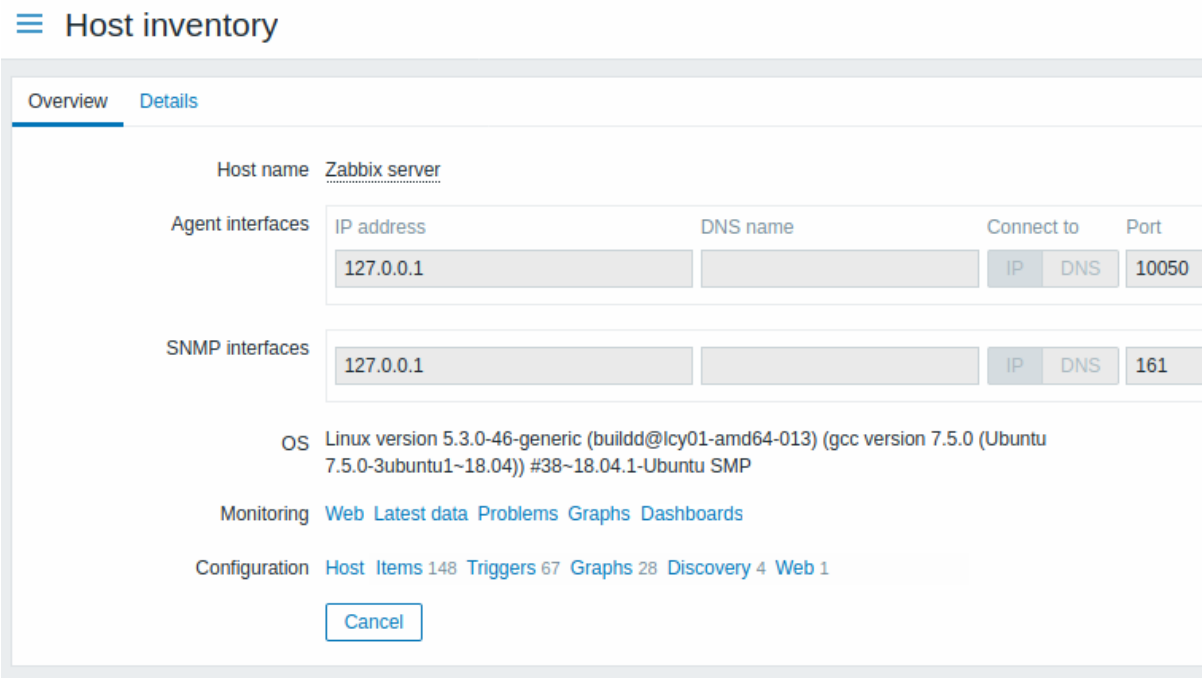


すべてのホストインベントリを表示するには、”Reset” ボタンを押します。

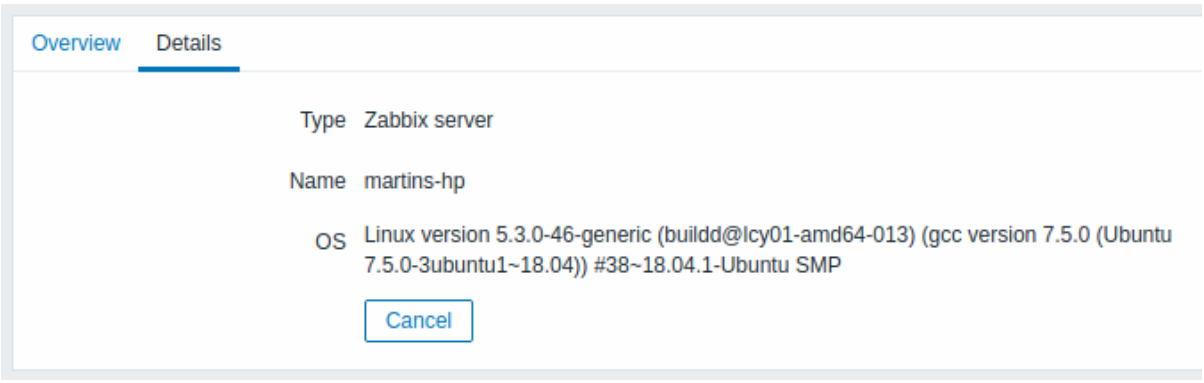
テーブルには主要なインベントリフィールドのみが表示されますが、そのホストで利用可能なすべてのインベントリ情報を表示することもできます。これを行うには、最初の列にあるホスト名をクリックします。

インベントリの詳細

概要タブには、ホストに関する一般的な情報、最新の監視データ、およびホスト設定オプションが表示されます。



詳細タブには、ホストで利用可能なすべてのインベントリの詳細が表示されます。



インベントリデータの完全性は、ホストでどの程度のインベントリ情報が管理されているかによって異なります。情報がまったく管理されていない場合、詳細タブは無効になります。

5 レポート

概要

レポートメニューには、システム情報、トリガー、収集されたデータなどのパラメーターの表示に焦点を当てた、事前定義されたユーザーがカスタマイズ可能なさまざまなレポートを含むいくつかのセクションがあります。

1 システム情報

概要

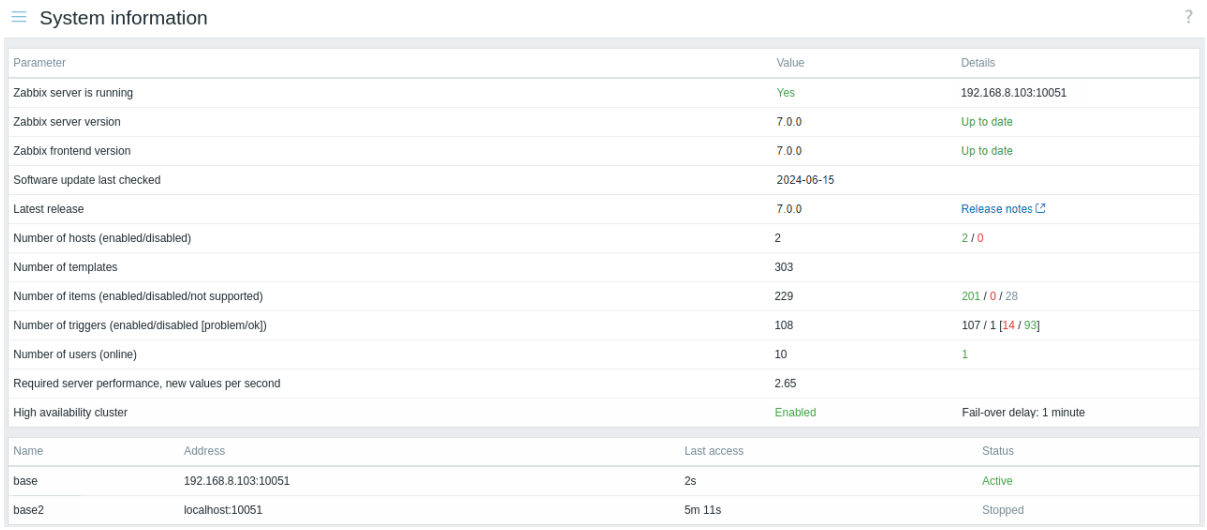
レポート → システム情報では、主要な Zabbix サーバーおよびシステムデータの概要が表示されます。システムデータは、内部アイテム を使用して収集されます。

高可用性構成では、システム情報の取得元（サーバーインスタンス）を切り替えることができます。これを行うには、*zabbix.conf.php* ファイルを編集し、$ZBX_SERVER または $ZBX_SERVER と $ZBX_SERVER_PORT の両方をコメント解除して、アクティブとして表示されているものとは別のサーバーに設定します。$ZBX_SERVER のみを設定した場合、$ZBX_SERVER_PORT にはデフォルト値 (10051) が使用されます。

高可用性構成が有効になっている場合、システム統計の下に高可用性ノードの詳細を示す別のブロックが表示されます。このブロックは、Zabbix の *Super Admin* ユーザーにのみ表示されます。

システム情報は、ダッシュボードのウィジェット としても利用できます。

システム統計



表示されるデータ:

パラメータ	値	詳細
Zabbix サーバーが稼働中	Zabbix サーバーのステータス: **Yes** - サーバーは稼働中 **No** - サーバーは稼働していません 注: 残りの情報を表示するには、Web インターフェースでサーバーが稼働している必要があり、かつサーバー上で少なくとも 1 つのトラッパープロセスが開始されている必要があります (zabbix_server.conf ファイルの StartTrappers パラメータ > 0)。	Zabbix サーバーの場所とポート。
Zabbix サーバーバージョン	現在のサーバーバージョン番号が表示されます。 注: Zabbix サーバーが稼働中の場合にのみ表示されます。	サーバーバージョンのステータスが表示されます: **Up to date** - 最新バージョンを使用中; **New update available** - より新しいバージョンが利用可能; **Outdated** - このバージョンの完全サポート期間は終了しています。 この情報は、Zabbix サーバーの設定でソフトウェア更新チェックが有効な場合にのみ利用できます。最後のソフトウェア更新チェックが 1 週間以上前に実行された場合、または現在のバージョンに関するデータが存在しない場合は何も表示されません。
Zabbix Web インターフェースバージョン	Zabbix Web インターフェースのバージョン番号が表示されます。	Zabbix Web インターフェースのバージョンステータスが表示されます: **Up to date** - 最新バージョンを使用中; **New update available** - より新しいバージョンが利用可能; **Outdated** - このバージョンの完全サポート期間は終了しています。 この情報は、Zabbix サーバーの設定でソフトウェア更新チェックが有効な場合にのみ利用できます。最後のソフトウェア更新チェックが 1 週間以上前に実行された場合、または現在のバージョンに関するデータが存在しない場合は何も表示されません。
ソフトウェア更新の最終確認	最後に Zabbix ソフトウェア更新チェックが行われた日付が表示されます。 この情報は、Zabbix サーバーの設定でソフトウェア更新チェックが有効な場合にのみ利用できます。	

パラメータ	値	詳細
最新リリース	現在の Zabbix バージョンに対して、より新しいリリースがある場合はその番号が表示されます。 この情報は、Zabbix サーバーの設定でソフトウェア更新チェックが有効な場合にのみ利用できます。 最後のソフトウェア更新チェックが 1 週間以上前に実行された場合、または現在のバージョンに関するデータが存在しない場合は何も表示されません。	利用可能な最新の Zabbix リリースのリリースノートへのリンクが表示されます。
ホスト数	設定されているホストの総数が表示されます。	監視中のホスト数/未監視のホスト数。
テンプレート数	テンプレートの総数が表示されます。	
アイテム数	アイテムの総数が表示されます。	ホストレベルの監視中/無効/未サポートのアイテム数。 無効なホスト上のアイテムは無効としてカウントされます。
トリガー数	トリガーの総数が表示されます。	ホストレベルの有効/無効のトリガー数、および有効なトリガーの”Problem”/”OK” 状態ごとの内訳。 ”OK” 状態に一覧表示されるトリガーには、ステータスが”Unknown” のトリガーも含まれます。 無効なアイテムに依存するトリガー、または無効なホストに割り当てられたトリガーは無効としてカウントされます。
ユーザー数	設定されているユーザーの総数が表示されます。	オンライン中のユーザー数。
必要なサーバーパフォーマンス (*1* 秒あたりの新規値数)	Zabbix サーバーが 1 秒あたりに処理すると想定される新しい値の数が表示されます。	必要なサーバーパフォーマンスは推定値であり、目安として役立ちます。実際に処理された値の正確な数値については、`zabbix[wcache,values,all]` の内部アイテムを使用してください。 監視中のホストにある有効なアイテムが計算に含まれます。ログアイテムは、アイテム更新間隔ごとに 1 つの値としてカウントされます。通常の間隔の値はカウントされますが、柔軟な間隔およびスケジューリング間隔の値はカウントされません。”nodata” メンテナンス期間中も計算は調整されません。トラッパーアイテムはカウントされません。
Zabbix サーバー上のグローバルスクリプト	サーバー設定で EnableGlobalScripts=0 を設定して Zabbix サーバー上のグローバルスクリプトが無効になっている場合、このフィールドには **Disabled** と表示されます。	
高可用性クラスター	Zabbix サーバーの高可用性クラスターのステータス: **Disabled** - スタンドアロンサーバー **Enabled** - 少なくとも 1 つの高可用性ノードが存在	有効な場合、フェイルオーバー遅延が表示されます。

システム情報には、次の条件でエラーメッセージも表示されます:

- 使用しているデータベースに必要な文字セットまたは照合順序（UTF-8）が設定されていません。
- データベースのバージョンがサポート範囲より低い、または高いです（*Super admin role* タイプのユーザーのみ利用可能）。
- TimescaleDB のハウスキーピングが正しく設定されていません（履歴テーブルまたはトレンドテーブルに圧縮チャンクが含まれているにもかかわらず、*Override item history period* または *Override item trend period* オプションが無効になっている）。

高可用性ノード

高可用性クラスターが有効になっている場合、各高可用性ノードのステータスを示す別のデータブロックが表示されます。



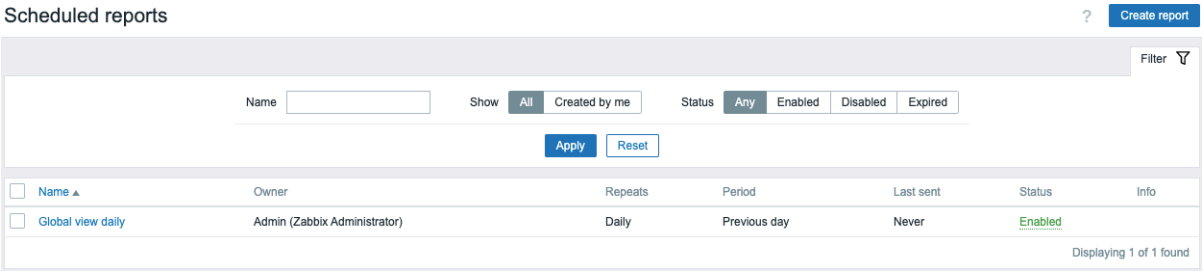
表示されるデータ:

Column	Description
Name	サーバー設定で定義されたノード名。
Address	ノードの IP アドレスとポート。
Last access	ノードが最後にアクセスした時刻。 セルにマウスオーバーすると、最後のアクセス時刻のタイムスタンプがロング形式で表示されます。
Status	ノードのステータス。テーブルの行は、以下の優先順位でこれらのステータス順に並べ替えられます: **Active** - ノードは稼働中で正常に動作しています **Unavailable** - フェイルオーバー遅延時間を超えてノードが確認されていません（原因を確認することをお勧めします） **Stopped** - ノードが停止されているか、起動できませんでした（起動または削除が必要な場合があります） **Standby** - ノードは稼働中で待機しています

2 定期レポート

概要

レポート → 定期レポートでは、十分な権限を持つユーザーが、ダッシュボードの PDF 版を定期的に生成し、指定した受信者にメールで送信するよう設定できます。



開始画面には定期レポートに関する情報が表示され、簡単に目的のレポートを見つけられるようフィルタリングできます。詳細は以下のフィルタの使用セクションを参照してください。

表示されるデータ:

Column	Description
Name	レポート名です。クリックすると、レポートの設定フォームが開きます。
Owner	レポートを作成したユーザーです。
Repeats	レポートの生成頻度です（毎日/毎週/毎月/毎年）。
Period	レポートの対象期間です（前日、前週、前月、または前年）。
Last sent	最新のレポートが送信された日時です。
Status	レポートの現在の状態です（有効/無効/期限切れ）。十分な権限を持つユーザーは、クリックして状態を変更できます。”Enabled” から”Disabled”（およびその逆）、”Expired” から”Disabled”（およびその逆）に変更できます。権限が不十分なユーザーには、状態はクリック可能ではありません。

Column	Description
Info	情報アイコンを表示します: 赤いアイコンは、レポートの生成に失敗したことを示します。アイコンにマウスオーバーすると、エラー情報を含むツールチップが表示されます。 黄色いアイコンは、レポートは生成されたものの一部またはすべての受信者への送信に失敗したこと、またはレポートの有効期限が切れていることを示します。アイコンにマウスオーバーすると、追加情報を含むツールチップが表示されます。

フィルターの使用

フィルターを使用して、レポートの一覧を絞り込むことができます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

利用可能なフィルタリングオプションは次のとおりです。

- 名前 - 名前の部分一致が可能
- 表示 - 現在のユーザーが作成したレポート、またはすべてのレポート
- ステータス - 「任意」(すべてのレポートを表示)、「有効」、「無効」、または「期限切れ」から選択

フィルターは、*Scheduled reports* セクション名の下にあります。右上隅の *Filter* タブをクリックすると、開いたり折りたたんだりできます。

一括更新

複数のレポートを一度に削除したり、ステータスを変更したい場合があります。個々のレポートを開いて編集する代わりに、そのための一括更新機能を使用できます。

複数のレポートを一括更新するには、次のようにします。

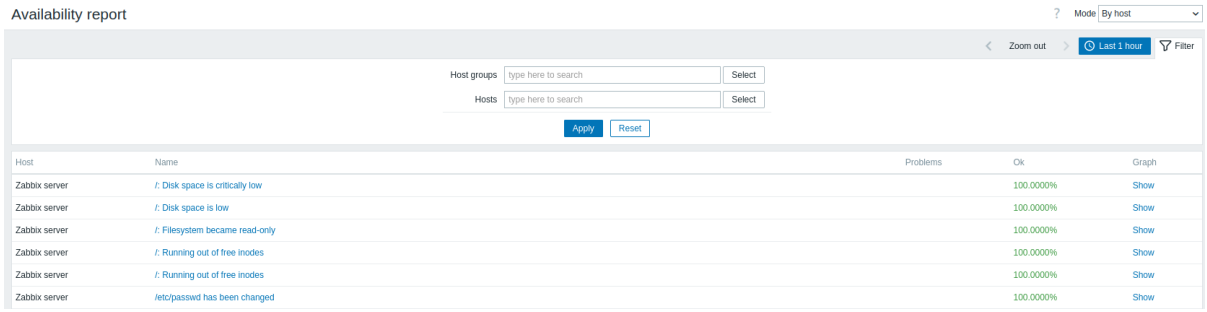
- 一覧で更新するレポートのチェックボックスを選択します
- 一覧の下にある必要なボタン（有効化、無効化、または 削除）をクリックして変更を適用します

3 稼働レポート

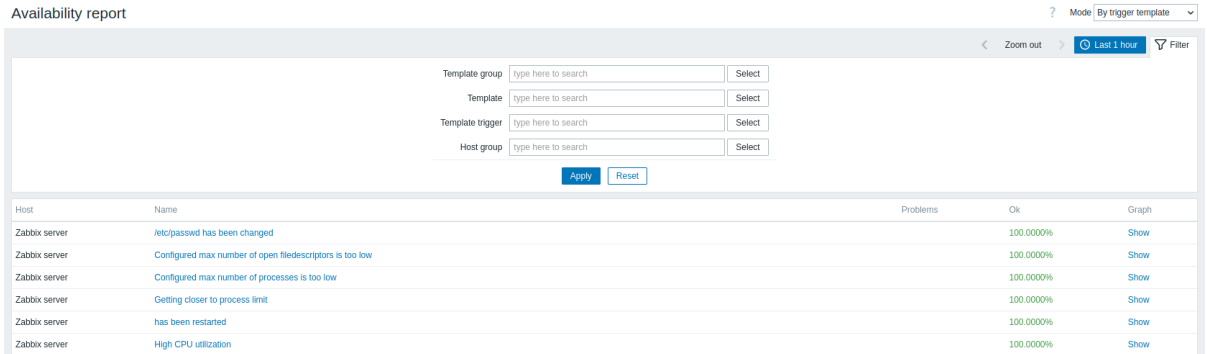
概要

レポート > 可用性レポートでは、各トリガーが問題/OK 状態にあった時間の割合を確認できます。

各状態について時間の割合がパーセンテージで表示されるため、システム内のさまざまな要素の可用性を簡単に判断できます。



右上隅のドロップダウンでは、選択モードを選択できます。つまり、ホストごとにトリガーを表示するか、テンプレートに属するトリガーを表示するかを指定できます。



トリガー名は、そのトリガーの最新イベントへのリンクになっています。

フィルターの使用

フィルターを使用すると、表示されるホストやトリガーの数を絞り込むことができます。
検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

フィルターは、*Availability report* セクション名の下にあります。
右側の *Filter* タブをクリックすると、開いたり折りたたんだりできます。

トリガーテンプレートによるフィルタリング

トリガーテンプレート別モードでは、以下に示す 1 つまたは複数のパラメータで結果をフィルタリングできます。

Parameter	Description
Template group	選択したテンプレートグループに属するテンプレートから継承されたトリガーによってホストをフィルタリングします。 親テンプレートグループを指定すると、ネストされたすべてのテンプレートグループも暗黙的に選択されます。
Template	選択したテンプレート（ネストされたテンプレートを含む）から継承されたトリガーによってホストをフィルタリングします。 ネストされたテンプレートが独自のトリガーを持つ場合、それらのトリガーは表示されません。
Template trigger	選択したトリガーによってホストをフィルタリングします。フィルタリングされたホストの他のトリガーは表示されません。
Host group	選択したホストグループに属するホストをフィルタリングします。

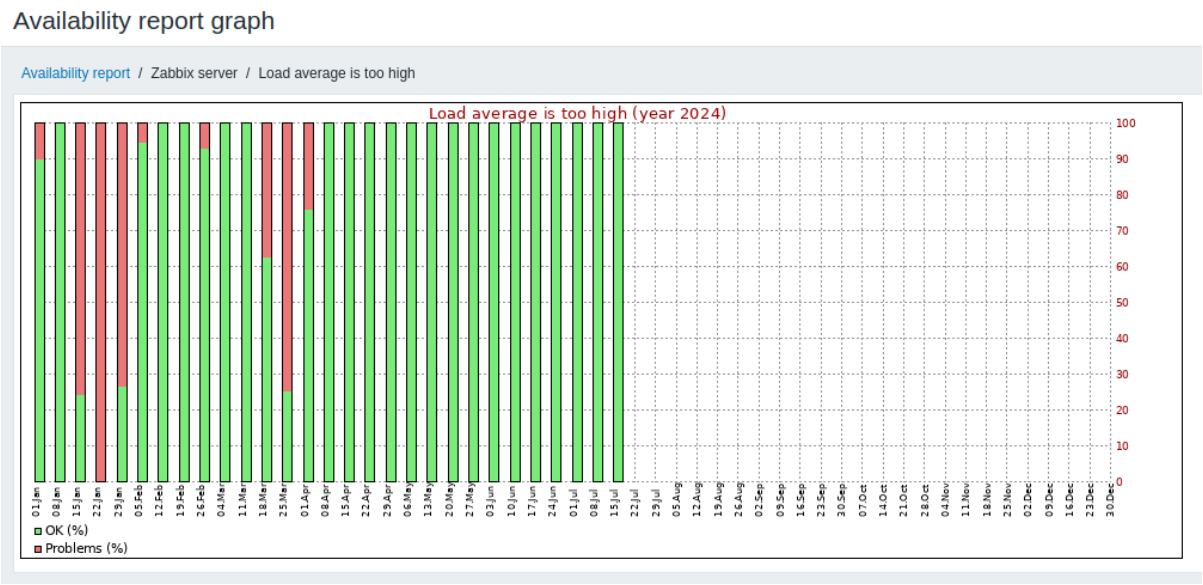
ホストによるフィルタリング

ホスト別モードでは、結果をホストまたはホストグループでフィルタリングできます。
親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。

期間セレクター

期間セレクターを使用すると、よく使われる期間をワンクリックで選択できます。このセレクターは、フィルターの横にある期間タブをクリックすると展開および折りたたみができます。

グラフ列の表示をクリックすると、可用性情報が棒グラフで表示されます。各バーは、現在の年の経過した各週を表します。



バーの緑色は正常時間を、赤色は障害時間を表します。

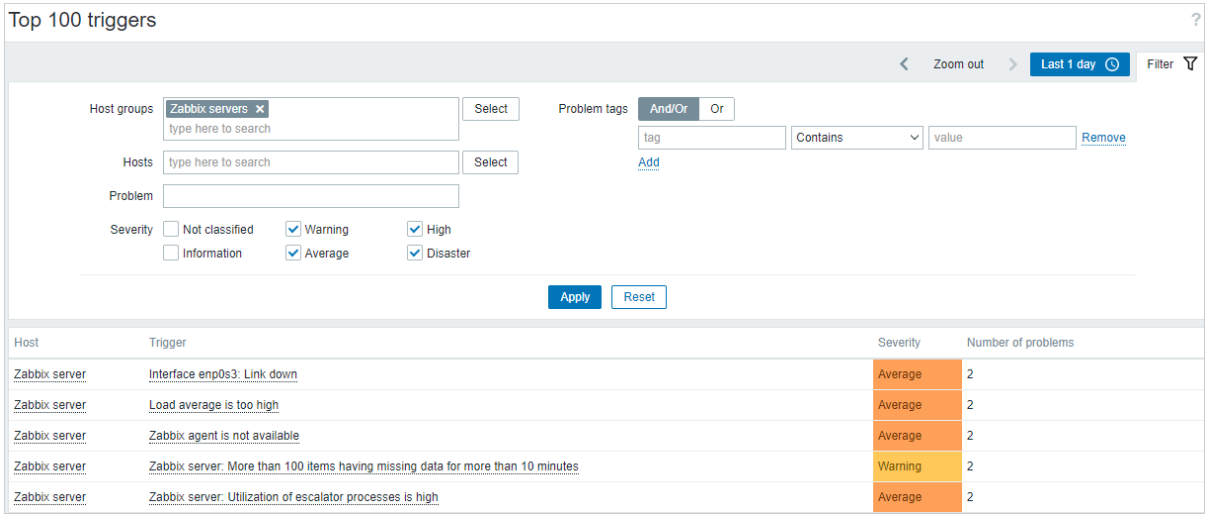
メンテナンス期間の影響

メンテナンスでは、可用性レポートから時間が自動的に除外されることはありません。
データ収集なしとして設定されたメンテナンスのみがポーリングを停止し（そのため問題は生成されません）、その結果、影響を受けるトリガーの可用性パーセンテージは変化しません。

4 障害発生数上位 **100** トリガー

概要

レポート → トップ *100* トリガーでは、選択した期間中に検出された障害の数が最も多いトリガーを確認できます。



ホスト列とトリガー列のエントリはいずれもリンクになっており、便利なオプションを利用できます。

- ホスト - ホスト名をクリックすると、ホストメニューが表示されます
- トリガー - トリガー名をクリックすると、最新のイベントへのリンク、各トリガーアイテムのシンプルグラフ、およびトリガー自体と各トリガーアイテムの設定フォームが表示されます

フィルターの使用

フィルターを使用すると、ホストグループ、ホスト、障害名、タグ、またはトリガーの深刻度でトリガーを表示できます。親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルターは、トップ *100* トリガーセクション名の下にあります。右側の *Filter* タブをクリックすると、開閉できます。

期間セレクター

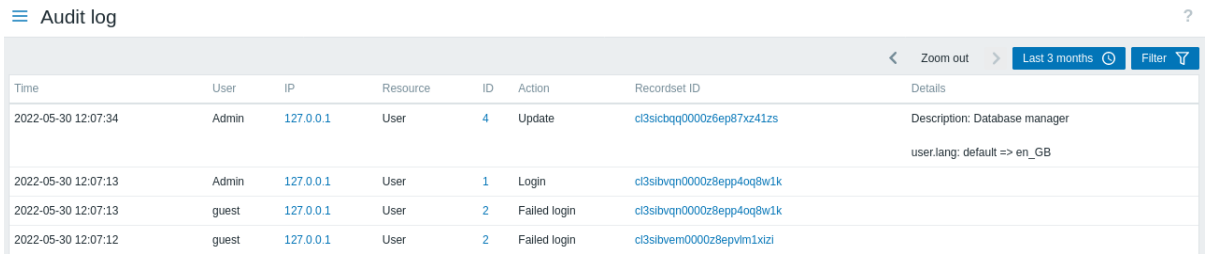
期間セレクターを使用すると、よく使われる期間をマウスクリック 1 回で選択できます。期間セレクターは、フィルターの横にある *Time period* タブをクリックすると、展開および折りたたみができます。

5 監査ログ

概要

レポート → 監査ログセクションでは、ユーザーおよびシステムのアクティビティの記録を表示できます。

Note:
監査記録を収集して表示するには、管理 → 監査ログ セクションで 監査ログを有効にするチェックボックスをオンにする必要があります。この設定が有効になっていない場合、アクティビティの履歴はデータベースに記録されず、監査ログにも表示されません。



監査ログには、次のデータが表示されます。

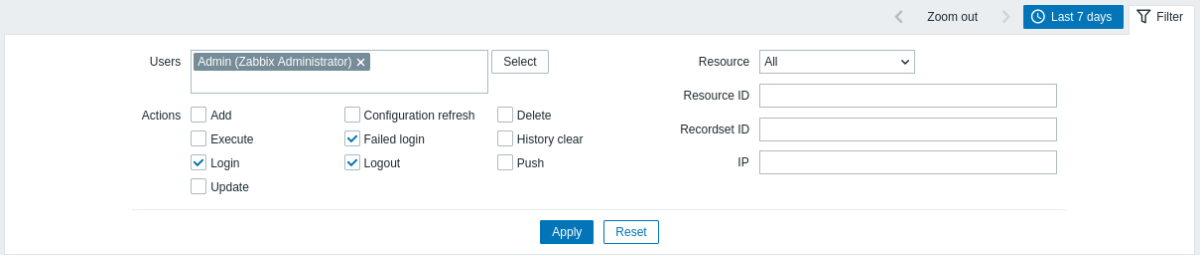
Column	Description
Time	監査記録のタイムスタンプ。
User	アクティビティを実行したユーザー。
IP	アクティビティが開始された IP。 ハイパーリンクをクリックすると、この IP で監査ログの記録がフィルタリングされます。
Resource	影響を受けたリソースの種類（*API token*、*Action*、*Authentication*、*Autoregistration* など）。

Column	Description
ID	影響を受けたリソースの ID。 ハイパーリンクをクリックすると、このリソース ID で監査ログの記録がフィルタリングされます。
Action	アクティビティの種類（*Add*、設定の更新、*Delete*、*Execute*、*Failed login*、*History clear*、*Login*、*Logout*、*Push*、*Update*）。
Recordset ID	同じ操作の結果として作成されたすべての監査ログ記録に共通する ID。 例えば、テンプレートをホストにリンクすると、継承された各テンプレートエンティティ（アイテム、トリガーなど）ごとに個別の監査ログ記録が作成されます。これらの記録はすべて同じ *Recordset ID* を持ちます。 ハイパーリンクをクリックすると、この *Recordset ID* で監査ログの記録がフィルタリングされます。
Details	リソースの説明と、実行されたアクティビティの詳細情報。 記録に 3 行以上含まれている場合は、追加の *Details* リンクが表示されます。このリンクをクリックすると、変更内容の完全な一覧を表示できます。

Note:
トラッパーアイテム またはHTTP エージェントアイテム（トラップが有効な場合）が何らかのデータを受信した場合、監査ログにエントリが追加されるのは、データが`history.push` API メソッドを使用して送信された場合のみであり、Zabbix sender ユーティリティを使用した場合は追加されません。

フィルターの使用

フィルターは、監査ログバーの下にあります。
右上隅の フィルタータブをクリックすると、開いたり折りたたんだりできます。



フィルターを使用すると、ユーザー、影響を受けたリソース、リソース ID、実行された操作（レコードセット *ID*）、および IP で記録を絞り込むことができます。
リソースによっては、フィルターで 1 つ以上の特定のアクションを選択できます。

検索パフォーマンスを向上させるため、すべてのデータはマクロを未解決のまま検索されます。

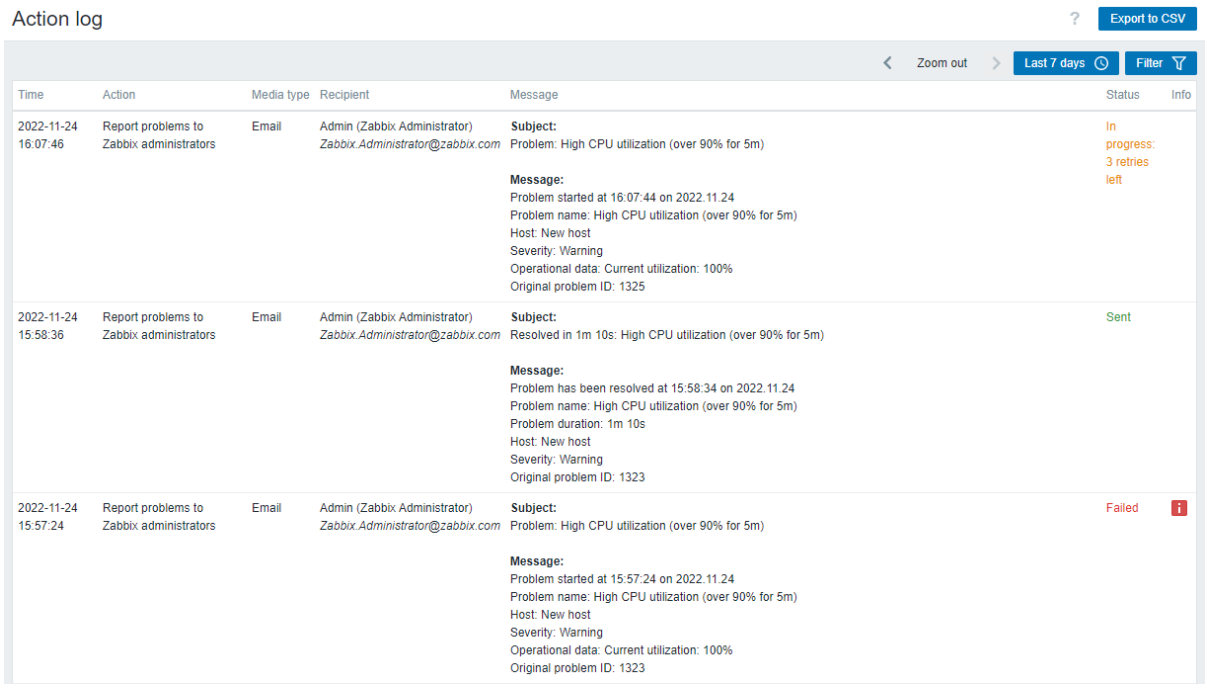
期間セレクター

期間セレクターを使用すると、よく必要とされる期間をマウスで 1 回クリックするだけで選択できます。
期間セレクターは、右上隅にある期間タブをクリックすることで展開および折りたたみできます。

6 アクションログ

概要

レポート → アクションログセクションでは、ユーザーはアクション内で実行された操作（通知、リモートコマンド）の詳細を確認できます。



表示されるデータ:

Column	Description
Time	操作のタイムスタンプ。
Action	操作を発生させたアクションの名前。
Media type	通知の送信に使用されたメディアタイプ（例: Email、Jira など）。 リモートコマンドを実行した操作の場合、この列は空になります。
Recipient	通知の受信者に関する情報 - ユーザー名、名前と姓（括弧内）、およびメディアタイプに応じた追加情報（メールアドレス、ユーザー名など）。 リモートコマンドを実行した操作の場合、この列は空になります。
Message	メッセージ、リモートコマンド、またはグローバルスクリプト名の内容。 リモートコマンドは、コロン記号で対象ホストと区切られます: `<host>:<command>`。たとえば、リモートコマンドが Zabbix server 上で実行された場合、情報は次の形式になります: `Zabbix server:<command>`。
Status	操作のステータス: 進行中 - 通知送信の操作が進行中です（サーバーが通知送信を再試行する残り回数も表示されます） 送信済み - 通知が送信されました 実行済み - リモートコマンドが実行されました 失敗 - 操作は完了していません
Info	操作の実行に関するエラー情報（存在する場合）。

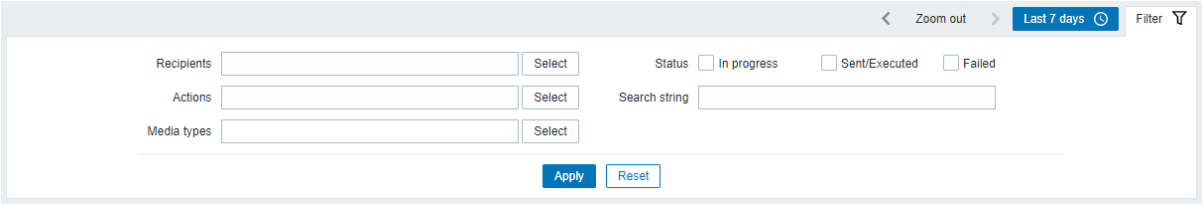
ボタン

ページ右上隅のボタンでは、次のオプションを利用できます。

Export to CSV	現在のページのアクションログレコードを CSV ファイルにエクスポートします。フィルターが適用されている場合は、フィルターされたレコードのみがエクスポートされます。 エクスポートされた CSV ファイルでは、「Recipient」列と「Message」列が複数の列、すなわち「Recipient's Zabbix username」、「Recipient's name」、「Recipient's surname」、「Recipient」、および「Subject」、「Message」、「Command」に分割されます。

フィルターの使用

フィルターは、アクションログバーの下にあります。
ページ右上の *Filter* タブをクリックすると、開いたり折りたたんだりできます。



フィルターを使用すると、通知の受信者、アクション、メディアタイプ、ステータス、
またはメッセージ/リモートコマンドの内容（*Search string*）でレコードを絞り込むことができます。
検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

期間セレクター

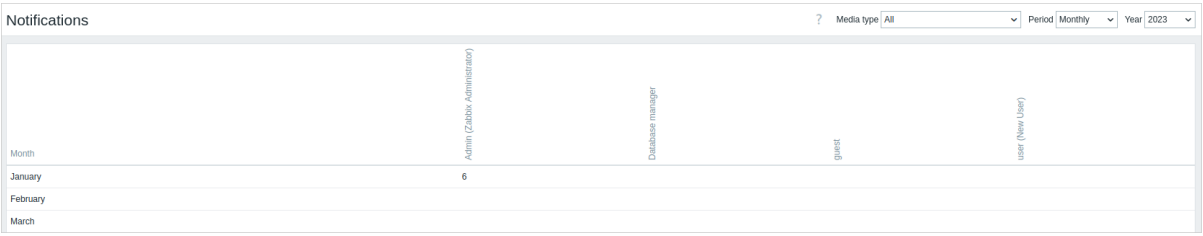
期間セレクターを使用すると、よく必要とされる期間をマウスで 1 回クリックするだけで選択できます。
期間セレクターは、フィルターの横にある期間タブをクリックすると展開および折りたたみができます。

7 通知レポート

概要

レポート → 通知セクションでは、各ユーザーに送信された通知数に関するレポートが表示されます。

右上隅のドロップダウンから、送信された通知について、メディアタイプ（またはすべて）、期間（各日/週/月/年のデータ）、および年を選択できます。



各列には、各システムユーザーごとの合計が表示されます。

6 データ収集

概要

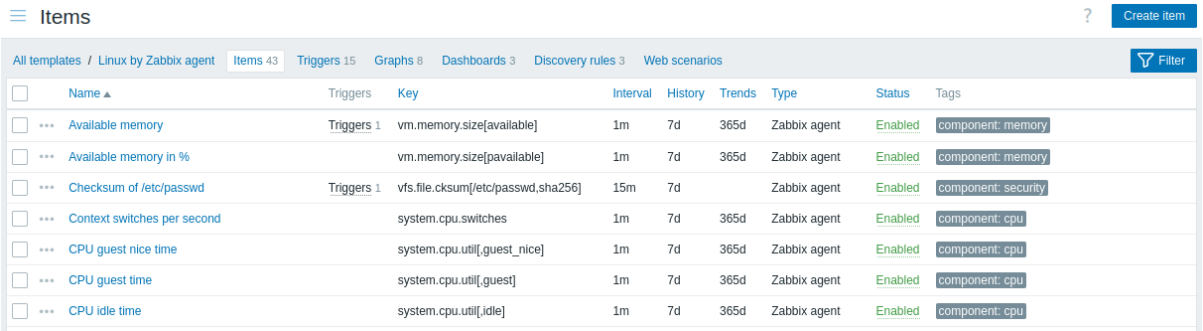
このメニューには、データ収集の設定に関連するセクションがあります。

1 アイテム

概要

テンプレートのアイテム一覧は、データ収集 → テンプレートで該当するテンプレートの アイテムをクリックすると表示できます。

既存のアイテムの一覧が表示されます。



表示されるデータ:

Column	Description
アイテムメニュー	三点アイコンをクリックすると、アイテムメニューが開きます。

Column	Description
テンプレート	アイテムが属するテンプレートです。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 この列は、フィルターで複数のテンプレートが選択されている場合、またはテンプレートが選択されていない場合にのみ表示されます。
名前	アイテム名が、アイテムの詳細への青いリンクとして表示されます。 アイテム名のリンクをクリックすると、アイテムの設定フォームが開きます。 アイテムが別のテンプレートから継承されている場合、アイテム名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのアイテム一覧が開きます。
トリガー	マウスカーソルをトリガーの上に移動すると、アイテムに関連付けられたトリガーを表示する情報ボックスが表示されます。 トリガーの数は灰色で表示されます。
キー	アイテムキーが表示されます。
間隔	チェックの頻度が表示されます。
履歴	アイテムデータの履歴が何日間保持されるかが表示されます。
トレンド	アイテムのトレンド履歴が何日間保持されるかが表示されます。
タイプ	アイテムタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
ステータス	アイテムのステータスが表示されます - 有効または 無効。ステータスをクリックすると、有効から無効（およびその逆）に変更できます。
タグ	アイテムタグが表示されます。 最大 3 つのタグ（name:value ペア）を表示できます。タグがそれより多い場合は、”...” リンクが表示され、マウスオーバーですべてのタグを確認できます。

新しいアイテムを設定するには、右上隅の アイテムの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

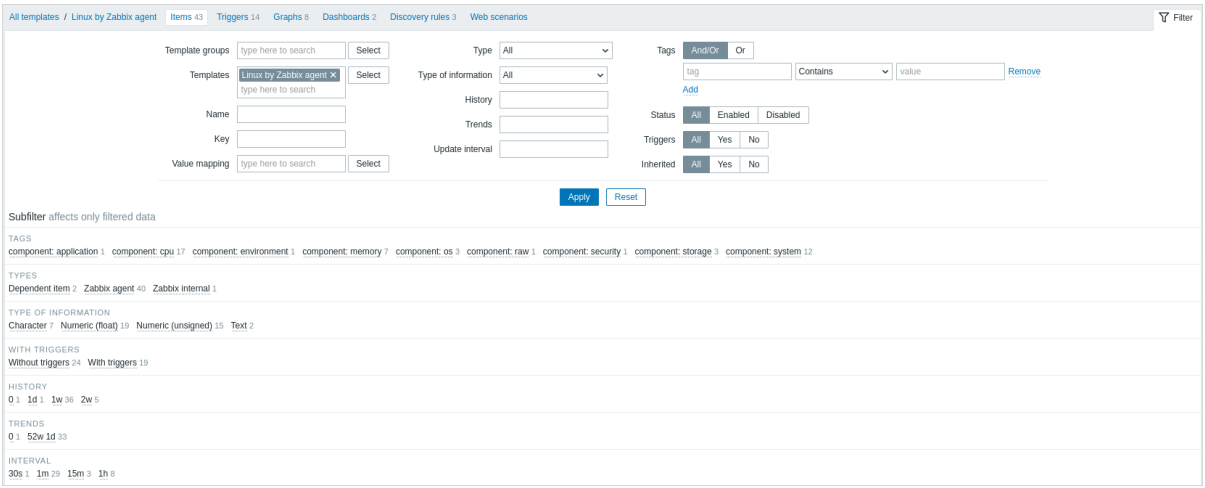
- *Enable* - アイテムのステータスを有効に変更します。
- *Disable* - アイテムのステータスを無効に変更します。
- *Copy* - アイテムを他のホストまたはテンプレートにコピーします。
- *Mass update* - 複数のアイテムのいくつかのプロパティを一度に更新します。
- *Delete* - アイテムを削除します。

これらのオプションを使用するには、対象のアイテムの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

アイテムの一覧には多数のアイテムが含まれる場合があります。フィルターを使用すると、その一部を絞り込み、探しているアイテムをすばやく見つけることができます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

Filter アイコンは右上隅にあります。これをクリックすると、目的の絞り込み条件を指定できるフィルターが開きます。

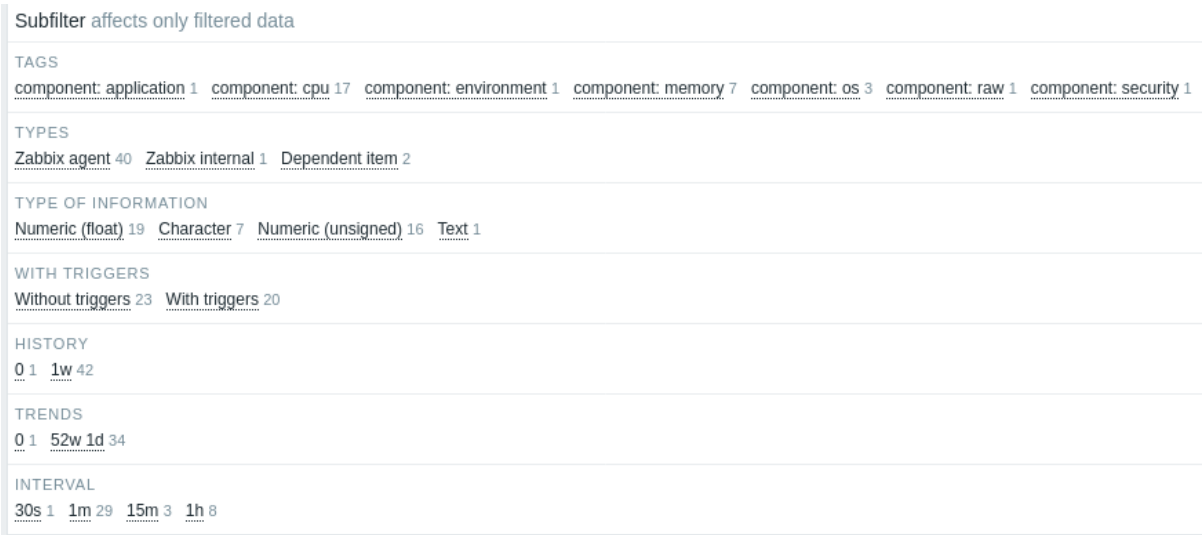


Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループも暗黙的に選択されます。
Templates	1 つ以上のテンプレートで絞り込みます。
Name	アイテム名で絞り込みます。
Key	アイテムキーで絞り込みます。
Value mapping	使用されている値のマッピングで絞り込みます。 *Templates* オプションが空の場合、このパラメータは表示されません。
Type	アイテムタイプ（Zabbix エージェント、SNMP エージェントなど）で絞り込みます。
Type of information	情報の型（数値 unsigned、float など）で絞り込みます。
History	アイテムの履歴を保持する期間で絞り込みます。
Trends	アイテムのトレンドを保持する期間で絞り込みます。
Update interval	アイテムの更新間隔で絞り込みます。
Tags	表示するアイテム数を制限するためのタグを指定します。特定のタグおよびタグ値を含めることも除外することも可能です。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別） **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外します **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない） 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です
Status	アイテムのステータス - *Enabled* または *Disabled* で絞り込みます。
Triggers	トリガーがある（またはない）アイテムで絞り込みます。
Inherited	リンクされたテンプレートから継承された（または継承されていない）アイテムで絞り込みます。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターによる絞り込みをさらに変更できます。

サブフィルターには、関連するアイテムにすばやくアクセスするためのクリック可能なリンクが含まれています。アイテムは、共通のエンティティ（タグ、アイテムタイプ、アイテムの更新間隔など）によって関連付けられています。エンティティをクリックすると、そのエンティティは灰色の背景で強調表示され、アイテムは即座に絞り込まれます（メインフィルターで 適用をクリックする必要はありません）。別のエンティティをクリックすると、絞り込み結果に追加されます。もう一度そのエンティティをクリックすると、絞り込みが解除されます。



サブフィルターは、フィルター済みデータに基づいて生成されます。このデータは 1000 レコードに制限されています。サブフィルターにさらに多くのレコードを表示したい場合は、*Administration -> General -> GUI* にある *Limit for search and filter results* パラメータの値を増やす必要があります。

メインフィルターとは異なり、サブフィルターはテーブルの更新要求ごとに更新されるため、利用可能な絞り込みオプションとその件数について常に最新の情報が表示されます。

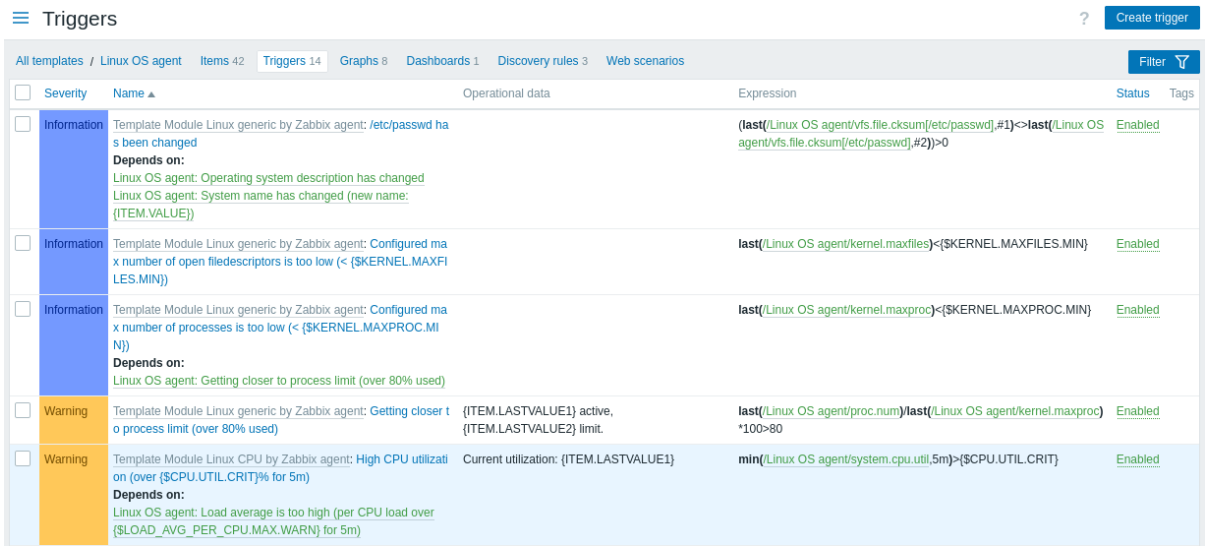
横方向に表示されるエンティティの数は 100 個までに制限されています。それを超える場合は、末尾に三点アイコンが表示されますが、クリックはできません。

各クリック可能なエンティティの横に表示される数字は、そのエンティティにグループ化されているアイテム数を示します（メインフィルターの結果に基づきます）。エンティティをクリックすると、他の利用可能なエンティティの数字には、現在の選択に追加できるアイテム数を示すプラス記号が付いて表示されます。アイテムを持たないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

2 トリガー

概要

テンプレートのトリガー一覧は、データ収集 → テンプレートから該当するテンプレートの トリガーをクリックしてアクセスできます。



表示されるデータ:

Column	Description
重大度	トリガーの重大度が、名前とセルの背景色の両方で表示されます。
テンプレート	トリガーが属するテンプレートです。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 この列は、フィルターで複数のテンプレートが選択されている場合、またはテンプレートが選択されていない場合にのみ表示されます。
名前	トリガー名が、トリガーの詳細への青いリンクとして表示されます。 トリガー名のリンクをクリックすると、トリガーの設定フォームが開きます。 トリガーが別のテンプレートから継承されている場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのトリガー一覧が開きます。
運用データ	トリガーの運用データ定義です。任意の文字列およびマクロを含めることができ、監視 → 障害で動的に展開されます。
条件式	トリガー条件式が表示されます。条件式内のテンプレート-アイテム部分はリンクとして表示され、アイテムの設定フォームに移動できます。
ステータス	トリガーのステータスが 有効または 無効として表示されます。ステータスをクリックすると、有効から 無効（およびその逆）に変更できます。
タグ	トリガーにタグが含まれている場合、この列にタグ名と値が表示されます。

新しいトリガーを設定するには、右上隅にある トリガーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 有効 - トリガーのステータスを有効に

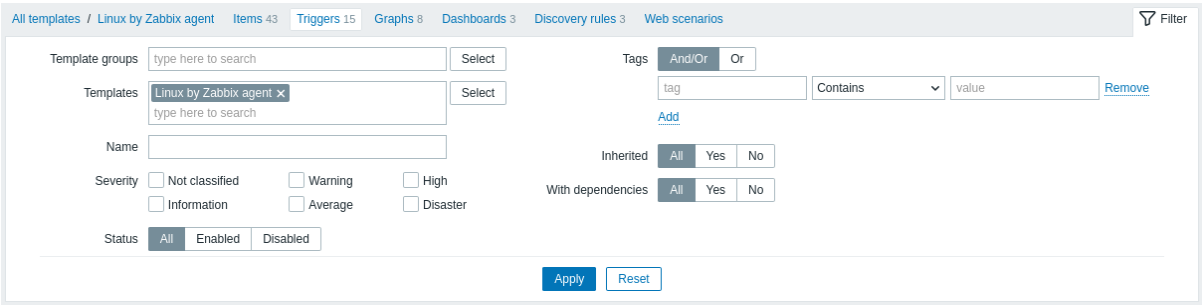
- 無効 - トリガーのステータスを無効に
- コピー - トリガーを他のホストまたはテンプレートにコピーします
- 一括更新 - 一度に複数のトリガーのいくつかのプロパティを更新します
- 削除 - トリガーの削除

これらのオプションを使用するには、トリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるトリガーのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

Filter アイコンは右上隅にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



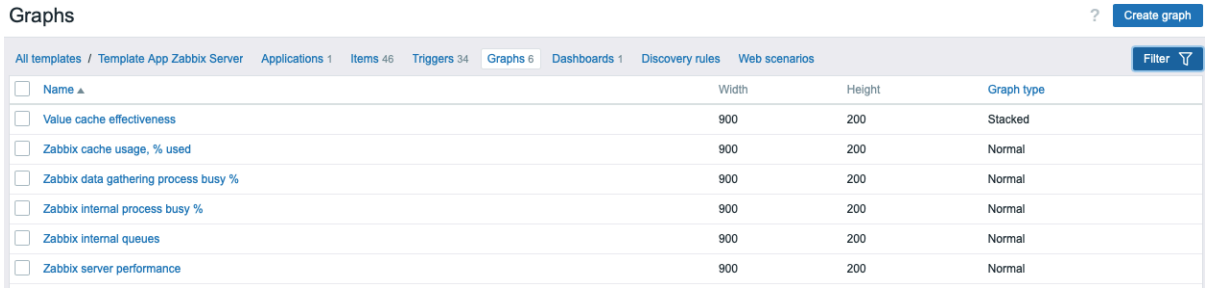
Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループも暗黙的に選択されます。
Templates	1 つ以上のテンプレートで絞り込みます。 上でテンプレートグループがすでに選択されている場合、テンプレートの選択はそれらのグループに限定されます。
Name	トリガー名で絞り込みます。
Severity	1 つまたは複数のトリガー深刻度で絞り込むには選択します。
Status	トリガーのステータスで絞り込みます。
Tags	トリガーのタグ名と値で絞り込みます。特定のタグおよびタグ値を含めることも除外することも可能です。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、以下の演算子を使用できます。 **Exists** - 指定したタグ名を含めます **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別） **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外します **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない） 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です タグ名およびタグ値のフィールドでは、マクロとmacro functions をサポートしています。
Inherited	リンクされたテンプレートから継承されたトリガー（または継承されていないトリガー）で絞り込みます。
With dependencies	依存関係のあるトリガー（または依存関係のないトリガー）で絞り込みます。

3 グラフ

概要

テンプレートのカスタムグラフ一覧には、データ収集 → テンプレートから、該当するテンプレートの グラフをクリックしてアクセスできます。

既存のグラフの一覧が表示されます。



表示されるデータ:

列	説明
テンプレート	グラフが属するテンプレートです。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 この列は、フィルターで複数のテンプレートが選択されている場合、またはテンプレートが選択されていない場合にのみ表示されます。
名前	カスタムグラフの名前が、グラフの詳細への青いリンクとして表示されます。 グラフ名のリンクをクリックすると、グラフの設定フォームが開きます。 グラフが別のテンプレートから継承されている場合、グラフ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、そのテンプレートレベルのグラフ一覧が開きます。
幅	グラフの幅が表示されます。
高さ	グラフの高さが表示されます。
グラフタイプ	グラフのタイプが表示されます - 標準、積み上げ、円、または 分割円。

新しいグラフを設定するには、右上隅にある グラフの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- コピー - グラフを他のホストまたはテンプレートにコピーします
- 削除 - グラフの削除

これらのオプションを使用するには、それぞれのグラフ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

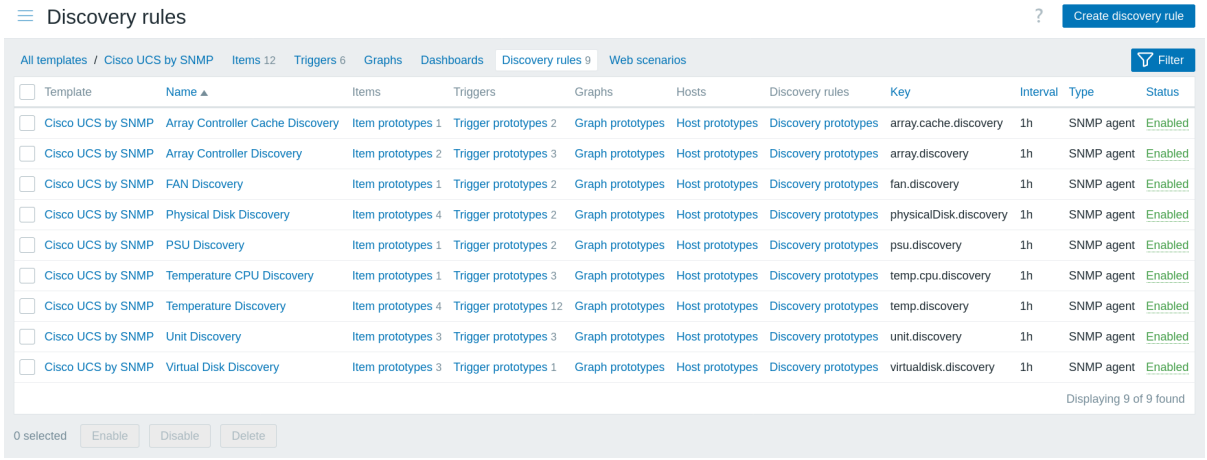
テンプレートグループおよびテンプレートでグラフをフィルタリングできます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

4 ディスカバリルール

概要

テンプレートのローレベルディスカバリルールの一覧は、データ収集 → テンプレートから対象のテンプレートの ディスカバリをクリックしてアクセスできます。

既存のローレベルディスカバリルールの一覧が表示されます。フィルター設定を変更することで、テンプレートに依存しないすべてのディスカバリルール、または特定のテンプレートグループのすべてのディスカバリルールを表示することもできます。



表示されるデータ:

Column	Description
Template	このテンプレートディスカバリルールが属するテンプレートです。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。
Name	ルール名です。青色のリンクとして表示されます。 ルール名をクリックすると、ローレベルディスカバリルールの設定フォームが開きます。 ディスカバリルールが別のテンプレートから継承されている場合、ルール名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルのディスカバリルール一覧が開きます。
Items	アイテムプロトタイプ一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数は灰色で表示されます。
Triggers	トリガープロトタイプ一覧へのリンクが表示されます。 既存のトリガープロトタイプ数は灰色で表示されます。
Graphs	グラフプロトタイプ一覧へのリンクが表示されます。 既存のグラフプロトタイプ数は灰色で表示されます。
Hosts	ホストプロトタイプ一覧へのリンクが表示されます。 既存のホストプロトタイプ数は灰色で表示されます。
Key	ディスカバリに使用されるアイテムキーが表示されます。
Interval	ディスカバリを実行する頻度が表示されます。
Type	ディスカバリに使用されるアイテムタイプが表示されます（Zabbix エージェント、SNMP エージェントなど）。
Status	ディスカバリルールのステータスが表示されます - 有効または 無効。ステータスをクリックすると、有効から 無効（およびその逆）に変更できます。

新しいローレベルディスカバリルールを設定するには、右上隅の ディスカバリルールの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

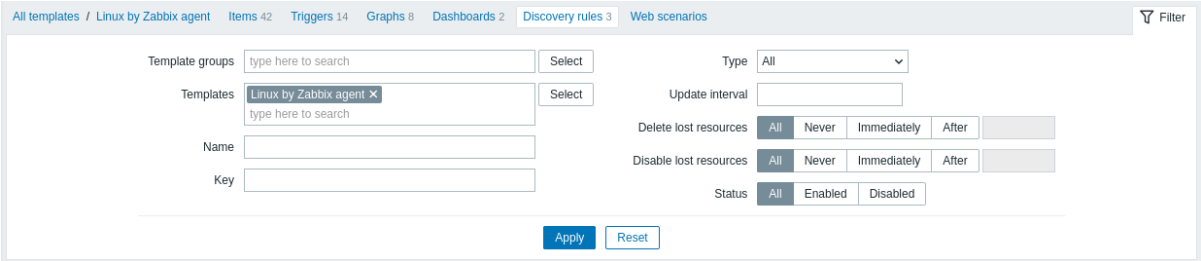
- 有効 - 低レベルのディスカバリルールのステータスを有効に変更します
- 無効 - 低レベルのディスカバリルールのステータスを無効に変更します
- 削除 - 低レベルのディスカバリルールを削除します

これらのオプションを使用するには、ディスカバリルール先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるディスカバリルールのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルターアイコンは右上隅にあります。これをクリックするとフィルターが開き、テンプレート、ディスカバリルール名、アイテムキー、アイテムタイプなど、必要な絞り込み条件を指定できます。



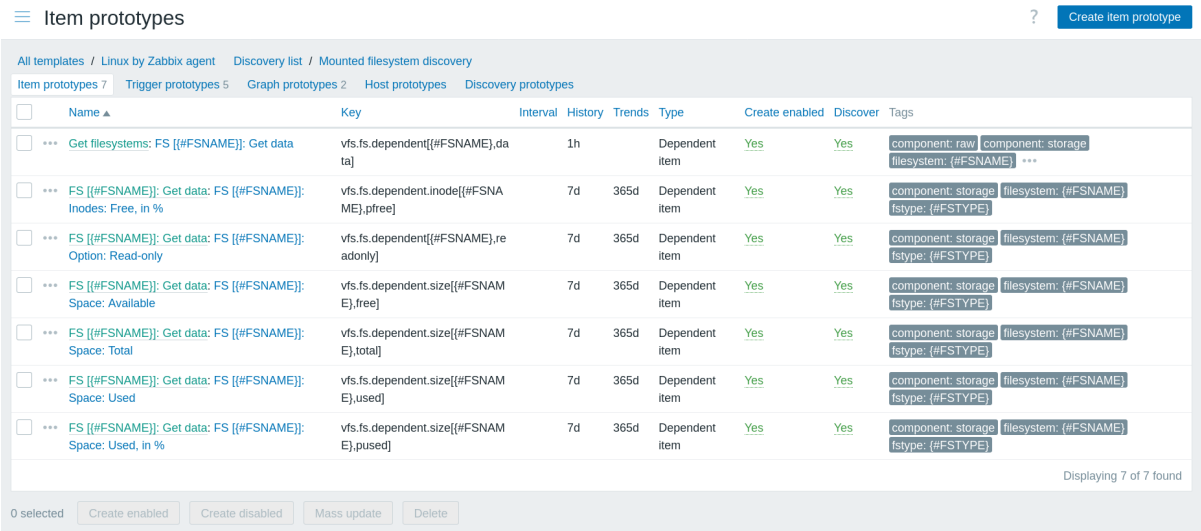
パラメータ	説明
テンプレートグループ	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループも暗黙的に選択されます。
テンプレート	1 つ以上のテンプレートで絞り込みます。
名前	ディスカバリルール名で絞り込みます。
キー	ディスカバリアイテムキーで絞り込みます。
タイプ	ディスカバリアイテムタイプで絞り込みます。
更新間隔	更新間隔で絞り込みます。 Zabbix トラッパーおよび依存アイテムでは使用できません。
失われたリソースの削除	失われたリソースの削除期間で絞り込みます。
失われたリソースの無効化	失われたリソースの無効化期間で絞り込みます。
ステータス	ディスカバリルールのステータス（すべて/有効/無効）で絞り込みます。

1 アイテムのプロトタイプ

概要

このセクションでは、テンプレート上のローレベルディスカバリルールに設定されたアイテムプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、アイテムプロトタイプは、ローレベルディスカバリ中に実際のホストアイテムを作成するための基礎となります。



表示されるデータ:

Column	Description
Name	アイテムプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、アイテムプロトタイプの設定フォームが開きます。 アイテムプロトタイプがリンクされたテンプレートに属している場合、アイテムプロトタイプ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのアイテムプロトタイプ一覧が開きます。 アイテムプロトタイプが「Dependent item」タイプの場合、アイテムプロトタイプ名の前にマスターアイテム名が緑色で表示されます（ホストレベルのアイテムプロトタイプの場合と同様です）。
Key	アイテムプロトタイプのキーが表示されます。

Column	Description
Interval	チェックの実行間隔が表示されます。
History	アイテムデータの履歴を何日間保持するかが表示されます。
Trends	アイテムのトレンド履歴を何日間保持するかが表示されます。
Type	アイテムプロトタイプのタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
Create enabled	このプロトタイプに基づいてアイテムを次の状態で作成します: **Yes** - 有効 **No** - 無効。クリックすると「Yes」と「No」を切り替えられます。
Discover	このプロトタイプに基づいてアイテムをディスカバリします: **Yes** - ディスカバリする **No** - ディスカバリしない。クリックすると「Yes」と「No」を切り替えられます。
Tags	アイテムプロトタイプのタグが表示されます。

新しいアイテムプロトタイプを設定するには、右上隅にある *Create item prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 有効として作成 - これらのアイテムを有効として作成します
- 無効として作成 - これらのアイテムを無効として作成します
- 一括更新 - これらのアイテムのプロトタイプを一括更新します
- 削除 - これらのアイテムのプロトタイプを削除します

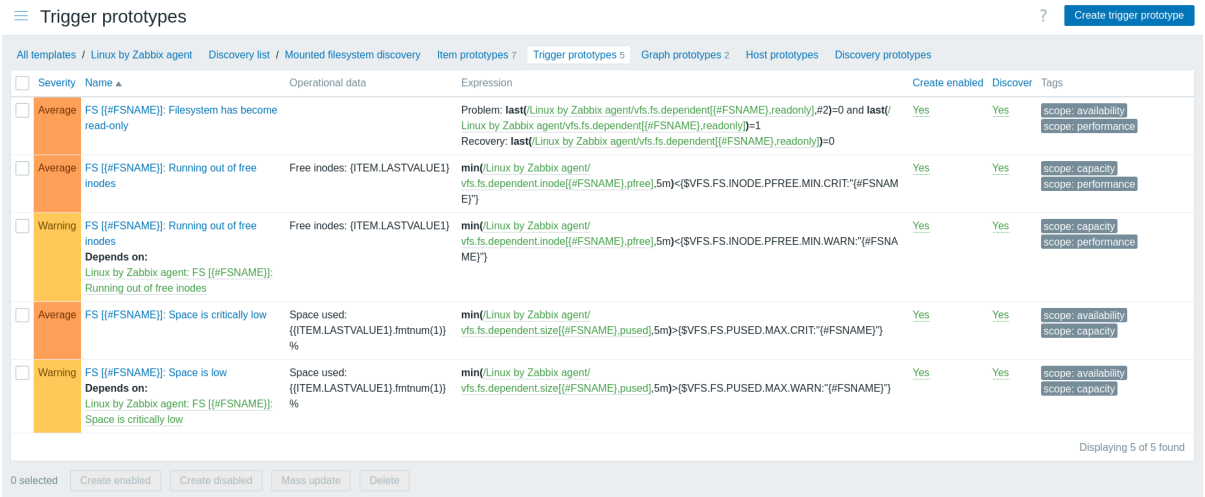
これらのオプションを使用するには、対象のアイテムのプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

2 トリガーのプロトタイプ

概要

このセクションでは、テンプレート上の低レベルディスカバリルールに設定されたトリガープロトタイプが表示されます。

テンプレートがホストにリンクされている場合、トリガープロトタイプは、低レベルディスカバリ中に実際のホストトリガーを作成するための基礎となります。



表示されるデータ:

Column	Description
Severity	トリガーの深刻度が、名前とセルの背景色の両方で表示されます。
Name	トリガープロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、トリガープロトタイプの設定フォームが開きます。 トリガープロトタイプがリンクされたテンプレートに属している場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのトリガープロトタイプ一覧が開きます。

Column	Description
Expression	トリガー条件式が表示されます。条件式内のテンプレート-アイテム部分はリンクとして表示され、アイテムの設定フォームに移動できます。
Operational data	トリガーの運用データの形式が表示されます。これには任意の文字列とマクロを含めることができ、*Monitoring* → *Problems* で動的に展開されます。
Create enabled	このプロトタイプに基づいてトリガーを次の状態で作成します: **Yes** - 有効 **No** - 無効。クリックすることで'Yes' と'No' を切り替えることができます。
Discover	このプロトタイプに基づくトリガーをディスカバリするかどうか: **Yes** - ディスカバリする **No** - ディスカバリしない。クリックすることで'Yes' と'No' を切り替えることができます。
Tags	トリガープロトタイプのタグが表示されます。

新しいトリガープロトタイプを設定するには、右上隅にある *Create trigger prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- *Create enabled* - これらのトリガーを有効として作成します
- *Create disabled* - これらのトリガーを無効として作成します
- *Mass update* - これらのトリガープロトタイプを一括更新します
- *Delete* - これらのトリガープロトタイプを削除します

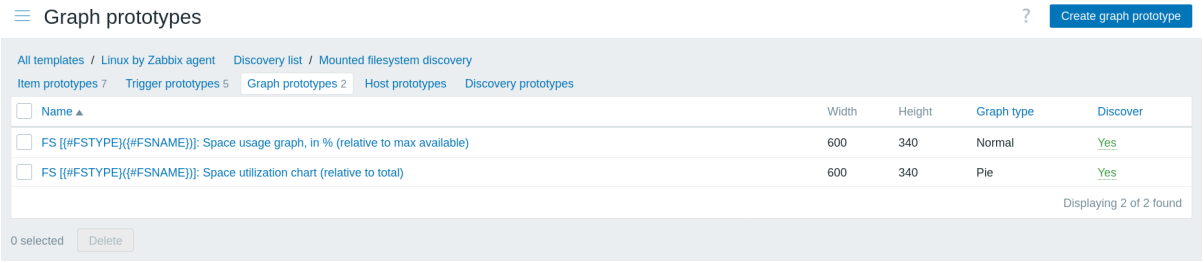
これらのオプションを使用するには、該当するトリガープロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

3 グラフのプロトタイプ

概要

このセクションでは、テンプレート上のローレベルディスカバリルールに設定されたグラフプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、グラフプロトタイプは、ローレベルディスカバリ中に実際のホストグラフを作成するための基礎となります。



表示されるデータ:

Column	Description
Name	グラフプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、グラフプロトタイプの設定フォームが開きます。 グラフプロトタイプがリンクされたテンプレートに属している場合、グラフ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのグラフプロトタイプ一覧が開きます。
Width	グラフプロトタイプの幅が表示されます。
Height	グラフプロトタイプの高さが表示されます。
Type	グラフプロトタイプのタイプが表示されます。- *Normal*、*Stacked*、*Pie*、または *Exploded*。
Discover	このプロトタイプに基づいてグラフをディスカバリします: **Yes** - ディスカバリする **No** - ディスカバリしない。**Yes** と **No** は、クリックすることで切り替えられます。

新しいグラフプロトタイプを設定するには、右上隅にある *Create graph prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- 削除 - これらのグラフプロトタイプを削除します

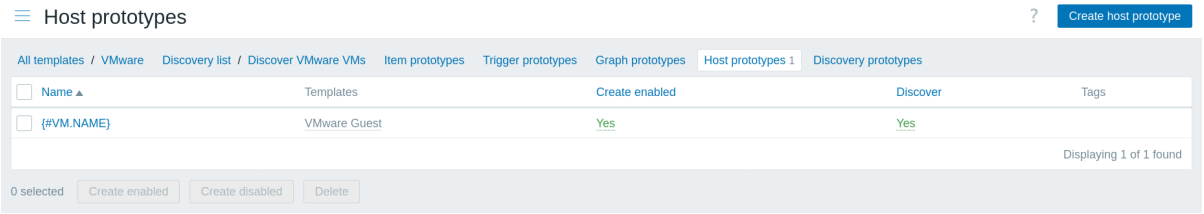
これらのオプションを使用するには、対象のグラフプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

4 ホストのプロトタイプ

概要

このセクションでは、テンプレート上のローレベルディスカバリルールに設定されたホストプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ホストプロトタイプは、ローレベルディスカバリ中に実際のホストを作成するための基礎となります。



表示されるデータ:

Column	Description
Name	ホストプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、ホストプロトタイプの設定フォームが開きます。 ホストプロトタイプがリンクされたテンプレートに属している場合、ホスト名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのホストプロトタイプ一覧が開きます。
Templates	ホストプロトタイプのテンプレートが表示されます。
Create enabled	このプロトタイプに基づいてホストを次の状態で作成します: **Yes** - 有効 **No** - 無効。**Yes** と **No** はクリックして切り替えることができます。
Discover	このプロトタイプに基づいてホストをディスカバリします: **Yes** - ディスカバリする **No** - ディスカバリしない。**Yes** と **No** はクリックして切り替えることができます。
Tags	ホストプロトタイプのタグが表示されます。

新しいホストプロトタイプを設定するには、右上隅にある *Create host prototype* ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- 自動生成時のステータス有効 - ホストを有効として作成します
- 自動生成無効 - ホストを無効として作成します
- 削除 - プロトタイプホストを削除する

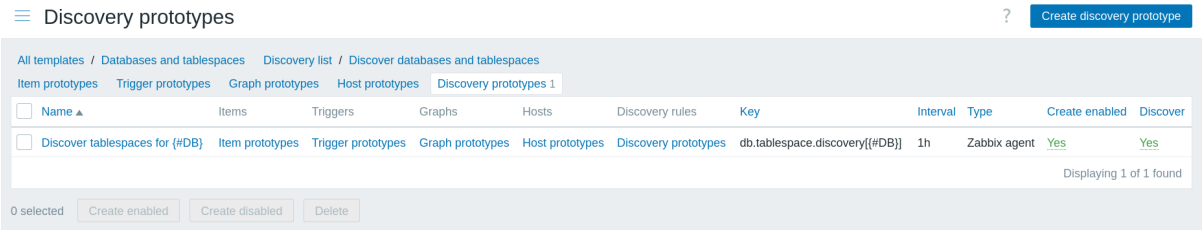
これらのオプションを使用するには、プロトタイプホストの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

5 ディスカバリのプロトタイプ

概要

このセクションでは、テンプレート上のローレベルディスカバリルールに設定されたディスカバリプロトタイプが表示されます。

テンプレートがホストにリンクされている場合、ディスカバリプロトタイプは、親ディスカバリルールの実際のディスカバリプロトタイプを作成するための基礎となります。



表示されるデータ:

Column	Description
Name	ディスカバリプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、ディスカバリプロトタイプの設定フォームが開きます。 ディスカバリプロトタイプがリンクされたテンプレートに属している場合、ディスカバリプロトタイプ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのディスカバリプロトタイプ一覧が開きます。
Items	アイテムプロトタイプ一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数は灰色で表示されます。
Triggers	トリガープロトタイプ一覧へのリンクが表示されます。 既存のトリガープロトタイプ数は灰色で表示されます。
Graphs	グラフプロトタイプ一覧へのリンクが表示されます。 既存のグラフプロトタイプ数は灰色で表示されます。
Hosts	ホストプロトタイプ一覧へのリンクが表示されます。 既存のホストプロトタイプ数は灰色で表示されます。
Discovery rules	ディスカバリプロトタイプ一覧へのリンクが表示されます。 既存のディスカバリプロトタイプ数は灰色で表示されます。
Key	ディスカバリに使用されるアイテムキーが表示されます。
Interval	ディスカバリを実行する頻度が表示されます。
Type	ディスカバリに使用されるアイテムタイプが表示されます（Zabbix エージェント、SNMP エージェントなど）。
Create enabled	ディスカバリプロトタイプの作成ステータスが表示されます - 作成有効 (*Yes*) または作成無効 (*No*)。 ステータスをクリックすると、*Yes* から *No*（およびその逆）に変更できます。
Discover	ディスカバリプロトタイプの検出ステータスが表示されます - 検出する (*Yes*) または検出しない (*No*)。ステータスをクリックすると、*Yes* から *No*（およびその逆）に変更できます。

新しいディスカバリプロトタイプを設定するには、右上隅にある *Create discovery prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- *Create enabled* - これらのディスカバリプロトタイプを *Enabled* として作成します
- *Create disabled* - これらのディスカバリプロトタイプを *Disabled* として作成します
- *Delete* - これらのディスカバリプロトタイプを削除します

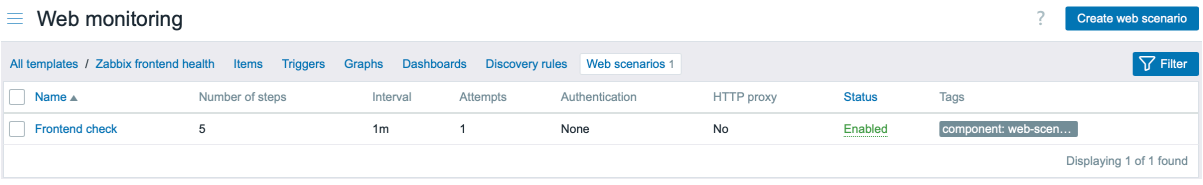
これらのオプションを使用するには、対象のディスカバリプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

5 Web シナリオ

概要

テンプレートのWeb シナリオ一覧には、データ収集 → テンプレートから、対象のテンプレートの *Web* をクリックしてアクセスできます。

既存の Web シナリオの一覧が表示されます。



表示されるデータ:

Column	Description
Name	Web シナリオの名前です。Web シナリオ名をクリックすると、Web シナリオの設定フォームが開きます。 Web シナリオが別のテンプレートから継承されている場合、Web シナリオ名の前にテンプレート名が灰色のリンクで表示されます。テンプレートリンクをクリックすると、そのテンプレートレベルの Web シナリオ一覧が開きます。
Number of steps	シナリオに含まれるステップ数です。
Update interval	シナリオの実行間隔です。
Attempts	Web シナリオのステップ実行時に行われる試行回数です。
Authentication	認証方式が表示されます - Basic、NTLM、Kerberos、Digest、または None。
HTTP proxy	HTTP プロキシ、または使用していない場合は「No」を表示します。
Status	Web シナリオのステータスが表示されます - *Enabled* または *Disabled*。 ステータスをクリックすると変更できます。
Tags	Web シナリオのタグが表示されます。 最大 3 つのタグ（name:value の組）が表示されます。タグがそれより多い場合は、「...」リンクが表示され、マウスオーバーですべてのタグを確認できます。

新しい Web シナリオを設定するには、右上隅にある *Web* シナリオの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

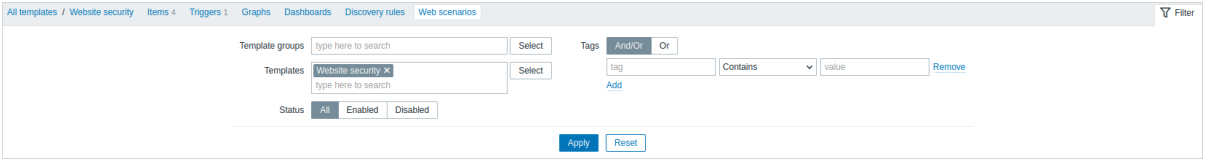
- 有効 - シナリオのステータスを有効にします
- 無効 - シナリオのステータスを無効にします
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、Web シナリオ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるシナリオのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

Filter リンクは Web シナリオの一覧の上部にあります。これをクリックするとフィルターが表示され、テンプレートグループ、テンプレート、ステータス、タグでシナリオを絞り込むことができます。

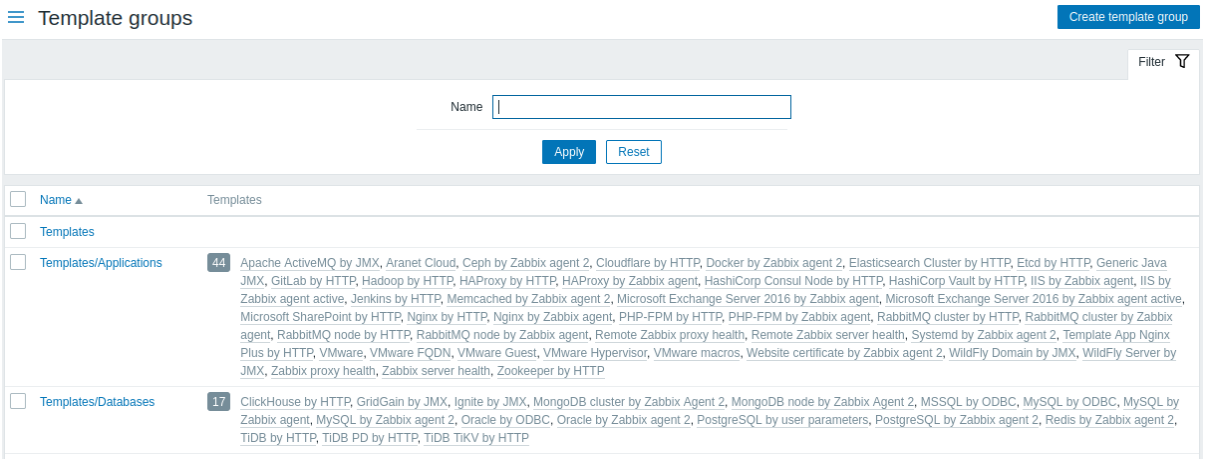


1 テンプレートグループ

概要

データ収集 → テンプレートグループセクションでは、ユーザーはテンプレートグループを設定および管理できます。

既存のテンプレートグループの一覧とその詳細が表示されます。テンプレートグループは名前で検索およびフィルタリングできます。



表示されるデータ:

列	説明
名前	テンプレートグループの名前です。グループ名をクリックすると、グループの設定フォームが開きます。
テンプレート	グループ内のテンプレート数（灰色で表示）の後に、グループメンバーの一覧が表示されます。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。 数字をクリックすると、このグループ内のテンプレート一覧が開きます。

一括編集オプション

複数のテンプレートグループを一度に削除するには、該当するグループの前にあるチェックボックスを選択し、一覧の下にある [削除] ボタンをクリックします。

フィルターの使用

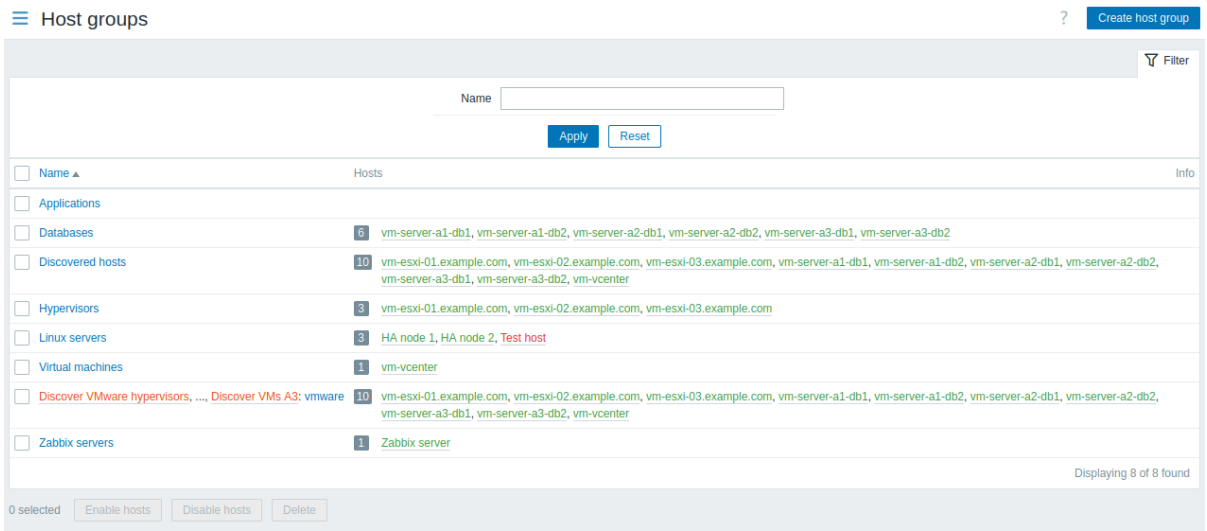
フィルターを使用して、関心のあるテンプレートグループのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

2 ホストグループ

概要

データ収集 → ホストグループセクションでは、ユーザーはホストグループを設定および管理できます。

既存のホストグループの一覧とその詳細が表示されます。ホストグループは名前で検索およびフィルタリングできます。



表示されるデータ:

Column	Description
Name	ホストグループの名前です。グループ名をクリックすると、グループの設定フォームが開きます。 ディスカバリされたホストグループは、プレフィックスとしてローレベルディスカバリルール名付きで表示されます。LLD ルール名をクリックすると、ホストプロトタイプの設定フォームが開きます。ディスカバリされたホストグループは、それをディスカバリしたすべての LLD ルールが削除されると削除されることに注意してください。
Hosts	グループ内のホスト数（グレーで表示）の後に、グループメンバーの一覧が表示されます。 ホスト名をクリックすると、ホストの設定フォームが開きます。 数値をクリックすると、ホスト一覧全体で、そのグループに属するホストがフィルタリングされます。
Info	ホストグループに関するエラー情報（存在する場合）が表示されます。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- ホストを有効 - グループに所属するホストのステータスを監視対象にします

- ホストを無効 - グループに所属するホストのステータスを監視対象外にします
- 削除 - ホストグループをを削除します

これらのオプションを使用するには、それぞれのホストグループの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

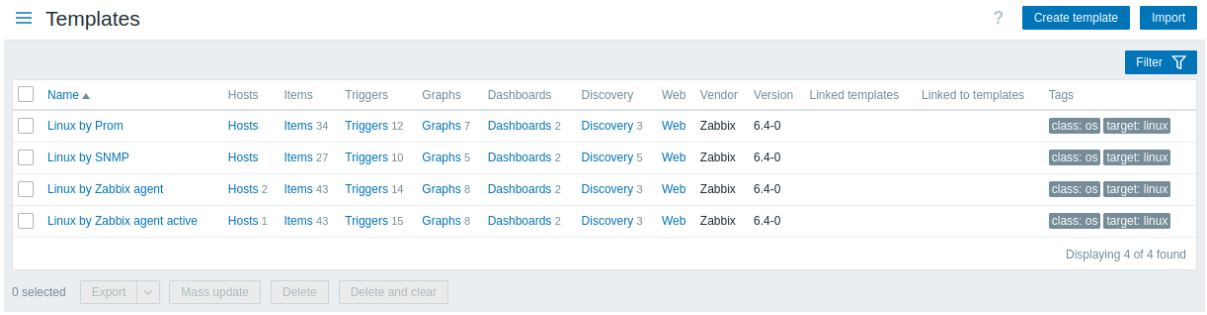
フィルタを使用して、任意のホストグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

3 テンプレート

概要

データ収集 → テンプレートセクションでは、ユーザーはテンプレートを設定および管理できます。

既存のテンプレートの一覧とその詳細が表示されます。



表示されるデータ:

Column	Description
Name	テンプレートの名前です。 テンプレート名をクリックすると、テンプレートの設定フォームが開きます。
Hosts	テンプレートがリンクされている編集可能なホストの数です。読み取り専用のホストは含まれません。 *Hosts* をクリックすると、そのテンプレートにリンクされているホストのみに絞り込まれたホスト一覧が開きます。
Entities (Items, Triggers, Graphs, Dashboards, Discovery, Web)	テンプレート内の各エンティティの数です（灰色で表示されます）。 エンティティ名をクリックすると、そのエンティティの一覧全体の中から、テンプレートに属するものだけが絞り込まれて表示されます。
Linked templates	そのテンプレートにリンクされているテンプレートです。
Linked to templates	そのテンプレートがリンクされているテンプレートです。
Vendor, *Version*	テンプレートのベンダーおよびバージョンです。テンプレート設定にその情報が含まれている場合に表示され、さらに標準搭載テンプレート、インポートされたテンプレート、またはTemplate API を通じて変更されたテンプレートに対してのみ表示されます。 標準搭載テンプレートでは、バージョンは次の形式で表示されます: Zabbix のメジャーバージョン、区切り文字（”-”）、リビジョン番号（テンプレートの新しいバージョンごとに増加し、Zabbix の各メジャーバージョンごとにリセットされます）。例: 7.0-0、7.0-3、7.4-0、7.4-3。
Tags	マクロが未解決の状態で表示される、テンプレートのタグです。

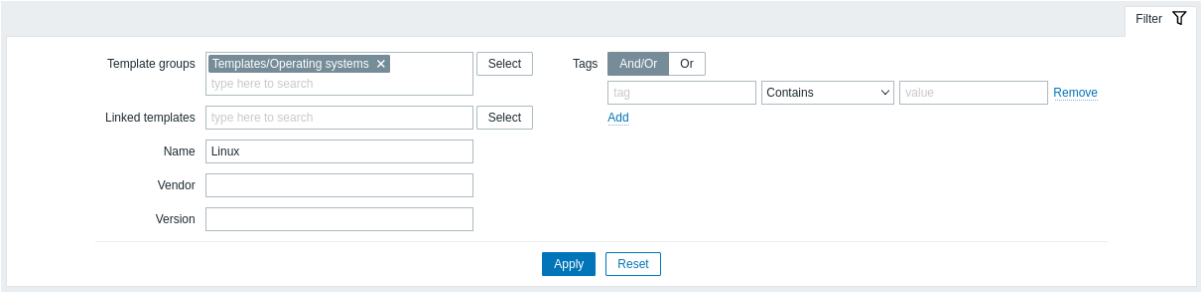
新しいテンプレートを設定するには、右上隅にある *Create template* ボタンをクリックします。

YAML、XML、または JSON ファイルからテンプレートをインポートするには、右上隅にある *Import* ボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるテンプレートのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルターリンクは、テンプレートの作成ボタンと インポートボタンの下にあります。これをクリックするとフィルターが表示され、テンプレートグループ、リンクされたテンプレート、名前、タグでテンプレートを絞り込めます。



Parameter	Description
Template groups	1 つ以上のテンプレートグループで絞り込みます。 親テンプレートグループを指定すると、ネストされたすべてのグループも暗黙的に選択されます。
Linked templates	直接リンクされているテンプレートで絞り込みます。
Name	テンプレート名で絞り込みます。
Vendor	テンプレートのベンダーで絞り込みます。
Version	テンプレートのバージョンで絞り込みます。
Tags	テンプレートのタグ名と値で絞り込みます。 絞り込み可能なのはテンプレートレベルのタグのみで、継承されたタグは対象外です。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致は常に大文字と小文字を区別します。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- エクスポート - テンプレートを YAML、XML、または JSON ファイルにエクスポートします。
- 一括更新 - 複数のテンプレートに対して、いくつかのプロパティを一度に更新します。
- 削除 - テンプレートを削除しますが、そのリンクされたエンティティ（アイテム、トリガーなど）はホストに残します。
- 削除してクリア - テンプレートと、そのリンクされたエンティティをホストから削除します。

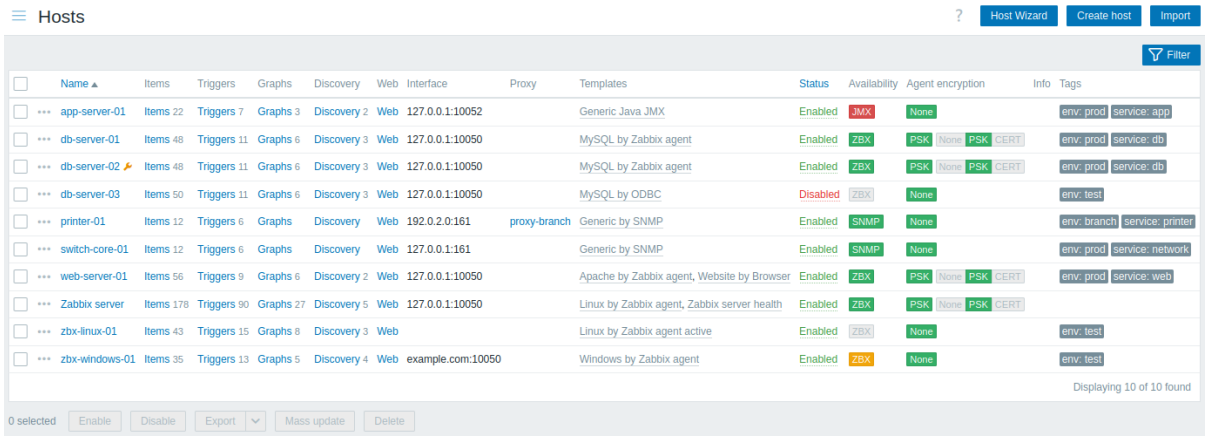
これらのオプションを使用するには、対象のテンプレートの前にあるチェックボックスを選択し、必要なボタンをクリックします。

4 ホスト

概要

データ収集 → ホストセクションでは、ユーザーはホストの設定および管理を行えます。

既存のホストの一覧とその詳細が表示されます。



表示されるデータ:

列	説明
ホストメニュー	三点アイコンをクリックすると、ホストメニューが開きます。
名前	ホストの名前です。 ホスト名をクリックすると、ホストの設定フォームが開きます。 ホスト名の後ろにあるオレンジ色のレンチアイコン は、このホストがメンテナンス中であることを示します。アイコンにマウスポインターを合わせると、メンテナンスの詳細が表示されます。
エンティティ *(アイテム、トリガー、グラフ、ディスカバリ、Web)*	エンティティ名をクリックすると、そのホストのアイテム、トリガーなどが表示されます。 各エンティティの数は灰色で表示されます。
インターフェース	ホストのメインインターフェースが表示されます。
プロキシ	割り当てられたプロキシがこの列に表示されます: <Proxy name> - ホストはスタンドアロンのプロキシによって監視されています（そのプロキシがプロキシグループの一部である場合でも同様です）。 <Proxy group name: proxy name> - ホストはプロキシグループによって監視されており、Zabbix server がそのホストを監視するプロキシを割り当てています。 <Proxy group name> - ホストはプロキシグループによって監視されていますが、プロキシが存在しない場合、または Zabbix server がそのホストを監視するプロキシを割り当てていない場合です。 何も表示されない - ホストはプロキシにもプロキシグループにも監視されていません。 この列は、監視対象フィルターオプションが「Any」、「Proxy」、または「Proxy group」に設定されている場合にのみ表示されます。
テンプレート	ホストにリンクされたテンプレートが表示されます。 リンクされたテンプレートに他のテンプレートが含まれている場合、それらは括弧内にカンマ区切りで表示されます。 テンプレート名をクリックすると、その設定フォームが開きます。
ステータス	ホストのステータスが表示されます - 有効または 無効。 ステータスをクリックすると、手動で変更できます。 失われた検出済みホストには情報アイコンが表示されます。ツールチップのテキストには、そのステータスの詳細が表示されます。

列	説明
可用性	設定されたインターフェースごとのホストの可用性が表示されます。 可用性アイコンは、Zabbix server 上でのホストインターフェースの現在のステータスを表します。そのため、Web インターフェースでホストを無効にした場合、その可用性は Zabbix server が設定変更を同期した後に更新されます。同様に、ホストを有効にした場合、その可用性は Zabbix server が設定変更を同期し、ホストをポーリングした後に更新されます。 可用性アイコンは、設定されているインターフェースタイプ（Agent、SNMP、IPMI、JMX）のみを表します。 アイコンにマウスを合わせると、同じタイプのすべてのインターフェースの一覧と、その詳細、ステータス、エラーを含むポップアップが表示されます。Agent インターフェースでは、ポップアップにインターフェース（パッシブ）とアクティブチェックが表示されます。ホストにアクティブチェックしかない場合、ホストに Agent インターフェースが設定されていなくても、Agent インターフェースアイコンが表示されます。 インターフェースを持たないホストでは、この列は空になります。 単一のホストインターフェースのステータスは、そのインターフェースを使用する有効なアイテムとホストとの接続によって決まります。ステータスは次のいずれかです: 利用可能 - ホストへの接続に成功しました。 利用不可 - ホストへの接続に失敗しました（タイムアウト、ファイアウォールの問題など）。 不明 - ホストへの接続がまだ試行されていないか、結果が不明です。 Zabbix server がインターフェースのステータスをどのように判定するかの詳細については、不明なインターフェースステータス および到達不能/利用不可ホストインターフェース設定 を参照してください。 単一タイプ（Agent、SNMP、IPMI、JMX）のすべてのホストインターフェースのステータスは、少なくとも 1 つの有効なアイテムで使用されているインターフェースによって決まります。ステータスはアイコンの色で示されます: 緑 - すべてのインターフェースが利用可能です。 黄 - 少なくとも 1 つのインターフェースが利用不可であり、かつ少なくとも 1 つが利用可能または不明です。 赤 - すべてのインターフェースが利用不可です。 灰 - 少なくとも 1 つのインターフェースが不明ですが、利用不可のものはありません。 アクティブチェックの可用性。ホスト上で少なくとも 1 つのアクティブチェックが有効な場合、アクティブチェックも上記の説明どおり Agent インターフェース全体の可用性に影響します。アクティブチェックの可用性を判定するために、heartbeat メッセージがエージェントのアクティブチェックスレッドで送信されます。heartbeat メッセージの頻度は、Zabbix agent またはagent 2 の設定にある `HeartbeatFrequency` パラメータで制御されます（デフォルトは 60 秒、範囲は 0～3600）。アクティブチェック heartbeat が 2 x HeartbeatFrequency 秒より古い場合、アクティブチェックは利用不可と見なされます。 注**:** バージョン 6.2.x より古い Zabbix エージェントはアクティブチェック heartbeat を送信しないため、それらのホストの可用性は不明のままになります。
エージェント暗号化	ホストへの接続およびホストからの接続に対する暗号化ステータスが表示されます: **None** - 暗号化なし。 **PSK** - 事前共有鍵を使用。 **Cert** - 証明書を使用。
情報	ホストに関するエラー情報（存在する場合）が表示されます。
タグ	マクロが未解決のホストのタグです。

新しいホストを設定するには、右上隅の *Host Wizard* または *Create host* ボタンをクリックします。YAML、XML、または JSON ファイルからホストをインポートするには、右上隅の *Import* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

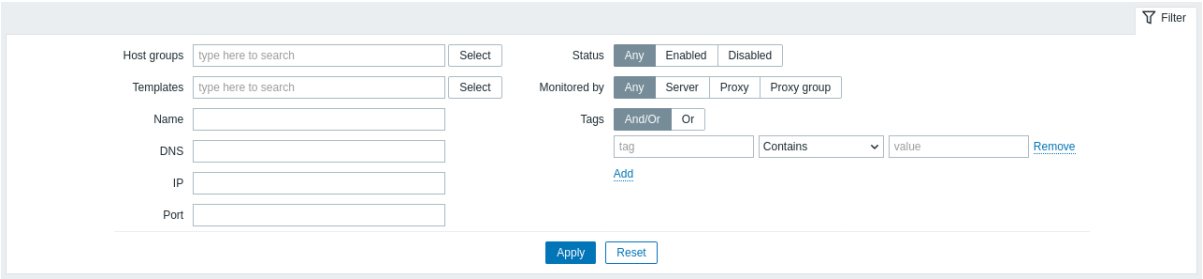
- 有効化 - ホストのステータスを 監視中に変更します。
- 無効化 - ホストのステータスを 監視しないに変更します。
- エクスポート - ホストを YAML、XML、または JSON ファイルにエクスポートします。
- 一括更新 - 複数のホストのいくつかのプロパティを一度に更新します。
- 削除 - ホストを削除します。

これらのオプションを使用するには、対象のホストの前にあるチェックボックスを選択し、その後、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるホストのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

Filter アイコンは右上隅にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



Parameter	Description
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。
Templates	リンクされたテンプレートで絞り込みます。
Name	表示用ホスト名で絞り込みます。
DNS	DNS 名で絞り込みます。
IP	IP アドレスで絞り込みます。
Port	ポート番号で絞り込みます。
Status	ホストのステータスで絞り込みます。
Monitored by	Zabbix server、プロキシ、またはプロキシグループによって監視されているホストで絞り込みます。
Proxies	ここで指定したプロキシによって監視されているホストで絞り込みます。このフィールドは、*Monitored by* フィールドで「Proxy」が選択されている場合にのみ使用できます。
Proxy groups	ここで指定したプロキシグループによって監視されているホストで絞り込みます。このフィールドは、*Monitored by* フィールドで「Proxy group」が選択されている場合にのみ使用できます。
Tags	ホストタグ名と値で絞り込みます。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

不明なインターフェースステータス

Zabbix サーバーは、以下の場合にホストインターフェース（エージェント、SNMP、IPMI、JMX）に対して「不明」ステータスを表示します。

- ホストが無効になっている。
- ホストがプロキシ、別のプロキシ、またはサーバーによって監視されるよう設定されている（以前にプロキシによって監視されていた場合）。
- ホストがオフラインと思われるプロキシによって監視されている（最大ハートビート間隔である 1 時間の間、プロキシから更新を受信していない）。
- そのインターフェースタイプを持つホストのすべてのアイテムが無効になっている。

- そのインターフェースタイプ用のポーラーが設定されていない（たとえば、サーバー設定パラメータStartAgentPollers またはStartSNMPPollers が 0 に設定されている）。

インターフェースの可用性は、Zabbix サーバーの設定キャッシュ同期後に「不明」に設定されます。

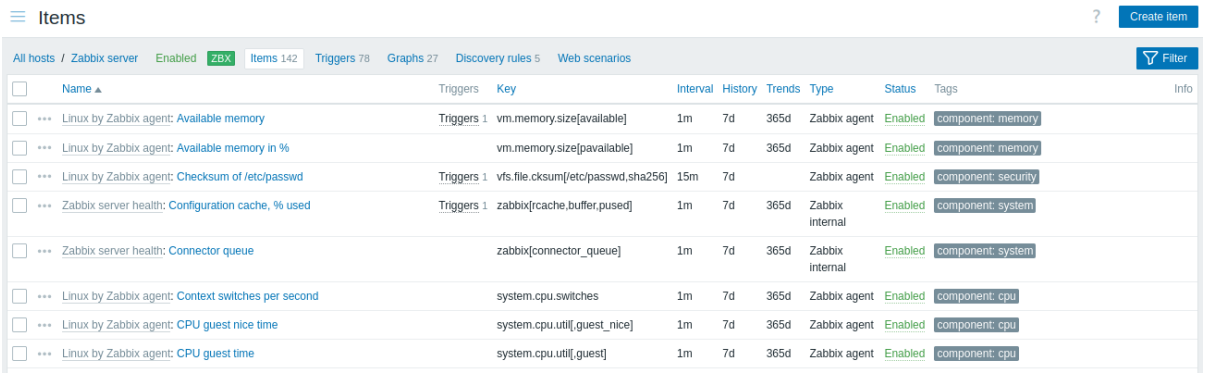
プロキシによって監視されているホストのインターフェース可用性（利用可能/利用不可）は、プロキシの設定キャッシュ同期後に復元されます。

1 アイテム

概要

ホストのアイテム一覧は、データ収集 → ホストから対象のホストの アイテムをクリックしてアクセスできます。

既存のアイテムの一覧が表示されます。



表示されるデータ:

列	説明
アイテムメニュー	三点アイコンをクリックすると、アイテムメニューが開きます。
ホスト	アイテムのホストです。 ホスト名をクリックすると、ホストの設定フォームが開きます。 この列は、フィルターで複数のホストが選択されている場合、またはホストが選択されていない場合にのみ表示されます。
名前	アイテム名が、アイテムの詳細への青いリンクとして表示されます。 アイテム名のリンクをクリックすると、アイテムの設定フォームが開きます。 ホストのアイテムがテンプレートに属している場合、テンプレート名がアイテム名の前に灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのアイテム一覧が開きます。 アイテムがアイテムプロトタイプから作成されている場合、その名前の前に、オレンジ色でローレベルディスカバリルール名が表示されます。ディスカバリルール名をクリックすると、アイテムプロトタイプの一覧が開きます。
トリガー	マウスカーソルを トリガーの上に移動すると、アイテムに関連付けられたトリガーを表示する情報ボックスが表示されます。 トリガー数は灰色で表示されます。
キー	アイテムキーが表示されます。
間隔	チェックの頻度が表示されます。 なお、パッシブアイテムは 今すぐ実行ボタンを押すことでも即時にチェックできます。
履歴	アイテムデータの履歴が何日間保持されるかが表示されます。
トレンド	アイテムのトレンド履歴が何日間保持されるかが表示されます。
タイプ	アイテムタイプが表示されます（Zabbix エージェント、SNMP エージェント、シンプルチェックなど）。
ステータス	アイテムのステータスが表示されます。有効、無効、または 未サポートです。ステータスをクリックすると手動で変更できます。有効から 無効（およびその逆）、未サポートから 無効（およびその逆）に変更できます。 失われた検出済みアイテムには情報アイコンが表示されます。ツールチップのテキストには、そのステータスの詳細が表示されます。
タグ	アイテムタグが表示されます。 最大 3 つのタグ（name:value の組）が表示されます。タグがそれより多い場合は、”...” リンクが表示され、マウスオーバーですべてのタグを確認できます。

列	説明
情報	アイテムが正常に動作している場合、この列にはアイコンは表示されません。エラーがある場合は、文字"i" の付いた四角いアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいアイテムを設定するには、右上隅の アイテムの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

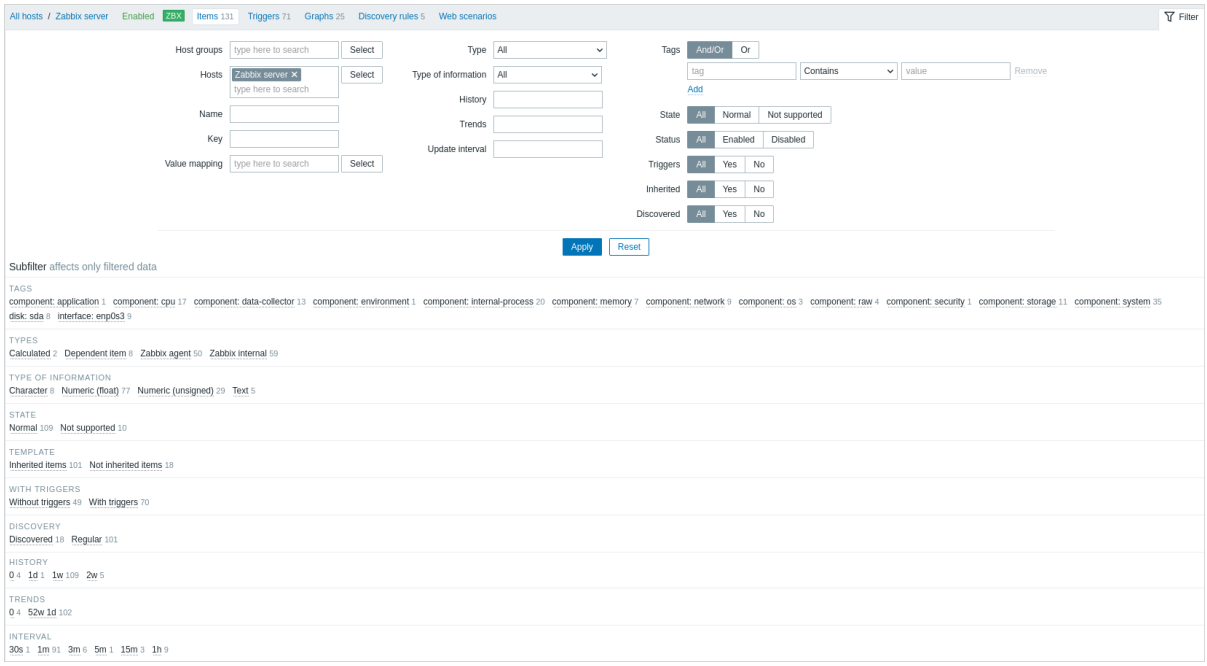
- 有効化 - アイテムのステータスを有効に変更します
- 無効化 - アイテムのステータスを無効に変更します
- 今すぐ実行 - 新しいアイテムの値のチェックを直ちに実行します。
  パッシブチェックでのみサポートされています（詳細はこちらを参照）。値を直ちにチェックする場合、設定キャッシュは更新されないため、値にはアイテム設定に対するごく最近の変更は反映されません。
- 履歴とトレンドをクリア - アイテムの履歴データおよびトレンドデータを削除します。
- コピー - アイテムを他のホストまたはテンプレートにコピーします。
- 一括更新 - 複数のアイテムのいくつかのプロパティを一度に更新します。
- 削除 - アイテムを削除します。

これらのオプションを使用するには、対象のアイテムの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用すると、関心のあるアイテムのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルタアイコンは右上隅にあります。これをクリックすると、目的のフィルタ条件を指定できるフィルタが開きます。



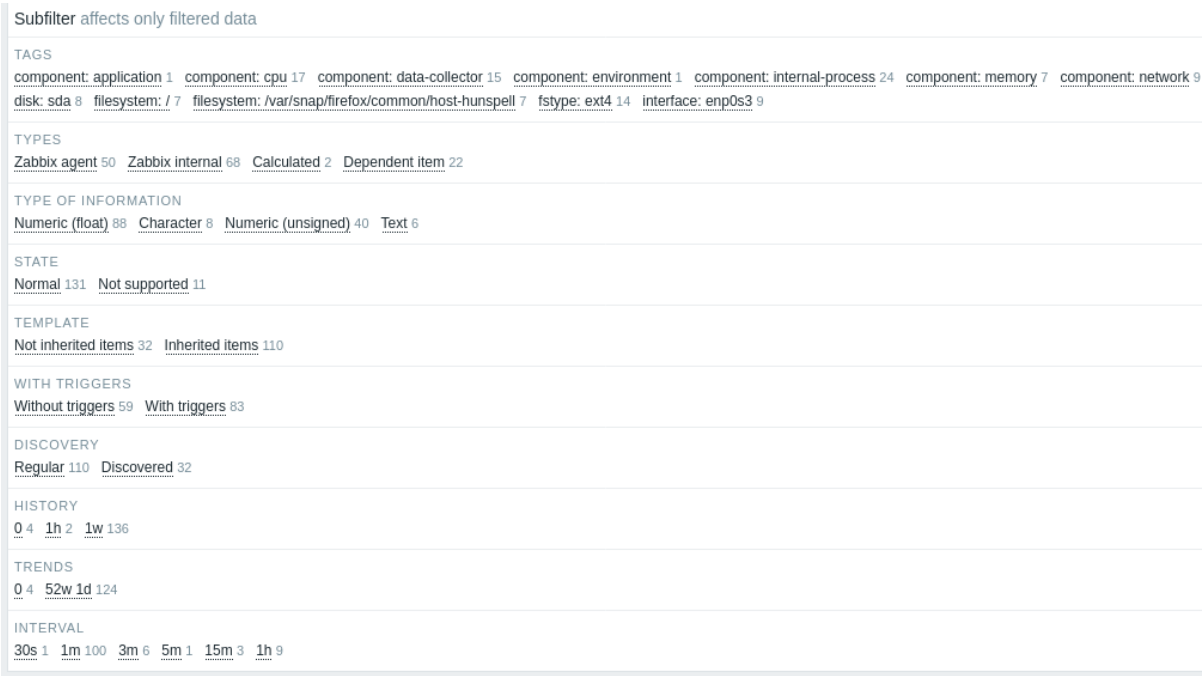
パラメータ	説明
ホストグループ	1 つ以上のホストグループでフィルタします。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
ホスト	1 つ以上のホストでフィルタします。
名前	アイテム名でフィルタします。
キー	アイテムキーでフィルタします。
値のマッピング	使用されている値のマップでフィルタします。 ホストオプションが空の場合、このパラメータは表示されません。
タイプ	アイテムタイプ（Zabbix エージェント、SNMP エージェントなど）でフィルタします。
情報の型	情報の型（符号なし数値、浮動小数点数など）でフィルタします。
履歴	アイテムの履歴を保持する期間でフィルタします。

パラメータ	説明
トレンド	アイテムのトレンドを保持する期間でフィルタします。
更新間隔	アイテムの更新間隔でフィルタします。
タグ	表示するアイテム数を絞り込むためのタグを指定します。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 存在する - 指定したタグ名を含めます 等しい - 指定したタグ名と値を含めます（大文字と小文字を区別） 含む - タグ値に入力した文字列を含む指定タグ名を含めます（部分一致、大文字と小文字を区別しない） 存在しない - 指定したタグ名を除外します 等しくない - 指定したタグ名と値を除外します（大文字と小文字を区別） 含まない - タグ値に入力した文字列を含む指定タグ名を除外します（部分一致、大文字と小文字を区別しない） 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - 1 つの条件を満たせば十分です
状態	アイテムの状態（正常または 未サポート）でフィルタします。
ステータス	アイテムのステータス（有効または 無効）でフィルタします。
トリガー	トリガーがある（またはない）アイテムでフィルタします。
継承	テンプレートから継承されたアイテム（または継承されていないアイテム）でフィルタします。
ディスカバリ	ローレベルディスカバリによって検出されたアイテム（または検出されていないアイテム）でフィルタします。

サブフィルターの使用

サブフィルターを使用すると、メインフィルターによる絞り込みをさらに調整できます。

サブフィルターには、関連するアイテムへすばやくアクセスするためのクリック可能なリンクが含まれています。アイテムは、共通のエンティティ（タグ、アイテムタイプ、アイテムの状態、アイテムの更新間隔など）によって関連付けられています。エンティティをクリックすると、そのエンティティは灰色の背景で強調表示され、アイテムは即座に絞り込まれます（メインフィルターで *Apply* をクリックする必要はありません）。別のエンティティをクリックすると、絞り込み結果に追加されます。同じエンティティをもう一度クリックすると、絞り込みが解除されます。



サブフィルターは、フィルター済みデータに基づいて生成されます。このデータは 1000 レコードに制限されています。サブフィルターでさらに多くのレコードを表示したい場合は、*Limit for search and filter results* パラメータの値を増やす必要があります（*Administration -> General -> GUI* 内）。

メインフィルターとは異なり、サブフィルターはテーブルの更新要求ごとに更新されるため、利用可能な絞り込みオプションとその件数に関する最新情報が常に反映されます。

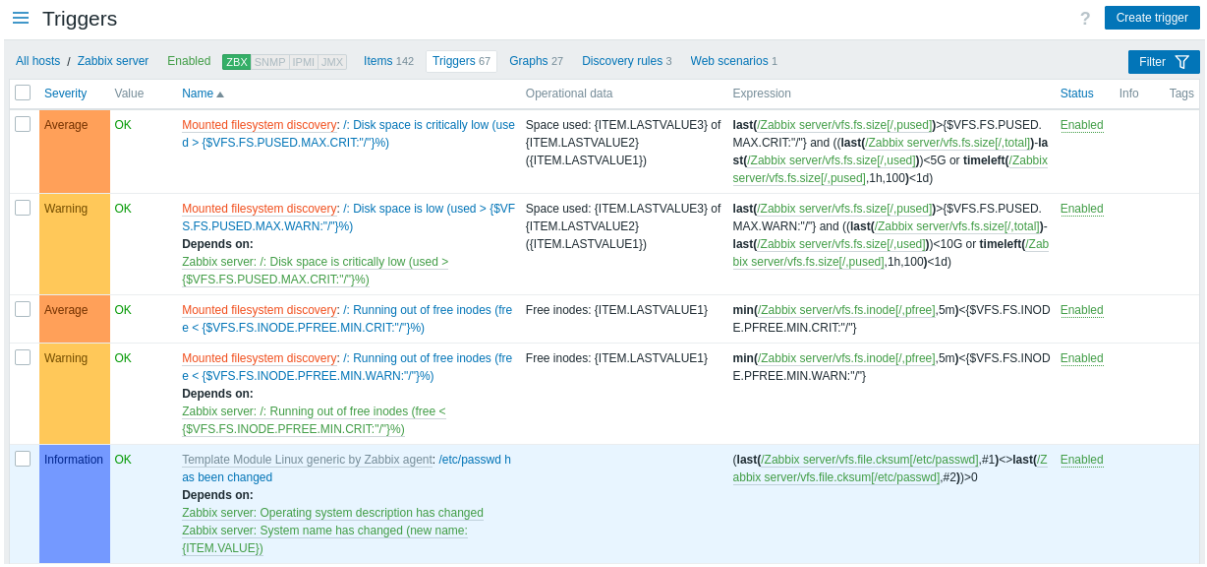
横方向に表示されるエンティティの数は 100 個までに制限されています。これを超える場合は、末尾に三点アイコンが表示されますが、クリックはできません。

各クリック可能なエンティティの横に表示される数字は、そこにグループ化されているアイテム数を示します（メインフィルターの結果に基づきます）。エンティティをクリックすると、他の利用可能なエンティティの数字にはプラス記号が付いて表示され、現在の選択に追加できるアイテム数を示します。アイテムを持たないエンティティは、以前にサブフィルターで選択されていない限り表示されません。

2 トリガー

概要

ホストのトリガー一覧は、データ収集 → ホストから、対象のホストの トリガーをクリックしてアクセスできます。



表示されるデータ:

Column	Description
Severity	トリガーの深刻度は、名前とセルの背景色の両方で表示されます。
Value	トリガーの値は次のように表示されます: **OK** - トリガーが OK 状態 **PROBLEM** - トリガーが障害状態
Host	トリガーのホスト。 ホスト名をクリックすると、ホストの設定フォームが開きます。 この列は、フィルターで複数のホストまたはホスト未選択の場合にのみ表示されます。
Name	トリガー名が、トリガーの詳細への青いリンクとして表示されます。 トリガー名のリンクをクリックすると、トリガーの設定フォームが開きます。 ホストのトリガーがテンプレートに属している場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのトリガー一覧が開きます。 トリガーがトリガープロトタイプから作成されている場合、その名前の前に低レベルディスカバリルール名がオレンジ色で表示されます。ディスカバリルール名をクリックすると、トリガープロトタイプ一覧が開きます。
Operational data	トリガーの運用データ定義が表示されます。これには任意の文字列とマクロを含めることができ、監視 → 障害で動的に展開されます。
Expression	トリガーの条件式が表示されます。条件式のホスト-アイテム部分はリンクとして表示され、アイテムの設定フォームに移動できます。
Status	トリガーのステータスが表示されます - *Enabled*、*Disabled*、または *Unknown*。ステータスをクリックすると、手動で変更できます - Enabled から Disabled（およびその逆）、Unknown から Disabled（およびその逆）。 無効化されたトリガーの障害は、Web インターフェースには表示されなくなりますが、削除はされません。 失われた検出済みトリガーには情報アイコンが表示されます。ツールチップのテキストには、そのステータスの詳細が示されます。

Column	Description
Info	すべてが正常に動作している場合、この列にはアイコンは表示されません。エラーがある場合は、文字「i」の付いた四角いアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。
Tags	トリガーにタグが含まれている場合、この列にタグ名と値が表示されます。

新しいトリガーを設定するには、右上隅の トリガーの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

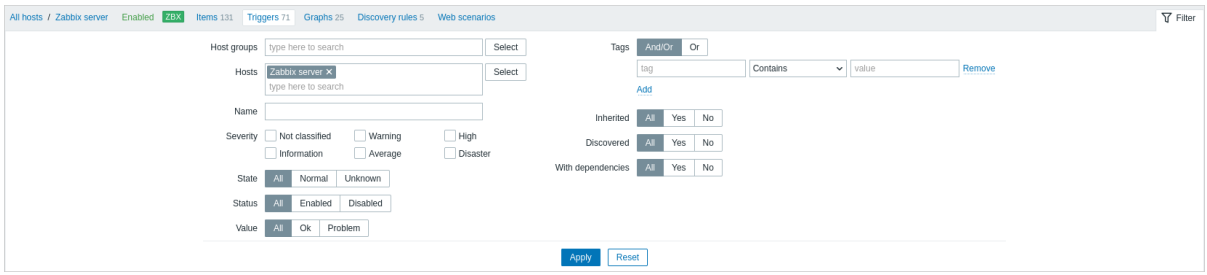
- 有効 - トリガーのステータスを有効に
- 無効 - トリガーのステータスを無効に
- コピー - トリガーを他のホストまたはテンプレートにコピーします
- 一括更新 - 一度に複数のトリガーのいくつかのプロパティを更新します
- 削除 - トリガーの削除

これらのオプションを使用するには、トリガーの先頭にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるトリガーのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルターアイコンは右上隅にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



Parameter	Description
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。 テンプレートのみを含むホストグループは選択できません。
Hosts	1 つ以上のホストで絞り込みます。 上でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに限定されます。
Name	トリガー名で絞り込みます。
Severity	1 つまたは複数のトリガーの深刻度で絞り込む場合に選択します。
State	トリガーの状態で絞り込みます。
Status	トリガーのステータスで絞り込みます。
Value	トリガーの値で絞り込みます。
Tags	トリガーのタグ名と値で絞り込みます。特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別） **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない） **Does not exist** - 指定したタグ名を除外します **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別） **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない） 条件には 2 つの計算タイプがあります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます **Or** - いずれか 1 つの条件を満たせば十分です タグ名フィールドとタグ値フィールドの両方で、マクロおよびマクロ関数をサポートしています。

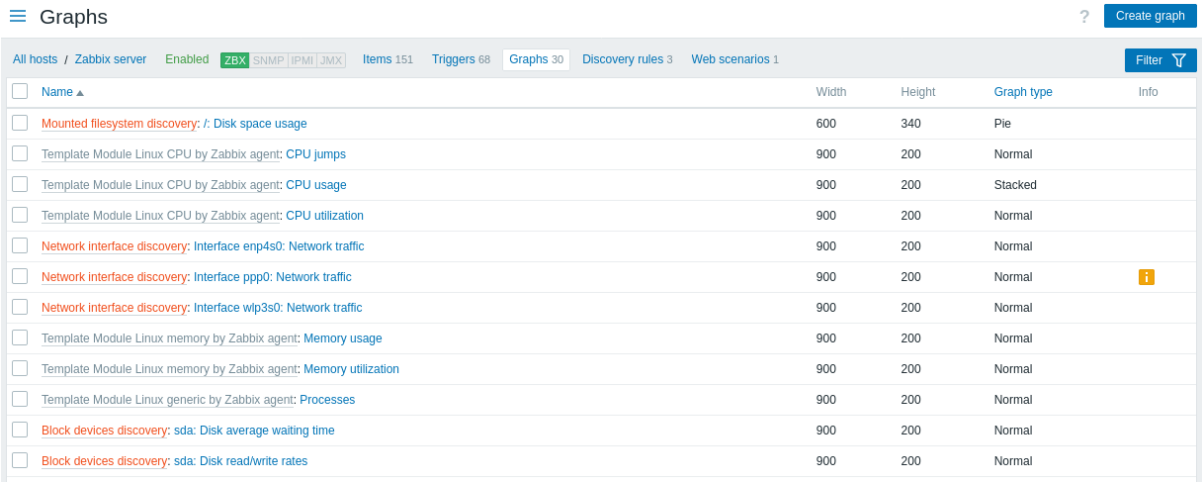
Parameter	Description
Inherited	テンプレートから継承されたトリガー（または継承されていないトリガー）で絞り込みます。
Discovered	ローレベルディスカバリによって検出されたトリガー（または検出されていないトリガー）で絞り込みます。
With dependencies	依存関係があるトリガー（または依存関係がないトリガー）で絞り込みます。

3 グラフ

概要

ホストのカスタムグラフ一覧には、データ収集 → ホストから対象のホストの グラフをクリックしてアクセスできます。

既存のグラフの一覧が表示されます。



表示されるデータ:

Column	Description
ホスト	グラフのホストです。 ホスト名をクリックすると、ホストの設定フォームが開きます。 この列は、フィルターで複数のホストが選択されている場合、またはホストが選択されていない場合にのみ表示されます。
名前	カスタムグラフの名前が、グラフの詳細への青色のリンクとして表示されます。 グラフ名のリンクをクリックすると、グラフの設定フォームが開きます。 ホストグラフがテンプレートに属している場合、テンプレート名がグラフ名の前に灰色のリンクとして表示されます。テンプレートのリンクをクリックすると、テンプレートレベルのグラフ一覧が開きます。 グラフがグラフプロトタイプから作成されている場合、その名前の前にローレベルディスカバリルール名がオレンジ色で表示されます。ディスカバリルール名をクリックすると、グラフプロトタイプの一覧が開きます。
Width	グラフの幅が表示されます。
Height	グラフの高さが表示されます。
Graph type	グラフタイプが表示されます - 標準、積み上げ、円、または 分解円。
Info	グラフが正常に動作している場合、この列にはアイコンは表示されません。エラーがある場合は、文字「i」の付いた四角いアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいグラフを設定するには、右上隅の グラフの作成ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

- コピー - グラフを他のホストまたはテンプレートにコピーします
- 削除 - グラフの削除

これらのオプションを使用するには、それぞれのグラフ先頭のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

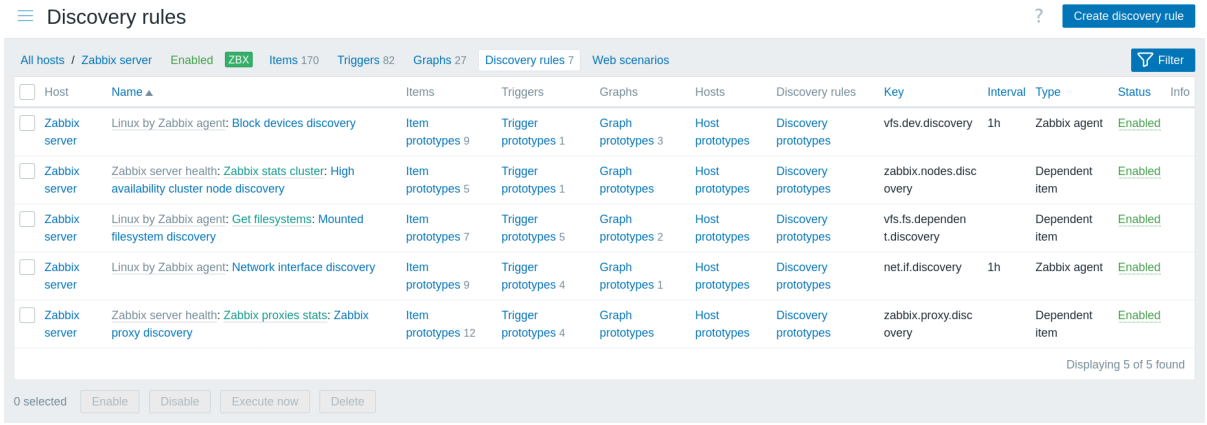
ホストグループおよびホストでグラフをフィルタリングできます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

4 ディスカバリルール

概要

ホストのローレベルディスカバリルールの一覧は、データ収集 → ホストから対象のホストの ディスカバリをクリックして表示できます。

既存のローレベルディスカバリルールの一覧が表示されます。フィルター設定を変更することで、ホストに依存しないすべてのディスカバリルール、または特定のホストグループのすべてのディスカバリルールを表示することもできます。



表示されるデータ:

Column	Description
Host	表示用のホスト名が表示されます。 ホスト名をクリックすると、ホストの設定フォームが開きます。 表示用のホスト名がない場合は、技術的なホスト名が表示されます。
Name	ルール名が青色のリンクとして表示されます。 ルール名をクリックすると、ローレベルディスカバリルールの設定フォームが開きます。 ディスカバリルールがテンプレートに属している場合、ルール名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、テンプレートレベルのルール一覧が開きます。
Items	アイテムのプロトタイプ一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数は灰色で表示されます。
Triggers	トリガーのプロトタイプ一覧へのリンクが表示されます。 既存のトリガープロトタイプ数は灰色で表示されます。
Graphs	グラフのプロトタイプ一覧へのリンクが表示されます。 既存のグラフプロトタイプ数は灰色で表示されます。
Hosts	ホストのプロトタイプ一覧へのリンクが表示されます。 既存のホストプロトタイプ数は灰色で表示されます。
Key	ディスカバリに使用されるアイテムキーが表示されます。
Interval	ディスカバリの実行頻度が表示されます。 なお、一覧の下にある 今すぐ実行ボタンを押すことで、ディスカバリを即時に実行することもできます。
Type	ディスカバリに使用されるアイテムタイプが表示されます（Zabbix エージェント、SNMP エージェントなど）。
Status	ディスカバリルールのステータスが表示されます。表示される値は 有効、無効、または 未サポートです。ステータスをクリックすると変更できます。有効から 無効（およびその逆）、未サポートから無効（およびその逆）に切り替えられます。
Info	問題がなければ、この列にはアイコンは表示されません。エラーがある場合は、文字”i” の付いた四角いアイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいローレベルディスカバリルールを設定するには、右上隅にある ディスカバリルールの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

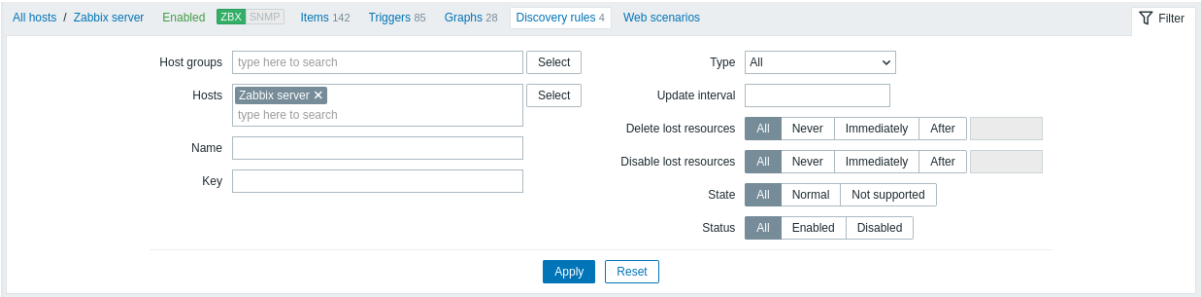
- *Enable* - ローレベルディスカバリルールのステータスを *Enabled* に変更します。
- *Disable* - ローレベルディスカバリルールのステータスを *Disabled* に変更します。
- *Execute now* - ディスカバリルールに基づいて、ただちにディスカバリを実行します。詳細はこちらを参照してください。なお、ディスカバリを即時実行した場合、設定キャッシュは更新されないため、ディスカバリルール設定に対するごく最近の変更は結果に反映されません。
- *Delete* - ローレベルディスカバリルールを削除します。

これらのオプションを使用するには、対象のディスカバリルールの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるディスカバリルールのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

Filter リンクはディスカバリルールの一覧の上にあります。これをクリックするとフィルターが表示され、ホストグループ、ホスト、名前、アイテムキー、アイテムタイプ、その他のパラメータでディスカバリルールを絞り込めます。

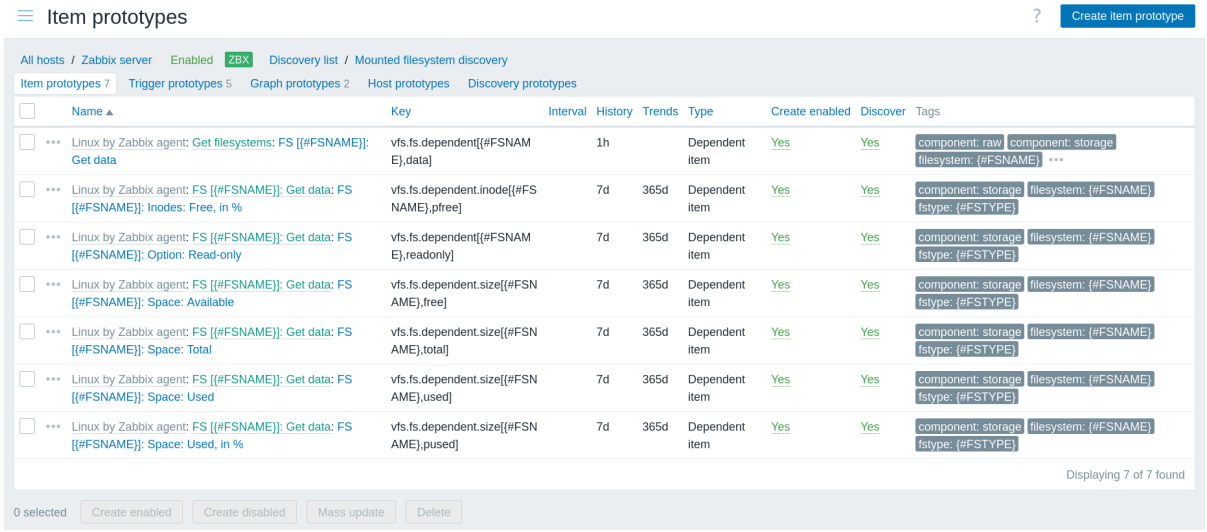


Parameter	Description
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。
Hosts	1 つ以上のホストで絞り込みます。
Name	ディスカバリルール名で絞り込みます。
Key	ディスカバリアイテムキーで絞り込みます。
Type	ディスカバリアイテムタイプで絞り込みます。
Update interval	更新間隔で絞り込みます。 Zabbix トラッパーおよび依存アイテムでは使用できません。
Delete lost resources	*Delete lost resources* 期間で絞り込みます。
Disable lost resources	*Disable lost resources* 期間で絞り込みます。
SNMP OID	SNMP OID で絞り込みます。 タイプとして *SNMP agent* が選択されている場合にのみ使用できます。
State	ディスカバリルールの状態（All/Normal/Not supported）で絞り込みます。
Status	ディスカバリルールのステータス（All/Enabled/Disabled）で絞り込みます。

1 アイテムのプロトタイプ

概要

このセクションでは、ホスト上のローレベルディスカバリルールのアイテムプロトタイプが表示されます。アイテムプロトタイプは、ローレベルディスカバリ中に作成される実際のホストアイテムの基礎となるものです。



表示されるデータ:

Column	Description
Name	アイテムプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、アイテムプロトタイプの設定フォームが開きます。 アイテムプロトタイプがテンプレートに属している場合、ルール名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、テンプレートレベルのアイテムプロトタイプ一覧が開きます。 アイテムプロトタイプのタイプが「Dependent item」の場合、アイテムプロトタイプ名の前にマスターアイテム名が緑色で表示されます。
Key	アイテムプロトタイプのキーが表示されます。
Interval	チェックの実行間隔が表示されます。
History	アイテムデータの履歴を保持する日数が表示されます。
Trends	アイテムのトレンド履歴を保持する日数が表示されます。
Type	アイテムプロトタイプのタイプが表示されます（Zabbix エージェント、SNMP エージェント、simple check など）。
Create enabled	このプロトタイプに基づいてアイテムを次の状態で作成します: **Yes** - 有効 **No** - 無効。クリックすると「Yes」と「No」を切り替えることができます。
Discover	このプロトタイプに基づいてアイテムをディスカバリします: **Yes** - ディスカバリする **No** - ディスカバリしない。クリックすると「Yes」と「No」を切り替えることができます。
Tags	アイテムプロトタイプのタグが表示されます。

新しいアイテムプロトタイプを設定するには、右上隅にある *Create item prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

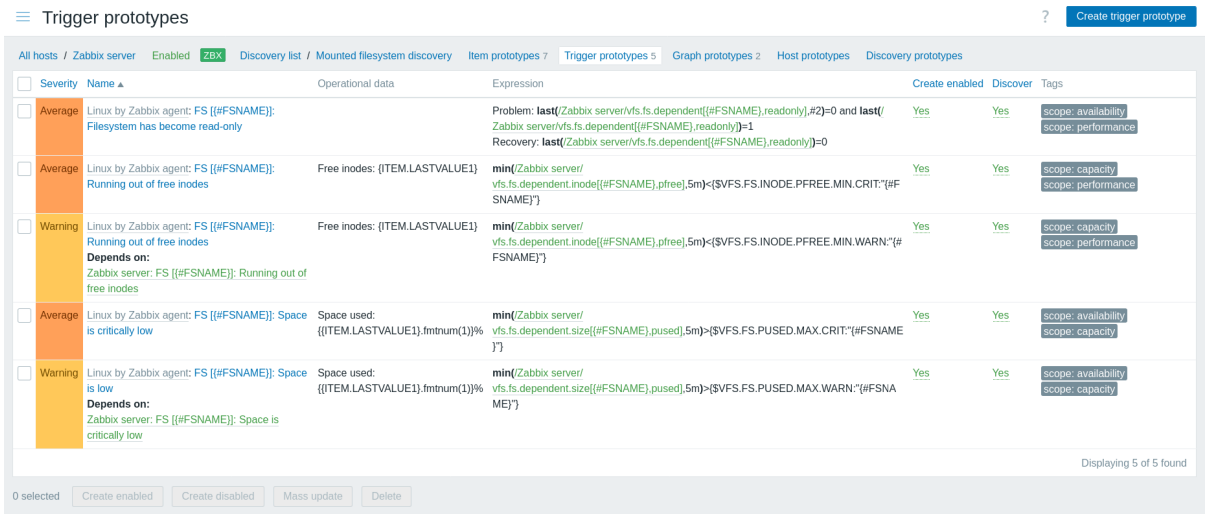
- 有効として作成 - これらのアイテムを有効として作成します
- 無効として作成 - これらのアイテムを無効として作成します
- 一括更新 - これらのアイテムのプロトタイプを一括更新します
- 削除 - これらのアイテムのプロトタイプを削除します

これらのオプションを使用するには、対象のアイテムのプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

2 トリガーのプロトタイプ

概要

このセクションでは、ホスト上のローレベルディスカバリルールのトリガープロトタイプが表示されます。
トリガープロトタイプは、ローレベルディスカバリ中に作成される実際のホストのトリガーの基礎となるものです。



表示されるデータ:

Column	Description
Name	トリガープロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、トリガープロトタイプの設定フォームが開きます。 トリガープロトタイプがリンクされたテンプレートに属している場合、トリガー名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのトリガープロトタイプ一覧が開きます。
Operational data	トリガーの運用データの形式が表示されます。これには任意の文字列およびマクロを含めることができ、*Monitoring → Problems* で動的に展開されます。
Create enabled	このプロトタイプに基づいてトリガーを次の状態で作成します: **Yes** - 有効 **No** - 無効。クリックすることで「Yes」と「No」を切り替えることができます。
Discover	このプロトタイプに基づくトリガーをディスカバリするかどうかを指定します: **Yes** - ディスカバリする **No** - ディスカバリしない。クリックすることで「Yes」と「No」を切り替えることができます。
Tags	トリガープロトタイプのタグが表示されます。

新しいトリガープロトタイプを設定するには、右上隅にある *Create trigger prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

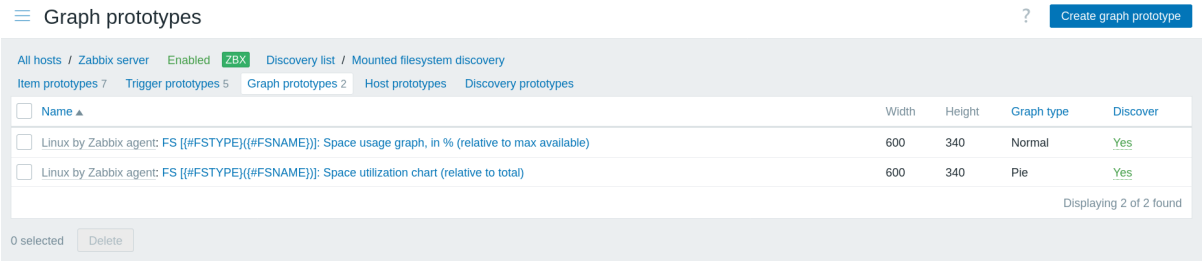
- *Create enabled* - これらのトリガーを有効として作成します
- *Create disabled* - これらのトリガーを無効として作成します
- *Mass update* - これらのトリガープロトタイプを一括更新します
- *Delete* - これらのトリガープロトタイプを削除します

これらのオプションを使用するには、該当するトリガープロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

3 グラフのプロトタイプ

概要

このセクションでは、ホスト上のローレベルディスカバリルールのグラフプロトタイプが表示されます。グラフプロトタイプは、ローレベルディスカバリ中に作成される実際のホストグラフの基礎となります。



表示されるデータ:

Column	Description
Name	グラフプロトタイプの名前が青いリンクとして表示されます。 名前をクリックすると、グラフプロトタイプの設定フォームが開きます。 グラフプロトタイプがリンクされたテンプレートに属している場合、グラフ名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのグラフプロトタイプ一覧が開きます。
Width	グラフプロトタイプの幅が表示されます。
Height	グラフプロトタイプの高さが表示されます。
Type	グラフプロトタイプのタイプが表示されます。- *Normal*、*Stacked*、*Pie*、または *Exploded* です。
Discover	このプロトタイプに基づいてグラフをディスカバリします: **Yes** - ディスカバリする **No** - ディスカバリしない。'Yes' と 'No' はクリックして切り替えることができます。

新しいグラフプロトタイプを設定するには、右上隅にある *Create graph prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

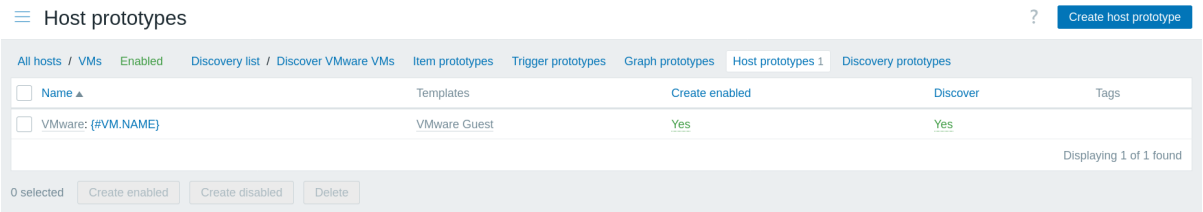
- *Delete* - これらのグラフプロトタイプを削除します

これらのオプションを使用するには、対象のグラフプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

4 ホストのプロトタイプ

概要

このセクションでは、ホスト上のローレベルディスカバリルールのホストプロトタイプが表示されます。
ホストプロトタイプは、ローレベルディスカバリ中に作成される実際のホストの基礎となるものです。



表示されるデータ:

Column	Description
Name	ホストプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、ホストプロトタイプの設定フォームが開きます。 ホストプロトタイプがリンクされたテンプレートに属している場合、ホスト名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのホストプロトタイプ一覧が開きます。
Templates	ホストプロトタイプのテンプレートが表示されます。
Create enabled	このプロトタイプに基づいてホストを次の状態で作成します: **Yes** - 有効 **No** - 無効。クリックすることで「Yes」と「No」を切り替えることができます。

Column	Description
Discover	このプロトタイプに基づいてホストをディスカバリします: **Yes** - ディスカバリする **No** - ディスカバリしない。クリックすることで「Yes」と「No」を切り替えることができます。
Tags	ホストプロトタイプのタグが表示されます。

新しいホストプロトタイプを設定するには、右上隅にある *Create host prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

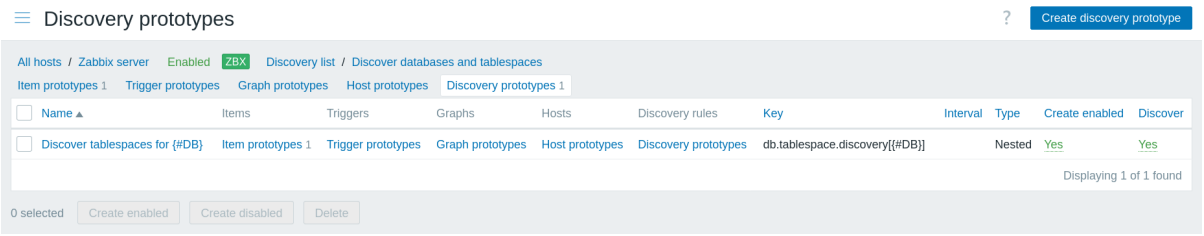
- 有効で作成 - これらのホストを有効として作成します
- 無効で作成 - これらのホストを無効として作成します
- 削除 - これらのホストプロトタイプを削除します

これらのオプションを使用するには、対象のホストプロトタイプの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

5 ディスカバリのプロトタイプ

概要

このセクションでは、ホスト上のローレベルディスカバリルールのディスカバリプロトタイプが表示されます。ディスカバリプロトタイプは、親ディスカバリルールにネストされたディスカバリルールです。



表示されるデータ:

Column	Description
Name	ディスカバリプロトタイプの名前が青色のリンクとして表示されます。 名前をクリックすると、ディスカバリプロトタイプの設定フォームが開きます。 ディスカバリプロトタイプがリンクされたテンプレートに属している場合、ホスト名の前にテンプレート名が灰色のリンクとして表示されます。テンプレートリンクをクリックすると、リンクされたテンプレートレベルのディスカバリプロトタイプ一覧が開きます。
Items	アイテムプロトタイプ一覧へのリンクが表示されます。 既存のアイテムプロトタイプ数は灰色で表示されます。
Triggers	トリガープロトタイプ一覧へのリンクが表示されます。 既存のトリガープロトタイプ数は灰色で表示されます。
Graphs	グラフプロトタイプ一覧へのリンクが表示されます。 既存のグラフプロトタイプ数は灰色で表示されます。
Hosts	ホストプロトタイプ一覧へのリンクが表示されます。 既存のホストプロトタイプ数は灰色で表示されます。
Discovery rules	ディスカバリプロトタイプ一覧へのリンクが表示されます。 既存のディスカバリプロトタイプ数は灰色で表示されます。
Key	ディスカバリに使用されるアイテムキーが表示されます。
Interval	ディスカバリの実行頻度が表示されます。 なお、一覧の下にある *Execute now* ボタンを押すことで、ディスカバリを即時に実行することもできます。
Type	ディスカバリに使用されるアイテムタイプが表示されます（Zabbix エージェント、SNMP エージェントなど）。
Create enabled	ディスカバリプロトタイプの作成ステータスが表示されます - 作成有効（*Yes*）または作成無効（*No*）。ステータスをクリックすると、*Yes* から *No*（およびその逆）に変更できます。
Discover	ディスカバリプロトタイプの検出ステータスが表示されます - 検出する（*Yes*）または検出しない（*No*）。ステータスをクリックすると、*Yes* から *No*（およびその逆）に変更できます。

新しいディスカバリプロトタイプを設定するには、右上隅の *Create discovery prototype* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

- *Create enabled* - これらのディスカバリプロトタイプを *Enabled* として作成します
- *Create disabled* - これらのディスカバリプロトタイプを *Disabled* として作成します
- *Delete* - これらのディスカバリプロトタイプを削除します

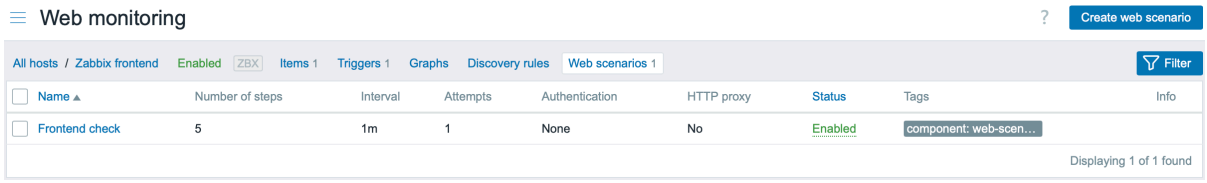
これらのオプションを使用するには、対象のディスカバリプロトタイプの前にあるチェックボックスを選択し、必要なボタンをクリックします。

5 Web シナリオ

概要

ホストのWeb シナリオ一覧は、データ収集 → ホストから対象のホストの *Web* をクリックしてアクセスできます。

既存の Web シナリオの一覧が表示されます。



表示されるデータ:

列	説明
名前	Web シナリオの名前です。Web シナリオ名をクリックすると、Web シナリオの設定フォームが開きます。 ホストの Web シナリオがテンプレートに属している場合、Web シナリオ名の前にテンプレート名が灰色のリンクで表示されます。テンプレートリンクをクリックすると、テンプレートレベルの Web シナリオ一覧が開きます。
ステップ数	シナリオに含まれるステップ数です。
更新間隔	シナリオの実行頻度です。
試行回数	Web シナリオのステップ実行時に行われる試行回数です。
認証	認証方式が表示されます - Basic、NTLM、Kerberos、Digest、または None。
HTTP プロキシ	HTTP プロキシ、または使用していない場合は「No」を表示します。
ステータス	Web シナリオのステータスが表示されます - 有効または 無効。 ステータスをクリックすると変更できます。
タグ	Web シナリオのタグが表示されます。 最大 3 つのタグ（name:value のペア）を表示できます。タグがそれより多い場合は”...” リンクが表示され、マウスオーバーですべてのタグを確認できます。
情報	すべてが正常に動作している場合、この列にはアイコンは表示されません。エラーがある場合は、文字”i” の付いた四角いアイコンが表示されます。アイコンにマウスオーバーすると、エラーの説明を含むツールチップが表示されます。

新しい Web シナリオを設定するには、右上隅の *Web* シナリオの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

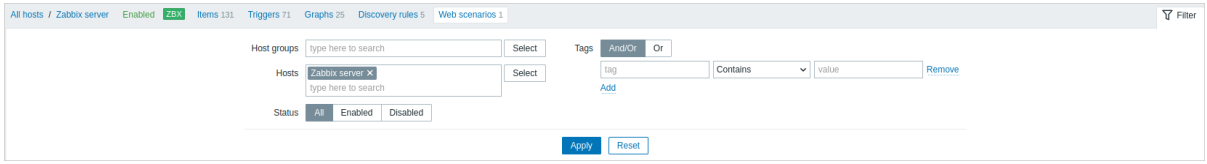
- 有効化 - シナリオのステータスを有効に変更します
- 無効化 - シナリオのステータスを無効に変更します
- 履歴とトレンドをクリア - シナリオの履歴データおよびトレンドデータをクリアします
- 削除 - Web シナリオを削除します

これらのオプションを使用するには、対象の Web シナリオの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるシナリオのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

Filter リンクは Web シナリオの一覧の上にあります。これをクリックするとフィルターが開き、必要な絞り込み条件を指定できます。



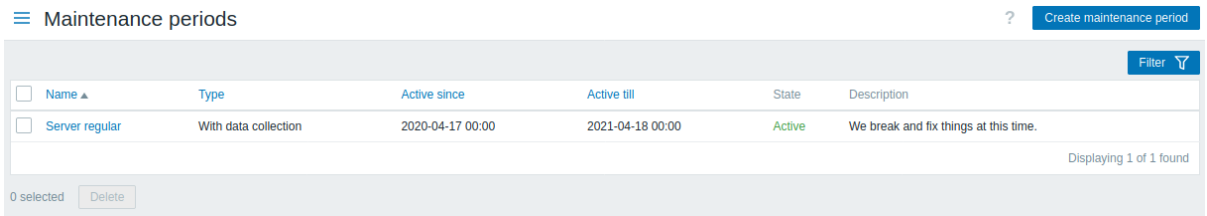
Parameter	Description
Host groups	1 つ以上のホストグループで絞り込みます。 親ホストグループを指定すると、ネストされたすべてのホストグループも暗黙的に選択されます。
Hosts	1 つ以上のホストで絞り込みます。 上でホストグループがすでに選択されている場合、ホストの選択はそれらのグループに制限されます。
Status	Web シナリオのステータスで絞り込みます。
Tags	Web シナリオのタグ名と値で絞り込みます。 特定のタグおよびタグ値を含めることも除外することもできます。複数の条件を設定できます。タグ名の一致では常に大文字と小文字が区別されます。 各条件では、次の演算子を使用できます。 **Exists** - 指定したタグ名を含めます。 **Equals** - 指定したタグ名と値を含めます（大文字と小文字を区別）。 **Contains** - タグ値に入力した文字列が含まれる指定タグ名を含めます（部分一致、大文字と小文字を区別しない）。 **Does not exist** - 指定したタグ名を除外します。 **Does not equal** - 指定したタグ名と値を除外します（大文字と小文字を区別）。 **Does not contain** - タグ値に入力した文字列が含まれる指定タグ名を除外します（部分一致、大文字と小文字を区別しない）。 条件の計算タイプは 2 種類あります。 **And/Or** - すべての条件を満たす必要があります。同じタグ名を持つ条件は Or 条件でグループ化されます。 **Or** - いずれか 1 つの条件を満たせば十分です。

5 メンテナンス

概要

データ収集 → メンテナンスセクションでは、ユーザーはホストのメンテナンス期間を設定および管理できます。

既存のメンテナンス期間の一覧とその詳細が表示されます。



表示されるデータ:

Column	Description
Name	メンテナンス期間の名前です。メンテナンス期間名をクリックすると、メンテナンス期間の設定フォームが開きます。
Type	メンテナンスのタイプが表示されます: データ収集ありまたは データ収集なし
Active since	実行されるメンテナンス期間が有効になる日時です。 注: この時刻によってメンテナンス期間が有効化されるわけではありません。メンテナンス期間は別途設定する必要があります。
Active till	実行されるメンテナンス期間が有効でなくなる日時です。

Column	Description
State	メンテナンス期間の状態です: **Approaching** - まもなく有効になります **Active** - 有効です **Expired** - すでに有効ではありません
Description	メンテナンス期間の説明が表示されます。

新しいメンテナンス期間を設定するには、右上隅にある メンテナンス期間の作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、次の一括編集オプションを利用できます。

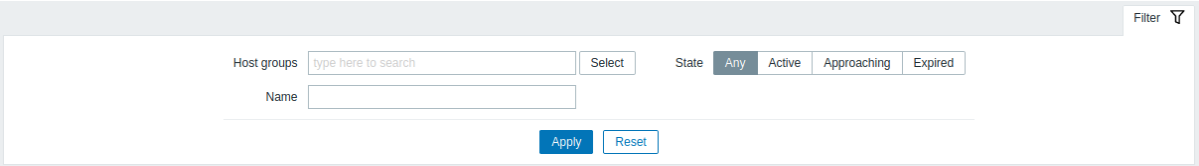
- 削除 - メンテナンス期間を削除します

このオプションを使用するには、対象のメンテナンス期間の前にあるチェックボックスを選択し、削除をクリックします。

フィルターの使用

フィルターを使用して、関心のあるメンテナンス期間のみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

メンテナンス期間の一覧の上に、フィルターリンクがあります。これをクリックするとフィルターが表示され、ホストグループ、名前、状態でメンテナンス期間を絞り込めます。



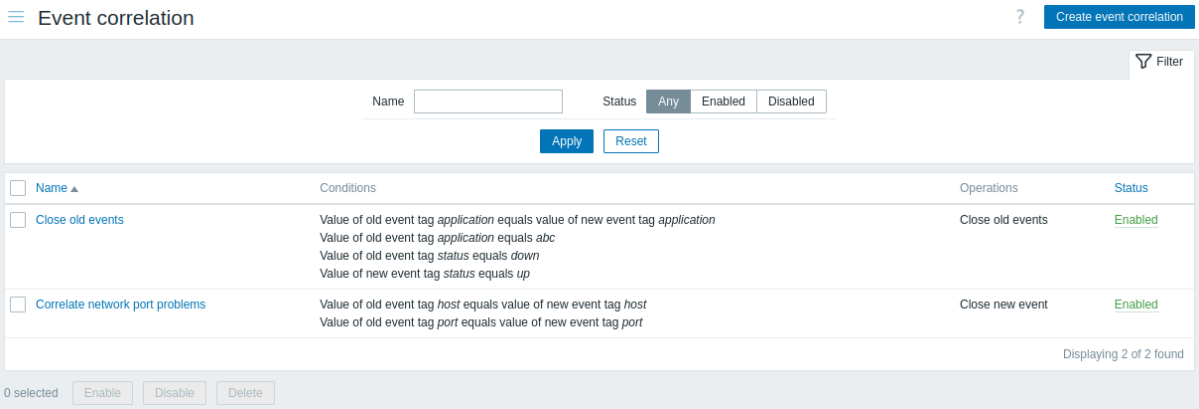
メンテナンス中のキューの計算

Attention:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算を参照してください。

6 イベント相関関係

概要

データ収集 → イベント相関セクションでは、ユーザーは Zabbix イベントのグローバル相関ルールを設定および管理できます。



表示されるデータ:

Column	Description
Name	相関ルールの名前です。相関ルール名をクリックすると、ルールの設定フォームが開きます。
Conditions	相関ルールの条件が表示されます。
Operations	相関ルールの操作が表示されます。
Status	相関ルールのステータスが表示されます - 有効または 無効。 ステータスをクリックすると変更できます。

新しい相関ルールを設定するには、右上隅にある イベント相関の作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

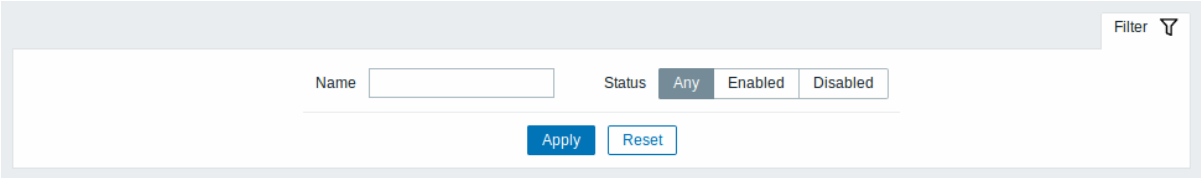
- *Enable* - 相関ルールのステータスを *Enabled* に変更します
- *Disable* - 相関ルールのステータスを *Disabled* に変更します
- *Delete* - 相関ルールを削除します

これらのオプションを使用するには、対象の相関ルールの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のある相関ルールのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

相関ルールの一覧の上にある *Filter* リンクを利用できます。これをクリックすると、名前とステータスで相関ルールを絞り込めるフィルターが表示されます。

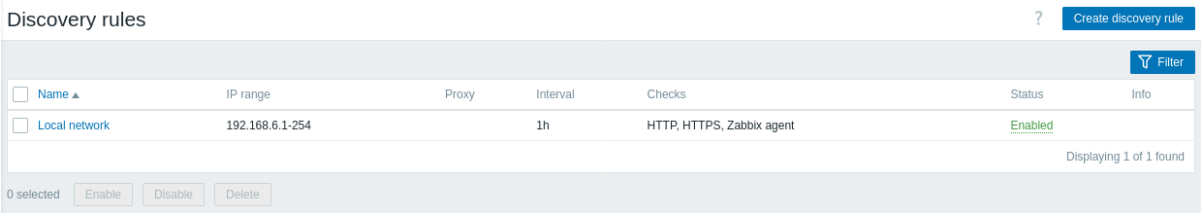


7 ディスカバリ

概要

データ収集 → ディスカバリセクションでは、ユーザーはディスカバリルールを設定および管理できます。

既存のディスカバリルールの一覧と、その詳細が表示されます。



表示されるデータ:

列	説明
名前	ディスカバリルールの名前です。ディスカバリルール名をクリックすると、ディスカバリルールの設定フォームが開きます。
IP 範囲	ネットワークスキャンに使用する IP アドレスの範囲が表示されます。
プロキシ	ディスカバリをプロキシが実行する場合、プロキシ名が表示されます。
間隔	ディスカバリを実行する頻度が表示されます。
チェック	ディスカバリに使用されるチェックの種類が表示されます。
ステータス	ディスカバリルールのステータスが表示されます。-有効または 無効。 ステータスをクリックすると変更できます。
情報	すべてが正常に動作している場合、この列には何も表示されません。エラーがある場合は、文字「i」の付いた赤い情報アイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいディスカバリルールを設定するには、右上隅にある ディスカバリルールの作成ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

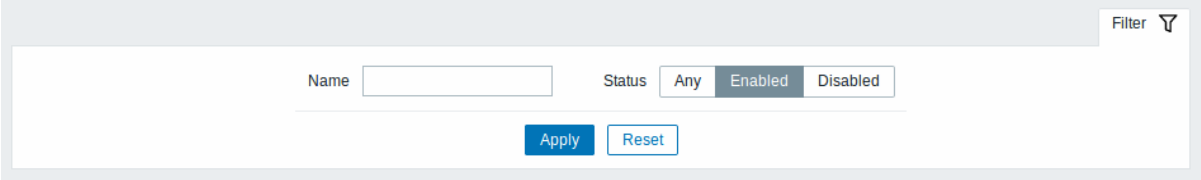
- *Enable* - ディスカバリルールのステータスを *Enabled* に変更します
- *Disable* - ディスカバリルールのステータスを *Disabled* に変更します
- *Delete* - ディスカバリルールを削除します

これらのオプションを使用するには、対象のディスカバリルールの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるディスカバリルールのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

Filter リンクはディスカバリルールの一覧の上にあります。これをクリックすると、名前とステータスでディスカバリルールを絞り込めるフィルターが表示されます。



7 アラート

概要

このメニューには、Zabbix でのアラートの設定に関連するセクションがあります。

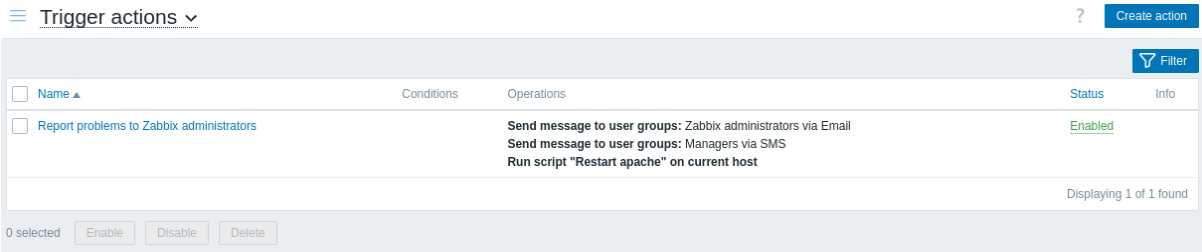
1 アクション

概要

アラート → アクションセクションでは、ユーザーはアクションを設定および管理できます。

表示されるアクションは、選択したイベントソース（トリガー、サービス、ディスカバリ、自動登録、内部アクション）に割り当てられているアクションです。

別のイベントソースに切り替えるには、アラートメニューセクションの アクションをクリックします。また、左上隅にあるタイトルのドロップダウンを使用してソースを切り替えることもできます。



表示データ:

Column	Description
Name	アクション名です。アクション名をクリックすると、アクションの設定フォームが開きます。
Conditions	アクションの条件が表示されます。
Operations	アクションの実行内容が表示されます。 実行内容の一覧には、通知に使用されるメディアタイプ（メール、SMS、またはスクリプト）に加え、通知の受信者の名前と姓（ユーザー名の後の括弧内）も表示されます。 アクションの実行内容は、選択した実行内容のタイプに応じて、通知またはリモートコマンドのいずれにもなります。
Status	アクションのステータスが表示されます。*Enabled* または *Disabled* です。 ステータスをクリックすると変更できます。 エスカレーションの進行中にアクションが無効化された場合に何が起こるかの詳細については、エスカレーションセクションを参照してください。
Info	すべてが正常に動作している場合、この列にはアイコンは表示されません。アップグレード後やローレベルディスカバリ後にアクションの実行内容または条件が欠落しているなどのエラーがある場合は、警告アイコンが表示されます。アイコンにマウスカーソルを合わせると、エラーの説明を含むツールチップが表示されます。

新しいアクションを設定するには、右上隅の *Create action* ボタンをクリックします。

Super admin 権限のないユーザーには、権限設定に従ってアクションが表示されます。つまり、特定の権限制限により、Super admin 権限のないユーザーは完全なアクション一覧を表示できない場合があります。以下の条件を満たす場合、Super admin 権限のないユーザーにもアクションが表示されます。

- ユーザーが、アクション条件内のホストグループ、ホスト、テンプレート、およびトリガーに対して読み書きアクセス権を持っている
- ユーザーが、アクションの実行内容、復旧時の実行内容、および更新時の実行内容内のホストグループ、ホスト、およびテンプレートに対して読み書きアクセス権を持っている
- ユーザーが、アクションの実行内容、復旧時の実行内容、および更新時の実行内容内のユーザーグループおよびユーザーに対して読み取りアクセス権を持っている

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

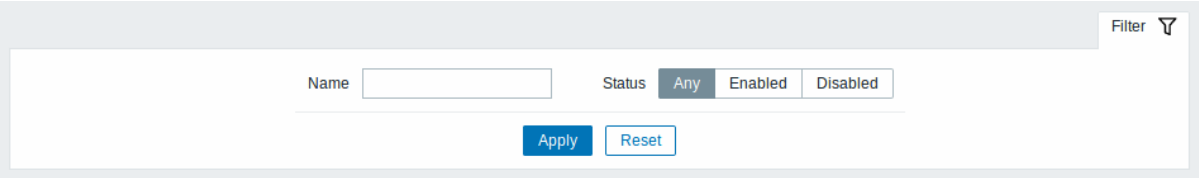
- 有効 - アクションステータスを有効に変更します
- 無効 - アクションステータスを無効に変更します
- 削除 - アクションを削除します

これらのオプションを使用するには、それぞれのアクションの前あるにチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して任意のアクションのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクはアクションリストの上にあります。クリックすると名前とステータスでアクションをフィルタリングできるフィルターが使用可能になります。



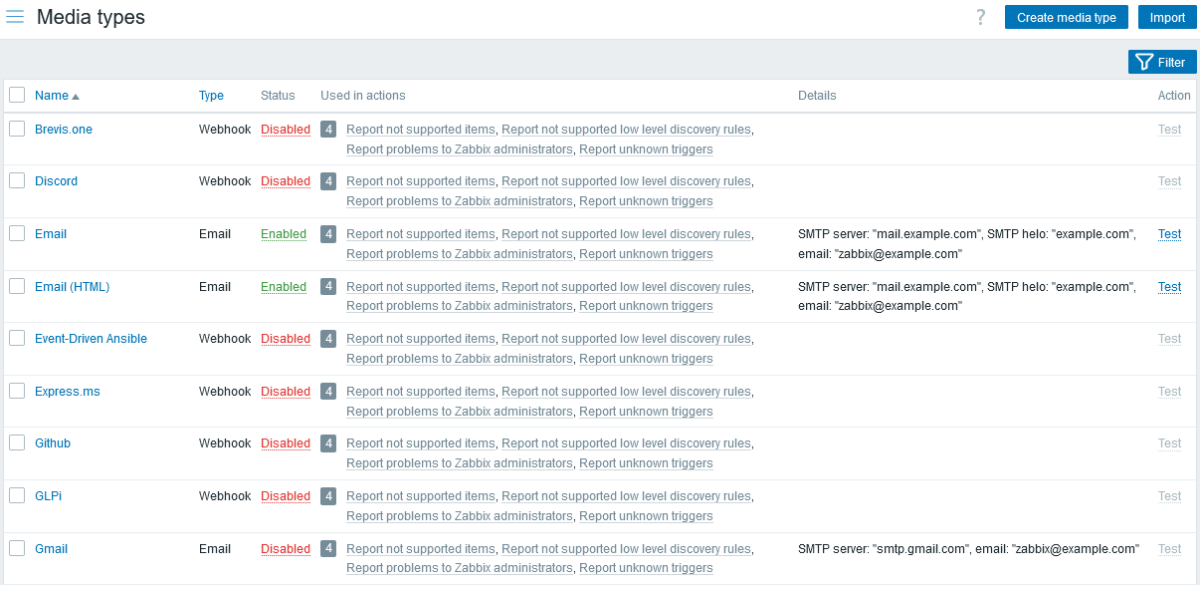
2 メディアタイプ

概要

Alerts → Media types セクションでは、ユーザーはメディアタイプ情報を設定および管理できます。

メディアタイプ情報には、通知の配信チャネルとしてメディアを使用するための一般的な手順が含まれます。通知の送信先となる個別のメールアドレスなどの具体的な詳細は、各ユーザーごとに保持されます。

既存のメディアタイプの一覧とその詳細が表示されます。



表示されるデータ:

Column	Description
Name	メディアタイプの名前です。名前をクリックすると、メディアタイプの設定フォームが開きます。
Type	メディアの種類（メール、SMS など）が表示されます。
Status	メディアタイプのステータスが表示されます - *Enabled* または *Disabled*。 ステータスをクリックすると変更できます。
Used in actions	メディアタイプが使用されているアクションが表示され、その前にそれらのアクションの総数が表示されます。 アクション名をクリックすると、アクション設定フォームが開きます。ユーザーにそのアクションへの権限がない場合、名前はクリックできません。
Details	メディアタイプの詳細情報が表示されます。
Actions	次の操作を利用できます: **Test** - クリックするとテスト用フォームが開き、メディアタイプのパラメータ（例: テスト用の件名と本文を含む受信者アドレス）を入力して、設定済みのメディアタイプが正しく動作することを確認するためのテストメッセージを送信できます。あわせて参照: Email、Webhook、またはScript のメディアタイプテスト。

新しいメディアタイプを設定するには、右上隅にある *Create media type* ボタンをクリックします。

メディアタイプをインポートするには、右上隅にある *Import* ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

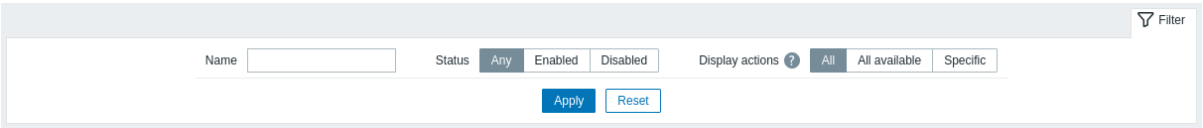
- 有効 - メディアタイプのステータスを *Enabled* に変更します
- 無効 - メディアタイプのステータスを *Disabled* に変更します
- エクスポート - メディアタイプを YAML, XML または JSON ファイルとしてエクスポートします
- 削除 - メディアタイプを削除します

これらのオプションを使用するには、それぞれのメディアタイプの前のチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用すると、関心のあるメディアタイプのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

メディアタイプの一覧の上に、*Filter* リンクがあります。これをクリックするとフィルターが表示され、名前とステータスでメディアタイプを絞り込めます。さらに、フィルターを使用して、メディアタイプの使用範囲（アクションの操作の詳細で定義される *Send to media type* パラメータ）に基づき、*Used in actions* 列に表示するアクションを絞り込むこともできます。



3 スクリプト

概要

アラート > スクリプトセクションでは、ユーザー定義のグローバルスクリプトを設定および管理できます。

グローバルスクリプトは、設定されたスコープおよびユーザー権限に応じて、次の場所から実行できます。

- 各種 Web インターフェース上のホストメニュー から（ダッシュボード、障害、最新データ、マップなど）
- イベントメニュー から
- アクションの実行内容として実行可能

スクリプトは、Zabbix エージェント、Zabbix サーバー（プロキシ）、または Zabbix サーバーのみで実行されます。
あわせてコマンド実行 も参照してください。

Zabbix エージェントおよび Zabbix プロキシの両方で、リモートスクリプトはデフォルトで無効になっています。
有効にするには、次のようにします。

- Zabbix エージェントで実行されるリモートコマンドの場合:
  - 許可する各コマンドについて、agent 設定に AllowKey=system.run[<command>,*] パラメータを追加します。* は wait モードおよび nowait モードを表します。
- Zabbix プロキシで実行されるリモートコマンドの場合:

- 警告: リモートコマンドが **Zabbix** プロキシによって監視されている **Zabbix** エージェント上で実行される場合、**Zabbix** プロキシでリモートコマンドを有効にする必要はありません。ただし、Zabbix プロキシ上でリモートコマンドを実行する必要がある場合は、プロキシ設定で *EnableRemoteCommands* パラメータを'1' に設定してください。

Zabbix サーバーでのグローバルスクリプト実行は、サーバー設定 で `EnableGlobalScripts=0` を設定することで無効にできます。
新規インストールでは、Zabbix 7.0 以降、Zabbix サーバーでのグローバルスクリプト実行はデフォルトで無効になっています。

既存のスクリプトの一覧とその詳細が表示されます。



表示されるデータ:

Column	Description
Name	スクリプト名。スクリプト名をクリックすると、スクリプトの設定フォームが開きます。
Scope	スクリプトのスコープ - アクションの実行内容、手動ホストアクション、または手動イベントアクション。この設定により、スクリプトを利用できる場所が決まります。
Used in actions	スクリプトが使用されているすべてのアクションが、該当アクションの総数とともに表示されます。アクション名をクリックすると、アクション設定フォームが開きます。ユーザーにそのアクションへの権限がない場合、名前はクリックできません。
Type	スクリプトタイプが表示されます - *URL*、*Webhook*、*Script*、*SSH*、*Telnet*、または *IPMI* コマンド。
Execute on	スクリプトが Zabbix エージェント、Zabbix プロキシ、サーバー、または Zabbix サーバーのみのどこで実行されるかが表示されます。
Commands	スクリプト内で実行されるすべてのコマンドが表示されます。 webhook の場合、ここには何も表示されません。
User group	スクリプトを利用できるユーザーグループが表示されます（すべてのユーザーグループが対象の場合は *All*）。
Host group	スクリプトを利用できるホストグループが表示されます（すべてのホストグループが対象の場合は *All*）。
Host access	ホストグループに対する権限レベルが表示されます - *Read* または *Write*。必要な権限レベルを持つユーザーのみがスクリプトを実行できます。

新しいスクリプトを設定するには、右上隅の *Create script* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、次の一括編集オプションを利用できます。

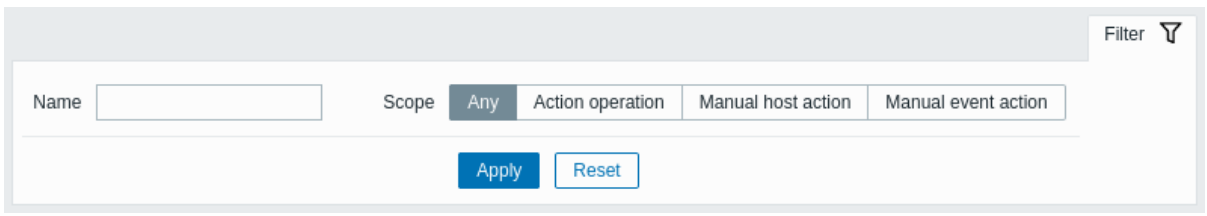
- 削除 - スクリプトを削除します

このオプションを使用するには、対象のスクリプトの前にあるチェックボックスを選択し、削除をクリックします。

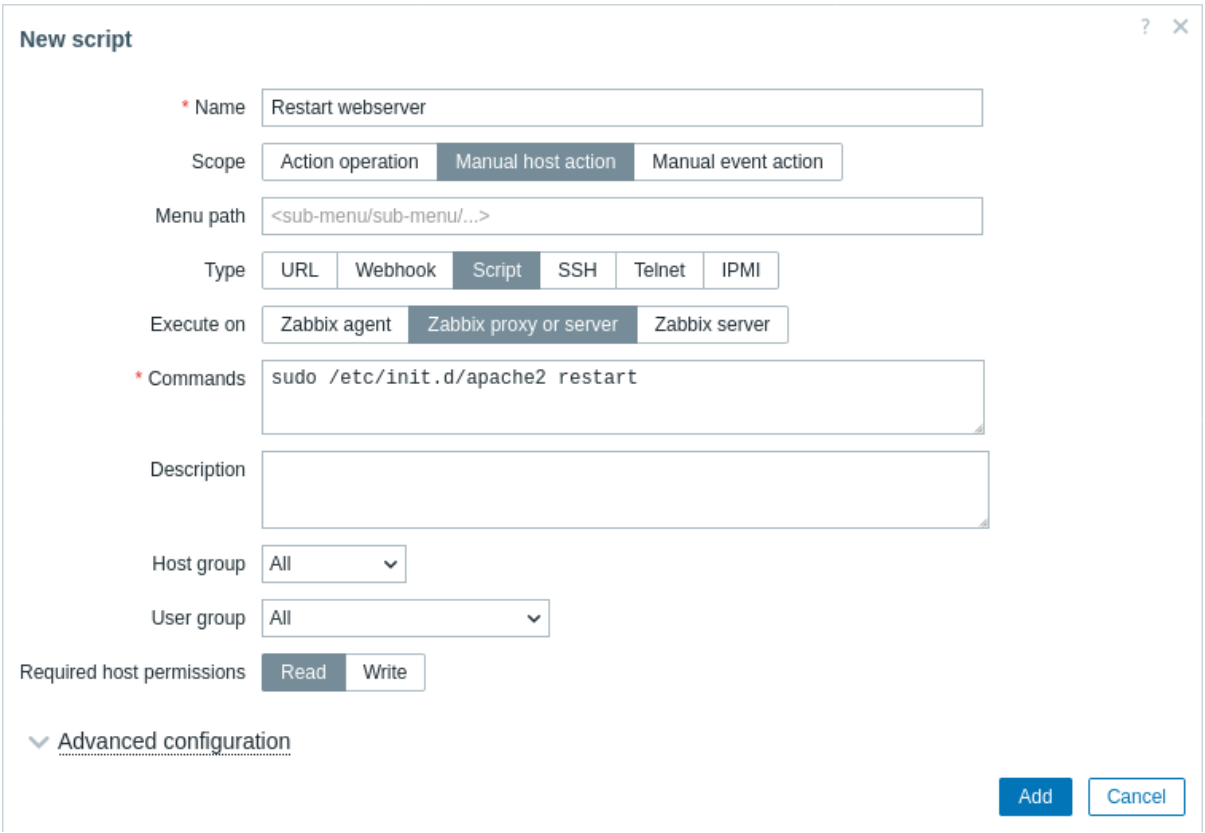
フィルターの使用

フィルターを使用すると、関心のあるスクリプトのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

フィルターリンクはスクリプト一覧の上にあります。
これをクリックすると、名前とスコープでスクリプトを絞り込めるフィルターが表示されます。



グローバルスクリプトの設定



Note:
セキュリティを強化するため、グローバルスクリプトではプレーンなマクロの代わりにマクロ関数を使用することを推奨します。マクロは自動的にエスケープされないためです。

スクリプト属性:

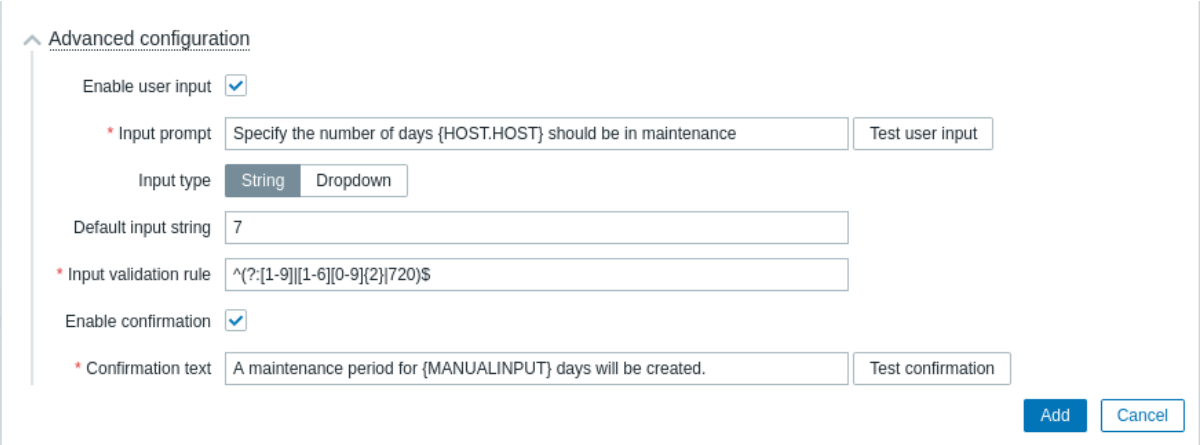
Parameter	Description
Name	スクリプトの一意の名前。 例: `Clear /tmp filesystem`
Scope	スクリプトのスコープ - アクションの実行、手動ホストアクション、または手動イベントアクション。この設定により、スクリプトをどこで使用できるかが決まります。つまり、アクション実行のリモートコマンド、ホストメニュー、またはイベントメニューです。 スコープを「アクションの実行」に設定すると、*Alerts* > *Actions* にアクセスできるすべてのユーザーがこのスクリプトを利用できます。 スクリプトが実際にアクションで使用されている場合、そのスコープを「アクションの実行」以外に変更することはできません。 マクロのサポート スコープは、利用可能なマクロの範囲に影響します。たとえば、ユーザー関連のマクロ ({USER.*}) は、スクリプトを起動したユーザーに関する情報を渡せるように、スクリプトでサポートされています。ただし、アクションの実行は自動的に実行されるため、スクリプトのスコープがアクションの実行である場合はサポートされません。 {MANUALINPUT} マクロを使用すると、スクリプト実行時に手動入力を指定できます。これは手動ホストアクションおよび手動イベントアクションのスクリプトでサポートされます。 他にどのマクロがサポートされているかを確認するには、サポートされるマクロテーブルで「Trigger-based notifications and commands/Trigger-based commands」、「Manual host action scripts」、「Manual event action scripts」を検索してください。マクロがスペースを含む値（たとえばホスト名）に展開される可能性がある場合は、必要に応じて引用符で囲むことを忘れないでください。
Menu path	スクリプトの表示先となるメニューパス。たとえば `Default` または `Default/` を指定すると、対応するディレクトリにスクリプトが表示されます。メニューはネスト可能で、たとえば `Main menu/Sub menu1/Sub menu2` のように指定できます。監視セクションのホスト/イベントメニューからスクリプトにアクセスする場合、指定したディレクトリに従って整理されます。 このフィールドは、*Scope* で「手動ホストアクション」または「手動イベントアクション」が選択されている場合にのみ表示されます。

Parameter	Description
Type	対応するボタンをクリックしてスクリプトタイプを選択します: **URL**、**Webhook**、**Script**、**SSH**、**Telnet**、または **IPMI** コマンド。 **URL** タイプは、*Scope* で「手動ホストアクション」または「手動イベントアクション」が選択されている場合にのみ使用できます。
Script type: **URL**	
URL	ホストメニューまたはイベントメニューからすばやくアクセスするための URL を指定します。 マクロおよびカスタムのユーザーマクロをサポートします。マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。 このフィールドで {MANUALINPUT} マクロを使用すると、たとえば次のように、スクリプト実行時に手動入力を指定できるようになります: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` マクロ値は URL エンコードしてはいけません。
Open in new window	URL を新しいブラウザータブで開くか、同じブラウザータブで開くかを決定します。
Script type: **Webhook**	
Parameters	webhook 変数を属性と値のペアとして指定します。 あわせて参照: Webhook メディア設定。 パラメータ値では、マクロおよびカスタムのユーザーマクロをサポートします。マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。
Script	パラメータフィールド内、またはその横にある鉛筆アイコンをクリックすると開くモーダルエディターで、JavaScript コードを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。 あわせて参照: Webhook メディア設定、Additional Javascript objects。
Timeout	JavaScript の実行タイムアウト（1～60 秒、デフォルトは 30 秒）。 30s、1m などの時間サフィックスをサポートします。
Script type: **Script**	
Execute on	対応するボタンをクリックして、次のいずれかでシェルスクリプトを実行します: **Zabbix agent** - スクリプトはホスト上で Zabbix エージェントによって実行されます (system.run アイテムが許可されている場合) **Zabbix proxy or server** - スクリプトは Zabbix プロキシまたはサーバーによって実行されます。どちらで実行されるかは、そのホストがプロキシとサーバーのどちらによって監視されているかによって決まります。 EnableRemoteCommands により有効化されている場合は、プロキシ上で実行されます。 グローバルスクリプトがサーバーのEnableGlobalScripts パラメータで有効になっている場合は、サーバー上で実行されます。 **Zabbix server** - スクリプトは Zabbix サーバーのみで実行されます。 このオプションは、サーバーのEnableGlobalScripts パラメータによってグローバルスクリプトが無効化されている場合は使用できません。
Commands	スクリプト内で実行するコマンドのフルパスを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。カスタムのユーザーマクロをサポートします。
Script type: **SSH**	
Authentication method	認証方式を選択します - パスワードまたは公開鍵。
Username	ユーザー名を入力します。
Password	パスワードを入力します。 このフィールドは、認証方式として「Password」が選択されている場合に使用できます。
Public key file	公開鍵ファイルへのパスを入力します。 このフィールドは、認証方式として「Public key」が選択されている場合に使用できます。
Private key file	秘密鍵ファイルへのパスを入力します。 このフィールドは、認証方式として「Public key」が選択されている場合に使用できます。
Passphrase	パスフレーズを入力します。 このフィールドは、認証方式として「Public key」が選択されている場合に使用できます。
Port	Zabbix が接続する対象ホスト上のリモート SSH サービスのポートを入力します。
Commands	コマンドを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。カスタムのユーザーマクロをサポートします。
Script type: **Telnet**	
Username	ユーザー名を入力します。
Password	パスワードを入力します。
Port	Zabbix が接続する対象ホスト上のリモート Telnet サービスのポートを入力します。
Commands	コマンドを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。カスタムのユーザーマクロをサポートします。

Parameter	Description
Script type: **IPMI**	
Command	IPMI コマンドを入力します。 マクロのサポートはスクリプトのスコープに依存します（上記の *Scope* を参照）。カスタムのユーザーマクロをサポートします。
Description	スクリプトの説明を入力します。
Host group	スクリプトを利用可能にするホストグループを選択します（すべてのホストグループに対して利用可能にする場合は *All*）。
User group	スクリプトを利用可能にするユーザーグループを選択します（すべてのユーザーグループに対して利用可能にする場合は *All*）。 このフィールドは、*Scope* で「手動ホストアクション」または「手動イベントアクション」が選択されている場合にのみ表示されます。
Required host permissions	ホストグループに対する権限レベルを選択します - *Read* または *Write*。必要な権限レベルを持つユーザーのみがスクリプトの実行にアクセスできます。 このフィールドは、*Scope* で「手動ホストアクション」または「手動イベントアクション」が選択されている場合にのみ表示されます。
Advanced configuration	*Advanced configuration* ラベルをクリックすると、高度な設定オプションが表示されます。 このフィールドは、*Scope* で「手動ホストアクション」または「手動イベントアクション」が選択されている場合にのみ表示されます。

高度な設定

高度な設定オプションは、折りたたみ可能な高度な設定セクションで利用できます。



パラメータ	説明
ユーザー入力を有効化	スクリプトを実行する前に手動のユーザー入力を有効にするには、チェックボックスをオンにします。 手動のユーザー入力は、スクリプト内の {MANUALINPUT} マクロ値を置き換えます。 あわせて参照: 手動のユーザー入力。
入力プロンプト	カスタムのユーザー入力を促す任意のテキストを入力します。このテキストは、手動入力ポップアップの入力フィールドの上に表示されます。 手動入力ポップアップのプレビューを表示するには、ユーザー入力をテストをクリックします。プレビューでは、入力文字列が入力検証ルール（以下のパラメータを参照）に準拠しているかどうかもテストできます。 マクロおよびユーザーマクロのサポートは、スクリプトのスコープに依存します（一般的なスクリプト設定パラメータのスコープを参照）。
入力タイプ	手動入力のタイプを選択します。 文字列 - 単一の文字列。 ドロップダウン - 複数のドロップダウンオプションから値を選択します。

パラメータ	説明
ドロップダウンオプション	ユーザー入力用ドロップダウンの一意の値を、カンマ区切りのリストで入力します。 ドロップダウンに空のオプションを含めるには、リストの先頭、中間、または末尾に余分なカンマを追加します。 このフィールドは、入力タイプとして「ドロップダウン」が選択されている場合にのみ表示されます。
デフォルト入力文字列	ユーザー入力のデフォルト文字列を入力します（未設定も可）。 このフィールドは、入力検証ルールフィールドで指定された正規表現に対して検証されます。 ここで入力した値は、手動入力ポップアップにデフォルトで表示されます。 このフィールドは、入力タイプとして「文字列」が選択されている場合にのみ表示されます。
入力検証ルール	ユーザー入力文字列を検証するための正規表現を入力します。 グローバル正規表現をサポートしています。 このフィールドは、入力タイプとして「文字列」が選択されている場合にのみ表示されます。
確認を有効化	スクリプトを実行する前に確認メッセージを表示するには、チェックボックスをオンにします。この機能は、危険を伴う可能性のある操作（再起動スクリプトなど）や、長時間かかる可能性のある操作で特に役立ちます。
確認テキスト	上のチェックボックスで有効にした確認ポップアップ用の任意の確認テキストを入力します（例: リモートシステムが再起動されます。よろしいですか?)。テキストの表示例を確認するには、フィールドの横にある確認をテストをクリックします。 マクロおよびカスタムのユーザーマクロをサポートしています。 注: 確認メッセージのテスト時には、マクロは展開されません。

手動のユーザー入力と確認メッセージの両方が設定されている場合、それらは連続したポップアップウィンドウに表示されます。

手動ユーザー入力

手動ユーザー入力を使用すると、スクリプトの実行ごとにカスタムパラメータを指定できます。
これにより、1 つのパラメータだけが異なる複数の類似したユーザースクリプトを作成する必要がなくなります。

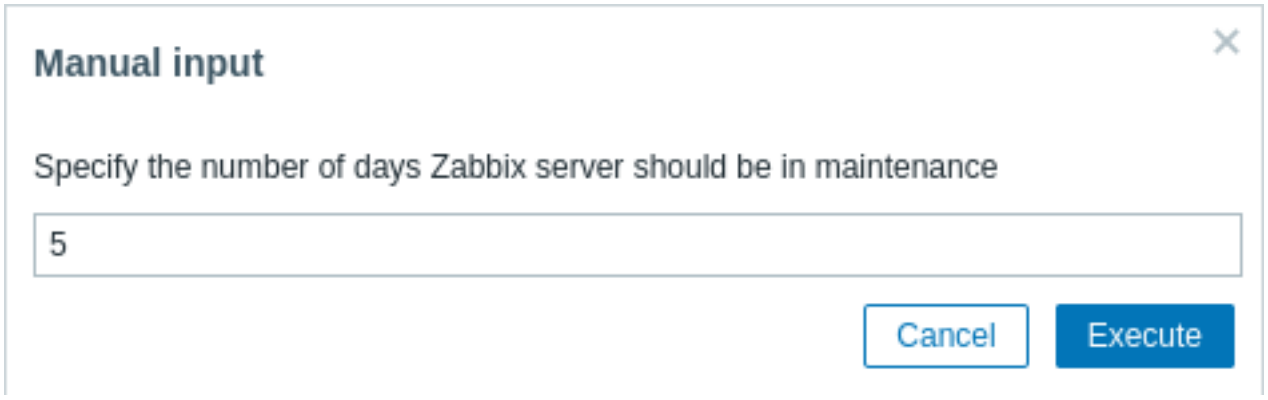
例えば、実行時に異なる整数値や異なる URL アドレスをスクリプトに渡したい場合があります。

手動ユーザー入力を有効にするには:

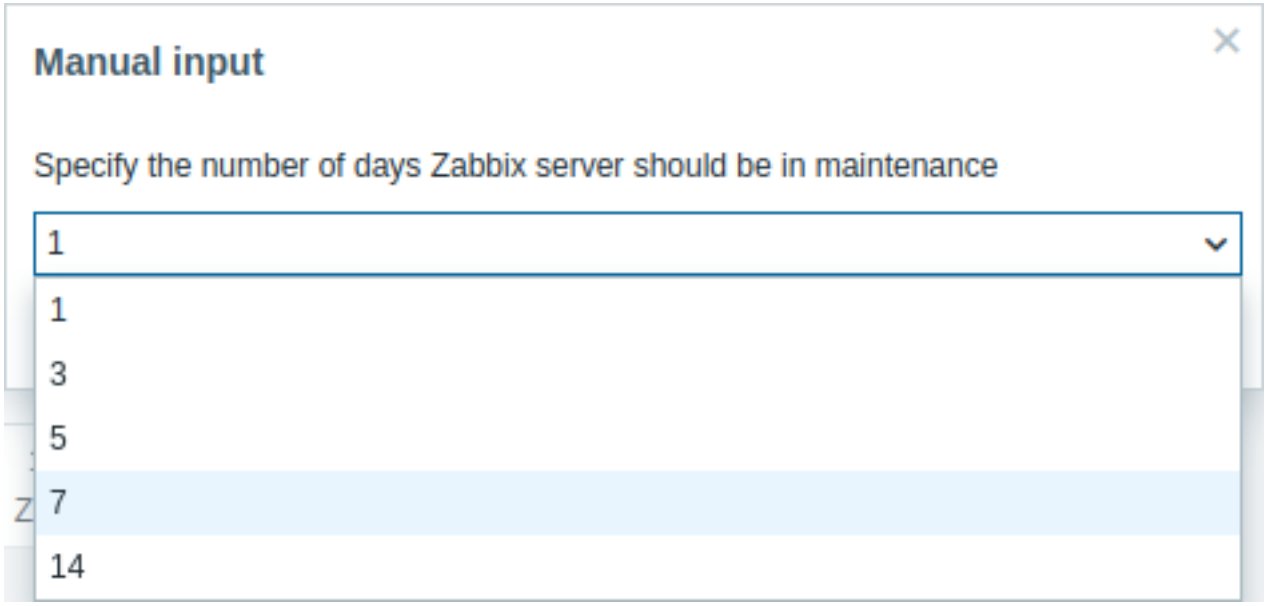
- 必要な箇所で、スクリプト（コマンド、スクリプト、スクリプトパラメータ）内、または URL スクリプトの URL フィールドで {MANUALINPUT} マクロを使用します。
- 高度なスクリプト設定 で、手動ユーザー入力を有効にし、入力オプションを設定します。

ユーザー入力を有効にすると、スクリプトの実行前に、ユーザーにカスタム値の入力を求める 手動入力ポップアップが表示されます。
入力された値は、スクリプト内の {MANUALINPUT} に置き換えられます。

設定に応じて、ユーザーには文字列値の入力が求められます:



または、あらかじめ定義されたオプションのドロップダウンから値を選択します:



手動ユーザー入力は、スコープが「手動ホストアクション」または「手動イベントアクション」であるスクリプトでのみ使用できます。

スクリプトの実行と結果

Zabbix サーバーによって実行されるスクリプトは、コマンド実行ページで説明されている順序で実行されます。

スクリプトの結果は、スクリプトの実行後に表示されるポップアップウィンドウに表示されます。スクリプトの戻り値は標準出力です。

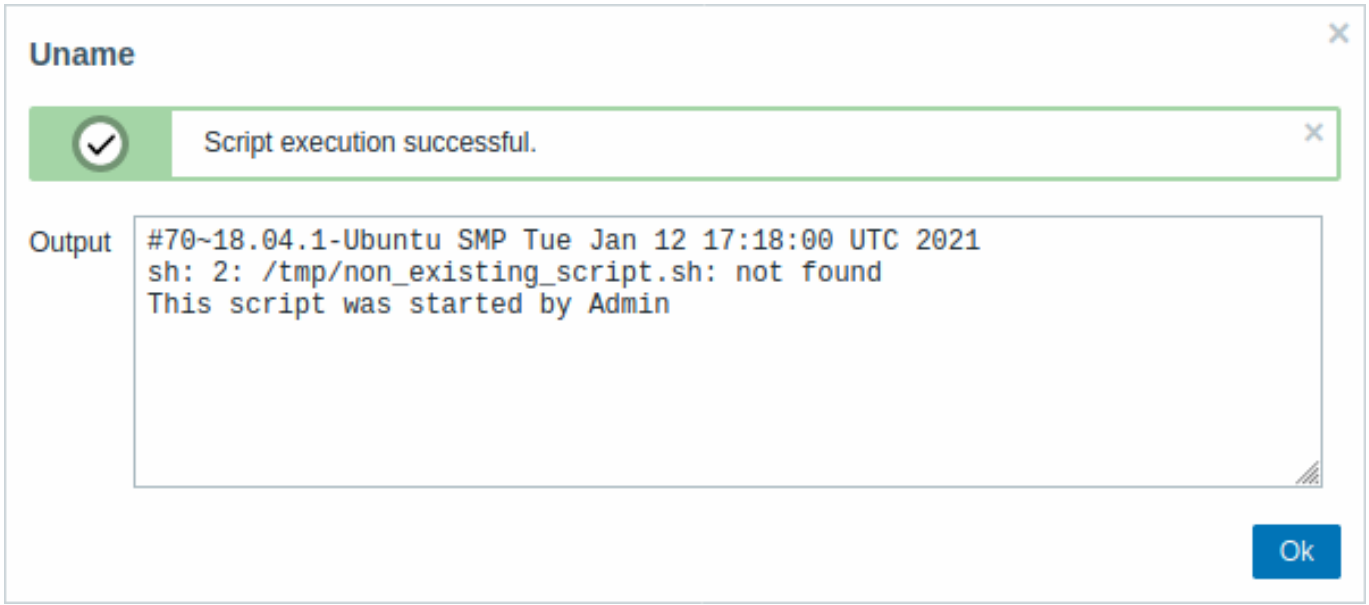
- スクリプトが正常に終了した場合（終了コード 0）、戻り値は 16MB に制限されます（切り捨てられる末尾の空白文字を含む）。
- スクリプトがエラーで終了した場合（終了コードが 0 以外）、戻り値は 2KB に制限された標準エラーです。

Zabbix は、デフォルトでは拡張されたスクリプト出力を保存しません。出力の完全な詳細を保持するには、スクリプト自体にロギングを実装できます（例: 出力をローカルのログファイルにリダイレクトする）。

Zabbix サーバーまたは Zabbix プロキシ上で実行されるスクリプトには、データベースの制限も適用されることに注意してください。

以下は、スクリプトと結果ウィンドウの例です。

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



スクリプトの結果には、スクリプト自体は表示されません。

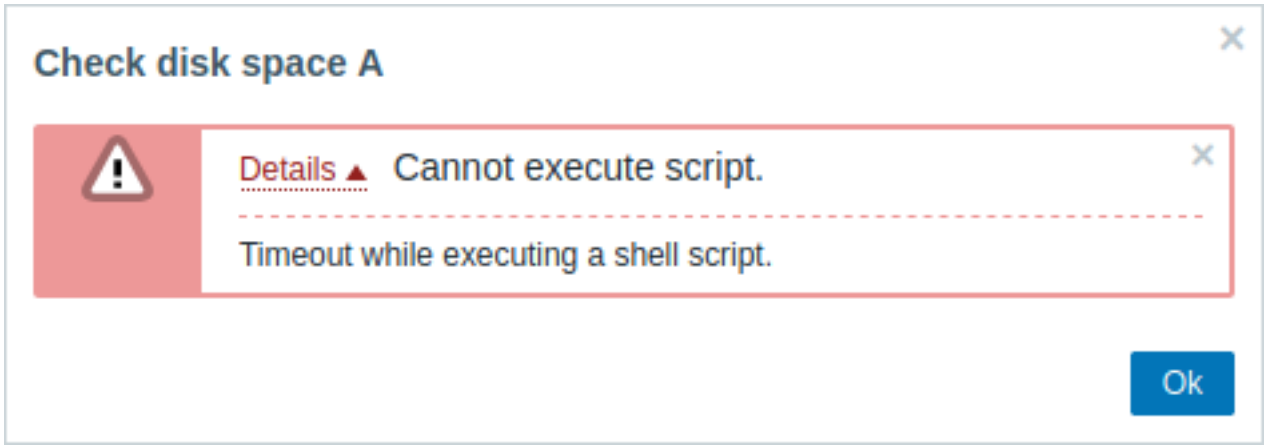
スクリプトのタイムアウト

Zabbix エージェント

スクリプトの実行中にタイムアウトが発生する場合があります。

以下は、Zabbix エージェント上で実行されるスクリプトと、その結果ウィンドウの例です。

```
sleep 5
df -h
```



この場合のエラーメッセージは次のとおりです。

```
Timeout while executing a shell script.
```

このような状況を回避するには、Zabbix エージェント設定 およびZabbix サーバー設定 の Timeout パラメータを調整するのではなく、スクリプト自体（上記の例では「5」）を最適化することを推奨します。
ただし、アクティブモードの Zabbix エージェントでは、Zabbix サーバー設定 の Timeout パラメータを、Zabbix エージェント設定 の RefreshActiveChecks パラメータより少なくとも数秒長く設定する必要があります。
これにより、サーバーはエージェントからアクティブチェックの結果を受信するための十分な時間を確保できます。
なお、アクティブなエージェントでのスクリプト実行は、Zabbix エージェント 7.0 以降でサポートされています。

Zabbix エージェント設定 で Timeout パラメータが変更されている場合、次のエラーメッセージが表示されます。

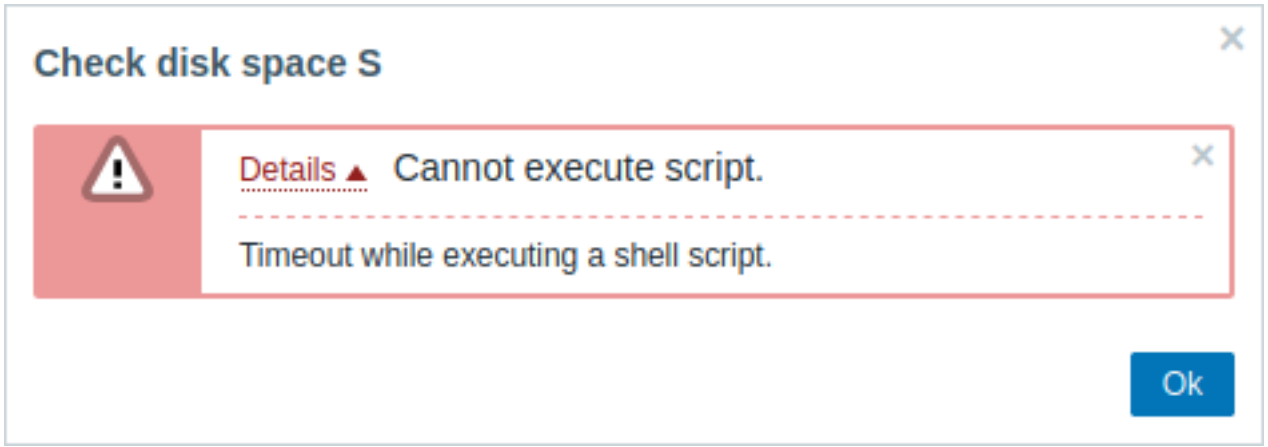
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

これは、Zabbix エージェント設定 では変更が行われているものの、Zabbix サーバー設定 の Timeout パラメータもあわせて変更する必要があることを意味します。

Zabbix サーバー/プロキシ

Zabbix サーバー上で実行されるスクリプトの例と、その結果ウィンドウを以下に示します。

```
sleep 11
df -h
```



また、Zabbix サーバー設定を変更して TrapperTimeout パラメータを対応する値（この場合は > 11）に調整するのではなく、スクリプト自体を最適化することも推奨されます。

8 ユーザー

概要

このメニューには、Zabbix でのユーザーの設定に関連するセクションがあります。このメニューは、SuperAdminタイプのユーザーのみが利用できます。

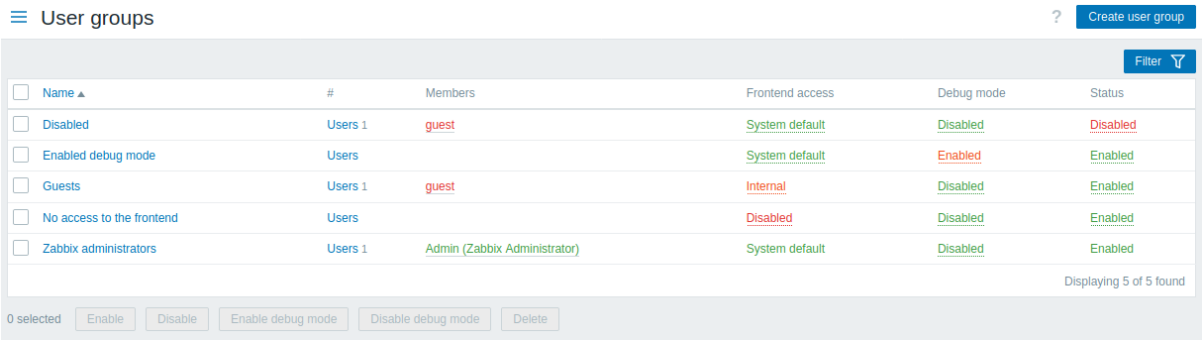
1 ユーザーグループ

概要

ユーザー → ユーザーグループセクションでは、システムのユーザーグループをメンテナンスできます。

ユーザーグループ

既存のユーザーグループの一覧と、その詳細が表示されます。



表示されるデータ:

Column	Description
Name	ユーザーグループの名前です。ユーザーグループ名をクリックすると、ユーザーグループの設定フォームが開きます。
#	グループ内のユーザー数です。*Users* をクリックすると、ユーザー一覧で該当するユーザーのみが表示されます。
Members	ユーザーグループに属する各ユーザーのユーザー名です（括弧内に名と姓を表示）。ユーザー名をクリックすると、ユーザー設定フォームが開きます。無効なグループのユーザーは赤色で表示されます。
Frontend access	Web インターフェースへのアクセスレベルが表示されます: **System default** - ユーザーは Zabbix、LDAP、または HTTP によって認証されます（グローバルに設定された認証方式に応じます）。 **Internal** - ユーザーは Zabbix によって認証されます。HTTP 認証がグローバルのデフォルトである場合は無視されます。 **LDAP** - ユーザーは LDAP によって認証されます。HTTP 認証がグローバルのデフォルトである場合は無視されます。 **Disabled** - このグループの Zabbix Web インターフェースへのアクセスは禁止されます。 現在のレベルをクリックすると、変更できます。
Debug mode	デバッグモードの状態が表示されます - *Enabled* または *Disabled*。状態をクリックすると変更できます。
Status	ユーザーグループの状態が表示されます - *Enabled* または *Disabled*。状態をクリックすると変更できます。

新しいユーザーグループを設定するには、右上隅にある *Create user group* ボタンをクリックします。

一括編集オプション

リストの下のボタンは、いくつかの一括編集オプションを提供します。

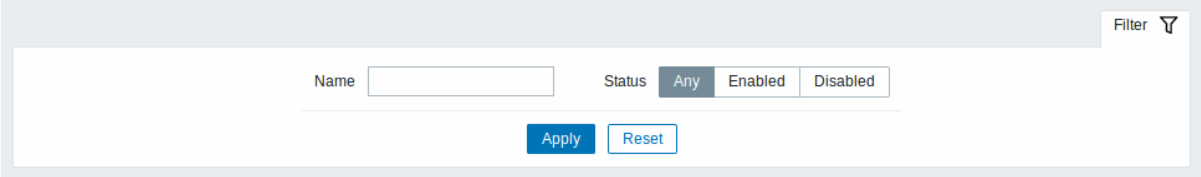
- 有効 - ユーザーグループのステータスを有効に
- 無効 - ユーザーグループのステータスを無効に
- デバッグモードを有効 - ユーザーグループのデバッグモードを有効に
- デバッグモードを無効 - ユーザーグループのデバッグモードを無効に
- 削除 - ユーザーグループを削除

これらのオプションを使用するには、それぞれのユーザーグループの前にチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルタの使用

フィルタを使用して、関心のあるユーザーグループのみを表示できます。検索パフォーマンスを向上させるために、データは未解決のマクロで検索されます。

フィルターリンクは、ユーザーグループのリストの上にあります。それをクリックすると、名前とステータスでユーザーグループをフィルタリングできます。



2 ユーザーの役割

概要

ユーザー > ユーザーロールセクションでは、ユーザーロールを作成できます。

ユーザーロールでは、最初に選択したユーザータイプ（*User*、*Admin*、*Super admin*）に基づいて、きめ細かな権限を作成できます。

ユーザータイプを選択すると、そのユーザータイプで利用可能なすべての権限が付与されます（デフォルトでチェックされています）。

権限は、そのユーザータイプで利用可能な権限の範囲内でのみ取り消すことができ、ユーザータイプで利用可能な範囲を超えて拡張することはできません。

利用できない権限のチェックボックスはグレーアウト表示されます。ユーザーは、この要素への直接 URL をブラウザに入力しても、その要素にアクセスすることはできません。

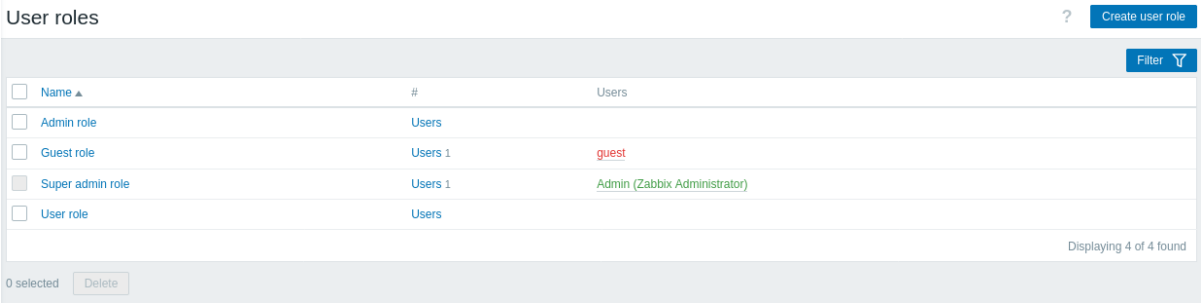
Note:
一部の UI 要素へのアクセスを制限しても、そのページを開けなくなるだけであり、インターフェースの他の部分にある基盤データへアクセスする機能自体がなくなるわけではありません。

ユーザーロールはシステムユーザーに割り当てることができます。各ユーザーに割り当てられるロールは 1 つだけです。

デフォルトのユーザーロール

デフォルトでは、Zabbix は 4 つのユーザーロールで設定されており、それぞれに事前定義された権限セットがあります。

- ゲストロール
- ユーザーロール
- 管理者ロール
- 特権管理者ロール

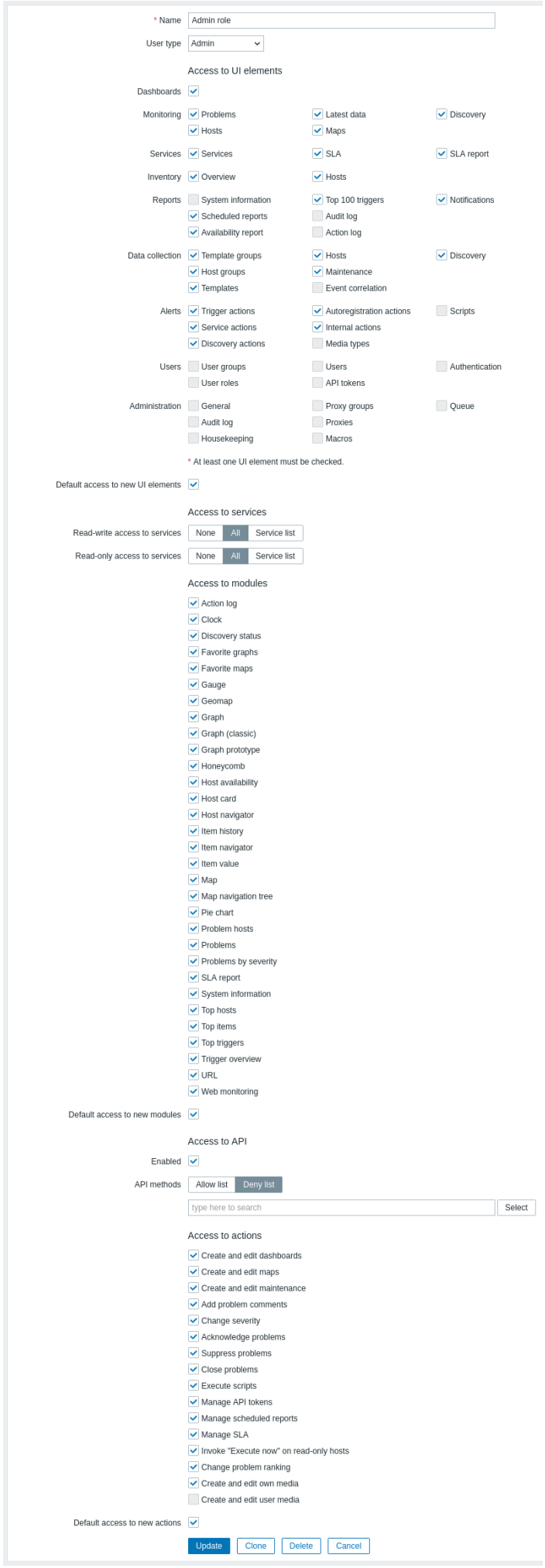


これらは、Zabbix の主要なユーザータイプに基づいています。それぞれのロールに割り当てられているすべてのユーザーの一覧が表示されます。無効化されたグループに含まれるユーザーは赤色で表示されます。ゲストロールは、一部の Web インターフェースセクションを表示する権限のみを持つユーザータイプロールです。

Note:
デフォルトの特権管理者ロールは、変更または削除できません。これは、無制限の権限を持つ特権管理者ユーザーが Zabbix に少なくとも 1 人存在している必要があるためです。特権管理者タイプのユーザーは、自身のロールの設定を変更できますが、ユーザータイプは変更できません。

設定

新しいロールを作成するには、右上隅にある ユーザーロールの作成ボタンをクリックします。既存のロールを更新するには、ロール名をクリックして設定フォームを開きます。


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
type here to search
Select
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Default access to new actions
Update
Clone
Delete
Cancel

利用可能な権限が表示されます。特定の権限を取り消すには、
そのチェックボックスの選択を外します。

利用可能な権限と、Zabbix で事前設定された各ユーザーロールのデフォルト値については、以下で説明します。

デフォルト権限

UI 要素へのアクセス

メニューセクションへのデフォルトアクセスは、ユーザータイプによって異なります。詳細は、Permissions ページを参照してください。

その他のオプションへのアクセス

パラメータ		説明	デフォルトのユーザーロール			
			Super admin role	*Admin role*	*User role*	*Guest role*
新しい *UI* 要素へのデフォルトアクセス		このオプションは、Zabbix アップグレード後に新しいメニューセクションへどのようにアクセス可能にするかを指定します。モジュールの既存のメニューセクションには影響しません。	Yes	Yes	Yes	Yes
サービスへのアクセス	サービスへの読み書きアクセス	サービスへの読み書きアクセスを選択します: **None** - アクセス不可 **All** - すべてのサービスに読み書き可能 **Service list** - 読み書きアクセスを許可するサービスを選択 読み書きアクセスが許可されている場合、読み取り専用アクセス設定より優先され、子サービスに動的に継承されます。	All	All	None	None

	タグ付きサービスへの読み書きアクセス	タグ名と、必要に応じて値を指定して、タグに一致するサービスに対して追加で読み書きアクセスを許可します。 このオプションは、サービスへの読み書きアクセスパラメータで'Service list' が選択されている場合に使用できます。 読み書きアクセスが許可されている場合、読み取り専用アクセス設定より優先され、子サービスに動的に継承されます。				
	サービスへの読み取り専用アクセス	サービスへの読み取り専用アクセスを選択します: **None** - アクセス不可 **All** - すべてのサービスに読み取り専用でアクセス可能 **Service list** - 読み取り専用アクセスを許可するサービスを選択 読み取り専用アクセスは読み書きアクセスより優先されず、子サービスに動的に継承されます。			All	All
	タグ付きサービスへの読み取り専用アクセス	タグ名と、必要に応じて値を指定して、タグに一致するサービスに対して追加で読み取り専用アクセスを許可します。 このオプションは、サービスへの読み取り専用アクセスパラメータで'Service list' が選択されている場合に使用できます。 読み取り専用アクセスは読み書きアクセスより優先されず、子サービスに動的に継承されます。				
モジュールへのアクセス						
	<Module name>	特定のモジュールへのアクセスを許可/拒否します。このセクションには、有効化されているモジュールのみが表示されます。現在無効化されているモジュールへのアクセスを許可または制限することはできません。	Yes	Yes	Yes	Yes
	新しいモジュールへのデフォルトアクセス	このオプションは、Zabbix アップグレード後に新しいモジュールおよびウィジェットへどのようにアクセス可能にするかを指定します。また、*Administration* > *General* > *Modules section* に追加されたモジュールおよびウィジェットにも適用されます。				
API へのアクセス						
	Enabled	API へのアクセスを有効/無効にします。	Yes	Yes	Yes	No

	API methods	検索フィールドで指定した API メソッドを許可するには *Allow list*、拒否するには *Deny list* を選択します。一部の API メソッドを許可し、他を拒否することはできません。 検索フィールドでメソッド名の入力を開始し、自動補完リストからメソッドを選択します。 また、Select ボタンを押して、このユーザータイプで利用可能な完全な一覧からメソッドを選択することもできます。Access to actions ブロック内の特定のアクションがオフになっている場合、ユーザーはそのアクションに関連する API メソッドを使用できません。 ワイルドカードをサポートしています。例: `dashboard.*` ('dashboard.' API サービスのすべてのメソッド) `*` (任意のメソッド) `*.export` (すべての API サービスの'.export' 名を持つメソッド)。 メソッドが 1 つも指定されていない場合、*Allow/Deny list* ルールは無視されます。				
アクションへのアクセス	ダッシュボードの作成と編集	このチェックボックスをオフにすると、対応する要素に対する `.create`、`.update`、`.delete` API メソッドの使用権限も取り消されます。	Yes	Yes	Yes	No
	マップの作成と編集					
	メンテナンスの作成と編集				No	
	障害コメントの追加	このチェックボックスをオフにすると、`event.acknowledge` API メソッドによる対応するアクションの実行権限も取り消されます。			Yes	
	深刻度の変更					
	障害の確認					
	障害の抑制					
	障害のクローズ					
	スクリプトの実行	このチェックボックスをオフにすると、`script.execute` API メソッドの使用権限も取り消されます。				
	API トークンの管理	このチェックボックスをオフにすると、すべての `token.` API メソッドの使用権限も取り消されます。				
	定期レポートの管理	このチェックボックスをオフにすると、すべての `report.` API メソッドの使用権限も取り消されます。			No	
	SLA の管理	SLA を管理する権限を有効/無効にします。				
	読み取り専用ホストでの「Execute now」の実行	読み取り専用ホストのアイテムに対して、最新データで"Execute now" オプションの使用を許可します。			Yes	
	障害ランキングの変更	障害ランキングを原因から症状へ、またはその逆に変更することを許可します。				
	自分自身のメディアの作成と編集	自分自身のメディアの作成/編集を許可します。				

ユーザーメディアの作成と編集	ユーザーのメディアの作成/編集を許可します。このオプションは Super admin ユーザーでのみ使用できます。	No	No
新しいアクションへのデフォルトアクセス	このオプションは、Zabbix アップグレード後に新しいアクションへどのようにアクセス可能にするかを指定します。	Yes	Yes

関連項目:

- ユーザーの設定

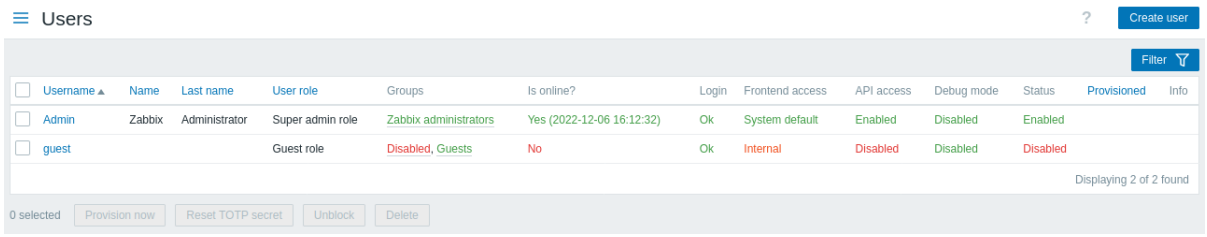
3 ユーザー

概要

ユーザー → ユーザーセクションでシステムのユーザーを管理できます。

ユーザー

既存のユーザーの一覧とその詳細が表示されます。



表示されるデータ:

Column	Description
Username	Zabbix へのログインに使用するユーザー名です。ユーザー名をクリックすると、ユーザーの設定フォームが開きます。
Name	ユーザーの名です。
Last name	ユーザーの姓です。
User role	ユーザーロールが表示されます。
Groups	ユーザーが所属するグループの一覧が表示されます。ユーザーグループ名をクリックすると、ユーザーグループの設定フォームが開きます。無効なグループは赤色で表示されます。
Is online?	ユーザーのオンライン状態が表示されます - *Yes* または *No*。最後のユーザー操作時刻が括弧内に表示されます。
Login	ユーザーのログイン状態が表示されます - *Ok* または *Blocked*。ユーザーは、*Administration → General → Other* セクションで設定されたログイン失敗回数の上限（デフォルトでは 5 回）を超えると、一時的にブロックされることがあります。*Blocked* をクリックすると、ユーザーのブロックを解除できます。
Frontend access	Web インターフェースへのアクセスレベルが表示されます - *System default*、*Internal*、*LDAP*、または *Disabled*。表示内容は、ユーザーグループ全体に設定されている値によって決まります。
API access	API アクセス状態が表示されます - *Enabled* または *Disabled*。表示内容は、ユーザーロールに設定されている値によって決まります。
Debug mode	デバッグモードの状態が表示されます - *Enabled* または *Disabled*。表示内容は、ユーザーグループ全体に設定されている値によって決まります。
Status	ユーザーの状態が表示されます - *Enabled* または *Disabled*。表示内容は、ユーザーグループ全体に設定されている値によって決まります。
Provisioned	ユーザーが最後にプロビジョニングされた日付が表示されます。 LDAP/SAML からの JIT プロビジョニングによって作成されたユーザーに使用されます。
Info	エラーに関する情報が表示されます。 ユーザーグループに所属していないユーザーには黄色の警告が表示されます。 ロールがないユーザー、およびロールもユーザーグループもないユーザーには赤色の警告が表示されます。

新しいユーザーを設定するには、右上隅にある *Create user* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

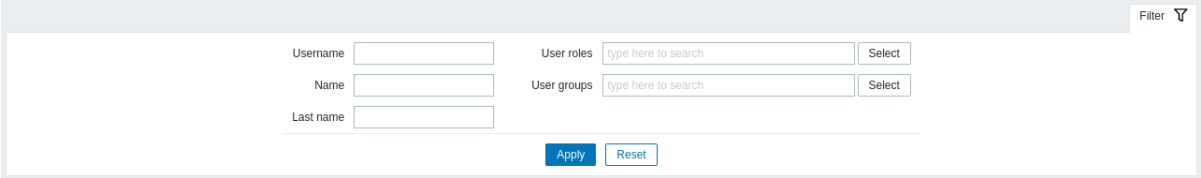
- 今すぐプロビジョニング - LDAP からユーザー情報を更新します（このオプションは、LDAP ユーザーが選択されている場合にのみ有効です）
- *TOTP* シークレットをリセット - すべての TOTP 方式のユーザー TOTP シークレットをリセットし、ユーザーセッションを削除します（このオプションは、MFA が有効な場合にのみ有効です。TOTP シークレットを持たないユーザーについては、セッションは削除されません）
- ブロック解除 - ブロックされたユーザーのシステムアクセスを再度有効にします
- 削除 - ユーザーを削除します

これらのオプションを使用するには、対象のユーザーの前にあるチェックボックスをオンにしてから、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるユーザーのみを表示できます。
検索パフォーマンスを向上させるため、データはマクロを未展開のまま検索されます。

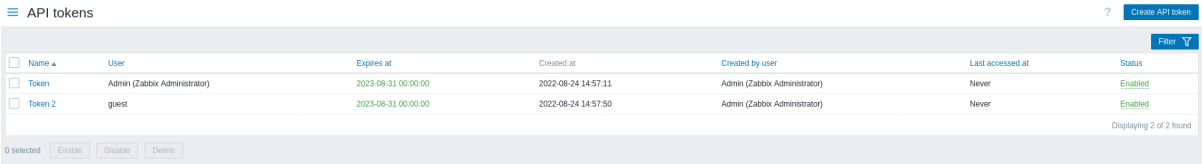
ユーザー一覧の上にある フィルターリンクを利用できます。これをクリックすると、ユーザー名、名、姓、ユーザーロール、およびユーザーグループでユーザーを絞り込めるフィルターが表示されます。



4 API トークン

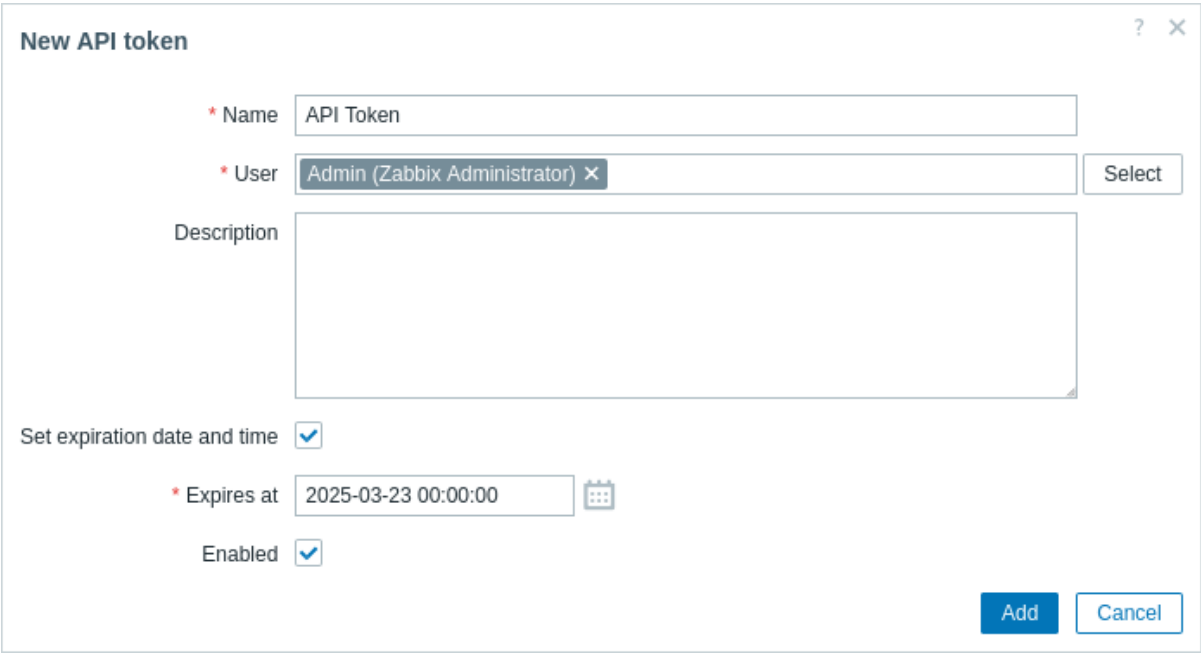
概要

このセクションでは、API トークンの作成と管理を行えます。



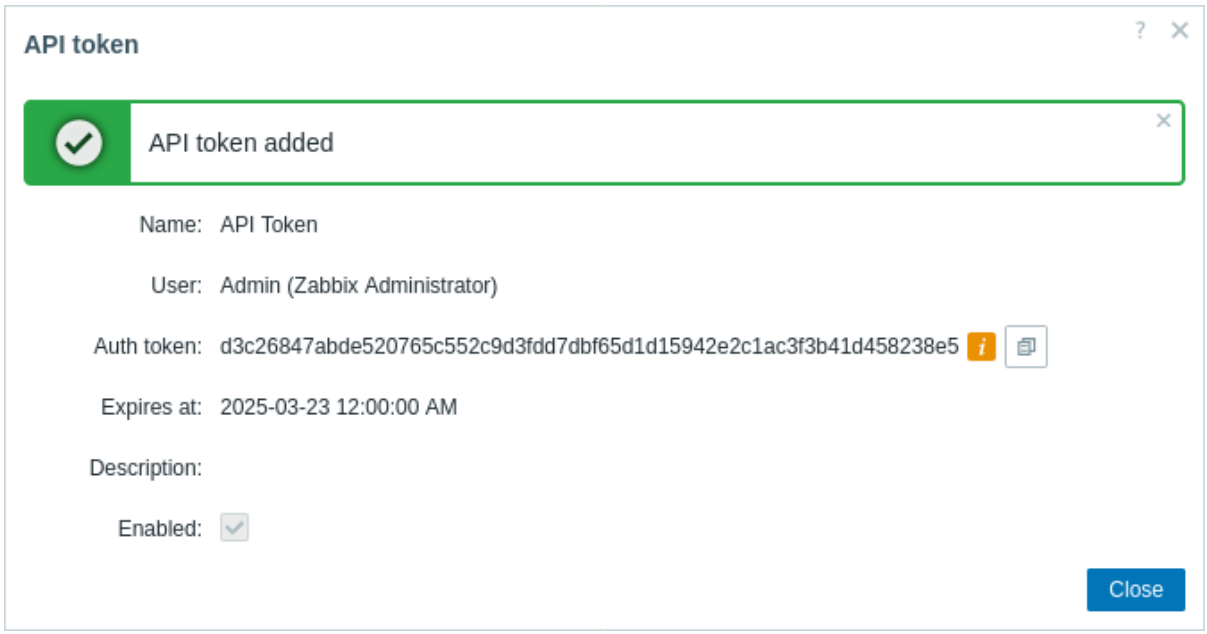
API トークンは、名前、トークンが割り当てられているユーザー、有効期限、トークンを作成したユーザー、またはステータス（有効/無効）で絞り込むことができます。リスト内のトークンのステータスをクリックすると、トークンをすばやく有効/無効にできます。また、リスト内でトークンを選択し、リストの下にある *Enable/Disable* ボタンをクリックすることで、トークンを一括で有効/無効にすることもできます。

新しいトークンを作成するには、右上隅の *Create API token* ボタンを押し、トークン設定画面で必要な項目を入力します。



Parameter	Description
名前	トークンの表示名。
ユーザー	トークンを割り当てるユーザー。ユーザーをすばやく選択するには、ユーザー名、名、または姓の入力を開始し、自動補完リストから必要なユーザーを選択します。あるいは、*Select* ボタンを押して、完全なユーザー一覧からユーザーを選択することもできます。トークンを割り当てられるユーザーは 1 人のみです。
説明	任意のトークンの説明。
有効期限の日時を設定	トークンに有効期限を設定しない場合は、このチェックボックスをオフにします。
有効期限	カレンダーアイコンをクリックしてトークンの有効期限を選択するか、YYYY-MM-DD hh:mm:ss 形式で日付を手動入力します。
有効	無効状態のトークンを作成する必要がある場合は、このチェックボックスをオフにします。

トークンを作成するには、*Add* を押します。



Auth token の値をコピーし、ページを閉じる前に安全な場所に保存してから、Close を押します。トークンが一覧に表示されます。

Warning:
Auth token の値は、後で再表示することはできません。この値は、トークン作成直後にのみ利用可能です。保存したトークンを紛失した場合は再生成する必要があり、その際には新しい認証文字列が作成されます。

トークン名をクリックすると、名前、説明、有効期限の設定、またはトークンのステータスを編集できます。なお、トークンの割り当て先ユーザーを変更することはできません。変更を保存するには、*Update* ボタンを押します。トークンを紛失した場合や漏えいした場合は、*Regenerate* ボタンを押して新しいトークン値を生成できます。確認ダイアログが表示され、この操作を続行すると以前に生成されたトークンが無効になるため、確認を求められます。

Administration メニューセクションにアクセスできないユーザーでも、user role の権限で *Manage API tokens* が許可されている場合に限り、*User profile* → *API tokens* section で自分に割り当てられたトークンの詳細を表示および変更できます。

5 認証

概要

ユーザー → 認証セクションでは、Zabbix のユーザー認証方式と内部パスワード要件を指定できます。

利用可能な認証方式は、内部認証、HTTP、LDAP、SAML、および MFA 認証です。

デフォルト認証

デフォルトでは、Zabbix はすべてのユーザーに対して 内部 Zabbix 認証を使用します。

デフォルトの認証方式は、システム全体でLDAP に変更できます。これを行うには、*LDAP* タブに移動して LDAP パラメータを設定し、その後 *Authentication* タブに戻って、*Default authentication* セレクターを LDAP に切り替えます。

認証方式は、ユーザーグループ レベルで細かく調整できることに注意してください。LDAP 認証がグローバルに設定されている場合でも、一部のユーザーグループは引き続き Zabbix によって認証できます。これらのグループでは、frontend access を Internal に設定する必要があります。

内部認証がグローバルに使用されている場合でも、特定のユーザーグループに対してのみ LDAP 認証を有効にすることも可能です。この場合、LDAP 認証の詳細は、frontend access を LDAP に設定する必要がある特定のユーザーグループに対して指定して使用できます。ユーザーが LDAP 認証を持つユーザーグループに少なくとも 1 つ含まれている場合、そのユーザーは内部認証方式を使用できません。

HTTP、SAML 2.0、および MFA の認証方式は、デフォルトの認証方式に加えて使用できます。

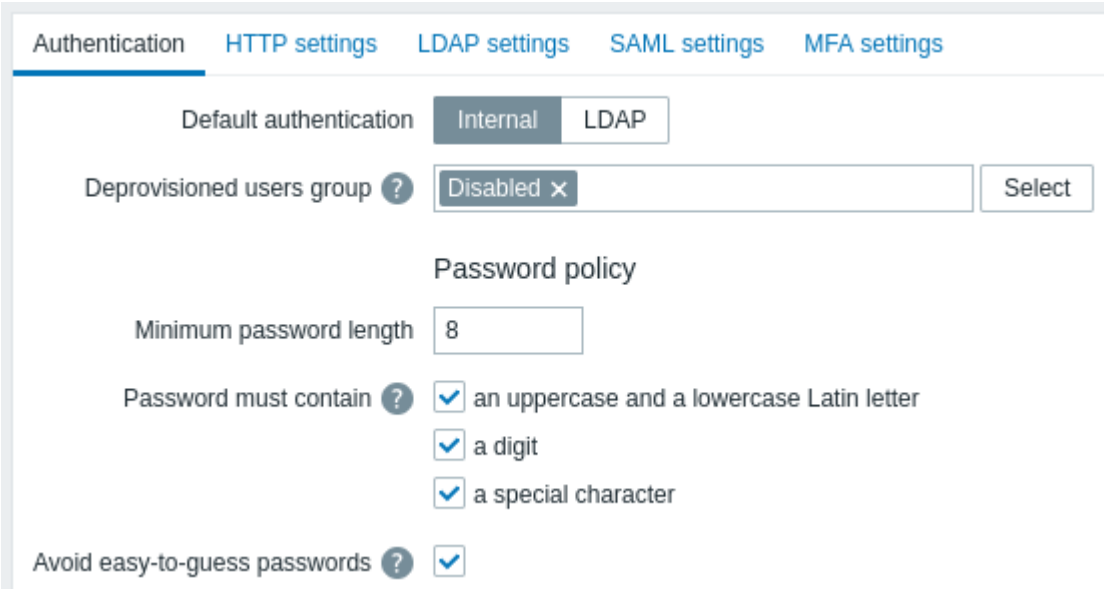
Zabbix はジャストインタイム (JIT) プロビジョニングをサポートしており、外部ユーザーが初めて認証した際に Zabbix 内にユーザーアカウントを作成し、それらのユーザーアカウントをプロビジョニングできます。JIT プロビジョニングは LDAP および SAML でサポートされています。

関連情報:

- HTTP authentication
- LDAP authentication
- SAML authentication
- MFA authentication

設定

認証タブでは、デフォルトの認証方式の設定、プロビジョニング解除済みユーザー用のグループの指定、および Zabbix ユーザーのパスワード複雑性要件の設定ができます。



設定パラメータ:

Parameter	Description
デフォルトの認証	Zabbix のデフォルト認証方式として、内部または *LDAP* を選択します。
プロビジョニング解除済みユーザーグループ	プロビジョニング解除済みユーザー用のユーザーグループを指定します。この設定は JIT プロビジョニングでのみ必要で、LDAP または SAML システムから Zabbix に作成されたものの、以後はプロビジョニングする必要がなくなったユーザーに関係します。 無効化されたユーザーグループを指定する必要があります。
最小パスワード長	デフォルトでは、最小パスワード長は 8 に設定されています。サポート範囲: 1～70。72 文字を超えるパスワードは切り捨てられることに注意してください。
パスワードに含める必要がある文字	1 つまたは複数のチェックボックスを選択して、パスワードに指定した文字の使用を必須にします: - 英大文字および英小文字 - 数字 - 特殊文字 各オプションで使用される文字の一覧は、疑問符にマウスオーバーするとヒントとして表示されます。

Parameter	Description
推測されやすいパスワードを避ける	チェックすると、パスワードは次の要件に照らして確認されます: - ユーザーの名、姓、またはユーザー名を含んではならない - 一般的なパスワードまたはコンテキスト固有のパスワードのいずれでもあってはならない 一般的なパスワードおよびコンテキスト固有のパスワードの一覧は、NCSC の「Top 100k passwords」の一覧、SecLists の「Top 1M passwords」の一覧、および Zabbix のコンテキスト固有パスワードの一覧から自動生成されます。内部ユーザーは、この一覧に含まれるパスワードを設定できません。これらのパスワードは、一般的に使用されているため脆弱と見なされるためです。

パスワード複雑性要件の変更は既存ユーザーのパスワードには影響しませんが、既存ユーザーがパスワードを変更する場合、新しいパスワードは現在の要件を満たす必要があります。要件一覧を示すヒントは、ユーザープロファイル および *Users* → *Users* メニューからアクセスできるユーザー設定フォーム の *Password* フィールドの横に表示されます。

1 HTTP

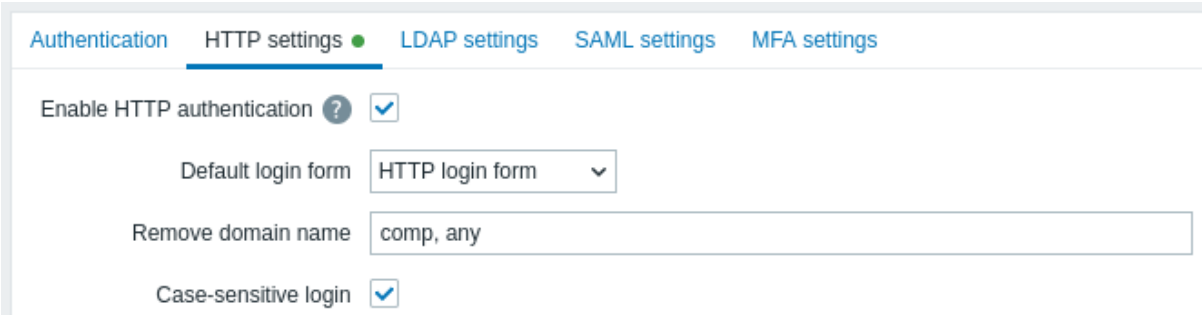
概要

HTTP または Web サーバーベースの認証 （例: BasicAuthentication、NTLM/Kerberos）を使用して、ユーザー名とパスワードを確認できます。ただし、ユーザーは Zabbix にも存在している必要がありますが、その Zabbix パスワードは使用されません。

Attention:
注意してください！Web サーバー認証を有効にする前に、認証が設定され、正しく動作していることを必ず確認してください。

HTTP 認証は、ユーザー > 認証セクションの *HTTP* 設定タブで対応するオプションを設定することで、Web インターフェースで無効にできます。HTTP 認証が無効になっている場合、HTTP 認証オプションのタブは Web インターフェースに表示されません。なお、Web インターフェースを再インストール（setup.php の実行）すると、HTTP 認証の設定を含む認証設定がリセットされます。

設定



設定パラメータ:

Parameter	Description
HTTP 認証を有効にする	チェックボックスをオンにして HTTP 認証を有効にします。 にマウスカーソルを合わせると、Web サーバー認証の場合は、Web インターフェースアクセス が LDAP/Internal に設定されている場合でも、すべてのユーザーが Zabbix ではなく Web サーバーによって認証されることを警告するヒントボックスが表示されます。
デフォルトのログインフォーム	未認証ユーザーを次のいずれに誘導するかを指定します: **Zabbix** ログインフォーム - 標準の Zabbix ログインページ。 **HTTP** ログインフォーム - HTTP ログインページ。 Web サーバーベースの認証は、`index_http.php` ページに対してのみ有効にすることを推奨します。デフォルトのログインフォームが「HTTP ログインページ」に設定されている場合、Web サーバー認証モジュールが `$_SERVER` 変数に有効なユーザーログインを設定すると、ユーザーは自動的にログインされます。 サポートされる `$_SERVER` キーは `PHP_AUTH_USER`、`REMOTE_USER`、`AUTH_USER` です。
ドメイン名を削除	ユーザー名から削除するドメイン名のカンマ区切りリスト。 例: `comp,any` - ユーザー名が'Admin@any' または'comp\Admin' の場合、ユーザーは'Admin' としてログインします。ユーザー名が'notacompany\Admin' の場合、ログインは拒否されます。

Parameter	Description
大文字と小文字を区別するログイン	チェックボックスをオフにすると、ユーザー名の大文字と小文字を区別するログインを無効にします（デフォルトで有効）。 大文字と小文字を区別するログインを無効にすると、たとえば Zabbix ユーザーが”Admin” または”ADMIN” であっても、”admin” としてログインできます。 大文字と小文字を区別するログインが無効で、類似したユーザー名を持つ複数の Zabbix ユーザー（例: Admin と admin）が存在する場合、それらのユーザーのログインは常に次のエラーメッセージとともに拒否されることに注意してください: ”Authentication failed: supplied credentials are not unique.”

Note:
HTTP 認証情報を使用してログインできない内部ユーザー（デフォルトで HTTP ログインフォームが設定されている場合）で 401 エラーになる場合は、基本認証ディレクティブに `ErrorDocument 401 /index.php?form=default` の行を追加することを検討してください。これにより、通常の Zabbix ログインフォームにリダイレクトされます。

2 LDAP

概要

外部 LDAP 認証 を使用して、ユーザー名とパスワードを確認できます。

Zabbix の LDAP 認証は、少なくとも Microsoft Active Directory および OpenLDAP で動作します。

LDAP サインインのみが設定されている場合、ユーザーは Zabbix にも存在している必要がありますが、その Zabbix パスワードは使用されません。
認証が成功すると、Zabbix はローカルのユーザー名を、LDAP から返されたユーザー名属性と照合します。

ユーザープロビジョニング

LDAP ユーザーに対して、JIT（just-in-time）ユーザープロビジョニングを設定できます。この場合、ユーザーがあらかじめ Zabbix に存在している必要はありません。ユーザーアカウントは、ユーザーが初めて Zabbix にログインしたときに作成できます。

LDAP ユーザーが LDAP のログイン名とパスワードを入力すると、Zabbix はこのユーザーが存在するかどうかをデフォルトの LDAP サーバーで確認します。ユーザーが存在し、まだ Zabbix にアカウントを持っていない場合、Zabbix 内に新しいユーザーが作成され、ユーザーはログインできるようになります。

JIT プロビジョニングによって作成されたユーザーは、作成時点でデフォルトとして設定されている LDAP サーバー（ディレクトリ）に関連付けられます。後でデフォルト LDAP サーバーを変更しても、すでにプロビジョニングされたユーザーに関連付けられている LDAP サーバーは変更または更新されません。

Attention:
JIT プロビジョニングが有効な場合、*Authentication* タブでプロビジョニング解除済みユーザー用のユーザーグループを指定する必要があります。

JIT プロビジョニングでは、LDAP の変更に基づいてプロビジョニング済みユーザーアカウントを更新することもできます。たとえば、ユーザーがある LDAP グループから別の LDAP グループに移動された場合、Zabbix でもそのユーザーはあるグループから別のグループへ移動されます。ユーザーが LDAP グループから削除された場合、Zabbix でもそのグループから削除され、他のどのグループにも属していなければ、プロビジョニング解除済みユーザー用のユーザーグループに追加されます。プロビジョニング済みユーザーアカウントは、設定されたプロビジョニング期間に基づいて、またはユーザーが Zabbix にログインしたときに更新されます。

バックグラウンドプロビジョニングは、ユーザーが Zabbix Web インターフェースを操作しているとき、または少なくともブラウザで Web インターフェースのページを開いているときに、Zabbix Web インターフェースによって実行されることに注意してください。ユーザーをプロビジョニングするための専用バックグラウンドプロセスはありません。

LDAP は、認証および検索のためにディレクトリへバインドする 3 つの方法をサポートしています。

- 匿名バインド — *Bind DN* / *Bind password* を指定せず、LDAP サーバーが匿名クエリを許可している場合です。
- 専用バインドユーザー（サービスアカウント）— 特定の LDAP アカウントを *Bind DN* / *Bind password* に設定し、Zabbix が検索およびプロビジョニングに使用します。これは、エンドユーザーの認証情報なしで Zabbix が検索やバックグラウンドプロビジョニングを実行できるため、推奨される最も柔軟なオプションです。
- 直接ユーザーバインド — Zabbix は、ユーザーがログイン時に入力した認証情報を使用してバインドを実行します（*Bind DN* / *Bind password* は設定しません）。これは、Base DN に `uid=%{user}` のようなプレースホルダーを含めることで設定します。このモードでは、Zabbix がユーザーのパスワードにアクセスできるのは対話的ログイン中のみです。その結果、ユーザーのログインセッション外で LDAP への認証を必要とするプロビジョニング操作（たとえば、*Provision now* ボタンの使用や、ユーザ

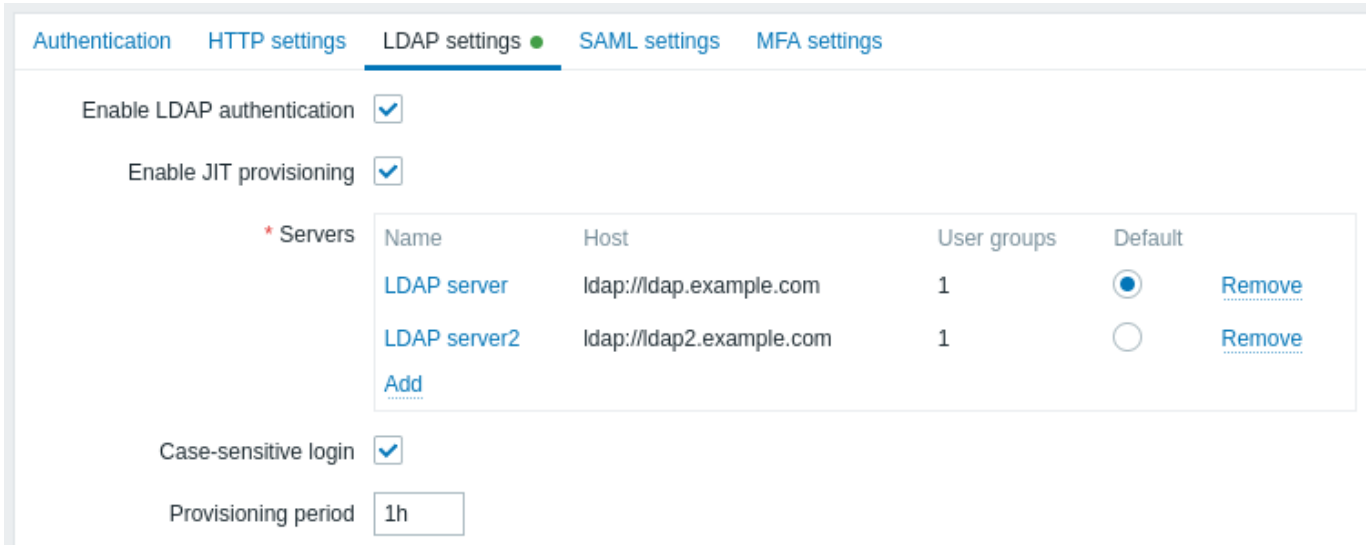
ーがアクティブにログインしていないときのバックグラウンドプロビジョニングの実行）は認証できないため、動作しません。
直接ユーザーバインドにおけるプロビジョニングおよび更新は、ユーザーがログインした時点でのみ行われます。

複数のサーバー

必要に応じて、複数の LDAP サーバーを定義できます。
例えば、異なるユーザーグループの認証に別々のサーバーを使用できます。
LDAP サーバーを設定すると、ユーザーグループの設定で、それぞれのユーザーグループに必要な LDAP サーバーを選択できるようになります。

ユーザーが複数のユーザーグループに属し、かつ複数の LDAP サーバーに存在する場合、認証には LDAP サーバーの一覧を名前の昇順で並べ替えた際の先頭のサーバーが使用されます。

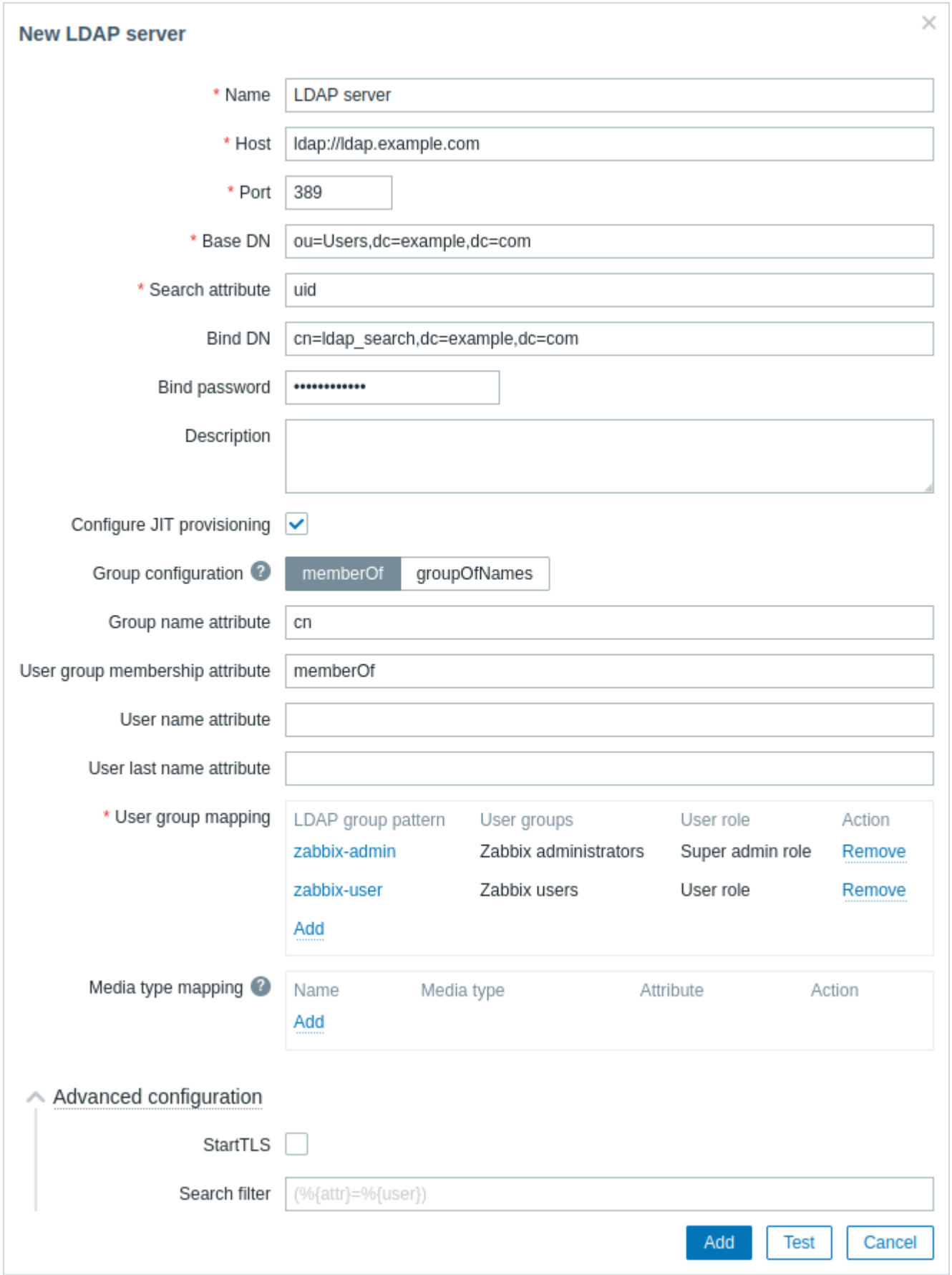
設定



設定パラメータ:

Parameter	Description
Enable LDAP authentication	チェックボックスをオンにして、LDAP 認証を有効にします。
Enable JIT provisioning	チェックボックスをオンにして、JIT プロビジョニングを有効にします。
Servers	*Add* をクリックして LDAP サーバーを設定します（以下のLDAP サーバー設定を参照）。
Case-sensitive login	ユーザー名の大文字・小文字を区別するログインを無効にするには、チェックボックスをオフにします（デフォルトで有効）。 大文字・小文字を区別するログインを無効にすると、たとえば Zabbix ユーザーが”Admin” または”ADMIN” であっても、”admin” としてログインできます。 大文字・小文字を区別するログインが無効で、類似したユーザー名を持つ複数の Zabbix ユーザー（例: Admin と admin）が存在する場合、それらのユーザーのログインは常に次のエラーメッセージで拒否されることに注意してください: ”Authentication failed: supplied credentials are not unique.”
Provisioning period	プロビジョニング期間、つまりログイン中のユーザーが Web インターフェースの使用中にどのくらいの頻度でプロビジョニングされるかを設定します。

LDAP サーバーの設定



LDAP サーバーの設定パラメータ:

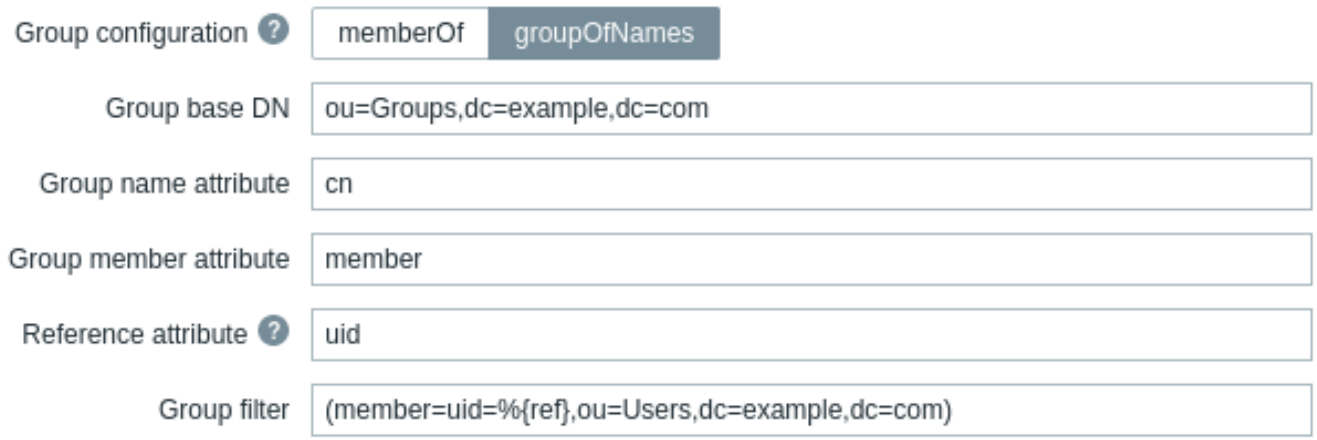
Parameter	Description
Name	Zabbix 設定内の LDAP サーバー名。

Parameter	Description
Host	LDAP サーバーのホスト名、IP、または URI。例: ldap.example.com, 127.0.0.1, ldap://ldap.example.com セキュアな LDAP サーバーの場合は、*ldaps* プロトコルとホスト名を使用します。例: ldaps://ldap.example.com OpenLDAP 2.x.x 以降では、ldap://hostname:port または ldaps://hostname:port の形式の完全な LDAP URI を使用できます。
Port	LDAP サーバーのポート。デフォルトは 389 です。 セキュアな LDAP 接続のポート番号は通常 636 です。 完全な LDAP URI を使用する場合は使用されません。
Base DN	LDAP サーバー内のユーザーアカウントへのベースパス: ou=Users,ou=system (OpenLDAP の場合), DC=company,DC=com (Microsoft Active Directory の場合) uid=%{user},dc=example,dc=com (直接ユーザーバインドの場合。以下の注記を参照)
Search attribute	検索に使用する LDAP アカウント属性: uid (OpenLDAP の場合), sAMAccountName (Microsoft Active Directory の場合)
Bind DN	LDAP サーバーへのバインドおよび検索に使用する LDAP アカウント。例: uid=ldap_search,ou=system (OpenLDAP の場合), CN=ldap_search,OU=user_group,DC=company,DC=com (Microsoft Active Directory の場合) 匿名バインドもサポートされています。ただし、匿名バインドを許可すると、ドメイン設定が未認可ユーザーに公開される可能性があります（ユーザー、コンピューター、サーバー、グループ、サービスなどの情報）。セキュリティ上の理由から、LDAP ホストでは匿名バインドを無効にし、代わりに認証済みアクセスを使用してください。
Bind password	LDAP サーバーへのバインドおよび検索に使用するアカウントの LDAP パスワード。
Description	LDAP サーバーの説明。
Configure JIT provisioning	このチェックボックスをオンにすると、JIT プロビジョニングに関連するオプションが表示されます。
Group configuration	グループ設定方法を選択します: **memberOf** - ユーザーとそのグループメンバーシップ属性を検索する方法 **groupOfNames** - member 属性を通じてグループを検索する方法 memberOf の方が高速なため推奨されます。LDAP サーバーが `memberOf` をサポートしていない場合、またはグループフィルタリングが必要な場合は groupOfNames を使用してください。
Group name attribute	`memberOf` 属性内のすべてのオブジェクトからグループ名を取得する属性を指定します（*User group membership attribute* フィールドを参照） グループ名はユーザーグループマッピングに必要です。
User group membership attribute	ユーザーが所属するグループの情報を含む属性を指定します（例: `memberOf`）。 たとえば、memberOf 属性には次のような情報が含まれる場合があります: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` このフィールドは memberOf メソッドでのみ使用できます。
User name attribute	ユーザーの名を含む属性を指定します。
User last name attribute	ユーザーの姓を含む属性を指定します。
User group mapping	LDAP ユーザーグループパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングします。 これは、プロビジョニングされたユーザーに Zabbix でどのユーザーグループ/ロールが割り当てられるかを判断するために必要です。 マッピングを追加するには *Add* をクリックします。 *LDAP group pattern* フィールドはワイルドカードをサポートします。グループ名は既存のグループと一致している必要があります。 1 人の LDAP ユーザーが複数の Zabbix ユーザーグループに一致する場合、そのユーザーはそれらすべてのメンバーになります。 1 人のユーザーが複数の Zabbix ユーザーロールに一致する場合、その中で最も高い権限レベルのロールが割り当てられます。
Media type mapping	通知送信のために、ユーザーの LDAP media 属性（例: email）を Zabbix ユーザーメディアにマッピングします（属性値はメディアの*Send to* フィールドとして使用されます）。
Advanced configuration	*Advanced configuration* ラベルをクリックすると、高度な設定オプションが表示されます（以下を参照）。
StartTLS	LDAP サーバーへの接続時に StartTLS 操作を使用するには、このチェックボックスをオンにします。 サーバーが StartTLS をサポートしていない場合、接続は失敗します。 StartTLS は *ldaps* プロトコルを使用するサーバーでは使用できません。

Parameter	Description
Search filter	LDAP でユーザーを認証する際に使用するカスタム文字列を定義します。次のプレースホルダーがサポートされています: `%{attr}` - 検索属性名 (uid, sAMAccountName) `%{user}` - 認証するユーザーのユーザー名の値 たとえば、大文字小文字を区別しない LDAP または Microsoft Active Directory 環境で大文字小文字を区別する検索を実行するには、文字列を次のように定義できます: `(%{attr}:caseExactMatch:=%{user})`。 フィルターをカスタマイズしない場合、LDAP はデフォルトの `(%{attr}=%{user})` を使用します。

Note:
直接ユーザーバインド用に LDAP サーバーを設定するには、*Base DN* パラメータに属性 uid=%{user} を追加し（例: *uid=%{user},dc=example,dc=com*）、*BindDN* および *Bind password* パラメータは空のままにします。認証時には、プレースホルダー%{user} はログイン時に入力されたユーザー名に置き換えられます。
 直接ユーザーバインドでは、Zabbix がユーザーの認証情報にアクセスできるのは対話型ログイン中のみです。そのため、対話型ログインなしで動作するプロビジョニングタスク（たとえば、*Provision now* ボタンや`user.provision` API メソッド）は、直接ユーザーバインドを使用して認証するユーザーを無視します。これは、Zabbix がそれらのユーザーに代わってバインドできないためです。エンドユーザーがログインしていなくても Web インターフェースからプロビジョニングおよび LDAP 検索を機能させる必要がある場合は、匿名バインドまたは専用のバインドユーザー（サービスアカウント）を使用してください。

以下のフィールドは、*Group configuration* メソッドとして”groupOfNames” を選択した場合に固有のものです:



Parameter	Description
Group base DN	LDAP サーバー内のグループへのベースパス。
Group name attribute	指定したグループのベースパス内でグループ名を取得する属性を指定します。 グループ名はユーザーグループマッピングに必要です。
Group member attribute	LDAP 内でグループのメンバー情報を含む属性を指定します（例: `member`）。
Reference attribute	グループフィルター用の参照属性を指定します（*Group filter* フィールドを参照）。 その後、グループフィルター内で `%{ref}` を使用して、ここで指定した属性の値を取得します。
Group filter	ユーザーが所属するグループを取得するためのフィルターを指定します。 たとえば、`(member=uid=%{ref},ou=Users,dc=example,dc=com)` は、グループの member 属性が `uid=User1,ou=Users,dc=example,dc=com` である場合に”User1” に一致し、”User1” が所属するグループを返します。

Warning:
証明書に問題がある場合、セキュアな LDAP 接続 (ldaps) を機能させるために、/etc/openldap/ldap.conf 設定ファイルに `TLS_REQCERT allow` 行を追加する必要があることがあります。これにより、LDAP カタログへの接続のセキュリティが低下する可能性があります。

Note:
実際のユーザーアカウント（Zabbix Web インターフェースへのログインに使用されるもの）を使用する代わりに、LDAP 内で最小限の権限を持つ、LDAP サーバーへのバインドおよび検索専用の LDAP アカウント（*Bind DN*）を別途作成することを推奨します。
この方法はより高いセキュリティを提供し、ユーザーが LDAP サーバーで自分のパスワードを変更した場合でも *Bind password* を変更する必要がありません。
上の表では、これは *ldap_search* アカウント名です。

アクセスのテスト

テストボタンを使用すると、ユーザーアクセスをテストできます。

Parameter	Description
Login	テストする LDAP ユーザー名（Zabbix Web インターフェースの現在のユーザー名があらかじめ入力されています）。このユーザー名は LDAP サーバーに存在している必要があります。 テストユーザーを認証できない場合、Zabbix は LDAP 認証を有効化しません。
User password	テストする LDAP ユーザーパスワード。

3 SAML

概要

SAML 2.0 認証 を使用して、Zabbix にサインインできます。

SAML サインインのみが設定されている場合、ユーザーは Zabbix にも存在している必要がありますが、その Zabbix パスワードは使用されません。認証が成功すると、Zabbix はローカルのユーザー名を、SAML によって返されたユーザー名属性と照合します。

ユーザープロビジョニング

SAML ユーザーに対して、JIT（just-in-time）ユーザープロビジョニングを設定できます。
この場合、ユーザーがあらかじめ Zabbix に存在している必要はありません。
ユーザーアカウントは、ユーザーが初めて Zabbix にログインしたときに作成できます。

Attention:
JIT プロビジョニングを有効にする場合は、*Authentication* タブでプロビジョニング解除されたユーザー用のユーザーグループを指定する必要があります。

JIT プロビジョニングに加えて、SCIM（System for Cross-domain Identity Management）プロビジョニングを有効にして設定することもできます。これは、ユーザープロビジョニングによって作成されたユーザーに対する継続的なユーザーアカウント管理です。
SCIM プロビジョニングでは、Zabbix への認証に Zabbix API token（Super admin 権限付き）が必要です。

たとえば、ユーザーがある SAML グループから別の SAML グループに移動された場合、Zabbix でもそのユーザーはあるグループから別のグループへ移動されます。
ユーザーが SAML グループから削除された場合、そのユーザーは Zabbix 内の該当グループからも削除され、他のどのグループにも属していなければ、プロビジョニング解除されたユーザー用のユーザーグループに追加されます。

SCIM が有効化され設定されている場合、SAML ユーザーは Zabbix にログインした時点でプロビジョニングされ、その後は SAML での変更に基づいて継続的に更新されます。
すでに存在する SAML ユーザーはプロビジョニングされず、更新されるのはプロビジョニング済みのユーザーのみです。
また、ユーザーがプロビジョニングまたは更新される際に、ユーザーには有効なメディアのみが追加される点に注意してください。

SCIM が有効でない場合、SAML ユーザーは Zabbix にログインした時点でプロビジョニングされます（その後の更新も同様です）。

Note:
SAML 認証が有効な場合、ユーザーはローカルログインと SAML シングルサインオンのどちらかを選択してログインできます。
JIT プロビジョニングを使用する場合は、シングルサインオンのみ可能です。

ID プロバイダーの設定

Zabbix で動作させるには、SAML ID プロバイダー（onelogin.com, auth0.com, okta.com など）を以下のように設定する必要があります。

- *Assertion Consumer URL* は <path_to_zabbix_ui>/index_sso.php?acs に設定してください
- *Single Logout URL* は <path_to_zabbix_ui>/index_sso.php?sls に設定してください

<path_to_zabbix_ui> の例: https://example.com/zabbix/ui, http://another.example.com/zabbix, http://<any_public_

Zabbix の設定

Attention:
Web インターフェースで SAML 認証を使用する場合は、php-openssl をインストールする必要があります。

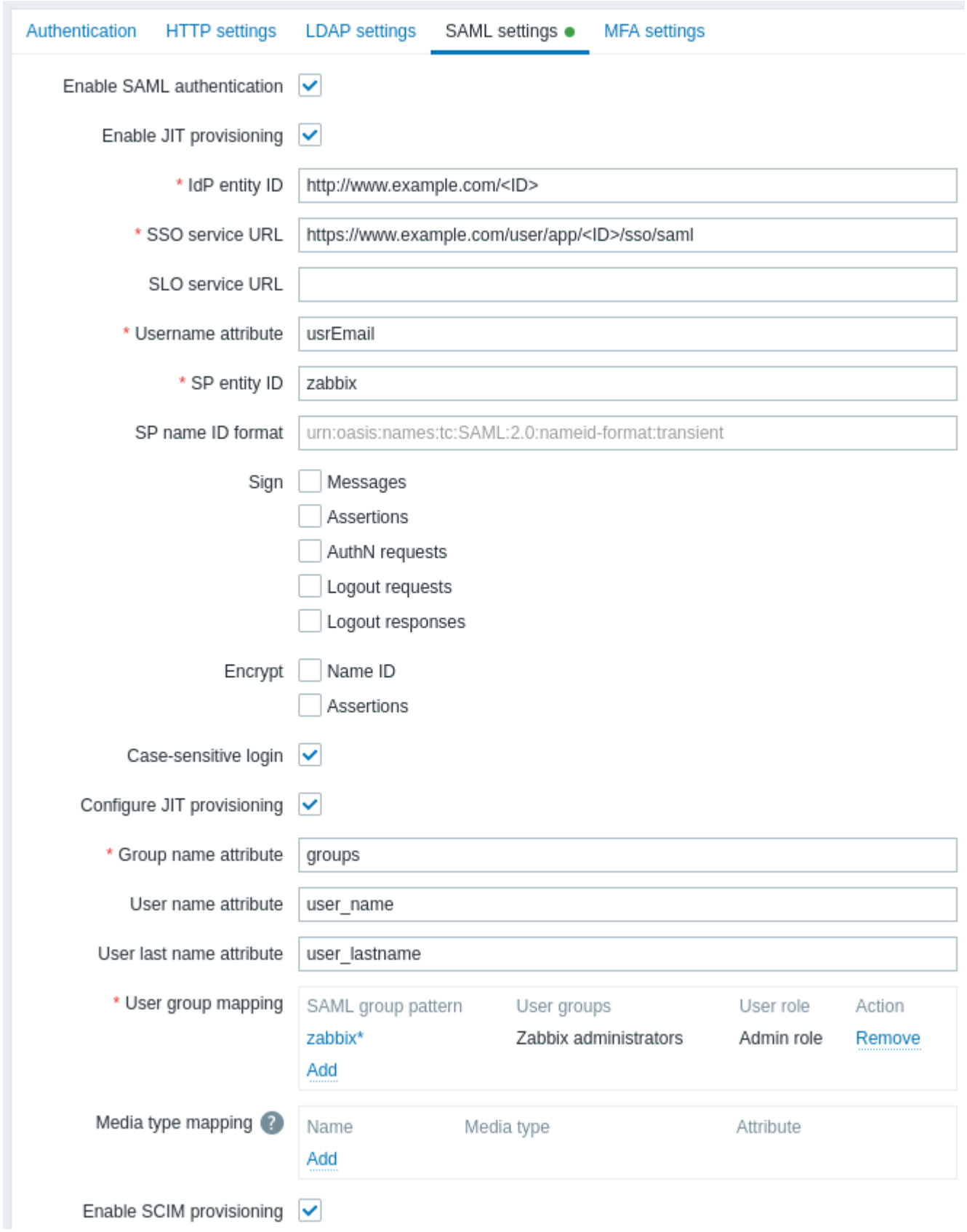
SAML 認証を使用するには、Zabbix を次のように設定する必要があります。

1. 秘密鍵と証明書は、カスタムパスがzabbix.conf.phpで指定されていない限り、*ui/conf/certs/* に保存する必要があります。

デフォルトでは、Zabbix は次の場所を参照します。

- ui/conf/certs/sp.key - SP 秘密鍵ファイル
- ui/conf/certs/sp.crt - SP 証明書ファイル
- ui/conf/certs/idp.crt - IDP 証明書ファイル

2. 最も重要な設定はすべて Zabbix の Web インターフェースで設定できます。ただし、追加の設定を設定ファイルで指定することも可能です。



Zabbix の Web インターフェースで利用可能な設定パラメータ:

Parameter	Description
Enable SAML authentication	チェックボックスをオンにして、SAML 認証を有効にします。
Enable JIT provisioning	チェックボックスをオンにして、JIT ユーザープロビジョニングを有効にします。
IDP entity ID	SAML ID プロバイダー内の一意のエンティティ識別子です。
SSO service URL	ログイン時にユーザーがリダイレクトされる URL です。

Parameter	Description
SLO service URL	ログアウト時にユーザーがリダイレクトされる URL です。空のままにすると、SLO サービスは使用されません。
Username attribute	Zabbix へのログイン時にユーザー名として使用される SAML 属性です。 サポートされる値の一覧は、ID プロバイダーによって決まります。 例: uid userprincipalname samaccountname username userusername urn:oid:0.9.2342.19200300.100.1.1 urn:oid:1.3.6.1.4.1.5923.1.1.1.13 urn:oid:0.9.2342.19200300.100.1.44
SP entity ID	一意のサービスプロバイダー識別子です（一致しない場合、操作は拒否されます）。 URL または任意の文字列データを指定できます。
SP name ID format	レスポンスで特定の name ID 形式を要求します。 例: urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified urn:oasis:names:tc:SAML:2.0:nameid-format:transient
Sign	チェックボックスをオンにして、SAML 署名を有効にする対象エンティティを選択します: *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	チェックボックスをオンにして、SAML 暗号化を有効にする対象エンティティを選択します: *Name ID* *Assertions*
Case-sensitive login	チェックボックスをオフにすると、ユーザー名の大文字・小文字を区別するログインを無効にします（デフォルトで有効）。 大文字・小文字を区別するログインを無効にすると、たとえば Zabbix ユーザーが「Admin」または「ADMIN」であっても、「admin」としてログインできます。 大文字・小文字を区別するログインが無効で、類似したユーザー名を持つ複数の Zabbix ユーザー（例: Admin と admin）が存在する場合、それらのユーザーのログインは常に次のエラーメッセージで拒否されることに注意してください: ”Authentication failed: supplied credentials are not unique.”
Configure JIT provisioning	このチェックボックスをオンにすると、JIT ユーザープロビジョニングに関連するオプションが表示されます。
Group name attribute	JIT ユーザープロビジョニング用のグループ名属性を指定します。
User name attribute	JIT ユーザープロビジョニング用のユーザー名属性を指定します。
User last name attribute	JIT ユーザープロビジョニング用のユーザー姓属性を指定します。
User group mapping	SAML ユーザーグループパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングします。 これは、プロビジョニングされたユーザーに Zabbix でどのユーザーグループ/ロールが付与されるかを判断するために必要です。 *Add* をクリックしてマッピングを追加します。 *SAML group pattern* フィールドはワイルドカードをサポートしています。グループ名は既存のグループと一致している必要があります。 SAML ユーザーが複数の Zabbix ユーザーグループに一致する場合、そのユーザーはそれらすべてのメンバーになります。 ユーザーが複数の Zabbix ユーザーロールに一致する場合、その中で最も高い権限レベルが付与されます。
Media type mapping	通知送信のために、ユーザーの SAML メディア属性（例: email）を Zabbix ユーザーメディアにマッピングします。
Enable SCIM provisioning	このチェックボックスをオンにして、SCIM 2.0 プロビジョニングを有効にします。

次の例では、サインインおよびユーザープロビジョニングのための SAML ID プロバイダーを Zabbix に設定する方法を確認できます。

- Microsoft Azure AD
- Okta
- Onelogin

SCIM プロビジョニングに関する注意事項

SCIM プロビジョニングでは、ID プロバイダー側で Zabbix Web インターフェースへのパスを指定し、その末尾に api_scim.php を追加します。例:

```
https://<path-to-zabbix-ui>/api_scim.php
```

Zabbix で使用されるユーザー属性 (username、user name、user lastname、およびメディア属性) は、カスタム属性として追加する必要があります。また、必要に応じて、外部ネームスペースはユーザースキーマ `urn:ietf:params:scim:schemas:core:2.0:User` と同じである必要があります。

詳細設定

追加の SAML パラメータは、Zabbix の Web インターフェース設定ファイル（*zabbix.conf.php*）で設定できます。

- $SSO['SP_KEY'] = '<SP 秘密鍵ファイルへのパス >';
- $SSO['SP_CERT'] = '<SP 証明書ファイルへのパス >';
- $SSO['IDP_CERT'] = '<IDP 証明書ファイルへのパス >';
- $SSO['SETTINGS']

Note:
`$SSO['SETTINGS']` 配列は、*SAML PHP Toolkit* ライブラリ（Zabbix に付属）で想定されているものと同じ構造に従う必要があります。利用可能な設定オプションの完全な説明については、公式のライブラリドキュメントを参照してください。

$SSO['SETTINGS'] の一部として設定できるのは、次のオプションのみです。

- *strict*
- *baseurl*
- *compress*
- *contactPerson*
- *organization*
- *sp*（この一覧で指定されているオプションのみ）
  - *attributeConsumingService*
  - *x509certNew*
- *idp*（この一覧で指定されているオプションのみ）
  - *singleLogoutService*（1 つのオプションのみ）
    - *responseUrl*
  - *certFingerprint*
  - *certFingerprintAlgorithm*
  - *x509certMulti*
- *security*（この一覧で指定されているオプションのみ）
  - *signMetadata*
  - *wantNameId*
  - *requestedAuthnContext*
  - *requestedAuthnContextComparison*
  - *wantXMLValidation*
  - *relaxDestinationValidation*
  - *destinationStrictlyMatches*
  - *rejectUnsolicitedResponsesWithInResponseTo*
  - *signatureAlgorithm*
  - *digestAlgorithm*
  - *lowercaseUrlencoding*

その他のすべてのオプションはデータベースから取得され、上書きすることはできません。*debug* オプションは無視されます。

さらに、Zabbix UI がプロキシまたはロードバランサーの背後にある場合は、カスタムの *use_proxy_headers* オプションを使用できます。

- *false*（デフォルト）- このオプションを無視します。
- *true* - ベース URL の構築に X-Forwarded-* HTTP ヘッダーを使用します。

ロードバランサーを使用して Zabbix インスタンスに接続しており、ロードバランサーでは TLS/SSL を使用し、Zabbix では使用していない場合は、'baseurl'、'strict'、'use_proxy_headers' パラメータを次のように指定する必要があります。

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

設定例:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Kerberos/ADFS を使用した Web インターフェースの設定

Zabbix の Web インターフェース設定ファイル（*zabbix.conf.php*）を使用して、Kerberos 認証および ADFS による SSO を設定できます。

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

この場合、SAML 設定の *SP name ID* フィールドには次を設定します。

`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

4 MFA

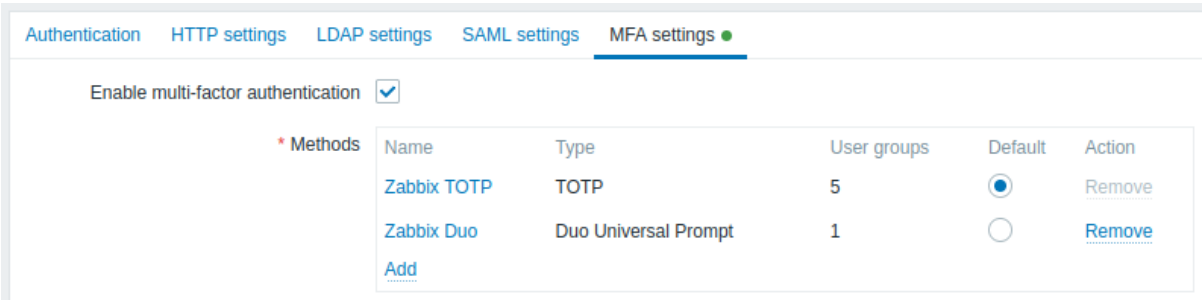
概要

多要素認証（MFA）を使用して Zabbix にサインインでき、ユーザー名とパスワードだけの場合に比べて、追加のセキュリティ層を提供します。

MFA では、ユーザーが Zabbix に存在している必要があり、ログイン時に Zabbix の認証情報を入力し、さらに別の手段で本人確認を行う必要があります。通常は、ユーザーのスマートフォン上の認証アプリで生成されたコードを使用します。

複数の MFA 方式が利用可能であり、ユーザーは自身のセキュリティ要件や好みに最も適した方法を選択できます。
これらの方式には、Time-Based One-Time Password（TOTP）と Duo Universal Prompt があります。

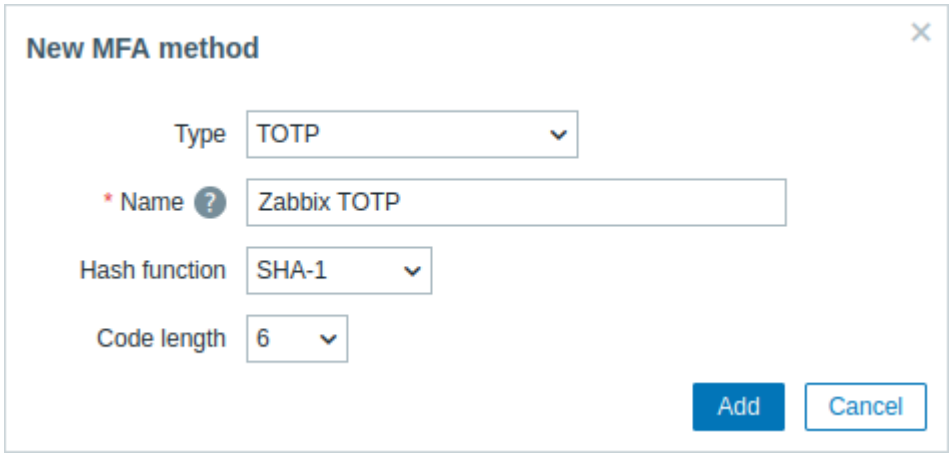
設定



設定パラメータ:

Parameter	Description
Enable multi-factor authentication	チェックボックスをオンにして、多要素認証を有効にします。
Methods	*Add* をクリックして MFA 方式を設定します（以下の方式の設定を参照）。

メソッドの設定



メソッド設定パラメータ:

Parameter	Description
Type	MFA メソッドのタイプを選択します: **TOTP** - 認証アプリを使用して時間ベースのワンタイムパスワードを生成します。 **Duo Universal Prompt** - Duo 認証サービスを使用して多要素認証を提供します。
Name	認証アプリ内ですべての MFA ユーザーにアカウント名として表示される名前を入力します（例: "Zabbix"）。
Hash function	TOTP コードの生成に使用するハッシュ関数（SHA-1、SHA-256、または SHA-512）を選択します。 このパラメータは、MFA メソッドのタイプが"TOTP" に設定されている場合に使用できます。 現在、多くのアプリケーションはこれらの関数をサポートしていないため、SHA-256 または SHA-512 を選択すると互換性が大幅に制限される可能性があることに注意してください。
Code length	認証コードの長さ（6 または 8）を選択します。 このパラメータは、MFA メソッドのタイプが"TOTP" に設定されている場合に使用できます。
API hostname	Duo 認証サービスから提供された API ホスト名を入力します。 このパラメータは、MFA メソッドのタイプが"Duo Universal Prompt" に設定されている場合に使用できます。
Client ID	Duo 認証サービスから提供されたクライアント ID を入力します。 このパラメータは、MFA メソッドのタイプが"Duo Universal Prompt" に設定されている場合に使用できます。
Client secret	Duo 認証サービスから提供されたクライアントシークレットを入力します。 このパラメータは、MFA メソッドのタイプが"Duo Universal Prompt" に設定されている場合に使用できます。

設定例

このセクションでは、Time-Based One-Time Password（TOTP）およびDuo Universal Prompt を使用して MFA を設定する例を紹介します。

TOTP

TOTP では、ユーザーは認証アプリ（たとえば、Google Authenticator アプリ）を使用して本人確認を行う必要があります。

1. Zabbix の *Users* → *Authentication* で MFA 設定を開き、多要素認証を有効にします。

2. 次の設定で新しい MFA method を追加します。

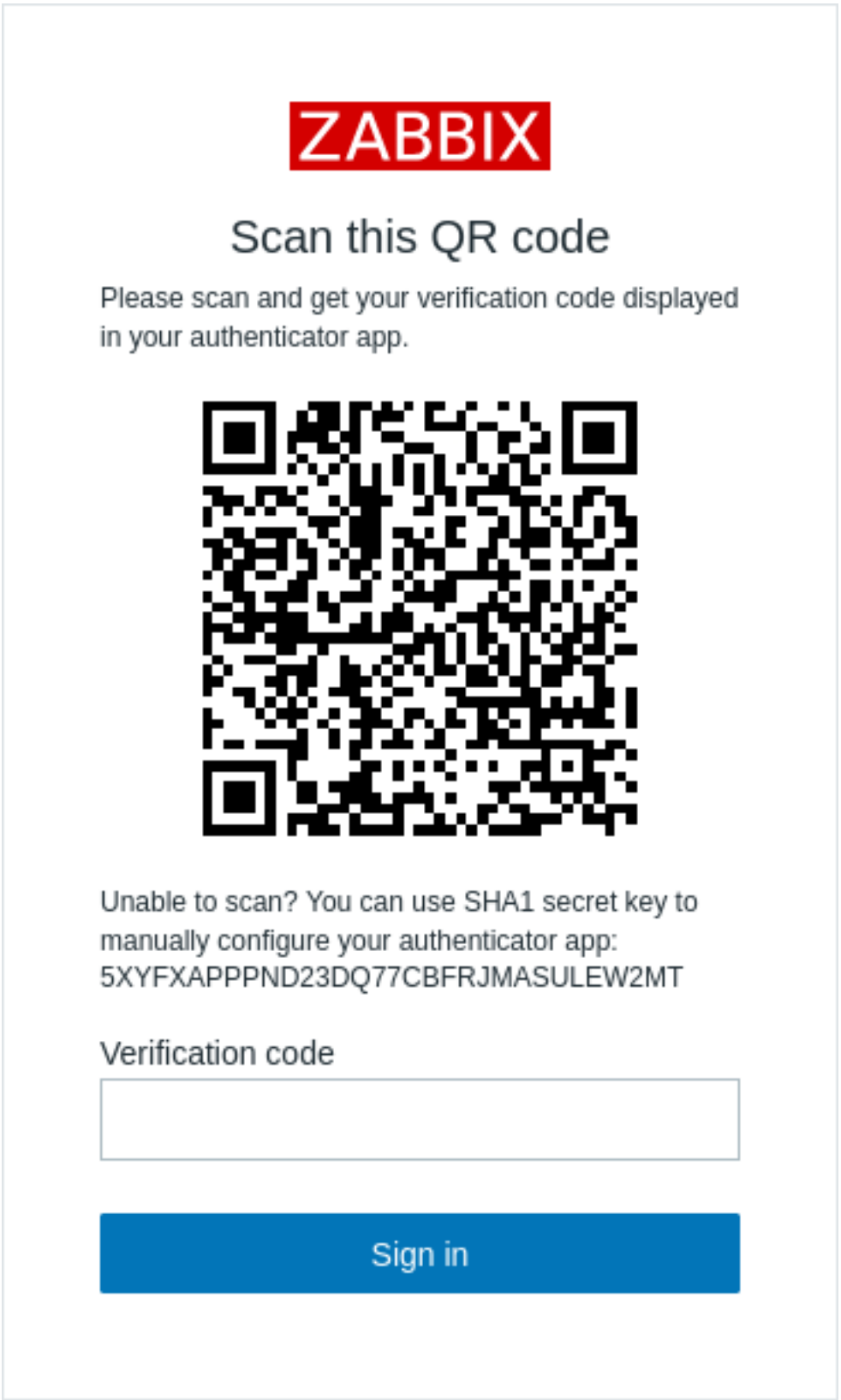
- Type: TOTP
- Name: Zabbix TOTP
- Hash function: SHA-1
- Code length: 6

3. *Add* ボタンをクリックし、続いて *Update* ボタンをクリックします。

4. *Users* → *User groups* に移動し、次の設定で新しいユーザーグループを作成します。

- Group name: TOTP group
- Users: Admin
- Multi-factor authentication: Default（または、デフォルトとして設定されていない場合は「Zabbix TOTP」）

5. Zabbix からログアウトし、認証情報を使用して再度ログインします。
ログインに成功すると、MFA への登録を求められ、QR コードとシークレットキーが表示されます。



6. QR コードをスキャンするか、シークレットキーを Google Authenticator アプリに入力します。
アプリが確認コードを生成するので、それを入力してログイン処理を完了します。

7. 以降のログインでは、Google Authenticator アプリから確認コードを取得し、ログイン時に入力します。

Duo Universal Prompt

Duo Universal Prompt では、ユーザーは認証アプリ Duo Mobile を使用して本人確認を行う必要があります。

Attention:
Duo Universal Prompt MFA 方式を使用するには、php-curl 拡張機能のインストール、HTTPS 経由での Zabbix へのアクセス、および Duo サーバーへの外向き接続の許可が必要です。
さらに、Web サーバーで Content Security Policy (CSP) を有効にしている 場合は、仮想ホストの設定ファイル内の CSP ディレクティブに”duo.com” を追加してください。

1. Duo Signup で無料の Duo 管理者アカウントに登録します。

2. Duo Admin Panel を開き、*Applications* → *Protect an Application* に移動し、*Web SDK* アプリケーションを検索して、*Protect* をクリックします。

3. Zabbix で MFA 方式を設定するために必要な認証情報（Client ID、Client secret、API hostname）を控えます。

4. Zabbix の *Users* → *Authentication* にある MFA 設定に移動し、多要素認証を有効にします。

5. 次の設定で新しい MFA 方式 を追加します。

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: （Duo の API hostname を使用）
- Client ID: （Duo の Client ID を使用）
- Client secret: （Duo の Client secret を使用）

6. *Add* ボタンをクリックし、その後 *Update* ボタンをクリックします。

7. *Users* → *User groups* に移動し、次の設定で新しいユーザーグループを作成します。

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default（または、デフォルトとして設定されていない場合は”Zabbix Duo”）

8. Zabbix からログアウトし、認証情報を使用して再度ログインします。
ログインに成功すると、MFA への登録を求められ、Duo にリダイレクトされます。
Duo の設定を完了し、ログインするためにスマートフォンの Duo アプリでユーザー認証を行います。

9. 以降のログインでは、Duo アプリが提供する適切な MFA 方式（確認コードの取得、プッシュ通知への応答、またはハードキーの使用など）を使用し、ログイン時に必要な情報を入力します。

9 管理

概要

管理メニューは、Zabbix の管理機能のためのものです。このメニューは、SuperAdmin ユーザータイプのユーザーのみが利用できます。

1 一般設定

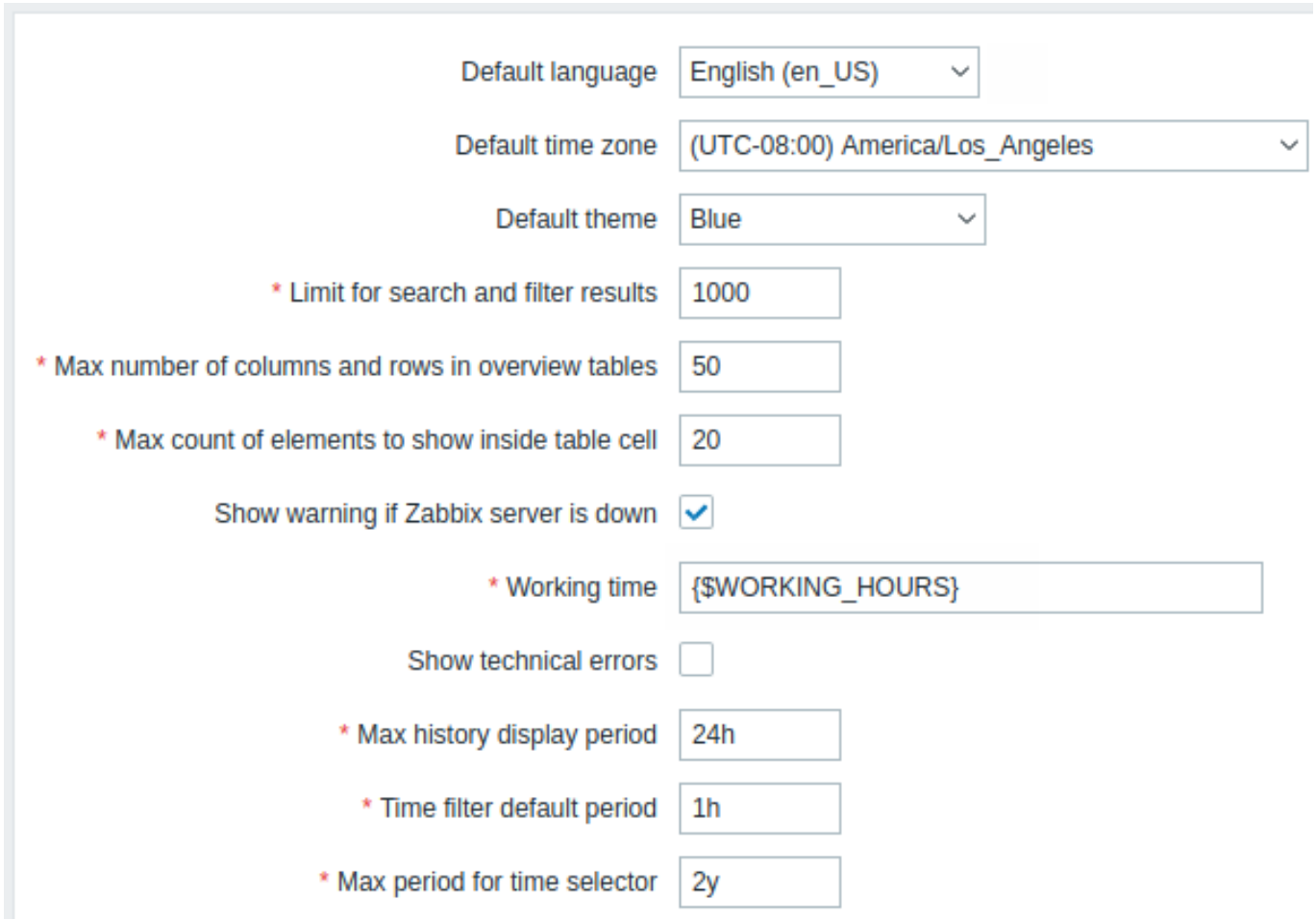
概要

Administration → *General* セクションには、Web インターフェース関連のデフォルト設定や Zabbix のカスタマイズを行うための複数のサブセクションが含まれています。

利用可能なサブセクションの一覧は、*Administration* メニューセクションで *General* を押すと表示されます。
また、左上隅にあるタイトルのドロップダウンを使用して、サブセクションを切り替えることもできます。

GUI

このセクションでは、Web インターフェースに関連するいくつかのデフォルト設定をカスタマイズできます。



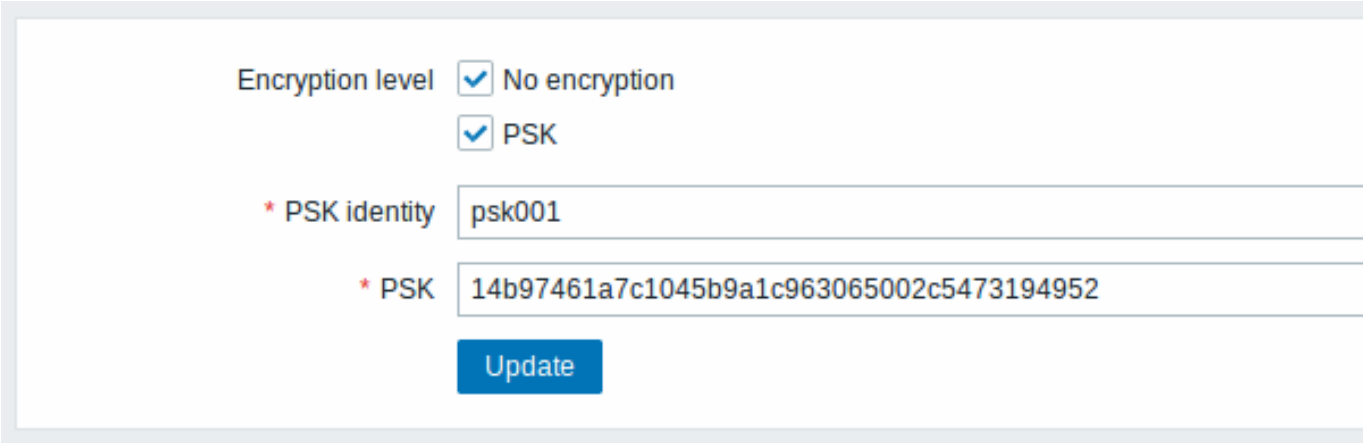
設定パラメータ:

パラメータ	説明
デフォルトの言語	プロファイルで言語を指定していないユーザーおよびゲストユーザーに対するデフォルトの言語です。 詳細は、追加の Web インターフェース言語のインストールを参照してください。
デフォルトのタイムゾーン	プロファイルでタイムゾーンを指定していないユーザーおよびゲストユーザーに対するデフォルトのタイムゾーンです。
デフォルトのテーマ	プロファイルでテーマを指定していないユーザーおよびゲストユーザーに対するデフォルトのテーマです。
検索およびフィルター結果の上限	Web インターフェースのリストに表示される要素（行）の最大数です。たとえば、データ収集 > ホストなどです。 注: たとえば「50」に設定すると、影響を受けるすべての Web インターフェースのリストで最初の 50 要素のみが表示されます。あるリストに 50 を超える要素が含まれている場合、そのことは *"Displaying 1 to 50 of **50+** found"* の「+」記号で示されます。また、フィルタリングを使用していて一致件数が 50 を超える場合でも、最初の 50 件のみが表示されます。 このパラメータの値を増やすと、Web サーバー側でパフォーマンスの低下やメモリ消費の増加を招く可能性があることに注意してください。
概要テーブルの <*br*> 最大列数および行数	トリガーの概要ダッシュボードウィジェットに表示する列数および行数の最大値です。同じ上限が列と行の両方に適用されます。表示されている数を超える行や列が存在する場合、システムはテーブルの下部に次の警告を表示します: 「Not all results are displayed. Please provide more specific search criteria.」
テーブルセル内に表示する <*br*> 要素数の最大値	1 つのテーブルセル内に表示されるエントリについて、ここで設定した数を超えては表示されません。
Zabbix サーバー停止時に警告を表示	このパラメータを有効にすると、Zabbix サーバーに到達できない場合（停止している可能性がある場合）に、ブラウザウィンドウに警告メッセージが表示されます。メッセージは、ユーザーがページを下にスクロールしても表示されたままです。メッセージにマウスカーソルを合わせると一時的に非表示になり、その下の内容を確認できます。
稼働時間	このシステム全体のパラメータは、稼働時間を定義します。グラフでは、稼働時間は白い背景で、非稼働時間は灰色で表示されます。 時間形式の説明については、期間指定ページを参照してください。 ユーザーマクロをサポートしています。

パラメータ	説明
技術的エラーを表示	チェックすると、登録済みのすべてのユーザーが技術的エラー（PHP/SQL）を表示できるようになります。チェックしない場合、この情報はZabbix Super Admins およびデバッグモード が有効なユーザーグループに属するユーザーのみが利用できます。
履歴表示期間の最大値	監視 > 最新データにおける履歴データ、監視 > ホストにおけるホストのWeb シナリオ情報、および *Top items* ダッシュボードウィジェットに表示する最大期間です。 許容範囲: 24 時間（デフォルト）~1 週間。時間サフィックス（例: 1w（1 週間）、36h（36 時間））をサポートしています。
タイムフィルターのデフォルト期間	グラフおよびダッシュボードでデフォルトで使用される期間です。許容範囲: 1 分～10 年（デフォルト: 1 時間）。 時間サフィックス（例: 10m（10 分）、5w（5 週間））をサポートしています。 注: ユーザーがグラフの表示中に期間を変更すると、その期間はユーザー設定として保存され、グローバルデフォルトまたは以前のユーザー選択を置き換えます。
タイムセレクターの最大期間	グラフおよびダッシュボードで利用可能な最大期間です。ユーザーはそれより古いデータを表示できません。許容範囲: 1 年～10 年（デフォルト: 2 年）。 時間サフィックス（例: 1y（1 年）、365w（365 週間））をサポートしています。

自動登録

このセクションでは、アクティブエージェントの自動登録に対する暗号化レベルを設定できます。



アスタリスクが付いたパラメータは必須です。

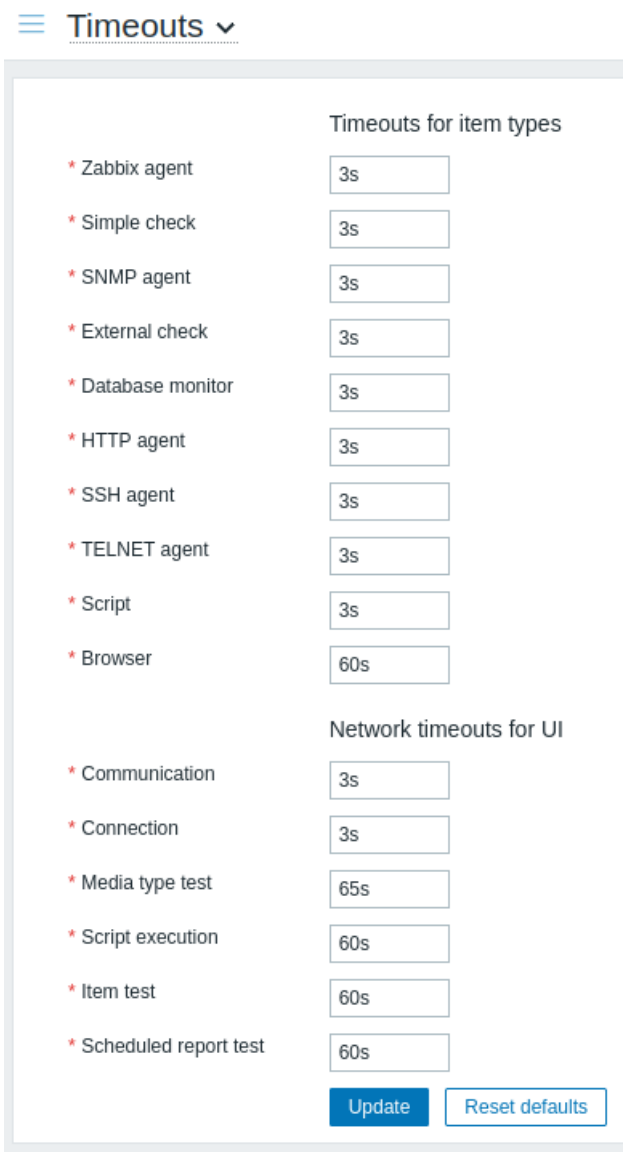
設定パラメータ:

Parameter	Description
Encryption level	暗号化レベルとして、次のオプションのいずれか 1 つまたは両方を選択します: **No encryption** - 暗号化されていない接続を許可します **PSK** - 事前共有鍵を使用した TLS 暗号化接続を許可します
PSK identity	事前共有鍵の識別文字列を入力します。 このフィールドは、*Encryption level* として「PSK」が選択されている場合にのみ使用できます。 PSK identity には機密情報を含めないでください。受信側にどの PSK を使用するかを通知するため、ネットワーク上を暗号化されずに送信されます。
PSK	事前共有鍵（16 進文字を偶数個並べたもの）を入力します。 最大長: Zabbix が GnuTLS または OpenSSL ライブラリを使用している場合は 512 桁の 16 進数（256 バイト PSK）、Zabbix が mbed TLS（PolarSSL）ライブラリを使用している場合は 64 桁の 16 進数（32 バイト PSK）。 例: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 このフィールドは、*Encryption level* として「PSK」が選択されている場合にのみ使用できます。

参照: 安全な自動登録

タイムアウト

このセクションでは、グローバルなアイテムタイプ別タイムアウトおよびネットワークタイムアウトを設定できます。
このフォーム内のすべてのフィールドは必須です。

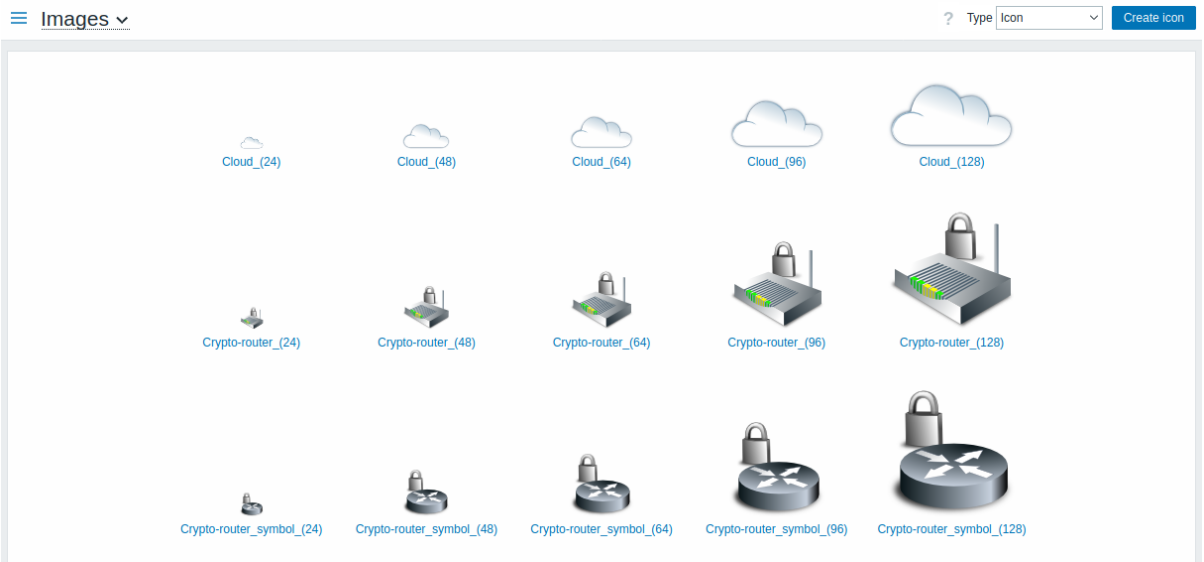


パラメータ	説明
アイテムタイプのタイムアウト	監視対象のアイテムからの応答を待機する秒数（タイプに基づく）です。 許容範囲: 1 - 600s（デフォルト: 3s、Browser アイテムタイプのデフォルト: 60s）。 時間サフィックス（例: 30s、1m）およびユーザーマクロ をサポートしています。 サポートされるアイテムタイプ: - Zabbix エージェント（パッシブチェックとアクティブチェックの両方） - Simple check（`icmpping*`、`vmware.*` アイテムを除く） - SNMP エージェント（SNMP `walk[OID]` および `get[OID]` アイテムのみ） - External check - Database monitor - HTTP エージェント - SSH エージェント - TELNET エージェント - Script - Browser プロキシが使用され、かつタイムアウトが設定されている場合、プロキシのタイムアウト設定がグローバル設定より優先されることに注意してください。特定のアイテムにタイムアウトが設定されている場合は、それらがプロキシおよびグローバル設定より優先されます。
UI のネットワークタイムアウト	

パラメータ		説明
	通信	アイドル状態のソケットを閉じるまで待機する秒数です（以前に Zabbix サーバーへの接続が確立されていても、Web インターフェースがこの時間内にデータの読み取り/送信処理を完了できない場合、接続は切断されます）。許容範囲: 1 - 300s（デフォルト: 3s）。
	接続	Zabbix サーバーへの接続試行を中止するまで待機する秒数です。許容範囲: 1 - 30s（デフォルト: 3s）。
	メディアタイプテスト	メディアタイプのテスト時に応答を待機する秒数です。許容範囲: 1 - 300s（デフォルト: 65s）。
	スクリプト実行	スクリプト実行時に応答を待機する秒数です。許容範囲: 1 - 300s（デフォルト: 60s）。 このタイムアウトは、長さがさまざまなスクリプト実行チェーン全体に対するものです。たとえば、スクリプトがエージェント上で実行される場合、サーバー（場合によってはプロキシも含む）からエージェントへ送られ、戻ってくる往復全体が対象となります。 エージェント上で実行されるスクリプトには、server またはproxy のタイムアウトが適用されます。 アクティブなエージェントでのみ実行されるスクリプトについては、デフォルトのサーバー/プロキシタイムアウトを引き上げる必要がある可能性があります。サーバー/プロキシタイムアウトはアクティブチェックの更新頻度より高くする必要があります。そうしないと、アクティブなエージェントがスクリプトを受信する前にサーバー/プロキシタイムアウトを超過する可能性があります。 関連項目: スクリプトタイムアウト
	アイテムテスト	アイテムのテスト時に返されるデータを待機する秒数です。許容範囲: 1 - 600s（デフォルト: 60s）。
	定期レポートテスト	定期レポートのテスト時に返されるデータを待機する秒数です。許容範囲: 1 - 300s（デフォルト: 60s）。

画像

画像セクションには、Zabbix で利用可能なすべての画像が表示されます。画像はデータベースに保存されます。

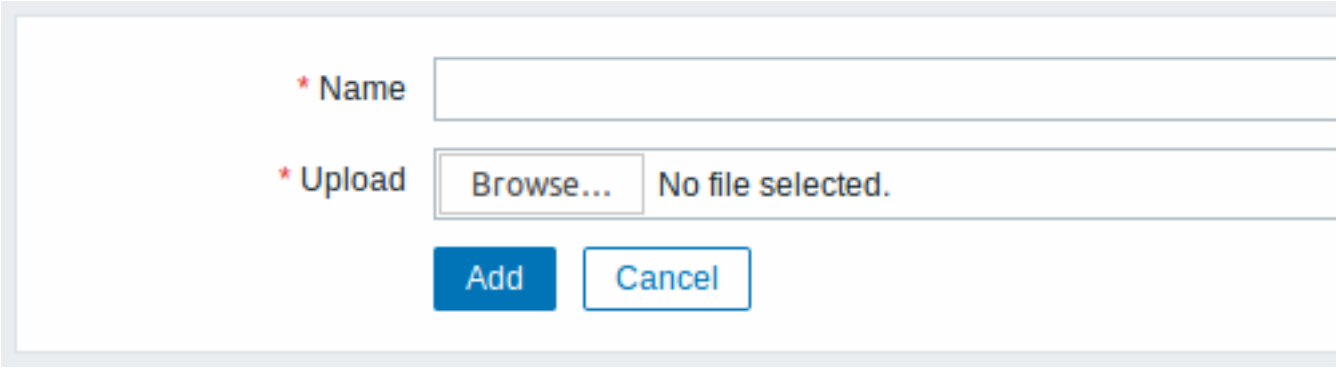


Type ドロップダウンでは、アイコン画像と背景画像を切り替えることができます。

- アイコンは、ネットワークマップの要素を表示するために使用されます
- 背景画像は、ネットワークマップの背景画像として使用されます

画像の追加

右上隅にある *Create icon* または *Create background* ボタンをクリックすると、独自の画像を追加できます。



画像の属性:

Parameter	Description
Name	画像の一意な名前。
Upload	ローカルシステムから Zabbix にアップロードするファイル（PNG、JPEG、GIF、または WebP）を選択します。 アップロード時に PNG へ変換される他の形式もアップロードできる場合があります。画像処理には GD ライブラリが使用されるため、サポートされる形式は使用しているライブラリのバージョンに依存します（Zabbix では 2.0.28 以上が必要です）。

Note:
アップロードファイルの最大サイズは、ZBX_MAX_IMAGE_SIZE の値によって制限されており、1024x1024 バイト、または 1 MB です。

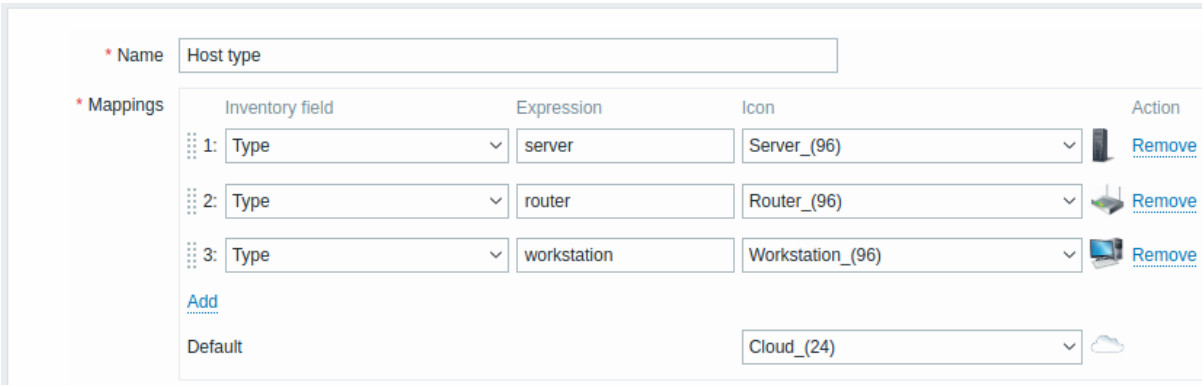
画像サイズが 1 MB に近く、MySQL 設定パラメータ `max_allowed_packet` がデフォルトの 1MB に設定されている場合、画像のアップロードに失敗することがあります。この場合は、max_allowed_packet パラメータを増やしてください。

アイコンのマッピング

このセクションでは、特定のホストを特定のアイコンにマッピングできます。マッピングの作成には、ホストインベントリフィールドの情報が使用されます。

作成したマッピングは、ネットワークマップの設定で使用して、一致するホストに適切なアイコンを自動的に割り当てることができます。

新しいアイコンマップを作成するには、右上隅にある アイコンマップの作成をクリックします。



設定パラメータ:

Parameter	Description
Name	アイコンマップの一意の名前。
Mappings	マッピングの一覧。マッピングの順序によって、どれが優先されるかが決まります。ドラッグ＆ドロップでマッピングを一覧内で上下に移動できます。
Inventory field	一致を検索するために参照されるホストインベントリフィールド。
Expression	一致条件を記述する正規表現。
Icon	式に一致した場合に使用するアイコン。
Default	使用するデフォルトのアイコン。

正規表現

このセクションでは、Web インターフェース内の複数の場所で使用できるカスタム正規表現を作成できます。詳細については、正規表現 セクションを参照してください。

トリガー表示オプション

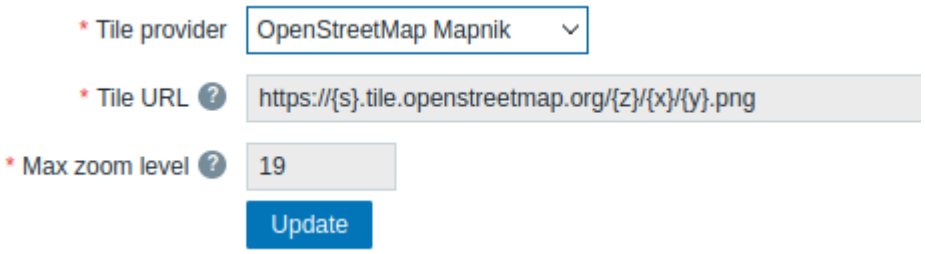
このセクションでは、Web インターフェースでのトリガーのステータス表示方法、およびトリガーの深刻度の名前と色をカスタマイズできます。



パラメータ	説明
Use custom event status colors	このパラメータをチェックすると、確認済み/未確認の障害に対する色のカスタマイズが有効になります。
Unacknowledged PROBLEM events, *Acknowledged PROBLEM events*, *Unacknowledged RESOLVED events*, *Acknowledged RESOLVED events*	新しいカラーコードを入力するか、色をクリックして表示されるパレットから新しい色を選択します。 *blinking* チェックボックスをオンにすると、ステータス変更時にトリガーが一定時間点滅し、より目立つようになります。
Display OK triggers for	OK 状態のトリガーを表示する期間です。許容範囲: 0～24 時間。時間サフィックス（例: 5m、2h、1d）をサポートしています。
On status change triggers blink for	トリガーが点滅する時間の長さです。許容範囲: 0～24 時間。時間サフィックス（例: 5m、2h、1d）をサポートしています。
Not classified, *Information*, *Warning*, *Average*, *High*, *Disaster*	システムのデフォルトの代わりに表示する、カスタムの深刻度名および/または色です。 新しいカラーコードを入力するか、色をクリックして表示されるパレットから新しい色を選択します。 ここで入力したカスタム深刻度名は、すべてのロケールで使用されることに注意してください。特定のユーザー向けに他の言語へ翻訳する必要がある場合は、トリガーの深刻度のカスタマイズページを参照してください。

地理マップ

このセクションでは、地理マップタイルサービスプロバイダーの選択と、Geomap ダッシュボードウィジェット のサービスプロバイダー設定を行えます。地理マップを使用した可視化を提供するために、Zabbix はオープンソースの JavaScript 対話型地図ライブラリ Leaflet を使用しています。定義済みのタイルプロバイダーを含むサードパーティのタイルプロバイダーが提供する画像の品質について、Zabbix は管理できない点にご注意ください。



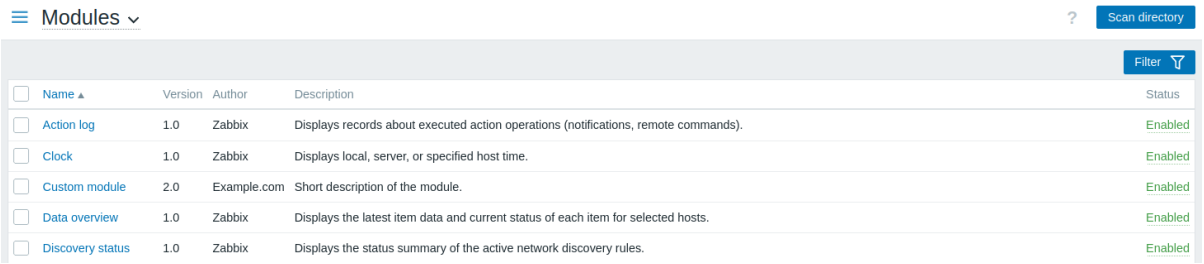
Parameter	Description
タイルプロバイダー	利用可能なタイルサービスプロバイダーのいずれかを選択するか、*Other* を選択して別のタイルプロバイダーまたはセルフホストのタイルを追加します（カスタムタイルサービスプロバイダーの使用 を参照）。
タイル URL	地理マップ上でタイルレイヤーを読み込み、表示するための URL テンプレート（最大 2048 文字）。このフィールドは、*Tile provider* が *Other* に設定されている場合のみ編集できます。 次のプレースホルダーがサポートされています。 *{s}* は利用可能なサブドメインのいずれかを表します。 *{z}* は URL 内のズームレベルパラメータを表します。 *{x}* と *{y}* はタイル座標を表します。 *{r}* は、retina タイルを読み込むために URL に”@2x” を追加する際に使用できます。 例: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
帰属表示テキスト	マップ上の小さなテキストボックスに表示される、タイルプロバイダーの帰属表示テキスト。このフィールドは、*Tile provider* が *Other* に設定されている場合のみ表示されます。
最大ズームレベル	マップの最大ズームレベル。このフィールドは、*Tile provider* が *Other* に設定されている場合のみ編集できます。

カスタムタイルサービスプロバイダーの使用

Geomap ウィジェットは、カスタムのセルフホストまたはサードパーティのタイルプロバイダーサービスからラスタータイル画像を読み込むことができます。カスタムサードパーティタイルプロバイダーサービスまたはセルフホストタイルフォルダーまたはサーバーを使用するにはタイルプロバイダーフィールドでその他を選択し、適切なプレースホルダーを使用してタイル *URL* フィールドでカスタム URL を指定します。

モジュール

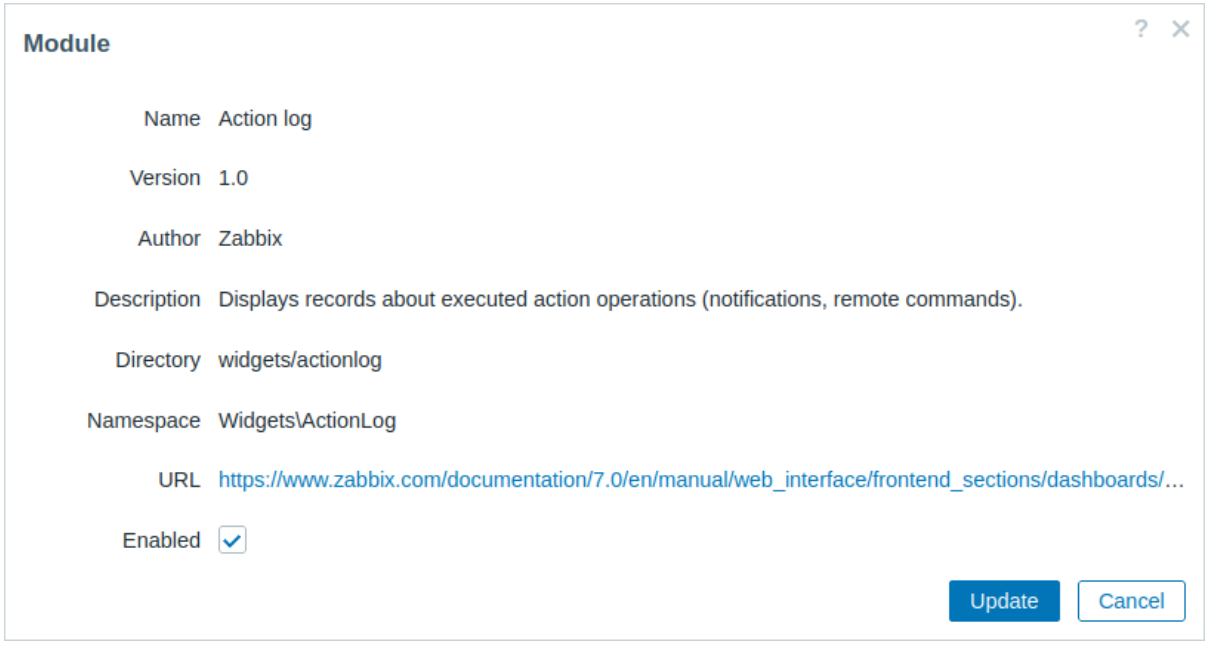
このセクションでは、カスタムのWeb インターフェースモジュールおよび組み込みモジュールを管理できます。



カスタムモジュールを登録/登録解除するには、ディレクトリをスキャンをクリックします。登録されたモジュールは一覧に表示され、登録解除されたモジュールは一覧から削除されます。

一覧内のモジュールステータスをクリックすると、モジュールを有効/無効にできます。また、一覧でモジュールを選択し、一覧の下にある有効化/無効化ボタンをクリックすることで、モジュールを一括で有効/無効にすることもできます。

一覧内のモジュール名をクリックすると、ポップアップウィンドウでその詳細を表示できます。

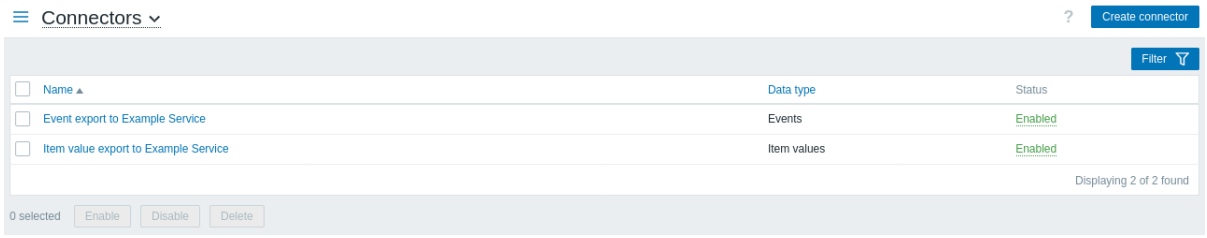


モジュールのステータスは、モジュール詳細のポップアップウィンドウでも更新できます。これを行うには、*Enabled* チェックボックスをオン/オフにしてから、更新をクリックします。

モジュールは、名前またはステータス（有効/無効）でフィルタリングできます。

コネクタ

このセクションでは、HTTP 経由で Zabbix データを外部システムへストリーミングするためのコネクタを設定できます。



新しいコネクタを設定するには、コネクタの作成をクリックします。

コネクタは名前またはステータス（有効/無効）で絞り込めます。リスト内のコネクタのステータスをクリックすると、そのコネクタを有効/無効にできます。
また、リストでコネクタを選択してから、リストの下にある 有効化/無効化ボタンをクリックすることで、コネクタを一括で有効/無効にすることもできます。

その他

このセクションでは、その他の各種 Web インターフェースパラメータを設定できます。



パラメータ	説明
Web インターフェース *URL*	Zabbix Web インターフェースの URL（最大 2048 文字）。 このパラメータは、Zabbix ウェブサービスが Web インターフェースと通信するために使用され、定期レポートを有効にするには指定が必要です。
ディスカバリされたホスト用のホストグループ	ネットワークディスカバリ およびエージェント自動登録 によってディスカバリされたホストは、ここで選択したホストグループに自動的に配置されます。
デフォルトのホストインベントリモード	ホストインベントリのデフォルトモードです。新しいホストまたはホストプロトタイプがサーバーまたは Web インターフェースによって作成される際、ホストのディスカバリ/自動登録時に ホストインベントリモードの設定操作で上書きされない限り、この設定が適用されます。

パラメータ	説明
データベース停止メッセージ用のユーザーグループ	アラームメッセージ送信用のユーザーグループ、または「なし」です。 Zabbix サーバーはバックエンドデータベースの可用性に依存しています。データベースなしでは動作できません。データベースが停止した場合、選択したユーザーに Zabbix から通知できます。通知は、ここで設定したユーザーグループに対して、有効なユーザーメディア設定を使用して送信されます。通知は次のメディアタイプを使用して送信されます（有効な場合）: メール、SMS、カスタムアラートスクリプト。webhook メディア設定が構成され有効になっていても、通知はトリガーされません。 Zabbix サーバーは停止せず、データベースが復旧するまで待機して処理を継続します。 通知は次の内容で構成されます。 `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> は空の値として定義されている場合はメッセージに追加されず、<DB Port> はデフォルト値（"0"）の場合は追加されません。アラートマネージャー（Zabbix サーバーの特別なプロセス）は、10 秒ごとにデータベースへの新しい接続の確立を試みます。データベースが引き続き停止している場合、アラートマネージャーはアラート送信を繰り返しますが、その頻度は 15 分に 1 回を超えません。
一致しない *SNMP* トラップをログに記録	対応する SNMP インターフェースが見つからない場合に、SNMP トラップをログに記録します。

認証

Parameter	Description
ログイン試行回数	ログインの可能性がブロックされるまでの、失敗したログイン試行回数。
ログインブロック間隔	ログイン試行回数の上限を超えた場合に、ログインが禁止される期間。許可される範囲: 30～3600 秒。時間のサフィックス（例: 90s、5m、1h）をサポートしています。

シークレットの保存

Parameter	Description
Vault provider	ユーザーマクロの値を保存するためのシークレット管理ソフトウェアを選択します - *HashiCorp Vault*（デフォルト）または *CyberArk Vault*。
Resolve secret vault macros by	シークレット Vault マクロの解決方法を選択します: **Zabbix server** - シークレットは Zabbix サーバーによって取得され、必要に応じてプロキシに転送されます（デフォルト）。 **Zabbix server and proxy** - シークレットは Zabbix サーバーとプロキシの両方によって取得され、それぞれが独立してマクロを解決できます。

関連項目: シークレットの保存。

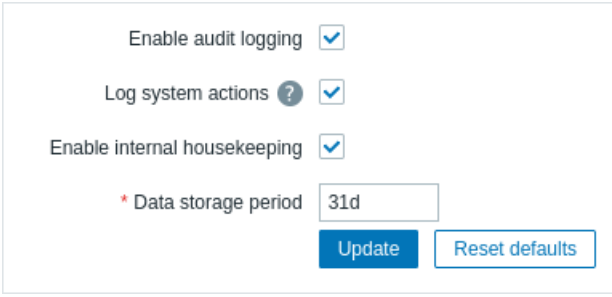
セキュリティ

Parameter	Description
URI スキームを検証	このチェックボックスをオフにすると、URI スキームの検証を無効にします（デフォルトで有効）。オンの場合、許可する URI スキームのカンマ区切りリストを指定できます（デフォルト: http,https,ftp,file,mailto,tel,ssh）。これは、URI が使用される Web インターフェース内のすべてのフィールド（例: マップ要素の URL）に適用されます。
X-Frame-Options HTTP ヘッダーを使用	このチェックボックスをオフにすると、HTTP X-Frame-options ヘッダーを無効にします（非推奨）。オンの場合、HTTP X-Frame-options ヘッダーの値を指定できます。サポートされる値: **SAMEORIGIN**（デフォルト）または **'self'**（シングルクォート必須）- ページ自体と同一オリジンのフレーム内でのみページを表示できます。 **DENY** または **'none'**（シングルクォート必須）- 表示を試みるサイトに関係なく、ページをフレーム内に表示できません。 スペース区切りのホスト名文字列; リストに **'self'**（シングルクォート必須）を追加すると、ページ自体と同一オリジンのフレーム内でページを表示できます。 **'self'** または **'none'** をシングルクォートなしで使用すると、ホスト名として扱われることに注意してください。

Parameter	Description
iframe サンドボックス化を使用	このチェックボックスをオフにすると、取得した URL コンテンツをサンドボックスに配置する機能を無効にします（非推奨）。 オンの場合、iframe サンドボックス化の例外を指定できます。指定しない制限は引き続き適用されます。このフィールドが空の場合、sandbox 属性のすべての制限が適用されます。 詳細については、`sandbox` 属性の説明を参照してください。

2 監査ログ

概要



このセクションでは、監査ログの設定を構成できます。以下のパラメータが利用可能です：

パラメータ	説明
監査ログの有効化	監査ログを有効化（デフォルト）または無効化します。
システム操作のログ記録	サーバー（システムユーザー）が実行する低レベル検出、ネットワーク検出、自動登録活動の監査ログ記録を有効化（デフォルト）または無効化します。
内部ハウスキーピングを有効化	監査ログレコードの内部ハウスキーピングを有効化（デフォルト）または無効化します。
データ保存期間	ハウスキーパーによって削除される前に監査ログレコードを保持する日数です。 ハウスキーピングが有効化されている場合、必須です。 デフォルト: 31 日間。

監査ログ圧縮パラメータは、管理 > ハウスキーピングセクション内の履歴、傾向、監査ログ圧縮ブロックで設定可能です。このブロックはTimescaleDB使用時に表示されます。

3 ハウスキーピング

概要

Administration > *Housekeeping* セクションでは、古くなったデータやユーザーによって削除されたデータをデータベースから自動的に削除する、Zabbix の内部ハウスキーピングを設定できます。これにより、データベースが無制限に肥大化するのを防ぎ、不要なディスク容量の消費やデータベース性能の低下を回避できます。

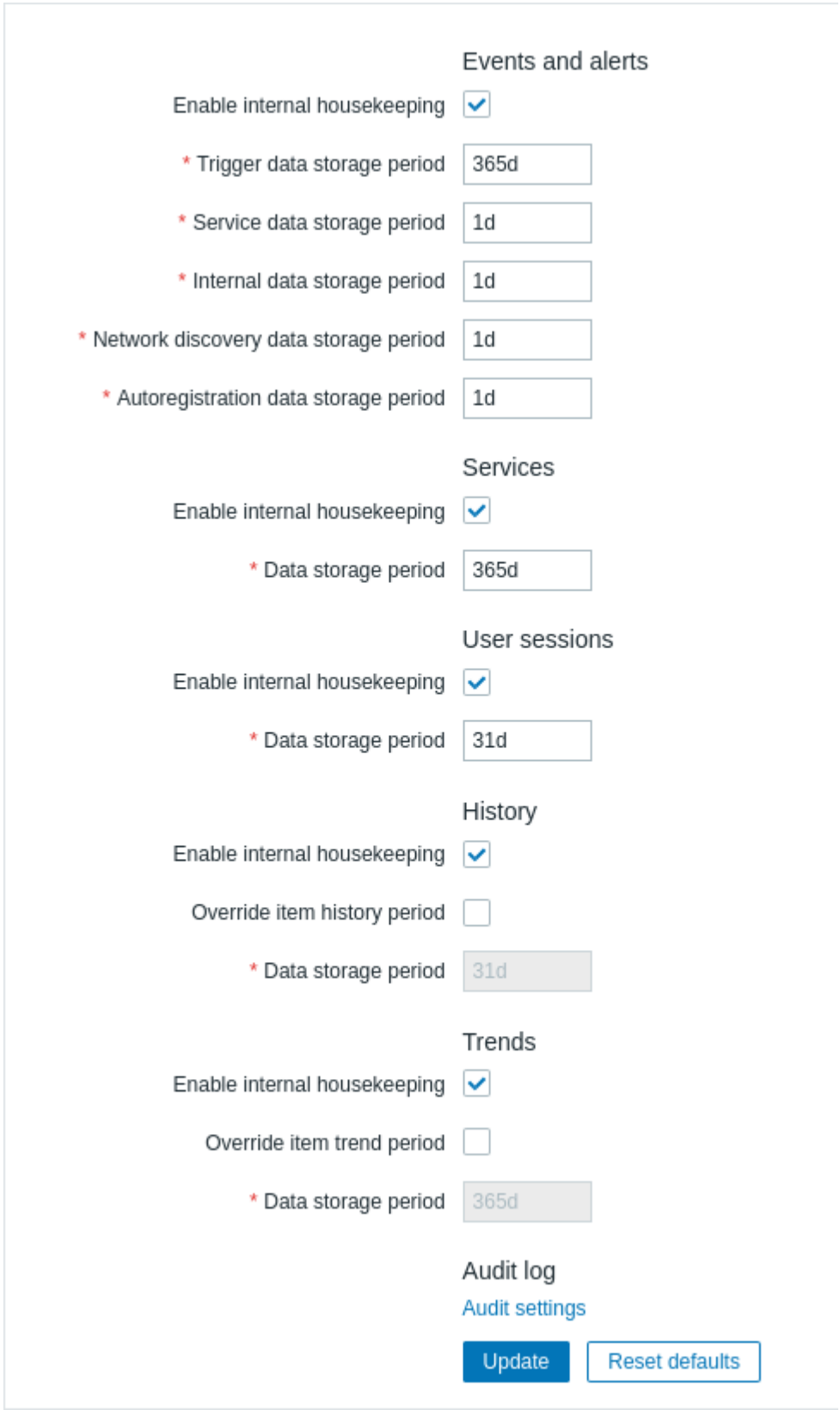
ハウスキーピングは、以下の種類のデータに対して有効化および設定できます。

- トリガー、サービス、内部データ、ネットワークディスカバリ、自動登録によるイベントとアラート
- サービス
- ユーザーセッション
- アイテムのヒストリとトレンド

Note:
監査ログのハウスキーピングは、別途設定します。

設定

以下のパラメータを使用できます。



Parameter	Description
内部ハウスキーピングを有効化	内部ハウスキーピングを有効または無効にします（デフォルトで有効）。 有効な場合、Zabbix サーバーのhousekeeper processが、データ保存期間を超えたデータをデータベースから自動的に削除します。

Parameter	Description
データ保存期間	housekeeper によって削除されるまで、データを保持する期間を指定します。 内部ハウスキーピングが有効な場合は必須です。 範囲: 1 日（履歴は 1 時間）~25 年、または”0”。時間のサフィックス（例: 1d、1w）をサポートしています。 イベントとアラートについては、データ保存期間はトリガー、サービス、内部データ、ネットワークディスカバリ、自動登録ごとに個別に設定されます。 housekeeper は、障害に関連付けられていないイベントのみを削除します。たとえば、データ保存期間より古い障害/復旧イベントであっても、未解決の障害に関連付けられている場合は削除されません。housekeeper が古いエンティティを削除する際は、まず障害を削除し、その後イベントを削除します。 アイテム/トリガーが削除されると、その障害も削除される点に注意してください。ただし、関連する障害/復旧イベントは、housekeeper によって削除されるまでデータベース内に残ります。 履歴およびトレンドについては、データ保存期間によって、内部ハウスキーピングが無効であっても、監視 > 最新データでデータが表示され続ける期間も決まります。
アイテムの履歴期間を上書き	有効にすると、アイテム設定で指定された履歴/トレンド保存期間は、データ保存期間の設定で上書きされます（保存しないオプションが有効なアイテムを除く）。 このオプションは、内部ハウスキーピングが無効で外部 housekeeper を使用している場合でも使用できます。
アイテムのトレンド期間を上書き	

TimescaleDB を使用している場合、履歴、トレンド、および監査ログの圧縮セクションが利用可能になります。

Note:
TimescaleDB では、自動パーティショニングの利点を最大限に活用するために、履歴およびトレンドに対して アイテムの履歴期間を上書き、アイテムのトレンド期間を上書き、および 内部ハウスキーピングを有効化を有効にしてください。これらのオプションが無効な場合でも、履歴テーブルおよびトレンドテーブル内のデータは引き続きパーティショニングされますが、housekeeper は古くなったパーティションを削除せず、設定に関する警告が表示されます。古くなったパーティションの削除を有効にすると、Zabbix サーバーおよび Web インターフェースは削除されたアイテムを追跡しなくなり、それらのアイテムの履歴は古いパーティションが削除された時点で消去されます。

デフォルトに戻すボタンを使用すると、加えた変更を元に戻すことができます。

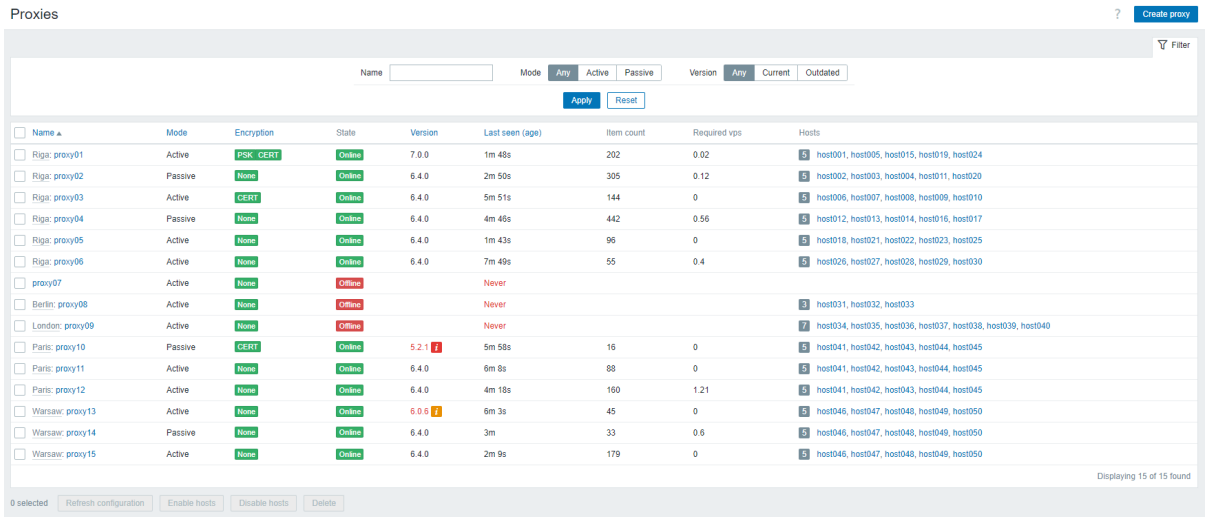
4 プロキシ

概要

分散監視 用のプロキシは、Zabbix Web インターフェースの *Administration* → *Proxies* セクションで設定できます。

プロキシ

既存のプロキシの一覧とその詳細が表示されます。



表示されるデータ:

Column	Description
Name	プロキシの名前。 プロキシ名をクリックすると、プロキシの設定フォームが開きます。 プロキシがプロキシグループに属している場合、グループ名がプロキシ名の前に灰色のリンクとして表示されます。グループ名をクリックすると、プロキシグループの設定フォームが開きます。
Mode	プロキシモード - アクティブまたは パッシブ。
Encryption	プロキシからの接続に対する暗号化の状態: なし - 暗号化なし; **PSK** - 事前共有鍵を使用; **Cert** - 証明書を使用。
State	プロキシの状態: **Unknown** - Zabbix サーバーの停止中にプロキシが作成された、またはサーバーがまだ状態を更新していない; **Online** - フェイルオーバー期間内にプロキシが Zabbix サーバーと通信した（パッシブプロキシはサーバー要求に応答し、アクティブプロキシは要求を送信した）; **Offline** - フェイルオーバー期間内にプロキシが Zabbix サーバーと通信していない。
Version	プロキシのバージョン（3 桁のバージョン番号）。プロキシが古い、またはサポート対象外の場合、バージョン番号が強調表示（赤色）され、情報ステータスアイコン（黄色または赤色）が表示されます。詳細はアイコンにマウスオーバーすると確認できます。
Last seen (age)	プロキシが最後にサーバーに認識された時刻。
Item count	プロキシに割り当てられた有効なホスト上の、有効なアイテム数。
Required vps	必要なプロキシ性能（1 秒あたりに収集する必要がある値の数）。
Hosts	プロキシに割り当てられた有効なホスト数と、プロキシによって監視されているホストの一覧。 ホスト名をクリックすると、ホスト設定フォームが開きます。

新しいプロキシを設定するには、右上隅の *Create proxy* ボタンをクリックします。

一括編集オプション

一覧の下にあるボタンでは、いくつかの一括編集オプションを利用できます。

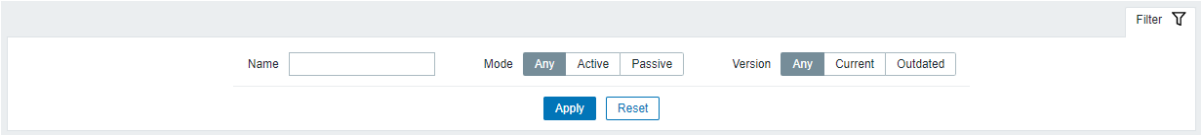
- 設定の更新 - プロキシの設定を更新します。
- ホストの有効化 - プロキシによって監視されているホストのステータスを 監視中に変更します。
- ホストの無効化 - プロキシによって監視されているホストのステータスを 監視しないに変更します。
- 削除 - プロキシを削除します。

これらのオプションを使用するには、対象のプロキシの前にあるチェックボックスを選択し、必要なボタンをクリックします。

フィルターの使用

フィルターを使用して、関心のあるプロキシのみを表示できます。検索パフォーマンスを向上させるため、データはマクロを未解決のまま検索されます。

プロキシの一覧の上にあるフィルターリンクを利用できます。これをクリックするとフィルターが表示され、名前、モード、バージョンでプロキシを絞り込めます。フィルターオプション *Outdated* では、古い（部分的にサポートされる）プロキシと未サポートのプロキシの両方が表示されることに注意してください。



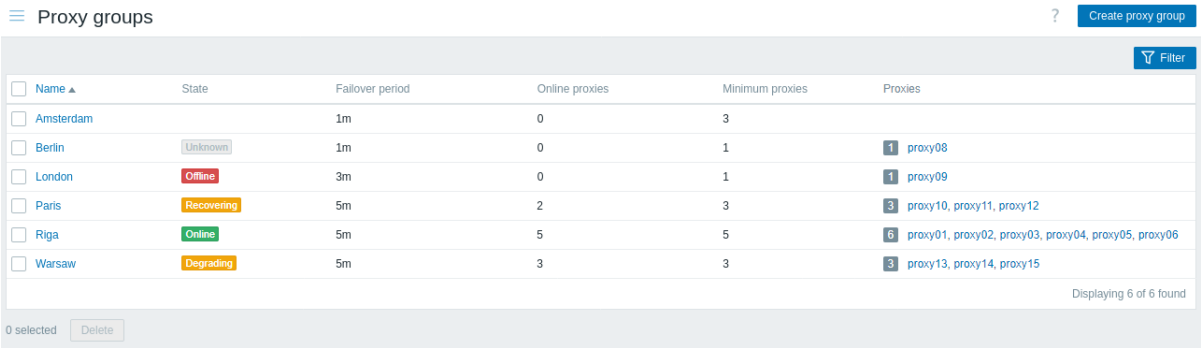
5 プロキシグループ

概要

管理 → プロキシグループでプロキシグループを設定できます。プロキシグループは、プロキシ間のホストの自動分散とプロキシ間の高可用性を実現するプロキシ負荷分散で使用されます。

プロキシグループ

既存のプロキシグループとその詳細の一覧が表示されます。



表示データ:

列名	説明
名前	プロキシグループ名。クリックするとプロキシグループの設定フォームが開きます。
状態	プロキシグループの状態（グループに少なくとも 1 つのプロキシが含まれる場合に表示）： 不明 - Zabbix サーバーがダウン中にグループが作成されたか、サーバーがまだ状態を更新していない状態; オンライン - フェイルオーバー期間内に最小限のプロキシ数が Zabbix サーバーと通信した状態; 劣化中 - オンラインプロキシが最小プロキシ数しきい値を下回ったため、グループがまもなくオフライン状態に移行します； オフライン - フェイルオーバー期間内に Zabbix サーバーと通信したプロキシが最小数未満でした； 回復中 - オンラインプロキシが最小プロキシ数しきい値に達したため、グループがまもなくオンライン状態に戻ります。 プロキシグループの状態は、zabbix[proxy group,<name>,state] およびzabbix[proxy group,discovery] アイテムで監視できます。
フェイルオーバー期間	プロキシグループ内のプロキシがオンラインと見なされるために Zabbix サーバーと通信しなければならない期間。
オンラインプロキシ最小プロキシ数	オンライン状態のプロキシの数（グループ最小値を下回ると赤色で表示）。 プロキシグループをオンライン状態に維持するために必要なオンラインプロキシの最小数。
プロキシ	グループ内のプロキシ数と、プロキシ設定フォームへのリンク付きのプロキシ一覧。 一覧表示されるプロキシの最大数は、テーブルセル内に表示する要素の最大数 の値によって制限されます。

新しいプロキシグループを設定するには、右上隅のプロキシグループを作成ボタンをクリックしてください。

一括編集オプション

リスト下部のボタンで一括編集が可能です：

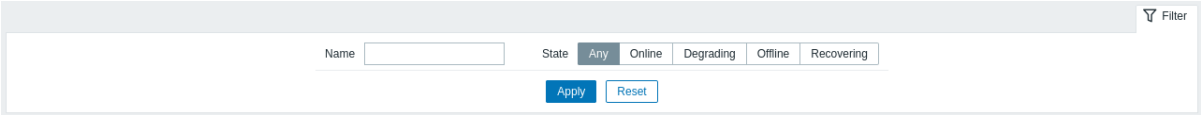
-削除 - プロキシグループを削除します。

これらのオプションを使用するには、対象のプロキシグループの前のチェックボックスを選択し、ボタンをクリックしてください。

フィルターの使用

フィルターを使用すると、関心のあるプロキシグループのみを表示できます。検索パフォーマンス向上のため、データはマクロが未解決の状態で検索されます。

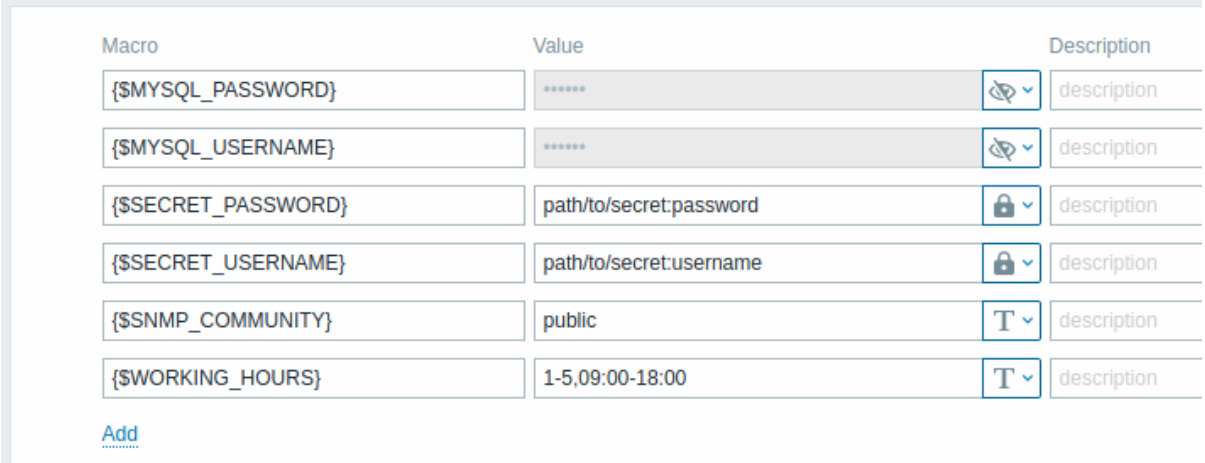
プロキシグループ一覧の上部に「フィルター」リンクが表示されます。クリックするとフィルター画面が開き、プロキシグループを名前とステータスで絞り込めます。



6 マクロ

概要

このセクションでは、システム全体のユーザーマクロを名前と値のペアとして定義できます。マクロの値はプレーンテキスト、シークレットテキスト、または Vault シークレットとして保持できることに注意してください。説明の追加もサポートされています。



7 キュー

概要

管理 → キューセクションでは、更新待ちのアイテムが表示されます。

理想的には、このセクションを開いたときにすべてが「緑」であるべきで、これはキュー内にアイテムが存在しないことを意味します。すべてのアイテムが遅延なく更新されていれば、待機中のアイテムはありません。ただし、サーバーの性能不足により、一部のアイテムで遅延が発生する場合があり、その情報がこのセクションに表示されます。詳細については、キュー セクションを参照してください。

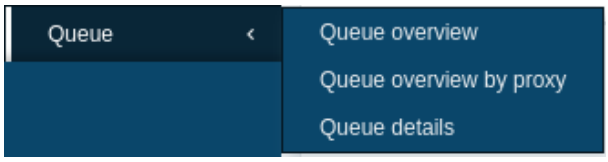
Note:
Zabbix プロキシはメンテナンス期間を認識しません。詳細はメンテナンス中のキューの計算 を参照してください。

Note:
キューは、Zabbix サーバーが稼働している場合にのみ利用できます。接続の問題やエージェントの正常な動作不良によりアイテムのインターフェースが利用不可になった場合、そのアイテムはキューにカウントされません。

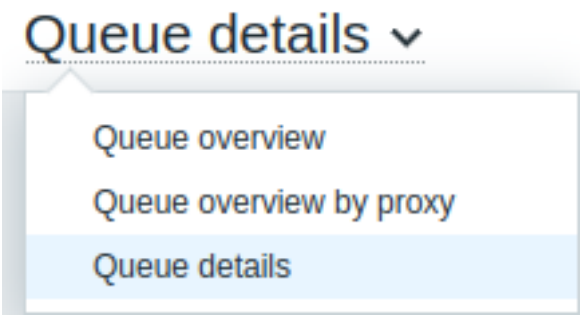
管理 → キューセクションには、次のページがあります。

- キューの概要 — アイテムタイプ別のキューを表示します。
- プロキシごとのキューの概要 — プロキシ別のキューを表示します。
- キューの詳細 — 遅延しているアイテムの一覧を表示します。

利用可能なページの一覧は、管理メニューセクションで キューを押すと表示されます。また、左上隅にあるタイトルのドロップダウンを使用してページを切り替えることもできます。



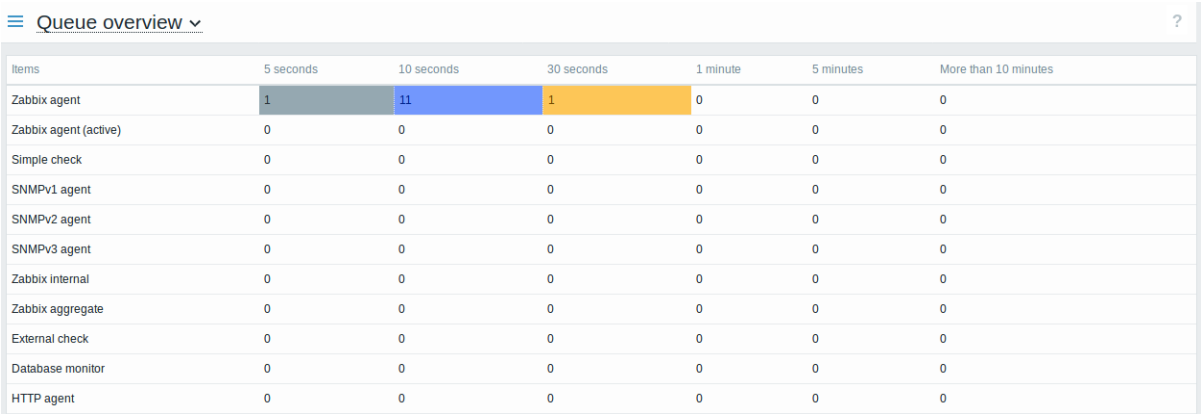
第 3 レベルメニュー。



タイトルのドロップダウン。

キュー概要

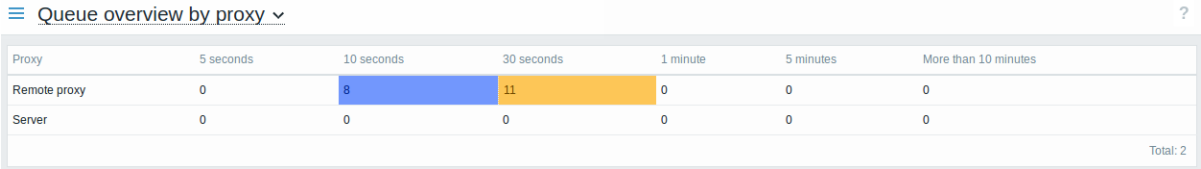
この画面では、問題が 1 つまたは複数のアイテムタイプに関連しているかどうかを簡単に特定できます。



各行にはアイテムタイプが含まれています。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5〜10 秒/10〜30 秒/30〜60 秒/1〜5 分/5〜10 分または 10 分以上です。

プロキシごとのキュー概要

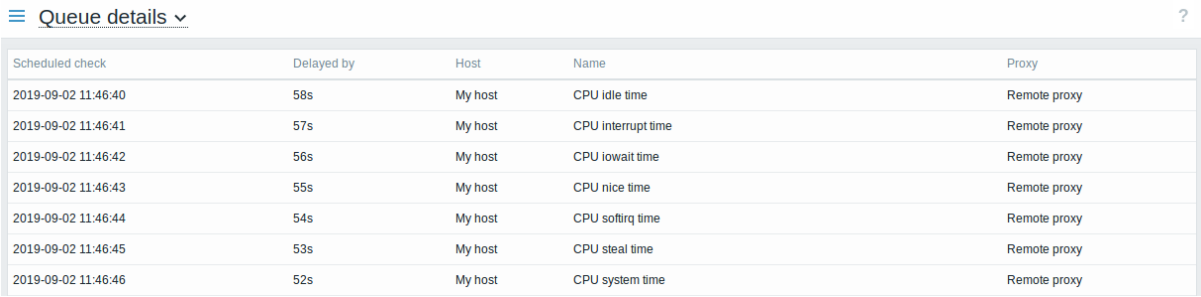
この画面では、問題がプロキシの 1 つに関連しているか、サーバーに関連しているかを簡単に特定できます。



各行にはプロキシが含まれ、サーバーはリストの最後にあります。各列には、待機中のアイテムの数が表示されます。待機中は、それぞれ 5〜10 秒/10〜30 秒/30〜60 秒/1〜5 分/5〜10 分または 10 分以上です。

待機中のアイテムの一覧

この画面では、各待機中のアイテムが一覧表示されます。



表示されるデータ:

列	説明
Scheduled check	チェックが予定されていた時刻が表示されます。

列	説明
Delayed by	遅延時間が表示されます。
Host	アイテムのホストが表示されます。
Name	待機中のアイテムの名前が表示されます。
Proxy	ホストがプロキシによって監視されている場合、プロキシ名が表示されます。

考えられるエラーメッセージ

データが表示されず、次のエラーメッセージが表示される場合があります。



この場合のエラーメッセージは次のとおりです。

```
Cannot display item queue. Permission denied
```

これは、*zabbix.conf.php* ファイル内の PHP 設定パラメータ $ZBX_SERVER、または $ZBX_SERVER と $ZBX_SERVER_PORT の両方が、異なるデータベースを使用する既存の Zabbix サーバーを指している場合に発生します。

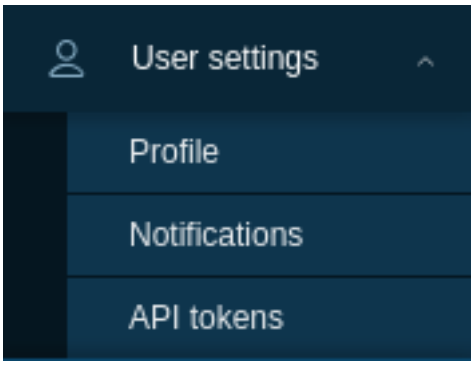
3 ユーザー設定

概要

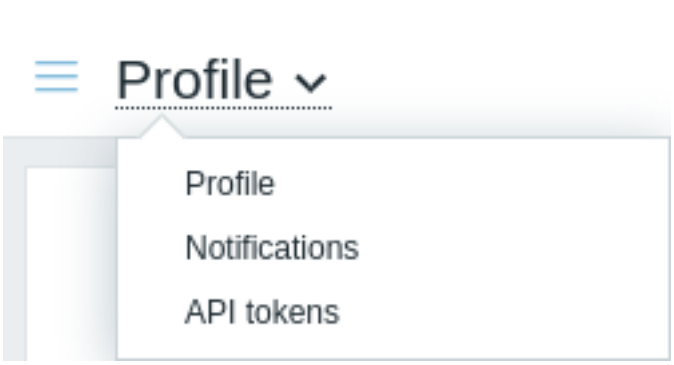
ユーザーのロール権限に応じて、ユーザー設定セクションには次のページが含まれる場合があります。

- プロフィールまたは ユーザープロフィール - 特定の Zabbix Web インターフェース機能をカスタマイズするためのページ。
- 通知 - 現在のユーザーの通知をカスタマイズするためのページ。
- *API* トークン - 現在のユーザーに割り当てられた API トークンを管理するためのページ。

利用可能なページの一覧は、Zabbix メニューの下部付近にある  ユーザーアイコンをクリックすると表示されます（*guest* ユーザーでは利用できません）。また、左上隅にあるタイトルのドロップダウンを使用してページを切り替えることもできます。



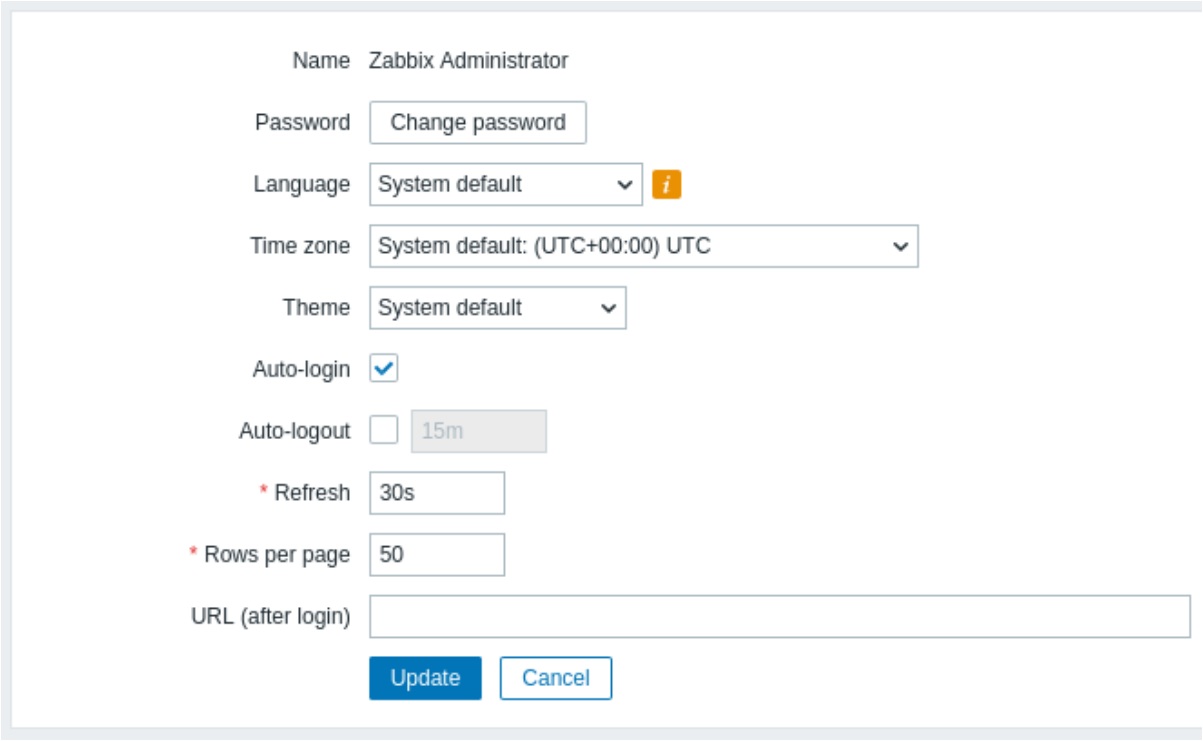
第 3 レベルメニュー。



タイトルのドロップダウン。

ユーザープロファイル

ユーザープロファイルセクションでは、カスタムのインターフェース言語、カラーテーマ、リストに表示する行数などを設定できます。ここで行った変更は、現在のユーザーにのみ適用されます。



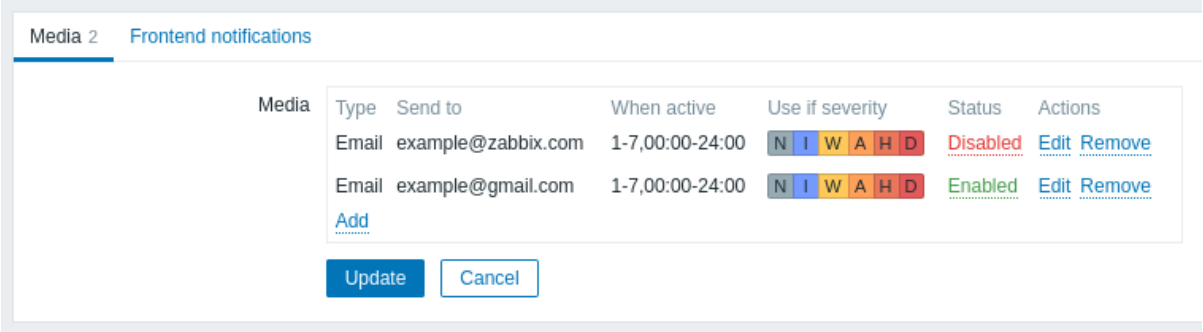
Parameter	Description
Name	ユーザーの名前と姓が表示されます。 名前と姓が設定されていない場合は、ユーザー名が表示されます。
Password	*Change password* ボタンをクリックすると、*Old password*、*New password*、*New password (once again)* の 3 つのフィールドが表示されます。 パスワードの変更に成功すると、ユーザーはすべてのアクティブなセッションからログアウトされます。 パスワードを変更できるのは、Zabbix の内部認証を使用しているユーザーのみであることに注意してください。
Language	使用するインターフェース言語を選択するか、**System default** を選択してシステムのデフォルト設定を使用します。 *English (en_US)* を選択すると、Web インターフェースで米国式の時刻/日付形式も有効になります。 詳細については、追加の Web インターフェース言語のインストールを参照してください。
Time zone	タイムゾーンを選択して、ユーザーレベルでグローバルのタイムゾーンを上書きするか、**System default** を選択してグローバルのタイムゾーン設定を使用します。
Theme	プロファイル専用のカラーテーマを選択します。 **System default** - システムのデフォルト設定を使用; **Blue** - 標準のブルーテーマ; **Dark** - 代替のダークテーマ; **High-contrast light** - 高コントラストのライトテーマ; **High-contrast dark** - 高コントラストのダークテーマ。
Auto-login	このチェックボックスをオンにすると、Zabbix がユーザーを記憶し、30 日間自動的にログインします。 *Remember for 30 days* を有効にしてログインした場合: - 自動ログアウトタイマーはリセットされます（30 日間の期間が終了するまでセッションは維持されます）。 - 30 日以内の以降のアクセス時に自動ログインが有効になります。 *Remember for 30 days* を使用せずにログインした場合: - Auto-login パラメータはクリアされます。 - 自動ログアウトは通常どおり設定されたタイムアウトに従います。 この機能にはブラウザの Cookie が使用されます。

Parameter	Description
Auto-logout	このチェックボックスをオンにすると、設定した秒数後に自動的にログアウトされます（最小 90 秒、最大 1 日）。 なお、*Remember for 30 days* を有効にしてログインした場合、自動ログアウトタイマーは上書きされ、セッションは全期間にわたってアクティブなままになります。 時間サフィックスがサポートされています。例: 90s、5m、2h、1d。 このオプションは、次の場合には機能しないことに注意してください。 * *Monitoring* メニューのページで、バックグラウンドで情報の更新が実行されている場合。一定間隔でデータを更新するページ（ダッシュボード、グラフ、最新データなど）を開いたままにしていると、セッションの有効期間が延長され、自動ログアウト機能は無効になります。 * *Remember me for 30 days* オプションをチェックしてログインした場合。 * SSO Identity Provider (IdP) を介して認証が行われる場合、自動ログアウトで終了するのは Zabbix セッションのみです。IdP との SSO セッションはアクティブなままのため、ユーザーはパスワードを入力せずに再度ログインできる可能性があります。 *Auto-logout* には”0” も指定でき、この場合、プロファイル設定の更新後に自動ログアウト機能は無効になります。
Refresh	*Monitoring* メニューのページ上の情報をどのくらいの頻度で更新するかを設定します（最小 0 秒、最大 1 時間）。 時間サフィックスがサポートされています。例: 30s、90s、1m、1h。
Rows per page	リストの 1 ページあたりに表示する行数を設定します。行数が少ないほど（表示するレコード数が少ないほど）、読み込み時間は短くなります。
URL (after login)	ログイン後に表示する特定の URL を設定します。デフォルトの *Dashboards* の代わりに、たとえば *Monitoring* > *Triggers* の URL を指定できます。

通知

通知セクションでは、現在のユーザー通知をカスタマイズするためのオプションを提供します。

メディアタブでは、通知の配信に使用するメディアタイプやアドレス、およびそれらを使用するタイミングなど、ユーザーのメディアの詳細を指定できます。



メディアタイプが無効になっている場合:

- 名前の後に黄色の情報アイコンが表示されます。
- ステータス列に「Disabled」と表示されます。

Note:
ユーザー自身のメディア詳細を変更する権限は、ユーザーロールに基づいて付与または取り消すことができます（*Create and edit own media* オプションを参照）。他のユーザーのメディア詳細を変更するスーパー管理者ユーザーの権限も、ユーザーロールに基づいて付与または取り消すことができます（*Create and edit user media* オプションを参照）。

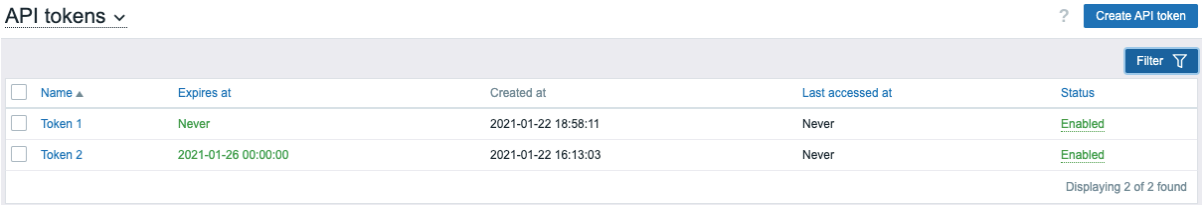
プロビジョニングされたユーザーについては、次の点に注意してください。

- プロビジョニングされたユーザーメディアは削除できません。
- プロビジョニングされたユーザーメディアは無効化/有効化できます。
- 有効期間、深刻度で使用、有効などのプロビジョニングされたユーザーメディア項目は手動で編集できます。
- プロビジョニングされたユーザーには、追加のユーザーメディア（たとえば追加のメールアドレス）を手動で追加できます。
- 手動で追加したユーザーメディアは削除できます。

Web インターフェース通知タブでは、グローバル通知を設定できます。

API トークン

API トークンセクションでは、ユーザーに割り当てられているトークンの表示、トークンの詳細の編集、および新しいトークンの作成ができます。このセクションは、ユーザーロールの設定で *Manage API tokens* アクションが許可されている場合にのみ、ユーザーが利用できます。



API トークンは、名前、有効期限、またはステータス（有効/無効）でフィルタリングできます。一覧内のトークンのステータスをクリックすると、トークンをすばやく有効化/無効化できます。また、一覧で複数のトークンを選択し、一覧の下にある *Enable/Disable* ボタンをクリックすることで、複数のトークンを一度に有効化/無効化することもできます。

Attention:
ユーザーは、Zabbix で自分に割り当てられたトークンの *Auth token* の値を表示できません。*Auth token* の値は、トークンの作成直後に一度だけ表示されます。トークンを紛失した場合は、再生成する必要があります。

1 グローバル通知

概要

グローバル通知は、Zabbix の Web インターフェース内で現在の画面にリアルタイムの障害を直接表示するための方法を提供します。

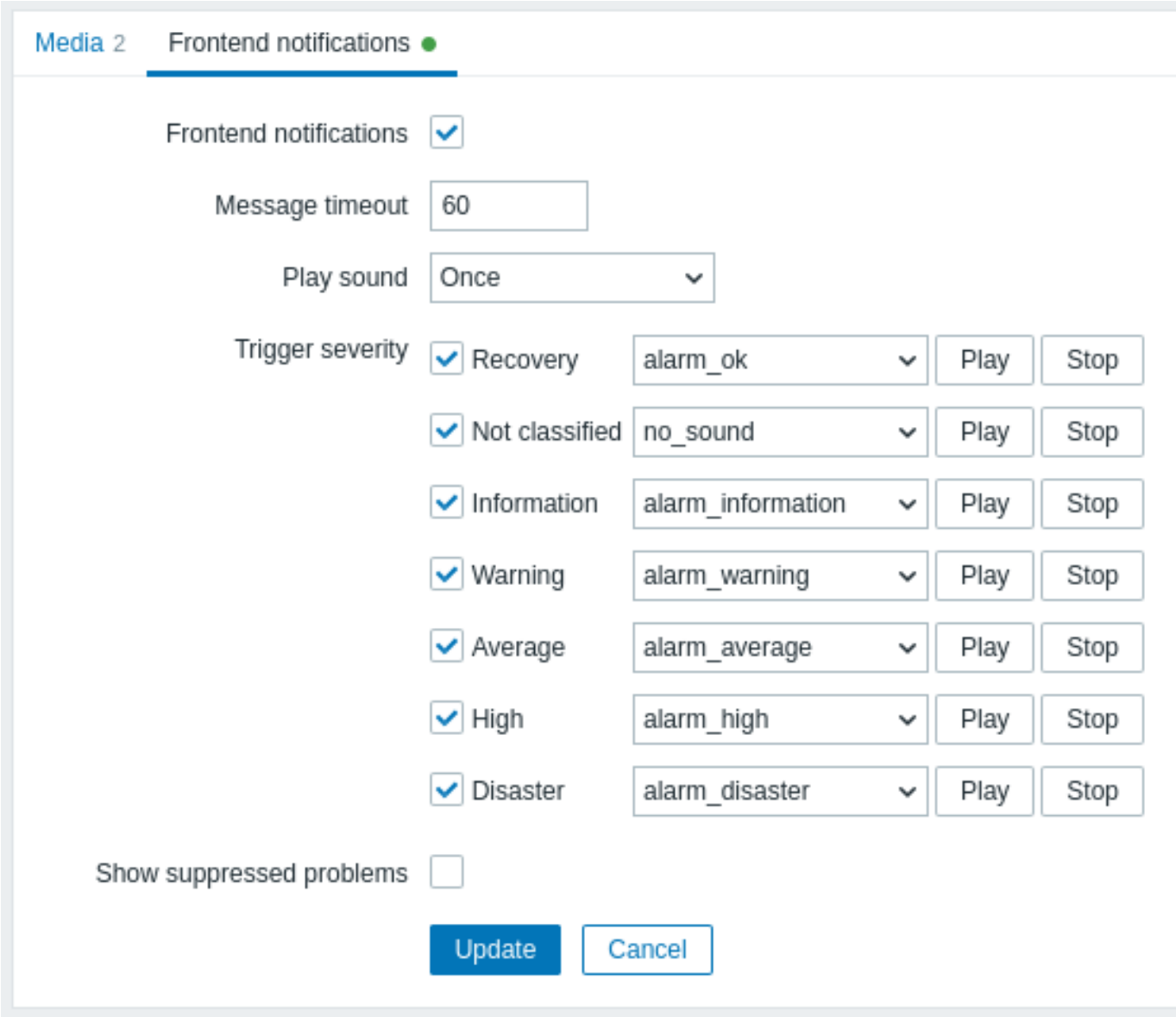
グローバル通知がない場合、障害または ダッシュボードセクション以外で作業していると、現在の障害に関する情報を受け取ることができません。グローバル通知により、Zabbix の Web インターフェース内の現在の場所に関係なく、この情報が表示されます。

グローバル通知には、メッセージの表示 と音の再生 の両方が含まれます。

Attention:
最近のブラウザのバージョンでは、音の自動再生が無効になっている場合があります（デフォルト）。その場合は、この設定を手動で有効にする必要があります。

設定

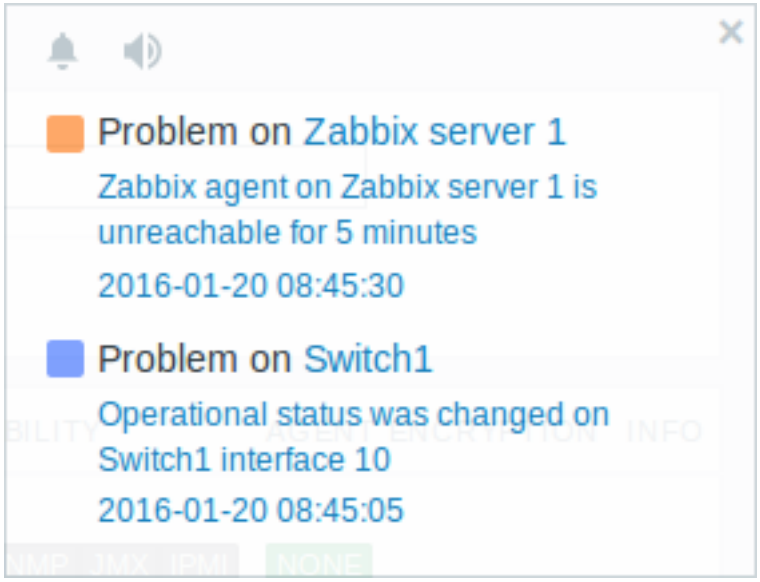
グローバル通知は、通知 セクションの **Web** インターフェース通知タブで、ユーザーごとに有効化できます。



Parameter	Description
Web インターフェース通知	チェックボックスをオンにすると、グローバル通知が有効になります。
メッセージのタイムアウト	メッセージを表示する期間を設定します。デフォルトでは、メッセージは画面上に 60 秒間表示されます。 時間のサフィックス をサポートしています。例: 30s、5m、2h、1d。
サウンドを再生	サウンドを再生する期間を設定します。 **1** 回 - サウンドは 1 回だけ最後まで再生されます。 **10** 秒 - サウンドは 10 秒間繰り返し再生されます。 メッセージのタイムアウト - メッセージが表示されている間、サウンドが繰り返し再生されます。
トリガーの深刻度	グローバル通知およびサウンドを有効にするトリガーの深刻度を設定します。深刻度ごとに適切なサウンドを選択することもできます。 深刻度が 1 つも選択されていない場合、メッセージは表示されません。 また、復旧メッセージは、選択された深刻度に対してのみ表示されます。たとえば、*Recovery* と *Disaster* が選択されている場合、*Disaster* 深刻度のトリガーに対する障害および復旧について、グローバル通知が表示されます。
抑制された障害を表示	ホストのメンテナンスにより通常は抑制されて表示されない障害についても、通知を表示するにはチェックボックスをオンにします。

表示されるグローバルメッセージ

メッセージが届くと、右側にあるフローティングセクションに表示されます。セクションヘッダーをドラッグすると、このセクションを自由に移動できます。



このセクションでは、いくつかのコントロールを利用できます。

- **Snooze** ボタンは、現在アクティブなアラーム音を消音します。
- **Mute/Unmute** ボタンは、アラーム音を再生する/しないを切り替えます。

2 ブラウザのサウンド

概要

サウンドはグローバル通知で使用されます。

Zabbix の Web インターフェースでサウンドを再生するには、ユーザープロファイルの *Frontend notifications* タブで *Frontend notifications* を有効にし、すべてのトリガー深刻度にチェックを入れる必要があります。さらに、グローバル通知ポップアップウィンドウでもサウンドを有効にする必要があります。

何らかの理由でデバイス上で音声を再生できない場合、グローバル通知ポップアップウィンドウの ボタンは常に「ミュート」状態のままとなり、 ボタンにマウスオーバーすると「このデバイスでは通知音声をサポートできません」というメッセージが表示されます。

デフォルトの音声クリップを含むサウンドは、MP3 形式のみサポートされています。

Zabbix の Web インターフェースのサウンドは、Linux 上の最近の Firefox および Opera ブラウザー、ならびに Windows 上の Chrome、Firefox、Microsoft Edge、Opera ブラウザーで正常にテストされています。

Attention:
最近のブラウザーのバージョンでは、サウンドの自動再生が（デフォルトで）無効になっている場合があります。その場合は、この設定を手動で有効にする必要があります。

4 グローバル検索

Zabbix の Web インターフェースでは、ホスト、ホストグループ、テンプレート、およびテンプレートグループを検索できます。

検索入力ボックスは、メニュー内の Zabbix ロゴの下にあります。検索は、*Enter* キーを押すか、 検索アイコンをクリックして開始できます。



入力した文字列が名前の一部に含まれるホストがある場合、ドロップダウンが表示され、そのようなホストがすべて一覧表示されます（一致した部分はオレンジ色で強調表示されます）。また、ホストの表示名が検索文字列として入力された技術名に一致する場合にも、そのホストがドロップダウンに表示されます。この場合、該当するホストは一覧表示されますが、強調表示はされません。

検索可能な属性

ホストは、次のプロパティで検索できます。

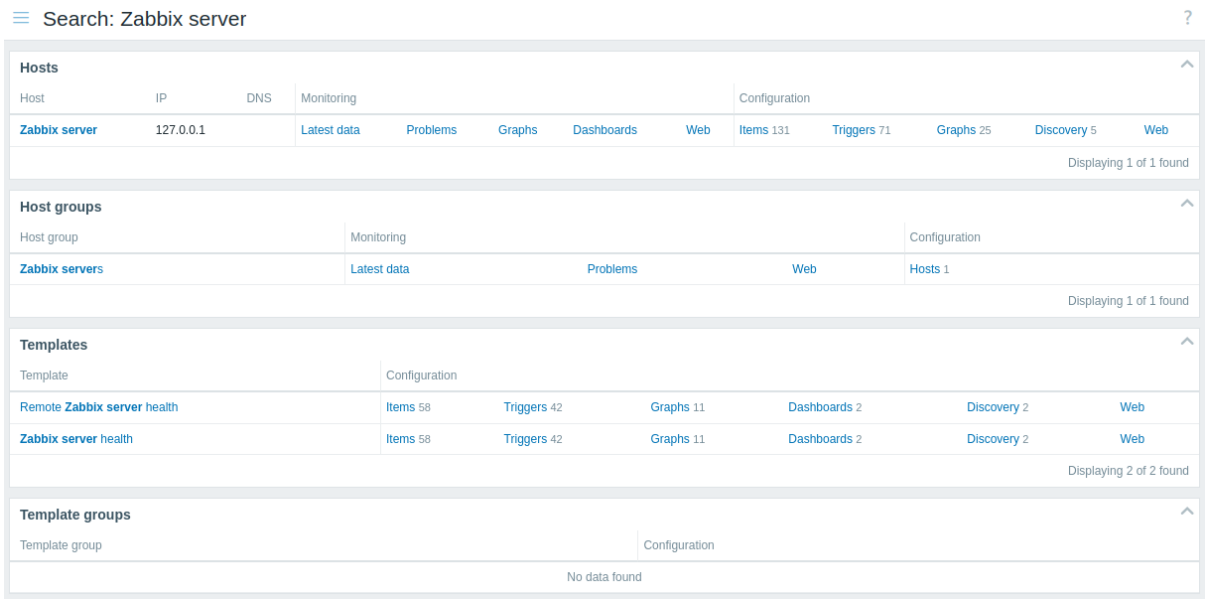
- ホスト名
- 表示名
- IP アドレス
- DNS 名

テンプレートは、名前または表示名で検索できます。表示名（テンプレート/ホストの）とは異なる名前で検索した場合、検索結果ではその名前が表示名の下に括弧付きで表示されます。

ホストグループおよびテンプレートグループは、名前で検索できます。親グループを指定すると、ネストされたすべてのグループも暗黙的に選択されます。

検索結果

検索結果は、ホスト、ホストグループ、テンプレート、テンプレートグループの 4 つの個別ブロックで構成されます。



各ブロックは個別に折りたたみ/展開できます。エントリ数は各ブロックの下部に表示されます。例えば、*13* 件中 *13* 件を表示のように表示されます。エントリがない場合、エントリ数は表示されません。1 つのブロック内に表示されるエントリ総数は 100 件に制限されています。

各エントリには、監視データおよび設定データへのリンクがあります。リンクの完全な一覧を参照してください。

すべての設定データ（アイテム、トリガー、グラフなど）については、見つかったエンティティの数が、エンティティ名の横に灰色で表示されます。エンティティ数が 0 の場合は、数値は表示されないことに注意してください。

有効なホストは青色、無効なホストは赤色で表示されます。

利用可能なリンク

各エントリについて、次のリンクを利用できます:

- Hosts
  - Monitoring
    - Latest data
    - Problems
    - Graphs
    - Host dashboards
    - Web scenarios
  - Configuration
    - Items
    - Triggers
    - Graphs
    - Discovery rules
    - Web scenarios
- Host groups
  - Monitoring
    - Latest data
    - Problems
    - Web scenarios
  - Configuration
    - Hosts
- Templates
  - Configuration
    - Items
    - Triggers
    - Graphs
    - Template dashboards
    - Discovery rules
    - Web scenarios
- Template groups
  - Configuration
    - Templates

5 フロントエンドメンテナンスモード

概要

アクセスを制限するために、Zabbix の Web インターフェースを一時的に無効化できます。これは、ユーザーによって開始される変更から Zabbix データベースを保護し、その整合性を維持するのに役立ちます。

Zabbix の Web インターフェースがメンテナンスモードの間は、データベースを安全に停止してメンテナンス作業を実行できます。

定義された IP アドレスからのユーザーは、メンテナンスモード中でも通常どおり Web インターフェースを操作できます。

設定

メンテナンスモードを有効にするには、maintenance.inc.php ファイル（Web サーバー上の Zabbix HTML ドキュメントディレクトリの /conf にあります）を開き、以下の行のコメントを解除します。

```
// メンテナンスモード。
define('ZBX_DENY_GUI_ACCESS', 1);

// Webインターフェースへの接続を許可するIPアドレスの配列（任意）。
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// 警告画面に表示されるメッセージ（任意）。
$ZBX_GUI_ACCESS_MESSAGE = '15:00までMySQLデータベースをアップグレードしています。しばらくお待ちください..
```

Note:
ほとんどの場合、maintenance.inc.php ファイルは、Web サーバー上の Zabbix HTML ドキュメントディレクトリの /conf にあります。
ただし、一部のオペレーティングシステムおよび Web サーバーでは、異なる場所が使用される場合があります。
例えば、以下の環境での場所は次のとおりです。

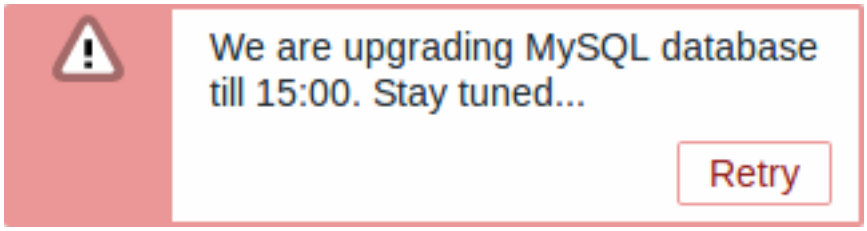
- SUSE および RedHat では /etc/zabbix/web/maintenance.inc.php
- Debian 系システムでは /usr/share/zabbix/conf/

あわせてPHP ファイルのコピー も参照してください。

Parameter	Details
ZBX_DENY_GUI_ACCESS	任意の値で定義されている場合、メンテナンスモードが有効になります。 メンテナンスモードを無効にするには、コメントアウトするか削除してください。
ZBX_GUI_ACCESS_IP_RANGE	接続を許可する IP アドレスの配列（任意）。 例えば: array('192.168.1.1', '192.168.1.2')
ZBX_GUI_ACCESS_MESSAGE	メンテナンスについてユーザーに通知するメッセージ（任意）。 未定義の場合、デフォルトメッセージ *'Zabbix is under maintenance'* が使用されます。

表示

メンテナンスモード中に Zabbix Web インターフェースへアクセスしようとすると、ユーザーには次の画面が表示されます。メンテナンス終了時にユーザーの操作なしで通常の状態に戻れるよう、この画面は 30 秒ごとに更新されます。



ZBX_GUI_ACCESS_IP_RANGE で定義された IP アドレスは、通常どおり Web インターフェースにアクセスできます。

6 ページパラメーター

概要

ほとんどの Zabbix Web インターフェースページは、表示内容を制御するさまざまな HTTP GET パラメータをサポートしています。これらは、URL の後に疑問符 (?) を付け、その後に parameter=value の組み合わせをアンパサンド (&) で区切って指定することで渡すことができます。

監視 > 障害

以下のパラメータがサポートされています。

Parameter	Description	Example
show	フィルターオプション *Show*。 指定可能な値: 1 - 最近の障害; 2 - すべて; 3 - 障害状態。	show=1
name	フィルターオプション *Problem*: 自由形式の文字列。	name=Zabbix agent

Parameter	Description	Example
severities	フィルターオプション *Severity*: `severities[*]=*` 形式の、選択した深刻度の配列（* は深刻度レベルに置き換えてください）。 指定可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 致命的。	severities[3]=3
inventory	フィルターオプション *Host inventory*: インベントリフィールド `[field],[value]` の配列	inventory[0][field]=type&invento
evaltype	フィルターオプション *Tags*: タグの評価方法。 指定可能な値: 0 - and/or; 2 - or。	evaltype=0
tags	フィルターオプション *Tags*: 定義されたタグ `[tag], [operator], [value]` の配列 指定可能なoperatorの値: 0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。	tags[0][tag]=target&tags[0][ope
show_tags	フィルターオプション *Show tags*。 指定可能な値: 0 - なし; 1 - 1 つ; 2 - 2 つ; 3 - 3 つ。	show_tags=3
tag_name_format	フィルターオプション *Tag name*。 指定可能な値: 0 - 完全名; 1 - 短縮名; 2 - なし。	tag_name_format=1
tag_priority	フィルターオプション *Tag display priority*: タグ表示優先度のカンマ区切り文字列	tag_priority=customer, target
show_suppressed	フィルターオプション *Show suppressed problems*。 指定可能な値: 1 - 表示する; 2 - 表示しない。	show_suppressed=1
acknowledgement_status	フィルターオプション *Acknowledgement status*。 指定可能な値: 0 - すべて; 1 - 未確認; 2 - 確認済み。	acknowledgement_status=0
acknowledged_by_me	フィルターオプション *By me*。`acknowledgement_status=2` の場合のみサポートされます。 指定可能な値: 0 - 無効; 1 - 有効。	acknowledged_by_me=1

Parameter	Description	Example
compact_view	フィルターオプション *Compact view*。 指定可能な値: 0 - 無効; 1 - 有効。	compact_view=1
highlight_row	フィルターオプション *Highlight whole row*。 指定可能な値: 0 - 無効; 1 - 有効。 Zabbix 7.4.3 より前のバージョンでは、このオプションは `compact_view=1` の場合のみサポートされます。	highlight_row=1
filter_name	フィルタープロパティオプション *Name*: 自由形式の文字列	filter_name=Databases
filter_show_counter	フィルタープロパティオプション *Show number of records*。 指定可能な値: 0 - 無効; 1 - 有効。	filter_show_counter=1
filter_custom_time	フィルタープロパティオプション *Set custom time period*。 指定可能な値: 0 - 無効; 1 - 有効。	filter_custom_time=1
sort	ソート対象の列。 指定可能な値: clock - *Time* 列でソート; host - *Host* 列でソート; severity - *Severity* 列でソート; name - *Problem* 列でソート。	sort=clock
sortorder	結果のソート順。 指定可能な値: DESC - 降順; ASC - 昇順。	sortorder=DESC
age_state	フィルターオプション *Age less than*。`show=3` の場合のみサポートされます。 指定可能な値: 0 - age パラメータを無効化; 1 - age パラメータを有効化。	age_state=1
age	フィルターオプション *Age less than*: 整数、日数。`age_state=1` かつ `show=3` の場合のみサポートされます。	age=7
groupids	フィルターオプション *Host groups*: ホストグループ ID の配列。	groupids[]=4
hostids	フィルターオプション *Hosts*: ホスト ID の配列	hostids[]=10084
triggerids	フィルターオプション *Triggers*: トリガー ID の配列	triggerids[]=22382
show_timeline	フィルターオプション *Show timeline*。`compact_view=1` ではサポートされません。 指定可能な値: 0 - 表示しない; 1 - 表示する。	show_timeline=1
details	フィルターオプション *Show details*。 指定可能な値: 0 - 表示しない; 1 - 表示する。	details=1
from	日付範囲の開始。相対指定が可能です（例: now-1m）。 `filter_custom_time=1` の場合のみサポートされます	from=now-2h
to	日付範囲の終了。相対指定が可能です（例: now-1m）。 `filter_custom_time=1` の場合のみサポートされます	to=now

関連項目: 障害ページのフィルターオプション。

キオスクモード

サポートされている Web インターフェースのページでは、URL パラメータを使用してキオスクモードを有効化できます。例えば、ダッシュボードでは次のようになります。

- /zabbix.php?action=dashboard.view&kiosk=1 - キオスクモードを有効化
- /zabbix.php?action=dashboard.view&kiosk=0 - 通常モードを有効化

スライドショー

ダッシュボードでスライドショーをアクティブにするには次の通りです。

- /zabbix.php?action=dashboard.view&slideshow=1 - スライドショーをアクティブに

7 定義

概要

Web インターフェース内の多くの項目は Web インターフェース自体を使用して設定できますが、一部のカスタマイズは現在のところ定義ファイルを編集することでのみ可能です。

このファイルは defines.inc.php で、Zabbix の HTML ドキュメントディレクトリの /include にあります。

パラメータ

このファイル内で、ユーザーにとって関心のある可能性があるパラメータ:

- ZBX_MIN_PERIOD

グラフの最小期間（秒単位）。デフォルトでは 1 分です。

- GRAPH_YAXIS_SIDE_DEFAULT

シンプルグラフにおける Y 軸のデフォルトの位置、およびカスタムグラフにアイテムを追加する際のドロップダウンボックスのデフォルト値です。指定可能な値: 0 - 左、1 - 右。

デフォルト: 0

- ZBX_SESSION_NAME

Zabbix Web インターフェースのセッションクッキー名として使用される文字列です。

デフォルト: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Vault response のデータキャッシュを無効化するために使用される TTL タイムアウト（秒単位）です。Vault response のキャッシュを無効にするには 0 を設定します。

デフォルト: 60

- SUBFILTER_VALUES_PER_GROUP

グループごとのサブフィルター値の数です（たとえば、最新データ のサブフィルター）。

デフォルト: 1000

- ZBX_MAX_WIDGET_LINES

表示するウィジェット行の最大数です。

デフォルト: 1000

8 独自テーマ作成

概要

デフォルトで Zabbix はいくつか事前定義されたテーマを提供します。独自のテーマを作成するには、ここに記載されている手順に従ってください。何かいいものを作成した場合は、作業の結果を Zabbix コミュニティと自由に共有してください。

ステップ 1

独自のテーマを定義するには、CSS ファイルを作成し、それを `assets/styles/`フォルダー（たとえば、*custom-theme.css*）に保存します。別のテーマからファイルをコピーして、それを元にテーマを作成するか、最初から始めることができます。

ステップ 2

APP::getThemes() メソッドによって返されるテーマのリストにテーマを追加します。このためには APP クラスの ZBase::getThemes() メソッドをオーバーライドします。これは *include/classes/core/APP.php* の閉じ中括弧の前に次のコードを追加することで実行できます。

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme')
    ]);
}
```

Attention:
引用符の最初のペアで指定する名前は、拡張子なしのテーマファイルの名前と一致する必要があることに注意してください。

複数のテーマを追加するには、最初のテーマの下にそれらをリストします。次に例を示します。

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme'),
        'anothertheme' => _('Another theme'),
        'onemoretheme' => _('One more theme')
    ]);
}
```

最後のテーマを除くすべてのテーマには、末尾にカンマが必要であることに注意してください。

Note:
グラフの色を変更するには、エントリを *graph_theme* データベーステーブルに追加する必要があります。

ステップ 3

新しいテーマをアクティブにします。

Zabbix フロントエンドで、新しいテーマをデフォルトのテーマに設定したり、ユーザープロファイルで各ユーザーのテーマを変更することができます。

新しいルックアンドフィールをお楽しみください！

9 デバッグモード

概要

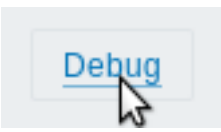
デバッグモードは、フロントエンドページの問題を診断するために使用できます。

設定

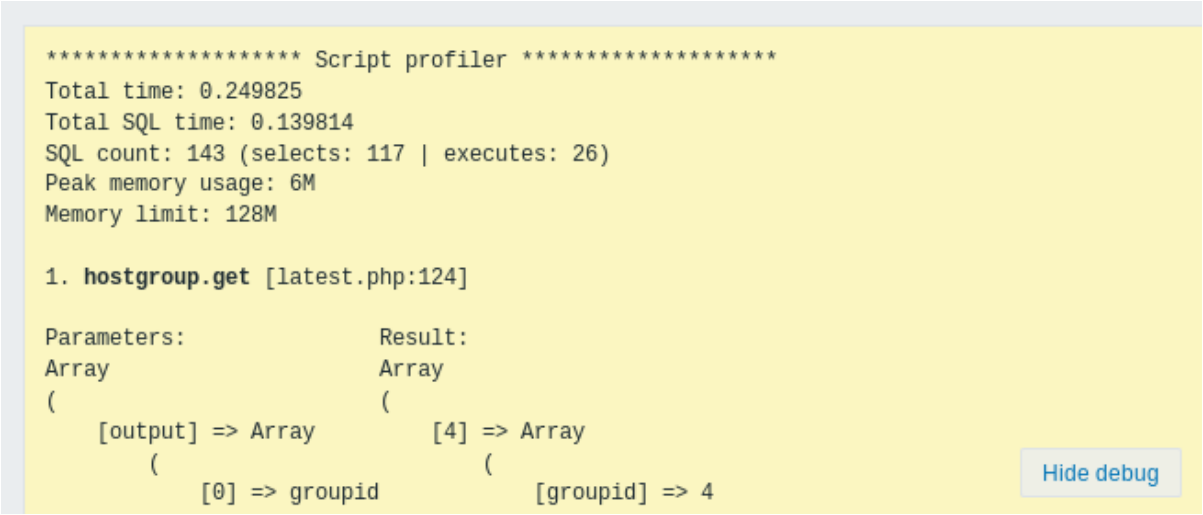
デバッグモードは、ユーザーグループに属する個々のユーザーに対して有効化できます。

- ユーザーグループを設定するとき
- 設定済みのユーザーグループを表示するとき

ユーザーグループでデバッグモードが有効になっている場合、そのグループのユーザーにはブラウザウィンドウの右下隅に *Debug* ボタンが表示されます。



Debug ボタンをクリックすると、ページ内容の下に新しいウィンドウが開き、ページの SQL 統計情報に加えて、API 呼び出しの一覧と個々の SQL 文が表示されます。



ページでパフォーマンスの問題が発生した場合、このウィンドウを使用して問題の根本原因を調査できます。

Warning:
デバッグモードを有効にすると、Web インターフェースのパフォーマンスに悪影響があります。

10 Zabbix 用 Cookie

概要

このページでは、Zabbix で使用される Cookie の一覧を示します。

Name	説明	値	Expires/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	base64 でエンコードされた JSON として保存される Zabbix Web インターフェースのセッションデータ		セッション（ブラウジングセッションの終了時に期限切れ）	+	+（Web サーバーで HTTPS が有効な場合のみ）
tab	アクティブなタブ番号。この Cookie は複数のタブがあるページ（例: ホスト、トリガー、またはアクションの設定ページ）でのみ使用され、ユーザーがプライマリタブから別のタブ（タグタブや依存関係タブなど）へ移動したときに作成されます。 0 はプライマリタブに使用されます。	例: 1	セッション（ブラウジングセッションの終了時に期限切れ）	-	-
browserwarning_ignore	ブラウザーの使用に関する警告を無視するかどうか。	yes	セッション（ブラウジングセッションの終了時に期限切れ）	-	-
system-message-ok	ページが再読み込みされるとすぐに表示するメッセージ。	プレーンテキストメッセージ	セッション（ブラウジングセッションの終了時に期限切れ）またはページの再読み込み直後	+	-
system-message-error	ページが再読み込みされるとすぐに表示するエラーメッセージ。	プレーンテキストメッセージ	セッション（ブラウジングセッションの終了時に期限切れ）またはページの再読み込み直後	+	-

Note:
Web サーバーのディレクティブによって Zabbix Cookie に'HttpOnly' フラグを強制することはサポートされていません。

脚注

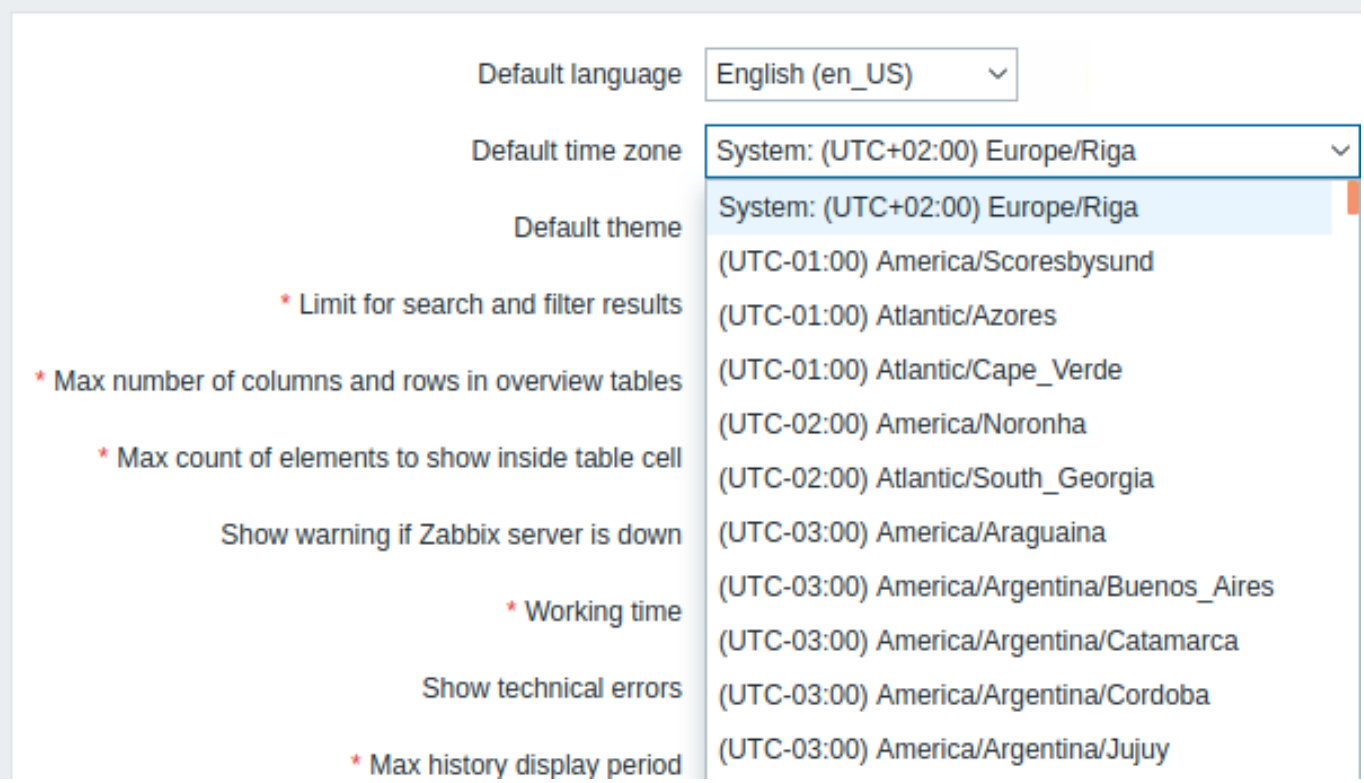
[1] `HttpOnly` が'true' の場合、Cookie は HTTP プロトコル経由でのみアクセス可能になります。これは、その Cookie が JavaScript などのスクリプト言語からアクセスできなくなることを意味します。この設定は、XSS 攻撃による認証情報の盗難を効果的に軽減するのに役立ちます（ただし、すべてのブラウザーでサポートされているわけではありません）。

[2] `Secure` は、その Cookie がクライアントからの安全な HTTPS 接続経由でのみ送信されるべきであることを示します。'true' に設定されている場合、安全な接続が存在するときにのみ Cookie が設定されます。

11 タイムゾーン

概要

Web インターフェースのタイムゾーンは、Web インターフェースでグローバルに設定でき、個々のユーザーごとに調整することもできます。



System が選択されている場合、Web サーバーのタイムゾーンが Web インターフェースに使用されます（設定されている場合は php.ini の'date.timezone' の値を含む）。一方、Zabbix サーバーは実行中のマシンのタイムゾーンを使用します。

Note:
Zabbix サーバーが指定されたグローバル/ユーザーのタイムゾーンを使用するのは、通知内でマクロを展開する場合のみです（例: {EVENT.TIME} はユーザーごとに異なるタイムゾーンに展開されることがあります）。また、通知が送信される時間制限にも使用されます（ユーザーのメディア設定の「When active」設定を参照）。

タイムゾーンの選択は、Web インターフェースの時刻/日付形式には影響しません。代わりに、インターフェース言語を調整できます（インストール時、またはユーザー設定で設定可能です）。*English (en_US)* を選択すると、Web インターフェースで米国式の時刻/日付形式も有効になります。

設定

グローバルタイムゾーンは、以下の方法で設定できます。

- インストール時に Web インターフェースで手動設定できます
- 管理 → 一般 → *GUI* で変更できます

ユーザーレベルのタイムゾーンは、以下の方法で設定できます。

- ユーザーの設定/更新 時に設定できます
- 各ユーザーが自分のユーザープロファイルで設定できます

関連情報: スケジュール間隔を使用する際のタイムゾーンの調整。

12 パスワードのリセット

概要　このセクションでは、Zabbix のユーザーパスワードをリセットする手順について説明します。

ステップ　Zabbix パスワードを忘れてログインできない場合は、Zabbix 管理者に問い合わせてください。

スーパー管理者ユーザーは、ユーザー設定フォーム 内のすべてのユーザーのパスワードを変更できます。

スーパー管理者がパスワードを忘れてログインできない場合は、次の SQL クエリを実行して、デフォルトパスワードをスーパー管理者ユーザーに適用する必要があります (「Admin」をスーパー管理者ユーザー名に置き換えます)

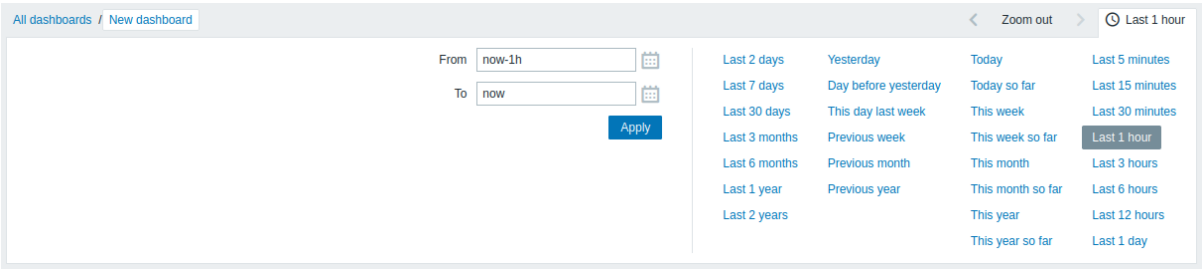
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username = 'Admin';

このクエリを実行すると、ユーザー名'Admin' のパスワードは *zabbix* に設定されます。最初のログイン時に必ずパスワードをデフォルトから変更してください。

13 期間セレクター

概要

Time period セレクターでは、よく使用される期間をマウスクリック 1 回で選択できます。これは、右上隅にある *Time period* タブをクリックすると展開または折りたたみできます。



Today、*This week* などのオプションでは、将来の時間/日を含む期間全体が表示されます。*Today so far*、*This week so far* などのオプションでは、経過した時間のみが表示されます。

期間を選択すると、 矢印ボタンをクリックして、時間を前後に移動できます。*Zoom out* ボタンを使用すると、期間を各方向に 50% ずつズームアウトできます。

Note:
グラフ（*Custom time period* を持つものを除く）では、マウスの左ボタンでグラフ内の領域をハイライトして、表示する期間を選択することもできます。左マウスボタンを離すと、グラフはハイライトした領域にズームインします。グラフ内をダブルクリックしてズームアウトすることもできます。

From/To フィールドには、選択した期間が絶対時刻構文（形式 `Y-m-d H:i:s`）または相対時刻構文で表示されます。相対時刻期間には、1 つまたは複数の数学演算（- または +）を含めることができます。例えば、`now-1d` や `now-1d-2h+5m` です。

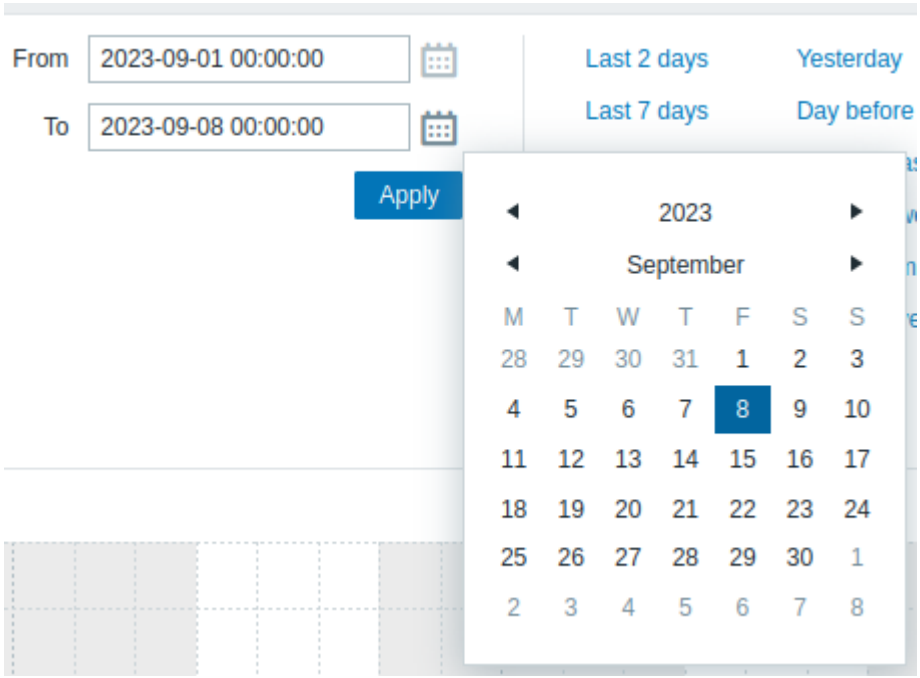
次の相対時刻の省略形がサポートされています。

- `now`
- `s`（秒）
- `m`（分）
- `h`（時間）
- `d`（日）
- `w`（週）
- `M`（月）
- `y`（年）

Time period セレクターでは精度もサポートされています（例えば、`now-1d/M` の `/M`）。精度の詳細は次のとおりです。

Precision	From	To
m	Y-m-d H:m:00	Y-m-d H:m:59
h	Y-m-d H:00:00	Y-m-d H:59:59
d	Y-m-d 00:00:00	Y-m-d 23:59:59
w	その週の月曜日 00:00:00	その週の日曜日 23:59:59
M	その月の初日 00:00:00	その月の最終日 23:59:59
y	その年の 1 月 1 日 00:00:00	その年の 12 月 31 日 23:59:59

Date picker を使用して期間を選択することもできます。開くには、*From/To* フィールドの横にあるカレンダーアイコンをクリックします。



Note:
日付ピッカー内では、Tab、Shift+Tab、およびキーボードの矢印キーを使用して、年/月/日付の間を移動できます。Enter を押すと選択が確定します。

例

From	To	選択期間
now/d	now/d	今日の 00:00 - 23:59
now/d	now/d+1d	今日の 00:00 - 明日の 23:59
now/w	now/w	今週の月曜日 00:00:00 - 日曜日 23:59:59
now-1y/w	now-1y/w	1 年前の月曜日 00:00:00 - 日曜日 23:59:59 の週

Attention:
To パラメータに”now/M+1M” を使用すると、31 日が加算される場合があり、月の日数によっては日付が 1~3 日ずれることがあります。たとえば、1 月に使用した場合、想定される 2 月 28 日ではなく 3 月 2 日になる可能性があります。この問題を回避するには、月の長さを正確に考慮して調整する”now/M-3d+1M/M” を使用してください。同様に、*From* を過去方向に設定する場合は、”now/M+3d-1M/M” を使用してください。

19. ベストプラクティス

概要　このセクションでは、Zabbix を設定するためのベストプラクティスについて説明します。

これらのプラクティスは Zabbix の動作に必須ではありませんが、最適で安全な使用エクスペリエンスを実現するために実装することを強くお勧めします。

1 セキュリティのベストプラクティス

概要

このセクションでは、Zabbix を安全にセットアップするためのベストプラクティスを説明しています。

Zabbix の機能には必須ではありませんが、システムのセキュリティを向上させるために推奨されます。

UTF-8 エンコーディング

Zabbix がサポートするエンコーディングは UTF-8 のみです。セキュリティ上の欠陥なく動作することが知られています。ユーザーは他のエンコーディングを使用する場合、既知のセキュリティ上の問題があることを留意してください。

Windows インストーラーのパス　Windows インストーラーを使用する場合は、インストーラーによって提供されるデフォルトのパスを使用することをお勧めします。適切なアクセス許可なしでカスタムパスを使用すると、インストールのセキュリティが損なわれる可能性があります。

ユーザー定義のグローバルスクリプト内のマクロ

セキュリティを強化するため、ユーザー定義のグローバルスクリプトでは、プレーンなマクロの代わりにマクロ関数を使用することを推奨します。マクロは自動的にエスケープされないためです。

Zabbix セキュリティアドバイザリと CVE データベース

*Zabbix セキュリティアドバイザリと CVE データベース*を参照してください。

HTML メールテンプレート

HTML メールで使用するメッセージテンプレートを作成または編集する場合は、必ずすべてのマクロを `htmlencode()` マクロ関数で囲んでください。例えば、次のようになります。

```
<b>Problem started</b> at {{EVENT.TIME}.htmlencode()} on {{EVENT.DATE}.htmlencode()}<br><b>Problem name:</
{{EVENT.NAME}.htmlencode()}<br><b>Host:</b> {{HOST.NAME}.htmlencode()}
<br><b>Severity:</b>
{{EVENT.SEVERITY}.htmlencode()}<br><b>Operational
data:</b> {{EVENT.OPDATA}.htmlencode()}<br><b>Original problem ID:</b> {{EVENT.ID}.htmlencode()}<br>{{TRIG
```

`htmlencode()` を使用すると、マクロ値に含まれる HTML 文字がエスケープされ、通知への HTML の挿入を防止できます（例えば、攻撃者が悪意のあるリンクやフィッシングリンクを通知に挿入することを防げます）。

注意: Zabbix が提供するデフォルトの HTML メールメッセージでは、すでにマクロに `htmlencode()` が適用されています。このガイダンスは、既存のテンプレートを編集する場合にも、新しいテンプレートを作成する場合にも適用されます。テンプレートを使用して HTML メールを送信する前に、マクロがエンコードされていることを確認してください。

1 アクセス制御

概要

このセクションでは、アクセス制御を安全に設定するためのベストプラクティスについて説明します。

最小権限の原則

ユーザーアカウントは常に、できるだけ少ない権限で実行する必要があります。これは、Zabbix Web インターフェースのユーザーアカウント、データベースユーザー、Zabbix サーバー/プロキシ/エージェントのプロセスユーザーには、目的の機能を実行するために必要な権限のみが与えられる必要があることを意味します。

> **Attention:**
> 'zabbix' ユーザーに追加の権限を与えると、設定ファイルにアクセスし、インフラストラクチャのセキュリティを侵害する可能性のある操作を実行できるようになります。

ユーザーアカウントの権限を設定するときは、Zabbix Web インターフェースのユーザータイプを考慮する必要があります。*Admin* ユーザータイプは、*Super Admin* ユーザータイプより権限が少ないにもかかわらず、設定を管理し、カスタムスクリプトを実行できることに注意してください。

Note:
一部の情報は、権限を持たないユーザーでも利用できます。たとえば、アラート → スクリプトは *Super Admin* ユーザーのみが利用できますが、スクリプトは Zabbix API を通じて取得することもできます。この場合、スクリプトのアクセス許可を制限し、スクリプトから機密情報 (アクセス資格情報など) を除外すると、グローバルスクリプトで利用可能な機密情報の公開を避けることができます。

Zabbix エージェントのセキュアユーザー

デフォルトでは、Zabbix サーバーと Zabbix エージェントのプロセスは単一の'zabbix' ユーザーを共有します。Zabbix エージェントがサーバー設定の機密情報 (データベースのログイン情報など) にアクセスできないようにするには、エージェントを別のユーザーとして実行する必要があります。

1. セキュアユーザーを作成します。
2. このユーザーをエージェント設定ファイルの`User`パラメータで指定します。
3. 管理者権限でエージェントを再起動します。指定したユーザーに権限が与えられます。

SSL 設定への書き込み権限の無効化 (Windows)

Windows 上で Zabbix エージェントをコンパイルし、OpenSSL を保護されていないディレクトリ (例: `c:\openssl-64bit`、`C:\OpenSSL-Win64-111-static`、`C:\dev\openssl`) に配置している場合は、管理者以外のユーザーによるこのディレクトリへの書き込み権限を必ず無効にしてください。無効にしないと、Zabbix エージェント 2 は権限のないユーザーが変更できるパスから SSL 設定を読み込み、潜在的なセキュリティ脆弱性が発生します。

Zabbix コンポーネントのセキュリティの強化

Zabbix コンポーネントのセキュリティを強化するために、一部の機能をオフにすることができます。

- Zabbix サーバーでのグローバルスクリプトの実行は、サーバー設定で EnableGlobalScripts=0 を設定することで無効にできます。
- Zabbix プロキシでのグローバルスクリプトの実行は、デフォルトで無効になっています (プロキシ設定で EnableRemoteCommands=1 を設定することで有効にできます)。
- Zabbix エージェントでのグローバルスクリプトの実行は、デフォルトで無効になっています (エージェント設定で許可されている各コマンドに対して、AllowKey=system.run[<command>,*] パラメータを追加することで有効にできます)。
- ユーザーの HTTP 認証は、フロントエンド設定ファイル (zabbix.conf.php) で `$ALLOW_HTTP_AUTH=false` を設定することで無効にできます。フロントエンドを再インストールすると (setup.php を実行すると)、このパラメータが削除されることに注意してください。

Windows 上の Zabbix エージェントによる UNC パスへのアクセス

Windows 上の Zabbix エージェントは、`vfs.file.*`、`vfs.dir.*`、`modbus.get`、`perf_counter*` などのアイテムで UNC パス (`\\server\share\file.txt` のような SMB 共有) をたどります。これは、状況によってはセキュリティリスクとなる可能性があります。

Windows が UNC パスへのアクセスを要求されると、そのサーバーに対して認証を試みます。これは、Zabbix エージェントへの悪意のあるリクエストによって、NTLM ハッシュがリクエスト元のサーバーに漏えいする可能性があることを意味します。必要に応じて、ユーザーはAllowKey およびDenyKey 設定パラメータを使用して、これを軽減できます。

1 MySQL/MariaDB のセキュリティ保護

概要

このセクションでは、MySQL/MariaDB データベースを保護するためのベストプラクティスについて説明します。

Note:
基本的なセットアップについては、デフォルトのMySQL/MariaDB データベースの作成手順を参照してください。この手順には、Zabbix データベースに対する完全な権限を持つ'zabbix' ユーザーの作成が含まれます。このユーザーは、Zabbix をアップグレードするときにデータベース構造を変更するために必要な権限も持つデータベース所有者です。

セキュリティを強化するには、最小限の権限を持つ追加のデータベースロールとユーザーを作成することをお勧めします。これらのロールとユーザーは、最小権限の原則に基づいて、つまり、意図した機能を実行するために必要な権限のみを設定する必要があります。

ユーザーロールの作成

対応する権限を持つ次のロールを作成します。

- **zbx_srv** - Zabbix サーバーとプロキシを実行するためのロール

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - Zabbix フロントエンドと API を実行するためのロール。

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - テーブルバックアップのロール

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

Note:
テーブルの復元とアップグレードはデータベース所有者が実行する必要があります。

- **zbx_part** - データベースのパーティション化に対する権限セットが制限されたロール。このロールは特定のデータベーステーブルに対する権限を付与するため、データベースの作成後にのみ作成できることに注意してください。

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

ロールを作成した後、ユーザーに割り当てることができます。

ユーザーロールの割り当て

To assign the created user roles, create users and assign the relevant roles to them. 作成したユーザーロールを割り当てるには、ユーザーを作成し、関連するロールをユーザーに割り当てます。<user>、<host>、<role>、<password> は必要に応じて置き換えてください。

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- For MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

たとえば、Zabbix サーバーとプロキシを実行するためのロールを作成して割り当てるには、次のようにします。

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 PostgreSQL/TimescaleDB のセキュリティ保護

概要

このセクションでは、PostgreSQL データベースを保護するためのベストプラクティスについて説明します。

Note:
基本的なセットアップについては、デフォルトのPostgreSQL データベースの作成手順を参照してください。この手順には、Zabbix データベースに対する完全な権限を持つ'zabbix' ユーザーの作成が含まれます。このユーザーは、Zabbix をアップグレードするときにデータベース構造を変更するために必要な権限も持つデータベース所有者です。

セキュリティを強化するには、安全なスキーマ使用パターンを構成し、最小限の権限を持つ追加のデータベースロールとユーザーを作成することをお勧めします。これらのロールとユーザーは、最小権限の原則((/manual/best_practices/security/access_control#principle-of-least-privilege) に基づいて、つまり、意図した機能を実行するために必要な権限のみを設定する必要があります。

データベースのセットアップ

データベース所有者となるユーザーを作成し、Zabbix データベースを作成します。データベースの所有者は、データベースの作成で指定されているユーザーです。

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

Note:
データベースのクリーンなインストールまたはアップグレードは、データベース所有者が実行する必要があります。これは、データベースオブジェクトをドロップする、あるいはその定義を変更する権限がデータベース所有者に固有の特権であり、許可または取り消されることができないためです。

Attention:
このページの次のコマンドは、PostgreSQL への接続が zabbix データベースに作成されている間に実行する必要があります。

zabbix スキーマを作成し、データベースの所有者 (usr_owner) をこのスキーマの所有者に設定します。

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

安全なスキーマ使用パターンを設定します。

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Note: search_path should point to the "zabbix" schema:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

データベースを設定した後、ユーザーロールの作成に進みます。

ユーザーロールの作成

対応する権限を持つ次の役割を作成します。

- **zbx_srv** - Zabbix サーバーとプロキシの実行の役割:

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - Zabbix フロントエンドと API を実行するための役割:

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - テーブルバックアップの役割:

```
CREATE ROLE zbx_bckp;
GRANT CONNECT ON DATABASE zabbix TO zbx_bckp;
GRANT USAGE ON SCHEMA zabbix TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT ON TABLES TO zbx_bckp;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

Note:
テーブルの復元は、データベースの所有者によってのみ可能です。

役割を作成した後、ユーザーに割り当てることができます。

ユーザーロールの割り当て

作成されたユーザーロールを割り当てるには、ユーザーを作成し関連するロールを割り当てます。<user> と <role>、<password> は適切に置き換えてください。

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

たとえば、Zabbix サーバーとプロキシを実行するロールを作成して割り当てるには、

```
CREATE USER usr_srv WITH ENCRYPTED password 'password';
GRANT zbx_srv TO usr_srv;
```

TimescaleDB を使用したデータベースパーティショニング

データベースパーティショニングは、TimescaleDB によって促進されます。TimescaleDB を利用するには、Zabbix サーバーにはデータベース所有者の権限が必要です。

PostgreSQL zabbix スキーマが既に zabbix データベースを作成していた場合、次のコマンドで TimescaleDB を有効にすることができます。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

2 暗号化

概要

このセクションでは、暗号化を安全に設定するためのベストプラクティスについて説明します。

Zabbix の Web インターフェースの SSL のセットアップ

RHEL ベースのシステムでは、mod_ssl パッケージをインストールします:

```
dnf install mod_ssl
```

SSL 鍵用のディレクトリを作成します:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

SSL 証明書を作成します:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

プロンプトに適切に入力します。最も重要な行は、Common Name を指定する行です。サーバーに関連付けるドメイン名を入力する必要があります。ドメイン名がない場合は、代わりにパブリック IP アドレスを入力することができます。

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:example.com
Email Address []:
```

Apache SSL 設定ファイル (/etc/httpd/conf.d/ssl.conf) を編集します:

```
DocumentRoot "/usr/share/zabbix"
ServerName example.com:443
SSLCertificateFile /etc/httpd/ssl/apache-selfsigned.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/apache-selfsigned.key
```

変更を反映させるため Apache のサービスを再起動します:

```
systemctl restart httpd.service
```

3 Web サーバー

概要

このセクションでは、Web サーバーを安全に設定するためのベストプラクティスについて説明します。

ルート URL から Zabbix の SSL へのリダイレクトを強制する

RHEL ベースのシステムでは、Apache の設定 (`/etc/httpd/conf/httpd.conf`) に仮想ホストを追加し、ドキュメントルートから Zabbix の SSL URL への恒久的なリダイレクトを設定します。`example.com` は実際のサーバー名に置き換えてください。

```
#### 行を追加:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

変更を適用するには、Apache サービスを再起動します。

```
systemctl restart httpd.service
```

Web サーバーで HTTP Strict Transport Security (HSTS) を有効にする

Zabbix Web コンソールをプロトコルダウングレード攻撃から保護するには、Web サーバーでHSTSポリシーを有効にすることをお勧めします。

Apache 構成で Zabbix フロントエンドの HSTS ポリシーを有効にするには、次の手順に従います。

1. 仮想ホストの構成ファイルを見つけます。

- RHEL ベースのシステムの場合、`/etc/httpd/conf/httpd.conf`
- Debian/Ubuntu の場合、`/etc/apache2/sites-available/000-default.conf`

2. 次のディレクティブを仮想ホストの設定ファイルに追加します。

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. 変更を適用させるため、Apache サービスを再起動します。

```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Zabbix でセキュアなセッションクッキーと SameSite セッションクッキーを強制する

Zabbix を構成する場合、セキュリティを強化し、クロスサイトリクエストフォージェリ (CSRF) 攻撃を防ぐために、セッションクッキーにセキュアな属性と SameSite 属性を強制することが不可欠です。ただし、`SameSite=Strict` を強制すると、次のような特定のシナリオで問題が発生する可能性があります。

- 同じドメインの iframe を埋め込むと、ダッシュボード URL ウィジェットに” ユーザーはログインしていません” と表示される。
- HTTPS ではなく HTTP 経由でダッシュボードにアクセスするユーザーは、ログインの問題に直面する可能性がある。
- 特定の Zabbix メニューセクションまたはホストへの URL を共有できない。

これらの問題を軽減するには、ユーザーが SameSite ポリシーを調整する方法が必要です。

1. セキュアなクッキー

`secure` フラグを設定すると、クッキーが HTTPS 経由でのみ送信されるようになり、暗号化されていない接続での露出が防止されます。

Zabbix でセキュアクッキーを有効にするには、Web サーバー構成で次の設定を追加または変更します:

Apache の場合:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Nginx の場合:

```
proxy_cookie_path / "/; Secure";
```

Zabbix フロントエンドが HTTPS 経由でアクセスされていることを確認します。そうでない場合、Secure フラグ付きのクッキーは送信されません。

2. SameSite 属性の構成

Web サーバー設定で SameSite 属性を適用することもできます:

Apache の場合:

```
<IfModule mod_headers.c>
Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Nginx(バージョン 1.19.3+) の場合:

```
proxy_cookie_flags ~ samesite=Strict; # 特異性のために~を'zbx_session'に置き換えます
```

Web サーバーでコンテンツセキュリティポリシー (CSP) を有効にする

Zabbix Web インターフェースをクロスサイトスクリプティング (XSS)、データインジェクション、および同様の攻撃から保護するには、Web サーバーでコンテンツセキュリティポリシーを有効にすることをお勧めします。これを行うには、HTTP ヘッダーを返すように Web サーバーを設定します。

> **Attention:**
> 次の CSP ヘッダー設定は、デフォルトの Zabbix Web インターフェースインストールおよびすべてのコンテンツがサイトのドメイン (サブドメインを除く) から発信されている場合にのみ適用されます。たとえば、サイトのサブドメインまたは外部ドメインのコンテンツを表示するように*URL* ウィジェットを構成している場合、別の CSP ヘッダー構成が必要になる場合があります。*OpenStreetMap* を別のマップエンジンに追加するか、外部 CSS またはウィジェットを追加します。Duo Universal Prompt 多要素認証方式を使用している場合は、仮想ホストの構成ファイルの CSP ディレクティブに”duo.com” を必ず追加してください。

Apache 構成で Zabbix Web インターフェースの CSP を有効にするには、次の手順に従います。

1. 仮想ホストの設定ファイルを見つけます。

- RHEL ベースのシステムの場合、/etc/httpd/conf/httpd.conf
- Debian/Ubuntu の場合、/etc/apache2/sites-available/000-default.conf

2. 次のディレクティブを仮想ホストの設定ファイルに追加します。

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. 変更を適用させるため、Apache サービスを再起動します。

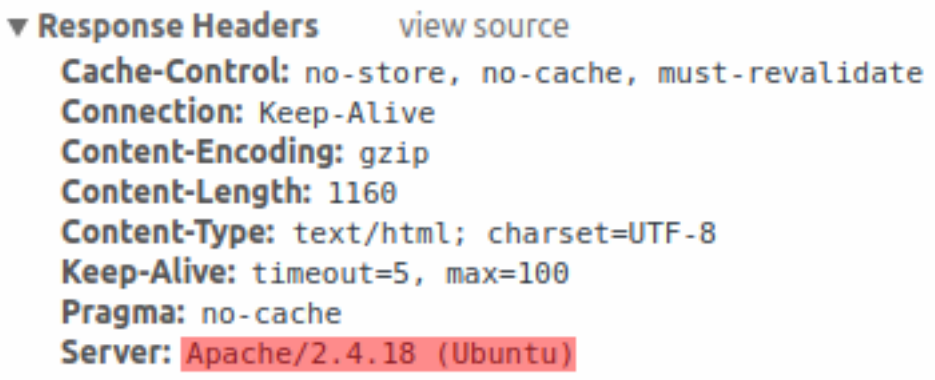
```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Web サーバー情報の公開を無効にする

セキュリティを向上させるために、すべての Web サーバーの署名を無効にすることをお勧めします。

デフォルトでは、Web サーバーはソフトウェア署名を公開します。



署名を無効にするには、Apache 構成ファイルに次のパラメータを追加します。

```
ServerSignature Off
ServerTokens Prod
```

PHP 署名 (X-Powered-By HTTP ヘッダー) は、php.ini 設定ファイルを変更することで無効にできます (デフォルトでは、署名は無効になっています)。

```
expose_php = Off
```

構成ファイルの変更を適用するには、Web サーバーの再起動が必要です。

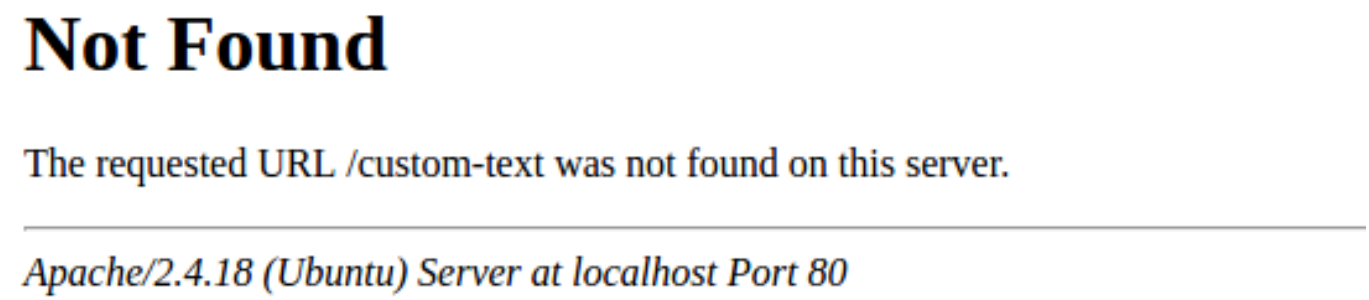
セキュリティを強化するには、Apache で *mod_security* ツール (パッケージ *libapache2-mod-security2*) を使用できます。このツールを使用すると、サーバー署名からバージョンのみを削除するのではなくサーバー署名を削除できます。*mod_security* をインストールした後、"SecServerSignature" を任意の値に設定することで、サーバー署名を任意の値に変更できます。

ソフトウェア署名を削除/変更する方法については、Web サーバーのドキュメントを参照してください。

Web サーバーのデフォルトエラーページを無効にする

情報漏洩を避けるため、デフォルトのエラーページを無効にすることをお勧めします。

デフォルトでは、Web サーバーは組み込みのエラーページを使用します。



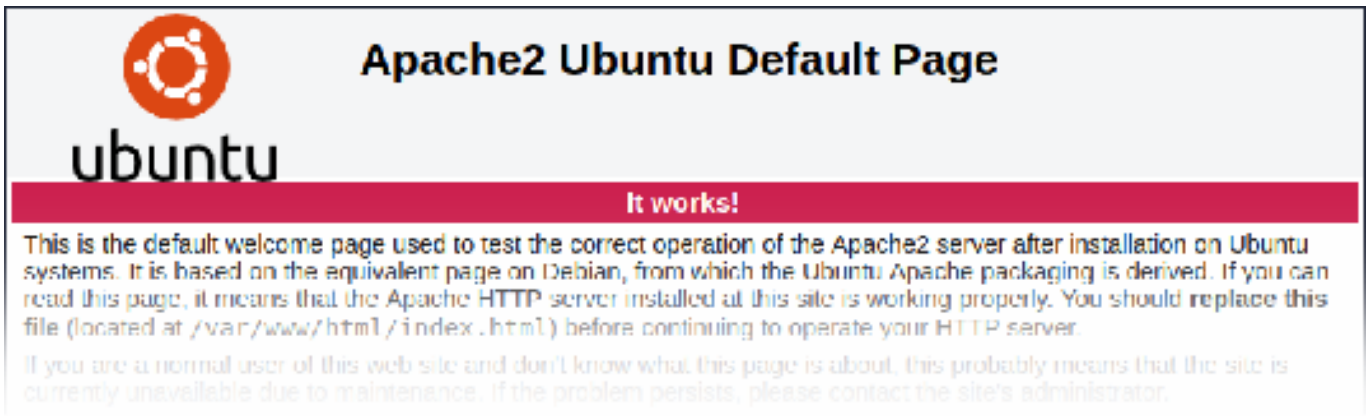
これらのデフォルトのエラーページは置き換えるか削除する必要があります。たとえば、"ErrorDocument" ディレクティブを使用して、Apache Web サーバーのカスタムエラーページ/テキストを定義できます。

デフォルトのエラーページを置き換えたり削除したりする方法については、Web サーバーのドキュメントを参照してください。

Web サーバーのテストページを削除する

情報漏洩を避けるため、Web サーバのテストページを削除することをお勧めします。

By default, the Apache web server webroot contains the `index.html` test page: デフォルトでは、Apache Web サーバーの Webroot には `index.html` テストページが含まれています。



デフォルトのテストページを削除する方法については、Web サーバーのドキュメントを参照してください。

X-Frame-Options HTTP 応答ヘッダーを設定する

デフォルトでは、Zabbix は *Use X-Frame-Options HTTP header* パラメーターが `SAMEORIGIN` に設定されて構成されています。つまり、ページ自体と同じオリジンを持つフレームにのみコンテンツをロードできます。

外部 URL からコンテンツをプルする Zabbix フロントエンド要素 (つまり、URLダッシュボードウィジェット) は、すべてのサンドボックス制限が有効になった状態で、取得したコンテンツをサンドボックスに表示します。

これらの設定により、Zabbix フロントエンドのセキュリティが強化され、XSS およびクリックジャッキング攻撃に対する保護が提供されます。*Super admin* ユーザーは、必要に応じて *Use iframe sandboxing* および *Use X-Frame-Options HTTP header* パラメータを変更できます。デフォルト設定を変更する前に、リスクとメリットを慎重に比較検討してください。iframe サンドボックスまたは X-Frame-Options HTTP ヘッダーを完全にオフにすることはお勧めできません。

一般的なパスワードリストのファイルを非表示にする

パスワード総当たり攻撃の複雑さを増すために、`ui/data/top_passwords.txt` ファイルへのアクセスを制限することをお勧めします。このファイルには、最も一般的でコンテキスト固有のパスワードのリストが含まれており、ユーザーがそのようなパスワードを設定できないようにします (パスワードポリシーで 推測しやすいパスワードを避けるパラメーターが有効になっている場合)。

`top_passwords.txt` ファイルへのアクセスを制限するには、Web サーバーの構成を変更します。

Apache では、`.htaccess` ファイルを使用してファイルアクセスを制限できます。

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

NGINX では、`location` ディレクティブを使用してファイルアクセスを制限できます。

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

2 設定のベストプラクティス

概要

このセクションでは、最適なパフォーマンスと使いやすさを実現するために Zabbix を設定する際のベストプラクティスを概説します。
これらの推奨事項は、Zabbix 開発者の助言と、Zabbix トレーナーおよびサポートエンジニアの実務経験に基づいています。

すべての Zabbix インストール環境は固有であり、これらのガイドラインの一部はお使いの特定の設定には適さない場合があります。ただし、一般的によくある潜在的な問題を回避するため、可能な限りこれらのガイドラインに従うことをお勧めします。

> **Note:**
> このページはさらに改善できると思われる場合は、ぜひご意見をお聞かせください。該当するテキストを選択し、**ctrl+Enter** を押して、誤りの報告やフィードバックの共有を行ってください。

ホストとアイテム　ホストの定義

Zabbix におけるホストは、物理マシンやデバイスではなく、論理的なエンティティです。監視の目的に応じて、データベースや、たとえば仮想マシン用に個別のホストを作成できます。あるいは、*John's laptop* のような汎用的なホストを作成し、そのホスト配下ですべてのメトリクスを監視することもできます。

ベストプラクティスとしては、仮想マシン、データベース、コンテナ、ネットワークスイッチなど、それぞれ独立したインスタンスごとに個別のホストを作成することです。この方法を採用することで、次のことが可能になります。

1. ホストごとに個別のアイテム、トリガー、アラート通知を持たせることで、監視データの煩雑化を防ぐことができます。
2. ユーザーアクセスレベルをきめ細かく調整できます。user-roles を設定して、特定のホストの表示および/または設定のみにアクセスを許可できます。あわせて、最小権限の原則 も参照してください。

重複したアイテムを持つホスト

Network switch 1 や *Network switch 2* のように、類似したホストが複数ある場合、Zabbix にはホストをすばやく再作成するための複数の方法が用意されています。
Clone ボタンを押すだけで、すべてのメトリクスを含めてホストを簡単に複製できますが、この場合、後でアイテムを更新する必要が生じた際には、各ホストで手動で更新しなければなりません。

ベストプラクティスは、必要なすべてのメトリクスを含むテンプレートを作成することです。たとえば、*Network switch template* です。

次に、類似したホストをホストグループにまとめます。上記の例では、*Network switches* などにできます。
これで、データ収集 -> ホストセクションでホストグループによってすべてのホストをフィルタリングし、*Mass update* ボタンを使用して、すべてのネットワークスイッチにテンプレートをリンクできます。

依存アイテム

対象エンティティへのリクエスト数を最小限に抑えるため、Zabbix ではマスターアイテムと依存アイテムを作成できます。
この場合、マスターアイテムは 1 回のリクエストで大量の情報を収集します。
その後、依存アイテムを設定して、前処理によってその収集結果から特定のデータを抽出し、個別のメトリクスとして保存できます。

例えば、マスターアイテムは複数のメトリクスを含む JSON または XML レスポンスを収集したり、複数列のデータを返すデータベースクエリ（例: オープン接続数、中断された接続数、同時接続の最大許容数、起動以降の累積総接続数）を実行したりできます。依存アイテムは、必要な各値を個別に解析して保存します。

この構成におけるベストプラクティスは、収集直後にマスターアイテムの履歴を破棄し、依存アイテムのデータのみを保持することです。

サーバーとプロキシ

すべてのホストが Zabbix サーバーと同じローカルネットワーク内にあり、スケーラビリティやパフォーマンスに関する懸念がない場合は、プロキシは不要なことがあります。より大規模または複雑な環境では、Zabbix サーバーでホストを直接監視するだけでは不十分な場合があります。プロキシを追加し、ホストの一部をそのプロキシに割り当てることで、負荷をより均等に分散できます。

ベストプラクティスとして、次の場合は Zabbix プロキシを追加します。

1. ファイアウォールの内側にある複数のホストを、さまざまなメトリクス収集方法を使用して監視している場合。プロキシがホストからデータを収集して Zabbix サーバーに転送するため、複数のファイアウォールポートを開放する必要性を減らせます。
2. リモート拠点、支店、またはネットワークを監視している場合。Zabbix サーバーとリモート拠点の間でネットワーク障害が発生した場合でも、リモート拠点に配置された Zabbix プロキシは引き続きデータを収集し、ネットワーク接続が復旧すると収集済みデータを Zabbix サーバーに送信します。
3. 大規模な導入環境で、Zabbix サーバーの負荷を軽減し、パフォーマンスを向上させたい場合。大規模な導入環境の定義は非常に幅広く、ホスト数だけでなく、1 秒あたりに収集される値の数にも依存します。

シークレットマクロ

secret ユーザーマクロは、シークレットテキストまたはシークレット vault マクロとして使用できます。

シークレット vault マクロを使用する際のセキュリティを強化するには、マクロ値が Zabbix サーバーおよび Zabbix プロキシによって個別に取得されるように設定することを推奨します。デフォルトでは、シークレットマクロの値は Zabbix サーバーによって取得され、Zabbix プロキシに伝播されます。

20. API

概要　Zabbix API を使用すると、Zabbix の設定をプログラムから取得および変更でき、履歴データにもアクセスできます。主に次の用途で広く使用されています。

- Zabbix と連携する新しいアプリケーションの作成
- Zabbix のサードパーティ製ソフトウェアへの統合
- 定型作業の自動化

Zabbix API は HTTP ベースの API であり、Web インターフェースの一部として提供されています。この API は JSON-RPC 2.0 プロトコルを使用しており、これは次の 2 つを意味します。

- API は個別のメソッドの集合で構成されている
- クライアントと API 間のリクエストおよびレスポンスは JSON 形式でエンコードされる

プロトコルおよび JSON の詳細については、JSON-RPC 2.0 specification および JSON format homepage を参照してください。

Zabbix の機能を Python アプリケーションに統合する方法の詳細については、Python library for Zabbix を参照してください。

> **Note:**
> 設定と履歴データの両方を含む Zabbix でのユーザーアクセスは、ユーザータイプ、割り当てられたユーザーロール、およびユーザーグループ に依存します。

構造　API は、名目上、別々の API にグループ化された多数のメソッドから構成されています。それぞれのメソッドはある特定のタスクを実行します。例えば、host.create メソッドは *host* API に属し、新しいホストを作成に使用されます。歴史的には、API は" クラス" と呼ばれることもあります。

> **Note:**
> ほとんどの API は、少なくとも get, create, update, delete という 4 つのメソッドを持ち、それぞれデータの取得、作成、更新、削除を行います。しかし、一部の API では、全く別のメソッドを提供することもあります。

リクエストの実行　Web インターフェースの設定が完了したら、リモート HTTP リクエストを使用して API を呼び出すことができます。そのためには、Web インターフェースのディレクトリにある api_jsonrpc.php ファイルに対して HTTP POST リクエストを送信する必要があります。例えば、Zabbix の Web インターフェースが https://example.com/zabbix にインストールされている場合、apiinfo.version メソッドを呼び出す HTTP リクエストは次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"apiinfo.version","params":{},"id":1}'
```

リクエストには、Content-Type ヘッダーを次のいずれかの値に設定する必要があります: application/json-rpc、application/json、または application/jsonrequest。

リクエストオブジェクトには、次のプロパティを含める必要があります。

- jsonrpc - API で使用される JSON-RPC プロトコルのバージョン（Zabbix API は JSON-RPC バージョン 2.0 を実装しています）。
- method - 呼び出される API メソッド。
- params - API メソッドに渡されるパラメータ。
- id - リクエストの任意の識別子（省略した場合、API はそのリクエストを通知として扱います）。

リクエストが正しい場合、API から返されるレスポンスは次のようになります。

```
{
    "jsonrpc": "2.0",
    "result": "7.4.0",
    "id": 1
}
```

レスポンスオブジェクトには、次のプロパティが含まれます。

- jsonrpc - JSON-RPC プロトコルのバージョン。
- result - メソッドによって返されるデータ。
- id - 対応するリクエストの識別子。

ワークフローの例　次のセクションでは、いくつかの使用例を詳しく説明します。

認証　Zabbix のデータにアクセスするには、次のどちらかを行う必要があります。

- 既存のAPI トークンを使用 (Zabbix フロントエンドまたはトークン APIを使用して作成)
- user.loginメソッドで取得した認証トークンを使用。

たとえば、標準の *Admin* ユーザーとしてログインして新しい認証トークンを取得したい場合、JSON リクエストは次のようになります。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

資格情報を正しく指定した場合、API から返されるレスポンスにはユーザー認証トークンが含まれているはずです。

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

認証方法　”Authorization” ヘッダーを使用する場合

すべての API リクエストには、認証情報または API トークンが必要です。
リクエスト内で Authorization ヘッダーを使用して認証情報を指定できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
認証に関する問題が発生している場合は、Authorization header forwardingを参照してください。

Zabbix API は、ヘッダー名の大文字と小文字を区別せずに受け付けます（例: `authorization`、`Authorization`、`AUTHORIZATION` は同じものとして扱われます）。

Authorization ヘッダーは、クロスオリジンリクエスト（CORS）でサポートされています。

Zabbix クッキー

”zbx_session” クッキーは、(モジュールまたはカスタムウィジェットから) JavaScript を使用して実行される Zabbix UI からの API リクエストを承認するために使用されます。

ホストの取得　これで、有効なユーザー認証トークン（以下の例では変数として表されています）が得られたため、これを使用して Zabbix 内のデータにアクセスできます。例えば、host.get メソッドを使用して、設定済みのすべてのホストの ID、ホスト名、インターフェースを取得できます。

リクエスト:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data @data.json
```

Note:
`data.json` は JSON クエリを含むファイルです。ファイルの代わりに、`--data` 引数でクエリを渡すこともできます。

data.json

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

レスポンスオブジェクトには、要求されたホストに関するデータが含まれます。

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
```

```
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
パフォーマンス上の理由から、取得したいオブジェクトのプロパティは常に明示的に指定することを推奨します。これにより、すべてのデータを取得してしまうことを避けられます。

新しいアイテムの作成　先ほどの host.get リクエストで取得したデータを使って、"Zabbix サーバー" に新しいアイテムを作成してみましょう。これはitem.createメソッドを使用して実行できます。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

作成が成功した場合のレスポンスには、新しく作成されたアイテムの ID が含まれます。これは、以降のリクエストでアイテムを参照するために使用できます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
なお item.create メソッドや他の *create* メソッドも、オブジェクトの配列を受け入れて、1 回の API 呼び出しで複数のアイテムを作成することもできます。

Creating multiple triggers　したがって、*create* メソッドが配列を受け入れる場合、次のように複数のトリガーを追加することも可能です。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

作成が成功した場合のレスポンスには、新しく作成したトリガーの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

アイテムの更新　アイテムのステータスを"0" に設定することで、アイテムを有効にします。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
```

```
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

成功したレスポンスには、更新されたアイテムの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
item.update メソッドやその他の *update* メソッドは、オブジェクトの配列を受け入れ、1 回の API 呼び出しで複数のアイテムを更新することができます。

複数のトリガーの更新　ステータスを"0" に設定して、複数のトリガーを有効にします。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.update","params":[{"triggerid":"13938","status":0},{"triggeri
```

成功したレスポンスには、更新されたトリガーの ID が含まれます。

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Note:
これは推奨される更新方法です。host.massupdate などの一部の API メソッドを使用すると、より単純なコードを作成できます。ただし、これらのメソッドは将来のリリースで削除されるため、使用することはお勧めできません。

エラーハンドリング　今まで試したメソッドはすべて成功しました。しかし、API に対して間違った呼び出しを行おうとするとどうなるでしょうか？host.create を呼び出して別のホストを作成する際に、必須の groups パラメータを省略してみましょう。

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

レスポンスにはエラーメッセージが含まれます。

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
```

```
    "id": 7
}
```

エラーが発生した場合、レスポンスオブジェクトには `result` プロパティの代わりに、次のデータを持つ `error` プロパティが含まれます。

- `code` - エラーコード
- `message` - エラーの概要
- `data` - 詳細なエラーメッセージ

エラーは、不正な入力値の使用、セッションタイムアウト、存在しないオブジェクトへのアクセスの試行など、さまざまなケースで発生する可能性があります。アプリケーションは、この種のエラーを適切に処理できる必要があります。

API バージョン　API のバージョン管理を簡素化するために、Zabbix 2.0.4 以降、API のバージョンは Zabbix 自体のバージョンと一致していますが、apiinfo.versionメソッドを使用して、使用している API のバージョンを確認することもできます。これは、バージョン固有の機能を使用するようにアプリケーションを調整する場合に役立ちます。

Zabbix は、メジャーバージョン内では機能の下位互換性を保証します。メジャーリリース間で下位互換性のない変更を行う場合、通常は古い機能を次のリリースで非推奨として残し、その後のリリースで削除します。場合によっては、下位互換性を提供せずにメジャーリリース間で機能を削除することもあります。非推奨の機能には決して依存せず、できるだけ早く新しい代替機能に移行することが重要です。

> **Note:**
> API 変更ログ で、API に加えられたすべての変更を追うことができます。

参考文献　これで Zabbix API を使い始めるのに十分な知識が得られましたが、これが全てではありません。更に詳しく知りたい場合は、利用可能な API のリストを参照することをお勧めします。

メソッドリファレンス

このセクションでは、Zabbix API が提供する機能の概要を説明し、利用可能なクラスとメソッドの使い方を説明します。

監視　Zabbix API では、監視中に収集した履歴やその他のデータにアクセスすることができます。

ダッシュボード

ダッシュボードを管理し、それに基づいてスケジュールされたレポートを作成します。

Dashboard API | Template dashboard API | Report API

高可用性 (HA) クラスタ

サーバノードとそのステータスの一覧を取得します。

High availability cluster API

ヒストリ

Zabbix 監視プロセスによって収集された過去の値を取得し、表示やさらなる処理を行うことができます。

History API

トレンド

Zabbix サーバーで計算されたトレンドの値を取得し、表示やさらなる処理を行うことができます。

Trend API

イベント

トリガー、ネットワークディスカバリ、その他の Zabbix システムから生成されたイベントを取得し、より柔軟な状況管理やサードパーティツールの統合を行うことができます。

Event API

障害

指定されたパラメータに従って、障害を取得します。

Problem API

マップ

マップを設定して、IT インフラストラクチャの詳細な動的表示を作成します。

Map API

タスク

Zabbix サーバーのタスクマネージャと連動し、タスクの作成とレスポンスの取得を行います。

Task API

サービス　Zabbix API を使用すると、サービスの監視中に収集されたデータにアクセスできます。

サービスレベルアグリーメント (SLA)

サービスレベル目標（SLO）を定義し、サービスのパフォーマンスに関する詳細なサービスレベル指標（SLI）情報を取得します。

SLA API

サービス

サービスレベルのモニタリングのためにサービスを管理し、あらゆるサービスの詳細な SLA 情報を取得することができます。

Service API

データ収集　Zabbix API では、監視システムの設定を管理することができます。

ホストとホストグループ

ホストグループとホスト、およびホストインタフェース、ホストマクロ、メンテナンス期間など、ホストに関連するすべてのものを管理します。

Host API | Host group API | Host interface API | User macro API | Value map API | Maintenance API

アイテム

監視するアイテムを定義します

Item API

トリガー

システムの問題を通知するトリガーを構成します。トリガーの依存関係を管理します。

Trigger API

グラフ

収集したデータを見やすくするために、グラフを編集したり、グラフの item を分けたりすることができます。

Graph API | Graph item API

テンプレートとテンプレートグループ

テンプレートを管理し、ホストまたは他のテンプレートにリンクします。

Template API | Template group API | Value map API

ローレベルディスカバリ

動的なエンティティを監視するために、ローレベルディスカバリルール、item、trigger、グラフのプロトタイプを設定します。

LLD rule API | Item prototype API | Trigger prototype API | Graph prototype API | Host prototype API

イベント相関

カスタムイベントの相関ルールを作成します。

Correlation API

ネットワークディスカバリ

ネットワークレベルの検出ルールを管理し、新しいホストを自動的に検出して監視します。検出されたサービスとホストに関する情報に完全にアクセスできます。

Discovery rule API | Discovery check API | Discovered host API | Discovered service API

エクスポートとインポート

設定のバックアップ、マイグレーション、大規模な設定更新のために Zabbix の設定データをエクスポート、インポートすることができます。

Configuration API

Web 監視

Web アプリケーションやサービスを監視するための Web シナリオを設定します。

Web scenario API

アラート　Zabbix API を使用すると、監視システムのアクションとアラートを管理できます。

アクションとアラート

特定のイベントをユーザーに通知したり、リモートコマンドを自動的に実行するためのアクションと操作を定義します。生成されたアラートとその受信者に関する情報にアクセスできます。

Action API | Alert API

メディアタイプ

メディアの種類と、ユーザーがアラートを受け取る複数の方法を構成します。

Media type API

スクリプト

日々の作業を支援するスクリプトを設定・実行します。

Script API

ユーザー　Zabbix API を使用すると、監視システムのユーザーを管理できます。

ユーザーとユーザーグループ

Zabbix にアクセスするユーザを追加し、ユーザグループに割り当て、権限を付与します。ユーザーの権限を細かく管理するための役割を作成します。

User API | User group API | User directory API | User role API

API トークン

認可トークンを管理します。

Token API

認証

認証の設定オプションを変更します。

Authentication API

管理画面　Zabbix API を使用すると、監視システムの管理設定を変更することができます。

一般

特定のグローバル設定オプションを変更します。

Autoregistration API | Icon map API | Image API | Settings API | Regular expression API | Module API | Connector API

監査ログ

各ユーザーが行った設定の変更を追跡します。

Audit log API

ハウスキーピング

ハウスキーピングの設定をします。

Housekeeping API

プロキシとプロキシグループ

分散監視設定で使用される Zabbix プロキシを管理します。

Proxy API | Proxy group API

マクロ

マクロを管理します。

User macro API

API 情報　Zabbix API のバージョンを取得し、アプリケーションでバージョン固有の機能を使用できるようにします。

API info API

Alert

このクラスは、アラートを操作するためのものです。

オブジェクトリファレンス:

- Alert

利用可能なメソッド:

- alert.get - アラートを取得する

Alert オブジェクト

次のオブジェクトは、`alert`API に直接関連しています。

アラート

> **Note:**
> アラートは Zabbix サーバーによって作成され、API 経由で変更することはできません。

alert オブジェクトには、特定のアクション操作が正常に実行されたかどうかに関する情報が含まれます。次のプロパティがあります。

プロパティ	型	説明
alertid	ID	アラートの ID。
actionid	ID	アラートを生成したアクションの ID。
alerttype	integer	アラートのタイプ。 使用可能な値: 0 - メッセージ; 1 - リモートコマンド。
clock	timestamp	アラートが生成された時刻。
error	string	メッセージの送信またはコマンドの実行で問題が発生した場合のエラーテキスト。
esc_step	integer	アラートが生成されたアクションのエスカレーションステップ。
eventid	ID	アクションをトリガーしたイベントの ID。
mediatypeid	ID	メッセージの送信に使用されたメディアタイプの ID。
message	text	メッセージ本文。 プロパティの動作: - `alerttype` が”message” に設定されている場合に *supported*
retries	integer	Zabbix がメッセージの送信を試行した回数。
sendto	string	受信者のアドレス、ユーザー名、またはその他の識別子。 プロパティの動作: - `alerttype` が”message” に設定されている場合に *supported*

プロパティ	型	説明
status	integer	アクション操作が正常に実行されたかどうかを示すステータス。 `alerttype` が”message” に設定されている場合の使用可能な値: 0 - メッセージ未送信; 1 - メッセージ送信済み; 2 - 複数回の再試行後に失敗; 3 - 新しいアラートはまだアラートマネージャーによって処理されていません。 `alerttype` が”remote command” に設定されている場合の使用可能な値: 0 - コマンド未実行; 1 - コマンド実行済み; 2 - Zabbix エージェント上でコマンドの実行を試みましたが、利用できませんでした。
subject	string	メッセージ件名。 プロパティの動作: - `alerttype` が”message” に設定されている場合に *supported*
userid	ID	メッセージの送信先ユーザーの ID。
p_eventid	ID	アラートを生成した障害イベントの ID。
acknowledgeid	ID	アラートを生成した確認応答の ID。

alert.get

説明

`integer/array alert.get(object parameters)`

このメソッドは、指定されたパラメーターに従ってアラートを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
alertids	ID/array	指定した ID のアラートのみを返します。
actionids	ID/array	指定したアクションによって生成されたアラートのみを返します。
eventids	ID/array	指定したイベントによって生成されたアラートのみを返します。
groupids	ID/array	指定したホストグループ内のオブジェクトによって生成されたアラートのみを返します。
hostids	ID/array	指定したホスト上のオブジェクトによって生成されたアラートのみを返します。
mediatypeids	ID/array	指定したメディアタイプを使用したメッセージアラートのみを返します。
objectids	ID/array	指定したオブジェクトによって生成されたアラートのみを返します。
userids	ID/array	指定したユーザーに送信されたメッセージアラートのみを返します。
eventobject	integer	指定したタイプのオブジェクトに関連するイベントによって生成されたアラートのみを返します。 サポートされているオブジェクトタイプの一覧は、event の`object`を参照してください。 デフォルト: 0 - トリガー。

Parameter	Type	Description
eventsource	integer	指定したタイプのイベントによって生成されたアラートのみを返します。 サポートされているイベントタイプの一覧は、event のsourceを参照してください。 デフォルト: 0 - トリガーイベント。
time_from	timestamp	指定した時刻以降に生成されたアラートのみを返します。
time_till	timestamp	指定した時刻以前に生成されたアラートのみを返します。
selectHosts	query	アクションの実行をトリガーしたホストのデータを含むhostsプロパティを返します。
selectMediatypes	query	メッセージアラートに使用されたメディアタイプの配列を含むmediatypesプロパティを返します。 ユーザータイプに基づく制限については、mediatype.getを参照してください。
selectUsers	query	メッセージの送信先ユーザーの配列を含むusersプロパティを返します。 ユーザータイプに基づく制限については、user.getを参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: alertid, clock, eventid, mediatypeid, sendto, status。
countOutput	boolean	これらのパラメータについては、reference commentaryで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

アクション ID でアラートを取得する

アクション"3" によって生成されたすべてのアラートを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLEM: Zabbix agent on Linux server is unreachable for 5 minutes: ",
            "message": "Trigger: Zabbix agent on Linux server is unreachable for 5 minutes: \nTrigger stat
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- メディアタイプ
- ユーザー

ソース

CAlert::get() in *ui/include/classes/api/services/CAlert.php*.

API info

このクラスは、API に関するメタ情報を取得するためのものです。

利用可能なメソッド:

- apiinfo.version - Zabbix API のバージョンの取得

apiinfo.version

説明

`string apiinfo.version(array)`

このメソッドは、Zabbix API のバージョンを取得することができます。

> **Attention:**
> このメソッドは認証されていないユーザーのみが使用できます。

パラメーター

(array) このメソッドは空の配列を受け入れます。

戻り値

(string) Zabbix API のバージョンを返します。

> **Note:**
> Zabbix 2.0.4 以降、API のバージョンは Zabbix のバージョンと一致します。

例

API のバージョンの取得

Zabbix API のバージョンを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "7.4.0",
    "id": 1
}
```

ソース

CAPIInfo::version() in *ui/include/classes/api/services/CAPIInfo.php*.

Authentication

このクラスは、認証設定を操作するためのものです。

オブジェクトリファレンス:

- Authentication

利用可能なメソッド:

- authentication.get - 認証を取得する
- authentication.update - 認証を更新する

Authentication オブジェクト

次のオブジェクトは、`authentication`API に直接関連しています。

認証

認証オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
authentication_type	integer	デフォルトの認証。 使用可能な値: 0 - *(デフォルト)* 内部; 1 - LDAP。
http_auth_enabled	integer	HTTP 認証。 使用可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。 プロパティの動作: - Web インターフェース設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合に *supported*。

プロパティ	Type	説明
http_login_form	integer	デフォルトのログインフォーム。 使用可能な値: 0 - *(デフォルト)* Zabbix ログインフォーム; 1 - HTTP ログインフォーム。 プロパティの動作: - Web インターフェース設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合に *supported*。
http_strip_domains	string	削除するドメイン名。 プロパティの動作: - Web インターフェース設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合に *supported*。
http_case_sensitive	integer	HTTP の大文字・小文字を区別するログイン。 使用可能な値: 0 - オフ; 1 - *(デフォルト)* オン。 プロパティの動作: - Web インターフェース設定ファイル (*zabbix.conf.php*) で `$ALLOW_HTTP_AUTH` が有効な場合に *supported*。
ldap_auth_enabled	integer	LDAP 認証。 使用可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
ldap_case_sensitive	integer	LDAP の大文字・小文字を区別するログイン。 使用可能な値: 0 - オフ; 1 - *(デフォルト)* オン。
ldap_userdirectoryid	ID	LDAP 認証用のデフォルトユーザーディレクトリの ID。 `gui_access` が LDAP または System default に設定されているユーザーグループで使用されます。 プロパティの動作: - `ldap_auth_enabled` が”Enabled” に設定されている場合は *required*
saml_auth_enabled	integer	SAML 認証。 使用可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
saml_case_sensitive	integer	SAML の大文字・小文字を区別するログイン。 使用可能な値: 0 - オフ; 1 - *(デフォルト)* オン。
passwd_min_length	integer	パスワードの最小長要件。 使用可能な値の範囲は 1 から 70 です。 デフォルト: 8。

プロパティ	Type	説明
passwd_check_rules	integer	パスワードチェックルール。 使用可能なビットマップ値: 0 - パスワード長をチェック; 1 - パスワードに大文字と小文字のラテン文字が使用されているかをチェック; 2 - パスワードに数字が使用されているかをチェック; 4 - パスワードに特殊文字が使用されているかをチェック; 8 - *(デフォルト)* パスワードがよく使われるパスワードの一覧に含まれておらず、かつ”Zabbix” という単語やユーザーの名、姓、またはユーザー名の派生形を含まないことをチェック。 これはビットマスクフィールドです。使用可能なビットマップ値の任意の合計を指定できます（たとえば、すべてのルールをチェックする場合は 15）。
ldap_jit_status	integer	LDAP プロビジョニングのステータス。 使用可能な値: 0 - 設定済みの LDAP IdP に対して無効; 1 - 設定済みの LDAP IdP に対して有効。
saml_jit_status	integer	SAML プロビジョニングのステータス。 使用可能な値: 0 - 設定済みの SAML IdP に対して無効; 1 - 設定済みの SAML IdP に対して有効。
jit_provision_interval	string	ログイン済みユーザーに対する JIT プロビジョン要求の間隔。 秒、および月と年をサポートするサフィックス付き時間単位を受け付けます (3600s,60m,1h,1d,1M,1y)。最小値: 1h。 デフォルト: 1h。 LDAP プロビジョニングでのみ使用できます。
disabled_usrgrpid	ID	プロビジョニング解除されたユーザーを割り当てるユーザーグループの ID。 このユーザーグループは無効である必要があり、設定されている間は有効化または削除できません。 プロパティの動作: - `ldap_jit_status` が”Enabled for configured LDAP IdPs” に設定されている場合、または `saml_jit_status` が”Enabled for configured SAML IdPs” に設定されている場合は *required*
mfa_status	integer	多要素認証。 使用可能な値: 0 - 無効（設定済みのすべての MFA 方式に対して）; 1 - 有効（設定済みのすべての MFA 方式に対して）。
mfaid	ID	MFA が有効なユーザーグループのデフォルトのMFA 方式。 プロパティの動作: - `mfa_status` が”Enabled” に設定されている場合は *required*

authentication.get

説明

`object authentication.get(object parameters)`

このメソッドは、指定されたパラメーターに従って認証オブジェクトを取得することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	このパラメータについては、リファレンス解説で説明されています。

戻り値

(object) 認証オブジェクトを返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "authentication_type": "0",
        "http_auth_enabled": "0",
        "http_login_form": "0",
        "http_strip_domains": "",
        "http_case_sensitive": "1",
        "ldap_auth_enabled": "0",
        "ldap_case_sensitive": "1",
        "ldap_userdirectoryid": "0",
        "saml_auth_enabled": "0",
        "saml_case_sensitive": "0",
        "passwd_min_length": "8",
        "passwd_check_rules": "15",
        "jit_provision_interval": "1h",
        "saml_jit_status": "0",
        "ldap_jit_status": "0",
        "disabled_usrgrpid": "9",
        "mfa_status": "0",
        "mfaid": "0"
    },
    "id": 1
}
```

ソース

CAuthentication::get() in *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

説明

`object authentication.update(object authentication)`

このメソッドは、既存の認証設定を更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 更新する認証のプロパティ。

戻り値

(array) 更新したパラメーターの名前を配列で返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

ソース

CAuthentication::update() in *ui/include/classes/api/services/CAuthentication.php*.

Autoregistration

このクラスは、自動登録を操作するためのものです。

オブジェクトリファレンス:

- Autoregistration

利用可能なメソッド:

- autoregistration.get - 自動登録を取得する
- autoregistration.update - 自動登録を更新する

Autoregistration オブジェクト

次のオブジェクトは、autoregistrationAPI に直接関連しています。

自動登録

自動登録オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tls_accept	integer	自動登録で許可される受信接続の種類。 指定可能な値: 1 - 暗号化されていない接続を許可; 2 - PSK を使用した TLS を許可; 3 - 暗号化されていない接続と PSK を使用した TLS 接続の両方を許可。
tls_psk_identity	string	PSK ID。1 つの PSK に対して 1 つだけ対応付ける必要があります (autoregistration、hosts、proxies 全体で一意)。 受信側にどの PSK を使用するかを通知するため、PSK ID はネットワーク上で暗号化されずに送信されます。そのため、PSK ID には機密情報を含めないでください。 Property behavior: - *write-only*
tls_psk	string	事前共有鍵 (PSK)。32 桁以上の 16 進数である必要があります。 Property behavior: - *write-only*

autoregistration.get

説明

`object autoregistration.get(object parameters)`

このメソッドは、指定されたパラメーターに従って自動登録オブジェクトを取得することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータ。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	型	説明
output	query	このパラメータはリファレンスコメントで説明されています。

戻り値

(`object`) autoregistration オブジェクトを返します。

例

リクエスト:

```
{
   "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "tls_accept": "3"
    },
    "id": 1
}
```

ソース

CAutoregistration::get() in *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

説明

`object autoregistration.update(object autoregistration)`

このメソッドは、既存の自動登録を更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(object) 更新する自動登録のプロパティ。

戻り値

(boolean) 更新が成功すると、ブール値 true を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

ソース

CAutoregistration::update() in *ui/include/classes/api/services/CAutoregistration.php*.

Configuration

このクラスは、Zabbix の設定データのエクスポートとインポートを行うためのものです。

利用可能なメソッド:

- configuration.export - 設定のエクスポート
- configuration.import - 設定のインポート
- configuration.importcompare - インポートファイルと現在のシステムの要素の比較

configuration.export

説明

`string configuration.export(object parameters)`

このメソッドは、構成データをシリアル化された文字列としてエクスポートできます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) エクスポートするオブジェクトと使用する形式を定義するパラメータです。

パラメータ	Type	説明
format	string	データをエクスポートする形式です。 使用可能な値: yaml - YAML; xml - XML; json - JSON; raw - 未処理の PHP 配列。 パラメータの動作: - 必須
prettyprint	boolean	インデントを追加して、出力を人間にとって読みやすくします。 使用可能な値: true - インデントを追加する; false - *(デフォルト)* インデントを追加しない。
options	object	エクスポートするオブジェクトです。 options オブジェクトには次のパラメータがあります: host_groups - (array) エクスポートするホストグループの ID; hosts - (array) エクスポートするホストの ID; images - (array) エクスポートするイメージの ID; maps - (array) エクスポートするマップの ID; mediaTypes - (array) エクスポートするメディアタイプの ID; template_groups - (array) エクスポートするテンプレートグループの ID; templates - (array) エクスポートするテンプレートの ID。 *Admin* および *User* タイプのユーザーは、*read-only* または *read-write* の権限を持つオブジェクト、およびイメージのみをエクスポートできますが、メディアタイプはエクスポートできません。 パラメータの動作: - 必須

戻り値

(string) 要求された構成データを含むシリアル化された文字列を返します。

例

テンプレートのエクスポート

テンプレート「10571」の設定を XML 文字列としてエクスポートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>7.4</version><template_
    "id": 1
}
```

ソース

CConfiguration::export() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

説明

`boolean configuration.import(object parameters)`

このメソッドは、シリアル化された文字列から設定データをインポートすることができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) インポートするデータと、そのデータをどのように処理するかのルールを含むパラメータです。

Parameter	Type	Description
format	string	シリアライズされた文字列の形式。 指定可能な値: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameter behavior: - *required*
source	string	設定データを含むシリアライズされた文字列。 Parameter behavior: - *required*

Parameter	Type	Description
rules	object	新規および既存のオブジェクトをどのようにインポートするかのルール。 *Admin* タイプのユーザーは、自身が *read-write* permission を持つオブジェクトと、マップのみをインポートできます。たとえば、ホストおよびそのエンティティ（アイテム、トリガー、グラフなど）は、インポートされるホストが所属するホストグループに対して、そのユーザーのユーザーグループが権限を持っている場合にのみインポートできます。画像およびメディアタイプは、*Admin* タイプのユーザーではインポートできません。 `rules` パラメータの詳細は、以下の表で説明しています。 Parameter behavior: - *required*

Attention:
ルールが指定されていない場合、設定は更新されません。

`rules` オブジェクトは、以下のパラメータをサポートしています。

Parameter	Type	Description
discoveryRules	object	LLD ルールをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しい LLD ルールが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存の LLD ルールが更新されます。デフォルト: `false`; `deleteMissing` - (`boolean`) `true` に設定すると、インポートデータに存在しない LLD ルールがデータベースから削除されます。デフォルト: `false`.
graphs	object	グラフをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しいグラフが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のグラフが更新されます。デフォルト: `false`; `deleteMissing` - (`boolean`) `true` に設定すると、インポートデータに存在しないグラフがデータベースから削除されます。デフォルト: `false`.
host_groups	object	ホストグループをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しいホストグループが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のホストグループが更新されます。デフォルト: `false`.
template_groups	object	テンプレートグループをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しいテンプレートグループが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のテンプレートグループが更新されます。デフォルト: `false`.

Parameter	Type	Description
hosts	object	ホストをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいホストが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のホストが更新されます。デフォルト: false.
httptests	object	Web シナリオをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しい Web シナリオが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の Web シナリオが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しない Web シナリオがデータベースから削除されます。デフォルト: false.
images	object	画像をどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しい画像が作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の画像が更新されます。デフォルト: false.
items	object	アイテムをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいアイテムが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のアイテムが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しないアイテムがデータベースから削除されます。デフォルト: false.
maps	object	マップをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいマップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のマップが更新されます。デフォルト: false.
mediaTypes	object	メディアタイプをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいメディアタイプが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のメディアタイプが更新されます。デフォルト: false.
templateLinkage	object	テンプレートリンクをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされていないが、インポートデータ内には存在するテンプレートがリンクされます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされているが、インポートデータ内には存在しないテンプレートのリンクが解除されます。このとき、リンク解除されたテンプレートから継承されたエンティティ（アイテム、トリガーなど）は削除されません。デフォルト: false.

Parameter	Type	Description
templates	object	テンプレートをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいテンプレートが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートが更新されます。デフォルト: false.
templateDashboards	object	テンプレートダッシュボードをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいテンプレートダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートダッシュボードが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しないテンプレートダッシュボードがデータベースから削除されます。デフォルト: false.
triggers	object	トリガーをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいトリガーが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のトリガーが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しないトリガーがデータベースから削除されます。デフォルト: false.
valueMaps	object	ホストまたはテンプレートの値マップをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しい値マップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の値マップが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しない値マップがデータベースから削除されます。デフォルト: false.

戻り値

(boolean) インポートに成功した場合は true を返します。

例

テンプレートのインポート

XML 文字列に含まれるテンプレート設定をインポートします。XML 文字列内のアイテムまたはトリガーが不足している場合、それらはデータベースから削除され、それ以外は変更されません。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.import",
    "params": {
        "format": "xml",
        "rules": {
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "items": {
                "createMissing": true,
```

```
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": false
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>7.4</version><templ
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```


ソース

CConfiguration::import() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

説明

`array configuration.importcompare(object parameters)`

このメソッドは、インポートファイルを現在のシステム要素と比較し、このインポートファイルがインポートされる場合に何が変更されるかを表示することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) インポート可能なデータと、そのデータをどのように処理するかのルールを含むパラメータです。

パラメータ	型	説明
format	string	シリアライズされた文字列の形式。 指定可能な値: `yaml` - YAML; `xml` - XML; `json` - JSON. パラメータの動作: - 必須
source	string	設定データを含むシリアライズされた文字列。 パラメータの動作: - 必須

パラメータ	型	説明
rules	object	新規オブジェクトおよび既存オブジェクトをどのように比較するかのルール。 *Admin* および *User* タイプのユーザーは、新規オブジェクトを、*read-only* または *read-write* の権限を持つ既存オブジェクトとのみ比較できます。 `rules` パラメータの詳細は、以下の表で説明されています。 パラメータの動作: - 必須

Attention:
ルールが指定されていない場合、更新対象はなく、結果は空になります。

Note:
比較はホストグループとテンプレートに対してのみ実行されます。
トリガーとグラフは、インポートされたテンプレートに対してのみ比較され、それ以外はすべて「新規」と見なされます。

`rules` オブジェクトは以下のパラメータをサポートしています。

パラメータ	型	説明
discoveryRules	object	LLD ルールをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しい LLD ルールが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存の LLD ルールが更新されます。デフォルト: `false`; `deleteMissing` - (`boolean`) `true` に設定すると、インポートデータに存在しない LLD ルールがデータベースから削除されます。デフォルト: `false`.
graphs	object	グラフをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しいグラフが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のグラフが更新されます。デフォルト: `false`; `deleteMissing` - (`boolean`) `true` に設定すると、インポートデータに存在しないグラフがデータベースから削除されます。デフォルト: `false`.
host_groups	object	ホストグループをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しいホストグループが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のホストグループが更新されます。デフォルト: `false`.
template_groups	object	テンプレートグループをどのようにインポートするかのルール。 サポートされるパラメータ: `createMissing` - (`boolean`) `true` に設定すると、新しいテンプレートグループが作成されます。デフォルト: `false`; `updateExisting` - (`boolean`) `true` に設定すると、既存のテンプレートグループが更新されます。デフォルト: `false`.

パラメータ	型	説明
hosts	object	ホストをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいホストが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のホストが更新されます。デフォルト: false. このパラメータは出力に違いをもたらしません。 configuration.import との一貫性のためにのみ許可されています。
httptests	object	Web シナリオをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しい Web シナリオが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の Web シナリオが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しない Web シナリオがデータベースから削除されます。デフォルト: false.
images	object	イメージをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいイメージが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のイメージが更新されます。デフォルト: false. このパラメータは出力に違いをもたらしません。 configuration.import との一貫性のためにのみ許可されています。
items	object	アイテムをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいアイテムが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のアイテムが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しないアイテムがデータベースから削除されます。デフォルト: false.
maps	object	マップをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいマップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のマップが更新されます。デフォルト: false. このパラメータは出力に違いをもたらしません。 configuration.import との一貫性のためにのみ許可されています。
mediaTypes	object	メディアタイプをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいメディアタイプが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のメディアタイプが更新されます。デフォルト: false. このパラメータは出力に違いをもたらしません。 configuration.import との一貫性のためにのみ許可されています。

パラメータ	型	説明
templateLinkage	object	テンプレートリンクをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされていないが、インポートデータ内に存在するテンプレートがリンクされます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポート対象のホストまたはテンプレートにリンクされているが、インポートデータ内に存在しないテンプレートは、リンク解除されます。このとき、リンク解除されたテンプレートから継承されたエンティティ（アイテム、トリガーなど）は削除されません。デフォルト: false.
templates	object	テンプレートをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいテンプレートが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートが更新されます。デフォルト: false.
templateDashboards	object	テンプレートダッシュボードをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいテンプレートダッシュボードが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のテンプレートダッシュボードが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しないテンプレートダッシュボードがデータベースから削除されます。デフォルト: false.
triggers	object	トリガーをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しいトリガーが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存のトリガーが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しないトリガーがデータベースから削除されます。デフォルト: false.
valueMaps	object	ホストまたはテンプレートの値マップをどのようにインポートするかのルール。 サポートされるパラメータ: createMissing - (boolean) true に設定すると、新しい値マップが作成されます。デフォルト: false; updateExisting - (boolean) true に設定すると、既存の値マップが更新されます。デフォルト: false; deleteMissing - (boolean) true に設定すると、インポートデータに存在しない値マップがデータベースから削除されます。デフォルト: false.

戻り値

(array) 行われる設定の変更を含む配列を返します。

例

テンプレートのインポートの比較

XML 文字列に含まれるテンプレートを現在のシステム要素と比較し、このテンプレートをインポートした場合に何が変更されるかを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.importcompare",
    "params": {
        "format": "xml",
        "rules": {
            "discoveryRules": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "graphs": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "host_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "template_groups": {
                "createMissing": true,
                "updateExisting": true
            },
            "httptests": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "items": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "templateLinkage": {
                "createMissing": true,
                "deleteMissing": true
            },
            "templates": {
                "createMissing": true,
                "updateExisting": true
            },
            "templateDashboards": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "triggers": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            },
            "valueMaps": {
                "createMissing": true,
                "updateExisting": true,
                "deleteMissing": true
            }
        },
        "source": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>7.4</version><templ
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "templates": {
            "updated": [
                {
                    "before": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "after": {
                        "uuid": "5aef0444a82a4d8cb7a95dc4c0c85330",
                        "template": "New template",
                        "name": "New template",
                        "groups": [
                            {
                                "name": "Templates"
                            }
                        ]
                    },
                    "items": {
                        "added": [
                            {
                                "after": {
                                    "uuid": "648006da5971424ead0c47ddbbf1ea2e",
                                    "name": "CPU utilization",
                                    "key": "system.cpu.util",
                                    "value_type": "FLOAT",
                                    "units": "%"
                                },
                                "triggers": {
                                    "added": [
                                        {
                                            "after": {
                                                "uuid": "736225012c534ec480c2a66a91322ce0",
                                                "expression": "avg(/New template/system.cpu.util,3m)>70",
                                                "name": "CPU utilization too high on 'New host' for 3 minu
                                                "priority": "WARNING"
                                            }
                                        }
                                    ]
                                }
                            }
                        ],
                        "removed": [
                            {
                                "before": {
                                    "uuid": "6805d4c39a624a8bab2cc8ab63df1ab3",
                                    "name": "CPU load",
                                    "key": "system.cpu.load",
                                    "value_type": "FLOAT"
                                },
                                "triggers": {
                                    "removed": [
                                        {
```

```
                                        "before": {
                                            "uuid": "ab4c2526c2bc42e48a633082255ebcb3",
                                            "expression": "avg(/New template/system.cpu.load,3m)>2",
                                            "name": "CPU load too high on 'New host' for 3 minutes",
                                            "priority": "WARNING"
                                        }
                                    }
                                ]
                            }
                        }
                    ],
                    "updated": [
                        {
                            "before": {
                                "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                "name": "Zabbix agent ping",
                                "key": "agent.ping"
                            },
                            "after": {
                                "uuid": "7f1e6f1e48aa4a128e5b6a958a5d11c3",
                                "name": "Zabbix agent ping",
                                "key": "agent.ping",
                                "delay": "3m"
                            }
                        }
                    ]
                }
            }
        ]
    }
},
"id": 1
}
```


ソース

CConfiguration::importcompare() in *ui/include/classes/api/services/CConfiguration.php*.

Connector

このクラスは、コネクタオブジェクトを操作するためのものです。

オブジェクトリファレンス:

- Connector
- Tag filter

利用可能なメソッド:

- connector.create - 新しいコネクタを作成する
- connector.delete - コネクタを削除する
- connector.get - コネクタを取得する
- connector.update - コネクタを更新する

Connector オブジェクト

次のオブジェクトは、connectorAPI に直接関連しています。

コネクタ

コネクタオブジェクトには、以下のプロパティがあります。

Property	Type	Description
connectorid	ID	コネクタの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
name	string	コネクタの名前。 プロパティの動作: - 作成操作では 必須
url	string	エンドポイント URL、つまり受信側の URL。 ユーザーマクロを使用できます。 プロパティの動作: - 作成操作では 必須
protocol	integer	通信プロトコル。 指定可能な値: 0 - *(デフォルト)* Zabbix Streaming Protocol v1.0。
data_type	integer	データタイプ。 指定可能な値: 0 - *(デフォルト)* アイテムの値; 1 - イベント。
item_value_type	integer	送信するアイテムの値タイプの合計。 指定可能な値: 1 - 数値（浮動小数）; 2 - 文字列; 4 - ログ; 8 - 数値（符号なし）; 16 - テキスト; 32 - バイナリ。 デフォルト: 31 - すべてのアイテムタイプ（バイナリを除く）。 プロパティの動作: - `data_type` が「アイテムの値」に設定されている場合に サポート
max_records	integer	1 つのメッセージ内で送信できるイベントまたはアイテムの最大数。 指定可能な値: 0-2147483647（32 ビット符号付き整数の最大値）。 デフォルト: 0 - 無制限。
max_senders	integer	このコネクタで実行する送信プロセスの数。 指定可能な値: 1-100。 デフォルト: 1。
max_attempts	integer	試行回数。 指定可能な値: 1-5。 デフォルト: 1。
attempt_interval	string	再試行の間隔。 秒単位を受け付けます。 指定可能な値: 0s-10s。 デフォルト: 5s。 プロパティの動作: - `max_attempts` が 1 より大きい場合に サポート

Property	Type	Description
timeout	string	タイムアウト。 時間の接尾辞を使用できます（例: 30s、1m）。 ユーザーマクロを使用できます。 指定可能な値: 1s-60s。 デフォルト: 5s。
http_proxy	string	*[protocol]://[username[:password]@]proxy.example.com[:port]* の形式で指定する HTTP(S) プロキシ接続文字列。 ユーザーマクロを使用できます。
authtype	integer	HTTP 認証方式。 指定可能な値: 0 - *(デフォルト)* なし; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer。
username	string	ユーザー名。 ユーザーマクロを使用できます。 プロパティの動作: - `authtype` が「Basic」、「NTLM」、「Kerberos」、または「Digest」に設定されている場合に サポート
password	string	パスワード。 ユーザーマクロを使用できます。 プロパティの動作: - `authtype` が「Basic」、「NTLM」、「Kerberos」、または「Digest」に設定されている場合に サポート
token	string	Bearer トークン。 ユーザーマクロを使用できます。 プロパティの動作: - `authtype` が「Bearer」に設定されている場合は 必須
verify_peer	integer	ホストの証明書が正当であることを検証するかどうか。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
verify_host	integer	接続先のホスト名がホストの証明書内のものと一致することを検証するかどうか。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
ssl_cert_file	string	公開 SSL キーのファイルパス。 ユーザーマクロを使用できます。
ssl_key_file	string	秘密 SSL キーのファイルパス。 ユーザーマクロを使用できます。
ssl_key_password	string	SSL キーファイルのパスワード。 ユーザーマクロを使用できます。
description	text	コネクタの説明。
status	integer	コネクタを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

Property	Type	Description
tags_evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。

タグフィルター

タグフィルターを使用すると、一致するアイテムの値またはイベントのみをエクスポートできます。設定しない場合は、すべてがエクスポートされます。タグフィルターオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	タグ名。 プロパティの動作: - 必須
operator	integer	条件演算子。 使用可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 12 - 存在する; 13 - 存在しない。
value	string	タグの値。

connector.create

説明

`object connector.create(object/array connectors)`

このメソッドは、新しいコネクタオブジェクトを作成することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

作成するコネクタオブジェクト。(`object/array`)

このメソッドは、標準のコネクタプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
tags	array	コネクタのタグフィルター。

戻り値

(`object`) `connectorids` プロパティの下に作成されたコネクタの ID を含むオブジェクトを返します。返される ID の順序は、渡されたコネクタの順序と一致します。

例

コネクタの作成

タグフィルターを使用してトリガーイベントをエクスポートするためのコネクタを作成します。HTTP 認証は Bearer トークンを使用して実行されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.create",
    "params": [
        {
            "name": "Export of events",
            "data_type": 1,
            "url": "{$DATA_EXPORT_URL}",
            "authtype": 5,
            "token": "{$DATA_EXPORT_BEARER_TOKEN}",
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

ソース

CConnector::create() in *ui/include/classes/api/services/CConnector.php*.

connector.delete

説明

`object connector.delete(array connectorids)`

このメソッドは、コネクタのエントリを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するコネクタの ID。

戻り値

(object) connectorids プロパティの下にある、削除されたコネクタの ID を含むオブジェクトを返します。

例

複数のコネクタの削除

2 つのコネクタエントリを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CConnector::delete() in *ui/include/classes/api/services/CConnector.php*.

connector.get

説明

`integer/array connector.get(object parameters)`

このメソッドは、与えられたパラメーターに従ってコネクタオブジェクトを取得することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
connectorids	ID/array	指定された ID を持つコネクタのみを返します。
selectTags	query	コネクタのtag filterを含む `tags` プロパティを返します。 `count` をサポートします。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `connectorid`, `name`, `data_type`, `status`。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

すべてのコネクタを取得する

すべてのコネクタとそのプロパティに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Export of item values",
            "protocol": "0",
            "data_type": "0",
            "url": "{$DATA_EXPORT_VALUES_URL}",
            "item_value_type": "31",
            "authtype": "4",
            "username": "{$DATA_EXPORT_VALUES_USERNAME}",
            "password": "{$DATA_EXPORT_VALUES_PASSWORD}",
            "token": "",
            "max_records": "0",
            "max_senders": "4",
            "max_attempts": "2",
            "attempt_interval": "10s",
            "timeout": "10s",
            "http_proxy": "{$DATA_EXPORT_VALUES_PROXY}",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "{$DATA_EXPORT_VALUES_SSL_CERT_FILE}",
            "ssl_key_file": "{$DATA_EXPORT_VALUES_SSL_KEY_FILE}",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "component",
                    "operator": "0",
                    "value": "memory"
                }
            ]
        },
        {
            "connectorid": "2",
            "name": "Export of events",
            "protocol": "0",
            "data_type": "1",
```

```
            "url": "{$DATA_EXPORT_EVENTS_URL}",
            "item_value_type": "31",
            "authtype": "5",
            "username": "",
            "password": "",
            "token": "{$DATA_EXPORT_EVENTS_BEARER_TOKEN}",
            "max_records": "0",
            "max_senders": "2",
            "max_attempts": "1",
            "attempt_interval": "5s",
            "timeout": "5s",
            "http_proxy": "",
            "verify_peer": "1",
            "verify_host": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "description": "",
            "status": "1",
            "tags_evaltype": "0",
            "tags": [
                {
                    "tag": "scope",
                    "operator": "0",
                    "value": "performance"
                }
            ]
        }
    ],
    "id": 1
}
```

ソース

CConnector:get() in *ui/include/classes/api/services/CConnector.php*.

connector.update

説明

`object connector.update(object/array connectors)`

このメソッドは、既存のコネクタを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するコネクタのプロパティ。

各コネクタに対して `connectorid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のコネクタプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
tags	array	現在のタグフィルターを置き換えるコネクタのタグフィルター。

戻り値

(`object`) `connectorids` プロパティの下にある、更新されたコネクタの ID を含むオブジェクトを返します。

例

HTTP 認証タイプの変更

ID ”3” のコネクタの HTTP 認証タイプをベアラーに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

タグフィルターの更新

ID ”5” のコネクタのタグフィルターを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": [
        {
            "connectorid": 5,
            "tags_evaltype": 2,
            "tags": [
                {
                    "tag": "service",
                    "operator": 0,
                    "value": "mysqld"
                },
                {
                    "tag": "error",
                    "operator": 12,
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
```

```
    "id": 1
}
```

ソース

CConnector::update() in *ui/include/classes/api/services/CConnector.php*.

Graph

このクラスは、グラフを操作するためのものです。

オブジェクトリファレンス:

- Graph

利用可能なメソッド:

- graph.create - 新しいグラフの作成
- graph.delete - グラフの削除
- graph.get - グラフの取得
- graph.update - グラフの更新

Graph オブジェクト

次のオブジェクトは、graphAPI に直接関連しています。

グラフ

グラフオブジェクトには、以下のプロパティがあります。

Property	Type	Description
graphid	ID	グラフの ID。 Property behavior: - *read-only* - 更新操作で *required*
height	integer	グラフの高さ（ピクセル単位）。 Property behavior: - 作成操作で *required*
name	string	グラフの名前。 Property behavior: - 作成操作で *required*
width	integer	グラフの幅（ピクセル単位）。 Property behavior: - 作成操作で *required*
flags	integer	グラフの生成元。 指定可能な値: 0 - *(default)* 通常のグラフ; 4 - プロトタイプから変換されたグラフ。 Property behavior: - *read-only*
graphtype	integer	グラフのレイアウトタイプ。 指定可能な値: 0 - *(default)* 通常; 1 - 積み上げ; 2 - 円グラフ; 3 - 分離。

Property	Type	Description
percent_left	float	左パーセンタイル。 デフォルト: 0。
percent_right	float	右パーセンタイル。 デフォルト: 0。
show_3d	integer	円グラフおよび分離グラフを 3D で表示するかどうか。 指定可能な値: 0 - *(default)* 2D で表示; 1 - 3D で表示。
show_legend	integer	グラフに凡例を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(default)* 表示。
show_work_period	integer	グラフに稼働時間を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(default)* 表示。
show_triggers	integer	グラフにトリガー線を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(default)* 表示。
templateid	ID	親テンプレートグラフの ID。 Property behavior: - *read-only*
yaxismax	float	Y 軸の固定最大値。 デフォルト: 100。
yaxismin	float	Y 軸の固定最小値。 デフォルト: 0。
ymax_itemid	ID	Y 軸の最大値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは ymax_type が”calculated” に設定されているものとして描画されます。
ymax_type	integer	Y 軸の最大値の計算方法。 指定可能な値: 0 - *(default)* calculated; 1 - fixed; 2 - item。
ymin_itemid	ID	Y 軸の最小値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは ymin_type が”calculated” に設定されているものとして描画されます。
ymin_type	integer	Y 軸の最小値の計算方法。 指定可能な値: 0 - *(default)* calculated; 1 - fixed; 2 - item。
uuid	string	インポートされたグラフを既存のグラフに関連付けるために使用される汎用一意識別子。指定されていない場合は自動生成されます。 Property behavior: - グラフがテンプレートに属する場合に *supported*

graph.create

説明

`object graph.create(object/array graphs)`

このメソッドは、新しいグラフを作成することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成するグラフ。

標準のグラフプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	グラフ用に作成するグラフアイテム。 パラメータの動作: - 必須

戻り値

(object) graphids プロパティの下に作成されたグラフの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグラフの順序と一致します。

例

グラフの作成

2 つのアイテムからなるグラフを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.create",
    "params": {
        "name": "MySQL bandwidth",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
```

```
    "id": 1
}
```

関連項目

- グラフアイテム

ソース

CGraph::create() in *ui/include/classes/api/services/CGraph.php*.

graph.delete

説明

`object graph.delete(array graphIds)`

このメソッドは、グラフを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するグラフの ID。

戻り値

(object) graphids プロパティの下にある削除されたグラフの ID を含むオブジェクトを返します。

例

複数のグラフの削除

2 つのグラフを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

ソース

CGraph::delete() in *ui/include/classes/api/services/CGraph.php*.

graph.get

説明

integer/array graph.get(object parameters)

このメソッドは、指定したパラメーターに応じたグラフを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
graphids	ID/array	指定した ID のグラフのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループ内のホストまたはテンプレートに属するグラフのみを返します。
templateids	ID/array	指定したテンプレートに属するグラフのみを返します。
hostids	ID/array	指定したホストに属するグラフのみを返します。
itemids	ID/array	指定したアイテムを含むグラフのみを返します。
templated	boolean	true に設定した場合、テンプレートに属するグラフのみを返します。
inherited	boolean	true に設定した場合、テンプレートから継承されたグラフのみを返します。
expandName	flag	グラフ名内のマクロを展開します。
selectDiscoveryData	query	グラフのディスカバリオブジェクトデータを含む discoveryData プロパティを返します。グラフのディスカバリオブジェクトは、ディスカバリされたグラフを、その元となったグラフプロトタイプに関連付けます。 以下のプロパティがあります。 parent_graphid - (ID) グラフの作成元となったグラフプロトタイプの ID。 status - (int) グラフのディスカバリステータス: 0 - *(default)* グラフはディスカバリされています。 1 - グラフはすでにディスカバリされていません。 ts_delete - (timestamp) すでにディスカバリされなくなったグラフが削除される時刻。
selectDiscoveryRule	query	グラフを作成したローレベルディスカバリルールを含むdiscoveryRule プロパティを返します。
selectHostGroups	query	グラフが属するホストグループを含むhostgroups プロパティを返します。
selectTemplateGroups	query	グラフが属するテンプレートグループを含むtemplategroups プロパティを返します。
selectTemplates	query	グラフが属するテンプレートを含むtemplates プロパティを返します。
selectHosts	query	グラフが属するホストを含むhosts プロパティを返します。
selectItems	query	グラフで使用されるアイテムを含むitems プロパティを返します。
selectGraphItems	query	グラフで使用されるアイテムを含むgitems プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 text data type のプロパティはサポートしていません。 追加のプロパティをサポートしています。 host - グラフが属するホストの技術名。 hostid - グラフが属するホストの ID。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: graphid, name, graphtype。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	

Parameter	Type	Description
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGraphDiscovery	query	グラフのディスカバリオブジェクトを含む graphDiscovery プロパティを返します。グラフのディスカバリオブジェクトは、グラフをその作成元となったグラフプロトタイプに関連付けます。 このクエリは 非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストからグラフの取得

ホスト"10107" から全てのグラフを取得し、名前でソートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "612",
            "name": "CPU jumps",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "439",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
```

```
        },
        {
            "graphid": "613",
            "name": "CPU load",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "433",
            "show_work_period": "1",
            "show_triggers": "1",
            "graphtype": "0",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "614",
            "name": "CPU utilization",
            "width": "900",
            "height": "200",
            "yaxismin": "0",
            "yaxismax": "100",
            "templateid": "387",
            "show_work_period": "1",
            "show_triggers": "0",
            "graphtype": "1",
            "show_legend": "1",
            "show_3d": "0",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "1",
            "ymax_type": "1",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "0"
        },
        {
            "graphid": "645",
            "name": "Disk space usage /",
            "width": "600",
            "height": "340",
            "yaxismin": "0",
            "yaxismax": "0",
            "templateid": "0",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0",
            "percent_right": "0",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
```

```
            "flags": "4"
        }
    ],
    "id": 1
}
```

関連項目

- ディスカバリルール
- グラフアイテム
- アイテム
- ホスト
- ホストグループ
- テンプレート
- テンプレートグループ

ソース

CGraph::get() in *ui/include/classes/api/services/CGraph.php*.

graph.update

説明

`object graph.update(object/array graphs)`

このメソッドでは、既存のグラフを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object/array) 更新するグラフのプロパティ。

各グラフでは graphid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準のグラフプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	既存のグラフアイテムを置き換えるグラフアイテム。グラフアイテムに gitemid プロパティが定義されている場合は更新され、そうでない場合は新しいグラフアイテムが作成されます。

戻り値

(object) graphids プロパティの下で更新されたグラフの ID を含むオブジェクトを返す。

例

Y 軸スケールの最大値の設定

Y 軸スケールの最大値を固定値 100 に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

ソース

CGraph::update() in *ui/include/classes/api/services/CGraph.php*.

Housekeeping

このクラスは、ハウスキーピングを操作するためのものです。

オブジェクトリファレンス:

- Housekeeping

利用可能なメソッド:

- housekeeping.get - ハウスキーピングの取得
- housekeeping.update - ハウスキーピングの更新

Housekeeping オブジェクト

次のオブジェクトは、housekeepingAPI に直接関連しています。

ハウスキーピング

settings オブジェクトには以下のプロパティがあります。

Property	Type	Description
hk_events_mode	integer	イベントおよびアラートの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_events_trigger	string	トリガーデータの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 365d。
hk_events_service	string	サービスデータの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。
hk_events_internal	string	内部データの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。
hk_events_discovery	string	ネットワークディスカバリデータの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。
hk_events_autoreg	string	自動登録データの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 1d。

Property	Type	Description
hk_services_mode	integer	サービスの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_services	string	サービスデータの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 365d。
hk_audit_mode	integer	監査の内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_audit	string	監査データの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 31d。
hk_sessions_mode	integer	セッションの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_sessions	string	セッションデータの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 31d。
hk_history_mode	integer	履歴の内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_history_global	integer	アイテムの履歴期間を上書きします。 指定可能な値: 0 - 上書きしない; 1 - *(デフォルト)* 上書きする。
hk_history	string	履歴データの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 31d。
hk_trends_mode	integer	トレンドの内部ハウスキーピングを有効にします。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
hk_trends_global	integer	アイテムのトレンド期間を上書きします。 指定可能な値: 0 - 上書きしない; 1 - *(デフォルト)* 上書きする。
hk_trends	string	トレンドデータの保存期間。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 365d。
db_extension	string	設定フラグ DB 拡張。このフラグが”timescaledb” に設定されている場合、サーバーはハウスキーピングおよびアイテム削除に対する動作を変更します。 Property behavior: - *read-only*

Property	Type	Description
compression_availability	integer	データ圧縮がデータベース（またはその拡張機能）でサポートされているかどうか。 指定可能な値: 0 - 利用不可; 1 - 利用可能。 Property behavior: - *read-only*
compression_status	integer	履歴およびトレンドに対して TimescaleDB 圧縮を有効にします。 指定可能な値: 0 - *(デフォルト)* オフ; 1 - オン。
compress_older	string	指定した期間より古い履歴およびトレンドのレコードを圧縮します。秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 7d。

housekeeping.get

説明

`object housekeeping.get(object parameters)`

このメソッドは、指定されたパラメータに従ってハウスキーピングオブジェクトを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	このパラメータについては、リファレンスコメントで説明されています。

戻り値

(object) housekeeping オブジェクトを返します

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hk_events_mode": "1",
```

```
        "hk_events_trigger": "365d",
        "hk_events_service": "1d",
        "hk_events_internal": "1d",
        "hk_events_discovery": "1d",
        "hk_events_autoreg": "1d",
        "hk_services_mode": "1",
        "hk_services": "365d",
        "hk_audit_mode": "1",
        "hk_audit": "31d",
        "hk_sessions_mode": "1",
        "hk_sessions": "365d",
        "hk_history_mode": "1",
        "hk_history_global": "0",
        "hk_history": "31d",
        "hk_trends_mode": "1",
        "hk_trends_global": "0",
        "hk_trends": "365d",
        "db_extension": "",
        "compression_status": "0",
        "compress_older": "7d"
    },
    "id": 1
}
```


ソース

CHousekeeping ::get() in *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

説明

`object housekeeping.update(object housekeeping)`

このメソッドでは、既存のハウスキーピング設定を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

更新する（object）のハウスキーピングプロパティ。

戻り値

（array）更新されたパラメータ名の配列を返します。

例

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.update",
    "params": {
        "hk_events_mode": "1",
        "hk_events_trigger": "200d",
        "hk_events_internal": "2d",
        "hk_events_discovery": "2d"
    },
    "id": 1
}
```


レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

ソース

CHousekeeping::update() in *ui/include/classes/api/services/CHousekeeping.php*.

Image

このクラスは、画像を操作するためのものです。

オブジェクトリファレンス:

- Image

利用可能なメソッド:

- image.create - 新しい画像を作成する
- image.delete - 画像を削除する
- image.get - 画像を取得する
- image.update - 画像を更新する

Image オブジェクト

次のオブジェクトは、imageAPI に直接関連しています。

画像

画像オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
imageid	ID	画像の ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
name	string	画像の名前。 プロパティの動作: - 作成操作で 必須
imagetype	integer	画像の種類。 指定可能な値: 1 - *(デフォルト)* アイコン; 2 - 背景画像。 プロパティの動作: - 定数 - 作成操作で 必須
image	string	Base64 でエンコードされた画像。 エンコードされた画像の最大サイズは 1 MB です。最大サイズは、ZBX_MAX_IMAGE_SIZE 定数の値を変更することで調整できます。 サポートされる画像形式: PNG、JPEG、GIF、WebP。 プロパティの動作: - 作成操作で 必須

image.create

説明

`object image.create(object/array images)`

このメソッドでは、新しい画像を作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(`object/array`) 作成する画像。

このメソッドは、標準の画像プロパティを持つ画像を受け付けます。

戻り値

(`object`) `imageids` プロパティの下で作成された画像の ID を含むオブジェクトを返します。返される ID の順序は、渡された画像の順序と一致します。

例

アイコン画像の作成

クラウドアイコンを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```

背景画像の作成

パターンとして使用する小さな背景画像を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

ソース

CImage::create() in *ui/include/classes/api/services/CImage.php*.

image.delete

説明

`object image.delete(array imageIds)`

このメソッドは、画像を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する画像の ID。

戻り値

(object) imageids プロパティの下にある削除された画像の ID を含むオブジェクトを返します。

例

複数の画像の削除

2 つの画像を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

ソース

CImage::delete() in *ui/include/classes/api/services/CImage.php*.

image.get

説明

`integer/array image.get(object parameters)`

このメソッドでは、指定されたパラメータに従って画像を取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
imageids	ID/array	指定された ID を持つ画像のみを返します。
sysmapids	ID/array	指定されたマップで使用されている画像を返します。
select_image	flag	Base64 エンコードされた画像を含む `image` プロパティを返します。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `imageid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

画像を取得する

ID が"2" の画像のすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```


ソース

CImage::get() in *ui/include/classes/api/services/CImage.php*.

image.update

説明

`object image.update(object/array images)`

このメソッドでは、既存の画像を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) 更新する画像のプロパティ。

各画像には imageid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準の画像プロパティ を持つ画像を受け付けます。

戻り値

(object) imageids プロパティの下で更新された画像の ID を含むオブジェクトを返します。

例

画像名の変更

画像名を「Cloud icon」に変更します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
```

```
    "id": 1
}
```

ソース

CImage::update() in *ui/include/classes/api/services/CImage.php*.

LLD ルール

このクラスは、ローレベルディスカバリルールで動作するように設計されています。

オブジェクト参照:

- LLD ルール
  - HTTP ヘッダー
  - HTTP クエリフィールド
- LLD ルールフィルター
  - LLD ルールフィルター条件
- LLD マクロパス
- LLD ルール前処理
- LLD ルールオーバーライド
  - LLD ルールオーバーライドフィルター
    - LLD ルールオーバーライドフィルター条件
  - LLD ルールオーバーライド操作
    - LLD ルールオーバーライド操作ステータス
    - LLD ルールオーバーライド操作ディスカバリ
    - LLD ルールオーバーライド操作期間
    - LLD ルールオーバーライド操作履歴
    - LLD ルールオーバーライド操作トレンド
    - LLD ルールオーバーライド操作深刻度
    - LLD ルールオーバーライド操作タグ
    - LLD ルールオーバーライド操作テンプレート
    - LLD ルールオーバーライド操作インベントリ

利用可能なメソッド:

- discoveryrule.create - 新しい LLD ルールを作成
- discoveryrule.delete - LLD ルールを削除
- discoveryrule.get - LLD ルールを取得
- discoveryrule.update - LLD ルールを更新

LLD rule オブジェクト

次のオブジェクトは、`discoveryruleAPI` に直接関連しています。

LLD ルール

低レベルディスカバリルールオブジェクトには、以下のプロパティがあります。

Property	Type	Description
itemid	ID	LLD ルールの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須

Property	Type	Description
delay	string	LLD ルールの更新間隔。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）を受け付けます。また、オプションで 1 つ以上のカスタム間隔も指定でき、これらはすべてセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロを受け付けます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはテキストと混在したマクロはサポートされません。柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が”Zabbix agent” (0)、”Simple check” (3)、”Zabbix internal” (5)、”External check” (10)、”Database monitor” (11)、”IPMI agent” (12)、”SSH agent” (13)、”TELNET agent” (14)、”JMX agent” (16)、”HTTP agent” (19)、”SNMP agent” (20)、”Script” (21)、”Browser” (22) に設定されている場合、または `type` が”Zabbix agent (active)” (7) に設定され、かつ `key_` に”mqtt.get” が含まれない場合に 必須
hostid	ID	LLD ルールが属するホストの ID。 プロパティの動作: - 定数 - 作成操作で 必須
flags	integer	ディスカバリルールの生成元。 指定可能な値: 1 - 低レベルディスカバリルール; 5 - プロトタイプから変換された低レベルディスカバリルール。 プロパティの動作: - 読み取り専用
interfaceid	ID	LLD ルールのホストインターフェースの ID。 プロパティの動作: - LLD ルールがホストに属し、`type` が”Zabbix agent”、”IPMI agent”、”JMX agent”、または”SNMP agent” に設定されている場合に 必須 - LLD ルールがホストに属し、`type` が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合に サポート
key_	string	LLD ルールキー。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
name	string	LLD ルール名。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用

Property	Type	Description
type	integer	LLD ルールのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested。このタイプは、ルールの親がテンプレート（ディスカバリ時にホストプロトタイプへリンクされることを想定）またはディスカバリされたホスト（ルートに LLD ルールを持つ）の場合にのみ許可されます。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
url	string	URL 文字列。 {HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} およびユーザーマクロをサポートします。 プロパティの動作: - type が”HTTP agent” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用
allow_traps	integer	trapper アイテムと同様に値を設定できるようにします。 指定可能な値: 0 - *(デフォルト)* 受信データの受け入れを許可しない; 1 - 受信データの受け入れを許可する。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート
authtype	integer	認証方式。 type が”SSH agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest。 プロパティの動作: - type が”SSH agent” または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用（type が”HTTP agent” に設定されている場合）

Property	Type	Description
description	string	LLD ルールの説明。
error	string	LLD ルールの値の更新に問題がある場合のエラーテキスト。 プロパティの動作: - 読み取り専用
follow_redirects	integer	データ取得時にレスポンスのリダイレクトに従うかどうか。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - type が"HTTP agent" に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - type が"HTTP agent" に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が"HTTP agent" に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が"IPMI agent" に設定され、かつ key_ が"ipmi.get" に設定されていない場合に 必須 - type が"IPMI agent" に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が"JMX agent" に設定されている場合に サポート
lifetime	string	ディスカバリされなくなったアイテムが削除されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを受け付けます。 デフォルト: 7d。
lifetime_type	integer	失われた LLD リソースを削除するシナリオ。 指定可能な値: 0 - *(デフォルト)* lifetime のしきい値に達した後に削除; 1 - 削除しない; 2 - 即時削除。
enabled_lifetime	string	ディスカバリされなくなったアイテムが無効化されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを受け付けます。 デフォルト: 0。
enabled_lifetime_type	integer	失われた LLD リソースを無効化するシナリオ。 指定可能な値: 0 - lifetime のしきい値に達した後に無効化; 1 - 無効化しない; 2 - *(デフォルト)* 即時無効化。

Property	Type	Description
master_itemid	ID	マスターアイテムの ID。 ディスカバリルールを別のディスカバリルールのマスターアイテムにすることはできません。 プロパティの動作: - type が”Dependent item” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データのまま保存; 1 - JSON に変換。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
params	string	LLD ルールのタイプに応じた追加パラメータ: - SSH および Telnet LLD ルール用の実行スクリプト; - データベースモニタ LLD ルール用の SQL クエリ; - 計算 LLD ルール用の式; - Script および Browser LLD ルール用のスクリプト。 プロパティの動作: - type が”Database monitor”、”SSH agent”、”TELNET agent”、”Script”、または”Browser” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用（type が”Script” または”Browser” に設定されている場合）
parameters	object/array	type が”Script” または”Browser” に設定されている場合の追加パラメータ。 name と value プロパティを持つオブジェクトの配列で、name は一意でなければなりません。 プロパティの動作: - type が”Script” または”Browser” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
password	string	認証用パスワード。 プロパティの動作: - type が”JMX agent” に設定され、かつ username が設定されている場合に 必須 - type が”Simple check”、”Database monitor”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用（type が”HTTP agent” に設定されている場合）
post_type	integer	posts プロパティに格納される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
posts	string	HTTP(S) リクエスト本文データ。 プロパティの動作: - type が”HTTP agent” に設定され、かつ post_type が”JSON data” または”XML data” に設定されている場合に 必須 - type が”HTTP agent” に設定され、かつ post_type が”Raw data” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用

Property	Type	Description
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” に設定され、かつ `authtype` が”public key” に設定されている場合に 必須
publickey	string	公開鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” に設定され、かつ `authtype` が”public key” に設定されている場合に 必須
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、または”PUT” に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が”HEAD” に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が”SNMP agent” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用
ssl_cert_file	string	公開 SSL 鍵ファイルのパス。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
ssl_key_file	string	秘密 SSL 鍵ファイルのパス。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
ssl_key_password	string	SSL 鍵ファイルのパスワード。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用

Property	Type	Description
state	integer	LLD ルールの状態。 指定可能な値: 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	LLD ルールのステータス。 指定可能な値: 0 - *(デフォルト)* 有効な LLD ルール; 1 - 無効な LLD ルール。
status_codes	string	必要な HTTP ステータスコードの範囲。カンマ区切りで指定します。カンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
templateid	ID	親テンプレート LLD ルールの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒、またはサフィックス付きの時間単位（例: 30s、1m）を受け付けます。 ユーザーマクロも受け付けます。 指定可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの動作: - type が”Zabbix agent” (0)、”Simple check” (3) で key_ が”vmware.” および”icmpping” で始まらない場合、”Zabbix agent (active)” (7)、”External check” (10)、”Database monitor” (11)、”SSH agent” (13)、”TELNET agent” (14)、”HTTP agent” (19)、”SNMP agent” (20) で snmp_oid が”walk[” または”get[” で始まる場合、”Script” (21)、”Browser” (22) に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - type が”Zabbix trapper” に設定されている場合、または type が”HTTP agent” に設定され、かつ allow_traps が”Allow to accept incoming data” に設定されている場合に サポート
username	string	認証用ユーザー名。 プロパティの動作: - type が”SSH agent”、”TELNET agent” に設定されている場合、または type が”JMX agent” に設定され、かつ password が設定されている場合に 必須 - type が”Simple check”、”Database monitor”、または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用（type が”HTTP agent” に設定されている場合）
uuid	string	既存の LLD ルールにインポートした LLD ルールをリンクするために使用される汎用一意識別子。指定しない場合は自動生成されます。 プロパティの動作: - LLD ルールがテンプレートに属する場合に サポート

Property	Type	Description
verify_host	integer	接続先のホスト名がホスト証明書内の名前と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
verify_peer	integer	ホストの証明書が正当であることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用

HTTP ヘッダー

ヘッダーオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

LLD ルールフィルター

LLD ルールフィルターオブジェクトは、検出されたオブジェクトをフィルタリングするために使用できる一連の条件を定義します。次のプロパティがあります。

Property	Type	Description
conditions	object/array	結果のフィルタリングに使用するフィルター条件のセット。条件は、formula 内で配置された順序でソートされます。 Property behavior: - 必須

Property	Type	Description
evaltype	integer	フィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - 必須
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、各フィルター条件をその `formulaid` で参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 Property behavior: - 読み取り専用
formula	string	カスタム式を使用するフィルターの条件評価に使用する、ユーザー定義の式。この式には、各フィルター条件をその `formulaid` で参照する ID が含まれている必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致している必要があります。どの条件も未使用または省略されたままにすることはできません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は必須

LLD ルールのフィルター条件

LLD ルールのフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。以下のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する対象の LLD マクロ。 Property behavior: - *required*
value	string	比較対象の値。 Property behavior: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は *required*
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際には、この ID はユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。 Property behavior: - LLD rule filter object の `evaltype` が「カスタム式」に設定されている場合は *required*
operator	integer	条件演算子。 指定可能な値: 8 - *(default)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、discoveryrule.get およびdiscoveryrule.create メソッドのページにある例を参照してください。

LLD マクロパス

LLD マクロパスには、以下のプロパティがあります。

プロパティ	Type	説明
lld_macro	string	LLD マクロ。 プロパティの動作: - 必須
path	string	対応するマクロに割り当てられる値のセレクター。 プロパティの動作: - 必須

LLD ルールの前処理

LLD ルールの前処理オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
type	integer	前処理オプションのタイプ。 指定可能な値: 5 - 正規表現; 11 - XML XPath; 12 - JSONPath; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON 内のエラーをチェック; 17 - XML 内のエラーをチェック; 20 - ハートビート付きで未変更を破棄; 21 - JavaScript; 23 - Prometheus から JSON; 24 - CSV から JSON; 25 - 置換; 27 - XML から JSON; 28 - SNMP walk 値; 29 - SNMP walk から JSON; 30 - SNMP get 値。 プロパティの動作: - 必須
params	string	前処理オプションで使用される追加パラメータ。複数のパラメータは改行文字 (\n) で区切ります。 プロパティの動作: - type が”Regular expression” (5)、”XML XPath” (11)、”JSONPath” (12)、”Matches regular expression” (14)、”Does not match regular expression” (15)、”Check for error in JSON” (16)、”Check for error in XML” (17)、”Discard unchanged with heartbeat” (20)、”JavaScript” (21)、”Prometheus to JSON” (23)、”CSV to JSON” (24)、”Replace” (25)、”SNMP walk value” (28)、”SNMP walk to JSON” (29)、または”SNMP get value” (30) に設定されている場合は 必須

プロパティ	型	説明
error_handler	integer	前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定される; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - `type` が”Regular expression” (5)、”XML XPath” (11)、”JSONPath” (12)、”Matches regular expression” (14)、”Does not match regular expression” (15)、”Check for error in JSON” (16)、”Check for error in XML” (17)、”Prometheus to JSON” (23)、”CSV to JSON” (24)、”XML to JSON” (27)、”SNMP walk value” (28)、”SNMP walk to JSON” (29)、または”SNMP get value” (30) に設定されている場合は 必須
error_handler_params	string	エラーハンドラーパラメータ。 プロパティの動作: - `error_handler` が”Set custom value” または”Set custom error message” に設定されている場合は 必須

以下のパラメータおよびエラーハンドラーが、各前処理タイプでサポートされています。

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
5	Regular expression	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Matches regular expression	pattern[1]			0, 1, 2, 3
15	Does not match regular expression	pattern[1]			0, 1, 2, 3
16	Check for error in JSON	path[3]			0, 1, 2, 3
17	Check for error in XML	path[3]			0, 1, 2, 3

前処理タイプ	名前	パラメータ 1	パラメータ 2	パラメータ 3	サポートされるエラーハンドラー
20	ハートビート付きで未変更を破棄	seconds[4, 5]			
21	JavaScript	script[2]			
23	Prometheus から JSON	pattern[5, 6]			0, 1, 2, 3
24	CSV から JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
27	XML から JSON				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON[7]	フィールド名[2]	OID プレフィックス[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 正規表現
[2] 文字列
[3] JSONPath または XML XPath
[4] 正の整数（時間サフィックスをサポート。例: 30s, 1m, 2h, 1d）
[5] ユーザーマクロ
[6] 以下の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。各 Prometheus パターンの構成要素（メトリクス、ラベル名、ラベル値、メトリクス値）にはユーザーマクロを使用できます。
[7] 改行文字で区切られた複数の”Field name,OID prefix,Format records” レコードをサポートします。

LLD ルールのオーバーライド

LLD ルールのオーバーライドオブジェクトは、異なるプロトタイプオブジェクトのプロパティを上書きするために使用される一連のルール（フィルター、条件、操作）を定義します。次のプロパティがあります。

Property	Type	Description
name	string	一意のオーバーライド名。 プロパティの動作: - 必須
step	integer	オーバーライドの一意の順序番号。 プロパティの動作: - 必須
stop	integer	一致した場合、後続のオーバーライドの処理を停止します。 指定可能な値: 0 - *(デフォルト)* オーバーライドの処理を停止しない; 1 - フィルターに一致した場合、オーバーライドの処理を停止する。
filter	object	オーバーライドフィルター。
operations	object/array	オーバーライド操作。

LLD ルールのオーバーライドフィルター

LLD ルールのオーバーライドフィルターオブジェクトは、一連の条件を定義します。これらの条件が検出されたオブジェクトに一致した場合、オーバーライドが適用されます。以下のプロパティがあります。

Property	Type	Description
conditions	object/array	検出されたオブジェクトとの一致に使用するオーバーライドフィルター条件のセット。条件は、formula 内での配置順に並べ替えられます。 Property behavior: - *required*
evaltype	integer	オーバーライドフィルター条件の評価方法。 使用可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - *required*
eval_formula	string	オーバーライドフィルター条件の評価に使用される生成済みの式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 Property behavior: - *read-only*
formula	string	カスタム式を使用するオーバーライドフィルターの条件評価に使用する、ユーザー定義の式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID を含める必要があります。式で使用する ID は、オーバーライドフィルター条件で定義されたものと完全に一致していなければなりません。未使用または省略された条件があってはなりません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は *required*

LLD ルールのオーバーライドフィルター条件

LLD ルールのオーバーライドフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。以下のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する対象の LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は 必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際は、この ID をユーザーが定義する必要がありますが、その後に条件をリクエストすると新たに生成されます。 プロパティの動作: - LLD ルールのオーバーライドフィルターオブジェクト の `evaltype` が「カスタム式」に設定されている場合は 必須
operator	integer	条件演算子。 指定可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

LLD ルールのオーバーライド操作

LLD ルールのオーバーライド操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。以下のプロパティがあります。

プロパティ	Type	説明
operationobject	integer	アクションを実行する検出済みオブジェクトのタイプ。 使用可能な値: 0 - アイテムのプロトタイプ; 1 - トリガーのプロトタイプ; 2 - グラフのプロトタイプ; 3 - ホストのプロトタイプ; 4 - ディスカバリのプロトタイプ。 プロパティの動作: - 必須
operator	integer	オーバーライド条件のoperator。 使用可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 8 - 一致する; 9 - 一致しない。
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフ、またはホストのプロトタイプ名に一致させるパターン。
opstatus	object	アイテム、トリガー、およびホストのプロトタイプオブジェクト用のオーバーライド操作ステータスオブジェクト。
opdiscover	object	オーバーライド操作のディスカバリステータスオブジェクト（すべてのオブジェクトタイプ）。
opperiod	object	アイテムのプロトタイプオブジェクト用のオーバーライド操作期間（更新間隔）オブジェクト。
ophistory	object	アイテムのプロトタイプオブジェクト用のオーバーライド操作履歴オブジェクト。

プロパティ	Type	説明
optrends	object	アイテムのプロトタイプオブジェクト用のオーバーライド操作トレンドオブジェクト。
opseverity	object	トリガーのプロトタイプオブジェクト用のオーバーライド操作深刻度オブジェクト。
optag	object/array	トリガーおよびホストのプロトタイプオブジェクト用のオーバーライド操作タグオブジェクト。
optemplate	object/array	ホストのプロトタイプオブジェクト用のオーバーライド操作テンプレートオブジェクト。
opinventory	object	ホストのプロトタイプオブジェクト用のオーバーライド操作インベントリオブジェクト。

LLD ルールのオーバーライド操作ステータス

検出されたオブジェクトに設定される LLD ルールのオーバーライド操作ステータスです。
以下のプロパティがあります。

プロパティ	Type	説明
status	integer	選択したオブジェクトのステータスをオーバーライドします。 指定可能な値: 0 - 有効で作成; 1 - 無効で作成。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作 discover

LLD ルールのオーバーライド操作 discover は、検出されたオブジェクトに設定される discover ステータスです。以下のプロパティがあります。

Property	Type	Description
discover	integer	選択したオブジェクトの discover ステータスをオーバーライドします。 指定可能な値: 0 - はい。オブジェクトの検出を継続します。 1 - いいえ。新しいオブジェクトは検出されず、既存のオブジェクトは失われたものとしてマークされます。 Property behavior: *- required*

LLD ルールの上書き操作期間

LLD ルールの上書き操作期間は、検出されたアイテムに設定される更新間隔の値です。以下のプロパティがあります。

プロパティ	Type	説明
delay	string	アイテムプロトタイプの更新間隔を上書きします。 秒、または接尾辞付きの時間単位（例: 30s、1m、2h、1d）を受け付けます。また、必要に応じて、セミコロンで区切られた 1 つ以上のカスタム間隔も指定できます。カスタム間隔には、柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロまたは LLD マクロを使用できます。使用する場合、値は単一のマクロである必要があります。複数のマクロ、またはテキストと混在したマクロはサポートされていません。柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - *required*

LLD ルールのオーバーライド操作履歴

検出されたアイテムに設定される LLD ルールのオーバーライド操作履歴の値です。以下のプロパティがあります。

プロパティ	Type	説明
history	string	履歴データをどのくらいの期間保存するかを示す時間単位であるアイテムプロトタイプの履歴をオーバーライドします。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作のトレンド

LLD ルールのオーバーライド操作のトレンド値は、ディスカバリされたアイテムに設定されます。以下のプロパティがあります。

Property	Type	Description
trends	string	アイテムプロトタイプのトレンドをオーバーライドします。これは、トレンドデータをどのくらいの期間保存するかを示す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 Property behavior: - *required*

LLD ルールのオーバーライド操作の深刻度

検出されたトリガーに設定される、LLD ルールのオーバーライド操作の深刻度の値です。以下のプロパティがあります。

Property	Type	Description
severity	integer	トリガープロトタイプの深刻度をオーバーライドします。 指定可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作タグ

LLD ルールのオーバーライド操作タグオブジェクトには、検出されたオブジェクトに設定されるタグ名と値が含まれます。次のプロパティがあります。

プロパティ	Type	説明
tag	string	新しいタグ名。 プロパティの動作: - 必須
value	string	新しいタグ値。

LLD ルールの上書き操作テンプレート

検出されたホストにリンクされる、LLD ルールの上書き操作テンプレートオブジェクトです。以下のプロパティがあります。

Property	Type	Description
templateid	ID	ホストプロトタイプのリンク済みテンプレートのテンプレートを上書きします。 プロパティの動作: - 必須

LLD ルールのオーバーライド操作インベントリ

検出されたホストに設定される、LLD ルールのオーバーライド操作インベントリモードの値です。以下のプロパティがあります。

Property	Type	Description
inventory_mode	integer	ホストプロトタイプのインベントリモードをオーバーライドします。 指定可能な値: -1 - 無効; 0 - *(デフォルト)* 手動; 1 - 自動。 プロパティの動作: - 必須

discoveryrule.create

説明

`object discoveryrule.create(object/array lldRules)`

このメソッドは、新しい LLD ルールを作成することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成する LLD ルール。

標準の LLD ルールプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	LLD ルールのLLD ルールフィルター。
preprocessing	object/array	LLD ルールの前処理オプション。
lld_macro_paths	object/array	LLD ルールのlld_macro_pathオプション。
overrides	object/array	LLD ルールのオーバーライドオプション。

戻り値

(object) itemids プロパティの下で作成された LLD ルールの ID を含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールの順序と一致します。

例

LLD ルールの作成

マウントされたファイルシステムをディスカバリするための Zabbix エージェント LLD ルールを作成します。
ディスカバリされたアイテムは 30 秒ごとに更新されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Mounted filesystem discovery",
        "key_": "vfs.fs.discovery",
        "hostid": "10197",
        "type": 0,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

フィルターの使用

結果をフィルタリングするための一連の条件を持つ LLD ルールを作成します。条件は論理演算子「and」を使用してグループ化されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
```

```
        "name": " フィルター付き LLD ルール",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex2",
                    "operator": "9"
                },
                {
                    "macro": "{#MACRO3}",
                    "value": "",
                    "operator": "12"
                },
                {
                    "macro": "{#MACRO4}",
                    "value": "",
                    "operator": "13"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

マクロパスを使用した LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "LLD マクロパスを持つ LLD ルール",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.path.1"
            },
```

```
            {
                "lld_macro": "{#MACRO2}",
                "path": "$.path.2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

条件を評価するためにカスタム式を使用するフィルターを持つ LLD ルールを作成します。
LLD ルールは、”{#MACRO1}” マクロ値が正規表現”regex1” と”regex2” の両方に一致し、かつ”{#MACRO2}” の値が”regex3” または”regex4” のいずれかに一致するオブジェクトのみをディスカバリする必要があります。式 ID ”A”、”B”、”C”、”D” は任意に選択されています。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Filtered LLD rule",
        "key_": "lld",
        "hostid": "10116",
        "type": 0,
        "interfaceid": "13",
        "delay": "30s",
        "filter": {
            "evaltype": 3,
            "formula": "(A and B) and (C or D)",
            "conditions": [
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex1",
                    "formulaid": "A"
                },
                {
                    "macro": "{#MACRO1}",
                    "value": "@regex2",
                    "formulaid": "B"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex3",
                    "formulaid": "C"
                },
                {
                    "macro": "{#MACRO2}",
                    "value": "@regex4",
                    "formulaid": "D"
                }
            ]
```

```
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

カスタムクエリフィールドとヘッダーの使用

カスタムクエリフィールドとヘッダーを使用して LLD ルールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
        "name": "API HTTP agent",
        "key_": "api_discovery_rule",
        "delay": "5s",
        "url": "http://127.0.0.1?discoverer.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "elements",
                "value": "2"
            }
        ],
        "headers": [
            {
                "name": "X-Type",
                "value": "api"
            },
            {
                "name": "Authorization",
                "value": "Bearer mF_A.B5f-2.1JcM"
            }
        ],
        "allow_traps": 1,
        "trapper_hosts": "127.0.0.1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
```

```
            "28336"
        ]
    },
    "id": 1
}
```

前処理を使用した LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": " 前処理を使用したディスカバリルール",
        "key_": "lld.with.preprocessing",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 20,
                "params": "20",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

オーバーライド付きの LLD ルールの作成

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": " データベースホストの検出",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "MySQL ホストの検出",
                "step": "1",
                "stop": "1",
                "filter": {
                    "evaltype": "2",
```

```
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mysqld\\.service$"
                    },
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^mariadb\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": " データベースホスト",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ],
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ]
                }
            ]
        },
        {
            "name": "PostgreSQL ホストの検出",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": " データベースホスト",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10263"
                        }
                    ],
```

```
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

スクリプト LLD ルールの作成

スクリプト LLD ルールを使用して、シンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.lldrule",
        "hostid": "12345",
        "type": 21,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [{
            "name": "host",
            "value": "{HOST.CONN}"
        }],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

無効化期間を指定し、削除を行わない LLD ルールを作成します。

エンティティが検出されなくなった後に無効化する期間をカスタマイズし、削除を行わないように設定した LLD ルールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "lld disable after 1h",
        "key_": "lld.disable",
        "hostid": "10001",
        "type": 2,
        "lifetime_type": 1,
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "1h"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

関連項目

- LLD ルールのフィルター
- LLD マクロパス
- LLD ルールの前処理

ソース

CDiscoveryRule::create() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

説明

`object discoveryrule.delete(array lldRuleIds)`

このメソッドは、LLD ルールを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除する LLD ルールの ID。

戻り値

(`object`) `ruleids` プロパティの下にある削除された LLD ルールの ID を含むオブジェクトを返します。

例

複数の LLD ルールの削除

2 つの LLD ルールを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
```

```
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRule::delete() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

説明

`integer/array discoveryrule.get(object parameters)`

このメソッドは、指定されたパラメータに従って LLD ルールを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
itemids	ID/array	指定した ID の LLD ルールのみを返します。
groupids	ID/array	指定したグループのホストに属する LLD ルールのみを返します。
hostids	ID/array	指定したホストに属する LLD ルールのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承された LLD ルールのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用する LLD ルールのみを返します。
monitored	boolean	`true` に設定した場合、監視対象ホストに属する有効な LLD ルールのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属する LLD ルールのみを返します。
templateids	ID/array	指定したテンプレートに属する LLD ルールのみを返します。
selectDiscoveryData	query	LLD ルールのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。LLD ルールのディスカバリオブジェクトは、検出された LLD ルールを、その検出元となった LLD ルールプロトタイプに関連付けます。
selectDiscoveryRulePrototypes	query	LLD ルールに属する LLD ルールプロトタイプを含む`discoveryRulePrototypes` プロパティを返します。 `count` をサポートします。
selectFilter	query	LLD ルールで使用されるフィルターのデータを含む`filter` プロパティを返します。

パラメータ	Type	説明
selectGraphs	query	LLD ルールに属するグラフプロトタイプを含むgraphs プロパティを返します。 count をサポートします。
selectHostPrototypes	query	LLD ルールに属するホストプロトタイプを含むhostPrototypes プロパティを返します。 count をサポートします。
selectHosts	query	LLD ルールが属するホストの配列を含むhosts プロパティを返します。
selectItems	query	LLD ルールに属するアイテムプロトタイプを含むitems プロパティを返します。 count をサポートします。
selectTriggers	query	LLD ルールに属するトリガープロトタイプを含むtriggers プロパティを返します。 count をサポートします。
selectLLDMacroPaths	query	各対応するマクロに割り当てられた値へのパスと LLD マクロの一覧を含むlld_macro_paths プロパティを返します。
selectPreprocessing	query	LLD ルールの前処理オプションを含むpreprocessing プロパティを返します。
selectOverrides	query	プロトタイプオブジェクトに対して実行される上書きフィルター、条件、および操作の一覧を含むlld_rule_overrides プロパティを返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 text data type のプロパティはサポートしていません。 追加のプロパティをサポートします: host - LLD ルールが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectItems, selectGraphs, selectTriggers。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: itemid, name, key_, delay, type, status。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ホストからディスカバリルールを取得する

特定のホスト ID のすべてのディスカバリルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Network interface discovery",
            "key_": "net.if.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22444",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Discovery of network interfaces as defined in global regular expression \"Netw
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
```

```
            "state": "0",
            "error": "",
            "parameters": []
        },
        {
            "itemid": "27426",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Mounted filesystem discovery",
            "key_": "vfs.fs.discovery",
            "delay": "1h",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "22450",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "119",
            "description": "Discovery of file systems of different types as defined in global regular expr
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


フィルター条件の取得

LLD ルール「24681」の名前とそのフィルター条件を取得します。
このフィルターは「and」評価タイプを使用しているため、formula プロパティは空で、eval_formula は自動的に生成されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "24681",
            "name": "Filtered LLD rule",
            "filter": {
                "evaltype": "1",
                "formula": "",
                "conditions": [
                    {
                        "macro": "{#MACRO1}",
                        "value": "@regex1",
                        "operator": "8",
                        "formulaid": "A"
                    },
                    {
                        "macro": "{#MACRO2}",
                        "value": "@regex2",
                        "operator": "9",
                        "formulaid": "B"
                    },
                    {
                        "macro": "{#MACRO3}",
                        "value": "",
                        "operator": "12",
                        "formulaid": "C"
                    },
                    {
                        "macro": "{#MACRO4}",
                        "value": "",
                        "operator": "13",
                        "formulaid": "D"
                    }
                ],
                "eval_formula": "A and B and C and D"
            }
        }
    ],
    "id": 1
}
```

URL による LLD ルールの取得

ホストの LLD ルールを、ルールの URL フィールド値によって取得します。サポートされるのは、LLD ルールに定義された URL 文字列との完全一致のみです。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
```

```
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28336",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10257",
            "name": "API HTTP agent",
            "key_": "api_discovery_rule",
            "delay": "5s",
            "status": "0",
            "trapper_hosts": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "1",
            "interfaceid": "5",
            "description": "",
            "lifetime": "30d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "http://127.0.0.1/discoverer.php",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "json"
                },
                {
                    "name": "elements",
                    "value": "2"
                }
            ],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [
                {
                    "name" : "X-Type",
                    "value": "api"
                },
                {
```

```
                    "name": "Authorization",
                    "value": "Bearer mF_A.B5f-2.1JcM"
                }
            ],
            "retrieve_mode": "0",
            "request_method": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "state": "0",
            "error": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


オーバーライドを持つ LLD ルールの取得

さまざまなオーバーライド設定を持つ LLD ルールを 1 つ取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "Discover database host",
            "overrides": [
                {
                    "name": "Discover MySQL host",
                    "step": "1",
                    "stop": "1",
                    "filter": {
                        "evaltype": "2",
                        "formula": "",
                        "conditions": [
                            {
                                "macro": "{#UNIT.NAME}",
                                "operator": "8",
                                "value": "^mysqld\\.service$",
                                "formulaid": "A"
                            },
                            {
                                "macro": "{#UNIT.NAME}",
```

```
                    "operator": "8",
                    "value": "^mariadb\\.service$",
                    "formulaid": "B"
                }
            ],
            "eval_formula": "A or B"
        },
        "operations": [
            {
                "operationobject": "3",
                "operator": "2",
                "value": "Database host",
                "opstatus": {
                    "status": "0"
                },
                "optag": [
                    {
                        "tag": "database",
                        "value": "mysql"
                    }
                ],
                "optemplate": [
                    {
                        "templateid": "10170"
                    }
                ]
            }
        ]
    },
    {
        "name": "Discover PostgreSQL host",
        "step": "2",
        "stop": "1",
        "filter": {
            "evaltype": "0",
            "formula": "",
            "conditions": [
                {
                    "macro": "{#UNIT.NAME}",
                    "operator": "8",
                    "value": "^postgresql\\.service$",
                    "formulaid": "A"
                }
            ],
            "eval_formula": "A"
        },
        "operations": [
            {
                "operationobject": "3",
                "operator": "2",
                "value": "Database host",
                "opstatus": {
                    "status": "0"
                },
                "optag": [
                    {
                        "tag": "database",
                        "value": "postgresql"
                    }
                ],
                "optemplate": [
                    {
```

```
                                "templateid": "10263"
                            }
                        ]
                    }
                ]
            }
        ]
    }
    ],
    "id": 1
}
```

関連項目

- グラフプロトタイプ
- ホスト
- アイテムプロトタイプ
- LLD ルールフィルター
- トリガープロトタイプ

ソース

CDiscoveryRule::get() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

説明

`object discoveryrule.update(object/array lldRules)`

このメソッドでは、既存の LLD ルールを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 更新する LLD ルールのプロパティ。

各 LLD ルールでは `itemid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準の LLD ルールプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	既存のフィルターを置き換えるためのLLD ルールフィルター。
preprocessing	object/array	既存の前処理オプションを置き換えるためのLLD ルール前処理オプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
lld_macro_paths	object/array	既存の lld_macro_path オプションを置き換えるための LLD ルールのlld_macro_pathオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
overrides	object/array	既存のオーバーライドオプションを置き換えるためのLLD ルールオーバーライドオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*

戻り値

(object) itemids プロパティの下で更新された LLD ルールの ID を含むオブジェクトを返します。

例

LLD ルールへのフィルターの追加

{#FSTYPE} マクロの内容が *@File systems for discovery* 正規表現に一致するように、フィルターを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

LLD マクロパスの追加

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
```

```
    "id": 1
}
```

トラップの無効化

ディスカバリルールの LLD トラップを無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

LLD ルールの前処理オプションの更新

前処理ルール「JSONPath」を使用して LLD ルールを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.path.to.json",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

LLD ルールのスクリプトの更新

LLD ルールのスクリプトを別のスクリプトで更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

LLD ルールの有効期間の更新

LLD ルールを更新し、検出されなくなったエンティティを 12 時間後に無効化し、7 日後に削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRule::update() in *ui/include/classes/api/services/CDiscoveryRule.php*.

LLD ルールのプロトタイプ

このクラスは、低レベルディスカバリルールのプロトタイプを扱うために設計されています。

オブジェクト参照:

- LLD ルールのプロトタイプ
  - HTTP ヘッダー

    - HTTP クエリフィールド
- LLD ルールのプロトタイプフィルター
    - LLD ルールのプロトタイプフィルター条件
- LLD プロトタイプマクロパス
- LLD ルールのプロトタイプ前処理
- LLD ルールのプロトタイプ上書き
    - LLD ルールのプロトタイプ上書きフィルター
        * LLD ルールのプロトタイプ上書きフィルター条件
    - LLD ルールのプロトタイプ上書き操作
        * LLD ルールのプロトタイプ上書き操作ステータス
        * LLD ルールのプロトタイプ上書き操作ディスカバリ
        * LLD ルールのプロトタイプ上書き操作期間
        * LLD ルールのプロトタイプ上書き操作履歴
        * LLD ルールのプロトタイプ上書き操作トレンド
        * LLD ルールのプロトタイプ上書き操作重大度
        * LLD ルールのプロトタイプ上書き操作タグ
        * LLD ルールのプロトタイプ上書き操作テンプレート
        * LLD ルールのプロトタイプ上書き操作インベントリ

利用可能なメソッド:

- discoveryruleprototype.create - 新しい LLD ルールのプロトタイプを作成
- discoveryruleprototype.delete - LLD ルールのプロトタイプを削除
- discoveryruleprototype.get - LLD ルールのプロトタイプを取得
- discoveryruleprototype.update - LLD ルールのプロトタイプを更新

LLD ルールプロトタイプオブジェクト

以下のオブジェクトは、`discoveryruleprototype` API に直接関連しています。

LLD ルールプロトタイプ

低レベルディスカバリルールプロトタイプオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	LLD ルールプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
ruleid	ID	親 LLD ルール/LLD ルールプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 作成操作で 必須

プロパティ	型	説明
delay	string	LLD ルールプロトタイプの更新間隔。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）を受け付け、必要に応じて 1 つ以上のカスタム間隔も指定できます。これらはすべてセミコロンで区切ります。カスタム間隔には、柔軟な間隔とスケジュール間隔を混在させることができます。 ユーザーマクロを受け付けます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはテキストと混在したマクロはサポートされません。柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - `type` が”Zabbix agent” (0)、”Simple check” (3)、”Zabbix internal” (5)、”External check” (10)、”Database monitor” (11)、”IPMI agent” (12)、”SSH agent” (13)、”TELNET agent” (14)、”JMX agent” (16)、”HTTP agent” (19)、”SNMP agent” (20)、”Script” (21)、”Browser” (22) に設定されている場合、または `type` が”Zabbix agent (active)” (7) に設定され、かつ `key_` に”mqtt.get” が含まれない場合は 必須
hostid	ID	LLD ルールプロトタイプが属するホストの ID。 プロパティの動作: - 定数 - 作成操作で 必須
flags	integer	LLD ルールプロトタイプの生成元。 指定可能な値: 3 - 低レベルディスカバリルールプロトタイプ; 7 - ディスカバリされた低レベルディスカバリルールプロトタイプ。 プロパティの動作: - 読み取り専用
interfaceid	ID	LLD ルールプロトタイプのホストインターフェースの ID。 プロパティの動作: - LLD ルールプロトタイプがホストに属し、`type` が”Zabbix agent”、”IPMI agent”、”JMX agent”、または”SNMP agent” に設定されている場合は 必須 - LLD ルールプロトタイプがホストに属し、`type` が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合は サポート
key_	string	LLD ルールプロトタイプキー。少なくとも 1 つの LLD マクロが必要です。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
name	string	LLD ルールプロトタイプの名前。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用

プロパティ	型	説明
type	integer	LLD ルールプロトタイプのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が”HTTP agent” に設定されている場合は 必須 - 継承されたオブジェクトでは 読み取り専用
allow_traps	integer	trapper アイテムと同様に値を設定できるようにします。 指定可能な値: 0 - *(デフォルト)* 受信データの受け入れを許可しない; 1 - 受信データの受け入れを許可する。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート
authtype	integer	認証方式。 type が”SSH agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM。 プロパティの動作: - type が”SSH agent” または”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用（type が”HTTP agent” に設定されている場合）
description	string	LLD ルールプロトタイプの説明。
error	string	LLD ルールプロトタイプの値の更新に問題がある場合のエラーテキスト。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
follow_redirects	integer	データ取得時にレスポンスのリダイレクトに従います。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が"IPMI agent" に設定され、かつ key_ が"ipmi.get" に設定されていない場合は 必須 - type が"IPMI agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - type が"JMX agent" に設定されている場合は サポート
lifetime	string	ディスカバリされなくなったアイテムが削除されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを受け付けます。 デフォルト: 7d。
lifetime_type	integer	失われた LLD リソースを削除するシナリオ。 指定可能な値: 0 - *(デフォルト)* lifetime しきい値に達した後に削除; 1 - 削除しない; 2 - 即時削除。
enabled_lifetime	string	ディスカバリされなくなったアイテムが無効化されるまでの期間。秒、サフィックス付きの時間単位、またはユーザーマクロを受け付けます。 デフォルト: 0。
enabled_lifetime_type	integer	失われた LLD リソースを無効化するシナリオ。 指定可能な値: 0 - lifetime しきい値に達した後に無効化; 1 - 無効化しない; 2 - *(デフォルト)* 即時無効化。
master_itemid	ID	マスターアイテムの ID。 ディスカバリルールを別のディスカバリルールのマスターアイテムにすることはできません。 プロパティの動作: - type が"Dependent item" に設定されている場合は 必須 - 継承されたオブジェクトでは 読み取り専用

プロパティ	型	説明
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データのまま保存; 1 - JSON に変換。 プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
params	string	LLD ルールプロトタイプのタイプに応じた追加パラメータ: - SSH および Telnet LLD ルール用の実行スクリプト; - データベースモニタ LLD ルール用の SQL クエリ; - 計算 LLD ルール用の数式; - Script および Browser LLD ルール用のスクリプト。 プロパティの動作: - type が"Database monitor"、"SSH agent"、"TELNET agent"、"Script"、または"Browser" に設定されている場合は 必須 - 継承されたオブジェクトでは 読み取り専用（type が"Script" または"Browser" に設定されている場合）
parameters	object/array	type が"Script" または"Browser" に設定されている場合の追加パラメータ。 name および value プロパティを持つオブジェクトの配列で、name は一意である必要があります。 プロパティの動作: - type が"Script" または"Browser" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
password	string	認証用パスワード。 プロパティの動作: - type が"JMX agent" に設定され、username が設定されている場合は 必須 - type が"Simple check"、"Database monitor"、"SSH agent"、"TELNET agent"、または"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用（type が"HTTP agent" に設定されている場合）
post_type	integer	posts プロパティに格納される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
posts	string	HTTP(S) リクエスト本文データ。 プロパティの動作: - type が"HTTP agent" に設定され、post_type が"JSON data" または"XML data" に設定されている場合は 必須 - type が"HTTP agent" に設定され、post_type が"Raw data" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - type が"SSH agent" に設定され、authtype が"public key" に設定されている場合は 必須

プロパティ	型	説明
publickey	string	公開鍵ファイル名。 プロパティの動作: - type が”SSH agent” に設定され、authtype が”public key” に設定されている場合は 必須
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 request_method が”GET”、”POST”、または”PUT” に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 request_method が”HEAD” に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - type が”SNMP agent” に設定されている場合は 必須 - 継承されたオブジェクトでは 読み取り専用
ssl_cert_file	string	公開 SSL 鍵ファイルのパス。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
ssl_key_file	string	秘密 SSL 鍵ファイルのパス。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
ssl_key_password	string	SSL 鍵ファイルのパスワード。 プロパティの動作: - type が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用

プロパティ	型	説明
state	integer	LLD ルールプロトタイプの状態。 指定可能な値: 0 - *(デフォルト)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	LLD ルールプロトタイプのステータス。 指定可能な値: 0 - *(デフォルト)* 有効な LLD ルールプロトタイプ; 1 - 無効な LLD ルールプロトタイプ。
status_codes	string	必要な HTTP ステータスコードの範囲。カンマ区切りで指定します。カンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - `type` が”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
templateid	ID	親テンプレート LLD ルールプロトタイプの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒、またはサフィックス付きの時間単位（例: 30s、1m）を受け付けます。 ユーザーマクロも受け付けます。 指定可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの動作: - `type` が”Zabbix agent” (0)、”Simple check” (3) で `key_` が”`vmware.`” および”`icmpping`” で始まらない場合、”Zabbix agent (active)” (7)、”External check” (10)、”Database monitor” (11)、”SSH agent” (13)、”TELNET agent” (14)、”HTTP agent” (19)、”SNMP agent” (20) で `snmp_oid` が”`walk[`” または”`get[`” で始まる場合、”Script” (21)、”Browser” (22) に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - `type` が”Zabbix trapper” に設定されている場合、または `type` が”HTTP agent” に設定され、`allow_traps` が”Allow to accept incoming data” に設定されている場合は サポート
username	string	認証用ユーザー名。 プロパティの動作: - `type` が”SSH agent”、”TELNET agent” に設定されている場合、または `type` が”JMX agent” に設定され、`password` が設定されている場合は 必須 - `type` が”Simple check”、”Database monitor”、または”HTTP agent” に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用（`type` が”HTTP agent” に設定されている場合）

プロパティ	型	説明
uuid	string	既存の LLD ルールプロトタイプとインポートされた LLD ルールプロトタイプを関連付けるために使用される汎用一意識別子。指定しない場合は自動生成されます。
verify_host	integer	プロパティの動作: - LLD ルールプロトタイプがテンプレートに属する場合は サポート 接続先のホスト名がホスト証明書内の名前と一致するかどうかを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。
verify_peer	integer	プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用 ホストの証明書が正当なものかどうかを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - type が"HTTP agent" に設定されている場合は サポート - 継承されたオブジェクトでは 読み取り専用

HTTP ヘッダー

ヘッダーオブジェクトには、次のプロパティがあります。

Property	Type	Description
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

LLD ルールプロトタイプフィルター

LLD ルールプロトタイプフィルターオブジェクトは、検出されたオブジェクトをフィルタリングするために使用できる一連の条件を定義します。次のプロパティがあります。

Property	Type	Description
conditions	object/array	結果のフィルタリングに使用するフィルター条件のセット。条件は、formula 内で配置された順序で並べ替えられます。 Property behavior: - 必須
evaltype	integer	フィルター条件の評価方法。 使用可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - 必須
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、各フィルター条件をその `formulaid` で参照する ID が含まれます。`eval_formula` の値は、カスタム式を持つフィルターでは `formula` の値と同じです。 Property behavior: - 読み取り専用
formula	string	カスタム式を持つフィルターの条件評価に使用する、ユーザー定義の式。この式には、各フィルター条件をその `formulaid` で参照する ID が含まれている必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致している必要があります。未使用または省略された条件があってはなりません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は必須

LLD ルールプロトタイプのフィルター条件

LLD ルールプロトタイプのフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。以下のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する対象の LLD マクロ。 Property behavior: - *required*
value	string	比較する値。 Property behavior: - `operator` が「正規表現に一致する」または「正規表現に一致しない」に設定されている場合は *required*
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際は、この ID をユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。 Property behavior: - LLD rule prototype filter object の `evaltype` が「カスタム式」に設定されている場合は *required*

プロパティ	Type	説明
operator	integer	条件演算子。 指定可能な値: 8 - *(default)* 正規表現に一致する; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、discoveryruleprototype.get および discoveryruleprototype.create メソッドのページにある例を参照してください。

LLD マクロパス

LLD マクロパスには、以下のプロパティがあります。

プロパティ	Type	説明
lld_macro	string	LLD マクロ。 プロパティの動作: - 必須
path	string	対応するマクロに割り当てられる値のセレクター。 プロパティの動作: - 必須

LLD ルールプロトタイプの前処理

LLD ルールプロトタイプの前処理オブジェクトには、以下のプロパティがあります。

Property	Type	Description
type	integer	前処理オプションのタイプ。 指定可能な値: 5 - 正規表現; 11 - XML XPath; 12 - JSONPath; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON 内のエラーをチェック; 17 - XML 内のエラーをチェック; 20 - ハートビート付きで未変更を破棄; 21 - JavaScript; 23 - Prometheus から JSON; 24 - CSV から JSON; 25 - 置換; 27 - XML から JSON; 28 - SNMP walk 値; 29 - SNMP walk から JSON; 30 - SNMP get 値。 プロパティの動作: - 必須

Property	Type	Description
params	string	前処理オプションで使用される追加パラメータ。複数のパラメータは改行文字 (\n) で区切ります。 プロパティの動作: - `type` が「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON 内のエラーをチェック」(16)、「XML 内のエラーをチェック」(17)、「ハートビート付きで未変更を破棄」(20)、「JavaScript」(21)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「置換」(25)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler	integer	前処理ステップの失敗時に使用されるアクションタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix サーバーによって設定される; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 プロパティの動作: - `type` が「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON 内のエラーをチェック」(16)、「XML 内のエラーをチェック」(17)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「XML から JSON」(27)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は 必須
error_handler_params	string	エラーハンドラーパラメータ。 プロパティの動作: - `error_handler` が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は 必須

以下のパラメータとエラーハンドラーが、各前処理タイプでサポートされています。

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
5	正規表現	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	正規表現に一致	pattern[1]			0, 1, 2, 3
15	正規表現に一致しない	pattern[1]			0, 1, 2, 3
16	JSON 内のエラーをチェック	path[3]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
17	XML 内のエラーをチェック	path[3]			0, 1, 2, 3
20	ハートビート付きで未変更を破棄	seconds[4, 5]			
21	JavaScript	script[2]			
23	Prometheus から JSON	pattern[5, 6]			0, 1, 2, 3
24	CSV から JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
27	XML から JSON				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON[7]	Field name[2]	OID prefix[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 正規表現
[2] 文字列
[3] JSONPath または XML XPath
[4] 正の整数（時間サフィックスをサポート。例: 30s, 1m, 2h, 1d）
[5] ユーザーマクロ
[6] 以下の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。各 Prometheus パターンの構成要素（メトリクス、ラベル名、ラベル値、メトリクス値）には、ユーザーマクロを使用できます。
[7] 改行文字で区切られた複数の「Field name,OID prefix,Format records」レコードをサポートします。

LLD ルールのプロトタイプ上書き

LLD ルールのプロトタイプ上書きオブジェクトは、異なるプロトタイプオブジェクトのプロパティを上書きするために使用される一連のルール（フィルター、条件、操作）を定義します。以下のプロパティがあります。

Property	Type	Description
name	string	一意の上書き名。 プロパティの動作: - 必須
step	integer	上書きの一意の順序番号。 プロパティの動作: - 必須
stop	integer	一致した場合、次の上書きの処理を停止します。 指定可能な値: 0 - *(デフォルト)* 上書きの処理を停止しない; 1 - フィルターが一致した場合に上書きの処理を停止する。
filter	object	上書きフィルター。
operations	object/array	上書き操作。

LLD ルールプロトタイプのオーバーライドフィルター

LLD ルールプロトタイプのオーバーライドフィルターオブジェクトは、検出されたオブジェクトに一致した場合にオーバーライドを適用する一連の条件を定義します。以下のプロパティがあります。

Property	Type	Description
conditions	object/array	検出されたオブジェクトとの照合に使用するオーバーライドフィルター条件のセット。条件は、formula 内での配置順に並べ替えられます。 Property behavior: - 必須
evaltype	integer	オーバーライドフィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - 必須
eval_formula	string	オーバーライドフィルター条件の評価に使用される生成済みの式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID が含まれます。カスタム式を使用するフィルターでは、`eval_formula` の値は `formula` の値と同じです。 Property behavior: - 読み取り専用
formula	string	カスタム式を使用するオーバーライドフィルターの条件評価に使用する、ユーザー定義の式。この式には、各オーバーライドフィルター条件をその `formulaid` で参照する ID を含める必要があります。式で使用する ID は、オーバーライドフィルター条件で定義されたものと完全に一致していなければなりません。未使用または省略された条件があってはなりません。 Property behavior: - `evaltype` が「custom expression」に設定されている場合は必須

LLD ルールプロトタイプのオーバーライドフィルター条件

LLD ルールプロトタイプのオーバーライドフィルター条件オブジェクトは、LLD マクロの値に対して実行する個別のチェックを定義します。以下のプロパティがあります。

プロパティ	Type	説明
macro	string	チェックを実行する対象の LLD マクロ。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - `operator` が「正規表現に一致」または「正規表現に一致しない」に設定されている場合は 必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際は、この ID をユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。 プロパティの動作: - LLD ルールプロトタイプのオーバーライドフィルターオブジェクト の `evaltype` が「カスタム式」に設定されている場合は 必須
operator	integer	条件演算子。 指定可能な値: 8 - *(デフォルト)* 正規表現に一致; 9 - 正規表現に一致しない; 12 - 存在する; 13 - 存在しない。

LLD ルールのプロトタイプ上書き操作

LLD ルールのプロトタイプ上書き操作は、プロトタイプオブジェクトに対して実行する条件とアクションの組み合わせです。次のプロパティがあります。

Property	Type	Description
operationobject	integer	アクションを実行する検出済みオブジェクトのタイプ。 使用可能な値: 0 - アイテムのプロトタイプ; 1 - トリガーのプロトタイプ; 2 - グラフのプロトタイプ; 3 - ホストのプロトタイプ。 プロパティの動作: - 必須
operator	integer	上書き条件の演算子。 使用可能な値: 0 - *(デフォルト)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 8 - 一致する; 9 - 一致しない。
value	string	選択したオブジェクトに応じて、アイテム、トリガー、グラフ、またはホストのプロトタイプ名に一致させるパターン。
opstatus	object	アイテム、トリガー、およびホストのプロトタイプオブジェクト用の上書き操作ステータスオブジェクト。
opdiscover	object	上書き操作の検出ステータスオブジェクト（すべてのオブジェクトタイプ）。
opperiod	object	アイテムのプロトタイプオブジェクト用の上書き操作期間（更新間隔）オブジェクト。
ophistory	object	アイテムのプロトタイプオブジェクト用の上書き操作履歴オブジェクト。

Property	Type	Description
optrends	object	アイテムのプロトタイプオブジェクト用の上書き操作トレンドオブジェクト。
opseverity	object	トリガーのプロトタイプオブジェクト用の上書き操作深刻度オブジェクト。
optag	object/array	トリガーおよびホストのプロトタイプオブジェクト用の上書き操作タグオブジェクト。
optemplate	object/array	ホストのプロトタイプオブジェクト用の上書き操作テンプレートオブジェクト。
opinventory	object	ホストのプロトタイプオブジェクト用の上書き操作インベントリオブジェクト。

LLD ルールのプロトタイプ上書き操作ステータス

検出されたオブジェクトに設定される、LLD ルールのプロトタイプ上書き操作ステータスです。
以下のプロパティがあります。

Property	Type	Description
status	integer	選択したオブジェクトのステータスを上書きします。 指定可能な値: 0 - 有効で作成; 1 - 無効で作成。 プロパティの動作: *- required*

LLD ルールのプロトタイプ上書き操作 discover

LLD ルールのプロトタイプ上書き操作 discover は、発見済みオブジェクトに設定される discover ステータスです。以下のプロパティがあります。

Property	Type	Description
discover	integer	選択したオブジェクトの discover ステータスを上書きします。 指定可能な値: 0 - はい。オブジェクトの検出を継続します。 1 - いいえ。新しいオブジェクトは検出されず、既存のオブジェクトは失われたものとしてマークされます。 プロパティの動作: - 必須

LLD ルールのプロトタイプ上書き操作期間

LLD ルールのプロトタイプ上書き操作期間は、検出されたアイテムに設定される更新間隔の値です。以下のプロパティがあります。

プロパティ	型	説明
delay	string	アイテムプロトタイプの更新間隔を上書きします。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）を受け付けます。また、必要に応じて、セミコロンで区切られた 1 つ以上のカスタム間隔も指定できます。カスタム間隔には、柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロまたは LLD マクロを使用できます。使用する場合、値は単一のマクロである必要があります。複数のマクロ、またはテキストと混在したマクロはサポートされていません。柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます（例: `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`）。 例: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` プロパティの動作: - 必須

LLD ルールのプロトタイプ上書き操作履歴

検出されたアイテムに設定される LLD ルールのプロトタイプ上書き操作履歴の値です。以下のプロパティがあります。

プロパティ	Type	説明
history	string	アイテムプロトタイプの履歴を上書きします。これは履歴データをどのくらいの期間保存するかを示す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: - 必須

LLD ルールのプロトタイプ上書き操作のトレンド

LLD ルールのプロトタイプ上書き操作のトレンド値は、検出されたアイテムに設定されます。以下のプロパティがあります。

プロパティ	Type	説明
trends	string	アイテムのプロトタイプのトレンドを上書きします。これは、トレンドデータを保存する期間を表す時間単位です。ユーザーマクロおよび LLD マクロも使用できます。 プロパティの動作: *- required*

LLD ルールのプロトタイプ上書き操作の深刻度

検出されたトリガーに設定される、LLD ルールのプロトタイプ上書き操作の深刻度の値です。以下のプロパティがあります。

Property	Type	Description
severity	integer	トリガープロトタイプの深刻度を上書きします。 指定可能な値: 0 - *(default)* 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 Property behavior: - *required*

LLD ルールのプロトタイプ上書き操作タグ

LLD ルールのプロトタイプ上書き操作タグオブジェクトには、ディスカバリされたオブジェクトに設定されるタグ名と値が含まれます。以下のプロパティがあります。

プロパティ	Type	説明
tag	string	新しいタグ名。 プロパティの動作: - 必須
value	string	新しいタグ値。

LLD ルールのプロトタイプ上書き操作テンプレート

検出されたホストにリンクされる、LLD ルールのプロトタイプ上書き操作テンプレートオブジェクトです。以下のプロパティがあります。

プロパティ	Type	説明
templateid	ID	ホストプロトタイプのリンク済みテンプレートのテンプレートを上書きします。 プロパティの動作: - 必須

LLD ルールのプロトタイプ上書き操作インベントリ

検出されたホストに設定される、LLD ルールのプロトタイプ上書き操作インベントリモードの値です。以下のプロパティがあります。

プロパティ	Type	説明
inventory_mode	integer	ホストプロトタイプのインベントリモードを上書きします。 指定可能な値: -1 - 無効; 0 - *(デフォルト)* 手動; 1 - 自動。 プロパティの動作: - 必須

discoveryruleprototype.create

説明

`object discoveryruleprototype.create(object/array lldRules)`

このメソッドは、新しい LLD ルールのプロトタイプを作成します。

Note:
このメソッドは、管理者および スーパー管理者のユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

パラメータ

(object/array) 作成する LLD ルールのプロトタイプ。

標準の LLD ルールプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	LLD ルール用のLLD ルールプロトタイプフィルター。
preprocessing	object/array	LLD ルールプロトタイプの前処理オプション。
lld_macro_paths	object/array	LLD ルールプロトタイプのlld_macro_pathオプション。
overrides	object/array	LLD ルールプロトタイプのオーバーライドオプション。

戻り値

(object) itemids プロパティの下で作成された LLD ルールプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡された LLD ルールプロトタイプの順序と一致します。

例

LLD ルールのプロトタイプの作成

データベースインスタンス内の表領域を検出するための LLD ルールのプロトタイプ（タイプ：ネスト）を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
        "name": "Discover tablespaces for {#DB}",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

関連項目

- LLD ルールのプロトタイプフィルター
- LLD マクロパス
- LLD ルールのプロトタイプ前処理

ソース

CDiscoveryRulePrototype::create() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

説明

`object discoveryruleprototype.delete(array lldRuleIds)`

このメソッドは、LLD ルールのプロトタイプを削除します。

> **Note:**
> このメソッドは、管理者およびスーパー管理者のユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

パラメーター

(array) 削除する LLD ルールプロトタイプの ID

戻り値

(object) ruleids プロパティの下にある削除された LLD ルールプロトタイプの ID を含むオブジェクトを返します。

例

複数の LLD ルールプロトタイプの削除

2 つの LLD ルールプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRulePrototype::delete() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

説明

`integer/array discoveryruleprototype.get(object parameters)`

このメソッドは、指定されたパラメータに従って LLD ルールのプロトタイプを取得します。

> **Note:**
> このメソッドは、あらゆるタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
itemids	ID/array	指定した ID を持つ LLD ルールのプロトタイプのみを返します。
groupids	ID/array	指定したグループのホストに属する LLD ルールのプロトタイプのみを返します。
hostids	ID/array	指定したホストに属する LLD ルールのプロトタイプのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承された LLD ルールのプロトタイプのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用する LLD ルールのプロトタイプのみを返します。
monitored	boolean	`true` に設定した場合、監視対象ホストに属する有効な LLD ルールのプロトタイプのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属する LLD ルールのプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属する LLD ルールのプロトタイプのみを返します。
selectDiscoveryRule	query	LLD ルールのプロトタイプの親 LLD ルールを含む`discoveryRule` プロパティを返します。
selectDiscoveryRulePrototype	query	LLD ルールのプロトタイプの親 LLD ルールプロトタイプを含む`discoveryRulePrototype` プロパティを返します。
selectDiscoveryRulePrototypes	query	LLD ルールのプロトタイプに属する子 LLD ルールのプロトタイプを含む`discoveryRulePrototypes` プロパティを返します。 `count` をサポートします。
selectFilter	query	LLD ルールのプロトタイプで使用されるフィルタのデータを含む`filter` プロパティを返します。
selectGraphs	query	LLD ルールのプロトタイプに属するグラフのプロトタイプを含む`graphs` プロパティを返します。 `count` をサポートします。
selectHostPrototypes	query	LLD ルールのプロトタイプに属するホストのプロトタイプを含む`hostPrototypes` プロパティを返します。 `count` をサポートします。
selectHosts	query	LLD ルールのプロトタイプが属するホストの配列を含む`hosts` プロパティを返します。
selectItems	query	LLD ルールのプロトタイプに属するアイテムのプロトタイプを含む`items` プロパティを返します。 `count` をサポートします。
selectTriggers	query	LLD ルールのプロトタイプに属するトリガーのプロトタイプを含む`triggers` プロパティを返します。 `count` をサポートします。
selectLLDMacroPaths	query	各対応するマクロに割り当てられた値へのパスと LLD マクロの一覧を含む`lld_macro_paths` プロパティを返します。
selectPreprocessing	query	LLD ルールのプロトタイプの前処理オプションを含む`preprocessing` プロパティを返します。
selectOverrides	query	プロトタイプオブジェクトに対して実行される上書きフィルタ、条件、および操作の一覧を含む`lld_rule_overrides` プロパティを返します。
filter	object	指定したフィルタに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしていません。 追加のプロパティをサポートします: `host` - LLD ルールのプロトタイプが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectItems`, `selectGraphs`, `selectTriggers`。

パラメータ	型	説明
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `itemid`, `name`, `key_`, `delay`, `type`, `status`。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ホストから検出ルールのプロトタイプを取得

特定のホスト ID のすべての検出ルールのプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Discover tablespaces for {#DB}",
            "key_": "db.tablespace.discovery[{#DB}]",
            "delay": "0",
            "history": "31d",
            "trends": "365d",
            "status": "0",
            "value_type": "4",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
```

```
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "3",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```


関連項目

- グラフプロトタイプ
- ホスト
- アイテムプロトタイプ
- LLD ルールプロトタイプフィルター
- トリガープロトタイプ

ソース

CDiscoveryRulePrototype::get() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

説明

`object discoveryruleprototype.update(object/array lldRules)`

このメソッドは、既存の LLD ルールのプロトタイプを更新します。

既に検出されたプロトタイプの更新には制限があることに注意してください。

> **Note:**
> このメソッドは、管理者およびスーパー管理者のユーザータイプでのみ使用できます。メソッドの呼び出し権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object/array) 更新する LLD ルールプロトタイプのプロパティ。

各 LLD ルールプロトタイプでは itemid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

標準の LLD ルールプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	既存のフィルターを置き換えるためのLLD ルールプロトタイプフィルター。
preprocessing	object/array	既存の前処理オプションを置き換えるためのLLD ルールプロトタイプの前処理オプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
lld_macro_paths	object/array	既存の lld_macro_path オプションを置き換えるための、LLD ルールプロトタイプのlld_macro_pathオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
overrides	object/array	既存のオーバーライドオプションを置き換えるためのLLD ルールプロトタイプのオーバーライドオプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*

戻り値

(object) itemids プロパティの下で更新された LLD ルールプロトタイプの ID を含むオブジェクトを返します。

例

LLD ルールプロトタイプの前処理オプションの更新

JSONPath 前処理ルールを使用して、LLD ルールプロトタイプを更新します。リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

ソース

CDiscoveryRulePrototype::update() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

MFA

このクラスは、MFA(多要素認証) メソッドを操作するためのものです。

オブジェクトリファレンス:

- MFA

利用可能なメソッド:

- mfa.create - 新しい MFA メソッドを作成する
- mfa.delete - MFA メソッドを削除する
- mfa.get - MFA メソッドを取得する
- mfa.update - MFA メソッドを更新する

MFA オブジェクト

次のオブジェクトは、`mfaAPI` に直接関連しています。

MFA

MFA（多要素認証）オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
mfaid	ID	MFA 方式の ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
type	integer	MFA 方式のタイプ。 指定可能な値: 1 - TOTP（Time-based One-Time Passwords）; 2 - Duo Universal Prompt。
name	string	MFA 方式の一意な名前。 プロパティの動作: - 作成操作では 必須
hash_function	integer	TOTP コードの生成に使用するハッシュ関数のタイプ。 指定可能な値: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512。 プロパティの動作: - `type` が「TOTP」に設定されている場合は 必須
code_length	integer	認証コードの長さ。 指定可能な値: 6 - 6 桁; 8 - 8 桁。 プロパティの動作: - `type` が「TOTP」に設定されている場合は 必須
api_hostname	string	Duo 認証サービスから提供される API ホスト名。 プロパティの動作: - `type` が「Duo Universal Prompt」に設定されている場合は 必須
clientid	string	Duo 認証サービスから提供されるクライアント ID。 プロパティの動作: - `type` が「Duo Universal Prompt」に設定されている場合は 必須

プロパティ	Type	説明
client_secret	string	Duo 認証サービスから提供されるクライアントシークレット。 プロパティの動作: - 書き込み専用 - type が「Duo Universal Prompt」に設定されている場合は 必須

mfa.create

説明

`object mfa.create(object/array MFA methods)`

このメソッドは、新しい MFA メソッドを作成します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロールをご覧ください。

パラメータ

(object/array) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受けつけます。

戻り値

(object) mfaids プロパティの下で作成された MFA メソッドの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムの順序と一致します。

例

MFA メソッドの作成

時間ベースのワンタイムパスワード (TOTP) を利用した”Zabbix TOTP” MFA メソッドを作成します。TOTP コード生成用のハッシュ関数は SHA-1 に設定し、検証コードの長さは 6 桁に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

```
{
"jsonrpc": "2.0",
"method": "mfa.create",
"params": {
```

```
"type": 1,
"name": "Zabbix TOTP",
"hash_function": 1,
"code_length": 6
},
"id": 1
}
```

レスポンス:

```
{
"jsonrpc": "2.0",
"result": {
"mfaids": [
"1"
]
},
"id": 1
}
```

ソース

CMfa::create() in *ui/include/classes/api/services/CMfa.php*.

mfa.delete

説明

`object mfa.delete(array mfaids)`

このメソッドは、MFA メソッドを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する MFA メソッドの ID。

戻り値

(object) mfaids プロパティの下にある削除された MFA メソッドの ID を含むオブジェクトを返します。

例

MFA メソッドの削除

MFA メソッドを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
```

```
    },
    "id": 1
}
```

ソース

CMfa::delete() in *ui/include/classes/api/services/CMfa.php*.

mfa.get

説明

`integer/array mfa.get(object parameters)`

このメソッドは、与えられたパラメータに基づいて MFA メソッドを取得します。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
mfaids	ID/array	指定した ID を持つ MFA メソッドのみを返します。
selectUsrgrps	query	MFA メソッドに関連付けられたユーザーグループ を含む `usrgrps` プロパティを返します。 `count` をサポートします。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 サポートされるプロパティ: `mfaid` - MFA メソッドの ID。 `type` - MFA メソッドのタイプ。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `name`。
search	object	指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索を実行します。 サポートされるプロパティ: `name`。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

名前による MFA メソッドの検索

名前に”Zabbix” が含まれるすべての MFA メソッドを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mfaid": "1",
            "type": "1",
            "name": "Zabbix TOTP 1",
            "hash_function": "1",
            "code_length": "6",
            "api_hostname": "",
            "clientid": ""
        },
        {
            "mfaid": "2",
            "type": "1",
            "name": "Zabbix TOTP 2",
            "hash_function": "3",
            "code_length": "8",
            "api_hostname": "",
            "clientid": ""
        }
    ],
    "id": 1
}
```

ソース

CMfa::get() in *ui/include/classes/api/services/CMfa.php*.

mfa.update

説明

`object mfa.update(object/array MFA methods)`

このメソッドは、既存の MFA メソッドを更新します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割を参照してください。

パラメータ

(object/array) 更新する MFA メソッドのプロパティ

mfaid プロパティは各項目に対して定義する必要があります。その他のプロパティはオプションです。渡されたプロパティのみが更新され、その他のプロパティは変更されません。

このメソッドは、標準 MFA メソッドのプロパティ を持つ MFA メソッドを受けつけます。

戻り値

(object) mfaids プロパティの下にある更新された MFA メソッドの ID を含むオブジェクトを返します。

例

Updating method properties

メソッドプロパティの更新

時間ベースのワンタイムパスワード（TOTP）を利用した”Zabbix TOTP” MFA メソッドの TOTP コード生成用ハッシュ関数と検証コードの長さを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CMfa::update() in *ui/include/classes/api/services/CMfa.php*.

Module

このクラスは、Web コンソールのモジュールを操作するためのものです。

オブジェクトリファレンス:

- Module

利用可能なメソッド:

- module.create - 新しいモジュールをインストールする
- module.delete - モジュールをアンインストールする
- module.get - モジュールを取得する
- module.update - モジュールを更新する

Module オブジェクト

次のオブジェクトは、moduleAPI に直接関連しています。

モジュール

モジュールオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
moduleid	ID	データベースに保存されているモジュールの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
id	string	モジュールのmanifest.json ファイルで開発者により定義される、一意のモジュール ID。 組み込みモジュールで使用可能な値: ダッシュボードウィジェット のプロパティ「type」の説明を参照してください。 プロパティの動作: - 作成操作で 必須
relative_path	string	Zabbix Web インターフェースのディレクトリからの相対パスで表した、モジュールのディレクトリへのパス。 使用可能な値: `widgets/*` - 組み込みウィジェットモジュール用; `modules/*` - サードパーティモジュール用。 プロパティの動作: - 作成操作で 必須
status	integer	モジュールが有効か無効かを示します。 使用可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
config	object	モジュール設定。

module.create

Description

`object module.create(object/array modules)`

このメソッドは、新しいフロントエンドモジュールをインストールします。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーロール をご覧ください。

> **Attention:**
> モジュールファイルは、モジュールの `relative_path` プロパティに一致する正しいサブディレクトリに手動で展開する必要があります。

パラメータ

`(object/array)` 作成するモジュール。

このメソッドは、標準モジュールプロパティを持つモジュールを受け入れます。

戻り値

`(object)` は、`moduleids` プロパティにインストールされているモジュールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたモジュールの順序と一致します。

例

モジュールのインストール

モジュールをステータス” 有効” でインストールします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

参照

- Module
- Frontend modules

ソース

CModule::create() in *ui/include/classes/api/services/CModule.php*.

module.delete

説明

`object module.delete(array moduleids)`

このメソッドは、モジュールをアンインストールすることができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

Attention:
モジュールファイルは手動で削除する必要があります。

パラメーター

(array) アンインストールするモジュールの ID。

戻り値

(object) moduleids プロパティの下にあるアンインストールされたモジュールの ID を含むオブジェクトを返します。

例

複数のモジュールのアンインストール

モジュール"27" と"28" をアンインストールします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
```

```
        "28"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

ソース

CModule::delete() in *ui/include/classes/api/services/CModule.php*.

module.get

説明

`integer/array module.get(object parameters)`

このメソッドは、指定したパラメーターに応じたモジュールを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
moduleids	ID/array	指定された ID を持つモジュールのみを返します。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `moduleid`, `relative_path`。
countOutput	boolean	これらのパラメータについては、リファレンス解説を参照してください。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ID でモジュールを取得する

モジュール”1”、”2”、”25” に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "moduleid": "1",
            "id": "actionlog",
            "relative_path": "widgets/actionlog",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "2",
            "id": "clock",
            "relative_path": "widgets/clock",
            "status": "1",
            "config": []
        },
        {
            "moduleid": "25",
            "id": "example",
            "relative_path": "modules/example_module",
            "status": "1",
            "config": []
        }
    ],
    "id": 1
}
```

参照

- Module
- Dashboard widget
- Frontend modules

ソース

CModule::get() in *ui/include/classes/api/services/CModule.php*.

module.update

説明

`object proxy.update(object/array proxies)`

このメソッドは、既存のプロキシを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 更新されるトークンプロパティ

tokenid プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) roleids プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

モジュールの無効化

モジュール"25" を無効化します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

参照

- Module
- Frontend modules

ソース

CModule::update() in *ui/include/classes/api/services/CModule.php*.

Proxy

このクラスは、プロキシを操作するためのものです。

オブジェクトリファレンス:

- Proxy

利用可能なメソッド:

- proxy.create - 新しいプロキシを作成する
- proxy.delete - プロキシを削除する
- proxy.get - プロキシを取得する
- proxy.update - プロキシを更新する

Proxy オブジェクト

次のオブジェクトは、proxyAPI に直接関連しています。

プロキシ

プロキシオブジェクトには、以下のプロパティがあります。

Property	Type	Description
proxyid	ID	プロキシの ID。 Property behavior: - 読み取り専用 - 更新操作で 必須
name	string	プロキシの名前。 Property behavior: - 作成操作で 必須
proxy_groupid	ID	プロキシグループの ID。 0 - プロキシはいずれのプロキシグループにも割り当てられていません。
local_address	string	アクティブエージェント用のアドレス。接続先の IP アドレスまたは DNS 名。 Property behavior: - proxy_groupid が 0 でない場合は 必須
local_port	string	接続先のローカルプロキシのポート番号。 ユーザーマクロをサポートします。 デフォルト: 10051。 Property behavior: - proxy_groupid が 0 でない場合に サポート
operating_mode	integer	プロキシのタイプ。 使用可能な値: 0 - アクティブプロキシ; 1 - パッシブプロキシ。 Property behavior: - 作成操作で 必須
description	text	プロキシの説明。
lastaccess	timestamp	プロキシが最後にサーバーへ接続した時刻。 Property behavior: - 読み取り専用
address	string	接続先の IP アドレスまたは DNS 名。 ユーザーマクロをサポートします。 Property behavior: - Zabbix プロキシの動作モードがパッシブの場合は 必須
port	string	接続先のポート番号。 ユーザーマクロをサポートします。 デフォルト: 10051。 Property behavior: - Zabbix プロキシの動作モードがパッシブの場合に サポート
allowed_addresses	string	アクティブ Zabbix プロキシの IP アドレスまたは DNS 名のカンマ区切りリスト。

Property	Type	Description
tls_connect	integer	ホストへの接続。 使用可能な値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。
tls_accept	integer	ホストからの接続。 使用可能なビットマップ値: 1 - *(デフォルト)* 暗号化なし; 2 - PSK; 4 - 証明書。 これはビットマスクフィールドです。使用可能なビットマップ値の任意の合計を指定できます（たとえば、PSK と証明書の場合は 6）。
tls_issuer	string	証明書の発行者。
tls_subject	string	証明書のサブジェクト。
tls_psk_identity	string	PSK ID。1 つの PSK に対して 1 つだけ対応している必要があります (autoregistration、hosts、proxies 全体で)。 PSK ID には機密情報を含めないでください。受信側に使用する PSK を通知するため、ネットワーク上を暗号化されずに送信されます。 Property behavior: - 書き込み専用 - `tls_connect` が”PSK” に設定されている場合、または `tls_accept` に”PSK” ビットが含まれている場合は 必須
tls_psk	string	事前共有鍵（PSK)。32 桁以上の 16 進数である必要があります。 Property behavior: - 書き込み専用 - `tls_connect` が”PSK” に設定されている場合、または `tls_accept` に”PSK” ビットが含まれている場合は 必須
custom_timeouts	integer	プロキシレベルでグローバルなアイテムタイムアウトを上書きするかどうか。 使用可能な値: 0 - *(デフォルト)* グローバル設定を使用; 1 - タイムアウトを上書き。
timeout_zabbix_agent	string	処理に `timeout_*` 秒を超えて費やしません。 秒、またはサフィックス付きの時間単位（例: 30s、1m）を受け付けます。 ユーザーマクロも使用できます。 使用可能な値の範囲: 1-600s。 デフォルト: ””。 Property behavior: - `custom_timeouts` が 1 に設定されている場合は 必須。
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

Property	Type	Description
version	integer	プロキシのバージョン。 3 部構成の Zabbix バージョン番号で、各部には 2 桁の 10 進数が使用されます。たとえば、バージョン 6.4.1 は 60401、バージョン 7.0.2 は 70002 です。 0 - 不明なプロキシバージョン。 Property behavior: - 読み取り専用
compatibility	integer	Zabbix サーバーのバージョンと比較したプロキシのバージョン。 使用可能な値: 0 - 未定義; 1 - 現行バージョン（プロキシとサーバーのメジャーバージョンが同じ）; 2 - 古いバージョン（プロキシのバージョンはサーバーより古いが、部分的にサポートされる）; 3 - 非サポートバージョン（プロキシのバージョンがサーバーの 1 つ前の LTS リリースより古い、またはサーバーのメジャーバージョンがプロキシのメジャーバージョンより古い）。 Property behavior: - 読み取り専用
state	integer	プロキシの状態。 使用可能な値: 0 - 不明; 1 - オフライン; 2 - オンライン。 Property behavior: - 読み取り専用

proxy.create

説明

`object proxy.create(object/array proxies)`

このメソッドは、新しいプロキシを作成することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 作成するプロキシ。

このメソッドは、標準のプロキシプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
hosts	array	プロキシによって監視されるホスト。 ホストがすでに別のプロキシによって監視されている場合、そのホストは現在のプロキシに再割り当てされます。 ホストでは、`hostid` プロパティのみを定義する必要があります。

戻り値

(`object`) `proxyids` プロパティの下に作成されたプロキシの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシの順序と一致します。

例

アクティブプロキシの作成

アクションプロキシ”Active proxy” を作成し、それによって監視されるホストを割り当てます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

パッシブプロキシの作成

パッシブプロキシ”Passive proxy” を作成し、それによって監視される 2 つのホストを割り当てます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Passive proxy",
        "operating_mode": "1",
        "address": "127.0.0.1",
        "port": "10051",
        "hosts": [
            {
                "hostid": "10192"
            },
            {
                "hostid": "10139"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
```

```
        ]
    },
    "id": 1
}
```

プロキシを作成して、プロキシグループへ追加

アクティブプロキシ"Active proxy" を作成し、ID"1" のプロキシグループに追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト
- プロキシグループ

ソース

CProxy::create() in *ui/include/classes/api/services/CProxy.php*.

proxy.delete

説明

`object proxy.delete(array proxies)`

このメソッドは、プロキシを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するプロキシの ID。

戻り値

(object) 削除されたプロキシの ID を proxyids プロパティに含むオブジェクトを返します。

例

複数のプロキシの削除

2 つのプロキシを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

ソース

CProxy::delete() in *ui/include/classes/api/services/CProxy.php*.

proxy.get

説明

`integer/array proxy.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってプロキシを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
proxyids	ID/array	指定された ID を持つプロキシのみを返します。
proxy_groupids	ID/array	指定されたプロキシグループに属するプロキシのみを返します。
selectAssignedHosts	query	プロキシに割り当てられたホストを含む`assignedHosts` プロパティを返します。 `count` をサポートします。
selectHosts	query	プロキシによって監視されているホストを含む`hosts` プロパティを返します。 `count` をサポートします。
selectProxyGroup	query	プロキシグループオブジェクトを含む`proxyGroup` プロパティを返します。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `proxyid`, `name`, `operating_mode`。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	

パラメータ	型	説明
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列。
- countOutput パラメータが使用されている場合は、取得したオブジェクト数。

例

すべてのプロキシを取得

設定されているすべてのプロキシとそのインターフェースを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "11",
            "name": "Active proxy",
            "proxy_groupid": "0",
            "local_address": "",
            "local_port": "10051",
            "operating_mode": "0",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "0",
            "timeout_zabbix_agent": "",
            "timeout_simple_check": "",
            "timeout_snmp_agent": "",
            "timeout_external_check": "",
            "timeout_db_monitor": "",
            "timeout_http_agent": "",
            "timeout_ssh_agent": "",
            "timeout_telnet_agent": "",
            "timeout_script": "",
            "last_access": "1693391880",
            "version": "70000",
```

```
            "compatibility": "1",
            "state": "1"
        },
        {
            "proxyid": "12",
            "name": "Passive proxy",
            "proxy_groupid": "1",
            "local_address": "127.0.0.1",
            "local_port": "10051",
            "operating_mode": "1",
            "description": "",
            "allowed_addresses": "",
            "address": "127.0.0.1",
            "port": "10051",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "custom_timeouts": "1",
            "timeout_zabbix_agent": "5s",
            "timeout_simple_check": "5s",
            "timeout_snmp_agent": "5s",
            "timeout_external_check": "5s",
            "timeout_db_monitor": "5s",
            "timeout_http_agent": "5s",
            "timeout_ssh_agent": "5s",
            "timeout_telnet_agent": "5s",
            "timeout_script": "5s",
            "lastaccess": "1693391875",
            "version": "60400",
            "compatibility": "2",
            "state": "2"
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- プロキシグループ

ソース

CProxy::get() in *ui/include/classes/api/services/CProxy.php*.

proxy.update

説明

`object proxy.update(object/array proxies)`

このメソッドは、既存のプロキシを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するプロキシのプロパティ。

各プロキシに対して `proxyid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

標準のプロキシプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
hosts	array	プロキシによって監視されるホスト。 ホストがすでに別のプロキシによって監視されている場合、そのホストは現在のプロキシに再割り当てされます。 ホストでは hostid プロパティのみを定義する必要があります。

戻り値

(object) proxyids プロパティの下に、更新されたプロキシの ID を含むオブジェクトを返します。

例

プロキシが監視するホストの変更

指定された 2 つのホストを監視するようにプロキシを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            },
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

プロキシステータスの変更

プロキシをアクティブプロキシに変更し、名前を”Active proxy” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

プロキシをプロキシグループへ追加

ID”5” のプロキシを更新し、ID””1 のプロキシループに追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト
- プロキシグループ

ソース

CProxy::update() in *ui/include/classes/api/services/CProxy.php*.

Settings

このクラスは、一般的な管理設定を操作するためのものです。

オブジェクトリファレンス:

- Settings

利用可能なメソッド:

- settings.get - 設定を取得する
- settings.update - 設定を更新する

Settings オブジェクト

次のオブジェクトは、`settings` API に直接関連しています。

設定

settings オブジェクトは、以下のプロパティを持ちます。

Property	Type	Description
default_lang	string	デフォルトのシステム言語。 デフォルト: en_US。
default_timezone	string	デフォルトのシステムタイムゾーン。 デフォルト: system - システムのデフォルト。 サポートされているタイムゾーンの完全な一覧については、PHP documentationを参照してください。
default_theme	string	デフォルトのテーマ。 指定可能な値: blue-theme - *(デフォルト)* Blue; dark-theme - Dark; hc-light - High-contrast light; hc-dark - High-contrast dark。
search_limit	integer	検索およびフィルター結果の上限。 デフォルト: 1000。
max_overview_table_size	integer	「データ概要」および「トリガー概要」ダッシュボードウィジェット内の列数と行数の最大値。 デフォルト: 50。
max_in_table	integer	テーブルセル内に表示する要素数の最大値。 デフォルト: 50。
server_check_interval	integer	Zabbix server が停止している場合に警告を表示します。 指定可能な値: 0 - 警告を表示しない; 10 - *(デフォルト)* 警告を表示する。
work_period	string	稼働時間。 デフォルト: 1-5,09:00-18:00。
show_technical_errors	integer	Super admin 以外のユーザー、およびデバッグモードが有効なユーザーグループに属していないユーザーに対して技術的エラー（PHP/SQL）を表示します。 指定可能な値: 0 - *(デフォルト)* 技術的エラーを表示しない; 1 - 技術的エラーを表示する。
history_period	string	「最新データ」、「Web」、および「データ概要」ダッシュボードウィジェットで履歴データを表示する最大期間。 秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 24h。
period_default	string	時間フィルターのデフォルト期間。 秒、または月と年をサポートするサフィックス付きの時間単位（30s, 1m, 2h, 1d, 1M, 1y）を受け付けます。 デフォルト: 1h。
max_period	string	時間フィルターの最大期間。 秒、または月と年をサポートするサフィックス付きの時間単位（30s, 1m, 2h, 1d, 1M, 1y）を受け付けます。 デフォルト: 2y。
severity_color_0	string	重大度「Not classified」の色（16 進カラーコード）。 デフォルト: 97AAB3。
severity_color_1	string	重大度「Information」の色（16 進カラーコード）。 デフォルト: 7499FF。

Property	Type	Description
severity_color_2	string	重大度「Warning」の色（16 進カラーコード）。 デフォルト: FFC859。
severity_color_3	string	重大度「Average」の色（16 進カラーコード）。 デフォルト: FFA059。
severity_color_4	string	重大度「High」の色（16 進カラーコード）。 デフォルト: E97659。
severity_color_5	string	重大度「Disaster」の色（16 進カラーコード）。 デフォルト: E45959。
severity_name_0	string	重大度「Not classified」の名前。 デフォルト: Not classified。
severity_name_1	string	重大度「Information」の名前。 デフォルト: Information。
severity_name_2	string	重大度「Warning」の名前。 デフォルト: Warning。
severity_name_3	string	重大度「Average」の名前。 デフォルト: Average。
severity_name_4	string	重大度「High」の名前。 デフォルト: High。
severity_name_5	string	重大度「Disaster」の名前。 デフォルト: Disaster。
custom_color	integer	カスタムイベントステータス色を使用します。 指定可能な値: 0 - *(デフォルト)* カスタムイベントステータス色を使用しない; 1 - カスタムイベントステータス色を使用する。
ok_period	string	OK トリガーの表示期間。 秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 5m。
blink_period	string	ステータス変更時のトリガー点滅期間。 秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 2m。
problem_unack_color	string	未確認の PROBLEM イベントの色（16 進カラーコード）。 デフォルト: CC0000。
problem_ack_color	string	確認済みの PROBLEM イベントの色（16 進カラーコード）。 デフォルト: CC0000。
ok_unack_color	string	未確認の RESOLVED イベントの色（16 進カラーコード）。 デフォルト: 009900。
ok_ack_color	string	確認済みの RESOLVED イベントの色（16 進カラーコード）。 デフォルト: 009900。
problem_unack_style	integer	未確認の PROBLEM イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。

Property	Type	Description
problem_ack_style	integer	確認済みの PROBLEM イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
ok_unack_style	integer	未確認の RESOLVED イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
ok_ack_style	integer	確認済みの RESOLVED イベントの点滅。 指定可能な値: 0 - 点滅を表示しない; 1 - *(デフォルト)* 点滅を表示する。
url	string	Web インターフェース URL。
discovery_groupid	ID	検出されたホストが自動的に配置されるホストグループの ID。
default_inventory_mode	integer	デフォルトのホストインベントリモード。 指定可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
alert_usrgrpid	ID	データベース停止アラームメッセージの送信先となるユーザーグループの ID。 ”0” に設定した場合、アラームメッセージは送信されません。
snmptrap_logging	integer	一致しない SNMP トラップをログに記録します。 指定可能な値: 0 - 一致しない SNMP トラップをログに記録しない; 1 - *(デフォルト)* 一致しない SNMP トラップをログに記録する。
login_attempts	integer	ログインフォームがブロックされるまでのログイン失敗回数。 デフォルト: 5。
login_block	string	ログイン失敗回数が login_attempts フィールドで定義された値を超えた場合に、ログインフォームがブロックされる時間間隔。 秒、またはサフィックス付きの時間単位を受け付けます。 デフォルト: 30s。
validate_uri_schemes	integer	URI スキームを検証します。 指定可能な値: 0 - 検証しない; 1 - *(デフォルト)* 検証する。
uri_valid_schemes	string	有効な URI スキーム。 デフォルト: http,https,ftp,file,mailto,tel,ssh。
x_frame_options	string	X-Frame-Options HTTP ヘッダー。 デフォルト: SAMEORIGIN。
iframe_sandboxing_enabled	integer	iframe サンドボックス化を使用します。 指定可能な値: 0 - 使用しない; 1 - *(デフォルト)* 使用する。
iframe_sandboxing_exceptions	string	iframe サンドボックス化の例外。

Property	Type	Description
connect_timeout	string	Zabbix server との接続タイムアウト。 指定可能な値の範囲: 1-30s。 デフォルト: 3s。 Property behavior: - *required*
socket_timeout	string	ネットワークのデフォルトタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 3s。 Property behavior: - *required*
media_type_test_timeout	string	メディアタイプテストのネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 65s。 Property behavior: - *required*
item_test_timeout	string	アイテムテストのネットワークタイムアウト。 指定可能な値の範囲: 1-600s。 デフォルト: 60s。 Property behavior: - *required*
script_timeout	string	スクリプト実行のネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 Property behavior: - *required*
report_test_timeout	string	定期レポートテストのネットワークタイムアウト。 指定可能な値の範囲: 1-300s。 デフォルト: 60s。 Property behavior: - *required*
auditlog_enabled	integer	監査ログを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
auditlog_mode	integer	サーバー（System user）によって実行されるローレベルディスカバリ、ネットワークディスカバリ、および自動登録アクティビティの監査ログを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

Property	Type	Description
ha_failover_delay	string	フェイルオーバー遅延（秒）。 デフォルト: 1m。 Property behavior: - *read-only*
geomaps_tile_provider	string	ジオマップのタイルプロバイダー。 指定可能な値: `OpenStreetMap.Mapnik` - *(デフォルト)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery。 `geomaps_tile_url`、`geomaps_max_zoom`、`geomaps_attribution` のカスタム値を指定するために空文字列もサポートします。
geomaps_tile_url	string	ジオマップのタイル URL。 Property behavior: - `geomaps_tile_provider` が空文字列に設定されている場合に *supported*
geomaps_max_zoom	integer	ジオマップの最大ズームレベル。 指定可能な値: 0-30。 Property behavior: - `geomaps_tile_provider` が空文字列に設定されている場合に *supported*
geomaps_attribution	string	ジオマップの帰属表示テキスト。 Property behavior: - `geomaps_tile_provider` が空文字列に設定されている場合に *supported*
vault_provider	integer	Vault プロバイダー。 指定可能な値: 0 - *(デフォルト)* HashiCorp Vault; 1 - CyberArk Vault。
proxy_secrets_provider	integer	シークレットマクロ値の解決方法: 指定可能な値: 0 - *(デフォルト)* server のみ; 1 - server とプロキシがそれぞれ独立して解決。
timeout_zabbix_agent	string	処理に `timeout_*` 秒を超えて費やしません。 秒、またはサフィックス付きの時間単位（例: 30s, 1m）を受け付けます。 ユーザーマクロも受け付けます。 指定可能な値の範囲: 1-600s。 デフォルト: 3s。 `timeout_browser` のデフォルト: 60s。 Property behavior: - *required*
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		

Property	Type	Description
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

settings.get

説明

`object settings.get(object parameters)`

このメソッドを使用すると、指定されたパラメータに従って設定オブジェクトを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは 1 つのパラメータのみをサポートします。

パラメータ	Type	説明
output	query	このパラメータについては、リファレンス解説で説明されています。

戻り値

(object) 設定オブジェクトを返します

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
```

```
        "severity_name_0": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "geomaps_max_zoom": "0",
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```


ソース

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

説明

`object settings.update(object settings)`

このメソッドでは、既存の共通設定を更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

更新する（object）設定プロパティ。

戻り値

（array）更新されたパラメータ名の配列を返します。

例

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

ソース

CSettings::update() in *ui/include/classes/api/services/CSettings.php*.

SLA

このクラスは、IT インフラストラクチャおよびビジネスサービスのパフォーマンスを評価するために使用される SLA（Service Level Agreement）オブジェクトを扱うように設計されています。

オブジェクト参照:

- SLA
- SLA スケジュール
- SLA 除外ダウンタイム
- SLA サービスタグ

利用可能なメソッド:

- sla.create - 新しい SLA を作成
- sla.delete - SLA を削除
- sla.get - SLA を取得
- sla.getsli - SLA の Service Level Indicator（SLI）データを取得
- sla.update - SLA を更新

SLA オブジェクト

次のオブジェクトは、`sla`(Service Level Agreement) API に直接関連しています。

SLA

SLA オブジェクトには、以下のプロパティがあります。

Property	Type	Description
slaid	ID	SLA の ID。 Property behavior: - *read-only* - 更新操作で *required*
name	string	SLA の名前。 Property behavior: - 作成操作で *required*
period	integer	SLA のレポート期間。 指定可能な値: 0 - 日次; 1 - 週次; 2 - 月次; 3 - 四半期ごと; 4 - 年次。 Property behavior: - 作成操作で *required*
slo	float	パーセントで表される、許容可能な最小のサービスレベル目標（SLO）。サービスレベル指標（SLI）がこれを下回ると、SLA は障害/未達成状態と見なされます。 指定可能な値: 0～100（小数点以下最大 4 桁）。 Property behavior: - 作成操作で *required*
effective_date	integer	SLA の発効日。 指定可能な値: UTC での日時タイムスタンプ。 Property behavior: - 作成操作で *required*
timezone	string	レポートのタイムゾーン。例: `Europe/London`、`UTC`。 サポートされているタイムゾーンの完全な一覧については、PHP documentationを参照してください。 Property behavior: - 作成操作で *required*
status	integer	SLA のステータス。 指定可能な値: 0 - *(default)* 無効な SLA; 1 - 有効な SLA。
description	string	SLA の説明。

SLA スケジュール

SLA スケジュールオブジェクトは、接続されたサービスが正常に稼働していることが予定されている期間を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
period_from	integer	繰り返し発生する週次期間の開始時刻（この時刻を含む）。 指定可能な値: 秒数（日曜日からの経過秒数）。 プロパティの動作: - 必須
period_to	integer	繰り返し発生する週次期間の終了時刻（この時刻を含まない）。 指定可能な値: 秒数（日曜日からの経過秒数）。 プロパティの動作: - 必須

SLA excluded downtime

excluded downtime オブジェクトは、接続されたサービスが SLI に影響を与えることなく、正常に機能していないスケジュールを定義します。例えば、計画的なメンテナンスなどです。以下のプロパティがあります。

プロパティ	Type	説明
name	string	除外するダウンタイムの名前。 プロパティの動作: - 必須
period_from	integer	除外するダウンタイムの開始時刻（この時刻を含む）。 指定可能な値: タイムスタンプ。 プロパティの動作: - 必須
period_to	integer	除外するダウンタイムの終了時刻（この時刻を含まない）。 指定可能な値: タイムスタンプ。 プロパティの動作: - 必須

SLA サービスタグ

SLA サービスタグオブジェクトは、SLA の計算に含めるサービスを関連付けます。以下のプロパティがあります。

プロパティ	Type	説明
tag	string	SLA サービスタグ名。 プロパティの動作: - 必須
operator	integer	SLA サービスタグの演算子。 指定可能な値: 0 - *(デフォルト)* 次と等しい; 2 - 含む。
value	string	SLA サービスタグの値。

sla.create

説明

`object sla.create(object/array SLAs)`

このメソッドは、新しい SLA オブジェクトを作成します。

Note:
このメソッドは、管理者およびスーパー管理者のユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(object/array) 作成する SLA オブジェクト。

標準 SLA プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
service_tags	array	SLA 用に作成するSLA サービスタグ。 パラメータの動作: - 必須
schedule	array	SLA 用に作成するSLA スケジュール。 空のパラメータを指定した場合は、24x7 スケジュールとして解釈されます。 デフォルト: 24x7 スケジュール。
excluded_downtimes	array	SLA 用に作成するSLA 除外ダウンタイム。

戻り値

(object) slaids プロパティの下に、作成された SLA の ID を含むオブジェクトを返します。返される ID の順序は、渡された SLA の順序と一致します。

例

SLA の作成

以下の条件で SLA エントリを作成するよう指示します。* SQL エンジン関連サービスの稼働率を追跡すること。* 土曜日の最後の 1 時間を除く、すべての平日のカスタムスケジュールであること。* 発効日が 2022 年の最終日であること。* 7 月 4 日の午前 0 時に開始する、1 時間 15 分の計画停止時間があること。* SLA の週次レポート計算を有効にすること。* 許容される最小 SLO が 99.9995% であること。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": " データベース稼働時間",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": " 主要なデータベースエンジンに優れた稼働時間を提供します。",
            "effective_date": 1672444800,
            "status": 1,
            "schedule": [
                {
                    "period_from": 0,
                    "period_to": 601200
                }
            ],
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
```

```
                }
            ],
            "excluded_downtimes": [
                {
                    "name": " ソフトウェアバージョンアップグレードの展開",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php 内の CSla::create()。

sla.delete

説明

`object sla.delete(array slaids)`

このメソッドは、SLA エントリを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する SLA の ID。

戻り値

(object) 削除された SLA の ID を slaids プロパティの下に含むオブジェクトを返します。

例

複数の SLA の削除

2 つの SLA エントリを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php 内の CSla::delete()。

sla.get

説明

`integer/array sla.get(object parameters)`

このメソッドを使用すると、指定されたパラメーターに従って SLA オブジェクトを取得できます。

Note:
このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくはユーザーの役割を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
slaids	ID/array	指定された ID を持つ SLA のみを返します。
serviceids	ID/array	特定のサービスに一致する SLA のみを返します。
selectSchedule	query	SLA スケジュールを含む `schedule` プロパティを返します。 `count` をサポートします。
selectExcludedDowntimes	query	SLA の除外ダウンタイムを含む `excluded_downtimes` プロパティを返します。 `count` をサポートします。
selectServiceTags	query	SLA のサービスタグを含む `service_tags` プロパティを返します。 `count` をサポートします。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクト数

例

すべての SLA の取得

すべての SLA とそのプロパティに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.get",
    "params": {
        "output": "extend",
        "selectSchedule": ["period_from", "period_to"],
        "selectExcludedDowntimes": ["name", "period_from", "period_to"],
        "selectServiceTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "1": {
            "slaid": "1",
            "name": "Database Uptime",
            "period": "1",
            "slo": "99.9995",
            "effective_date": "1672444800",
            "timezone": "America/Toronto",
            "status": "1",
            "description": "Provide excellent uptime for main SQL database engines.",
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mysql"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "postgresql"
                }
            ],
            "schedule": [
                {
                    "period_from": "0",
                    "period_to": "601200"
                }
            ],
            "excluded_downtimes": [
                {
                    "name": "Software version upgrade rollout",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
```

```
        }
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php 内の CSla:get()。

sla.getsli

説明

`object sla.getsli(object parameters)`

このメソッドを使用すると、サービスレベルアグリーメント (SLA) のサービスレベル指標 (SLI) データを計算できます。

Note:
このメソッドは、あらゆるタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(object) SLA ID、レポート期間、および必要に応じて SLI の計算対象となるサービスの ID を含むパラメータです。

パラメータ	Type	説明
slaid	ID	可用性情報を返す SLA の ID。 パラメータの動作: - 必須
period_from	timestamp	SLI をレポートする開始タイムスタンプ（この値を含む）。 指定可能な値: Unix タイムスタンプ。
period_to	timestamp	SLI をレポートする終了タイムスタンプ（この値を含む）。 指定可能な値: Unix タイムスタンプ。
periods	integer	レポートする期間の数。 指定可能な値: 1-100
serviceids	ID/array	SLI を返すサービスの ID。

期間の分割

次の表は、パラメータの組み合わせに基づいて返される期間スライスの配置を示しています。

Note:
返される期間は、最初に利用可能な期間（SLA の有効日付に基づく）より前にはならず、現在の期間を超えることもありません。

パラメータ			返される期間
period_from	**period_to**	**periods**	
-	-	-	現在の期間を含む直近 20 期間。
-	-	N	直近 N 期間。
-	指定あり	-	period_to より前の直近 20 期間。
-	指定あり	N	period_to より前の直近 N 期間。
指定あり	-	-	period_from から始まる最初の 20 期間。
指定あり	-	N	period_from から始まる最初の N 期間。
指定あり	指定あり	-	指定した範囲内で最大 100 期間。
指定あり	指定あり	N	指定した範囲内で N 期間。

戻り値

(object) 計算結果を返します。

プロパティ	Type	説明
periods	array	報告対象期間の一覧。 各報告対象期間は、以下で構成されるオブジェクトとして表されます。 - period_from - (timestamp) 報告対象期間の開始タイムスタンプ(含む)。 - period_to - (timestamp) 報告対象期間の終了タイムスタンプ(含まない)。 期間は period_from でソートされ、最も早い期間が先頭に表示されます。
serviceids	array	報告対象期間内のサービス ID の一覧。 一覧の並び順は定義されていません。sla.getsli メソッドに serviceids パラメータが渡された場合でも同様です。
sli	array	各報告対象期間およびサービスの SLI データ(二次元配列として)。 periods プロパティのインデックスが、sli プロパティの第 **1** 次元として使用されます。 serviceids プロパティのインデックスが、sli プロパティの第 **2** 次元として使用されます。

SLI データ

報告期間およびサービスごとに返される SLI データは、以下で構成されます。

プロパティ	Type	説明
uptime	integer	スケジュールされた稼働時間内において、除外されたダウンタイムを差し引いた後に、サービスが *OK* 状態であった時間。
downtime	integer	スケジュールされた稼働時間内において、除外されたダウンタイムを差し引いた後に、サービスが *not OK* 状態であった時間。
sli	float	稼働時間と停止時間に基づく SLI(総稼働時間に対する割合)。
error_budget	integer	SLI および SLO に基づくエラーバジェット(秒単位)。
excluded_downtimes	array	この報告期間における除外されたダウンタイムの配列。 各オブジェクトには、以下のパラメータが含まれます。 - name - (string) 除外されたダウンタイムの名前。 - period_from - (timestamp) 除外されたダウンタイムの開始タイムスタンプ(この時刻を含む)。 - period_to - (timestamp) 除外されたダウンタイムの終了タイムスタンプ(この時刻を含まない)。 除外されたダウンタイムは period_from でソートされ、最も早い期間が先に表示されます。

例

日次 SLA の SLI の計算

ID「1」の SLA にリンクされた、ID「1」および「4」のサービスの SLI データを取得します。
「1761861599」(2025 年 10 月 30 日 23:59:59 GMT+0200)までの単一期間のデータを取得します。
SLA のレポート期間は日次であるため、SLI データは「1761775200」(2025 年 10 月 30 日 00:00:00 GMT+0200)から「1761861600」(2025 年 10 月 31 日 00:00:00 GMT+0200)まで取得されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
```

```
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1761775200,
                "period_to": 1761861600
            }
        ],
        "serviceids": [
            1,
            4
        ],
        "sli": [
            [
                {
                    "uptime": 43843,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance OCT",
                            "period_from": 1761825600,
                            "period_to": 1761829200
                        }
                    ]
                },
                {
                    "uptime": 32225,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


月次 SLA の SLI の計算

ID が「5」の SLA に関連付けられている、ID「50」、「60」、「70」のサービスの SLI データを取得します。
「1635724800」（2021 年 11 月 01 日 00:00:00 UTC）から開始する 3 期間分のデータを取得します。
SLA のレポート期間は月次であるため、SLI データは次の 3 か月分について取得されます。

- 「1635724800」（2021 年 11 月 01 日 00:00:00 UTC）から「1638316800」（2021 年 12 月 01 日 00:00:00 UTC）まで
- 「1638316800」（2021 年 12 月 01 日 00:00:00 UTC）から「1640995200」（2022 年 01 月 01 日 00:00:00 UTC）まで
- 「1640995200」（2022 年 01 月 01 日 00:00:00 UTC）から「1643673600」（2022 年 02 月 01 日 00:00:00 UTC）まで

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "periods": [
            {
                "period_from": 1635724800,
                "period_to": 1638316800
            },
            {
                "period_from": 1638316800,
                "period_to": 1640995200
            },
            {
                "period_from": 1640995200,
                "period_to": 1643673600
            }
        ],
        "serviceids": [
            50,
            60,
            70
        ],
        "sli": [
            [
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
                            "period_from": 1637836212,
                            "period_to": 1638316800
                        }
                    ]
                },
                {
                    "uptime": 1186212,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": [
                        {
                            "name": "Maintenance Nov 25 - Dec 01",
```

```
                    "period_from": 1637836212,
                    "period_to": 1638316800
                }
            ]
        },
        {
            "uptime": 1186212,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Nov 25 - Dec 01",
                    "period_from": 1637836212,
                    "period_to": 1638316800
                }
            ]
        }
    ],
    [
        {
            "uptime": 1147548,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Dec 02 - Dec 10",
                    "period_from": 1638439200,
                    "period_to": 1639109652
                }
            ]
        },
        {
            "uptime": 1147548,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Dec 02 - Dec 10",
                    "period_from": 1638439200,
                    "period_to": 1639109652
                }
            ]
        },
        {
            "uptime": 1147548,
            "downtime": 0,
            "sli": 100,
            "error_budget": 0,
            "excluded_downtimes": [
                {
                    "name": "Maintenance Dec 02 - Dec 10",
                    "period_from": 1638439200,
                    "period_to": 1639109652
                }
            ]
        }
    ],
    [
        {
```

```
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


ソース

CSla::getSli() in *ui/include/classes/api/services/CSla.php*

sla.update

Description

`object sla.update(object/array slaids)`

このメソッドで、既存の SLA エントリを更新できます。

> **Note:**
> このメソッドは *Admin* および *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

パラメータ

(object/array) 更新する SLA プロパティ。

各 SLA について slaid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

標準 SLA プロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
service_tags	array	現在の SLA サービスタグを置き換えるためのSLA サービスタグ。
schedule	array	現在のものを置き換えるためのSLA スケジュール。 パラメータを空で指定した場合は、24x7 スケジュールとして解釈されます。
excluded_downtimes	array	現在のものを置き換えるためのSLA 除外ダウンタイム。

戻り値

(object) slaids プロパティの下に、更新された SLA の ID を含むオブジェクトを返します。

例

サービスタグの更新

NoSQL 関連サービスの SLA ID ”5” を月次で計算するように設定し、スケジュールや除外するダウンタイムは変更せず、SLO を 95% に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": [
        {
            "slaid": "5",
            "name": "NoSQL Database engines",
            "slo": "95",
            "period": 2,
            "service_tags": [
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "redis"
                },
                {
                    "tag": "database",
                    "operator": "0",
                    "value": "mongodb"
                }
            ]
        }
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA のスケジュール変更

ID ”5” の SLA を 24 時間 365 日のスケジュールに切り替えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
```

```
    },
    "id": 1
}
```

SLA の除外ダウンタイムの変更

2022 年 4 月 6 日に 4 時間の RAM アップグレードのためのダウンタイムを計画的に追加します。ただし、ID ”5” の SLA については、7 月 4 日に既に計画されているソフトウェアアップグレードは維持します（新たに定義する必要があります）

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "excluded_downtimes": [
            {
                "name": "Software version upgrade rollout",
                "period_from": "1648760400",
                "period_to": "1648764900"
            },
            {
                "name": "RAM upgrade",
                "period_from": "1649192400",
                "period_to": "1649206800"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CSla.php 内の CSla::update()。

Token

このクラスは、トークンを操作するためのものです。

オブジェクトリファレンス:

- Token

利用可能なメソッド:

- token.create - 新しいトークンを作成する
- token.delete - トークンを削除する
- token.get - トークンを取得する
- token.update - トークンを更新する
- token.generate - トークンを生成する

Token オブジェクト

次のオブジェクトは、tokenAPI に直接関連しています。

トークン

トークンオブジェクトには、以下のプロパティがあります。

Property	Type	Description
tokenid	ID	トークンの ID。 Property behavior: - *read-only* - 更新操作で *required*
name	string	トークンの名前。 Property behavior: - 作成操作で *required*
description	text	トークンの説明。
userid	ID	トークンが割り当てられているユーザーの ID。 デフォルト: *current user*。 Property behavior: - *constant*
lastaccess	timestamp	トークンが最後に認証された日時。 トークンが一度も認証されていない場合は”0”。 Property behavior: - *read-only*
status	integer	トークンのステータス。 指定可能な値: 0 - （デフォルト）有効なトークン; 1 - 無効なトークン。
expires_at	timestamp	トークンの有効期限の日時。 有効期限のないトークンの場合は”0”。
created_at	timestamp	トークンの作成日時。 Property behavior: - *read-only*
creator_userid	ID	トークンを作成したユーザーの ID。 Property behavior: - *read-only*

token.create

説明

`object token.create(object/array tokens)`

このメソッドは、新しいトークンを作成することができます。

Note:
他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理の権限が必要です。

Attention:
このメソッドで作成されたトークンも、使用可能になる前に生成する必要があります。

パラメータ

`(object/array)` トークンの作成

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) tokenids プロパティの下で作成されたトークンの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトークンの順序と一致します。

例

トークンの作成

有効期限が切れることのない有効なトークンを作成し、ID2 のユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

2021 年 1 月 21 日に有効期限が切れるトークンを作成します。このトークンは現在のユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

ソース

CToken::create() in *ui/include/classes/api/services/CToken.php*.

token.delete

説明

`object token.delete(array tokenids)`

このメソッドは、トークンを削除することができます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理の権限が必要です。

パラメーター

(array) 削除するトークンの ID。

戻り値

(object) tokenids プロパティの下にある削除されたトークンの ID を含むオブジェクトを返します。

例

複数のトークンの削除

2 つのトークンを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

ソース

CToken::delete() in *ui/include/classes/api/services/CToken.php*.

token.generate

説明

`object token.generate(array tokenids)`

このメソッドでは、トークンを生成できます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーロールに *Manage API tokens* permission が必要です。

> **Attention:**
> トークンは、このメソッドで生成する前に作成 されている必要があります。

パラメータ

(array) 生成するトークンの ID

戻り値

(array) 生成されたトークンの ID を tokenid プロパティに、生成された認可文字列を token プロパティに含むオブジェクトの配列を返します。

プロパティ	Type	説明
tokenid	ID	トークンの ID。
token	string	このトークン用に生成された認可文字列。

例

複数のトークンを生成する

2 つのトークンを生成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

ソース

CToken::generate() in *ui/include/classes/api/services/CToken.php*.

token.get

説明

`integer/array token.get(object parameters)`

このメソッドでは、指定されたパラメーターに従ってトークンを取得できます。

> **Note:**
> *Super admin* ユーザータイプのみが他のユーザーのトークンを表示できます。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
tokenids	ID/array	指定した ID のトークンのみを返します。
userids	ID/array	指定したユーザー用に作成されたトークンのみを返します。

パラメータ	Type	説明
token	string	指定した認証トークン用に作成されたトークンのみを返します。
valid_at	timestamp	指定した日時の時点で有効（期限切れでない）なトークンのみを返します。
expired_at	timestamp	指定した日時の時点で期限切れ（無効）であるトークンのみを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントを参照してください。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トークンの取得

ID"2" のトークンのデータをすべて取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "name": "The Token",
            "description": "",
            "userid": "1",
            "lastaccess": "0",
            "status": "0",
            "expires_at": "1609406220",
            "created_at": "1611239454",
            "creator_userid": "1"
        }
    ],
    "id": 1
}
```

ソース

CToken::get() in *ui/include/classes/api/services/CToken.php*.

token.update

説明

`object token.update(object/array tokens)`

このメソッドは、既存のトークンを更新することができます。

> **Note:**
> 他のユーザーのトークンを管理するには、ユーザーの役割に *API* トークンの管理の権限が必要です。

パラメータ

(object/array) 更新されるトークンプロパティ

`tokenid` プロパティはトークンごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。

このメソッドは標準トークンプロパティを持つトークンを受け入れます。

戻り値

(object) `tokenids` プロパティの下で更新されたトークンの ID を含むオブジェクトを返します。

例

トークンの有効期限変更

トークンから有効期限を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CToken::update() in *ui/include/classes/api/services/CToken.php*.

User

このクラスは、ユーザーを操作するためのものです。

オブジェクトリファレンス:

- User
- Media

利用可能なメソッド:

- user.checkauthentication - ユーザーセッションの確認と延長をする
- user.create - 新しいユーザーを作成する
- user.delete - ユーザーを削除する
- user.get - ユーザーを取得する
- user.login - API にログインする
- user.logout - API からログアウトする
- user.provision - LDAP ユーザーのプロビジョニングをする
- user.resettotp - ユーザーの TOTP シークレットをリセットする
- user.unblock - ユーザーブロックを解除する
- user.update - ユーザーを更新する

User オブジェクト

次のオブジェクトは、userAPI に直接関連しています。

ユーザー

ユーザーオブジェクトには、以下のプロパティがあります。

Property	Type	Description
userid	ID	ユーザーの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
username	string	ユーザー名。 プロパティの動作: - 作成操作では 必須 - ユーザーがユーザーディレクトリ にリンクされており (userdirectoryid が"0" ではない有効な値に設定されている)、かつユーザーディレクトリのプロビジョニングステータスが有効であり (User directory object の provision_status が"1" に設定されている)、さらにすべての LDAP または SAML プロビジョニングの認証ステータスが有効である場合 (Authentication object の ldap_jit_status が"Enabled for configured LDAP IdPs" に設定されている、またはAuthentication object の saml_jit_status が"Enabled for configured SAML IdPs" に設定されている)、プロビジョニングされたユーザー では 読み取り専用
passwd	string	ユーザーのパスワード。 ユーザーがユーザーディレクトリ にリンクされている場合、このパラメータの値は空文字列にできます。 プロパティの動作: - 書き込み専用
roleid	ID	ユーザーのロールの ID。 ロールを持たないユーザーは、その LDAP/SAML 情報が Zabbix で設定されたユーザーグループマッピングと一致する場合に限り、LDAP またはSAML 認証を使用してのみ Zabbix にログインできることに注意してください。
attempt_clock	timestamp	最後にログインに失敗した試行の時刻。 プロパティの動作: - 読み取り専用
attempt_failed	integer	直近のログイン失敗回数。 プロパティの動作: - 読み取り専用

Property	Type	Description
attempt_ip	string	最後にログインに失敗した試行元の IP アドレス。 プロパティの動作: - 読み取り専用
autologin	integer	自動ログインを有効にするかどうか。 指定可能な値: 0 - *(デフォルト)* 自動ログイン無効; 1 - 自動ログイン有効。
autologout	string	ユーザーセッションの有効期間。秒、または接尾辞付きの時間単位を受け付けます。0s に設定すると、セッションは期限切れになりません。 デフォルト: 15m。
lang	string	ユーザーの言語の言語コード。例: en_US。 デフォルト: default - システムデフォルト。
name	string	ユーザーの名。
provisioned	integer	ユーザーがプロビジョニングされているかどうか。 指定可能な値: 0 - プロビジョニングされていない; 1 - プロビジョニング済み。 プロパティの動作: - 読み取り専用
refresh	string	自動更新間隔。秒、または接尾辞付きの時間単位（例: 30s、90s、1m、1h）を受け付けます。 デフォルト: 30s。
rows_per_page	integer	1 ページあたりに表示するオブジェクト行数。 デフォルト: 50。
surname	string	ユーザーの姓。
theme	string	ユーザーのテーマ。 指定可能な値: default - *(デフォルト)* システムデフォルト; blue-theme - Blue; dark-theme - Dark; hc-light - ハイコントラストライト; hc-dark - ハイコントラストダーク。
ts_provisioned	timestamp	最新のプロビジョニング操作が実行された時刻。 プロパティの動作: - 読み取り専用 - *Super admin* タイプのユーザーで サポート
url	string	ログイン後にユーザーをリダイレクトするページの URL。
userdirectoryid	ID	ユーザーがリンクされているユーザーディレクトリ の ID。 プロビジョニング（作成または更新）およびユーザーディレクトリにリンクされたユーザーのログインに使用されます。 ログイン操作では、このプロパティの値は、ユーザーが所属するユーザーグループ の userdirectoryid プロパティよりも優先されます。 デフォルト: 0。 プロパティの動作: - 読み取り専用 - *Super admin* タイプのユーザーで サポート

Property	Type	Description
timezone	string	ユーザーのタイムゾーン。例: Europe/London、UTC。 デフォルト: default - システムデフォルト。 サポートされているタイムゾーンの完全な一覧については、PHP documentation を参照してください。

メディア

メディアオブジェクトには以下のプロパティがあります。

Property	Type	Description
mediaid	ID	ユーザーのメディアの ID。 プロパティの動作: - 読み取り専用
mediatypeid	ID	ユーザーのメディアで使用されるメディアタイプの ID。 プロパティの動作: - 必須
sendto	string/array	受信者のアドレス、ユーザー名、またはその他の識別子。 メディアタイプ の type が”Email” に設定されている場合、値は配列として表されます。メディアタイプ のその他のタイプでは、値は文字列として表されます。 プロパティの動作: - 必須
active	integer	メディアが有効かどうか。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
severity	integer	通知を送信するトリガーの深刻度。 指定可能なビットマップ値: 1 - 未分類; 2 - 情報; 4 - 警告; 8 - 平均; 16 - 高; 32 - 障害。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、平均、高、障害の場合は 48）。 デフォルト: 63。
period	string	通知を送信できる時間。time period またはセミコロンで区切られたユーザーマクロで指定します。 デフォルト: 1-7,00:00-24:00。
provisioned	integer	ユーザーがプロビジョニングされているかどうか。 指定可能な値: 0 - プロビジョニングされていない; 1 - プロビジョニング済み。

Property	Type	Description
userdirectory_mediaid	ID	プロビジョニングされたメディアのユーザーディレクトリメディアマッピング ID。 プロパティの動作: - 読み取り専用 - *Super admin* タイプのユーザーで サポート

user.checkAuthentication

説明

`object user.checkAuthentication`

このメソッドは、ユーザーセッションを確認し、延長します。

Attention:
`sessionid` パラメータを使用して `user.checkAuthentication` メソッドを呼び出すと、デフォルトでユーザーセッションが延長されます。

パラメータ

このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
extend	boolean	ユーザーセッションを延長するかどうか。 デフォルト値: ”true”。 値を”false” に設定すると、延長せずにユーザーセッションを確認できます。 パラメータの動作: - `sessionid` が設定されている場合は *supported*
sessionid	string	ユーザーの認証トークン。 パラメータの動作: - `token` が設定されていない場合は *required*
token	string	ユーザーのAPI トークン。 パラメータの動作: - `sessionid` が設定されていない場合は *required*

戻り値

(`object`) ユーザーに関する情報を含むオブジェクトを返します。

標準のユーザープロパティに加えて、以下の情報が返されます。

プロパティ	Type	説明
auth_type	integer	ユーザーのデフォルト認証。 指定可能な値の一覧については、Authentication object の `authentication_type` プロパティを参照してください。
debug_mode	integer	ユーザーに対してデバッグモードが有効か無効か。 指定可能な値の一覧については、User group object の `debug_mode` プロパティを参照してください。
deprovisioned	boolean	ユーザーがプロビジョニング解除済みユーザーグループに属しているかどうか。

プロパティ	Type	説明
gui_access	string	ユーザーの Web インターフェースへの認証方法。 指定可能な値の一覧については、User group object の gui_access プロパティを参照してください。
secret	string	32 文字のランダムな文字列。ユーザーログイン時に生成されます。 API トークンを使用してユーザーセッションを確認する場合、secret プロパティは返されません。
sessionid	string	認証トークン。以降の API リクエストで使用する必要があります。 API トークンを使用してユーザーセッションを確認する場合、sessionid プロパティは返されません。
type	integer	ユーザータイプ。 指定可能な値の一覧については、Role object の type プロパティを参照してください。
userip	string	ユーザーの IP アドレス。

例

認証トークンを使用し認証を確認

ユーザー認証トークンを使用してユーザーセッションを確認し、延長します。また、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "deprovisioned": false,
```

```
        "auth_type": 0,
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2",
        "secret": "0e329b933e46984e49a5c1051ecd0751"
    },
    "id": 1
}
```

API トークンを使用した認証の確認

ユーザー API トークンを使用してユーザーセッションを確認し、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919338",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "ts_provisioned": "0",
        "debug_mode": 0,
        "deprovisioned": false,
        "gui_access": "1",
        "mfaid": 0,
        "auth_type": 0,
        "type": 3,
        "userip": "127.0.0.1"
    },
    "id": 1
}
```

ソース

ui/include/classes/api/services/CUser.php の CUser::checkAuthentication()。

user.create

説明

`object user.create(object/array users)`

このメソッドでは、新しいユーザーを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。
> メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

> **Note:**
> ユーザーパスワードの強度は、Authentication API で定義されたパスワードポリシールールに従って検証されます。詳細はAuthentication API を参照してください。

パラメータ

(object/array) 作成するユーザー。

このメソッドは、標準のユーザープロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
usrgrps	array	ユーザーの追加先となるユーザーグループ。 ユーザーグループでは、usrgrpid プロパティのみを定義する必要があります。
medias	array	作成するユーザーメディア。

戻り値

(object) userids プロパティの下で作成されたユーザーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーの順序と一致します。

例

ユーザーの作成

新しいユーザーを作成し、そのユーザーをユーザーグループに追加して、そのユーザー用の新しいメディアを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe123",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

関連項目

- 認証
- メディア
- ユーザーグループ
- ロール

ソース

CUser::create() in *ui/include/classes/api/services/CUser.php*.

user.delete

説明

`object user.delete(array users)`

このメソッドは、ユーザーを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するユーザーの ID。

戻り値

(object) userids プロパティの下にある削除されたユーザーの ID を含むオブジェクトを返します。

例

複数のユーザーの削除

2 つのユーザーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
```

```
    "id": 1
}
```

ソース

CUser::delete() in *ui/include/classes/api/services/CUser.php*.

user.get

説明

`integer/array user.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってユーザーを取得できます。

Note:
このメソッドは、任意のタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Note:
ユーザーメディア、権限、またはロール情報を要求する場合、*Admin* および *User* タイプのユーザーは、自分自身のユーザーに関するデータのみ取得できます。例については、Retrieving users as *Admin* を参照してください。

パラメータ	型	説明
mediaids	ID/array	指定したメディアを使用しているユーザーのみを返します。
mediatypeids	ID/array	指定したメディアタイプを使用しているユーザーのみを返します。
userids	ID/array	指定した ID を持つユーザーのみを返します。
usrgrpids	ID/array	指定したユーザーグループに属するユーザーのみを返します。
getAccess	flag	ユーザー権限に関する追加情報を加えます。 各ユーザーに対して以下のプロパティを追加します。 gui_access - (integer) ユーザーの Web インターフェース認証方式。ユーザーグループオブジェクト の gui_access プロパティを参照して、使用可能な値の一覧を確認してください。 debug_mode - (integer) ユーザーに対してデバッグが有効かどうかを示します。使用可能な値: 0 - デバッグ無効、1 - デバッグ有効。 users_status - (integer) ユーザーが無効化されているかどうかを示します。使用可能な値: 0 - ユーザー有効、1 - ユーザー無効。
selectMedias	query	ユーザーが使用しているメディアを`medias` プロパティで返します。
selectMediatypes	query	ユーザーが使用しているメディアタイプを`mediatypes` プロパティで返します。 ユーザータイプに基づく制限については、`mediatype.get` を参照してください。
selectUsrgrps	query	ユーザーが属しているユーザーグループを`usrgrps` プロパティで返します。 ユーザータイプに基づく制限については、`usergroup.get` を参照してください。
selectRole	query	ユーザーロールを`role` プロパティで返します。

パラメータ	型	説明
filter	object	指定したフィルターに完全一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしていません。
output	query	*Admin* および *User* タイプのユーザーが、自分のユーザーグループ内のユーザーに関するデータを要求する場合に使用可能なユーザーオブジェクト のプロパティ: `userid`, `name`, `surname`, `username`。 返されるユーザーオブジェクト のプロパティ。 *Admin* および *User* タイプのユーザーは、以下のプロパティのみ取得できます。 - 自分自身のユーザーについて: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`。 - 自分のユーザーグループ内のユーザーについて: `userid`, `name`, `surname`, `username`。 デフォルト: `extend`。
search	object	指定したパターンに一致する結果を返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索を実行します。 `string` および `text` データ型 のプロパティのみサポートしています。
sortfield	string/array	*Admin* および *User* タイプのユーザーが、自分のユーザーグループ内のユーザーに関するデータを要求する場合に使用可能なユーザーオブジェクト のプロパティ: `name`, `surname`, `username`。 指定したプロパティで結果をソートします。 使用可能な値: `userid`, `username`。
countOutput	boolean	これらのパラメータについては、reference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ユーザーの取得

設定されているすべてのユーザーを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "refresh": "0s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "3",
            "username": "user",
            "name": "Zabbix",
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
```

```
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
    "id": 1
}
```

管理者としてユーザーを取得する

管理者タイプのユーザーとして、自分のユーザーに関する詳細データと、ユーザーグループ内のユーザーの限定データを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "userid": "3",
            "username": "database-admin",
            "name": "John",
            "surname": "Doe",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "default",
```

```
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "2",
            "provisioned": "0",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```


ユーザーデータの取得

ID「12」のユーザーのデータを取得します。

リクエスト:


```
{
    "jsonrpc": "2.0",
```

```
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "type": "1",
              "readonly": "0"
          }
      }
  ],
  "id": 1
}
```

関連項目

- メディア
- メディアタイプ
- ユーザーグループ
- ロール

ソース

CUser::get() in *ui/include/classes/api/services/CUser.php*.

user.login

説明

`string/object user.login(object parameters)`

このメソッドを使用すると、API にログインして認証トークンを生成できます。

Warning:
このメソッドを使用する場合は、多数の未終了セッションレコードが生成されるのを防ぐために、user.logout も実行する必要があります。

Attention:
このメソッドは、多要素認証が有効なユーザーグループ に属していない、未認証のユーザーのみが利用できます。

パラメータ

(object) ユーザー名とパスワードを含むパラメータ。

このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
password	string	ユーザーパスワード。 パラメータの動作: - 必須

パラメータ	Type	説明
username	string	ユーザー名。 パラメータの動作: - 必須
userData	flag	認証されたユーザーの情報を返します。

戻り値

(string/object) userData パラメータが使用されている場合、認証されたユーザーに関する情報を含むオブジェクトを返します。

標準のユーザープロパティに加えて、以下の情報が返されます。

プロパティ	Type	説明
auth_type	integer	ユーザーのデフォルト認証。 使用可能な値の一覧については、Authentication object の authentication_type プロパティを参照してください。
debug_mode	integer	ユーザーに対してデバッグモードが有効か無効か。 使用可能な値の一覧については、User group object の debug_mode プロパティを参照してください。
deprovisioned	boolean	ユーザーがプロビジョニング解除済みユーザーグループに属しているかどうか。
gui_access	string	ユーザーの Web インターフェースへの認証方式。 使用可能な値の一覧については、User group object の gui_access プロパティを参照してください。
mfaid	integer	ログイン時にユーザーに対して使用するMFA method の ID。 MFA がグローバルに無効化されている場合、またはユーザーが属するすべてのユーザーグループで無効化されている場合は"0" を返します。
secret	string	32 文字のランダムな文字列。ユーザーのログイン時に生成されます。
sessionid	string	認証トークン。以降の API リクエストで使用する必要があります。
type	integer	ユーザータイプ。 使用可能な値の一覧については、Role object の type プロパティを参照してください。
userip	string	ユーザーの IP アドレス。

Note:
ユーザーが 1 回以上の認証失敗の後に正常に認証された場合、このメソッドは attempt_clock、attempt_failed、attempt_ip プロパティの現在の値を返し、その後それらをリセットします。

userData パラメータが使用されていない場合、このメソッドは認証に必要な認証トークンを返します。

例

ユーザーの認証

ユーザーを認証します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

認証されたユーザーの情報の取得

認証を行い、ユーザーに関する追加情報を返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userid": "1",
        "username": "Admin",
        "name": "Zabbix",
        "surname": "Administrator",
        "url": "",
        "autologin": "1",
        "autologout": "0",
        "lang": "ru_RU",
        "refresh": "0",
        "theme": "default",
        "attempt_failed": "0",
        "attempt_ip": "127.0.0.1",
        "attempt_clock": "1355919038",
        "rows_per_page": "50",
        "timezone": "Europe/Riga",
        "roleid": "3",
        "userdirectoryid": "0",
        "type": 3,
        "userip": "127.0.0.1",
        "debug_mode": 0,
        "gui_access": "0",
        "mfaid": "1",
        "deprovisioned": false,
        "auth_type": 0,
        "sessionid": "5b56eee8be445e98f0bd42b435736e42",
        "secret": "cd0ba923319741c6586f3d866423a8f4"
    },
    "id": 1
}
```

参照

- user.logout

ソース

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.logout

説明

`string/object user.logout(array)`

このメソッドを使用すると、API からログアウトして、現在の認証トークンを無効にすることができます。

> **Note:**
> このメソッドはどのタイプのユーザーでも利用可能です。このメソッドを呼び出す権限は、ユーザーの役割設定で取り消すことができます。詳しくはユーザーの役割を参照してください。

パラメータ

(array) このメソッドは空の配列を受け入れます。

戻り値

(boolean) ユーザーが正常にログアウトされた場合は `true` を返します

例

ログアウト

API からログアウトします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

参照

- user.login

ソース

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.provision

説明

`object user.provision(object/array users)`

このメソッドは、LDAP ユーザーをプロビジョニングできます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーの役割設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

パラメータ

(array) プロビジョニングするユーザーの ID

戻り値

(object) userids プロパティの下でプロビジョニングされたユーザーの ID を含むオブジェクトを返します。

例

複数ユーザーのプロビジョニング

2 つのユーザーをプロビジョニングします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CUser::provision() in *ui/include/classes/api/services/CUser.php*.

user.resettotp

説明

`object user.resettotp(object/array users)`

このメソッドは、ユーザーの TOTP シークレットをリセットします。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割をご覧ください。

パラメータ

（配列）TOTP シークレットをリセットするユーザーの ID

> **Note:**
> 指定されたユーザーのユーザーセッションも削除されます（リクエストを送信したユーザーを除く）。

戻り値

（object）TOTP シークレットがリセットされたユーザーの ID を `userids` プロパティに格納したオブジェクトを返します。

例

複数のユーザーの TOTP シークレットのリセット

2 人のユーザーの TOTP シークレットをリセットします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
```

```
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

参照

- MFA object

ソース

CUser::resettotp() in *ui/include/classes/api/services/CUser.php*.

user.unblock

説明

`object user.unblock(array userids)`

このメソッドは、ユーザーのブロックを解除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(array) ユーザーブロックを解除する ID

戻り値

(object) userids プロパティの下にあるブロックされていないユーザーの ID を含むオブジェクトを返します。

例

複数のユーザーのブロック解除

2 人のユーザーのブロックを解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
```

```
            "1",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CUser::unblock() in *ui/include/classes/api/services/CUser.php*.

user.update

説明

`object user.update(object/array users)`

このメソッドでは、既存のユーザーを更新できます。

Note:
このメソッドは、任意のタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

Note:
ユーザーパスワードの強度は、Authentication API で定義されたパスワードポリシールールに従って検証されます。詳細については、Authentication APIを参照してください。

パラメータ

(object/array) 更新するユーザープロパティ。

各ユーザーについて userid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のユーザープロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
current_passwd	string	ユーザーの現在のパスワード。 ユーザーがユーザーディレクトリ にリンクされている場合、このパラメータの値は空文字列にできます。 パラメータの動作: - *write-only* - User object の passwd が設定され、ユーザーが自分自身のユーザーパスワードを変更する場合は 必須
usrgrps	array	既存のユーザーグループを置き換えるユーザーグループ。 ユーザーグループでは usrgrpid プロパティのみを定義する必要があります。
medias	array	既存の、プロビジョニングされていないユーザーメディアを置き換えるユーザーメディア。メディアを更新する際、プロビジョニングされたメディアは省略できます。

戻り値

(object) userids プロパティの下で更新されたユーザーの ID を含むオブジェクトを返します。

例

ユーザー名の変更

ユーザー名を John Doe に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

ユーザーの役割の変更

ユーザーの役割を変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

参照

- Authentication

ソース

CUser::update() in *ui/include/classes/api/services/CUser.php*.

Web シナリオ

このクラスは、Web シナリオを操作するために設計されています。

オブジェクト参照:

- Web シナリオ
- シナリオステップ
    - HTTP フィールド

- Web シナリオタグ

利用可能なメソッド:

- httptest.create - 新しい Web シナリオを作成
- httptest.delete - Web シナリオを削除
- httptest.get - Web シナリオを取得
- httptest.update - Web シナリオを更新

Web scenario オブジェクト

次のオブジェクトは、`webcheck`API に直接関連しています。

Web シナリオ

Web シナリオオブジェクトには、以下のプロパティがあります。

Property	Type	Description
httptestid	ID	Web シナリオの ID。 Property behavior: - *read-only* - 更新操作で *required*
hostid	ID	Web シナリオが属するホストの ID。 Property behavior: - *constant* - 作成操作で *required*
name	string	Web シナリオの名前。 Property behavior: - 作成操作で *required*
agent	string	Web シナリオで使用されるユーザーエージェント文字列。 デフォルト: Zabbix
authentication	integer	Web シナリオで使用される認証方式。 指定可能な値: 0 - *(default)* なし; 1 - Basic HTTP 認証; 2 - NTLM 認証; 3 - Kerberos 認証; 4 - Digest 認証。
delay	string	Web シナリオの実行間隔。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）、あるいはユーザーマクロを指定できます。 デフォルト: 1m。
headers	array	リクエスト実行時に送信されるHTTP ヘッダー。
http_password	string	Basic HTTP、NTLM、Kerberos、または Digest 認証で使用されるパスワード。
http_proxy	string	*http://[username[:password]@]proxy.example.com[:port]* の形式で指定する、Web シナリオで使用されるプロキシ。
http_user	string	Basic HTTP、NTLM、Kerberos、または Digest 認証で使用されるユーザー名。
retries	integer	失敗する前に、Web シナリオが各ステップの実行を試行する回数。 デフォルト: 1。
ssl_cert_file	string	クライアント認証に使用される SSL 証明書ファイルの名前（PEM 形式である必要があります）。
ssl_key_file	string	クライアント認証に使用される SSL 秘密鍵ファイルの名前（PEM 形式である必要があります）。
ssl_key_password	string	SSL 秘密鍵のパスワード。

Property	Type	Description
status	integer	Web シナリオが有効かどうか。 指定可能な値: 0 - *(default)* 有効; 1 - 無効。
templateid	ID	親テンプレートの Web シナリオの ID。 Property behavior: - *read-only*
variables	array	Web シナリオの変数。
verify_host	integer	接続先のホスト名が、そのホストの証明書内のホスト名と一致することを検証するかどうか。 指定可能な値: 0 - *(default)* ホスト検証をスキップ; 1 - ホストを検証。
verify_peer	integer	ホストの証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(default)* ピア検証をスキップ; 1 - ピアを検証。
uuid	string	グローバル一意識別子。インポートした Web シナリオを既存のものにリンクするために使用されます。指定しない場合は自動生成されます。 Property behavior: - Web シナリオがテンプレートに属する場合に *supported*

シナリオステップ

シナリオステップオブジェクトは、特定の Web シナリオチェックを定義します。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
name	string	シナリオステップの名前。 Property behavior: - *required*
no	integer	Web シナリオ内のステップの順序番号。 Property behavior: - *required*
url	string	チェック対象の URL。 Property behavior: - *required*
follow_redirects	integer	HTTP リダイレクトに従うかどうか。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。
headers	array	リクエスト実行時に送信されるHTTP ヘッダー。シナリオステップのヘッダーは、Web シナリオに指定されたヘッダーを上書きします。
posts	string/array	HTTP POST 変数。文字列（生の POST データ）またはHTTP フィールドの配列（フォームフィールドデータ）として指定します。
required	string	レスポンス内に存在しなければならないテキスト。
retrieve_mode	integer	シナリオステップが取得する必要のある HTTP レスポンスの部分。 指定可能な値: 0 - *(デフォルト)* 本文のみ; 1 - ヘッダーのみ; 2 - ヘッダーと本文。

Property	Type	Description
status_codes	string	必要な HTTP ステータスコードの範囲。カンマ区切りで指定します。
timeout	string	リクエストのタイムアウト（秒）。秒数、接尾辞付きの時間単位、またはユーザーマクロを指定できます。 デフォルト: 15s。最大: 1h。最小: 1s。
variables	array	シナリオステップの変数。
query_fields	array	クエリフィールド - リクエスト実行時に URL へ追加されるHTTP フィールド の配列。

HTTP フィールド

HTTP フィールドオブジェクトは、Web シナリオ変数、HTTP ヘッダー、POST フィールドまたはクエリフィールドを指定するために使用される名前と値を定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	ヘッダー/変数/POST または GET フィールドの名前。 プロパティの動作: - 必須
value	string	ヘッダー/変数/POST または GET フィールドの値。 プロパティの動作: - 必須

Web シナリオタグ

Web シナリオタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	Web シナリオタグ名。 プロパティの動作: - 必須
value	string	Web シナリオタグの値。

httptest.create

説明

`object httptest.create(object/array webScenarios)`

このメソッドでは、新しい Web シナリオを作成できます。

Note:
Web シナリオを作成すると、一連のWeb 監視アイテムが自動的に作成されます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

`(object/array)` 作成する Web シナリオ。

このメソッドは、標準の Web シナリオプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
steps	array	シナリオステップ。 パラメータの動作: - 必須

パラメータ	Type	説明
tags	array	Web シナリオタグ。

戻り値

(object) httptestids プロパティの下で作成された Web シナリオの ID を含むオブジェクトを返します。返される ID の順序は、渡された Web シナリオの順序と一致します。

例

Web シナリオの作成

会社のホームページを監視するための Web シナリオを作成します。このシナリオには 2 つのステップがあり、ホームページと「About」ページを確認して、HTTP ステータスコード 200 が返されることを確認します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.create",
    "params": {
        "name": "Homepage check",
        "hostid": "10085",
        "steps": [
            {
                "name": "Homepage",
                "url": "http://example.com",
                "status_codes": "200",
                "no": 1
            },
            {
                "name": "Homepage / About",
                "url": "http://example.com/about",
                "status_codes": "200",
                "no": 2
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- シナリオステップ

ソース

CHttpTest::create() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

説明

`object httptest.delete(array webScenarioIds)`

このメソッドは、Web シナリオを削除することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する Web シナリオの ID。

戻り値

(object) httptestids プロパティの下にある削除された Web シナリオの ID を含むオブジェクトを返します。

例

複数の Web シナリオの削除

2 つの Web シナリオを削除します

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

ソース

CHttpTest::delete() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

説明

integer/array httptest.get(object parameters)

このメソッドでは、指定されたパラメータに従って Web シナリオを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
groupids	ID/array	指定したホストグループに属する Web シナリオのみを返します。
hostids	ID/array	指定したホストに属する Web シナリオのみを返します。

パラメータ	型	説明
httptestids	ID/array	指定した ID を持つ Web シナリオのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承された Web シナリオのみを返します。
monitored	boolean	`true` に設定した場合、監視対象ホストに属する有効な Web シナリオのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属する Web シナリオのみを返します。
templateids	ID/array	指定したテンプレートに属する Web シナリオのみを返します。
expandName	flag	Web シナリオ名内のマクロを展開します。
expandStepName	flag	シナリオステップ名内のマクロを展開します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	array	指定したタグを持つ Web シナリオのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべての Web シナリオを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
selectHosts	query	Web シナリオが属するホストを、`hosts`プロパティに配列として返します。
selectSteps	query	Web シナリオのステップを`steps`プロパティに返します。 `count` をサポートします。
selectTags	query	Web シナリオのタグを`tags`プロパティに返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `httptestid`, `name`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

`(integer/array)` 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

Web シナリオの取得

Web シナリオ「9」に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "httptestid": "9",
            "name": "Homepage check",
            "delay": "1m",
            "status": "0",
            "variables": [],
            "agent": "Zabbix",
            "authentication": "0",
            "http_user": "",
            "http_password": "",
            "hostid": "10084",
            "templateid": "0",
            "http_proxy": "",
            "retries": "1",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "headers": [],
            "steps": [
                {
                    "httpstepid": "36",
                    "httptestid": "9",
                    "name": "Homepage",
                    "no": "1",
                    "url": "http://example.com",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "variables": [
                        {
                            "name":"{var}",
                            "value":"12"
                        }
                    ],
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "headers": [],
                    "query_fields": []
                },
                {
                    "httpstepid": "37",
                    "httptestid": "9",
                    "name": "Homepage / About",
                    "no": "2",
```

```
                    "url": "http://example.com/about",
                    "timeout": "15s",
                    "posts": "",
                    "required": "",
                    "status_codes": "200",
                    "variables": [],
                    "follow_redirects": "1",
                    "retrieve_mode": "0",
                    "headers": [],
                    "query_fields": []
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- シナリオステップ

ソース

CHttpTest::get() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

説明

`object httptest.update(object/array webScenarios)`

このメソッドでは、既存の Web シナリオを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
> このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object/array) 更新する Web シナリオのプロパティ。

各 Web シナリオに対して `httptestid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準の Web シナリオプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
steps	array	既存のステップを置き換えるシナリオステップ。
tags	array	Web シナリオタグ。

戻り値

(object) `httptestid` プロパティの下で更新された Web シナリオの ID を含むオブジェクトを返します。

例

Web シナリオの有効化

Web シナリオを有効化します。つまり、そのステータスを「0」に設定します。

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
```

```
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- シナリオステップ

ソース

CHttpTest::update() in *ui/include/classes/api/services/CHttpTest.php*.

アイコンマップ

このクラスは、アイコンマップを操作するために設計されています。

オブジェクト参照:

- アイコンマップ
- アイコンマッピング

利用可能なメソッド:

- iconmap.create - 新しいアイコンマップを作成
- iconmap.delete - アイコンマップを削除
- iconmap.get - アイコンマップを取得
- iconmap.update - アイコンマップを更新

Icon map オブジェクト

次のオブジェクトは、`iconmap`API に直接関連しています。

アイコンマップ

アイコンマップオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
iconmapid	ID	アイコンマップの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
default_iconid	ID	デフォルトアイコンの ID。 プロパティの動作: - 作成操作で 必須
name	string	アイコンマップの名前。 プロパティの動作: - 作成操作で 必須

アイコンマッピング

アイコンマッピングオブジェクトは、特定のインベントリフィールド値を持つホストに使用する特定のアイコンを定義します。以下のプロパティがあります。

プロパティ	Type	説明
iconid	ID	アイコンマッピングで使用されるアイコンの ID。 Property behavior: - *required*
expression	string	インベントリフィールドと照合するための式。 Property behavior: - *required*
inventory_link	integer	ホストインベントリフィールドの ID。 サポートされているインベントリフィールドの一覧については、ホストインベントリオブジェクトを参照してください。 Property behavior: - *required*
sortorder	integer	アイコンマップ内でのアイコンマッピングの位置。 Property behavior: - *read-only*

iconmap.create

説明

`object iconmap.create(object/array iconMaps)`

このメソッドでは、新しいアイコンマップを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 作成するアイコンマップ。

標準のアイコンマップのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
mappings	array	アイコンマップ用に作成するアイコンマッピング。 パラメータの動作: - 必須

戻り値

(`object`) `iconmapids` プロパティの下で作成されたアイコンマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイコンマップの順序と一致します。

例

アイコンマップの作成

異なるタイプのホストを表示するためのアイコンマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.create",
    "params": {
```

```
        "name": "Type icons",
        "default_iconid": "2",
        "mappings": [
            {
                "inventory_link": 1,
                "expression": "server",
                "iconid": "3"
            },
            {
                "inventory_link": 1,
                "expression": "switch",
                "iconid": "4"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- アイコンマッピング

ソース

CIconMap::create() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

説明

`object iconmap.delete(array iconMapIds)`

このメソッドは、アイコンマップを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアイコンマップの ID。

戻り値

(object) iconmapids プロパティの下にある削除されたアイコンマップの ID を含むオブジェクトを返します。

例

複数のアイコンマップの削除

2 つのアイコンマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
```

```
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CIconMap::delete() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

説明

`integer/array iconmap.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってアイコンマップを取得できます。

Note:
このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
iconmapids	ID/array	指定した ID のアイコンマップのみを返します。
sysmapids	ID/array	指定したマップで使用されているアイコンマップのみを返します。
selectMappings	query	使用されているアイコンマッピングを含む`mappings` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `iconmapid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

アイコンマップを取得する

アイコンマップ「3」に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mappings": [
                {
                    "iconmappingid": "3",
                    "iconmapid": "3",
                    "iconid": "6",
                    "inventory_link": "1",
                    "expression": "server",
                    "sortorder": "0"
                },
                {
                    "iconmappingid": "4",
                    "iconmapid": "3",
                    "iconid": "10",
                    "inventory_link": "1",
                    "expression": "switch",
                    "sortorder": "1"
                }
            ],
            "iconmapid": "3",
            "name": "Host type icons",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

関連項目

- アイコンマッピング

ソース

CIconMap::get() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

説明

`object iconmap.update(object/array iconMaps)`

このメソッドは、既存のアイコンマップを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 更新するアイコンマップのプロパティ。

各アイコンマップでは `iconmapid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

標準のアイコンマッププロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
mappings	array	既存のアイコンマッピングを置き換えるアイコンマッピング。

戻り値

(object) `iconmapids` プロパティの下にある更新されたアイコンマップの ID を含むオブジェクトを返します

例

アイコンマップの名前変更

アイコンマップの名前を”OS アイコン” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- アイコンマッピング

ソース

CIconMap::update() in *ui/include/classes/api/services/CIconMap.php*.

アイテム

このクラスはアイテムを操作するために設計されています。

オブジェクト参照:

- Item

    - HTTP header
    - HTTP query field
- Item tag
- Item preprocessing

利用可能なメソッド:

- item.create - 新しいアイテムを作成
- item.delete - アイテムを削除
- item.get - アイテムを取得
- item.update - アイテムを更新

Item オブジェクト

次のオブジェクトは、itemAPI に直接関連しています。

アイテム

> **Note:**
> Web アイテムは、Zabbix API 経由で直接作成、更新、削除することはできません。

アイテムオブジェクトには、以下のプロパティがあります。

Property	Type	Description
itemid	ID	アイテムの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
delay	string	アイテムの更新間隔。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）を受け付けます。また、必要に応じて、セミコロンで区切られた 1 つ以上のカスタム間隔も指定できます。カスタム間隔には、柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロを受け付けます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはテキストと混在したマクロはサポートされません。柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます（例: {$FLEX_INTERVAL}/{$FLEX_PERIOD}）。 例: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr プロパティの動作: - type が”Zabbix agent” (0)、”Simple check” (3)、”Zabbix internal” (5)、”External check” (10)、”Database monitor” (11)、”IPMI agent” (12)、”SSH agent” (13)、”TELNET agent” (14)、”Calculated” (15)、”JMX agent” (16)、”HTTP agent” (19)、”SNMP agent” (20)、”Script” (21)、”Browser” (22) に設定されている場合、または type が”Zabbix agent (active)” (7) に設定され、かつ key_ に”mqtt.get” が含まれない場合に 必須
hostid	ID	アイテムが属するホストまたはテンプレートの ID。 プロパティの動作: - 不変 - 作成操作で 必須

Property	Type	Description
interfaceid	ID	アイテムのホストインターフェースの ID。 プロパティの動作: - アイテムがホストに属し、type が”Zabbix agent”、”IPMI agent”、”JMX agent”、”SNMP trap”、または”SNMP agent” に設定されている場合に 必須 - アイテムがホストに属し、type が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合に サポート - ディスカバリされたオブジェクトでは 読み取り専用
key_	string	アイテムキー。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
name	string	アイテム名。 ユーザーマクロをサポートします。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
name_resolved	string	ユーザーマクロを展開したアイテム名。 プロパティの動作: - 読み取り専用
type	integer	アイテムのタイプ。 指定可能な値: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用

Property	Type	Description
url	string	URL 文字列。 ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が”HTTP agent” に設定されている場合に 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
value_type	integer	アイテムの情報タイプ。 指定可能な値: 0 - 数値（浮動小数）; 1 - 文字列; 2 - ログ; 3 - 数値（符号なし）; 4 - テキスト; 5 - バイナリ。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
allow_traps	integer	trapper アイテムと同様に値を投入できるようにします。 0 - *(デフォルト)* 受信データの受け付けを許可しない; 1 - 受信データの受け付けを許可する。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - ディスカバリされたオブジェクトでは 読み取り専用
authtype	integer	認証方式。 type が”SSH agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest。 プロパティの動作: - type が”SSH agent” または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクト（type が”HTTP agent” の場合）またはディスカバリされたオブジェクトでは 読み取り専用
description	string	アイテムの説明。 プロパティの動作: - ディスカバリされたオブジェクトでは 読み取り専用
error	string	アイテム値の更新に問題がある場合のエラーテキスト。 プロパティの動作: - 読み取り専用

Property	Type	Description
flags	integer	アイテムの生成元。 指定可能な値: 0 - 通常のアイテム; 4 - プロトタイプから変換されたアイテム。 プロパティの動作: - 読み取り専用
follow_redirects	integer	データ取得時にレスポンスのリダイレクトに従うかどうか。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
history	string	履歴データを保存する期間を表す時間単位。 ユーザーマクロも受け付けます。 デフォルト: 31d。 プロパティの動作: - ディスカバリされたオブジェクトでは 読み取り専用
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
inventory_link	integer	アイテムによって設定されるホストインベントリフィールドの ID。 サポートされるホストインベントリフィールドとその ID の一覧については、ホストインベントリページを参照してください。 デフォルト: 0。 プロパティの動作: - `value_type` が”numeric float”、”character”、”numeric unsigned”、または”text” に設定されている場合に サポート - ディスカバリされたオブジェクトでは 読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - `type` が”IPMI agent” に設定され、かつ `key_` が”ipmi.get” に設定されていない場合に 必須 - `type` が”IPMI agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用

Property	Type	Description
jmx_endpoint	string	JMX agent のカスタム接続文字列。 デフォルト値: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - `type` が”JMX agent” に設定されている場合に サポート - ディスカバリされたオブジェクトでは 読み取り専用
lastclock	timestamp	アイテム値が最後に更新された時刻。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は、*Administration → General* メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの動作: - 読み取り専用
lastns	integer	アイテム値が最後に更新された時点のナノ秒。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は、*Administration → General* メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの動作: - 読み取り専用
lastvalue	string	アイテムの最新値。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は、*Administration → General* メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの動作: - 読み取り専用
logtimefmt	string	ログエントリ内の時刻の形式。 プロパティの動作: - `value_type` が”log” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
master_itemid	ID	マスターアイテムの ID。 プロパティの動作: - `type` が”Dependent item” に設定されている場合に 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 0 - *(デフォルト)* 生データのまま保存; 1 - JSON に変換。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用

Property	Type	Description
params	string	アイテムのタイプに応じた追加パラメータ: - SSH agent および TELNET agent アイテムでは実行するスクリプト; - Database monitor アイテムでは SQL クエリ; - Calculated アイテムでは数式; - Script および Browser アイテムではスクリプト。 プロパティの動作: - `type` が”Database monitor”、”SSH agent”、”TELNET agent”、”Calculated”、”Script”、または”Browser” に設定されている場合に 必須 - 継承されたオブジェクト（`type` が”Script” または”Browser” の場合）またはディスカバリされたオブジェクトでは 読み取り専用
parameters	object/array	`type` が”Script” または”Browser” に設定されている場合の追加パラメータ。`name` と `value` プロパティを持つオブジェクトの配列で、`name` は一意である必要があります。 プロパティの動作: - `type` が”Script” または”Browser” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
password	string	認証用パスワード。 プロパティの動作: - `type` が”JMX agent” に設定され、かつ `username` が設定されている場合に 必須 - `type` が”Simple check”、”SSH agent”、”TELNET agent”、”Database monitor”、または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクト（`type` が”HTTP agent” の場合）またはディスカバリされたオブジェクトでは 読み取り専用
post_type	integer	`posts` プロパティに格納される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ; 2 - JSON データ; 3 - XML データ。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
posts	string	HTTP(S) リクエスト本文データ。 プロパティの動作: - `type` が”HTTP agent” に設定され、かつ `post_type` が”JSON data” または”XML data” に設定されている場合に 必須 - `type` が”HTTP agent” に設定され、かつ `post_type` が”Raw data” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
prevvalue	string	アイテムの前回値。 デフォルトでは、直近 24 時間以内の値のみが表示されます。この期間は、*Administration → General* メニューセクションの *Max history display period* パラメータの値を変更することで延長できます。 プロパティの動作: - 読み取り専用
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” に設定され、かつ `authtype` が”public key” に設定されている場合に 必須 - ディスカバリされたオブジェクトでは 読み取り専用

Property	Type	Description
publickey	string	公開鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” に設定され、かつ `authtype` が”public key” に設定されている場合に 必須 - ディスカバリされたオブジェクトでは 読み取り専用
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、または”PUT” に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が”HEAD” に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
snmp_oid	string	SNMP OID。 プロパティの動作: - `type` が”SNMP agent” に設定されている場合に 必須 - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
ssl_cert_file	string	公開 SSL 鍵ファイルのパス。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
ssl_key_file	string	秘密 SSL 鍵ファイルのパス。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用

Property	Type	Description
ssl_key_password	string	SSL 鍵ファイルのパスワード。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
state	integer	アイテムの状態。 指定可能な値: 0 - *(*デフォルト*)* 正常; 1 - サポートされていない。 プロパティの動作: - 読み取り専用
status	integer	アイテムのステータス。 指定可能な値: 0 - *(*デフォルト*)* 有効なアイテム; 1 - 無効なアイテム。
status_codes	string	必要な HTTP ステータスコードの範囲。カンマ区切りで指定します。 カンマ区切りリストの一部としてユーザーマクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
templateid	ID	親テンプレートアイテムの ID。 ヒント: アイテムが属するテンプレートを指定するには、hostid プロパティを使用します。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータのポーリングリクエストのタイムアウト。 秒、またはサフィックス付きの時間単位（例: 30s、1m）を受け付けます。 ユーザーマクロも受け付けます。 指定可能な範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの動作: - type が”Zabbix agent” (0)、”Simple check” (3) で key_ が”vmware.” および”icmpping” で始まらない場合、”Zabbix agent (active)” (7)、”External check” (10)、”Database monitor” (11)、”SSH agent” (13)、”TELNET agent” (14)、”HTTP agent” (19)、”SNMP agent” (20) で snmp_oid が”walk[” または”get[” で始まる場合、”Script” (21)、”Browser” (22) に設定されている場合に サポート - 継承されたオブジェクトおよびディスカバリされたオブジェクトでは 読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - ディスカバリされたオブジェクトでは *readonly* - type が”Zabbix trapper” に設定されている場合、または type が”HTTP agent” に設定され、かつ allow_traps が”Allow to accept incoming data” に設定されている場合に サポート

Property	Type	Description
trends	string	トレンドデータを保存する期間を表す時間単位。 ユーザーマクロも受け付けます。 デフォルト: 365d。 プロパティの動作: - `value_type` が”numeric float” または”numeric unsigned” に設定されている場合に サポート - ディスカバリされたオブジェクトでは 読み取り専用
units	string	値の単位。 プロパティの動作: - `value_type` が”numeric float” または”numeric unsigned” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
username	string	認証用ユーザー名。 プロパティの動作: - `type` が”SSH agent”、”TELNET agent” に設定されている場合、または `type` が”JMX agent” に設定され、かつ `password` が設定されている場合に 必須 - `type` が”Simple check”、”Database monitor”、または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクト（`type` が”HTTP agent” の場合）またはディスカバリされたオブジェクトでは 読み取り専用
uuid	string	既存のアイテムにインポートしたアイテムを関連付けるために使用される汎用一意識別子。指定しない場合は自動生成されます。 プロパティの動作: - アイテムがテンプレートに属する場合に サポート
valuemapid	ID	関連付けられた値マップの ID。 プロパティの動作: - `value_type` が”numeric float”、”character”、または”numeric unsigned” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
verify_host	integer	接続先のホスト名がホスト証明書内のホスト名と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用
verify_peer	integer	ホスト証明書が正当なものであることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは 読み取り専用

HTTP ヘッダー

ヘッダーオブジェクトには、次のプロパティがあります。

プロパティ	Type	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータの名前。 プロパティの動作: - 必須
value	string	パラメータの値。 プロパティの動作: - 必須

アイテムタグ

アイテムタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	アイテムタグ名。 プロパティの動作: - 必須
value	string	アイテムタグの値。

アイテムの前処理

アイテムの前処理オブジェクトには、以下のプロパティがあります。

Property	Type	Description
type	integer	前処理オプションのタイプです。 指定可能な値: 1 - カスタム乗数; 2 - 右トリム; 3 - 左トリム; 4 - トリム; 5 - 正規表現; 6 - 真偽値から 10 進数; 7 - 8 進数から 10 進数; 8 - 16 進数から 10 進数; 9 - 単純変化; 10 - 1 秒あたりの変化; 11 - XML XPath; 12 - JSONPath; 13 - 範囲内; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON 内のエラーをチェック; 17 - XML 内のエラーをチェック; 18 - 正規表現を使用してエラーをチェック; 19 - 未変更を破棄; 20 - ハートビート付きで未変更を破棄; 21 - JavaScript; 22 - Prometheus パターン; 23 - Prometheus から JSON; 24 - CSV から JSON; 25 - 置換; 26 - 未サポートをチェック; 27 - XML から JSON; 28 - SNMP walk 値; 29 - SNMP walk から JSON; 30 - SNMP get 値。 Property behavior: - *required*
params	string	前処理オプションで使用される追加パラメータです。 複数のパラメータは改行文字 (\n) で区切られます。 type が「未サポートをチェック」に設定されている場合、パラメータは <scope>[\n<pattern>] 構文に従います。ここで、*pattern* は正規表現、*scope* は次のいずれかです: -1 - 任意のエラーに一致; 0 - エラーメッセージが *pattern* に一致するかをチェック; 1 - エラーメッセージが *pattern* に一致しないかをチェック。 Property behavior: - type が「カスタム乗数」(1)、「右トリム」(2)、「左トリム」(3)、「トリム」(4)、「正規表現」(5)、「XML XPath」(11)、「JSONPath」(12)、「範囲内」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON 内のエラーをチェック」(16)、「XML 内のエラーをチェック」(17)、「正規表現を使用してエラーをチェック」(18)、「ハートビート付きで未変更を破棄」(20)、「JavaScript」(21)、「Prometheus パターン」(22)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「置換」(25)、「未サポートをチェック」(26)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は *required*

Property	Type	Description
error_handler	integer	前処理ステップの失敗時に使用されるアクションタイプです。 指定可能な値: 0 - エラーメッセージは Zabbix server によって設定される; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 type が「未サポートをチェック」に設定されている場合の指定可能な値: 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 Property behavior: - type が「カスタム乗数」(1)、「正規表現」(5)、「真偽値から 10 進数」(6)、「8 進数から 10 進数」(7)、「16 進数から 10 進数」(8)、「単純変化」(9)、「1 秒あたりの変化」(10)、「XML XPath」(11)、「JSONPath」(12)、「範囲内」(13)、「正規表現に一致」(14)、「正規表現に一致しない」(15)、「JSON 内のエラーをチェック」(16)、「XML 内のエラーをチェック」(17)、「正規表現を使用してエラーをチェック」(18)、「Prometheus パターン」(22)、「Prometheus から JSON」(23)、「CSV から JSON」(24)、「未サポートをチェック」(26)、「XML から JSON」(27)、「SNMP walk 値」(28)、「SNMP walk から JSON」(29)、または「SNMP get 値」(30) に設定されている場合は *required*
error_handler_params	string	エラーハンドラーのパラメータです。 Property behavior: - error_handler が「カスタム値を設定」または「カスタムエラーメッセージを設定」に設定されている場合は *required*

以下のパラメータとエラーハンドラーが、各前処理タイプでサポートされています。

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
1	カスタム乗数	number[1, 6]			0, 1, 2, 3
2	右トリム	文字のリスト[2]			
3	左トリム	文字のリスト[2]			
4	トリム	文字のリスト[2]			
5	正規表現	pattern[3]	output[2]		0, 1, 2, 3
6	真偽値から 10 進数				0, 1, 2, 3
7	8 進数から 10 進数				0, 1, 2, 3
8	16 進数から 10 進数				0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
9	単純変化				0, 1, 2, 3
10	1 秒あたりの変化				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[4]			0, 1, 2, 3
13	範囲内	min[1, 6]	max[1, 6]		0, 1, 2, 3
14	正規表現に一致	pattern[3]			0, 1, 2, 3
15	正規表現に一致しない	pattern[3]			0, 1, 2, 3
16	JSON 内のエラーをチェック	path[4]			0, 1, 2, 3
17	XML 内のエラーをチェック	path[4]			0, 1, 2, 3
18	正規表現を使用してエラーをチェック	pattern[3]	output[2]		0, 1, 2, 3
19	未変更を破棄				
20	ハートビート付きで未変更を破棄	seconds[5, 6]			
21	JavaScript	script[2]			
22	Prometheus パターン	pattern[6, 7]	`value`, `label`, `function`	output[8, 9]	0, 1, 2, 3
23	Prometheus から JSON	pattern[6, 7]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
24	CSV から JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	置換	search string[2]	replacement[2]		
26	未サポートをチェック	scope[1]	pattern[3, 6]		1, 2, 3
27	XML から JSON				0, 1, 2, 3
28	SNMP walk 値	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk から JSON[10]	フィールド名[2]	OID プレフィックス[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get 値	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 浮動小数点数 (整数は暗黙的に浮動小数点値に変換されます)
[2] 文字列
[3] 正規表現
[4] JSONPath または XML XPath
[5] 正の整数 (時間サフィックスをサポート。例: 30s, 1m, 2h, 1d)
[6] ユーザーマクロ
[7] 次の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。各 Prometheus パターンの構成要素 (メトリクス、ラベル名、ラベル値、メトリクス値) にはユーザーマクロを使用できます。
[8] 次の構文に従う Prometheus 出力: 2 番目のパラメータとして `label` が選択されている場合は `<label name>` (ユーザーマクロを使用可能)。
[9] 集計関数のいずれか: 2 番目のパラメータとして `function` が選択されている場合は `sum`, `min`, `max`, `avg`, `count`。
[10] 改行文字で区切られた複数の「Field name,OID prefix,Format records」レコードをサポートします。

item.create

説明

```
object item.create(object/array items)
```

このメソッドでは、新しいアイテムを作成できます。

> **Note:**
> Web 監視アイテムは Zabbix API 経由では作成できません。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するアイテム。

標準のアイテムプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
preprocessing	array	アイテムの前処理オプション。
tags	array	アイテムタグ。

戻り値

(object) itemids プロパティの下に、作成されたアイテムの ID を含むオブジェクトを返します。
返される ID の順序は、渡されたアイテムの順序と一致します。

例

アイテムの作成

ID が"30074" のホスト上の空きディスク容量を監視するために、2 つのアイテムタグを持つ数値型の Zabbix エージェントアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "/home/joe/ の空きディスク容量",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "30074",
        "type": 0,
        "value_type": 3,
        "interfaceid": "30084",
        "tags": [
            {
                "tag": "component",
                "value": "storage"
            },
            {
                "tag": "equipment",
                "value": "workstation"
            }
        ],
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

ホストインベントリのアイテムの作成

ホストの「OS」インベントリフィールドを設定するための Zabbix エージェントのアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "uname",
        "key_": "system.uname",
        "hostid": "30021",
        "type": 0,
        "interfaceid": "30007",
        "value_type": 1,
        "delay": "10s",
        "inventory_link": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

前処理を使用したアイテムの作成

カスタム乗数を使用してアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Device uptime",
        "key_": "sysUpTime",
        "hostid": "10084",
        "type": 20,
        "snmp_oid": "SNMPv2-MIB::sysUpTime.0",
        "value_type": 1,
        "delay": "60s",
        "interfaceid": "83",
        "preprocessing": [
            {
                "type": 1,
                "params": "0.01",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

依存アイテムの作成

ID 24759 のマスターアイテムに対する依存アイテムを作成します。
同じホスト上の依存関係のみ許可されるため、マスターアイテムと依存アイテムは同じ hostid を持っている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
      "hostid": "30074",
      "name": "Dependent test item",
      "key_": "dependent.item",
      "type": 18,
      "master_itemid": "24759",
      "value_type": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムの作成

JSON レスポンスの前処理を使用して、POST リクエストメソッドのアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "url":"http://127.0.0.1/http.php",
        "query_fields": [
            {
                "name": "mode",
                "value": "json"
            },
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value": "100"
            }
        ],
        "interfaceid": "1",
        "type": 19,
```

```
        "hostid": "10254",
        "delay": "5s",
        "key_": "json",
        "name": "HTTP agent example JSON",
        "value_type": 0,
        "output_format": 1,
        "preprocessing": [
            {
                "type": 12,
                "params": "$.random",
                "error_handler": 0,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

スクリプトアイテムの作成

スクリプトアイテムを使用して、シンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.item",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
```

```
    },
    "id": 1
}
```

ソース

CItem::create() in *ui/include/classes/api/services/CItem.php*.

item.delete

説明

`object item.delete(array itemIds)`

このメソッドは、アイテムを削除することができます。

> **Note:**
> Web アイテムは、Zabbix API を使用しての削除はできません。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアイテムの ID。

戻り値

(object) itemids プロパティの下にある削除されたアイテムの ID を含むオブジェクトを返します。

例

複数のアイテムの削除

2 つのアイテムを削除します。
マスターアイテムが削除されると、依存アイテムとアイテムのプロトタイプは自動的に削除されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

ソース

CItem::delete() in *ui/include/classes/api/services/CItem.php*.

item.get

説明

`integer/array item.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってアイテムを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	型	説明
itemids	ID/array	指定した ID のアイテムのみを返します。
groupids	ID/array	指定したグループのホストに属するアイテムのみを返します。
templateids	ID/array	指定したテンプレートに属するアイテムのみを返します。
hostids	ID/array	指定したホストに属するアイテムのみを返します。
proxyids	ID/array	指定したプロキシによって監視されているアイテムのみを返します。
interfaceids	ID/array	指定したホストインターフェースを使用するアイテムのみを返します。
graphids	ID/array	指定したグラフで使用されているアイテムのみを返します。
triggerids	ID/array	指定したトリガーで使用されているアイテムのみを返します。
webitems	flag	結果に Web アイテムを含めます。
inherited	boolean	`true` に設定すると、テンプレートから継承されたアイテムのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するアイテムのみを返します。
monitored	boolean	`true` に設定すると、監視中のホストに属する有効なアイテムのみを返します。
group	string	指定した名前のグループに属するアイテムのみを返します。
host	string	指定した名前のホストに属するアイテムのみを返します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	array	指定したタグを持つアイテムのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのアイテムを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
with_triggers	boolean	`true` に設定すると、トリガーで使用されているアイテムのみを返します。
selectHosts	query	アイテムが属するホストの配列を持つ`hosts` プロパティを返します。
selectInterfaces	query	アイテムが使用するホストインターフェースの配列を持つ`interfaces` プロパティを返します。
selectTriggers	query	アイテムが使用されているトリガーを持つ`triggers` プロパティを返します。 `count` をサポートします。
selectGraphs	query	アイテムを含むグラフを持つ`graphs` プロパティを返します。 `count` をサポートします。

パラメータ	型	説明
selectDiscoveryData	query	アイテムディスカバリオブジェクトのデータを持つ discoveryData プロパティを返します。アイテムディスカバリオブジェクトは、ディスカバリされたアイテムを、そのアイテムの検出元であるアイテムプロトタイプに関連付けます。 以下のプロパティを持ちます: parent_itemid - (string) アイテムの作成元となったアイテムプロトタイプの ID; key_ - (string) アイテムプロトタイプのキー; status - (int) アイテムディスカバリのステータス: 0 - *(デフォルト)* アイテムはディスカバリされています; 1 - アイテムはすでにディスカバリされていません; ts_delete - (timestamp) すでにディスカバリされていないアイテムが削除される時刻; ts_disable - (timestamp) すでにディスカバリされていないアイテムが無効化される時刻; disable_source - (int) アイテムが LLD ルールによって無効化されたか、手動で無効化されたかを示すインジケーター: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	アイテムを作成した LLD ルールを持つdiscoveryRule プロパティを返します。
selectPreprocessing	query	アイテムの前処理オプションを持つpreprocessing プロパティを返します。
selectTags	query	tags プロパティでアイテムタグを返します。
selectValueMap	query	アイテムの値マップを持つvaluemap プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 text データ型 のプロパティはサポートしません。 追加のプロパティをサポートします: host - アイテムが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectGraphs - 結果は name でソートされます; selectTriggers - 結果は description でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: itemid, name, key_, delay, history, trends, type, status。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	アイテムディスカバリオブジェクトを持つ itemDiscovery プロパティを返します。アイテムディスカバリオブジェクトは、アイテムを、その作成元であるアイテムプロトタイプに関連付けます。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

キーでアイテムを検索する

特定のホスト ID について、トリガーで使用されているすべてのアイテムのうち、アイテムキーに”system.cpu” という語を含むものを取得し、結果を名前でソートします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "42269",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "CPU utilization",
            "key_": "system.cpu.util",
            "delay": "0",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "%",
            "logtimefmt": "",
            "templateid": "42267",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "CPU utilization in %.",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "42264",
```

```
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "CPU utilization"
        },
        {
            "itemid": "42259",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (15m avg)",
            "key_": "system.cpu.load[all,avg15]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "42219",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
```

```
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (15m avg)"
        },
        {
            "itemid": "42249",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (1m avg)",
            "key_": "system.cpu.load[all,avg1]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "42209",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
```

```
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (1m avg)"
        },
        {
            "itemid": "42257",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (5m avg)",
            "key_": "system.cpu.load[all,avg5]",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "42217",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
```

```
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Load average (5m avg)"
        },
        {
            "itemid": "42260",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Number of CPUs",
            "key_": "system.cpu.num",
            "delay": "1m",
            "history": "7d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "42220",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "1",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
```

```
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Number of CPUs"
        }
    ],
    "id": 1
}
```

キーで依存アイテムを検索する

ID が”10116” のホストから、キーに”apache” という単語を含むすべての依存アイテムを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25550",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Days",
            "key_": "apache.status.uptime.days",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
```

```
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
        "password": "",
        "publickey": "",
        "privatekey": "",
        "flags": "0",
        "interfaceid": "0",
        "description": "",
        "inventory_link": "0",
        "evaltype": "0",
        "jmx_endpoint": "",
        "master_itemid": "25545",
        "timeout": "",
        "url": "",
        "query_fields": [],
        "posts": "",
        "status_codes": "200",
        "follow_redirects": "1",
        "post_type": "0",
        "http_proxy": "",
        "headers": [],
        "retrieve_mode": "0",
        "request_method": "0",
        "output_format": "0",
        "ssl_cert_file": "",
        "ssl_key_file": "",
        "ssl_key_password": "",
        "verify_peer": "0",
        "verify_host": "0",
        "allow_traps": "0",
        "uuid": "",
        "state": "0",
        "error": "",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0",
        "name_resolved": "Days"
    },
    {
        "itemid": "25555",
        "type": "18",
        "snmp_oid": "",
        "hostid": "10116",
        "name": "Hours",
        "key_": "apache.status.uptime.hours",
        "delay": "0",
        "history": "90d",
        "trends": "365d",
        "status": "0",
        "value_type": "3",
        "trapper_hosts": "",
        "units": "",
        "logtimefmt": "",
        "templateid": "0",
        "valuemapid": "0",
        "params": "",
        "ipmi_sensor": "",
        "authtype": "0",
        "username": "",
```

```
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "25545",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0",
            "name_resolved": "Hours"
        }
    ],
    "id": 1
}
```


HTTP エージェントアイテムの検索

特定のホスト ID について、POST ボディタイプが XML の HTTP エージェントアイテムを検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "itemid": "28252",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10255",
            "name": "template item",
            "key_": "ti",
            "delay": "30s",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "localhost",
            "query_fields": [
                {
                    "name": "mode",
                    "value": "xml"
                }
            ],
            "posts": "<body>\r\n<![CDATA[{$MACRO}<foo></bar>]]>\r\n</body>",
            "status_codes": "200",
            "follow_redirects": "0",
            "post_type": "3",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "1",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "uuid": "",
            "state": "0",
            "error": "",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "",
```

```
            "prevvalue": "",
            "name_resolved": "template item"
        }
    ],
    "id": 1
}
```

前処理ルールを持つアイテムの取得

特定のホスト ID について、すべてのアイテムとその前処理ルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP agent example JSON",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

関連項目

- ディスカバリルール
- グラフ
- ホスト
- ホストインターフェース
- トリガー

ソース

CItem::get() in *ui/include/classes/api/services/CItem.php*.

item.update

説明

`object item.update(object/array items)`

このメソッドでは、既存のアイテムを更新できます。

> **Note:**
> Web アイテムは Zabbix API 経由では更新できません。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object/array) 更新するアイテムのプロパティ。

各アイテムに対して `itemid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のアイテムプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
preprocessing	array	現在の前処理オプションを置き換えるためのアイテムの前処理オプション。 パラメータの動作: - 継承されたオブジェクトまたはディスカバリされたオブジェクトでは *read-only*
tags	array	アイテムタグ。 パラメータの動作: - ディスカバリされたオブジェクトでは *read-only*

戻り値

(object) `itemids` プロパティの下で更新されたアイテムの ID を含むオブジェクトを返します。

例

アイテムの有効化

アイテムを有効化します。つまり、ステータスを”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

依存アイテムの更新

依存アイテム名とマスターアイテム ID を更新します。同じホスト上の依存関係のみ許可されるため、マスターアイテムと依存アイテムは同じ hostid を持っている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Dependent item updated name",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムの更新

アイテム値のトラップを有効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

前処理を使用したアイテムの更新

アイテムの前処理ルール「範囲内」を使用してアイテムを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

スクリプトアイテムの更新

スクリプトアイテムを別のスクリプトで更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItem::update() in *ui/include/classes/api/services/CItem.php*.

アイテムのプロトタイプ

このクラスは、アイテムのプロトタイプを操作するために設計されています。

オブジェクト参照:

- アイテムのプロトタイプ
  - HTTP ヘッダー
  - HTTP クエリフィールド
- アイテムのプロトタイプタグ
- アイテムのプロトタイプ前処理

利用可能なメソッド:

- itemprototype.create - 新しいアイテムのプロトタイプを作成
- itemprototype.delete - アイテムのプロトタイプを削除
- itemprototype.get - アイテムのプロトタイプを取得

- itemprototype.update - アイテムのプロトタイプを更新

Item prototype オブジェクト

次のオブジェクトは、itemprototypeAPI に直接関連しています。

アイテムプロトタイプ

アイテムプロトタイプオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	アイテムプロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
delay	string	アイテムプロトタイプの更新間隔。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）を受け付けます。また、必要に応じて、セミコロンで区切られた 1 つ以上のカスタム間隔も指定できます。カスタム間隔には、柔軟な間隔とスケジューリング間隔を混在させることができます。 ユーザーマクロおよび LLD マクロを受け付けます。使用する場合、値は単一のマクロでなければなりません。複数のマクロ、またはテキストと混在したマクロはサポートされません。柔軟な間隔は、スラッシュで区切られた 2 つのマクロとして記述できます（例: {$FLEX_INTERVAL}/{$FLEX_PERIOD}）。 例: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr プロパティの動作: - type が”Zabbix agent” (0)、”Simple check” (3)、”Zabbix internal” (5)、”External check” (10)、”Database monitor” (11)、”IPMI agent” (12)、”SSH agent” (13)、”TELNET agent” (14)、”Calculated” (15)、”JMX agent” (16)、”HTTP agent” (19)、”SNMP agent” (20)、”Script” (21)、”Browser” (22) に設定されている場合、または type が”Zabbix agent (active)” (7) に設定され、かつ key_ に”mqtt.get” が含まれない場合に 必須
hostid	ID	アイテムプロトタイプが属するホストの ID。 プロパティの動作: - 定数 - 作成操作で 必須
interfaceid	ID	アイテムプロトタイプのホストインターフェースの ID。 プロパティの動作: - アイテムプロトタイプがホストに属し、type が”Zabbix agent”、”IPMI agent”、”JMX agent”、”SNMP trap”、または”SNMP agent” に設定されている場合に 必須 - アイテムプロトタイプがホストに属し、type が”Simple check”、”External check”、”SSH agent”、”TELNET agent”、または”HTTP agent” に設定されている場合に サポート
key_	string	アイテムプロトタイプキー。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用

プロパティ	型	説明
name	string	アイテムプロトタイプの名前。 ユーザーマクロをサポートします。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
type	integer	アイテムプロトタイプのタイプ。 指定可能な値: 0 - Zabbix エージェント; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix エージェント (active); 10 - External check; 11 - Database monitor; 12 - IPMI エージェント; 13 - SSH エージェント; 14 - TELNET エージェント; 15 - Calculated; 16 - JMX エージェント; 17 - SNMP trap; 18 - 依存アイテム; 19 - HTTP エージェント; 20 - SNMP エージェント; 21 - Script; 22 - Browser。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
url	string	URL 文字列。 LLD マクロ、ユーザーマクロ、{HOST.IP}、{HOST.CONN}、{HOST.DNS}、{HOST.HOST}、{HOST.NAME}、{HOST.PORT}、{ITEM.ID}、{ITEM.KEY} をサポートします。 プロパティの動作: - type が”HTTP agent” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用
value_type	integer	アイテムプロトタイプの情報の型。 指定可能な値: 0 - 数値（浮動小数）; 1 - 文字列; 2 - ログ; 3 - 数値（符号なし）; 4 - テキスト; 5 - バイナリ。 プロパティの動作: - 作成操作で 必須 - 継承されたオブジェクトでは 読み取り専用
allow_traps	integer	trapper アイテムと同様に値の投入を許可します。 0 - *(デフォルト)* 受信データの受け入れを許可しない; 1 - 受信データの受け入れを許可する。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート

プロパティ	型	説明
authtype	integer	認証方式。 type が”SSH agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* パスワード; 1 - 公開鍵。 type が”HTTP agent” に設定されている場合の指定可能な値: 0 - *(デフォルト)* なし; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest。 プロパティの動作: - type が”SSH agent” または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用（type が”HTTP agent” に設定されている場合）
description	string	アイテムプロトタイプの説明。
flags	integer	アイテムプロトタイプの生成元。 指定可能な値: 2 - アイテムプロトタイプ; 6 - ディスカバリされたアイテムプロトタイプ プロパティの動作: - 読み取り専用
follow_redirects	integer	データ取得時にレスポンスのリダイレクトに従います。 指定可能な値: 0 - リダイレクトに従わない; 1 - *(デフォルト)* リダイレクトに従う。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
headers	array	HTTP リクエスト実行時に送信されるヘッダーの配列。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
history	string	履歴データを保存する期間を表す時間単位。 ユーザーマクロおよび LLD マクロも受け付けます。 デフォルト: 31d。
http_proxy	string	HTTP(S) プロキシ接続文字列。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
ipmi_sensor	string	IPMI センサー。 プロパティの動作: - type が”IPMI agent” に設定され、かつ key_ が”ipmi.get” に設定されていない場合に 必須 - type が”IPMI agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用

プロパティ	型	説明
jmx_endpoint	string	JMX エージェントのカスタム接続文字列。 デフォルト: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi プロパティの動作: - `type` が”JMX agent” に設定されている場合に サポート
logtimefmt	string	ログエントリ内の時刻の形式。 プロパティの動作: - `value_type` が”log” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
master_itemid	ID	マスターアイテムの ID。 プロパティの動作: - `type` が”Dependent item” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用
output_format	integer	レスポンスを JSON に変換するかどうか。 指定可能な値: 0 - *(デフォルト)* 生データのまま保存; 1 - JSON に変換。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
params	string	アイテムプロトタイプのタイプに応じた追加パラメータ: - SSH エージェントおよび TELNET エージェントのアイテムプロトタイプでは実行するスクリプト; - Database monitor のアイテムプロトタイプでは SQL クエリ; - Calculated のアイテムプロトタイプでは数式; - Script および Browser のアイテムプロトタイプではスクリプト。 プロパティの動作: - `type` が”Database monitor”、”SSH agent”、”TELNET agent”、”Calculated”、”Script”、または”Browser” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用（`type` が”Script” または”Browser” に設定されている場合）
parameters	object/array	`type` が”Script” または”Browser” に設定されている場合の追加パラメータ。`name` と `value` プロパティを持つオブジェクトの配列で、`name` は一意である必要があります。 プロパティの動作: - `type` が”Script” または”Browser” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
password	string	認証用パスワード。 プロパティの動作: - `type` が”JMX agent” に設定され、かつ `username` が設定されている場合に 必須 - `type` が”Simple check”、”SSH agent”、”TELNET agent”、”Database monitor”、または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用（`type` が”HTTP agent” に設定されている場合）

プロパティ	型	説明
post_type	integer	`posts` プロパティに格納される POST データ本文のタイプ。 指定可能な値: 0 - *(デフォルト)* 生データ。 2 - JSON データ。 3 - XML データ。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
posts	string	HTTP(S) リクエスト本文データ。 プロパティの動作: - `type` が”HTTP agent” に設定され、かつ `post_type` が”JSON data” または”XML data” に設定されている場合に 必須 - `type` が”HTTP agent” に設定され、かつ `post_type` が”Raw data” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
privatekey	string	秘密鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” に設定され、かつ `authtype` が”public key” に設定されている場合に 必須
publickey	string	公開鍵ファイル名。 プロパティの動作: - `type` が”SSH agent” に設定され、かつ `authtype` が”public key” に設定されている場合に 必須
query_fields	array	HTTP リクエスト実行時に送信されるクエリフィールドの配列。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
request_method	integer	リクエストメソッドのタイプ。 指定可能な値: 0 - *(デフォルト)* GET; 1 - POST; 2 - PUT; 3 - HEAD。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
retrieve_mode	integer	レスポンスのどの部分を保存するか。 `request_method` が”GET”、”POST”、または”PUT” に設定されている場合の指定可能な値: 0 - *(デフォルト)* 本文; 1 - ヘッダー; 2 - 本文とヘッダーの両方を保存。 `request_method` が”HEAD” に設定されている場合の指定可能な値: 1 - ヘッダー。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用

プロパティ	型	説明
snmp_oid	string	SNMP OID。 プロパティの動作: - type が”SNMP agent” に設定されている場合に 必須 - 継承されたオブジェクトでは 読み取り専用
ssl_cert_file	string	公開 SSL 鍵ファイルのパス。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
ssl_key_file	string	秘密 SSL 鍵ファイルのパス。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
ssl_key_password	string	SSL 鍵ファイルのパスワード。 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
status	integer	アイテムプロトタイプのステータス。 指定可能な値: 0 - *(デフォルト)* 有効なアイテムプロトタイプ; 1 - 無効なアイテムプロトタイプ; 3 - サポートされていないアイテムプロトタイプ。
status_codes	string	必要な HTTP ステータスコードの範囲。カンマ区切りで指定します。 また、カンマ区切りリストの一部としてユーザーマクロまたは LLD マクロもサポートします。 例: 200,200-{$M},{$M},200-400 プロパティの動作: - type が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
templateid	ID	親テンプレートのアイテムプロトタイプの ID。 プロパティの動作: - 読み取り専用
timeout	string	アイテムデータ取得リクエストのタイムアウト。 秒、またはサフィックス付きの時間単位（例: 30s、1m）を受け付けます。 ユーザーマクロおよび LLD マクロも受け付けます。 指定可能な値の範囲: 1-600s。 デフォルト: ”” - プロキシ/グローバル設定を使用。 プロパティの動作: - type が”Zabbix agent” (0)、”Simple check” (3) で key_ が”vmware.” および”icmpping” で始まらない場合、”Zabbix agent (active)” (7)、”External check” (10)、”Database monitor” (11)、”SSH agent” (13)、”TELNET agent” (14)、”HTTP agent” (19)、”SNMP agent” (20) で snmp_oid が”walk[” または”get[” で始まる場合、”Script” (21)、”Browser” (22) に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
trapper_hosts	string	許可されたホスト。 プロパティの動作: - type が”Zabbix trapper” に設定されている場合、または type が”HTTP agent” に設定され、かつ allow_traps が”Allow to accept incoming data” に設定されている場合に サポート

プロパティ	型	説明
trends	string	トレンドデータを保存する期間を表す時間単位。 ユーザーマクロおよび LLD マクロも受け付けます。 デフォルト: 365d。 プロパティの動作: - `value_type` が”numeric float” または”numeric unsigned” に設定されている場合に サポート
units	string	値の単位。 プロパティの動作: - `value_type` が”numeric float” または”numeric unsigned” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
username	string	認証用ユーザー名。 プロパティの動作: - `type` が”SSH agent” または”TELNET agent” に設定されている場合、または `type` が”JMX agent” に設定され、かつ `password` が設定されている場合に 必須 - `type` が”Simple check”、”Database monitor”、または”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用（`type` が”HTTP agent” に設定されている場合）
uuid	string	既存のアイテムプロトタイプにインポートしたアイテムプロトタイプを関連付けるために使用される汎用一意識別子。指定しない場合は自動生成されます。 プロパティの動作: - アイテムプロトタイプがテンプレートに属する場合に サポート
valuemapid	ID	関連付けられた値マップの ID。 プロパティの動作: - `value_type` が”numeric float”、”character”、または”numeric unsigned” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
verify_host	integer	接続先のホスト名がホストの証明書内の名前と一致することを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
verify_peer	integer	ホストの証明書が正当であることを検証するかどうか。 指定可能な値: 0 - *(デフォルト)* 検証しない; 1 - 検証する。 プロパティの動作: - `type` が”HTTP agent” に設定されている場合に サポート - 継承されたオブジェクトでは 読み取り専用
discover	integer	アイテムプロトタイプのディスカバリステータス。 指定可能な値: 0 - *(デフォルト)* 新しいアイテムがディスカバリされる; 1 - 新しいアイテムはディスカバリされず、既存のアイテムは失われたものとしてマークされる。

HTTP ヘッダー

ヘッダーオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
name	string	HTTP ヘッダー名。 プロパティの動作: - 必須
value	string	ヘッダー値。 プロパティの動作: - 必須

HTTP クエリフィールド

クエリフィールドオブジェクトは、URL パラメータを指定するために使用される名前と値を定義します。次のプロパティがあります。

プロパティ	Type	説明
name	string	パラメータの名前。 Property behavior: *- required*
value	string	パラメータの値。 Property behavior: *- required*

アイテムプロトタイプタグ

アイテムプロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	アイテムプロトタイプのタグ名。 Property behavior: *- required*
value	string	アイテムプロトタイプのタグ値。

アイテムプロトタイプの前処理

アイテムプロトタイプの前処理オブジェクトには、以下のプロパティがあります。

Property	Type	Description
type	integer	前処理オプションのタイプ。 指定可能な値: 1 - カスタム乗数; 2 - 右トリム; 3 - 左トリム; 4 - トリム; 5 - 正規表現; 6 - 真偽値を 10 進数に変換; 7 - 8 進数を 10 進数に変換; 8 - 16 進数を 10 進数に変換; 9 - 単純変化; 10 - 1 秒あたりの変化量; 11 - XML XPath; 12 - JSONPath; 13 - 範囲内; 14 - 正規表現に一致; 15 - 正規表現に一致しない; 16 - JSON 内のエラーチェック; 17 - XML 内のエラーチェック; 18 - 正規表現を使用したエラーチェック; 19 - 未変更なら破棄; 20 - ハートビート付きで未変更なら破棄; 21 - JavaScript; 22 - Prometheus パターン; 23 - Prometheus から JSON; 24 - CSV から JSON; 25 - 置換; 26 - 未サポートをチェック; 27 - XML から JSON; 28 - SNMP walk 値; 29 - SNMP walk から JSON; 30 - SNMP get 値。 Property behavior: - *required*
params	string	前処理オプションで使用される追加パラメータ。 複数のパラメータは改行文字 (\n) で区切ります。 `type` が”Check unsupported” に設定されている場合、パラメータは `<scope>[\n<pattern>]` 構文に従います。ここで、*pattern* は正規表現、*scope* は次のいずれかです: -1 - 任意のエラーに一致; 0 - エラーメッセージが *pattern* に一致するか確認; 1 - エラーメッセージが *pattern* に一致しないか確認。 Property behavior: - `type` が”Custom multiplier” (1), ”Right trim” (2), ”Left trim” (3), ”Trim” (4), ”Regular expression” (5), ”XML XPath” (11), ”JSONPath” (12), ”In range” (13), ”Matches regular expression” (14), ”Does not match regular expression” (15), ”Check for error in JSON” (16), ”Check for error in XML” (17), ”Check for error using regular expression” (18), ”Discard unchanged with heartbeat” (20), ”JavaScript” (21), ”Prometheus pattern” (22), ”Prometheus to JSON” (23), ”CSV to JSON” (24), ”Replace” (25), ”Check unsupported” (26), ”SNMP walk value” (28), ”SNMP walk to JSON” (29), または”SNMP get value” (30) に設定されている場合は *required*

Property	Type	Description
error_handler	integer	前処理ステップが失敗した場合に使用されるアクションタイプ。 指定可能な値: 0 - エラーメッセージは Zabbix server によって設定される; 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 type が”Check unsupported” に設定されている場合の指定可能な値: 1 - 値を破棄; 2 - カスタム値を設定; 3 - カスタムエラーメッセージを設定。 Property behavior: - type が”Custom multiplier” (1), ”Regular expression” (5), ”Boolean to decimal” (6), ”Octal to decimal” (7), ”Hexadecimal to decimal” (8), ”Simple change” (9), ”Change per second” (10), ”XML XPath” (11), ”JSONPath” (12), ”In range” (13), ”Matches regular expression” (14), ”Does not match regular expression” (15), ”Check for error in JSON” (16), ”Check for error in XML” (17), ”Check for error using regular expression” (18), ”Prometheus pattern” (22), ”Prometheus to JSON” (23), ”CSV to JSON” (24), ”Check unsupported” (26), ”XML to JSON” (27), ”SNMP walk value” (28), ”SNMP walk to JSON” (29), または”SNMP get value” (30) に設定されている場合は *required*
error_handler_params	string	エラーハンドラーパラメータ。 Property behavior: - error_handler が”Set custom value” または”Set custom error message” に設定されている場合は *required*

各前処理タイプでサポートされるパラメータとエラーハンドラーは以下のとおりです。

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
1	Custom multiplier	数値[1, 6]			0, 1, 2, 3
2	Right trim	文字リスト[2]			
3	Left trim	文字リスト[2]			
4	Trim	文字リスト[2]			
5	Regular expression	パターン[3]	出力[2]		0, 1, 2, 3
6	Boolean to decimal				0, 1, 2, 3
7	Octal to decimal				0, 1, 2, 3
8	Hexadecimal to decimal				0, 1, 2, 3
9	Simple change				0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
10	Change per second				0, 1, 2, 3
11	XML XPath	パス[4]			0, 1, 2, 3
12	JSONPath	パス[4]			0, 1, 2, 3
13	In range	最小値[1, 6]	最大値[1, 6]		0, 1, 2, 3
14	Matches regular expression	パターン[3]			0, 1, 2, 3
15	Does not match regular expression	パターン[3]			0, 1, 2, 3
16	Check for error in JSON	パス[4]			0, 1, 2, 3
17	Check for error in XML	パス[4]			0, 1, 2, 3
18	Check for error using regular expression	パターン[3]	出力[2]		0, 1, 2, 3
19	Discard unchanged				
20	Discard unchanged with heartbeat	秒[5, 6]			
21	JavaScript	スクリプト[2]			
22	Prometheus pattern	パターン[6, 7]	`value`, `label`, `function`	出力[8, 9]	0, 1, 2, 3
23	Prometheus to JSON	パターン[6, 7]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
24	CSV to JSON	文字[2]	文字[2]	0,1	0, 1, 2, 3
25	Replace	検索文字列[2]	置換文字列[2]		
26	Check un-sup-ported	スコープ[1]	パターン[3, 6]		1, 2, 3
27	XML to JSON				0, 1, 2, 3
28	SNMP walk value	OID[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数		0, 1, 2, 3
29	SNMP walk to JSON[10]	フィールド名[2]	OID プレフィックス[2]	形式: 0 - 変更なし 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数	0, 1, 2, 3
30	SNMP get value	形式: 1 - Hex-STRING から UTF-8 2 - Hex-STRING から MAC 3 - BITS から整数			0, 1, 2, 3

[1] 整数または浮動小数点数
[2] 文字列
[3] 正規表現
[4] JSONPath または XML XPath
[5] 正の整数 (時間サフィックスをサポート。例: 30s, 1m, 2h, 1d)
[6] ユーザーマクロ、LLD マクロ
[7] 次の構文に従う Prometheus パターン: `<metric name>{<label name>="<label value>", ...} == <value>`。各 Prometheus パターンの構成要素 (メトリクス、ラベル名、ラベル値、メトリクス値) には、ユーザーマクロまたは LLD マクロを使用できます。
[8] 次の構文に従う Prometheus 出力: 2 番目のパラメータとして `label` が選択されている場合は `<label name>` (ユーザーマクロまたは LLD マクロを使用可能)。
[9] 2 番目のパラメータとして `function` が選択されている場合の集計関数のいずれか: `sum`, `min`, `max`, `avg`, `count`。
[10] 改行文字で区切られた複数の”Field name,OID prefix,Format records” レコードをサポートします。

itemprototype.create

説明

`object itemprototype.create(object/array itemPrototypes)`

このメソッドでは、新しいアイテムのプロトタイプを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 作成するアイテムプロトタイプ。

標準のアイテムプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
ruleid	ID	このアイテムが属するLLD ルールの ID。 パラメータの動作: - 必須
preprocessing	array	アイテムプロトタイプの前処理オプション。
tags	array	アイテムプロトタイプのタグ。

戻り値

(object) itemids プロパティの下で作成されたアイテムプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアイテムプロトタイプの順序と一致します。

例

アイテムのプロトタイプの作成

検出されたファイルシステム上の空きディスク容量を監視するためのアイテムのプロトタイプを作成します。検出されたアイテムは、30 秒ごとに更新される数値型の Zabbix エージェントアイテムである必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "{#FSNAME} の空きディスク容量",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

前処理を使用したアイテムプロトタイプの作成

1 秒あたりの変化量とカスタム乗数を 2 番目のステップとして使用するアイテムを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Incoming network traffic on {#IFNAME}",
        "key_": "net.if.in[{#IFNAME}]",
        "hostid": "10001",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "delay": "60s",
        "units": "bps",
        "interfaceid": "1155",
        "preprocessing": [
            {
                "type": 10,
                "params": "",
                "error_handler": 0,
                "error_handler_params": ""
            },
            {
                "type": 1,
                "params": "8",
                "error_handler": 2,
                "error_handler_params": "10"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

依存アイテムのプロトタイプの作成

ID 44211 のマスターアイテムのプロトタイプに対して、依存アイテムのプロトタイプを作成します。
同じホスト（テンプレート/ディスカバリルール）上の依存関係のみ許可されるため、
マスターアイテムと依存アイテムは同じ hostid と ruleid を持っている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Dependent test item prototype",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムのプロトタイプを作成

ユーザーマクロ、クエリフィールド、カスタムヘッダーを使用した URL でアイテムのプロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "type": "19",
        "hostid": "10254",
        "ruleid": "28256",
        "interfaceid": "2",
        "name": "api item prototype example",
        "key_": "api_http_item",
        "value_type": 3,
        "url": "{$URL_PROTOTYPE}",
        "query_fields": [
            {
                "name": "min",
                "value": "10"
            },
            {
                "name": "max",
                "value" "100"
            }
        ],
        "headers": [
            {
                "name": "X-Source",
                "value": "api"
            }
        ],
        "delay": "35"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

スクリプトアイテムのプロトタイプを作成

スクリプトアイテムのプロトタイプを使用して、シンプルなデータ収集を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Script example",
        "key_": "custom.script.itemprototype",
        "hostid": "12345",
        "type": 21,
        "value_type": 4,
        "params": "var request = new HttpRequest();\nreturn request.post(\"https://postman-echo.com/post\"
        "parameters": [
            {
                "name": "host",
                "value": "{HOST.CONN}"
            }
        ],
        "timeout": "6s",
        "delay": "30s"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::create() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.delete

説明

`object itemprototype.delete(array itemPrototypeIds)`

このメソッドは、アイテムのプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアイテムのプロトタイプの ID。

戻り値

(object) prototypeids プロパティの下にある削除されたアイテムのプロトタイプの ID を含むオブジェクトを返します。

例

複数のアイテムのプロトタイプの削除

2 つのアイテムのプロトタイプを削除します。
マスターアイテムまたはアイテムのプロトタイプが削除されると、依存アイテムのプロトタイプは自動的に削除されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::delete() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.get

説明

`integer/array itemprototype.get(object parameters)`

このメソッドは、指定されたパラメータに従ってアイテムのプロトタイプを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
discoveryids	ID/array	指定された LLD ルールに属するアイテムプロトタイプのみを返します。
graphids	ID/array	指定されたグラフプロトタイプで使用されているアイテムプロトタイプのみを返します。
hostids	ID/array	指定されたホストに属するアイテムプロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたアイテムプロトタイプのみを返します。
itemids	ID/array	指定された ID を持つアイテムプロトタイプのみを返します。
monitored	boolean	`true` に設定すると、監視対象ホストに属する有効なアイテムプロトタイプのみを返します。
templated	boolean	`true` に設定すると、テンプレートに属するアイテムプロトタイプのみを返します。
templateids	ID/array	指定されたテンプレートに属するアイテムプロトタイプのみを返します。
triggerids	ID/array	指定されたトリガープロトタイプで使用されているアイテムプロトタイプのみを返します。

Parameter	Type	Description
selectDiscoveryData	query	アイテムプロトタイプのディスカバリオブジェクトデータを含む`discoveryData`プロパティを返します。アイテムプロトタイプのディスカバリオブジェクトは、検出されたアイテムプロトタイプを、それが検出された元のアイテムプロトタイプに関連付けます。 以下のプロパティがあります: `parent_itemid` - (`string`) アイテムプロトタイプの作成元となったアイテムプロトタイプの ID; `key_` - (`string`) アイテムプロトタイプのキー; `status` - (`int`) アイテムプロトタイプのディスカバリステータス: 0 - *(default)* アイテムプロトタイプは検出されている; 1 - アイテムプロトタイプはすでに検出されていない; `ts_delete` - (`timestamp`) すでに検出されていないアイテムプロトタイプが削除される時刻; `ts_disable` - (`timestamp`) すでに検出されていないアイテムプロトタイプが無効化される時刻; `disable_source` - (`int`) アイテムプロトタイプが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(default)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	アイテムプロトタイプが属するローレベルディスカバリルールを含む`discoveryRule`プロパティを返します。
selectDiscoveryRulePrototype	query	アイテムプロトタイプが属する親 LLD ルールプロトタイプを含む`discoveryRulePrototype`プロパティを返します。
selectGraphs	query	アイテムプロトタイプが使用されているグラフプロトタイプを含む`graphs`プロパティを返します。 `count` をサポートします。
selectHosts	query	アイテムプロトタイプが属するホストの配列を含む`hosts`プロパティを返します。
selectTags	query	アイテムプロトタイプのタグを`tags`プロパティで返します。
selectTriggers	query	アイテムプロトタイプが使用されているトリガープロトタイプを含む`triggers`プロパティを返します。 `count` をサポートします。
selectPreprocessing	query	アイテムプロトタイプの前処理オプションを含む`preprocessing`プロパティを返します。
selectValueMap	query	アイテムプロトタイプの値マップを含む`valuemap`プロパティを返します。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が照合対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` data type のプロパティはサポートしません。 追加のプロパティをサポートします: `host` - アイテムプロトタイプが属するホストの技術名。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectGraphs` - 結果は `name` でソートされます; `selectTriggers` - 結果は `description` でソートされます。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	

Parameter	Type	Description
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

LLD ルールからアイテムのプロトタイプを取得する

特定の LLD ルール ID に対するすべてのアイテムのプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23077",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10079",
            "name": "Incoming network traffic on en0",
            "key_": "net.if.in[en0]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "bps",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "",
```

```
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        },
        {
            "itemid": "10010",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10001",
            "name": "Processor load (1 min average per core)",
            "key_": "system.cpu.load[percpu,avg1]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "The processor load is calculated as system CPU load divided by number of CPU c
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
```

```
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        }
    ],
    "id": 1
}
```

依存アイテムの検索

特定のアイテム ID に対する依存アイテムを 1 つ検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "25547",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10116",
            "name": "Seconds",
            "key_": "apache.status.uptime.seconds",
            "delay": "0",
            "history": "90d",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
```

```
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "0",
            "interfaceid": "0",
            "description": "",
            "evaltype": "0",
            "master_itemid": "25545",
            "jmx_endpoint": "",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


HTTP エージェントのアイテムプロトタイプを検索

特定のホスト ID に対して、リクエストメソッドが HEAD の HTTP エージェントのアイテムプロトタイプを検索します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "28257",
            "type": "19",
            "snmp_oid": "",
            "hostid": "10254",
            "name": "discovered",
```

```
            "key_": "item[{#INAME}]",
            "delay": "{#IUPDATE}",
            "history": "90d",
            "trends": "30d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "",
            "logtimefmt": "",
            "templateid": "28255",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "2",
            "description": "",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "{#IURL}",
            "query_fields": [],
            "posts": "",
            "status_codes": "",
            "follow_redirects": "0",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "3",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "parameters": []
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- グラフプロトタイプ
- トリガープロトタイプ

ソース

CItemPrototype::get() in *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

説明

`object itemprototype.update(object/array itemPrototypes)`

このメソッドでは、既存のアイテムのプロトタイプを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
> このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object/array) 更新するアイテムプロトタイプのプロパティ。

各アイテムプロトタイプでは itemid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のアイテムプロトタイププロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
preprocessing	array	現在の前処理オプションを置き換えるためのアイテムプロトタイプの前処理オプション。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
tags	array	アイテムプロトタイプのタグ。

戻り値

(object) itemids プロパティの下で更新されたアイテムプロトタイプの ID を含むオブジェクトを返します。

例

アイテムプロトタイプのインターフェースの変更

ディスカバリされたアイテムで使用されるホストインターフェースを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

依存アイテムのプロトタイプを更新

依存アイテムのプロトタイプを、新しいマスターアイテムのプロトタイプ ID で更新します。許可されるのは同じホスト（テンプレート/ディスカバリルール）上の依存関係のみであるため、マスターアイテムと依存アイテムは同じ hostid および ruleid を持っている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

HTTP エージェントアイテムのプロトタイプを更新

クエリフィールドを変更し、すべてのカスタムヘッダーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
        "headers": []
    }
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

アイテムの前処理オプションの更新

アイテムの前処理ルール「カスタム乗数」を持つアイテムのプロトタイプを更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
```

```
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

スクリプトアイテムのプロトタイプの更新

スクリプトアイテムのプロトタイプを別のスクリプトで更新し、以前のスクリプトで使用されていた不要なパラメータを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

ソース

CItemPrototype::update() in *ui/include/classes/api/services/CItemPrototype.php*.

アクション

このクラスは、アクションを操作するために設計されています。

オブジェクト参照:

- アクション
- アクションの実行内容
    - アクション実行メッセージ
    - アクション実行条件
- アクションのリカバリ実行内容

- アクションの更新時実行内容
- アクションフィルター
  - アクションフィルター条件

利用可能なメソッド:

- action.create - 新しいアクションを作成
- action.delete - アクションを削除
- action.get - アクションを取得
- action.update - アクションを更新

Action オブジェクト

次のオブジェクトは、`action`API に直接関連しています。

アクション

アクションオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
actionid	ID	アクションの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
esc_period	string	デフォルトの実行ステップ期間。少なくとも 60 秒以上である必要があります。秒数、接尾辞付きの時間単位、またはユーザーマクロを指定できます。 プロパティの動作: - `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合に サポート
eventsource	integer	アクションが処理するイベントの種類。 サポートされているイベントタイプの一覧については、`event source` プロパティを参照してください。 プロパティの動作: - 定数 - 作成操作で 必須
name	string	アクションの名前。 プロパティの動作: - 作成操作で 必須
status	integer	アクションが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
pause_symptoms	integer	イベントが症状イベントである場合にエスカレーションを一時停止するかどうか。 指定可能な値: 0 - 症状の問題ではエスカレーションを一時停止しない; 1 - *(デフォルト)* 症状の問題ではエスカレーションを一時停止する。 プロパティの動作: - `eventsource` が「トリガーによって作成されたイベント」に設定されている場合に サポート

プロパティ	型	説明
pause_suppressed	integer	メンテナンス期間中にエスカレーションを一時停止するかどうか。 指定可能な値: 0 - エスカレーションを一時停止しない; 1 - *(デフォルト)* エスカレーションを一時停止する。 プロパティの動作: - `eventsource` が「トリガーによって作成されたイベント」に設定されている場合に サポート
notify_if_canceled	integer	エスカレーションがキャンセルされたときに通知するかどうか。 指定可能な値: 0 - エスカレーションがキャンセルされても通知しない; 1 - *(デフォルト)* エスカレーションがキャンセルされたときに通知する。 プロパティの動作: - `eventsource` が「トリガーによって作成されたイベント」に設定されている場合に サポート

アクションの実行内容

アクション実行内容オブジェクトは、アクションが実行されたときに実施される操作を定義します。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
operationtype	integer	操作の種類。 指定可能な値: 0 - メッセージ送信; 1 - グローバルスクリプト; 2 - ホストの追加; 3 - ホストの削除; 4 - ホストグループへの追加; 5 - ホストグループからの削除; 6 - テンプレートのリンク; 7 - テンプレートのリンク解除; 8 - ホストの有効化; 9 - ホストの無効化; 10 - ホストインベントリモードの設定; 13 - ホストタグの追加; 14 - ホストタグの削除。 Action object の `eventsource` が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合の指定可能な値: 0 - 「メッセージ送信」; 1 - 「グローバルスクリプト」。 Action object の `eventsource` が「内部イベント」に設定されている場合の指定可能な値: 0 - 「メッセージ送信」。 Property behavior: - *required*

Property	Type	Description
esc_period	string	エスカレーションステップの期間（秒）。60 秒より大きい必要があります。秒数、サフィックス付きの時間単位、またはユーザーマクロを指定できます。0 または 0s に設定した場合、デフォルトのアクションエスカレーション期間が使用されます。 デフォルト: 0s。 Property behavior: - *supported* if Action object の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合
esc_step_from	integer	エスカレーションを開始するステップ。 デフォルト: 1。 Property behavior: - *supported* if Action object の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合
esc_step_to	integer	エスカレーションを終了するステップ。 デフォルト: 1。 Property behavior: - *supported* if Action object の `eventsource` が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合
evaltype	integer	実行条件の評価方法。 指定可能な値: 0 - *(default)* And/Or; 1 - And; 2 - Or。
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトでは `scriptid` プロパティが定義されている必要があります。 Property behavior: - *required* if `operationtype` が「グローバルスクリプト」に設定されている場合
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループでは `groupid` プロパティが定義されている必要があります。 Property behavior: - *required* if `operationtype` が「グローバルスクリプト」に設定され、`opcommand_hst` が設定されていない場合
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストでは `hostid` プロパティが定義されている必要があります。 Property behavior: - *required* if `operationtype` が「グローバルスクリプト」に設定され、`opcommand_grp` が設定されていない場合
opconditions	array	トリガーアクションで使用される実行条件。 実行条件オブジェクトについては、以下で詳しく説明しています。

Property	Type	Description
opgroup	array	ホストを追加するホストグループ。 ホストグループでは groupid プロパティが定義されている必要があります。 Property behavior: - *required* if operationtype が「ホストグループへの追加」または「ホストグループからの削除」に設定されている場合
opmessage	object	実行内容によって送信されるメッセージに関するデータを含むオブジェクト。 実行メッセージオブジェクトについては、以下で詳しく説明しています。 Property behavior: - *required* if operationtype が「メッセージ送信」に設定されている場合
opmessage_grp	array	メッセージの送信先ユーザーグループ。 ユーザーグループでは usrgrpid プロパティが定義されている必要があります。 Property behavior: - *required* if operationtype が「メッセージ送信」に設定され、opmessage_usr が設定されていない場合
opmessage_usr	array	メッセージの送信先ユーザー。 ユーザーでは userid プロパティが定義されている必要があります。 Property behavior: - *required* if operationtype が「メッセージ送信」に設定され、opmessage_grp が設定されていない場合
optemplate	array	ホストにリンクするテンプレート。 テンプレートでは templateid プロパティが定義されている必要があります。 Property behavior: - *required* if operationtype が「テンプレートのリンク」または「テンプレートのリンク解除」に設定されている場合
opinventory	object	ホストに設定するインベントリモード。 インベントリでは inventory_mode プロパティが定義されている必要があります。 Property behavior: - *required* if operationtype が「ホストインベントリモードの設定」に設定されている場合
optag	array	追加または削除するホストタグ。 タグでは tag プロパティが定義されている必要があります。 value プロパティは任意です。 Property behavior: - *supported* if operationtype が「ホストタグの追加」または「ホストタグの削除」に設定されている場合

アクション実行メッセージ

実行メッセージオブジェクトには、実行によって送信されるメッセージに関するデータが含まれます。次のプロパティがあります。

プロパティ	型	説明
default_msg	integer	デフォルトのアクションメッセージ本文および件名を使用するかどうか。 指定可能な値: 0 - 実行のデータを使用; 1 - *(デフォルト)* メディアタイプのデータを使用。
mediatypeid	ID	メッセージの送信に使用されるメディアタイプの ID。 プロパティの動作: - アクション実行オブジェクト、アクション復旧実行オブジェクト、またはアクション更新実行オブジェクト の `operationtype` が「メッセージ送信」に設定されている場合、またはアクション更新実行オブジェクト の `operationtype` が「関係者全員に通知」に設定されている場合に *supported*
message	string	実行メッセージの本文。 プロパティの動作: - `default_msg` が「実行のデータを使用」に設定されている場合に *supported*
subject	string	実行メッセージの件名。 プロパティの動作: - `default_msg` が「実行のデータを使用」に設定されている場合に *supported*

アクション実行条件

アクション実行条件オブジェクトは、現在の実行を行うために満たす必要がある条件を定義します。以下のプロパティがあります。

プロパティ	Type	説明
conditiontype	integer	条件のタイプ。 指定可能な値: 14 - イベント確認済み。 プロパティの動作: - 必須
value	string	比較する値。 プロパティの動作: - 必須
operator	integer	条件演算子。 指定可能な値: 0 - *(デフォルト)* =

以下の演算子と値が、各実行条件タイプでサポートされています。

条件	条件名	サポートされる演算子	期待される値
14	イベント確認済み	=	イベントが確認済みかどうか。 指定可能な値: 0 - 未確認; 1 - 確認済み。

アクションの復旧時実行内容

アクションの復旧時実行内容オブジェクトは、障害が解決されたときに実行される操作を定義します。
復旧時実行内容が可能なのは、トリガー、内部、およびサービスアクションのみです。
以下のプロパティがあります。

Property	Type	Description
operationtype	integer	操作の種類。 アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合の使用可能な値: 0 - メッセージ送信; 1 - グローバルスクリプト; 11 - 関係者全員に通知。 アクションオブジェクト の eventsource が「内部イベント」に設定されている場合の使用可能な値: 0 - メッセージ送信; 11 - 関係者全員に通知。 プロパティの動作: - 必須
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトでは scriptid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定されている場合は 必須
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループでは groupid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、かつ opcommand_hst が設定されていない場合は 必須
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストでは hostid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、かつ opcommand_grp が設定されていない場合は 必須
opmessage	object	復旧時実行内容によって送信されるメッセージに関するデータを含むオブジェクト。 実行内容メッセージオブジェクトの詳細は、上記の説明を参照してください。 プロパティの動作: - operationtype が「メッセージ送信」に設定されている場合は 必須
opmessage_grp	array	メッセージの送信先となるユーザーグループ。 ユーザーグループでは usrgrpid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定され、かつ opmessage_usr が設定されていない場合は 必須

Property	Type	Description
opmessage_usr	array	メッセージの送信先となるユーザー。 ユーザーでは userid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定され、かつ opmessage_grp が設定されていない場合は 必須

アクション更新操作

アクション更新操作オブジェクトは、障害が更新されたとき（コメントの追加、確認、深刻度の変更、または手動クローズ）に実行される操作を定義します。更新操作は、トリガーアクションおよびサービスアクションでのみ使用できます。このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
operationtype	integer	操作の種類。 指定可能な値: 0 - メッセージ送信; 1 - グローバルスクリプト; 12 - 関係者全員に通知。 プロパティの動作: - 必須
opcommand	object	実行するグローバルスクリプト。 グローバルスクリプトでは scriptid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「グローバルスクリプト」に設定されている場合は 必須
opcommand_grp	array	グローバルスクリプトを実行するホストグループ。 ホストグループでは groupid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、かつ opcommand_hst が設定されていない場合は 必須
opcommand_hst	array	グローバルスクリプトを実行するホスト。 ホストでは hostid プロパティを定義する必要があります。 プロパティの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定され、operationtype が「グローバルスクリプト」に設定され、かつ opcommand_grp が設定されていない場合は 必須
opmessage	object	更新操作によって送信されるメッセージに関するデータを含むオブジェクト。 操作メッセージオブジェクトについては、上記で詳しく説明しています。
opmessage_grp	array	メッセージの送信先となるユーザーグループ。 ユーザーグループでは usrgrpid プロパティを定義する必要があります。 プロパティの動作: - operationtype が「メッセージ送信」に設定され、opmessage_usr が設定されていない場合は 必須

Property	Type	Description
opmessage_usr	array	メッセージの送信先となるユーザー。 ユーザーでは `userid` プロパティを定義する必要があります。 プロパティの動作: - `operationtype` が「メッセージ送信」に設定され、`opmessage_grp` が設定されていない場合は 必須

アクションフィルター

アクションフィルターオブジェクトは、設定されたアクションの実行操作を実行するために満たす必要がある条件のセットを定義します。以下のプロパティがあります。

プロパティ	型	説明
conditions	array	結果をフィルタリングするために使用するフィルター条件のセット。条件は、formula 内で配置された順序でソートされます。 プロパティの動作: - 必須
evaltype	integer	フィルター条件の評価方法。 使用可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 プロパティの動作: - 必須
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、`formulaid` によって特定のフィルター条件を参照する ID が含まれます。カスタム式を持つフィルターでは、`eval_formula` の値は `formula` の値と同じです。 プロパティの動作: - 読み取り専用
formula	string	カスタム式を持つフィルターの条件評価に使用する、ユーザー定義の式。この式には、`formulaid` によって特定のフィルター条件を参照する ID を含める必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致している必要があります。どの条件も未使用のままにしたり、省略したりすることはできません。 プロパティの動作: - `evaltype` が「カスタム式」に設定されている場合は 必須

アクションフィルター条件

アクションフィルター条件オブジェクトは、アクションの実行操作を開始する前に確認しなければならない特定の条件を定義します。

プロパティ	Type	説明
conditiontype	integer	条件のタイプ。 アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」に設定されている場合の値: 0 - ホストグループ; 1 - ホスト; 2 - トリガー; 3 - イベント名; 4 - トリガーの深刻度; 6 - 時間帯; 13 - ホストテンプレート; 16 - 障害は抑制されている; 25 - イベントタグ; 26 - イベントタグ値。 アクションオブジェクト の eventsource が「ディスカバリルールによって作成されたイベント」に設定されている場合の値: 7 - ホスト IP; 8 - 検出されたサービスのタイプ; 9 - 検出されたサービスのポート; 10 - ディスカバリステータス; 11 - 稼働時間または停止時間の期間; 12 - 受信値; 18 - ディスカバリルール; 19 - ディスカバリチェック; 20 - プロキシ; 21 - ディスカバリオブジェクト。 アクションオブジェクト の eventsource が「アクティブエージェントの自動登録によって作成されたイベント」に設定されている場合の値: 20 - プロキシ; 22 - ホスト名; 24 - ホストメタデータ。 アクションオブジェクト の eventsource が「内部イベント」に設定されている場合の値: 0 - ホストグループ; 1 - ホスト; 13 - ホストテンプレート; 23 - イベントタイプ; 25 - イベントタグ; 26 - イベントタグ値。 アクションオブジェクト の eventsource が「サービスステータス更新時に作成されたイベント」に設定されている場合の値: 25 - イベントタグ; 26 - イベントタグ値; 27 - サービス; 28 - サービス名。 Property behavior: - *required*
value	string	比較する値。 Property behavior: - *required*

プロパティ	Type	説明
value2	string	比較する第 2 の値。 Property behavior: - アクションオブジェクト の `eventsource` が「トリガーによって作成されたイベント」に設定され、`conditiontype` がトリガーアクションで使用可能な任意の値に設定され、かつ条件タイプ（下記参照）が「26」の場合は *required* - アクションオブジェクト の `eventsource` が「内部イベント」に設定され、`conditiontype` が内部アクションで使用可能な任意の値に設定され、かつ条件タイプ（下記参照）が「26」の場合は *required* - アクションオブジェクト の `eventsource` が「サービスステータス更新時に作成されたイベント」に設定され、`conditiontype` がサービスアクションで使用可能な任意の値に設定され、かつ条件タイプ（下記参照）が「26」の場合は *required*
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。この ID はフィルター条件を変更する際にユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。
operator	integer	条件の演算子。 使用可能な値: 0 - *(default)* 等しい; 1 - 等しくない; 2 - 含む; 3 - 含まない; 4 - in; 5 - 以上; 6 - 以下; 7 - not in; 8 - 一致する; 9 - 一致しない; 10 - はい; 11 - いいえ。

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、action.get およびaction.create メソッドのページにある例を参照してください。

以下の演算子と値が、各条件タイプでサポートされています。

条件	条件名	サポートされる演算子	想定される値
0	ホストグループ	等しい, 等しくない	ホストグループ ID。
1	ホスト	等しい, 等しくない	ホスト ID。
2	トリガー	等しい, 等しくない	トリガー ID。
3	イベント名	含む, 含まない	イベント名。
4	トリガーの深刻度	等しい, 等しくない, 以上, 以下	トリガーの深刻度。サポートされるトリガー深刻度の一覧については、trigger `severity` property を参照してください。
5	トリガー値	等しい	トリガー値。サポートされるトリガー値の一覧については、trigger `value` property を参照してください。
6	時間帯	in, not in	イベントがトリガーされた時刻をtime period として指定します。

条件	条件名	サポートされる演算子	想定される値
7	ホスト IP	等しい, 等しくない	確認する 1 つまたは複数の IP 範囲をカンマ区切りで指定します。サポートされる IP 範囲の形式の詳細については、ネットワークディスカバリ設定 セクションを参照してください。
8	検出されたサービスのタイプ	等しい, 等しくない	検出されたサービスのタイプ。サービスのタイプは、そのサービスの検出に使用されたディスカバリチェックのタイプと一致します。サポートされるタイプの一覧については、discovery check type property を参照してください。
9	検出されたサービスのポート	等しい, 等しくない	1 つまたは複数のポート範囲をカンマ区切りで指定します。
10	ディスカバリステータス	等しい	検出されたオブジェクトのステータス。 使用可能な値: 0 - ホストまたはサービスが稼働中; 1 - ホストまたはサービスが停止中; 2 - ホストまたはサービスが検出された; 3 - ホストまたはサービスが失われた。
11	稼働時間または停止時間の期間	以上, 以下	検出されたオブジェクトが現在のステータスにある時間を秒単位で示します。
12	受信値	等しい, 等しくない, 以上, 以下, 含む, 含まない	Zabbix エージェント、SNMPv1、SNMPv2、または SNMPv3 のディスカバリチェック実行時に返される値。
13	ホストテンプレート	等しい, 等しくない	リンクされたテンプレート ID。
16	障害は抑制されている	はい, いいえ	値は不要です。「はい」演算子を使用すると障害が抑制されている必要があり、「いいえ」は抑制されていないことを意味します。
18	ディスカバリルール	等しい, 等しくない	ディスカバリルールの ID。
19	ディスカバリチェック	等しい, 等しくない	ディスカバリチェックの ID。
20	プロキシ	等しい, 等しくない	プロキシの ID。
21	ディスカバリオブジェクト	等しい	ディスカバリイベントをトリガーしたオブジェクトのタイプ。 使用可能な値: 1 - 検出されたホスト; 2 - 検出されたサービス。
22	ホスト名	含む, 含まない, 一致する, 一致しない	ホスト名。 自動登録条件では、*matches* および *does not match* 演算子で正規表現を使用できます。
23	イベントタイプ	等しい	特定の内部イベント。 使用可能な値: 0 - アイテムが「未サポート」状態; 1 - アイテムが「正常」状態; 2 - LLD ルールが「未サポート」状態; 3 - LLD ルールが「正常」状態; 4 - トリガーが「不明」状態; 5 - トリガーが「正常」状態。
24	ホストメタデータ	含む, 含まない, 一致する, 一致しない	自動登録されたホストのメタデータ。 *matches* および *does not match* 演算子では正規表現を使用できます。
25	タグ	等しい, 等しくない, 含む, 含まない	イベントタグ。

条件	条件名	サポートされる演算子	想定される値
26	タグ値	等しい, 等しくない, 含む, 含まない	イベントタグ値。
27	サービス	等しい, 等しくない	サービス ID。
28	サービス名	等しい, 等しくない	サービス名。

action.create

説明

`object action.create(object/array actions)`

このメソッドは、新しいアクションを作成することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 作成するアクション。

標準のアクションプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
filter	object	アクション用のアクションフィルターオブジェクト。
operations	array	アクション用に作成するアクション操作。
recovery_operations	array	アクション用に作成するアクション復旧操作。
update_operations	array	アクション用に作成するアクション更新操作。

戻り値

(`object`) `actionids` プロパティの下に作成されたアクションの ID を含むオブジェクトを返します。返される ID の順序は、渡されたアクションの順序と一致します。

例

トリガーアクションの作成

ホスト"10084" からのトリガー (名前に"memory" が含まれる) が障害状態になると実行されるトリガーアクションを作成する。アクションには 4 つの操作が構成されます。最初の即時操作では、メディアタイプ"1" を介してユーザーグループ"7" 内のすべてのユーザーにメッセージが送信されます。イベントが 30 分以内に解決されない場合、2 番目の操作ではグループ"2" のすべてのホストでスクリプト"5"(スコープ" アクション操作" のスクリプト) が実行されます。イベントが解決された場合は、回復操作により、問題に関するメッセージを受け取ったすべてのユーザーに通知されます。イベントが更新されると、確認/更新操作により、問題に関するメッセージを受信したすべてのユーザーに (カスタムの件名とメッセージを使用して) 通知します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "30m",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
```

```
                "conditiontype": 1,
                "operator": 0,
                "value": "10084"
            },
            {
                "conditiontype": 3,
                "operator": 2,
                "value": "memory"
            }
        ]
    },
    "operations": [
        {
            "operationtype": 0,
            "esc_step_from": 1,
            "esc_step_to": 1,
            "opmessage_grp": [
                {
                    "usrgrpid": "7"
                }
            ],
            "opmessage": {
                "default_msg": 1,
                "mediatypeid": "1"
            }
        },
        {
            "operationtype": 1,
            "esc_step_from": 2,
            "esc_step_to": 2,
            "opconditions": [
                {
                    "conditiontype": 14,
                    "operator": 0,
                    "value": "0"
                }
            ],
            "opcommand_grp": [
                {
                    "groupid": "2"
                }
            ],
            "opcommand": {
                "scriptid": "5"
            }
        }
    ],
    "recovery_operations": [
        {
            "operationtype": "11",
            "opmessage": {
                "default_msg": 1
            }
        }
    ],
    "update_operations": [
        {
            "operationtype": "12",
            "opmessage": {
                "default_msg": 0,
                "message": "Custom update operation message body",
                "subject": "Custom update operation message subject"
```

```
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

ディスカバリアクションの作成

検出されたホストにテンプレート”10001” をリンクするディスカバリアクションを作成します。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Discovery action",
        "eventsource": 1,
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "conditiontype": 21,
                    "operator": 0,
                    "value": "1"
                },
                {
                    "conditiontype": 10,
                    "operator": 0,
                    "value": "2"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 6,
                "optemplate": [
                    {
                        "templateid": "10001"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
```

```
            "18"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

アクション条件を評価するためのカスタム式”A and (B or C)” を使用するトリガーアクションを作成します。ホスト”10084” またはホスト”10106” からの” 警告” 以上の深刻度を持つトリガーが障害状態になると、このアクションは、メディアタイプ”1” を介してユーザーグループ”7” のすべてのユーザーにメッセージを送信します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Trigger action",
        "eventsource": 0,
        "esc_period": "15m",
        "filter": {
            "evaltype": 3,
            "formula": "A and (B or C)",
            "conditions": [
                {
                    "conditiontype": 4,
                    "operator": 5,
                    "value": "2",
                    "formulaid": "A"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10084",
                    "formulaid": "B"
                },
                {
                    "conditiontype": 1,
                    "operator": 0,
                    "value": "10106",
                    "formulaid": "C"
                }
            ]
        },
        "operations": [
            {
                "operationtype": 0,
                "esc_step_from": 1,
                "esc_step_to": 1,
                "opmessage_grp": [
                    {
                        "usrgrpid": "7"
                    }
                ],
                "opmessage": {
                    "default_msg": 1,
                    "mediatypeid": "1"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

エージェントの自動登録ルール作成

ホスト名に”SRV ” が含まれているか、メタデータに”AlmaLinux” が含まれている場合、ホストをホストグループ”2” に追加します。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers",
        "eventsource": "2",
        "filter": {
            "evaltype": "2",
            "conditions": [
                {
                    "conditiontype": "22",
                    "operator": "2",
                    "value": "SRV"
                },
                {
                    "conditiontype": "24",
                    "operator": "2",
                    "value": "AlmaLinux"
                }
            ]
        },
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

ホストタグとエージェント自動登録ルールの作成

ホストをホストグループ”2” に追加し、2 つのホストタグを追加する自動登録アクションを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers with tags",
        "eventsource": "2",
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            },
            {
                "operationtype": "13",
                "optag": [
                    {
                        "tag": "location",
                        "value": "office"
                    },
                    {
                        "tag": "city",
                        "value": "Riga"
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

関連項目

- アクションフィルター
- アクションの実行内容
- スクリプト

ソース

CAction::create() in *ui/include/classes/api/services/CAction.php*.

action.delete

説明

`object action.delete(array actionIds)`

このメソッドは、アクションを削除することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するアクションの ID。

戻り値

(object) actionids プロパティの下にある削除されたアクションの ID を含むオブジェクトを返します。

例

複数のアクションの削除

2 つのアクションを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
        ]
    },
    "id": 1
}
```

ソース

CAction::delete() in *ui/include/classes/api/services/CAction.php*.

action.get

説明

`integer/array action.get(object parameters)`

このメソッドは、指定したパラメーターに応じたアクションを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
actionids	ID/array	指定した ID のアクションのみを返します。
groupids	ID/array	アクションの条件で指定したホストグループを使用しているアクションのみを返します。

パラメータ	Type	説明
hostids	ID/array	アクションの条件で指定したホストを使用しているアクションのみを返します。
triggerids	ID/array	アクションの条件で指定したトリガーを使用しているアクションのみを返します。
mediatypeids	ID/array	メッセージ送信に指定したメディアタイプを使用しているアクションのみを返します。
usrgrpids	ID/array	指定したユーザーグループにメッセージを送信するよう設定されているアクションのみを返します。
userids	ID/array	指定したユーザーにメッセージを送信するよう設定されているアクションのみを返します。
scriptids	ID/array	指定したスクリプトを実行するよう設定されているアクションのみを返します。
selectFilter	query	アクション条件フィルターを含む`filter` プロパティを返します。
selectOperations	query	アクションの実行内容を含む`operations` プロパティを返します。
selectRecoveryOperations	query	アクションの復旧時の実行内容を含む`recovery_operations` プロパティを返します。
selectUpdateOperations	query	アクションの更新時の実行内容を含む`update_operations` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `actionid`, `name`, `status`。
countOutput	boolean	これらのパラメータについては、reference commentary を参照してください。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のどちらかを返します。

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合は、取得したオブジェクトの数

例

トリガーアクションの取得

設定されたすべてのトリガーアクションを、アクションの実行条件および実行内容と共に取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Report problems to Zabbix administrators",
            "eventsource": "0",
            "status": "1",
            "esc_period": "1h",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [],
                "eval_formula": ""
            },
            "operations": [
                {
                    "operationid": "3",
                    "actionid": "3",
                    "operationtype": "0",
                    "esc_period": "0",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": [
                        {
                            "default_msg": "1",
                            "subject": "",
                            "message": "",
                            "mediatypeid" => "0"
                        }
                    ],
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ]
                }
            ],
            "recovery_operations": [
                {
                    "operationid": "7",
                    "actionid": "3",
                    "operationtype": "11",
                    "evaltype": "0",
                    "opconditions": [],
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "{TRIGGER.STATUS}: {TRIGGER.NAME}",
                        "message": "Trigger: {TRIGGER.NAME}\r\nTrigger status: {TRIGGER.STATUS}\r\nTrigger
                        "mediatypeid": "0"
                    }
                }
            ],
            "update_operations": [
                {
                    "operationid": "31",
                    "operationtype": "12",
                    "evaltype": "0",
```

```
                    "opmessage": {
                        "default_msg": "1",
                        "subject": "",
                        "message": "",
                        "mediatypeid": "0"
                    }
                },
                {
                    "operationid": "32",
                    "operationtype": "0",
                    "evaltype": "0",
                    "opmessage": {
                        "default_msg": "0",
                        "subject": "Updated: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} updated problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.T
                        "mediatypeid": "1"
                    },
                    "opmessage_grp": [
                        {
                            "usrgrpid": "7"
                        }
                    ],
                    "opmessage_usr": []
                },
                {
                    "operationid": "33",
                    "operationtype": "1",
                    "evaltype": "0",
                    "opcommand": {
                        "scriptid": "3"
                    },
                    "opcommand_hst": [
                        {
                            "hostid": "10084"
                        }
                    ],
                    "opcommand_grp": []
                }
            ]
        }
    ],
    "id": 1
}
```


ディスカバリアクションの取得

設定されたすべてのディスカバリアクションを、アクションの条件と操作とともに取得します。このフィルターは"and" 評価タイプを使用するため、formula プロパティは空で、eval_formula が自動的に生成されます。

リクエスト：


```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "2",
            "name": "Auto discovery. Linux servers.",
            "eventsource": "1",
            "status": "1",
            "esc_period": "0s",
            "pause_suppressed": "1",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "conditiontype": "10",
                        "operator": "0",
                        "value": "0",
                        "value2": "",
                        "formulaid": "B"
                    },
                    {
                        "conditiontype": "8",
                        "operator": "0",
                        "value": "9",
                        "value2": "",
                        "formulaid": "C"
                    },
                    {
                        "conditiontype": "12",
                        "operator": "2",
                        "value": "Linux",
                        "value2": "",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A and B and C"
            },
            "operations": [
                {
                    "operationid": "1",
                    "actionid": "2",
                    "operationtype": "6",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
                    "evaltype": "0",
                    "opconditions": [],
                    "optemplate": [
                        {
                            "templateid": "10001"
                        }
                    ]
                },
                {
                    "operationid": "2",
                    "actionid": "2",
                    "operationtype": "4",
                    "esc_period": "0s",
                    "esc_step_from": "1",
                    "esc_step_to": "1",
```

```
                    "evaltype": "0",
                    "opconditions": [],
                    "opgroup": [
                        {
                            "groupid": "2"
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- アクションフィルター
- アクションの実行内容

ソース

CAction::get() in *ui/include/classes/api/services/CAction.php*.

action.update

説明

`object action.update(object/array actions)`

このメソッドは、既存のアクションを更新することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 更新するアクションのプロパティ。

各アクションでは actionid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のアクションプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
filter	object	現在のフィルターを置き換えるアクションフィルター オブジェクト。
operations	array	既存の実行内容を置き換えるアクションの実行内容。
recovery_operations	array	既存のリカバリー実行内容を置き換えるアクションのリカバリー実行内容。 パラメータの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合に *supported*
update_operations	array	既存の更新実行内容を置き換えるアクションの更新実行内容。 パラメータの動作: - アクションオブジェクト の eventsource が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合に *supported*

戻り値

(object) actionids プロパティの下で更新されたアクション ID を含むオブジェクトを返します。

例

アクションの無効化

ステータスを”1” に設定してアクションを無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- アクションフィルター
- アクション操作

ソース

CAction::update() in *ui/include/classes/api/services/CAction.php*.

イベント

このクラスはイベントを扱うために設計されています。

オブジェクト参照:

- イベント
- イベントタグ
- メディアタイプ URL

利用可能なメソッド:

- event.get - イベントを取得
- event.acknowledge - イベントを確認済みにする

Event オブジェクト

次のオブジェクトは、eventAPI に直接関連しています。

イベント

> **Note:**
> イベントは Zabbix サーバーによって作成され、API 経由で変更することはできません。

イベントオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
eventid	ID	イベントの ID。

プロパティ	型	説明
source	integer	イベントの種類。 指定可能な値: 0 - トリガーによって作成されたイベント; 1 - ディスカバリルールによって作成されたイベント; 2 - アクティブエージェントの自動登録によって作成されたイベント; 3 - 内部イベント; 4 - サービスステータス更新時に作成されたイベント。
object	integer	イベントに関連するオブジェクトの種類。 source が「トリガーによって作成されたイベント」に設定されている場合の指定可能な値: 0 - トリガー。 source が「ディスカバリルールによって作成されたイベント」に設定されている場合の指定可能な値: 1 - 発見されたホスト; 2 - 発見されたサービス。 source が「アクティブエージェントの自動登録によって作成されたイベント」に設定されている場合の指定可能な値: 3 - 自動登録されたホスト。 source が「内部イベント」に設定されている場合の指定可能な値: 0 - トリガー; 4 - アイテム; 5 - LLD ルール。 source が「サービスステータス更新時に作成されたイベント」に設定されている場合の指定可能な値: 6 - サービス。
objectid	ID	関連オブジェクトの ID。
acknowledged	integer	イベントが確認済みかどうか。
clock	timestamp	イベントが作成された時刻。
ns	integer	イベントが作成された時点のナノ秒。
name	string	解決済みのイベント名。
value	integer	関連オブジェクトの状態。 source が「トリガーによって作成されたイベント」または「サービスステータス更新時に作成されたイベント」に設定されている場合の指定可能な値: 0 - OK; 1 - 障害。 source が「ディスカバリルールによって作成されたイベント」に設定されている場合の指定可能な値: 0 - ホストまたはサービスが稼働中; 1 - ホストまたはサービスが停止中; 2 - ホストまたはサービスが発見された; 3 - ホストまたはサービスが失われた。 source が「内部イベント」に設定されている場合の指定可能な値: 0 - 「正常」状態; 1 - 「不明」または「未サポート」状態。 プロパティの動作: - source が「トリガーによって作成されたイベント」、「ディスカバリルールによって作成されたイベント」、「内部イベント」、または「サービスステータス更新時に作成されたイベント」に設定されている場合は *supported*

プロパティ	型	説明
severity	integer	イベントの現在の深刻度。 指定可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。
r_eventid	ID	復旧イベントの ID。
c_eventid	ID	グローバル相関ルールの下で現在のイベントを上書き（クローズ）するために使用されたイベントの ID。正確な相関ルールを特定するには `correlationid` を参照してください。 このパラメータは、イベントがグローバル相関ルールによってクローズされた場合にのみ定義されます。
cause_eventid	ID	原因イベントの ID。
correlationid	ID	障害のクローズを生成した相関ルールの ID。 このパラメータは、イベントがグローバル相関ルールによってクローズされた場合にのみ定義されます。
userid	ID	イベントをクローズしたユーザーの ID（イベントが手動でクローズされた場合）。
suppressed	integer	イベントが抑制されているかどうか。 指定可能な値: 0 - イベントは通常状態; 1 - イベントは抑制されている。
opdata	string	展開済みマクロを含む運用データ。
urls	array	アクティブなメディアタイプ URL。

イベントタグ

イベントタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	イベントタグ名。
value	string	イベントタグの値。

メディアタイプ URL

メディアタイプ URL オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	メディアタイプで定義された URL 名。
url	string	メディアタイプで定義された URL 値。

結果には、有効なイベントメニューエントリを持つアクティブなメディアタイプのエントリのみが含まれます。プロパティ内で使用されているマクロは展開されますが、いずれかのプロパティに未展開のマクロが含まれている場合、両方のプロパティが結果から除外されます。サポートされているマクロについては、サポートされているマクロを参照してください。

event.acknowledge

説明

```
object event.acknowledge(object/array parameters)
```

このメソッドを使用すると、イベントを更新できます。実行できる更新アクションは次のとおりです。

- イベントをクローズする。イベントがすでに解決済みの場合、このアクションはスキップされます。
- イベントを確認済みにする。イベントがすでに確認済みの場合、このアクションはスキップされます。
- イベントの確認を解除する。イベントが確認済みでない場合、このアクションはスキップされます。

- メッセージを追加する。
- イベントの重大度を変更する。イベントがすでに同じ重大度である場合、このアクションはスキップされます。
- イベントを抑制する。イベントがすでに抑制されている場合、このアクションはスキップされます。
- イベントの抑制を解除する。イベントが抑制されていない場合、このアクションはスキップされます。
- イベントのランクを原因に変更する。イベントがすでに原因としてランク付けされている場合、このアクションはスキップされます。
- イベントのランクを症状に変更する。イベントがすでに症状としてランク付けされている場合、このアクションはスキップされます。

Attention:
更新できるのはトリガーイベントのみです。
更新できるのは障害イベントのみです。
イベントをクローズする、またはイベントの重大度を変更するには、トリガーに対する読み取り/書き込み権限が必要です。
イベントをクローズするには、トリガーで手動クローズが許可されている必要があります。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object/array) イベントの ID と、実行する更新操作を含むパラメータです。

Parameter	Type	Description
eventids	ID/array	確認応答するイベントの ID。 Parameter behavior: - 必須
action	integer	イベント更新アクション。 指定可能なビットマップ値: 1 - 障害をクローズ; 2 - イベントを確認応答; 4 - メッセージを追加; 8 - 深刻度を変更; 16 - イベントの確認応答を解除; 32 - イベントを抑制; 64 - イベントの抑制を解除; 128 - イベントランクを原因に変更; 256 - イベントランクを症状に変更。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、34 はイベントの確認応答と抑制を意味します）。 Parameter behavior: - 必須
cause_eventid	ID	原因イベント ID。 Parameter behavior: - action に「イベントランクを症状に変更」ビットが含まれる場合は 必須
message	string	メッセージのテキスト。 Parameter behavior: - action に「メッセージを追加」ビットが含まれる場合は 必須

Parameter	Type	Description
severity	integer	イベントの新しい深刻度。 指定可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 Parameter behavior: - `action` に「深刻度を変更」ビットが含まれる場合は 必須
suppress_until	integer	イベントを抑制する期限の Unix タイムスタンプ。 ”0” に設定した場合、抑制は無期限になります。 Parameter behavior: - `action` に「イベントを抑制」ビットが含まれる場合は 必須

戻り値

(`object`) `eventids` プロパティの下で更新されたイベントの ID を含むオブジェクトを返します。

例

イベントの確認

1 つのイベントを確認し、メッセージを残します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
        "message": "Problem resolved."
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

イベントの深刻度の変更

複数のイベントの深刻度を変更し、メッセージを残します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
```

```
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
            "20428"
        ]
    },
    "id": 1
}
```

ソース

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

説明

`integer/array event.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってイベントを取得できます。

Attention:
このメソッドは、これらのイベントがまだハウスキーパーによって削除されていない場合、削除済みエンティティのイベントを返すことがあります。

Note:
このメソッドは、すべての種類のユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
eventids	ID/array	指定した ID のイベントのみを返します。
groupids	ID/array	指定したホストグループに属するオブジェクトによって作成されたイベントのみを返します。
hostids	ID/array	指定したホストに属するオブジェクトによって作成されたイベントのみを返します。
objectids	ID/array	指定したオブジェクトによって作成されたイベントのみを返します。
source	integer	指定したタイプのイベントのみを返します。 サポートされているイベントタイプの一覧は、event object pageを参照してください。 デフォルト: 0 - トリガーイベント。

Parameter	Type	Description
object	integer	指定したタイプのオブジェクトによって作成されたイベントのみを返します。 サポートされているオブジェクトタイプの一覧は、event object pageを参照してください。 デフォルト: 0 - トリガー。
acknowledged	boolean	`true` に設定すると、確認済みのイベントのみを返します。
action	integer	指定したevent update actionsが実行されたイベントのみを返します。複数のアクションを指定する場合は、許容されるビットマップ値の合計をビットマスクとして使用します（例: acknowledge と suppress event の場合は 34）。
action_userids	ID/array	イベント更新アクションを実行したユーザーの指定した ID を持つイベントのみを返します。
suppressed	boolean	`true` に設定すると、抑制されたイベントのみを返します。
symptom	boolean	`true` に設定すると、症状イベントのみを返します。
severities	integer/array	指定したイベント深刻度のイベントのみを返します。object が trigger の場合にのみ適用されます。
trigger_severities	integer/array	指定したトリガー深刻度のイベントのみを返します。object が trigger の場合にのみ適用されます。
evaltype	integer	タグのevaluation method。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	array	指定したタグを持つイベントのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのイベントを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist。
eventid_from	string	指定した ID 以上の ID を持つイベントのみを返します。
eventid_till	string	指定した ID 以下の ID を持つイベントのみを返します。
time_from	timestamp	指定した時刻以降に作成されたイベントのみを返します。
time_till	timestamp	指定した時刻以前に作成されたイベントのみを返します。
problem_time_from	timestamp	現在のステータスに関係なく、`problem_time_from` 以降に障害状態であったイベントのみを返します。source がトリガーイベントで、object がトリガーの場合にのみ適用されます。`problem_time_till` を指定する場合、このパラメータは必須です。
problem_time_till	timestamp	現在のステータスに関係なく、`problem_time_till` まで障害状態であったイベントのみを返します。source がトリガーイベントで、object がトリガーの場合にのみ適用されます。`problem_time_from` を指定する場合、このパラメータは必須です。
value	integer/array	指定した値を持つイベントのみを返します。

Parameter	Type	Description
selectAcknowledges	query	イベント更新を含む acknowledges プロパティを返します。イベント更新は新しい順にソートされます。 イベント更新オブジェクトは以下のプロパティを持ちます: acknowledgeid - (ID) 確認の ID。 userid - (ID) イベントを更新したユーザーの ID。 clock - (timestamp) イベントが更新された時刻。 message - (string) メッセージのテキスト。 action - (integer) 実行された更新アクション。 event.acknowledgeを参照してください。 old_severity - (integer) この更新アクション前のイベント深刻度。 new_severity - (integer) この更新アクション後のイベント深刻度。 suppress_until - (timestamp) イベントが抑制される期限。 taskid - (ID) 現在のイベントでランク変更が進行中の場合のタスク ID。 username - (string) イベントを更新したユーザーのユーザー名。 name - (string) イベントを更新したユーザーの名前。 surname - (string) イベントを更新したユーザーの姓。 count をサポートします。
selectAlerts	query	イベントによって生成されたアラートを含むalerts プロパティを返します。アラートは新しい順にソートされます。
selectHosts	query	イベントを作成したオブジェクトを含むホストを含むhosts プロパティを返します。トリガー、アイテム、または LLD ルールによって生成されたイベントでのみサポートされます。
selectRelatedObject	query	イベントを作成したオブジェクトを含む relatedObject プロパティを返します。返されるオブジェクトのタイプはイベントタイプによって異なります。
selectSuppressionData	query	アクティブなメンテナンスおよび手動抑制の一覧を含む suppression_data プロパティを返します: maintenanceid - (ID) メンテナンスの ID。 userid - (ID) イベントを抑制したユーザーの ID。 suppress_until - (integer) イベントが抑制される期限。
selectTags	query	イベントタグを含むtags プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 プロパティ名をキーとし、値として単一の値または照合対象の値の配列を指定するオブジェクトを受け付けます。 text data type のプロパティはサポートされません。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: eventid, objectid, clock。 groupBy と併用する場合の指定可能な値: objectid。 countOutput および groupBy と併用する場合の指定可能な値: objectid, rowscount。
groupBy	string/array	指定したプロパティで結果をグループ化します。指定したプロパティは結果に返されます。 指定可能な値: objectid。
countOutput	boolean	これらのパラメータはreference commentaryで説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	

Parameter	Type	Description
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されているが、groupBy パラメータが使用されていない場合、取得したオブジェクトの数
- groupBy パラメーターが使用されている場合、集計結果を含むオブジェクトの配列

例

トリガーイベントの取得

トリガー"22395" から最新のイベントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "suppressed": "0",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
```

```
                "tag": "scope",
                "value": "performance"
            },
            {
                "tag": "component",
                "value": "cpu"
            },
            {
                "tag": "class",
                "value": "os"
            },
            {
                "tag": "target",
                "value": "linux"
            }
        ],
        "urls": []
    },
    {
        "eventid": "4",
        "source": "0",
        "object": "0",
        "objectid": "22395",
        "clock": "1728657737",
        "value": "1",
        "acknowledged": "1",
        "ns": "460759366",
        "name": "Load average is too high (per CPU load over 1.5 for 5m)",
        "severity": "3",
        "r_eventid": "20",
        "c_eventid": "0",
        "correlationid": "0",
        "userid": "0",
        "cause_eventid": "0",
        "acknowledges": [
            {
                "acknowledgeid": "1",
                "userid": "1",
                "clock": "1728657938",
                "message": "Testing environment. Please disregard this alert.",
                "action": "38",
                "old_severity": "0",
                "new_severity": "0",
                "suppress_until": "1728744338",
                "taskid": "0",
                "username": "Admin",
                "name": "Zabbix",
                "surname": "Administrator"
            }
        ],
        "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
        "suppression_data": [
            {
                "maintenanceid": "0",
                "suppress_until": "1728744338",
                "userid": "1"
            }
        ],
        "suppressed": "1",
        "tags": [
            {
                "tag": "scope",
```

```
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```

期間を指定したイベントの取得

2012 年 10 月 17 日から 18 日の間に作成されたすべてのイベントを、時系列の新しい順で取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": " 履歴キャッシュの空き容量が 25% 未満",
            "severity": "3",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
```

```
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20616",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477814",
            "value": "0",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper プロセスのビジー率が 75% を超過",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper プロセスのビジー率が 75% を超過",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


指定したユーザーによって確認済みとされたイベントの取得

ID=10 のユーザーによって確認済みとされたイベントを取得します

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```

問題イベント数が多い上位トリガーの取得

指定した期間内の問題イベント数とともに、深刻度が「Warning」、「Average」、「High」、または「Disaster」の上位 5 件のトリガーを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "countOutput": true,
        "groupBy": "objectid",
        "source": 0,
        "object": 0,
        "value": 1,
        "time_from": 1672531200,
        "time_till": 1677628800,
        "trigger_severities": [2, 3, 4, 5],
        "sortfield": ["rowscount"],
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "objectid": "232124",
            "rowscount": "27"
        },
        {
            "objectid": "29055",
            "rowscount": "23"
        },
        {
            "objectid": "253731",
            "rowscount": "18"
        },
        {
            "objectid": "254062",
            "rowscount": "11"
        },
        {
            "objectid": "23216",
            "rowscount": "7"
        }
    ],
    "id": 1
}
```


参照

- Alert
- Item
- Host
- LLD rule
- Service
- Trigger

ソース

CEvent::get() in *ui/include/classes/api/services/CEvent.php*.

グラフアイテム

このクラスは、グラフアイテムを操作するために設計されています。

オブジェクト参照:

- グラフアイテム

利用可能なメソッド:

- graphitem.get - グラフアイテムを取得します

Graph item オブジェクト

次のオブジェクトは、graphitemAPI に直接関連しています。

グラフアイテム

> **Note:**
> グラフアイテムは graph API 経由でのみ変更できます。

グラフアイテムオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
gitemid	ID	グラフアイテムの ID。 プロパティの動作: - 読み取り専用
color	string	グラフアイテムの描画色（16 進数のカラーコード）。 プロパティの動作: - 作成操作では 必須
itemid	ID	アイテムの ID。 プロパティの動作: - 作成操作では 必須
calc_fnc	integer	表示されるアイテムの値。 指定可能な値: 1 - 最小値; 2 - *(デフォルト)* 平均値; 4 - 最大値; 7 - すべての値; 9 - 最後の値。円グラフおよび分割円グラフでのみ使用されます。
drawtype	integer	グラフアイテムの描画スタイル。 指定可能な値: 0 - *(デフォルト)* 線; 1 - 塗りつぶし領域; 2 - 太線; 3 - 点; 4 - 破線; 5 - グラデーション線。
graphid	ID	このグラフアイテムが属するグラフの ID。
sortorder	integer	グラフ内でのアイテムの位置。 デフォルト: "0" から始まり、エントリごとに 1 ずつ増加します。
type	integer	グラフアイテムのタイプ。 指定可能な値: 0 - *(デフォルト)* シンプル; 2 - グラフ合計。円グラフおよび分割円グラフでのみ使用されます。
yaxisside	integer	グラフアイテムの Y 軸スケールを描画するグラフの側。 指定可能な値: 0 - *(デフォルト)* 左側; 1 - 右側。

graphitem.get

説明

`integer/array graphitem.get(object parameters)`

このメソッドは、指定したパラメーターに応じたグラフアイテムを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
graphids	ID/array	指定したグラフに属するグラフアイテムのみを返します。
itemids	ID/array	指定したアイテム ID を持つグラフアイテムのみを返します。
type	integer	指定したタイプのグラフアイテムのみを返します。 サポートされているグラフアイテムタイプの一覧については、グラフアイテムオブジェクトのページを参照してください。
selectGraphs	query	アイテムが属するグラフの配列を含むgraphsプロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: gitemid。
countOutput	boolean	これらのパラメータについては、リファレンスコメントで説明されています。
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

グラフからグラフアイテムの取得

グラフで使用されているすべてのグラフアイテムを、アイテムとホストに関する追加情報とともに取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "gitemid": "1242",
            "graphid": "387",
            "itemid": "22665",
            "drawtype": "1",
            "sortorder": "1",
            "color": "FF5555",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1243",
            "graphid": "387",
            "itemid": "22668",
            "drawtype": "1",
            "sortorder": "2",
```

```
            "color": "55FF55",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        },
        {
            "gitemid": "1244",
            "graphid": "387",
            "itemid": "22671",
            "drawtype": "1",
            "sortorder": "3",
            "color": "009999",
            "yaxisside": "0",
            "calc_fnc": "2",
            "type": "0"
        }
    ],
    "id": 1
}
```


参照

- Graph

ソース

CGraphItem::get() in *ui/include/classes/api/services/CGraphItem.php*.

グラフプロトタイプ

このクラスは、グラフプロトタイプを操作するために設計されています。

オブジェクト参照:

- グラフプロトタイプ

利用可能なメソッド:

- graphprototype.create - 新しいグラフプロトタイプを作成
- graphprototype.delete - グラフプロトタイプを削除
- graphprototype.get - グラフプロトタイプを取得
- graphprototype.update - グラフプロトタイプを更新

Graph prototype オブジェクト

次のオブジェクトは、`graphprototype`API に直接関連しています。

グラフプロトタイプ

グラフプロトタイプオブジェクトには、以下のプロパティがあります。

Property	Type	Description
graphid	ID	グラフプロトタイプの ID。 Property behavior: - *read-only* - 更新操作で *required*
height	integer	グラフプロトタイプの高さ（ピクセル単位）。 Property behavior: - 作成操作で *required*
name	string	グラフプロトタイプの名前。 Property behavior: - 作成操作で *required*

Property	Type	Description
width	integer	グラフプロトタイプの幅（ピクセル単位）。 Property behavior: - 作成操作で *required*
flags	integer	グラフプロトタイプの生成元。 指定可能な値: 2 - グラフプロトタイプ; 6 - ディスカバリされたグラフプロトタイプ Property behavior: - *read-only*
graphtype	integer	グラフプロトタイプのレイアウトタイプ。 指定可能な値: 0 - *(default)* 通常; 1 - 積み上げ; 2 - 円グラフ; 3 - 分離。
percent_left	float	左パーセンタイル。 デフォルト: 0。
percent_right	float	右パーセンタイル。 デフォルト: 0。
show_3d	integer	ディスカバリされた円グラフおよび分離グラフを 3D で表示するかどうか。 指定可能な値: 0 - *(default)* 2D で表示; 1 - 3D で表示。
show_legend	integer	ディスカバリされたグラフに凡例を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(default)* 表示。
show_work_period	integer	ディスカバリされたグラフに稼働時間を表示するかどうか。 指定可能な値: 0 - 非表示; 1 - *(default)* 表示。
templateid	ID	親テンプレートのグラフプロトタイプの ID。 Property behavior: - *read-only*
yaxismax	float	Y 軸の固定最大値。
yaxismin	float	Y 軸の固定最小値。
ymax_itemid	ID	Y 軸の最大値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは `ymax_type` が”calculated” に設定されているものとして描画されます。
ymax_type	integer	Y 軸の最大値の計算方法。 指定可能な値: 0 - *(default)* calculated; 1 - fixed; 2 - item.
ymin_itemid	ID	Y 軸の最小値として使用されるアイテムの ID。 ユーザーが指定されたアイテムにアクセスできない場合、グラフは `ymin_type` が”calculated” に設定されているものとして描画されます。

Property	Type	Description
ymin_type	integer	Y 軸の最小値の計算方法。 指定可能な値: 0 - *(default)* calculated; 1 - fixed; 2 - item.
discover	integer	グラフプロトタイプのディスカバリ状態。 指定可能な値: 0 - *(default)* 新しいグラフがディスカバリされる; 1 - 新しいグラフはディスカバリされず、既存のグラフは失われたものとしてマークされる。
uuid	string	インポートされたグラフプロトタイプを既存のものにリンクするために使用される汎用一意識別子。指定されていない場合は自動生成されます。 Property behavior: - グラフプロトタイプがテンプレートに属している場合に *supported*

graphprototype.create

説明

`object graphprototype.create(object/array graphPrototypes)`

このメソッドでは、新しいグラフプロトタイプを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object/array) 作成するグラフプロトタイプ。

このメソッドは、標準のグラフプロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	グラフプロトタイプ用に作成されるグラフアイテム。グラフアイテムはアイテムとアイテムプロトタイプの両方を参照できますが、少なくとも 1 つのアイテムプロトタイプが含まれている必要があります。 Parameter behavior: - *required*

戻り値

(object) graphids プロパティの下に作成されたグラフプロトタイプの ID を含むオブジェクトを返します。
 返された ID の順番は、渡されたグラフプロトタイプの順番と一致する。

例

グラフプロトタイプの作成

2 つのアイテムを持つグラフプロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Disk space usage {#FSNAME}",
        "width": 900,
```

```
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

関連項目

- グラフアイテム

ソース

CGraphPrototype::create() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

説明

`object graphprototype.delete(array graphPrototypeIds)`

このメソッドは、グラフプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するグラフプロトタイプの ID。

戻り値

(object) graphids プロパティの下にある削除されたグラフプロトタイプの ID を含むオブジェクトを返します。

例

複数のグラフプロトタイプの削除

2 つのグラフプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

ソース

CGraphPrototype::delete() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

説明

`integer/array graphprototype.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってグラフプロトタイプを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。
このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
discoveryids	ID/array	指定したディスカバリルールに属するグラフプロトタイプのみを返します。
graphids	ID/array	指定した ID を持つグラフプロトタイプのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループ内のホストまたはテンプレートに属するグラフプロトタイプのみを返します。
hostids	ID/array	指定したホストに属するグラフプロトタイプのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承されたグラフプロトタイプのみを返します。
itemids	ID/array	指定したアイテムプロトタイプを含むグラフプロトタイプのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属するグラフプロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するグラフプロトタイプのみを返します。

Parameter	Type	Description
selectDiscoveryData	query	グラフプロトタイプのディスカバリオブジェクトデータを含む discoveryData プロパティを返します。グラフプロトタイプのディスカバリオブジェクトは、検出されたグラフプロトタイプを、その検出元となったグラフプロトタイプに関連付けます。 以下のプロパティがあります。 parent_graphid - (ID) グラフの作成元となったグラフプロトタイプの ID。 status - (int) グラフのディスカバリステータス: 0 - *(default)* グラフプロトタイプは検出されています。 1 - グラフプロトタイプはすでに検出されていません。 ts_delete - (timestamp) これ以上検出されないグラフプロトタイプが削除される時刻。
selectDiscoveryRule	query	グラフプロトタイプが属する LLD ルールを含むdiscoveryRule プロパティを返します。
selectDiscoveryRulePrototype	query	グラフプロトタイプが属する親 LLD ルールプロトタイプを含むdiscoveryRulePrototype プロパティを返します。
selectGraphItems	query	グラフプロトタイプで使用されるグラフアイテムを含むgitems プロパティを返します。
selectHostGroups	query	グラフプロトタイプが属するホストグループを含むhostgroups プロパティを返します。
selectHosts	query	グラフプロトタイプが属するホストを含むhosts プロパティを返します。
selectItems	query	グラフプロトタイプで使用されるアイテム およびアイテムプロトタイプを含む items プロパティを返します。
selectTemplateGroups	query	グラフプロトタイプが属するテンプレートグループを含むtemplategroups プロパティを返します。
selectTemplates	query	グラフプロトタイプが属するテンプレートを含むtemplates プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 text data type のプロパティはサポートしていません。 追加のプロパティをサポートしています。 host - グラフプロトタイプが属するホストの技術名。 hostid - グラフプロトタイプが属するホストの ID。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: graphid, name, graphtype, discovered。
countOutput	boolean	これらのパラメータについては、reference commentary を参照してください。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

LLD ルールから graph prototypes を取得する

LLD ルールからすべての graph prototypes を取得します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "auth": "038e1d7b1735c6a5436ee9eae095879e",
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "parent_itemid": "27426",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "discover": "0"
        }
    ],
    "id": 1
}
```

LLD ルールからグラフプロトタイプを取得する

LLD ルールからすべてのグラフプロトタイプを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "graphid": "1017",
            "name": "Disk space usage {#FSNAME}",
            "width": "600",
            "height": "340",
            "yaxismin": "0.0000",
            "yaxismax": "0.0000",
            "templateid": "442",
            "show_work_period": "0",
            "show_triggers": "0",
            "graphtype": "2",
            "show_legend": "1",
            "show_3d": "1",
            "percent_left": "0.0000",
            "percent_right": "0.0000",
            "ymin_type": "0",
            "ymax_type": "0",
            "ymin_itemid": "0",
            "ymax_itemid": "0",
            "flags": "2",
            "discover": "0"
        }
    ],
    "id": 1
}
```


関連項目

- ディスカバリルール
- グラフアイテム
- アイテム
- ホスト
- ホストグループ
- テンプレート
- テンプレートグループ

ソース

CGraphPrototype::get() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

説明

`object graphprototype.update(object/array graphPrototypes)`

このメソッドは、既存のグラフプロトタイプを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(`object/array`) 更新するグラフプロトタイプのプロパティ。

各グラフプロトタイプでは `graphid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のグラフプロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
gitems	array	既存のグラフアイテムを置き換えるグラフアイテム。グラフアイテムにgitemid プロパティが定義されている場合は更新され、そうでない場合は新しいグラフアイテムが作成されます。

戻り値

(object) graphids プロパティの下で更新された graph prototypes の ID を含むオブジェクトを返します。

例

グラフプロトタイプのサイズの変更

グラフプロトタイプのサイズを 1100 x 400 ピクセルに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

ソース

CGraphPrototype::update() in *ui/include/classes/api/services/CGraphPrototype.php*.

サービス

このクラスは、IT インフラストラクチャ/ビジネスサービスを扱うために設計されています。

オブジェクト参照:

- サービス
- ステータスルール
- サービスタグ
- サービスアラーム
- 障害タグ

利用可能なメソッド:

- service.create - 新しいサービスを作成
- service.delete - サービスを削除
- service.get - サービスを取得
- service.update - サービスを更新

Service オブジェクト

次のオブジェクトは、serviceAPI に直接関連しています。

サービス

サービスオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
serviceid	ID	サービスの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
algorithm	integer	ステータス計算ルール。子サービスが存在する場合にのみ適用されます。 指定可能な値: 0 - ステータスを OK に設定; 1 - すべての子に問題がある場合、最も重大なもの; 2 - 子サービスの中で最も重大なもの。 プロパティの動作: - 作成操作では 必須
name	string	サービス名。 プロパティの動作: - 作成操作では 必須
sortorder	integer	ソートに使用されるサービスの位置。 指定可能な値: 0-999。 プロパティの動作: - 作成操作では 必須
weight	integer	サービスの重み。 指定可能な値: 0-1000000。 デフォルト: 0。
propagation_rule	integer	ステータス伝播ルール。 指定可能な値: 0 - *(デフォルト)* サービスのステータスを変更せずそのまま伝播する; 1 - 指定された propagation_value だけ伝播されるステータスを増加させる（重大度を 1～5 段階上げる）; 2 - 指定された propagation_value だけ伝播されるステータスを減少させる（重大度を 1～5 段階下げる）; 3 - このサービスを無視する - ステータスは親サービスにまったく伝播されない; 4 - 指定された propagation_value を使用して固定のサービスステータスを設定する。 プロパティの動作: - propagation_value が設定されている場合は 必須

プロパティ	型	説明
propagation_value	integer	ステータス伝播値。 propagation_rule が"0" または"3" に設定されている場合の指定可能な値: 0 - 未分類。 propagation_rule が"1" または"2" に設定されている場合の指定可能な値: 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害。 propagation_rule が"4" に設定されている場合の指定可能な値: -1 - OK; 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害。 プロパティの動作: - propagation_rule が設定されている場合は 必須
status	integer	サービスが OK 状態か問題状態か。 サービスが問題状態の場合、status は次のいずれかに等しくなります: - 最も重大な問題の重大度; - 問題状態にある子サービスの中で最も高いステータス。 サービスが OK 状態の場合、status は -1 に等しくなります。 プロパティの動作: - 読み取り専用
description	string	サービスの説明。
uuid	string	インポートされたサービスを既存のサービスに関連付けるために使用される汎用一意識別子。指定されていない場合は自動生成されます。
created_at	integer	サービスが作成された日時の Unix タイムスタンプ。
readonly	integer	サービスへのアクセス。 指定可能な値: 0 - 読み書き可能; 1 - 読み取り専用。 プロパティの動作: - 読み取り専用

ステータスルール

ステータスルールオブジェクトには、以下のプロパティがあります。

Property	Type	Description
type	integer	(新しいステータス) ステータスを設定する条件。 使用可能な値: 0 - 少なくとも (N) 個の子サービスが (ステータス) ステータス以上である場合; 1 - 少なくとも (N%) の子サービスが (ステータス) ステータス以上である場合; 2 - (N) 個未満の子サービスが (ステータス) ステータス以下である場合; 3 - (N%) 未満の子サービスが (ステータス) ステータス以下である場合; 4 - (ステータス) ステータス以上の子サービスの重みの合計が少なくとも (W) である場合; 5 - (ステータス) ステータス以上の子サービスの重みの合計が少なくとも (N%) である場合; 6 - (ステータス) ステータス以下の子サービスの重みの合計が (W) 未満である場合; 7 - (ステータス) ステータス以下の子サービスの重みの合計が (N%) 未満である場合。 ここで: - N (W) は `limit_value`; - (ステータス) は `limit_status`; - (新しいステータス) は `new_status`。 プロパティの動作: - 必須
limit_value	integer	制限値。 使用可能な値: - N および W の場合: 1-100000; - N% の場合: 1-100。 プロパティの動作: - 必須
limit_status	integer	制限ステータス。 使用可能な値: -1 - OK; 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 プロパティの動作: - 必須
new_status	integer	新しいステータス値。 使用可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。 プロパティの動作: - 必須

サービスタグ

サービスタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	サービスタグ名。 プロパティの動作: - 必須
value	string	サービスタグの値。

サービスアラーム

Note:
サービスアラームは、Zabbix API を介して直接作成、更新、または削除することはできません。

サービスアラームオブジェクトは、サービスの状態変更を表します。以下のプロパティがあります。

プロパティ	Type	説明
clock	timestamp	サービスの状態変更が発生した時刻。
value	integer	サービスのステータス。 使用可能な値の一覧については、service `status` プロパティを参照してください。

障害タグ

障害タグを使用すると、サービスを障害イベントに関連付けることができます。障害タグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	障害タグ名。 プロパティの動作: - 必須
operator	integer	operator 条件のマッピング。 使用可能な値: 0 - *(デフォルト)* Equals; 2 - Contains.
value	string	障害タグの値。

service.create

説明

`object service.create(object/array services)`

このメソッドでは、新しいサービスを作成できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。
このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

`(object/array)` 作成するサービス。

標準のサービスプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
children	array	サービスにリンクする子サービス。 子サービスでは、serviceid プロパティのみを定義する必要があります。
parents	array	サービスにリンクする親サービス。 親サービスでは、serviceid プロパティのみを定義する必要があります。
tags	array	サービス用に作成するサービスタグ。
problem_tags	array	サービス用に作成する障害タグ。
status_rules	array	サービス用に作成するステータスルール。

戻り値

(object) serviceids プロパティの下で作成されたサービスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたサービスの順序と一致します。

例

サービスの作成

少なくとも 1 つの子プロセスに問題がある場合、問題状態に切り替えるサービスを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CService::create() in *ui/include/classes/api/services/CService.php*.

service.delete

説明

`object service.delete(array serviceIds)`

このメソッドは、サービスを削除することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するサービスの ID。

戻り値

(object) serviceids プロパティの下にある削除されたサービスの ID を含むオブジェクトを返します。

例

複数のサービスの削除

2 つのサービスを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CService::delete() in *ui/include/classes/api/services/CService.php*.

service.get

説明

`integer/array service.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってサービスを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
serviceids	ID/array	指定した ID を持つサービスのみを返します。
parentids	ID/array	指定した親サービスにリンクされているサービスのみを返します。
deep_parentids	flag	直接および間接のすべての子サービスを返します。parentids と一緒に使用します。
childids	ID/array	指定した子サービスにリンクされているサービスのみを返します。

Parameter	Type	Description
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	object/array	指定したタグを持つサービスのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのサービスを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
problem_tags	object/array	指定した障害タグを持つサービスのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのサービスを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
without_problem_tags	flag	障害タグを持たないサービスのみを返します。
slaids	ID/array	指定した SLA にリンクされているサービスのみを返します。
selectChildren	query	子サービスを含む `children` プロパティを返します。 `count` をサポートします。
selectParents	query	親サービスを含む `parents` プロパティを返します。 `count` をサポートします。
selectTags	query	サービスタグを含む`tags` プロパティを返します。 `count` をサポートします。
selectProblemEvents	query	障害イベントオブジェクトの配列を含む `problem_events` プロパティを返します。 障害イベントオブジェクトは以下のプロパティを持ちます: `eventid` - (`ID`) イベント ID; `severity` - (`string`) 現在のイベント重大度; `name` - (`string`) 解決済みイベント名。 `count` をサポートします。
selectProblemTags	query	障害タグを含む`problem_tags` プロパティを返します。 `count` をサポートします。
selectStatusRules	query	ステータスルールを含む`status_rules` プロパティを返します。 `count` をサポートします。

Parameter	Type	Description
selectStatusTimeline	object/array	指定した期間のサービス状態の変化を含む `status_timeline` プロパティを返します。 形式 `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - `period_from` は対象期間の開始日時（含む; integer timestamp）、`period_to` は対象期間の終了日時（含まない; integer timestamp）です。 指定した期間内の状態変化について、`start_value` プロパティと`alarms` 配列を含むエントリの配列を返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `serviceid`, `name`, `status`, `sortorder`, `created_at`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのサービスの取得

すべてのサービスとその依存関係に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "serviceid": "1",
            "name": "Zabbix cluster",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
```

```
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "8d77bd91b62347e4b79382912eb5df95",
            "created_at": "1761742392",
            "readonly": false,
            "parents": [],
            "children": [
                {
                    "serviceid": "2",
                    "name": "Zabbix server node 1",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "195911d26d7f4e218d6217079bcd5929",
                    "created_at": "1761742623",
                    "readonly": false
                },
                {
                    "serviceid": "3",
                    "name": "Zabbix server node 2",
                    "status": "-1",
                    "algorithm": "2",
                    "sortorder": "0",
                    "weight": "0",
                    "propagation_rule": "0",
                    "propagation_value": "0",
                    "description": "",
                    "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
                    "created_at": "1761742654",
                    "readonly": false
                }
            ]
        },
        {
            "serviceid": "2",
            "name": "Zabbix server node 1",
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "195911d26d7f4e218d6217079bcd5929",
            "created_at": "1761742623",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        },
        {
            "serviceid": "3",
            "name": "Zabbix server node 2",
```

```
            "status": "-1",
            "algorithm": "2",
            "sortorder": "0",
            "weight": "0",
            "propagation_rule": "0",
            "propagation_value": "0",
            "description": "",
            "uuid": "9fc659a30fe244f690dff25fc2a9db5c",
            "created_at": "1761742654",
            "readonly": false,
            "parents": [
                {
                    "serviceid": "1",
                    "name": "Zabbix cluster"
                }
            ],
            "children": []
        }
    ],
    "id": 1
}
```


ソース

CService::get() in *ui/include/classes/api/services/CService.php*.

service.update

説明

`object service.update(object/array services)`

このメソッドでは、既存のサービスを更新できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。
> このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object/array) 更新するサービスのプロパティ。

各サービスについて serviceid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

標準のサービスプロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
children	array	現在の子サービスを置き換える子サービス。 子サービスでは serviceid プロパティのみを定義する必要があります。
parents	array	現在の親サービスを置き換える親サービス。 親サービスでは serviceid プロパティのみを定義する必要があります。
tags	array	現在のサービスタグを置き換えるサービスタグ。
problem_tags	array	現在の問題タグを置き換える問題タグ。
status_rules	array	現在のステータスルールを置き換えるステータスルール。

戻り値

(object) serviceids プロパティの下で更新されたサービスの ID を含むオブジェクトを返します。

例

サービスの親の設定

ID が”3” のサービスを、ID が”5” のサービスの親にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

スケジュールされたダウンタイムの追加

ID が”4” のサービスに対して、毎週月曜日 22:00 から火曜日 10:00 までのスケジュールされたダウンタイムを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

ソース

CService::update() in *ui/include/classes/api/services/CService.php*.

スクリプト

このクラスはスクリプトを扱うために設計されています。

オブジェクト参照:

- Script
  - Webhook パラメータ
- Debug
  - ログエントリ

利用可能なメソッド:

- script.create - 新しいスクリプトを作成
- script.delete - スクリプトを削除
- script.execute - スクリプトを実行
- script.get - スクリプトを取得
- script.getscriptsbyevents - イベントのスクリプトを取得
- script.getscriptsbyhosts - ホストのスクリプトを取得
- script.update - スクリプトを更新

Script オブジェクト

次のオブジェクトは、`script`API に直接関連しています。

スクリプト

スクリプトオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
scriptid	ID	スクリプトの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
name	string	スクリプトの名前。 プロパティの動作: - 作成操作で 必須
type	integer	スクリプトのタイプ。 `scope` が「action operation」に設定されている場合の使用可能な値: 0 - スクリプト; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook。 `scope` が「manual host action」または「manual event action」に設定されている場合の使用可能な値: 6 - URL。 プロパティの動作: - 作成操作で 必須
command	string	実行するコマンド。 プロパティの動作: - `type` が「スクリプト」、「IPMI」、「SSH」、「TELNET」、または「Webhook」に設定されている場合に 必須

プロパティ	型	説明
scope	integer	スクリプトのスコープ。 使用可能な値: 1 - action operation; 2 - manual host action; 4 - manual event action。 プロパティの動作: - 作成操作で 必須
execute_on	integer	スクリプトの実行場所。 使用可能な値: 0 - Zabbix エージェント上で実行; 1 - Zabbix サーバー上で実行。これは、Zabbix サーバーでグローバルスクリプトの実行が有効になっている場合にのみ サポートされます; 2 - *(デフォルト)* Zabbix サーバーまたはプロキシ上で実行。 プロパティの動作: - type が「スクリプト」に設定されている場合に サポート
menu_path	string	スラッシュで区切られたフォルダで、ホストまたはイベントをクリックしたときに Web インターフェースでメニューのようなナビゲーションを構成します。 プロパティの動作: - scope が「manual host action」または「manual event action」に設定されている場合に サポート
authtype	integer	SSH スクリプトタイプで使用される認証方式。 使用可能な値: 0 - パスワード; 1 - 公開鍵。 プロパティの動作: - type が「SSH」に設定されている場合に サポート
username	string	認証に使用されるユーザー名。 プロパティの動作: - type が「SSH」または「TELNET」に設定されている場合に 必須
password	string	パスワード認証を使用する SSH スクリプトおよび TELNET スクリプトで使用されるパスワード。 プロパティの動作: - type が「SSH」で authtype が「password」に設定されている場合、または type が「TELNET」に設定されている場合に サポート
publickey	string	公開鍵認証を使用する SSH スクリプトで使用される公開鍵ファイル名。 プロパティの動作: - type が「SSH」で authtype が「public key」に設定されている場合に 必須
privatekey	string	公開鍵認証を使用する SSH スクリプトで使用される秘密鍵ファイル名。 プロパティの動作: - type が「SSH」で authtype が「public key」に設定されている場合に 必須
port	string	SSH および TELNET スクリプトで使用されるポート番号。 プロパティの動作: - type が「SSH」または「TELNET」に設定されている場合に サポート

プロパティ	型	説明
groupid	ID	スクリプトを実行できるホストグループの ID。 「0」に設定すると、スクリプトはすべてのホストグループで利用可能になります。 デフォルト: 0。
usrgrpid	ID	スクリプトの実行を許可されるユーザーグループの ID。 「0」に設定すると、スクリプトはすべてのユーザーグループで利用可能になります。 デフォルト: 0。
host_access	integer	プロパティの動作: - scope が「manual host action」または「manual event action」に設定されている場合に サポート スクリプトの実行に必要なホスト権限。 使用可能な値: 2 - *(デフォルト)* 読み取り; 3 - 書き込み。
confirmation	string	プロパティの動作: - scope が「manual host action」または「manual event action」に設定されている場合に サポート 確認ポップアップのテキスト。 Zabbix Web インターフェースからスクリプトを実行しようとしたときに、このポップアップが表示されます。
timeout	string	プロパティの動作: - scope が「manual host action」または「manual event action」に設定されている場合に サポート webhook スクリプトの実行タイムアウト（秒）。時間の接尾辞をサポートします（例: 30s、1m）。 使用可能な値: 1-60s。 デフォルト: 30s。
parameters	array	プロパティの動作: - type が「Webhook」に設定されている場合に 必須 webhook 入力パラメータの配列。
description	string	プロパティの動作: - type が「Webhook」に設定されている場合に サポート スクリプトの説明。
url	string	ユーザー定義 URL。 プロパティの動作: - type が「URL」に設定されている場合に 必須
new_window	integer	URL を新しいウィンドウで開きます。 使用可能な値: 0 - いいえ; 1 - *(デフォルト)* はい。 プロパティの動作: - type が「URL」に設定されている場合に サポート

プロパティ	型	説明
manualinput	integer	スクリプトがユーザー指定の入力を受け付けるかどうかを示します。 使用可能な値: 0 - *(デフォルト)* 無効; 1 - 有効; プロパティの動作: - `scope` が「manual host action」または「manual event action」に設定されている場合に サポート
manualinput_prompt	string	手動入力のプロンプトテキスト。 プロパティの動作: - `manualinput` が「Enabled」に設定されている場合に 必須
manualinput_validator	string	ユーザーが指定した入力を検証するために使用される文字列フィールドです。この文字列は、正規表現またはカンマ区切りの値セットのいずれかで構成されます。 プロパティの動作: - `manualinput` が「Enabled」に設定されている場合に 必須
manualinput_validator_type	integer	想定されるユーザー入力のタイプを決定します。 使用可能な値: 0 - *(デフォルト)* 文字列。manualinput_validator が正規表現として扱われることを示します; 1 - リスト。manualinput_validator が入力可能な値のカンマ区切りリストとして扱われることを示します。 プロパティの動作: - `manualinput` が「Enabled」に設定されている場合に サポート
manualinput_default_value	string	ユーザー入力を自動入力するためのデフォルト値。 プロパティの動作: - `manualinput_validator_type` が「String」に設定されている場合に サポート

webhook パラメータ

webhook スクリプトの呼び出し時に渡されるパラメータには、以下のプロパティがあります。

プロパティ	型	説明
name	string	パラメータ名。 プロパティの動作: - 必須
value	string	パラメータ値。マクロをサポートします。

デバッグ

実行された webhook スクリプトのデバッグ情報です。debug オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
logs	array	ログエントリの配列。
ms	string	スクリプトの実行時間（ミリ秒）。

ログエントリ

ログエントリオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
level	integer	ログレベル。
ms	string	ログエントリが追加されるまでに、スクリプトの実行開始から経過した時間（ミリ秒）。
message	string	ログメッセージ。

script.create

説明

`object script.create(object/array scripts)`

このメソッドでは、新しいスクリプトを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、User roles を参照してください。

パラメータ

(object/array) 作成するスクリプト

このメソッドは、標準のスクリプトプロパティを持つスクリプトを受け入れます

戻り値

(object) scriptids プロパティの下で作成されたスクリプトの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスクリプトの順序と一致します。

例

webhook スクリプトの作成

外部サービスに HTTP リクエストを送信する webhook スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Webhook script",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "scope": 1,
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

SSH スクリプトの作成

公開鍵認証を使用し、ホスト上で実行でき、コンテキストメニューを持つ SSH スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "SSH script",
        "command": "my script command",
        "type": 2,
        "authtype": 1,
        "username": "John",
        "publickey": "pub.key",
        "privatekey": "priv.key",
        "password": "secret",
        "port": "12345",
        "scope": 2,
        "menu_path": "All scripts/SSH",
        "usrgrpid": "7",
        "groupid": "4"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

カスタムスクリプトの作成

サーバーを再起動するカスタムスクリプトを作成します。このスクリプトはホストへの書き込みアクセスを必要とし、ユーザーに手動入力を求めます。入力が正常に送信されると、スクリプトは Web インターフェースに確認メッセージを表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Reboot server",
        "command": "reboot server {MANUALINPUT}",
        "type": 0,
        "scope": 2,
        "confirmation": "Are you sure you would like to reboot the server {MANUALINPUT}?",
        "manualinput": 1,
        "manualinput_prompt": "Which server you want to reboot?",
        "manualinput_validator": "[1-9]",
        "manualinput_validator_type": 0,
```

```
        "manualinput_default_value": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

URL タイプのスクリプト作成

同じウィンドウに表示され、確認テキストを含む、ホストスコープ用の URL タイプのスクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script",
        "type": 6,
        "scope": 2,
        "url": "http://zabbix/ui/zabbix.php?action=host.edit&hostid={HOST.ID}",
        "confirmation": "Edit host {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

手動入力が可能な URL 型スクリプト作成

新しいウィンドウで開き、手動入力が可能なイベントスコープ用の URL 型スクリプトを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL script with manual input",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Select a page to open:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
```

```
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

ソース

CScript::create() in *ui/include/classes/api/services/CScript.php*.

script.delete

説明

`object script.delete(array scriptIds)`

このメソッドは、スクリプトを削除することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するスクリプトの ID。

戻り値

(object) scriptids プロパティの下にある削除されたスクリプトの ID を含むオブジェクトを返します。

例

複数のスクリプトの削除

2 つのスクリプトを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
```

```
    "id": 1
}
```

ソース

CScript::delete() in *ui/include/classes/api/services/CScript.php*.

script.execute

説明

`object script.execute(object parameters)`

このメソッドでは、ホストまたはイベントでスクリプトを実行できます。ただし、URL タイプのスクリプトは除きます。これらは実行できません。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 実行するスクリプトの ID、およびホストの ID またはイベントの ID と manualinput 値を含むパラメータです。

パラメータ	Type	説明
scriptid	ID	実行するscriptの ID。
hostid	ID	Parameter behavior: - *required* スクリプトを実行するホストの ID。
eventid	ID	Parameter behavior: - `eventid` が設定されていない場合は *required* スクリプトを実行するeventの ID。
manualinput	string	Parameter behavior: - `hostid` が設定されていない場合は *required* {MANUALINPUT} マクロを置換してスクリプトを実行するための、ユーザー指定の値。

戻り値

(`object`) スクリプト実行の結果を返します。

プロパティ	型	説明
response	string	スクリプトが正常に実行されたかどうか。 指定可能な値 - `success`。
value	string	スクリプトの出力。
debug	object	webhook スクリプトが実行された場合は、debug オブジェクトを含みます。その他のスクリプトタイプでは、空のオブジェクトを含みます。

例

webhook スクリプトの実行

外部サービスに HTTP リクエストを送信する webhook スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
```

```
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "{\"status\":\"sent\",\"timestamp\":\"1611235391\"}",
        "debug": {
            "logs": [
                {
                    "level": 3,
                    "ms": 480,
                    "message": "[Webhook Script] HTTP status: 200."
                }
            ],
            "ms": 495
        }
    },
    "id": 1
}
```

カスタムスクリプトの実行

ホストで"ping" スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_req=1 tt
        "debug": []
    },
    "id": 1
}
```

手動入力を使用してカスタムスクリプトを実行する

ホスト上で、コマンド `ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac` を使用して"ping" スクリプトを実行します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
```

```
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "value": "PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.\n64 bytes from 127.0.0.1: icmp_seq=1 tt
        "debug": []
    },
    "id": 1
}
```


ソース

CScript::execute() in *ui/include/classes/api/services/CScript.php*.

script.get

説明

`integer/array script.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってスクリプトを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
groupids	ID/array	指定したホストグループで実行できるスクリプトのみを返します。
hostids	ID/array	指定したホストで実行できるスクリプトのみを返します。
scriptids	ID/array	指定した ID を持つスクリプトのみを返します。
usrgrpids	ID/array	指定したユーザーグループのユーザーが実行できるスクリプトのみを返します。
selectHostGroups	query	スクリプトを実行できるホストグループを含む`hostgroups` プロパティを返します。
selectHosts	query	スクリプトを実行できるホストを含む`hosts` プロパティを返します。
selectActions	query	スクリプトに関連付けられているアクションを含む`actions` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `scriptid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

すべてのスクリプトを取得

設定されているすべてのスクリプトを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
```

```
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "3",
            "name": " オペレーティングシステムを検出",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Webhook",
            "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "5",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
```

```
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": [
                {
                    "name": "token",
                    "value": "{$WEBHOOK.TOKEN}"
                },
                {
                    "name": "host",
                    "value": "{HOST.HOST}"
                },
                {
                    "name": "v",
                    "value": "2.2"
                }
            ]
        },
        {
            "scriptid": "5",
            "name": "URL",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "{HOST.NAME} に移動しますか?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action=latest.view&hostids[]={HOST.ID}",
            "new_window": "0",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator": "",
            "manualinput_validator_type": "0",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "6",
            "name": " ユーザー入力付き URL",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
```

```
            "groupid": "0",
            "description": "",
            "confirmation": "zabbix ページ {MANUALINPUT} を開きますか?",
            "type": "6",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
            "new_window": "0",
            "manualinput": "1",
            "manualinput_prompt": " 開くページを選択してください:",
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- ホストグループ

ソース

CScript::get() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

説明

`object script.getscriptsbyevents(object parameters)`

このメソッドは、指定されたイベントで利用可能なすべてのスクリプト、またはスクリプト ID が指定されている場合は特定のスクリプトを取得します。manualinput が指定されている場合は、{MANUALINPUT} マクロを指定された値に置き換えます。

> **Note:**
> このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(object/array) このメソッドは、以下のパラメータを持つオブジェクト、またはオブジェクトの配列を受け付けます。

Parameter	Type	Description
eventid	ID	スクリプトを返す対象のイベントの ID。 一意である必要があります。 パラメータの動作: - 必須
scriptid	ID	返すスクリプトの ID。
manualinput	string	ユーザーが指定した {MANUALINPUT} マクロ値の値。

戻り値

(object) は、イベント ID をプロパティとして、利用可能なスクリプトの配列を値として持つオブジェクトを返します。スクリプト ID が指定されている場合、関連付けられた値は特定のスクリプトを含む配列です。

Note:
このメソッドは、confirmation テキスト、manualinput prompt テキスト、および url 内のマクロを自動的に展開します。
manualinput パラメータが指定されている場合、{MANUALINPUT} マクロは指定された値に解決されます。

例

イベント ID でスクリプトを取得する

イベント"632" および"614" に使用できるすべてのスクリプトを取得します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Detect operating system",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "4",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
```

```
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
```

```
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        }
    ],
    "614": [
        {
            "scriptid": "3",
            "name": "Detect operating system",
            "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "7",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "1",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "1",
            "name": "Ping",
            "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
```

```
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open page *UNKNOWN*?",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "4",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "4",
```

```
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

manualinput 値を持つ特定のスクリプトの情報を取得します。

イベント”632” の ID”4”、manualinput 値”dashboard.view” を持つスクリプトの情報を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "632": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "4",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
```

```
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": "Zabbix page to open:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
          }
        ]
      },
      "id": 1
}
```


ソース

CScript::getScriptsByEvents() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

説明

`object script.getscriptsbyhosts(object parameters)`

このメソッドを使用すると、指定したホストで利用可能なすべてのスクリプト、またはスクリプト ID が指定されている場合は特定のスクリプトを取得できます。manualinput が指定されている場合、{MANUALINPUT} マクロは指定された値に置き換えられます。

> **Note:**
> このメソッドは、あらゆるタイプのユーザーが利用できます。
> このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) このメソッドは、以下のパラメータを持つオブジェクト、またはオブジェクトの配列を受け付けます。

パラメータ	Type	説明
hostid	ID	スクリプトを返す対象のホストの ID。 一意である必要があります。 Parameter behavior: *- required*
scriptid	ID	返すスクリプトの ID。
manualinput	string	ユーザーが指定した {MANUALINPUT} マクロ値。

戻り値

(object) ホスト ID をプロパティ、利用可能なスクリプトの配列を値として持つオブジェクトを返します。スクリプト ID が指定されている場合、対応する値はその特定のスクリプトを含む配列になります。

> **Note:**
> このメソッドは、`confirmation` テキスト、`manualinput prompt` テキスト、および `url` 内のマクロを自動的に展開します。
> manualinput パラメータが指定されている場合、{MANUALINPUT} マクロは指定された値に解決されます。

例

ホスト ID でスクリプトを取得

ホスト「30079」および「30073」で利用可能なすべてのスクリプトを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
       }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": " オペレーティングシステムを検出",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
```

```
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Zabbix ページを開く",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": " ページ *UNKNOWN* を開いてもよろしいですか？",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": " 開く Zabbix ページ:",
            "manualinput_validator_type": "0",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
```

```
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ],
        "30073": [
            {
                "scriptid": "3",
                "name": " オペレーティングシステムを検出",
                "command": "sudo /usr/bin/nmap -O {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "7",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
                "url": "",
                "new_window": "1",
                "manualinput": "0",
                "manualinput_prompt": "",
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            },
            {
                "scriptid": "1",
                "name": "Ping",
                "command": "/bin/ping -c 3 {HOST.CONN} 2>&1",
                "host_access": "2",
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "",
                "type": "0",
                "execute_on": "1",
                "timeout": "30s",
                "scope": "2",
                "port": "",
                "authtype": "0",
                "username": "",
                "password": "",
                "publickey": "",
                "privatekey": "",
                "menu_path": "",
```

```
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
            "manualinput_validator_type": "0",
            "manualinput_validator": "",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "4",
            "name": "Zabbix ページを開く",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": " ページ *UNKNOWN* を開いてもよろしいですか？",
            "type": "6",
            "execute_on": "2",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
            "new_window": "1",
            "manualinput": "1",
            "manualinput_prompt": " 開く Zabbix ページ:",
            "manualinput_validator_type": "1",
            "manualinput_validator": "dashboard.view,discovery.view",
            "manualinput_default_value": "",
            "parameters": []
        },
        {
            "scriptid": "2",
            "name": "Traceroute",
            "command": "/usr/bin/traceroute {HOST.CONN} 2>&1",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "",
            "type": "0",
            "execute_on": "1",
            "timeout": "30s",
            "scope": "2",
            "port": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "menu_path": "",
            "url": "",
            "new_window": "1",
            "manualinput": "0",
            "manualinput_prompt": "",
```

```
                "manualinput_validator_type": "0",
                "manualinput_validator": "",
                "manualinput_default_value": "",
                "parameters": []
            }
        ]
    },
    "id": 1
}
```

manualinput 値を持つ特定のスクリプトの情報を取得します。

ホスト”30079” 上の ID ”4”、manualinput 値”dashboard.view” を持つスクリプトの情報を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "30079": [
        {
          "scriptid": "4",
          "name": "Open Zabbix page",
          "command": "",
          "host_access": "2",
          "usrgrpid": "0",
          "groupid": "0",
          "description": "",
          "confirmation": "Are you sure you want to open page dashboard.view?",
          "type": "6",
          "execute_on": "2",
          "timeout": "30s",
          "scope": "2",
          "port": "",
          "authtype": "0",
          "username": "",
          "password": "",
          "publickey": "",
          "privatekey": "",
          "menu_path": "",
          "url": "http://localhost/ui/zabbix.php?action=dashboard.view",
          "new_window": "1",
          "manualinput": "1",
          "manualinput_prompt": "Zabbix page to open:",
          "manualinput_validator_type": "1",
          "manualinput_validator": "dashboard.view,discovery.view",
          "manualinput_default_value": "",
          "parameters": []
        }
      ]
```

```
    },
    "id": 1
}
```

ソース

CScript::getScriptsByHosts() in *ui/include/classes/api/services/CScript.php*.

script.update

説明

`object script.update(object/array scripts)`

このメソッドでは、既存のスクリプトを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) スクリプトのプロパティが更新されます。

`scriptid` プロパティはスクリプトごとに定義する必要があり、すべてのプロパティはオプションです。渡されたプロパティのみが更新され、他のプロパティは変更されません。例外は `type` プロパティが 5（Webhook）からその他に変更された場合です。その場合は `parameters` プロパティはクリーンアップされます。

戻り値

(object) `scriptids` プロパティの下で更新されたスクリプトの ID を含むオブジェクトを返します。

例

スクリプトコマンドの変更

スクリプトのコマンドを「"/bin/ping -c 10 {HOST.CONN} 2>&1"」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

スクリプトコマンドを変更して、手動入力を追加します

スクリプトのコマンドを"/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1" に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
```

```
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "Specify the number of ICMP packets to send with the ping command",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CScript::update() in *ui/include/classes/api/services/CScript.php*.

タスク

このクラスは、タスク（設定の再読み込みを行わずにアイテムやローレベルディスカバリルールをチェックするなど）を処理するために設計されています。

オブジェクト参照:

- タスク
  - '今すぐ実行' リクエストオブジェクト
  - 'プロキシ設定の更新' リクエストオブジェクト
  - '診断情報' リクエストオブジェクト
    * 統計リクエストオブジェクト
    * 統計結果オブジェクト

利用可能なメソッド:

- task.create - 新しいタスクを作成
- task.get - タスクを取得

タスクオブジェクト

以下のオブジェクトは task API に直接関連しています。

タスク

タスクオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
taskid	ID	タスクの ID。 プロパティの動作: - 読み取り専用

プロパティ	型	説明
type	integer	タスクのタイプ。 指定可能な値: 1 - 診断情報; 2 - プロキシ設定の更新; 6 - 今すぐ実行。 Zabbix 7.4.3 以降、*Admin* および *User* タイプのユーザーは「今すぐ実行」タスクを作成できます。 `task.get` は常に"7"（タスク実行サマリー）を返すことに注意してください。 プロパティの動作: - 必須
status	integer	タスクのステータス。 指定可能な値: 1 - 新規タスク; 2 - 実行中のタスク; 3 - タスクは完了しています; 4 - タスクは期限切れです。 プロパティの動作: - 読み取り専用
clock	timestamp	タスクが作成された時刻。 プロパティの動作: - 読み取り専用
ttl	integer	タスクの有効期限が切れるまでの秒数。 プロパティの動作: - 読み取り専用
proxyid	ID	診断情報の統計が収集される対象のプロキシの ID。 プロパティの動作: - `type` が「診断情報」または「プロキシ設定の更新」に設定されている場合に サポート
request	object	タスクタイプに応じたタスクリクエストオブジェクト: 「今すぐ実行」タスクのオブジェクトは以下で詳しく説明しています; 「プロキシ設定の更新」タスクのオブジェクトは以下で詳しく説明しています; 「診断情報」タスクのオブジェクトは以下で詳しく説明しています。 プロパティの動作: - 必須
result	object	診断情報タスクの結果オブジェクト。 結果の準備がまだできていない場合は NULL を含むことがあります。 結果オブジェクトは以下で詳しく説明しています。 プロパティの動作: - 読み取り専用

「今すぐ実行」リクエストオブジェクト

「今すぐ実行」タスクのリクエストオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
itemid	ID	アイテムおよびローレベルディスカバリルールの ID。 Zabbix 7.4.3 以降、*Admin* および *User* タイプのユーザーは、*read-write* permission を持つホスト、または *read* permission を持ち、かつそのロールで `invoke_execute_now` action が有効になっているホスト上のアイテムに対して「今すぐ実行」を行うことができます。ローレベルディスカバリ（LLD）ルールについても、*Admin* タイプのユーザーに同様に適用されます。

「プロキシ設定の更新」リクエストオブジェクト

「プロキシ設定の更新」タスクリクエストオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
proxyids	array	プロキシ ID。

「診断情報」リクエストオブジェクト

診断情報タスクのリクエストオブジェクトには、以下のプロパティがあります。すべての種類のプロパティに対する統計リクエストオブジェクトについては、以下で詳しく説明しています。

プロパティ	Type	説明
historycache	object	履歴キャッシュの統計リクエスト。サーバーおよびプロキシで使用できます。
valuecache	object	アイテムキャッシュの統計リクエスト。サーバーで使用できます。
preprocessing	object	前処理マネージャーの統計リクエスト。サーバーおよびプロキシで使用できます。
alerting	object	アラートマネージャーの統計リクエスト。サーバーで使用できます。
lld	object	LLD マネージャーの統計リクエスト。サーバーで使用できます。

統計リクエストオブジェクト

統計リクエストオブジェクトは、サーバー/プロキシの内部プロセスについて、どの種類の情報を収集するかを定義するために使用されます。以下のプロパティがあります。

プロパティ	型	説明
stats	query	返される統計オブジェクトのプロパティ。 診断情報統計の各タイプで利用可能なフィールドの一覧は、以下で詳しく説明されています。 デフォルト: `extend` は利用可能なすべての統計フィールドを返します。
top	object	返される統計値をソートおよび制限するためのオブジェクト。 診断情報統計の各タイプで利用可能なフィールドの一覧は、以下で詳しく説明されています。 例: { “source.alerts”: 10 }

診断情報リクエストの各タイプで利用可能な統計フィールドの一覧

診断情報リクエストプロパティの各タイプについて、以下の統計フィールドを要求できます。

診断タイプ	利用可能なフィールド	説明
historycache	items	キャッシュされたアイテム数。
	values	キャッシュされた値の数。
	memory	共有メモリの統計情報（空き領域、使用中チャンク数、空きチャンク数、空きチャンクの最大サイズ）
	memory.data	履歴データキャッシュの共有メモリ統計情報。
	memory.index	履歴インデックスキャッシュの共有メモリ統計情報。
valuecache	items	キャッシュされたアイテム数。
	values	キャッシュされた値の数。

診断タイプ	利用可能なフィールド	説明
	memory	共有メモリの統計情報（空き領域、使用中チャンク数、空きチャンク数、空きチャンクの最大サイズ）
	mode	値キャッシュのモード。
preprocessing	values	キューに入っている値の数。
	preproc.values	前処理ステップを持つキュー内の値の数。
alerting	alerts	キューに入っているアラート数。
lld	rules	キューに入っているルール数。
	values	キューに入っている値の数。

各種診断情報リクエストで利用可能なソートフィールドの一覧

以下の統計フィールドを使用して、要求された情報のソートおよび制限を行うことができます。

診断タイプ	利用可能なフィールド	Type
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

統計結果オブジェクト

統計結果オブジェクトは、タスクオブジェクトの `result` フィールドで取得されます。

プロパティ	型	説明
status	integer	タスク結果のステータス。 指定可能な値: -1 - タスクの実行中にエラーが発生した。 0 - タスク結果が作成された。 プロパティの動作: - 読み取り専用
data	string/object	特定の診断情報タスクの統計リクエストオブジェクトに応じた結果。 タスクの実行中にエラーが発生した場合は、エラーメッセージ文字列が含まれます。

task.create

説明

`object task.create(object/array tasks)`

このメソッドでは、タスクを作成できます。

> **Note:**
> このメソッドは、すべてのユーザータイプで利用できます（*Admin* および *User* タイプのユーザーは Zabbix 7.4.3 以降）。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

`(object/array)` 作成するタスク。

このメソッドは、標準のタスクプロパティを持つタスクを受け付けます。

なお、「今すぐ実行」タスクを作成できるのは、次の種類のアイテム/ディスカバリルールに対してのみです。

- Zabbix エージェント（パッシブ）
- シンプルチェック

- SNMP エージェント（v1/v2/v3）
- Zabbix 内部
- 外部チェック
- データベースモニタ
- HTTP エージェント
- IPMI エージェント
- SSH エージェント
- TELNET エージェント
- JMX エージェント
- 計算
- 依存アイテム
- スクリプト
- ブラウザ

アイテム/ディスカバリルールのタイプが「依存アイテム」の場合、そのマスターアイテムも上記のいずれかのタイプである必要があります。

戻り値

(object) taskids プロパティの下で作成されたタスクの ID を含むオブジェクトを返します。アイテムおよびローレベルディスカバリルールごとに 1 つのタスクが作成されます。返される ID の順序は、渡された itemids の順序と一致します。

例

タスクの作成

アイテムおよびローレベルディスカバリルールに対して「今すぐ実行」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

2 つのプロキシに対して「プロキシ設定の更新」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

「診断情報」タスクを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 1,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "proxyid": 0
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
```

```
}
```

関連項目

- Task
- ' 今すぐ実行' リクエストオブジェクト
- ' プロキシ設定の更新' リクエストオブジェクト
- ' 診断情報' リクエストオブジェクト

ソース

CTask::create() in *ui/include/classes/api/services/CTask.php*.

task.get

説明

`integer/array task.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってタスクを取得できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
taskids	ID/array	指定された ID を持つタスクのみを返します。
output	query	これらのパラメータについては、リファレンス解説で説明されています。
preservekeys	boolean	

戻り値

(integer/array) オブジェクトの配列を返します

例

ID でタスクを取得

ID「1」のタスクに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "taskid": "1",
            "type": "7",
            "status": "3",
            "clock": "1601039076",
```

```
            "ttl": "3600",
            "proxyid": null,
            "request": {
                "alerting": {
                    "stats": [
                        "alerts"
                    ],
                    "top": {
                        "media.alerts": 10
                    }
                },
                "lld": {
                    "stats": "extend",
                    "top": {
                        "values": 5
                    }
                }
            },
            "result": {
                "data": {
                    "alerting": {
                        "alerts": 0,
                        "top": {
                            "media.alerts": []
                        },
                        "time": 0.000663
                    },
                    "lld": {
                        "rules": 0,
                        "values": 0,
                        "top": {
                            "values": []
                        },
                        "time": 0.000442
                    }
                },
                "status": "0"
            }
        }
    ],
    "id": 1
}
```


参照

- Task
- Statistic result object

ソース

CTask::get() in *ui/include/classes/api/services/CTask.php*.

ダッシュボード

このクラスはダッシュボードを操作するために設計されています。

オブジェクト参照:

- ダッシュボード
- ダッシュボードページ
    - ダッシュボードウィジェット
        * ダッシュボードウィジェットフィールド
- ダッシュボードユーザーグループ
- ダッシュボードユーザー

利用可能なメソッド:

- dashboard.create - 新しいダッシュボードを作成
- dashboard.delete - ダッシュボードを削除
- dashboard.get - ダッシュボードを取得
- dashboard.update - ダッシュボードを更新

Dashboard オブジェクト

次のオブジェクトは、dashboardAPI に直接関連しています。

ダッシュボード

ダッシュボードオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
dashboardid	ID	ダッシュボードの ID。 プロパティの動作: - 読み取り専用 - 更新操作では必須
name	string	ダッシュボードの名前。 プロパティの動作: - 作成操作では必須
userid	ID	ダッシュボードの所有者であるユーザーの ID。
private	integer	ダッシュボードの共有タイプ。 指定可能な値: 0 - パブリックダッシュボード; 1 - *(デフォルト)* プライベートダッシュボード。
display_period	integer	デフォルトのページ表示期間（秒単位）。 指定可能な値: 10, 30, 60, 120, 600, 1800, 3600。 デフォルト: 30。
auto_start	integer	スライドショーを自動開始します。 指定可能な値: 0 - スライドショーを自動開始しない; 1 - *(デフォルト)* スライドショーを自動開始する。

ダッシュボードページ

ダッシュボードページオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
dashboard_pageid	ID	ダッシュボードページの ID。 プロパティの動作: - 読み取り専用
name	string	ダッシュボードページ名。 デフォルト: 空文字列。
display_period	integer	ダッシュボードページの表示期間（秒単位）。 指定可能な値: 0、10、30、60、120、600、1800、3600。 デフォルト: 0（デフォルトのページ表示期間が使用されます）。
widgets	array	ダッシュボードウィジェットオブジェクトの配列。

ダッシュボードウィジェット

ダッシュボードウィジェットオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
widgetid	ID	ダッシュボードウィジェットの ID。 プロパティの動作: - 読み取り専用
type	string	ダッシュボードウィジェットのタイプ。 指定可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリーステータス; favgraphs - お気に入りグラフ; favmaps - お気に入りマップ; gauge - ゲージ; geomap - ジオマップ; graph - グラフ（クラシック）; graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホストの可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemhistory - アイテム履歴; itemnavigator - アイテムナビゲーター; item - アイテムの値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; problemhosts - 障害ホスト; problems - 障害; problemsbysv - 深刻度別の障害; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - 上位ホスト; topitems - 上位アイテム; toptriggers - 上位トリガー; trigover - トリガーの概要; url - URL; web - Web 監視。 プロパティの動作: - 必須
name	string	カスタムウィジェット名。
x	integer	ダッシュボードの左端からの水平方向の位置。 指定可能な値の範囲は 0 から 71 です。
y	integer	ダッシュボードの上端からの垂直方向の位置。 指定可能な値の範囲は 0 から 63 です。
width	integer	ウィジェットの幅。 指定可能な値の範囲は 1 から 72 です。
height	integer	ウィジェットの高さ。 指定可能な値の範囲は 1 から 64 です。
view_mode	integer	ウィジェットの表示モード。 指定可能な値: 0 - *(デフォルト)* デフォルトのウィジェット表示; 1 - ヘッダーを非表示;

プロパティ	型	説明
fields	array	ダッシュボードウィジェットフィールドオブジェクトの配列。 プロパティの動作: - ダッシュボードウィジェットフィールド の各ウィジェットを参照

ダッシュボードウィジェットフィールド

ダッシュボードウィジェットフィールドオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
type	integer	ウィジェットフィールドのタイプ。 指定可能な値: 0 - 整数; 1 - 文字列; 2 - ホストグループ; 3 - ホスト; 4 - アイテム; 5 - アイテムのプロトタイプ; 6 - グラフ; 7 - グラフのプロトタイプ; 8 - マップ; 9 - サービス; 10 - SLA; 11 - ユーザー; 12 - アクション; 13 - メディアタイプ。 プロパティの動作: - 必須
name	string	ウィジェットフィールド名。 指定可能な値: ダッシュボードウィジェットフィールドを参照してください。 プロパティの動作: - 必須
value	mixed	タイプに応じたウィジェットフィールドの値。 指定可能な値: ダッシュボードウィジェットフィールドを参照してください。 プロパティの動作: - 必須

ダッシュボードユーザーグループ

ユーザーグループに基づくダッシュボード権限の一覧です。以下のプロパティがあります。

プロパティ	Type	説明
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 必須

プロパティ	Type	説明
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

ダッシュボードユーザー

ユーザーに基づくダッシュボード権限の一覧です。以下のプロパティがあります。

プロパティ	Type	説明
userid	ID	ユーザーの ID。 Property behavior: - *required*
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み取り/書き込み。 Property behavior: - *required*

dashboard.create

説明

`object dashboard.create(object/array dashboards)`

このメソッドは、新しいダッシュボードを作成することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 作成するダッシュボード。

標準のダッシュボードプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
pages	array	ダッシュボード用に作成するダッシュボードページ。ダッシュボードページは、指定された順序と同じ順序で並べられます。 パラメータの動作: - 必須
users	array	ダッシュボード上に作成するダッシュボードユーザー共有。
userGroups	array	ダッシュボード上に作成するダッシュボードユーザーグループ共有。

戻り値

(`object`) `dashboardids` プロパティの下に作成されたダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたダッシュボードの順序と一致します。

例

ダッシュボードの作成

単一のダッシュボードページで、タグ付きの障害ウィジェットと 2 種類の共有 (ユーザーグループとユーザー) を持つ、”My dashboard” という名前のダッシュボードを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "service"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "zabbix_server"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- ダッシュボードページ
- ダッシュボードウィジェット
- ダッシュボードウィジェットフィールド
- ダッシュボードユーザー
- ダッシュボードユーザーグループ

ソース

CDashboard::create() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

説明

`object dashboard.delete(array dashboardids)`

このメソッドは、ダッシュボードを削除することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するダッシュボードの ID。

戻り値

(object) dashboardids プロパティの下にある削除されたダッシュボードの ID を含むオブジェクトを返します。

例

複数のダッシュボードの削除

2 つのダッシュボードを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2",
            "3"
        ]
    },
```

```
    "id": 1
}
```

ソース

CDashboard::delete() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

説明

`integer/array dashboard.get(object parameters)`

このメソッドは、指定したパラメーターに応じたダッシュボードを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dashboardids	ID/array	指定した ID のダッシュボードのみを返します。
selectPages	query	ダッシュボードページを正しい順序で含む`pages` プロパティを返します。
selectUsers	query	ダッシュボードが共有されているユーザーを含む`users` プロパティを返します。
selectUserGroups	query	ダッシュボードが共有されているユーザーグループを含む`userGroups` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dashboardid`。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ID によるダッシュボードの取得

ダッシュボード"1" と"2" に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
```

```
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dashboardid": "1",
            "name": "Dashboard",
            "userid": "1",
            "private": "0",
            "display_period": "30",
            "auto_start": "1",
            "users": [],
            "userGroups": [],
            "pages": [
                {
                    "dashboard_pageid": "1",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "9",
                            "type": "systeminfo",
                            "name": "",
                            "x": "12",
                            "y": "8",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "8",
                            "type": "problemsbysv",
                            "name": "",
                            "x": "12",
                            "y": "4",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
                        {
                            "widgetid": "7",
                            "type": "problemhosts",
                            "name": "",
                            "x": "12",
                            "y": "0",
                            "width": "12",
                            "height": "4",
                            "view_mode": "0",
                            "fields": []
                        },
```

```
                {
                    "widgetid": "6",
                    "type": "discovery",
                    "name": "",
                    "x": "6",
                    "y": "9",
                    "width": "18",
                    "height": "4",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "5",
                    "type": "web",
                    "name": "",
                    "x": "0",
                    "y": "9",
                    "width": "18",
                    "height": "4",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "4",
                    "type": "problems",
                    "name": "",
                    "x": "0",
                    "y": "3",
                    "width": "12",
                    "height": "6",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "3",
                    "type": "favmaps",
                    "name": "",
                    "x": "8",
                    "y": "0",
                    "width": "12",
                    "height": "3",
                    "view_mode": "0",
                    "fields": []
                },
                {
                    "widgetid": "1",
                    "type": "favgraphs",
                    "name": "",
                    "x": "0",
                    "y": "0",
                    "width": "12",
                    "height": "3",
                    "view_mode": "0",
                    "fields": []
                }
            ]
        },
        {
            "dashboard_pageid": "2",
            "name": "",
            "display_period": "0",
            "widgets": []
```

```
                },
                {
                    "dashboard_pageid": "3",
                    "name": "Custom page name",
                    "display_period": "60",
                    "widgets": []
                }
            ]
        },
        {
            "dashboardid": "2",
            "name": "My dashboard",
            "userid": "1",
            "private": "1",
            "display_period": "60",
            "auto_start": "1",
            "users": [
                {
                    "userid": "4",
                    "permission": "3"
                }
            ],
            "userGroups": [
                {
                    "usrgrpid": "7",
                    "permission": "2"
                }
            ],
            "pages": [
                {
                    "dashboard_pageid": "4",
                    "name": "",
                    "display_period": "0",
                    "widgets": [
                        {
                            "widgetid": "10",
                            "type": "problems",
                            "name": "",
                            "x": "0",
                            "y": "0",
                            "width": "12",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "2",
                                    "name": "groupids",
                                    "value": "4"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- ダッシュボードページ
- ダッシュボードウィジェット

- ダッシュボードウィジェットフィールド
- ダッシュボードユーザー
- ダッシュボードユーザーグループ

ソース

CDashboard::get() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

説明

`object dashboard.update(object/array dashboards)`

このメソッドは、既存のダッシュボードを更新することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するダッシュボードのプロパティ。

各ダッシュボードでは `dashboardid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のダッシュボードプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
pages	array	既存のダッシュボードページを置き換えるダッシュボードページ。 ダッシュボードページは `dashboard_pageid` プロパティによって更新されます。`dashboard_pageid` プロパティを持たないオブジェクトについては新しいダッシュボードページが作成され、再利用されなかった既存のダッシュボードページは削除されます。ダッシュボードページは、指定された順序と同じ順序で並べられます。ダッシュボードページのうち、指定されたプロパティのみが更新されます。
users	array	既存の要素を置き換えるダッシュボードユーザー共有。
userGroups	array	既存の要素を置き換えるダッシュボードユーザーグループ共有。

戻り値

(`object`) `dashboardids` プロパティの下にある更新されたダッシュボードの ID を含むオブジェクトを返します。

例

ダッシュボードの名前変更

ダッシュボードの名前を”SQL server status” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

ダッシュボードページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換えて、3 番目のページとして新しいページを追加します。他のすべてのダッシュボードページを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

ダッシュボードの所有者の変更

Admin と super admin のみが使用できます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
```

```
        "userid": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- ダッシュボードページ
- ダッシュボードウィジェット
- ダッシュボードウィジェットフィールド
- ダッシュボードユーザー
- ダッシュボードユーザーグループ

ソース

CDashboard::update() in *ui/include/classes/api/services/CDashboard.php*.

Dashboard widget fields

このページには、ダッシュボードウィジェットパラメーターのナビゲーションリンクと、それぞれのダッシュボードウィジェットフィールドオブジェクトの可能なプロパティ値が含まれています。

各ウィジェットのパラメーターとプロパティ値を確認するには、個別のページに移動してください:

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

- Web monitoring

Attention:
非推奨のウィジェットは、今後のメジャーリリースで削除される予定です。

1 Action log

説明

これらのパラメータおよび対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、`dashboard.create` および `dashboard.update` メソッドで*Action log* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、その一方で、ウィジェットを誤って作成または更新してしまうリスクも生じます。*Action log* ウィジェットを正常に作成または更新するには、以下の表に示されているパラメータの動作を参照してください。

パラメータ

以下のパラメータは、アクションログウィジェットでサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
受信者	11	userids.0	ユーザー ID。 注意: 複数のユーザーを設定するには、プロパティ名の番号を増やして、ユーザーごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
アクション	12	actionids.0	アクション ID。 注意: 複数のアクションを設定するには、プロパティ名の番号を増やして、アクションごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
メディアタイプ	13	mediatypeids.0	メディアタイプ ID。 注意: 複数のメディアタイプを設定するには、プロパティ名の番号を増やして、メディアタイプごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
ステータス	0	statuses.0	0 - 処理中; 1 - 送信済み/実行済み; 2 - 失敗。 注意: 複数の値を設定するには、プロパティ名の番号を増やして、値ごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。

パラメータ		type	name	value
検索文字列		1	message	任意の文字列値。
期間		1	time_period._referenceDA	SHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータが"ABCDE"に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメータでのみ期間を設定することもできます。
	開始	1	time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.to が設定されている場合は必須
	終了	1	time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.from が設定されている場合は必須
エントリの並べ替え		0	sort_triggers	3 - 時刻（昇順）; 4 - *(デフォルト)* 時刻（降順）; 5 - タイプ（昇順）; 6 - タイプ（降順）; 7 - ステータス（昇順）; 8 - ステータス（降順）; 11 - 受信者（昇順）; 12 - 受信者（降順）。
表示行数		0	show_lines	指定可能な値の範囲は 1～100 です。 デフォルト: 25。

例

次の例は、アクションログウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アクションログウィジェットの設定

時間でソート (昇順) したアクション操作の詳細の 10 のエントリを表示するアクションログウィジェットを設定します。さらに、ユーザー"1" に電子メールを送信しようとしたが失敗したアクション操作のみの詳細を表示します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
```

```
        "pages": [
            {
                "widgets": [
                    {
                        "type": "actionlog",
                        "name": "Action log",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 10
                            },
                            {
                                "type": 0,
                                "name": "sort_triggers",
                                "value": 3
                            },
                            {
                                "type": 11,
                                "name": "userids.0",
                                "value": 1
                            },
                            {
                                "type": 13,
                                "name": "mediatypeids.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "statuses.0",
                                "value": 2
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

2 Clock

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで時計ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。時計ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

時計ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分
時間の形式	0	time_type	0 - *(デフォルト)* ローカル時間 1 - サーバー時間 2 - ホスト時間
時計のタイプ	0	clock_type	0 - *(デフォルト)* アナログ 1 - デジタル

時間の形式が” ホスト時間” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
アイテム	4	itemid.0	アイテムID。 パラメーターの動作: - 必須

時計のタイプが” デジタル” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
表示	0	show.0	1 - 日付 2 - *(デフォルト)* 時刻 3 - タイムゾーン 注意: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。

詳細設定

時計のタイプが" デジタル" に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
背景色	1	bg_color	16 進数の色コード (例: FF0000)。 デフォルト: "" (空)。

日付

時計のタイプが" デジタル" に設定され、表示が" 日付" に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	date_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	date_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。

時間

時計のタイプが" デジタル" に設定され、表示が" 時刻" に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	time_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	time_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。
秒	0	time_sec	0 - 無効 1 - *(デフォルト)* 有効
形式	0	time_format	0 - *(デフォルト)* 24 時間 1 - 12 時間

タイムゾーン

時計のタイプが" デジタル" に設定され、表示が" タイムゾーン" に設定されている場合、以下の詳細設定パラメーターがサポートされます。

パラメーター	type	name	value
太字	0	tzone_bold	0 - *(デフォルト)* 無効 1 - 有効
色	1	tzone_color	16 進数のカラーコード (例: FF0000)。 デフォルト: "" (空)。

パラメーター	type	name	value
タイムゾーン	1	tzone_timezone	有効なタイムゾーン文字列 (例: Europe/Riga、system、UTC など)。サポートされているタイムゾーンの完全なリストについては、PHP ドキュメントを参照してください。 デフォルト: local。 パラメーターの動作: - 時間の形式が" ローカル時間" または" サーバー時間" に設定されている場合はサポート
フォーマット	0	tzone_format	0 - (デフォルト) 短縮形式 1 - 完全形式 パラメーターの動作: - 時間の形式が" ローカル時間" または" サーバー時間" に設定されている場合はサポート

例

次の例は、時計ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

時計ウィジェットの設定

カスタマイズされたデジタル時計にローカルの日付、時間、およびタイムゾーンを表示する時計ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "clock",
                        "name": "Clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "clock_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
```

```
                                "name": "show.2",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "date_color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 0,
                                "name": "time_bold",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tzone_color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 1,
                                "name": "tzone_timezone",
                                "value": "Europe/Riga"
                            },
                            {
                                "type": 0,
                                "name": "tzone_format",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field

- dashboard.create
- dashboard.update

3 Discovery status

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでディスカバリのステータスウィジェットを設定できます。

パラメーター

ディスカバリのステータスウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

例

次の例は、ディスカバリのステータスウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ディスカバリのステータスウィジェットの設定

ディスカバリのステータスウィジェットを更新間隔を 15 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "discovery",
                        "name": "Discovery status",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 900
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
```

```
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

4 Favorite graphs

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでお気に入りのグラフウィジェットを設定できます。

パラメーター

お気に入りのグラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分

例

次の例は、お気に入りのグラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

お気に入りのグラフウィジェットの設定

お気に入りのグラフウィジェットを更新間隔を 10 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favgraphs",
                        "name": "Favorite graphs",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

5 Favorite maps

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでお気に入りのマップウィジェットを設定できます。

パラメーター

お気に入りのマップウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分

例

次の例は、お気に入りのマップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

お気に入りのマップウィジェットの設定

お気に入りのマップウィジェットを更新間隔を 10 分で設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "favmaps",
                        "name": "Favorite maps",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
```

```
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

6 Gauge

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでゲージウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ゲージウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

Gauge ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
Item	4	itemid.0	Item ID パラメーターの動作: - *Item (Widget)* が設定されていない場合は 必須です。

パラメーター		type	name	value
	Item (Widget)	1	itemid._reference	Item ID: ABCDE._itemid の代わりに、アイテムのデータソースとして互換性のあるウィジェット (*Reference* パラメーターが"ABCDE に設定されている) を設定します。 パラメーターの動作: - *Item* が設定されていない場合は 必須です。
最小値		1	min	任意の数値サフィックス (例: "1d", "2w", "4K", "8G") がサポートされています。 デフォルト: "0"
最大値		1	max	任意の数値サフィックス (例: "1d", "2w", "4K", "8G") がサポートされています。 デフォルト: "100"
値の円弧		1	value_arc_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)
円弧の背景		1	empty_color	16 進数カラーコード（例：FF0000） デフォルト："" (空)
背景		1	bg_color	16 進数カラーコード（例：FF0000） デフォルト："" (空)
表示		0	表示.0	1 - 説明; 2 - 値; 3 - 針; 4 - スケール; 5 - 値の円弧. 注: 複数の値を構成するには、プロパティ名に増分番号を付けて、各値のダッシュボードウィジェットフィールドオブジェクトを作成します。 デフォルト: 1、2、4、5 値「針」と「スケール」は、次の両方の場合にはサポートされません。 - 値が「値の円弧」の 表示のダッシュボードウィジェットフィールドオブジェクトが設定されていない。 - 表示円弧詳細設定 パラメータが「無効」に設定されている。 対応する値が設定されたダッシュボードウィジェットフィールドオブジェクトが設定されていない場合、表示オプションの詳細設定 パラメータはサポートされません。
上書きするホスト		1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット (その *Reference* パラメータを"ABCDE に設定) をホストのデータソースとして設定します。 DASHBOARD._hostid - ダッシュボードのホストセレクタ をホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合にはサポートされません。

Advanced configuration

Gauge ウィジェットでは、以下の詳細設定パラメータがサポートされています。

Note:
Thresholds プロパティ名（例：thresholds.0.color）の数字は、リスト内のしきい値の位置を示し、昇順でソートされます。ただし、しきい値が異なる順序で設定されている場合は、Zabbix フロントエンドでウィジェット設定を更新すると、値は昇順でソートされます。(例："thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5")

パラメータ	type	名前	値
角度	0	角度	指定可能な値: 180 *(デフォルト)* または 270
説明			
説明	1	説明	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ デフォルト: {ITEM.NAME}
サイズ	0	desc_size	指定可能な値の範囲は 1-100 です。 デフォルト: 15
垂直位置	0	desc_v_pos	0 - 上; 1 - *(デフォルト)* 下
太字	0	desc_bold	0 - *(デフォルト)* 無効; 1 - 有効
色	1	desc_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)
値			
小数点	0	decimal_places	指定可能な値の範囲は 1-10 です。 デフォルト: 2
サイズ	0	value_size	指定可能な値の範囲は 1-100 です。 デフォルト: 25
太字	0	value_bold	0 - *(デフォルト)* 無効; 1 - 有効
Color	1	value_color	16 進数のカラーコード（例：FF0000） デフォルト："" （空）
単位			
単位（チェックボックス）	0	units_show	0 - 無効 1 - （デフォルト）有効
単位（値）	1	units	任意の文字列値 パラメータの動作: - 単位（チェックボックス）が” 有効” に設定されている場合にサポートされます
サイズ	0	units_size	有効な値の範囲は 1-100 です。 デフォルト: 25 パラメータの動作: - 単位 (チェックボックス) が” 有効” に設定されている場合にサポートされます
太字	0	units_bold	0 - *(デフォルト)* 無効 1 - 有効 パラメータの動作: - 単位 (チェックボックス) が” 有効” に設定されている場合にサポートされます

パラメータ	type	名前	値
位置	0	units_pos	0 - 値の前 1 - 値の上 2 - *(デフォルト)* 値の後 3 - 値の下 パラメータの動作: - 単位 (チェックボックス) が" 有効" に設定されている場合に サポートされます。 このパラメータは、次の時間関連の単位 のいずれかに設定されている場合、無視されます。: unixtime、uptime、s
色	1	units_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)
値の円弧			
円弧サイズ	0	value_arc_size	有効な値の範囲は 1-100 です。 デフォルト: 20
ニードル			
色	1	needle_color	16 進数のカラーコード（例: FF0000） デフォルト: ""（空） パラメータの動作: - ダッシュボードウィジェットのフィールドオブジェクトで、表示に" 値円弧" が設定されている場合、または表示円弧が" 有効" に設定されている場合にサポートされます。
スケール			
単位の表示	0	scale_show_units	0 - 無効; 1 - *(デフォルト)* 有効 パラメータの動作: - 単位 (チェックボックス) が" 有効" に設定され、ダッシュボードウィジェットの 表示のフィールドオブジェクトで値が" 値の円弧" に設定されているか、表示円弧が" 有効" に設定されている場合に サポートされます。
サイズ	0	scale_size	指定可能な値の範囲は 1-100 です。 デフォルト: 15 パラメータの動作: - 表示のダッシュボードウィジェットのフィールドオブジェクトで値が" 値の円弧" に設定されているか、表示円弧が" 有効" に設定されている場合に サポートされます。
小数点以下の桁数	0	scale_decimal_places	有効な値の範囲は 1-10 です。 デフォルト: 0 パラメータの動作: - ダッシュボードウィジェットのフィールドオブジェクトで、表示に"Value arc" が設定されている場合、または表示 *arc* が" 有効" に設定されている場合にサポートされます。
しきい値			
色	1	thresholds.0.color	16 進数のカラーコード (例: FF0000)
しきい値	1	thresholds.0.threshold	任意の数値サフィックス (例: "1d"、"2w"、"4K"、"8G") がサポートされています。
ラベルを表示	0	th_show_labels	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - しきい値が設定されており、ダッシュボードウィジェットの 表示フィールドオブジェクトに値"Value arc" が設定されているか、表示アークが" 有効" に設定されている場合は サポートされます。

パラメータ	type	名前	値
円弧 'を表示	0	th_show_arc	0 - (デフォルト) 無効; 1 - 有効
円弧のサイズ	0	th_arc_size	パラメータの動作: - しきい値が設定されている場合は サポートされます。 可能な値の範囲は 1-100 です。 デフォルト: 5 パラメータの動作: - 円弧の表示が" 有効" に設定されている場合は サポートされます。

例

次の例は、ゲージウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ゲージウィジェットの設定

アイテム"44474" (インターフェース enp0s3: 送信ビット数) の値を表示するゲージウィジェットを設定します。さらに、しきい値などの複数の詳細オプションを使用して、ウィジェットを視覚的に微調整します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "gauge",
                        "name": "Gauge",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 44474
                            },
                            {
                                "type": 1,
                                "name": "min",
                                "value": "100000"
                            },
                            {
                                "type": 1,
                                "name": "max",
                                "value": "1000000"
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
```

```
                {
                    "type": 0,
                    "name": "show.1",
                    "value": 2
                },
                {
                    "type": 0,
                    "name": "show.2",
                    "value": 3
                },
                {
                    "type": 0,
                    "name": "show.4",
                    "value": 4
                },
                {
                    "type": 0,
                    "name": "show.5",
                    "value": 5
                },
                {
                    "type": 0,
                    "name": "angle",
                    "value": 270
                },
                {
                    "type": 0,
                    "name": "desc_size",
                    "value": 10
                },
                {
                    "type": 0,
                    "name": "desc_bold",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "decimal_places",
                    "value": 0
                },
                {
                    "type": 0,
                    "name": "value_bold",
                    "value": 1
                },
                {
                    "type": 0,
                    "name": "units_size",
                    "value": 15
                },
                {
                    "type": 0,
                    "name": "units_pos",
                    "value": 3
                },
                {
                    "type": 1,
                    "name": "needle_color",
                    "value": "3C3C3C"
                },
                {
                    "type": 1,
```

```
                                "name": "thresholds.0.color",
                                "value": "FF465C"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "700000"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "FFD54F"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "500000"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.color",
                                "value": "0EC9AC"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.threshold",
                                "value": "100000"
                            },
                            {
                                "type": 0,
                                "name": "th_show_labels",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "th_show_arc",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "th_arc_size",
                                "value": 15
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
```

```
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

7 Geomap

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで地理マップウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。地理マップウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

Geomap ウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメータ		type	name	value
	ホストグループ (ウィジェット)	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
タグ				
	評価タイプ	0	evaltype	0 - (デフォルト) And/Or; 2 - Or. このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - タグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - タグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の数字は、タグ評価リスト内のタグ順序を参照します。 パラメータの動作: - タグを構成する場合は 必須 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメータ	type	name	value
初期ビューパラメータ	1	default_view	カンマ区切りの 緯度、経度、ズームレベル (オプション、可能な値の範囲は 0-30) 例: 40.6892494,-74.0466891,10
参照	1	参照	5 文字からなる任意の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、地理マップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

地理マップウィジェットの設定

以下のタグ設定に基づいて、ホストグループ”2” と”22” のホストを表示する地理マップウィジェットを設定します。”component” という名前のタグに”node” という値が含まれる、または”location” という名前のタグに”New York” という値と等しいもの。さらに、マップの初期表示を座標”40.6892494”(緯度)、”-74.0466891”(経度)、ズームレベル”10” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "geomap",
                        "name": "Geomap",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 22
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "default_view",
                                "value": "40.6892494,-74.0466891,10"
                            },
                            {
```

```
                                "type": 0,
                                "name": "evaltype",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "node"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "location"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "New York"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
```

```
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

8 Graph

説明

これらのパラメータおよび各ダッシュボードウィジェットフィールドオブジェクトに対応するプロパティ値を使用すると、dashboard.create および dashboard.update メソッドで*Graph* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、一方でウィジェットを誤って作成または更新してしまうリスクも生じます。*Graph* ウィジェットを正常に作成または更新するには、以下の表に示されているパラメータの動作を参照してください。

パラメーター

グラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分
参照	1	reference	5 文字で構成される文字列値 (例: ABCDE または JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意でなければなりません。 パラメーターの動作: - 必須

データセット

以下のパラメータは、データセットの設定でサポートされています。

Note:
プロパティ名の最初の数字（例: ds.0.hosts.0、ds.0.items.0）は特定のデータセットを表し、2 番目の数字が存在する場合は、設定されたホストまたはアイテムを表します。

Parameter	type	name	value
データセットタイプ	0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。

Parameter	type	name	value
アイテム	4	ds.0.itemids.0	Item ID。 テンプレートダッシュボード でウィジェットを設定する場合は、テンプレート上で設定されたアイテムのみを指定する必要があります。 注意: 複数のアイテムを設定するには、プロパティ名の番号を増やして、各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 Parameter behavior: - データセットタイプが「アイテムリスト」に設定され、アイテム（ウィジェット）が設定されていない場合は 必須
アイテム（ウィジェット）	1	ds.0.itemids.0._reference	Item ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（*Reference* パラメータが"ABCDE" に設定されているもの）を、アイテムのデータソースとして設定します。 注意: 複数のウィジェットを設定するには、プロパティ名の番号を増やして、各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 Parameter behavior: - データセットタイプが「アイテムリスト」に設定され、アイテムが設定されていない場合は 必須
色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 Parameter behavior: - データセットタイプが「アイテムリスト」に設定されている場合は 必須
ホストパターン	1	ds.0.hosts.0	Host 名またはパターン（例: "Zabbix*"）。 Parameter behavior: - データセットタイプが「アイテムパターン」に設定されている場合は 必須 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
アイテムパターン	1	ds.0.items.0	Item 名またはパターン（例: "*: 1 秒あたりに処理された * 値の数"）。 テンプレートダッシュボード でウィジェットを設定する場合は、テンプレート上で設定されたアイテムのパターンのみを指定する必要があります。 Parameter behavior: - データセットタイプが「アイテムパターン」に設定されている場合は 必須
色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 Parameter behavior: - データセットタイプが「アイテムパターン」に設定され、カラーパレットが設定されていない場合に サポート
カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 指定可能な値は 0～11 です。 デフォルト: 0。 Parameter behavior: - データセットタイプが「アイテムパターン」に設定され、色が設定されていない場合に サポート

Parameter	type	name	value
描画	0	ds.0.type	0 - *(デフォルト)* 線; 1 - 点; 2 - 階段状; 3 - 棒。
積み重ね	0	ds.0.stacked	0 - *(デフォルト)* 無効; 1 - 有効。 Parameter behavior: - 描画が「線」、「階段状」、または「棒」に設定されている場合に サポート
幅	0	ds.0.width	指定可能な値は 1～10 です。 デフォルト: 1。 Parameter behavior: - 描画が「線」または「階段状」に設定されている場合に サポート
ポイントサイズ	0	ds.0.pointsize	指定可能な値は 1～10 です。 デフォルト: 3。 Parameter behavior: - 描画が「点」に設定されている場合に サポート
透明度	0	ds.0.transparency	指定可能な値は 1～10 です。 デフォルト: 5。
塗りつぶし	0	ds.0.fill	指定可能な値は 1～10 です。 デフォルト: 3。 Parameter behavior: - 描画が「線」または「階段状」に設定されている場合に サポート
欠損データ	0	ds.0.missingdatafunc	0 - *(デフォルト)* なし; 1 - 接続; 2 - 0 として扱う; 3 - 最後の既知の値。 Parameter behavior: - 描画が「線」または「階段状」に設定されている場合に サポート
ホストを上書き	1	ds.0.override_hostid.ABCDE.reference	ABCDE - *hostid* - 互換性のあるウィジェット（その”Reference” パラメータが”ABCDE” に設定されているもの）を、ホストのデータソースとして設定します。 DASHBOARD._hostid - ダッシュボードのホストセレクターを、ホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
Y 軸	0	ds.0.axisy	0 - *(デフォルト)* 左; 1 - 右。
タイムシフト	1	ds.0.timeshift	有効な時間文字列（例: 3600、1h など）。 時間サフィックス を使用できます。負の値も使用可能です。 デフォルト: "" （空）。

Parameter	type	name	value
集計関数	0	ds.0.aggregate_function	0 - *(デフォルト)* 未使用; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last。
集計間隔	1	ds.0.aggregate_interval	有効な時間文字列（例: 3600、1h など）。 時間サフィックス を使用できます。 デフォルト: 1h。
集計	0	ds.0.aggregate_grouping	0 - *(デフォルト)* 各アイテム; 1 - データセット。 Parameter behavior: - 集計関数が「min」、「max」、「avg」、「count」、「sum」、「first」、または「last」に設定されている場合に サポート
近似	0	ds.0.approximation	1 - min; 2 - *(デフォルト)* avg; 4 - max; 7 - all。
データセットラベル	1	ds.0.data_set_label	任意の文字列値。 デフォルト: ""（空）。

Display options

表示オプションの設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
履歴データの選択	0	source	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド
シンプルトリガー	0	simple_triggers	0 - *(デフォルト)* 無効; 1 - 有効
稼働時間	0	working_time	0 - *(デフォルト)* 無効; 1 - 有効

パラメータ	type	name	value
パーセンタイル線 (左)			
ステータス	0	percentile_left	0 - *(デフォルト)* 無効; 1 - 有効 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 左" に設定されている場合に サポートされます。
値	0	percentile_left_value	指定可能な値の範囲は 1-100 です。 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 左" に設定されている場合に サポートされます。
パーセンタイル線 *(右)*			
ステータス	0	percentile_right	0 - *(デフォルト)* 無効; 1 - 有効 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 右" に設定されている場合に サポートされます。
値	0	percentile_right_value	可能な値の範囲は 1-100 です。 パラメーターの動作: - *Y* 軸 (データセット 構成内) が" 右" に設定されている場合に サポートされます。

期間

以下のパラメータは、期間の設定でサポートされています。

パラメータ	type	name	value
期間	1	time_period._reference	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（*Reference* パラメータが「ABCDE」に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定することもできます。
From	1	time_period.from	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合は *supported* - time_period.to が設定されている場合は *required*
To	1	time_period.to	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合は *supported* - time_period.from が設定されている場合は *required*

Axes

軸の設定には、以下のパラメーターがサポートされています。

パラメーター	type	name	value
左 *Y* 軸	0	lefty	0 - 無効 1 - *(デフォルト)* 有効 パラメーターの動作: - *Y* 軸 (データセット 設定内) が" 左" に設定されている場合に サポートされます。
右 *Y*	0	righty	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - *Y* 軸 (データセット 設定内) が" 右" に設定されている場合に サポートされます。
スケール	0	lefty_scale righty_scale	0 - *(デフォルト)* 線形; 1 - 対数
最小値	1	lefty_min righty_min	任意の数値 デフォルト: "" (空)
最大値	1	lefty_max righty_max	任意の数値 デフォルト: "" (空)
単位 (タイプ)	0	lefty_units righty_units	0 - *(デフォルト)* 自動; 1 - 静的
単位 (値)	1	lefty_static_units	任意の文字列値。 デフォルト: "" (空)

パラメーター	type	name	value
X 軸	0	righty_static_units xaxis	0 - 無効; 1 - *(*デフォルト*)* 有効

Legend

凡例の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
凡例の表示	0	凡例	0 - 無効 1 - *(*デフォルト*)* 有効
最小値/平均値/最大値を表示	0	legend_statistic	パラメータの動作: - 凡例の表示が" 有効" に設定されている場合にサポートされます。 0 - *(*デフォルト*)* 無効; 1 - 有効
集計関数の表示	0	legend_aggregation	パラメータの動作: - 凡例の表示が" 有効" に設定されている場合にサポートされます。 0 - *(*デフォルト*)* 無効; 1 - 有効
行数	0	legend_lines_mode	パラメータの動作: - 凡例の表示が" 有効" に設定されている場合にサポートされます。 0 - *(*デフォルト*)* 固定 1 - 可変
行数/ 最大行数	0	legend_lines	パラメータの動作: - 凡例の表示が" 有効" に設定されている場合に サポートされます。 指定可能な値の範囲は 1-10 です。 デフォルト: 1
列数	0	legend_columns	パラメータの動作: - 凡例の表示が" 有効" に設定されている場合に サポートされます 可能な値の範囲は 1〜4 です。 デフォルト: 4
			パラメータの動作: - 凡例の表示が" 有効" に設定され、最小値/平均値/最大値の表示が" 無効" に設定されている場合にサポートされます。

Problems

問題の設定には、以下のパラメータがサポートされています。

パラメータ	type	name	value
問題を表示	0	show_problems	0 - *(*デフォルト*)* 無効; 1 - *(*デフォルト*)* 有効

パラメータ		type	name	value
選択された項目のみ		0	graph_item_problems	0 - (デフォルト) 有効
問題のあるホスト		1	problemhosts.0	ホスト 名 注: プロパティ名の数字は、設定されたホストを参照します。 複数のホストを設定するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを設定する場合、サポートされません。
重大度		0	severities.0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効) 注: 複数の値を設定するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトを値ごとに作成します。
問題		1	problem_name	問題イベント名 (大文字と小文字は区別されません。完全な名前またはその一部)
問題タグ				
	評価タイプ	0	evaltype	0 - (デフォルト) And/Or; 2 - Or
	タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 問題タグを構成する場合は 必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 問題のあるタグを構成する場合は必須

パラメータ	type	name	value
タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - 問題のあるタグを構成する場合は必須

Overrides

オーバーライドの設定には、以下のパラメーターがサポートされています。

Note:
プロパティ名の最初の数字（例: or.0.hosts.0、or.0.items.0）は特定のデータセットを表し、2 番目の数字は（存在する場合）設定されたホストまたはアイテムを表します。

パラメーター	type	名前	値
ホストパターン	1	または.0.hosts.0	ホスト の名前またはパターン (例: Zabbix*) このパラメーターは、テンプレートダッシュボード でウィジェットを設定する場合、サポートされません。 パラメーターの動作: - オーバーライドを設定する場合は 必須
アイテムパターン	1	または.0.items.0	アイテム の名前またはパターン (例: *: 1 秒あたりに処理される * 値の数) テンプレートダッシュボード でウィジェットを設定する場合は、テンプレートで設定されているアイテムのパターンのみを設定する必要があります。 パラメーターの動作: - オーバーライドを設定する場合は 必須
基本色	1	または.0.color	16 進数カラーコード (例: FF0000)
幅	0	または.0.width	指定可能な値の範囲は 1-10 です。
描画	0	or.0.type	0 - 線; 1 - ポイント; 2 - 階段; 3 - 棒グラフ
透明度	0	or.0.transparency	有効な値の範囲は 1-10 です。
塗りつぶし	0	or.0.fill	有効な値の範囲は 1-10 です。
ポイントサイズ	0	or.0.pointsize	有効な値の範囲は 1-10 です。
欠損データ	0	or.0.missingdatafunc	0 - なし; 1 - 接続済み; 2 - 0 として処理; 3 - 既知の最後
Y 軸	0	or.0.axisy	0 - 左; 1 - 右
時間シフト	1	または.0.timeshift	有効な時間文字列（例：3600、1h など） 時間接尾辞を使用できます。負の値も使用できます。

例

次の例は、グラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフウィジェットの設定

次のようにグラフウィジェットを設定します。

- 1 つのホスト上の合計 9 個のアイテムに対して、2 つのデータセットを設定します。
- 1 つ目のデータセットは「アイテムリスト」タイプで、3 つのアイテムで構成されます。これらは色はそれぞれ異なるものの、線の幅、透過率、塗りつぶしは同じ線で表されます。
- 2 つ目のデータセットは「アイテムパターン」タイプで、6 つのアイテムで構成され、集計が設定されており、カスタムの色、幅、透過率、塗りつぶしを持つ線で表されます。
- 2 つ目のデータセットには、カスタムのデータセットラベルも設定されています。
- グラフ内のデータは直近 3 時間の期間で表示されます。
- グラフ内の障害は、設定されたアイテムについてのみ表示されます。
- グラフには 2 つの Y 軸があり、右側の Y 軸には 2 つ目のデータセットの値のみが表示されます。
- グラフの凡例には、設定されたアイテムが 4 行で表示されるほか、データセットの最小値、最大値、平均値も表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "svggraph",
                        "name": "Graph",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "ds.0.dataset_type",
                                "value": 0
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.0",
                                "value": 23264
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.0",
                                "value": "FF0000"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.1",
                                "value": 23269
                            },
                            {
                                "type": 1,
                                "name": "ds.0.color.1",
                                "value": "BF00FF"
                            },
                            {
                                "type": 4,
                                "name": "ds.0.itemids.2",
                                "value": 23257
                            },
                            {
                                "type": 1,
```

```
            "name": "ds.0.color.2",
            "value": "0040FF"
        },
        {
            "type": 0,
            "name": "ds.0.width",
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.0.transparency",
            "value": 3
        },
        {
            "type": 0,
            "name": "ds.0.fill",
            "value": 1
        },
        {
            "type": 1,
            "name": "ds.1.hosts.0",
            "value": "Zabbix server"
        },
        {
            "type": 1,
            "name": "ds.1.items.0",
            "value": "*: Number of processed *values per second"
        },
        {
            "type": 1,
            "name": "ds.1.color",
            "value": "000000"
        },
        {
            "type": 0,
            "name": "ds.1.transparency",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.1.fill",
            "value": 0
        },
        {
            "type": 0,
            "name": "ds.1.axisy",
            "value": 1
        },
        {
            "type": 0,
            "name": "ds.1.aggregate_function",
            "value": 3
        },
        {
            "type": 1,
            "name": "ds.1.aggregate_interval",
            "value": "1m"
        },
        {
            "type": 0,
            "name": "ds.1.aggregate_grouping",
            "value": 1
```

```
                            },
                            {
                                "type": 1,
                                "name": "ds.1.data_set_label",
                                "value": "Number of processed values per second"
                            },
                            {
                                "type": 0,
                                "name": "graph_time",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_period.from",
                                "value": "now-3h"
                            },
                            {
                                "type": 1,
                                "name": "time_period.to",
                                "value": "now"
                            },
                            {
                                "type": 0,
                                "name": "legend_statistic",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "legend_lines",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "show_problems",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "YZABC"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

9 Graph (classic)

説明

これらのパラメータと、対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、dashboard.create および dashboard.update メソッドで*Graph (classic)* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、その一方で、ウィジェットを誤って作成または更新してしまうリスクも生じます。*Graph (classic)* ウィジェットの作成または更新を正常に行うため、以下の表に示されたパラメータの動作を参照してください。

パラメータ

以下のパラメータは、グラフ（クラシック）ウィジェットでサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ソース	0	source_type	0 - *(デフォルト)* グラフ; 1 - シンプルグラフ。
グラフ	6	graphid.0	Graph ID。 パラメータの動作: - *Source* が”Graph” に設定されている場合は 必須
グラフ（ウィジェット）	1	graphid._reference	Graph ID の代わりに: ABCDE._graphid - 互換性のあるウィジェット（*Reference* パラメータが”ABCDE” に設定されているもの）を、グラフのデータソースとして設定します。 パラメータの動作: - *Source* が”Simple graph” に設定され、かつ *Graph* が設定されていない場合は 必須

パラメータ		type	name	value
アイテム		4	itemid.0	Item ID。 パラメータの動作: - *Source* が”Simple graph” に設定され、かつ *Item (Widget)* が設定されていない場合は 必須
	アイテム（ウィジェット）	1	itemid._reference	Item ID の代わりに: ABCDE._itemid - 互換性のあるウィジェット（*Reference* パラメータが”ABCDE” に設定されているもの）を、アイテムのデータソースとして設定します。 パラメータの動作: - *Source* が”Simple graph” に設定され、かつ *Item* が設定されていない場合は 必須
期間		1	time_period._reference	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（*Reference* パラメータが”ABCDE” に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定することもできます。
	From	1	time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - *Time period* が設定されていない場合に サポート - time_period.to が設定されている場合は 必須
	To	1	time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - *Time period* が設定されていない場合に サポート - time_period.from が設定されている場合は 必須
凡例の表示		0	show_legend	0 - 無効; 1 - *(デフォルト)* 有効。
ホストの上書き		1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（*Reference* パラメータが”ABCDE” に設定されているもの）を、ホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターを、ホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
Reference		1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、グラフ（クラシック）ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフ（クラシック）ウィジェットの設定

アイテム"42269" のシンプルなグラフを表示するグラフ（クラシック）ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "name": "Graph (classic)",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 1
                            },
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42269
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "RSTUV"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

10 Graph prototype

説明

これらのパラメータおよび各ダッシュボードウィジェットフィールドオブジェクトに対応するプロパティ値を使用すると、dashboard.create および dashboard.update メソッドでグラフプロトタイプ ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、その一方で、ウィジェットを誤って作成または更新してしまうリスクも生じます。グラフプロトタイプウィジェットの作成または更新を正常に行うため、以下の表に示すパラメータの動作を参照してください。

パラメータ

以下のパラメータは、グラフプロトタイプウィジェットでサポートされています。

Parameter	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ソース	0	source_type	2 - *(デフォルト)* グラフプロトタイプ; 3 - シンプルなグラフプロトタイプ。
グラフプロトタイプ	7	graphid.0	グラフプロトタイプ ID。 パラメータの動作: - ソースが「グラフプロトタイプ」に設定されている場合は 必須

Parameter	type	name	value
アイテムプロトタイプ	5	itemid.0	アイテムプロトタイプ ID。 パラメータの動作: - ソースが「シンプルなグラフプロトタイプ」に設定されている場合は必須
期間	1	time_period._reference	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（参照パラメータが「ABCDE」に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、期間は 開始および 終了パラメータでのみ設定することもできます。
開始	1	time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）による有効な時刻文字列。 パラメータの動作: - 期間が設定されていない場合に サポート - time_period.to が設定されている場合は 必須
終了	1	time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）による有効な時刻文字列。 パラメータの動作: - 期間が設定されていない場合に サポート - time_period.from が設定されている場合は 必須
凡例の表示	0	show_legend	0 - 無効; 1 - *(デフォルト)* 有効。
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（参照パラメータが「ABCDE」に設定されているもの）をホストのデータソースとして設定します; DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。
列数	0	columns	指定可能な値は 1～24 です。 デフォルト: 2。
行数	0	rows	指定可能な値は 1～16 です。 デフォルト: 1。
参照	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、グラフのプロトタイプウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを

目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

グラフのプロトタイプウィジェットの設定

ローレベルディスカバリによってアイテムのプロトタイプ (ID: ”42316”) から作成された 3 つのグラフ (3 列、1 行) のグリッドを表示するグラフのプロトタイプウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graphprototype",
                        "name": "Graph prototype",
                        "x": 0,
                        "y": 0,
                        "width": 48,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "source_type",
                                "value": 3
                            },
                            {
                                "type": 5,
                                "name": "itemid.0",
                                "value": 42316
                            },
                            {
                                "type": 0,
                                "name": "columns",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "OPQWX"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

11 Honeycomb

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでハニカムウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ハニカムウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

Honeycomb ウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメータ		type	name	value
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストタグ				
	評価タイプ	0	evaltype_host	0 - *(デフォルト)* And/Or; 2 - Or. このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメータ		type	name	value
	タグ値	1	host_tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグ順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
アイテムパターンアイテムタグ		1	items.0	アイテム の名前またはパターン 注: 複数のアイテムパターンを構成するには、プロパティ名に増分番号を含む、アイテムパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメーターの動作: - 必須
	評価タイプ	0	evaltype_item	0 - (デフォルト) And/Or; 2 - Or
	タグ名	1	item_tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - アイテムタグを構成する場合は 必須
	演算子	0	item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを構成する場合は 必須
	タグ値	1	item_tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - アイテムタグを構成する場合は 必須

パラメータ	type	name	value
メンテナンス中のホストを表示	0	メンテナンス	0 - *(デフォルト)* 無効; 1 - 有効
表示	0	表示 0	1 - プライマリラベル; 2 - セカンダリラベル 注: 複数の値を設定するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。 デフォルト: 1、2
参照	1	参照	5 文字からなる任意の文字列値 (例: ABCDE または JBPNL) この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

Advanced configuration

Honeycomb ウィジェットでは、以下の詳細設定パラメータがサポートされています。

Note:
Thresholds プロパティ名（例：thresholds.0.color）の数字は、リスト内のしきい値の位置を示し、昇順でソートされます。ただし、しきい値が異なる順序で設定されている場合は、Zabbix フロントエンドでウィジェット設定を更新すると、値は昇順でソートされます。(例：`"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`)

パラメータ	type	名前	値
プライマリラベルタイプ	0	primary_label_type	0 - *(デフォルト)* テキスト; 1 - 値
テキスト	1	primary_label	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ デフォルト: {HOST.NAME>
小数点以下の桁数	0	primary_label_decimal_places	パラメータの動作: - タイプが" テキスト " に設定されている場合に サポートされます。 有効な値の範囲は 0-6 です。 デフォルト: 2
			パラメータの動作: - *Type* が"Value" に設定されている場合にサポートされます。

パラメータ	type	名前	値
Size (type)	0	primary_label_size_type	0 - *(デフォルト)* 自動; 1 - カスタム
Size	0	primary_label_size	有効な値の範囲は 1-100 です。 デフォルト: 20 パラメータの動作: - *Size* (type) が”Custom” に設定されている場合にサポートされます。
太字	0	primary_label_bold	0 - *(デフォルト)* 無効; 1 - 有効
色	1	primary_label_color	16 進数カラーコード (例: FF0000) デフォルト: 設定オブジェクトとユーザーオブジェクトの theme に基づきます。 1F2C33 (blue-theme) または”hc-light” EEEEEE (dark-theme) または”hc-dark”
単位 (チェックボックス)	0	primary_label_units_show	0 - 無効; 1 - *(デフォルト)* 有効 パラメータの動作: - タイプが” 値” に設定されている場合に サポートされます。
単位 (値)	1	primary_label_units	任意の文字列値 "" (空) パラメータの動作: - タイプが” 値” に設定され、単位 (チェックボックス) が” 有効” に設定されている場合に サポートされます。
位置	0	primary_label_units_pos	0 - 前の値 1 - *(デフォルト)* 後の値 パラメータの動作: - タイプが” 値” に設定され、単位 (チェックボックス) が” 有効” に設定されている場合にサポートされます。 このパラメータは、次の時間関連の単位のいずれかに設定されている場合、無視されます。: unixtime、uptime、s
セカンダリラベルタイプ	0	secondary_label_type	0 - テキスト 1 - *(デフォルト)* 値
テキスト	1	secondary_label	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ。 デフォルト: {{ITEM.LASTVALUE}.fmtnum(2)> パラメータの動作: - *Type* が”Text” に設定されている場合に サポートされます。
小数点以下の桁数	0	secondary_label_decimal_places	可能な値の範囲は 0-6 です。 デフォルト: 2 パラメータの動作: - *Type* が”Value” に設定されている場合に サポートされます。
サイズ (type)	0	secondary_label_size_type	0 - *(デフォルト)* 自動; 1 - カスタム

パラメータ	type	名前	値
サイズ	0	secondary_label_size	有効な値の範囲は 1-100 です。 デフォルト: 30 パラメータの動作: - サイズ (type) が" カスタム" に設定されている場合に サポートされます。
太字	0	secondary_label_bold	0 - 無効; 1 - *(デフォルト)* 有効
色	1	secondary_label_color	16 進数カラーコード（例：FF0000） デフォルト：設定オブジェクトおよびユーザーオブジェクトの theme に基づきます。 "blue-theme" または"hc-light" の場合は 1F2C33、 "dark-theme" または"hc-dark" の場合は EEEEEE
単位（チェックボックス）	0	secondary_label_units_show	0 - 無効 1 - （デフォルト）有効 パラメータの動作: - *Type* が"Value" に設定されている場合にサポートされます。
単位 (値)	1	secondary_label_units	任意の文字列値 "" (空) パラメータの動作: - タイプが" 値" に設定され、単位 (チェックボックス) が" 有効" に設定されている場合に サポートされます。
位置	0	secondary_label_position	0 - 前の値 1 - *(デフォルト)* 後の値 パラメータの動作: - タイプが" 値" に設定され、単位 (チェックボックス) が" 有効" に設定されている場合に サポートされます。 このパラメータは、次の時間関連の単位 のいずれかに設定されている場合、無視されます。: unixtime、uptime、s
背景色			
背景色	1	bg_color	16 進数カラーコード（例：FF0000） デフォルト：設定オブジェクトおよびユーザーオブジェクトの theme に基づきます。 "blue-theme" の場合は D9E7ED、 "dark-theme" の場合は 3D5059、 "hc-light" の場合は AAD7E9、 "hc-dark" の場合は 335463
しきい値			
色補間	0	補間	0 - 無効、 1 - *(デフォルト)* 有効
Color	1	thresholds.0.color	16 進数のカラーコード（例：FF0000）
Threshold	1	thresholds.0.threshold	任意の数値サフィックス（例："1d"、"2w"、"4K"、"8G"）がサポートされています。

例

次の例は、ハニカムウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

ハニカムウィジェットの設定

Zabbix サーバープロセスの使用率を表示するハニカムウィジェットを設定します。さらに、ハニカムセルのプライマリラベルを変更し、しきい値を設定してウィジェットを視覚的に微調整します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "honeycomb",
                        "name": "Honeycomb",
                        "x": "0",
                        "y": "0",
                        "width": "24",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "items.0",
                                "value": "Zabbix server: Utilization*"
                            },
                            {
                                "type": 1,
                                "name": "primary_label",
                                "value": "{ITEM.NAME}"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.color",
                                "value": "0EC9AC"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "0"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "FFD54F"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "70"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.color",
```

```
                                "value": "FF465C"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.2.threshold",
                                "value": "90"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "KSTMQ"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

12 Host availability

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでホスト稼働状況ウィジェットを設定できます。

Parameters

ホスト可用性ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(デフォルト)* 15 分
ホストグループ		2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: `ABCDE._hostgroupids` の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 テンプレートダッシュボード でウィジェットを構成する場合、このパラメータはサポートされません。
インターフェースタイプ		0	interface_type.0	0 - なし; 1 - Zabbix エージェント (パッシブチェック); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix エージェント (アクティブチェック) デフォルト: 1、2、3、4、5 (すべて有効) 注: 複数の値を設定するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。
レイアウト		0	layout	0 - *(デフォルト)* 水平; 1 - 垂直
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効

パラメーター	type	name	value
合計のみを表示	0	only_totals	0 - *(デフォルト)* 無効; 1 - 有効

例

次の例は、ホスト稼働状況ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

ホスト稼働状況ウィジェットの設定

”Zabbix エージェント” および”SNMP” インターフェースが設定されているホストグループ”4” のホストの可用性情報を (縦型レイアウトで) 表示するホスト稼働状況ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostavail",
                        "name": "Host availability",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "interface_type",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
```

```
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

13 Host card

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでホストカードウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ホストカードウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

ホストカードウィジェットでは、以下のパラメーターがサポートされています。

パラメーター		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホスト		3	hostid.0	ホスト ID パラメーターの動作: - ホスト *(ウィジェット/ダッシュボード)* が設定されていない場合は 必須 テンプレートダッシュボード でウィジェットを構成する場合、このパラメーターはサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostid._reference	ホスト ID の代わりに: `DASHBOARD.hostid` - ホストセレクターをホストのデータソースとして設定します。 `ABCDE._hostid` - 互換性のあるウィジェット (*Reference* パラメータが"ABCDE" に設定されている) をホストのデータソースとして設定します。 パラメータの動作: - ホストが設定されていない場合は必須です このパラメータは、テンプレートダッシュボード.
抑制された問題を表示		0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効
表示		0	sections.0	0 - ホストグループ; 1 - 説明; 2 - 監視; 3 - 可用性; 4 - 監視元; 5 - テンプレート; 6 - インベントリー; 7 - タグ. 注: プロパティ名の数字は、セクションリスト内のセクションの順序を表します。複数のセクションを設定するには、プロパティ名に番号を増分した、セクションごとにダッシュボードウィジェットフィールドオブジェクトを作成します。

表示が" インベントリー" に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	名前	値
インベントリフィールド	0	inventory.0	インベントリ ID 注: 複数のインベントリフィールドを構成するには、プロパティ名に増分番号を含む、各インベントリフィールドのダッシュボードウィジェットフィールドオブジェクトを作成します。

例

次の例は、*Host card* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

Configuring a *Host card* widget

”監視”、”可用性”、”監視元”、”インベントリ”、”タグ” の各セクションを表示する ホストカードウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostcard",
                        "name": "Host card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostid.0",
                                "value": 10084
                            },
                            {
                                "type": 0,
                                "name": "show_suppressed",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 4
                            },
                            {
                                "type": 0,
```

```
                                "name": "sections.3",
                                "value": 6
                            },
                            {
                                "type": 0,
                                "name": "sections.4",
                                "value": 7
                            },
                            {
                                "type": 0,
                                "name": "inventory.0",
                                "value": 25
                            },
                            {
                                "type": 0,
                                "name": "inventory.1",
                                "value": 26
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


参照

- Dashboard widget field
- dashboard.create
- dashboard.update

14 Host navigator

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでホストナビゲーターウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。ホストナビゲーターウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

ホストナビゲーターウィジェットでは、以下のパラメータがサポートされています。

パラメータ		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ		2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: `ABCDE._hostgroupids` の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (その *Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
ホストパターン		1	hosts.0	ホスト の名前またはパターン 注: 複数のホストパターンを構成するには、プロパティ名に増分番号を含む、ホストパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。複数のホストパターンの場合、パラメータ ホストグループはまったく設定しないか、設定されたホストパターンに一致するホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
ホストステータス		0	ステータス	-1 - *(デフォルト)* 任意; 0 - 有効; 1 - 無効 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
ホストタグ				

パラメータ		type	name	value
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or. このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	タグ名	1	host_tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - ホストタグを構成する場合は 必須 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
重大度		0	severities.0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効) 注: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。

パラメータ		type	name	value
メンテナンス中のホストを表示		0	maintenance	0 - *(デフォルト)* 無効; 1 - 有効。
問題を表示		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制なし; 2 - なし。
グループ化	属性	0	group_by.0.attribute	0 - ホストグループ; 1 - タグ値; 2 - 重大度 注: プロパティ名の数字は、グループ化属性リスト内の属性の順序を示します。 パラメータの動作: - グループ化を構成する場合は 必須
	値	1	group_by.0.tag_name	任意の文字列値 注: プロパティ名に含まれる数字は、*Attribute* パラメータで設定されたグループ化属性を参照します。 パラメータの動作: - *Group by* を設定し、*Attribute* が" タグ値" に設定されている場合は 必須
ホスト制限		0	show_lines	有効な値の範囲は 1-9999 です。 デフォルト: 100 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
参照		1	reference	5 文字の任意の文字列値 (例: ABCDE または JBPNL) この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、ホストナビゲーターウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

ホストナビゲーターウィジェットの設定

ホストグループ別にグループ化されたホストと、"city" タグの値別にグループ化されたホストを表示するホストナビゲーターウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "hostnavigator",
                        "name": "Host navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "city"
                            },
                            {
                                "type": 1,
                                "name": "reference",
                                "value": "SWKLB"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
```

```
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

15 アイテムカード

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでアイテムカードウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。アイテムカードウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

アイテムカードウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
Item	4	itemid.0	Item ID。 パラメーターの動作: - *Item (Widget)* が設定されていない場合は 必須です。

パラメーター		type	name	value
	Item (Widget)	1	itemid._reference	Item ID: ABCDE._itemid の代わりに、アイテムのデータソースとして互換性のあるウィジェット (*Reference* パラメーターが"ABCDE に設定されている) を設定します。 パラメーターの動作: - *Item* が設定されていない場合は 必須です。
表示		0	セクション.0	0 - 説明; 1 - エラーテキスト; 2 - メトリクス; 3 - 最新データ; 4 - 情報の種類; 5 - トリガー; 6 - ホストインターフェース; 7 - タイプ; 8 - ホストインベントリ; 9 - タグ. 注: プロパティ名の数字は、セクションリスト内のセクションの順序を示します。複数のセクションを構成するには、プロパティ名に増分番号を付けたダッシュボードウィジェットフィールドオブジェクトをセクションごとに作成します。
ホストの上書き		1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット (その *Reference* パラメータが"ABCDE に設定されている) をホストのデータソースとして設定します。 DASHBOARD._hostid - ダッシュボードのホストセレクター をホストのデータソースとして設定します。 デフォルト: "" (空) テンプレートダッシュボード でウィジェットを構成する場合、このパラメータはサポートされません。

スパークライン

表示が" 最新データ" に設定されている場合、以下のパラメータがサポートされます。

パラメータ	type	name	value
幅	0	sparkline.width	指定可能な値は 0～10 です。 デフォルト: 1。
塗りつぶし	0	sparkline.fill	指定可能な値は 0～10 です。 デフォルト: 3。
色	1	sparkline.color	16 進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。

パラメータ	type	name	value
期間	1	sparkline.time_period.from	DASHBOARD._timeperiod - データソースとして期間セレクター を設定します。 ABCDE._timeperiod - データソースとして互換性のあるウィジェット（その reference パラメータが ABCDE に等しいもの）を設定します。 デフォルト: ""（空） または、開始および 終了パラメータでのみ期間を設定することもできます。
開始	1	sparkline.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対 時間構文（now、now/d、now/w-1w など）の有効な時刻文字列。 デフォルト: now-1h。 パラメータの動作: - 期間が設定されていない場合に サポート - sparkline.time_period.to が設定されている場合は 必須
終了	1	sparkline.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対 時間構文（now、now/d、now/w-1w など）の有効な時刻文字列。 デフォルト: now。 パラメータの動作: - 期間が設定されていない場合に サポート - time_period.from が設定されている場合は 必須
履歴データ	0	sparkline.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

例

次の例は、*Item card* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムカードウィジェットの設定

”説明”、”最新データ”、”トリガー”、”タグ” の各セクションを表示する アイテムカードウィジェットを設定します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemcard",
                        "name": "Item card",
                        "x": 0,
                        "y": 0,
                        "width": 14,
```

```
                        "height": 7,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42257
                            },
                            {
                                "type": 0,
                                "name": "sections.0",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "sections.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "sections.2",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "sections.3",
                                "value": 9
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```


Response:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


参照

- Dashboard widget field

- dashboard.create
- dashboard.update

16 アイテムの履歴

説明

これらのパラメータおよび対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、dashboard.create および dashboard.update メソッドでアイテム履歴 ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、一方でウィジェットを誤って作成または更新してしまうリスクも生じます。アイテム履歴ウィジェットの作成または更新を確実に成功させるため、以下の表に示されたパラメータの動作を参照してください。

Parameters

アイテム履歴ウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
レイアウト	0	レイアウト	0 - *(デフォルト)* 水平; 1 - 垂直
列 (下記を参照)			
行を表示	0	show_lines	有効な値の範囲は 1～100 です。 デフォルト: 25
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット（*Reference* パラメータを"ABCDE" に設定）をホストのデータソースとして設定します。 DASHBOARD._hostid - ダッシュボードのホストセレクターをホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合、サポートされません。
詳細設定 (下記を参照)			
参照	1	reference	5 文字からなる任意の文字列値 (例: ABCDE または JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

列

列には、アイテムパラメーターの設定に応じて、共通パラメーターと追加パラメーターがあります。

Note:
列に関連するすべてのパラメーターにおいて、プロパティ名に含まれる番号 (例: columns.0.name) は、パラメーターが設定されている列を参照します。

以下のパラメータはすべての列でサポートされています。

パラメータ	type	name	value
Name	1	columns.0.name	任意の文字列値 パラメータの動作: - 必須
項目	4	columns.0.itemid	項目 ID テンプレートダッシュボードでウィジェットを設定する場合は、テンプレートで設定されている項目のみを設定する必要があります。 パラメーターの動作: - 必須
基本色	1	columns.0.base_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

設定された *Item* が数値型の項目の場合、以下の列パラメータがサポートされます。

Parameter	type	name	value
Display	0	columns.0.display	1 - *(default)* そのまま; 2 - Bar; 3 - Indicators.
Min	1	columns.0.min	任意の数値 Parameter behavior: - *Display* が”Bar” または”Indicators” に設定されている場合に サポートされます。
最大	1	columns.0.max	任意の数値 パラメータの動作: - 表示が” バー” または” インジケーター” に設定されている場合にサポートされます。
しきい値			
色	1	columns.0.thresholds.color	16 進数のカラーコード（例：FF0000）
しきい値	1	columns.0.thresholds.threshold	任意の数値 シンタックス（例：”1d”、”2w”、”4K”、”8G”）がサポートされています
履歴データ	0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - 傾向

設定された項目が文字、テキスト、またはログタイプの項目の場合、以下の列パラメータがサポートされます。

パラメータ	type	name	value
ハイライト			
ハイライト	1	columns.0.highlights.color	16 進数のカラーコード（例：FF0000）
しきい値	1	columns.0.highlights.pattern	任意の正規表現

パラメータ	type	name	value
表示	0	columns.0.display	1 - *(デフォルト)* そのまま; 4 - HTML; 5 - 単一行
単一行	0	columns.0.max_length	有効な値の範囲は 1～500 です。 デフォルト: 100 パラメータの動作: - *Display* が" 単一行" に設定されている場合に サポートされます
等幅フォントを使用する	0	columns.0.monospace_font	0 - *(デフォルト)* デフォルトフォントを使用する。 1 - 等幅フォントを使用する。
ローカル時刻を表示	0	columns.0.local_time	0 - *(デフォルト)* タイムスタンプを表示します。 1 - ローカル時刻を表示します。 パラメーターの動作: - *Item* がログタイプ項目に設定され、*Show timestamp* が"Enabled" に設定されている場合に サポートされます。

設定されたアイテムがバイナリタイプのアイテムの場合、以下の列パラメータがサポートされます。

パラメータ	type	名前	値
サムネイルを表示	1	columns.0.show_thumbnail	0 - *(デフォルト)* 無効; 1 - 有効

高度な設定

以下の高度な設定パラメータは、アイテム履歴ウィジェットでサポートされています。

パラメータ	type	name	value
新しい値	0	sortorder	0 - *(デフォルト)* 上部; 1 - 下部。

パラメータ	type	name	value
タイムスタンプを表示	0	show_timestamp	0 - *(デフォルト)* 無効; 1 - 有効。
列ヘッダーを表示	0	show_column_header	0 - オフ; 1 - 水平; 2 - *(デフォルト)* 垂直。
期間	1	time_period._reference	DASHBOARD._timeperiod - データソースとして期間セレクターを設定します; ABCDE._timeperiod - データソースとして互換性のあるウィジェット（*Reference* パラメータが"ABCDE" に設定されているもの）を設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定することもできます。
From	1	time_period.from	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.to が設定されている場合は必須
To	1	time_period.to	絶対（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合はサポート - time_period.from が設定されている場合は必須

例

次の例は、アイテムヒストリウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムヒストリウィジェットの設定

2 つの数値アイテム"42269" と"42270" の最新データを表示するアイテムヒストリウィジェットを設定します。さらに、アイテム列を縦に、列名を横に表示するように設定します。表示データは 15 行に制限し、タイムスタンプ列を別に用意します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
```

```
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemhistory",
                        "name": "Item history",
                        "x": "0",
                        "y": "0",
                        "width": "18",
                        "height": "6",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "0",
                                "name": "layout",
                                "value": "1"
                            },
                            {
                                "type": "1",
                                "name": "columns.0.name",
                                "value": "CPU utilization"
                            },
                            {
                                "type": "4",
                                "name": "columns.0.itemid",
                                "value": "42269"
                            },
                            {
                                "type": "1",
                                "name": "columns.1.name",
                                "value": "Memory utilization"
                            },
                            {
                                "type": "4",
                                "name": "columns.1.itemid",
                                "value": "42270"
                            },
                            {
                                "type": "0",
                                "name": "show_lines",
                                "value": "15"
                            },
                            {
                                "type": "0",
                                "name": "show_timestamp",
                                "value": "1"
                            },
                            {
                                "type": "0",
                                "name": "show_column_header",
                                "value": "1"
                            },
                            {
                                "type": "1",
                                "name": "reference",
                                "value": "KIVKD"
                            }
                        ]
                    }
                ]
            }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

17 アイテムナビゲーター

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでアイテムナビゲーターウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。アイテムナビゲーターウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

アイテムナビゲーターウィジェットでは次のパラメータがサポートされています。

パラメータ		type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(デフォルト)* 15 分	
ホストグループ		2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合にはサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: `ABCDE._hostgroupids` の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (その *Reference* パラメーターを"ABCDE に設定) を設定します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合にはサポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを設定するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトを各ホストに作成します。複数のホストの場合、パラメータ ホストグループはまったく設定しないか、設定されたホストが属する少なくとも 1 つのホストグループを設定する必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされません。
	ホスト *(ウィジェット/ダッシュボード)*	1	hostids._reference	ホスト ID の代わりに: `DASHBOARD.hostid` - ホストセレクター をホストのデータソースとして設定します。 `ABCDE._hostid` - 互換性のあるウィジェット (その *Reference* パラメーターを"ABCDE に設定) をホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合にはサポートされません。
ホストタグ				
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合にはサポートされません。
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or この

パラメータ		type	name	value
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リストのタグ順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 テンプレートダッシュボード でウィジェットを構成する場合、このパラメーターはサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグ順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 テンプレートダッシュボード でウィジェットを構成する場合、このパラメーターはサポートされません。
アイテムパターンアイテムタグ		1	items.0	アイテム の名前またはパターン 注: 複数のアイテムパターンを構成するには、プロパティ名に増分番号を含む各アイテムパターンのダッシュボードウィジェットフィールドオブジェクトを作成します。
	評価タイプ	0	item_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or
	タグ名	1	item_tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - アイテムタグを構成する場合は 必須

パラメータ		type	name	value
	演算子	0	item_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。
	タグ値	1	item_tags.0.value	パラメーターの動作: - アイテムタグを構成する場合は 必須 任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。
状態		0	状態	パラメーターの動作: - アイテムタグを構成する場合は 必須 -1 - *(デフォルト)* すべて; 0 - 正常; 1 - サポートされていません。
問題を表示		0	show_problems	0 - すべて; 1 - *(デフォルト)* 抑制されない; 2 - なし
グループ化属性	属性	0	group_by.0.attribute	0 - ホストグループ 1 - ホスト名 2 - ホストタグ値 3 - 項目タグ値 注: プロパティ名の数字は、グループ化属性リスト内の属性の順序を示します。
	値	1	group_by.0.tag_name	パラメーターの動作: - *Group by* を構成する場合は 必須 任意の文字列値 注: プロパティ名の数字は、*Attribute* パラメータで設定されたグループ化属性を参照します。
アイテム制限		0	show_lines	パラメータの動作: - *Group by* を構成し、*Attribute* が” ホストタグ値” または” アイテムタグ値” に設定されている場合は 必須 可能な値の範囲は 1 ～9999 です。 デフォルト: 100

パラメータ	type	name	value
参照	1	参照	5 文字からなる任意の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、アイテムナビゲーターウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムナビゲーターウィジェットの設定

最大 1000 個のアイテムをホスト別、さらに” コンポーネント” アイテムタグ値別にグループ化して表示するアイテムナビゲーターウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": "30",
        "auto_start": "1",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "itemnavigator",
                        "name": "Item navigator",
                        "x": "0",
                        "y": "0",
                        "width": "12",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": 0,
                                "name": "group_by.0.attribute",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "group_by.1.attribute",
                                "value": 3
                            },
                            {
                                "type": 1,
                                "name": "group_by.1.tag_name",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 1000
                            },
                            {
                                "type": 1,
```

```
                                "name": "reference",
                                "value": "DFNLK"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

18 アイテムの値

説明

これらのパラメータと、それぞれのダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、dashboard.create および dashboard.update メソッドでアイテム値 ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、その一方で、ウィジェットを誤って作成または更新してしまうリスクも生じます。アイテム値ウィジェットを正常に作成または更新するために、以下の表に示されたパラメータの動作を参照してください。

Parameters

アイテム値ウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
Item	4	itemid.0	Item ID パラメーターの動作: - *Item (Widget)* が設定されていない場合は 必須です。
Item (Widget)	1	itemid._reference	Item ID: ABCDE._itemid の代わりに、アイテムのデータソースとして互換性のあるウィジェット (*Reference* パラメーターが"ABCDE に設定されている) を設定します。 パラメーターの動作: - *Item* が設定されていない場合は 必須です。
表示	0	表示 0	1 - 説明 2 - 値 3 - 時間 4 - 変化インジケーター 5 - スパークライン デフォルト: 1、2、3、4 注: 複数の値を設定するには、プロパティ名に増分番号を付けた各値に対してダッシュボードウィジェットフィールドオブジェクトを作成します。
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット (その *Reference* パラメータを"ABCDE に設定) をホストのデータソースとして設定します。 DASHBOARD._hostid - ダッシュボードのホストセレクタ をホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合にはサポートされません。

高度な設定

以下の高度な設定パラメータは、アイテム値ウィジェットでサポートされています。

Note:
しきい値プロパティ名の番号（例: thresholds.0.color）は、昇順に並べ替えられたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix Web インターフェースでウィジェット設定を更新した後、値は昇順に並べ替えられます（例: "thresholds.0.threshold":"5" → "thresholds.0.threshold":"1"; "thresholds.1.threshold":"1" → "thresholds.1.threshold": "5"）。

パラメータ	type	name	value
背景色	1	bg_color	16 進数カラーコード（例: FF0000）。 デフォルト: ""（空）。

パラメータ		type	name	value
しきい値				
	色	1	thresholds.0.color	16 進数カラーコード（例: FF0000）。
	しきい値	1	thresholds.0.threshold	任意の文字列値。
集計関数		0	aggregate_function	0 - *(デフォルト)* 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
期間		1	time_period._reference	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（その reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定することもできます。 パラメータの動作: - *supported* if *Aggregation function* is set to ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last”
	From	1	time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now, now/d, now/w-1w など）で有効な時間文字列。 パラメータの動作: - *supported* if *Time period* is not set and *Aggregation function* is set to ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *required* if time_period.to is set
	To	1	time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now, now/d, now/w-1w など）で有効な時間文字列。 パラメータの動作: - *supported* if *Time period* is not set and *Aggregation function* is set to ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *required* if time_period.from is set
履歴データ		0	history	0 - *(デフォルト)* Auto; 1 - History; 2 - Trends.

Description

表示が” 説明” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
説明	1	説明	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{ITEM.*}、{INVENTORY.*}、ユーザーマクロ デフォルト: {ITEM.NAME}
水平位置	0	desc_h_pos	0 - 左 1 - *(デフォルト)* 中央 2 - 右 2 つ以上の要素 (説明、値、時間) で同じ 水平位置と 垂直位置を共有することはできません。
垂直位置	0	desc_v_pos	0 - 上; 1 - 中央; 2 - *(デフォルト)* 下 2 つ以上の要素 (説明、値、時刻) が同じ 水平位置と 垂直位置を共有することはできません。
サイズ	0	desc_size	指定可能な値の範囲は 1～100 です。 デフォルト: 15
太字	0	desc_bold	0 - *(デフォルト)* 無効; 1 - 有効
色	1	desc_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

Value

表示が” 値” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ		type	名前	値
小数点以下の桁数				
	小数点以下の桁数	0	decimal_places	有効な値の範囲は 1～10 です。 デフォルト: 2
	サイズ	0	decimal_size	有効な値の範囲は 1～100 です。 デフォルト: 35
位置				
	水平位置	0	value_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右 2 つ以上の要素 (説明、値、時間) が同じ 水平位置と 垂直位置を共有することはできません。

パラメータ		type	名前	値
	垂直位置	0	value_v_pos	0 - 上; 1 - *(デフォルト)* 中央; 2 - 下 2 つ以上の要素 (説明、値、時間) が同じ 水平位置と 垂直位置を共有することはできません。
	サイズ	0	value_size	有効な値の範囲は 1～100 です。 デフォルト: 45
	太字	0	value_bold	0 - 無効; 1 - *(デフォルト)* 有効
	色	1	値 _ 色	16 進数のカラーコード（例：FF0000） デフォルト："" (空)
単位				
	単位（チェックボックス）	0	単位表示	0 - 無効 1 - （デフォルト）有効
	単位（値）	1	単位	任意の文字列値
	位置	0	単位 _ 位置	0 - 前の値 1 - 上の値 2 - （デフォルト）後の値 3 - 下の値
	サイズ	0	単位 _ サイズ	可能な値の範囲は 1～100 です。 デフォルト：35
	太字	0	units_bold	0 - 無効; 1 - *(デフォルト)* 有効
	色	1	units_color	16 進数の色コード (例: FF0000) デフォルト: "" (空)

時間

表示が” 時間” に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
水平位置	0	time_h_pos	0 - 左; 1 - *(デフォルト)* 中央; 2 - 右 2 つ以上の要素（説明、値、時間）が同じ水平位置と垂直位置を共有することはできません。
垂直位置	0	time_v_pos	0 - *(デフォルト)* 上; 1 - 中央; 2 - 下 2 つ以上の要素（説明、値、時間）が同じ水平位置と垂直位置を共有することはできません。
サイズ	0	time_size	指定可能な値の範囲は 1～100 です。 デフォルト: 15
太字	0	time_bold	0 - *(デフォルト)* 無効 1 - 有効

パラメータ	type	名前	値
色	1	time_color	16 進数のカラーコード (例: FF0000) デフォルト: "" (空)

変更インジケーター

表示が" 変更インジケーター" に設定されている場合、以下の詳細設定パラメータがサポートされます。

パラメータ	type	名前	値
変更インジケーター ↑ の色	1	up_color	16 進数カラーコード（例：FF0000） デフォルト："" （空）
変更インジケーター ↓ の色	1	down_color	16 進数カラーコード（例：FF0000） デフォルト："" （空）
インジケーターの色変更	1	updown_color	16 進数カラーコード（例：FF0000） デフォルト："" （空）

スパークライン

Show が"Sparkline" に設定されている場合、以下の高度な設定パラメータを使用できます。

パラメータ	type	name	value
幅	0	sparkline.width	指定可能な値の範囲は 0～10 です。 デフォルト: 1。
塗りつぶし	0	sparkline.fill	指定可能な値の範囲は 0～10 です。 デフォルト: 3。
色	1	sparkline.color	16 進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。
期間	1	sparkline.time_period.reference	DASHBOARD._timeperiod - データソースとして期間セレクターを設定します。 ABCDE._timeperiod - データソースとして互換性のあるウィジェット（reference パラメータが ABCDE に等しいもの）を設定します。 デフォルト: DASHBOARD._timeperiod または、開始および 終了パラメータでのみ期間を設定することもできます。
開始	1	sparkline.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）で指定する有効な時刻文字列。 パラメータの動作: - sparkline.time_period.to が設定されている場合は 必須
終了	1	sparkline.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）で指定する有効な時刻文字列。 パラメータの動作: - sparkline.time_period.from が設定されている場合は 必須

パラメータ	type	name	value
履歴データ	0	sparkline.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。

例

次の例は、アイテムの値ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

アイテムの値ウィジェットの設定

アイテム"42266"(Zabbix エージェントの可用性) のアイテム値を表示するアイテムの値ウィジェットを設定します。さらにウィジェットを複数の詳細オプションを使用して、Zabbix エージェントの可用性ステータスに基づいて変更される動的な背景色といった視覚的な調整をします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "item",
                        "name": "Item value",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 4,
                                "name": "itemid.0",
                                "value": 42266
                            },
                            {
                                "type": 0,
                                "name": "show.0",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show.1",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "show.2",
                                "value": 3
                            },
                            {
```

```
            "type": 1,
            "name": "description",
            "value": "Agent status"
        },
        {
            "type": 0,
            "name": "desc_h_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "desc_v_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "desc_bold",
            "value": 1
        },
        {
            "type": 1,
            "name": "desc_color",
            "value": "F06291"
        },
        {
            "type": 0,
            "name": "value_h_pos",
            "value": 0
        },
        {
            "type": 0,
            "name": "value_size",
            "value": 25
        },
        {
            "type": 1,
            "name": "value_color",
            "value": "FFFF00"
        },
        {
            "type": 0,
            "name": "units_show",
            "value": 0
        },
        {
            "type": 0,
            "name": "time_h_pos",
            "value": 2
        },
        {
            "type": 0,
            "name": "time_v_pos",
            "value": 2
        },
        {
            "type": 0,
            "name": "time_size",
            "value": 10
        },
        {
            "type": 0,
            "name": "time_bold",
```

```
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "time_color",
                                "value": "9FA8DA"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.color",
                                "value": "E1E1E1"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.0.threshold",
                                "value": "0"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.color",
                                "value": "D1C4E9"
                            },
                            {
                                "type": 1,
                                "name": "thresholds.1.threshold",
                                "value": "1"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`

- dashboard.update

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでマップウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。マップウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

マップウィジェットでは、以下のパラメーターがサポートされています。

パラメータ	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - (デフォルト) 15 分
マップ	8	sysmapid.0	マップID。 パラメーターの動作: - マップ (ウィジェット) が設定されていない場合、必須
マップ (ウィジェット)	1	sysmapid._reference	ABCDE._mapid - マップのデータソースとして、マップナビゲーションウィジェット (参照パラメーターが"ABCDE" に設定されている) を設定します。 パラメーターの動作: - マップが設定されていない場合、必須
参照	1	reference	5 文字からなる任意の文字列値 (例: ABCDE や JBPNL)。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメーターの動作: - 必須

例

次の例は、マップウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

マップウィジェットの設定

マップ"1" を表示するマップウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
```

```
                "widgets": [
                    {
                        "type": "map",
                        "name": "Map",
                        "x": 0,
                        "y": 0,
                        "width": 54,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 8,
                                "name": "sysmapid.0",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

リンクされたマップウィジェットの設定

マップナビゲーションウィジェットにリンクされたマップウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
```

```
            "type": "map",
            "name": "Map",
            "x": 0,
            "y": 5,
            "width": 54,
            "height": 5,
            "view_mode": 0,
            "fields": [
                {
                    "type": 1,
                    "name": "sysmapid._reference",
                    "value": "ABCDE._mapid"
                }
            ]
        },
        {
            "type": "navtree",
            "name": "Map navigation tree",
            "x": 0,
            "y": 0,
            "width": 18,
            "height": 5,
            "view_mode": 0,
            "fields": [
                {
                    "type": 1,
                    "name": "navtree.1.name",
                    "value": "Element A"
                },
                {
                    "type": 1,
                    "name": "navtree.2.name",
                    "value": "Element B"
                },
                {
                    "type": 1,
                    "name": "navtree.3.name",
                    "value": "Element C"
                },
                {
                    "type": 1,
                    "name": "navtree.4.name",
                    "value": "Element A1"
                },
                {
                    "type": 1,
                    "name": "navtree.5.name",
                    "value": "Element A2"
                },
                {
                    "type": 1,
                    "name": "navtree.6.name",
                    "value": "Element B1"
                },
                {
                    "type": 1,
                    "name": "navtree.7.name",
                    "value": "Element B2"
                },
                {
                    "type": 0,
                    "name": "navtree.4.parent",
```

```
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.parent",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.6.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.7.parent",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.1.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.2.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.3.order",
            "value": 3
        },
        {
            "type": 0,
            "name": "navtree.4.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.5.order",
            "value": 2
        },
        {
            "type": 0,
            "name": "navtree.6.order",
            "value": 1
        },
        {
            "type": 0,
            "name": "navtree.7.order",
            "value": 2
        },
        {
            "type": 8,
            "name": "navtree.6.sysmapid",
            "value": 1
        },
        {
            "type": 1,
            "name": "reference",
            "value": "ABCDE"
        }
```

```
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update
- Map navigation tree

20 マップナビゲーションツリー

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでマップナビゲーションウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。マップナビゲーションウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

マップナビゲーションツリーウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - *(デフォルト)* 15 分
利用できないマップを表示	1	show_unavailable	0 - *(デフォルト)* 無効; 1 - 有効
参照	1	reference	5 文字の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

マップナビゲーションツリー要素の設定には、以下のパラメータがサポートされています。

パラメータ	タイプ	名前	値
名前	1	navtree.1.name	任意の文字列値 注: プロパティ名に含まれる数字は要素番号を設定します。
リンクされたマップ	8	navtree.1.sysmapid	マップ ID 注: プロパティ名に含まれる数字は、マップがリンクされている要素を参照します。
要素階層を作成するためのパラメーター	0	navtree.1.parent	親要素番号 注: プロパティ名に含まれる数字は、子要素を参照します。プロパティ値は親要素を参照します。
	0	navtree.1.order	マップナビゲーションツリー内の要素の位置 注: プロパティ名に含まれる数字は、要素番号を参照します。プロパティ値は、マップナビゲーションツリー内の要素の位置を参照します。親要素の位置は、マップナビゲーションツリー全体内で決定されます。子要素の位置は、親要素内で決定されます。

例

次の例は、マップナビゲーションウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

マップナビゲーションウィジェットの設定

以下のマップナビゲーションツリーを表示するマップナビゲーションウィジェットを設定します:

- 要素 A
  - 要素 A1
  - 要素 A2
- 要素 B
  - 要素 B1 (リンクされたマップウィジェットに表示できるリンクされたマップ"1" を含む)
  - 要素 B2
- 要素 C

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
```

```
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "navtree",
                        "name": "Map navigation tree",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "navtree.1.name",
                                "value": "Element A"
                            },
                            {
                                "type": 1,
                                "name": "navtree.2.name",
                                "value": "Element B"
                            },
                            {
                                "type": 1,
                                "name": "navtree.3.name",
                                "value": "Element C"
                            },
                            {
                                "type": 1,
                                "name": "navtree.4.name",
                                "value": "Element A1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.5.name",
                                "value": "Element A2"
                            },
                            {
                                "type": 1,
                                "name": "navtree.6.name",
                                "value": "Element B1"
                            },
                            {
                                "type": 1,
                                "name": "navtree.7.name",
                                "value": "Element B2"
                            },
                            {
                                "type": 0,
                                "name": "navtree.4.parent",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.5.parent",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "navtree.6.parent",
```

```
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "navtree.7.parent",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "navtree.1.order",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "navtree.2.order",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "navtree.3.order",
                            "value": 3
                        },
                        {
                            "type": 0,
                            "name": "navtree.4.order",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "navtree.5.order",
                            "value": 2
                        },
                        {
                            "type": 0,
                            "name": "navtree.6.order",
                            "value": 1
                        },
                        {
                            "type": 0,
                            "name": "navtree.7.order",
                            "value": 2
                        },
                        {
                            "type": 8,
                            "name": "navtree.6.sysmapid",
                            "value": 1
                        },
                        {
                            "type": 1,
                            "name": "reference",
                            "value": "HJQXF"
                        }
                    ]
                }
            ]
        }
    ],
    "userGroups": [
        {
            "usrgrpid": 7,
            "permission": 2
        }
```

```
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update
- Map

21 円グラフ

説明

これらのパラメータおよび対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、dashboard.create および dashboard.update メソッドで円グラフ ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの fields プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、その一方で、ウィジェットを誤って作成または更新してしまうリスクも生じます。円グラフウィジェットを正常に作成または更新するには、以下の表に示されたパラメータの動作を参照してください。

パラメーター

円グラフウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - *(デフォルト)* 1 分 120 - 2 分 600 - 10 分 900 - 15 分

データセット

以下のパラメータは、データセットの設定でサポートされています。

Note:
プロパティ名の最初の数字（例: ds.0.hosts.0、ds.0.items.0）は特定のデータセットを表し、2 番目の数字が存在する場合は、設定されたホストまたはアイテムを表します。

パラメータ		type	name	value
データセットタイプ		0	ds.0.dataset_type	0 - アイテムリスト; 1 - *(デフォルト)* アイテムパターン。
	アイテム	4	ds.0.itemids.0	Item ID。 ウィジェットをテンプレートダッシュボードで設定する場合は、テンプレート上で設定されたアイテムのみを指定する必要があります。 注意: 複数のアイテムを設定するには、プロパティ名の番号を増やして、各アイテムごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 Parameter behavior: - *required* if *Data set type* is set to ”Item list” and *Items (Widget)* is not set
	アイテム（ウィジェット）	1	ds.0.itemids.0._reference	Item ID の代わりに: `ABCDE._itemid` - アイテムのデータソースとして、互換性のあるウィジェット（*Reference* パラメータを”ABCDE” に設定）を指定します。 注意: 複数のウィジェットを設定するには、プロパティ名の番号を増やして、各ウィジェットごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 Parameter behavior: - *required* if *Data set type* is set to ”Item list” and *Items* is not set
	色	1	ds.0.color.0	16 進数のカラーコード（例: FF0000）。 Parameter behavior: - *required* if *Data set type* is set to ”Item list”
	アイテムタイプ	0	ds.0.type.0	0 - *(デフォルト)* 通常; 1 - 合計。 値” 合計” は、チャート全体で 1 つのアイテムに対してのみ設定できます。 Parameter behavior: - *supported* if *Data set type* is set to ”Item list”
	ホストパターン	1	ds.0.hosts.0	Host 名またはパターン（例: ”Zabbix*”）。 Parameter behavior: - *required* if *Data set type* is set to ”Item patterns” このパラメータは、テンプレートダッシュボードでウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
	アイテムパターン	1	ds.0.items.0	Item 名またはパターン（例: ”*: 1 秒あたりに処理された *values の数”）。 ウィジェットをテンプレートダッシュボードで設定する場合は、テンプレート上で設定されたアイテムに対するパターンのみを指定する必要があります。 Parameter behavior: - *required* if *Data set type* is set to ”Item patterns”
	色	1	ds.0.color	16 進数のカラーコード（例: FF0000）。 Parameter behavior: - *supported* if *Data set type* is set to ”Item patterns” and *Color palette* is not set
	カラーパレット	0	ds.0.color_palette	カラーパレットのインデックス。 指定可能な値は 0～11 です。 デフォルト: 0。 Parameter behavior: - *supported* if *Data set type* is set to ”Item patterns” and *Color* is not set
集計関数		0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - *(デフォルト)* last。
データセット集計		0	ds.0.dataset_aggregation	0 - *(デフォルト)* なし; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum。 Parameter behavior: - *supported* if *Item type* is set to ”Total”
データセットラベル		1	ds.0.data_set_label	任意の文字列値。 デフォルト: ""（空）。

Displaying options

表示オプションの設定には、以下のパラメータがサポートされています。

パラメータ		type	name	value
履歴データの選択		0	source	0 - *(default)* 自動; 1 - 履歴; 2 - トレンド
描画		0	draw_type	0 - *(default)* 円グラフ; 1 - ドーナツグラフ
	幅	0	width	20 - 半径の 20%。 30 - 半径の 30%。 40 - 半径の 40%。 50 - *(*デフォルト*)* 半径の 50% パラメータの動作: - *Draw* が”Doughnut” に設定されている場合に サポートされます
	ストローク幅	0	stroke	設定可能な値の範囲は 0～10 です。 デフォルト: 0
	合計値を表示	0	total_show	パラメータの動作: - *Draw* が”Doughnut” に設定されている場合に サポートされます 0 - *(*デフォルト*)* 無効; 1 - 有効。
	Size	0	value_size_type	パラメータの動作: - *Draw* が” ドーナツ” に設定されている場合に サポートされます 0 - *(*デフォルト*)* 自動; 1 - カスタム
	サイズ (カスタムサイズの値)	0	value_size	パラメータの動作: - 合計値を表示が” 有効” に設定されている場合に サポートされます 有効な値の範囲は 1～100 です。 デフォルト: 20
	小数点以下の桁数	0	decimal_places	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます。 有効な値の範囲は 0～6 です。 デフォルト: 2
	単位 (チェックボックス)	0	units_show	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます。 0 - *(*デフォルト*)* 無効; 1 - 有効
	単位 (値)	1	units	パラメータの動作: - 合計値の表示が” 有効” に設定されている場合に サポートされます 任意の文字列値 パラメータの動作: - 単位 (チェックボックス) が” 有効” に設定されている場合に サポートされます

パラメータ		type	name	value
	太字	0	value_bold	0 - *(デフォルト)* 無効; 1 - 有効。 パラメータの動作: - 合計値の表示が" 有効" に設定されている場合に サポートされます
	色	1	value_color	16 進数のカラーコード (例: FF0000)。 パラメータの動作: - 合計値の表示が" 有効" に設定されている場合に サポートされます
セクター間のスペース		0	space	指定可能な値の範囲は 0～10 です。 デフォルト: 1。
N% 未満のセクターをマージ (チェックボックス)		0	merge	0 - *(デフォルト)* 無効; 1 - 有効。
N% 未満のセクターをマージ (値)		0	merge_percent	指定可能な値の範囲は 1～10 です。 デフォルト: 1。 パラメータの動作: - *N*% 未満のセクターをマージ (チェックボックス) が" 有効" に設定されている場合に サポートされます。

パラメータ	type	name	value
N% 未満のセクターをマージ (色)	1	merge_color	16 進数のカラーコード (例: FF0000)。 パラメーターの動作: - *N*% 未満のセクターをマージ (チェックボックス) が" 有効" に設定されている場合に サポートされます

期間

以下のパラメータは、期間の設定でサポートされています。

パラメータ	type	name	value
期間	1	time_period._reference	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（*Reference* パラメータが"ABCDE" に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、*From* および *To* パラメータでのみ期間を設定することもできます。
From	1	time_period.from	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合は *supported* - time_period.to が設定されている場合は *required*
To	1	time_period.to	絶対時間形式（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されていない場合は *supported* - time_period.from が設定されている場合は *required*

Legend

凡例の設定では、以下のパラメータがサポートされています。

パラメータ	type	name	value
凡例の表示	0	legend	0 - 無効; 1 - *(デフォルト)* 有効

パラメータ	type	name	value
値の表示	0	legend_value	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - 凡例の表示が" 有効" に設定されている場合にサポートされます。
集計関数を表示	0	legend_aggregation	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - 凡例の表示が" 有効" に設定されている場合にサポートされます
行	0	legend_lines_mode	0 - *(デフォルト)* 固定; 1 - 可変 パラメータの動作: - 凡例の表示が" 有効" に設定されている場合にサポートされます
行数/ 最大行数	0	legend_lines	有効な値の範囲は 1～10 です。 デフォルト: 1. パラメータの動作: - 凡例の表示が" 有効" に設定されている場合に サポートされます。
列数	0	legend_columns	有効な値の範囲は 1～4 です。 デフォルト: 4. パラメータの動作: - 凡例の表示が" 有効" に設定され、値の表示が" 無効" に設定されている場合に サポートされます。

例

次の例は、円グラフウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

円グラフウィジェットの設定

円グラフウィジェットは、次のように設定します。

- 1 つのホスト上の合計 9 個のアイテムに対して、2 つのデータセットを設定します。
- 1 つ目のデータセットは「アイテムリスト」タイプで、すべて「通常」タイプの 3 つのアイテムで構成され、それぞれ異なる色で表示されます。
- 2 つ目のデータセットは「アイテムパターン」タイプで、6 つのアイテムで構成され、各アイテムに集計が設定され、カスタム色で表示されます。
- 2 つ目のデータセットには、カスタムデータセットラベルも設定されています。
- 円グラフ内のデータは、カスタム幅のドーナツチャートとして表示され、中央には単位付きの合計値が表示されます。
- 円グラフ内のデータは、直近 3 時間のカスタム期間について表示および集計されます。
- 円グラフの凡例には、設定されたアイテムが 4 行で表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "piechart",
                        "name": "Pie chart",
                        "x": 0,
```

```
                "y": 0,
                "width": 24,
                "height": 5,
                "view_mode": 0,
                "fields": [
                    {
                        "type": 0,
                        "name": "ds.0.dataset_type",
                        "value": 0
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.0",
                        "value": 23264
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.0",
                        "value": "FF0000"
                    },
                    {
                        "type": 0,
                        "name": "ds.0.type.0",
                        "value": 0
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.1",
                        "value": 23269
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.1",
                        "value": "BF00FF"
                    },
                    {
                      "type": 0,
                      "name": "ds.0.type.1",
                      "value": 0
                    },
                    {
                        "type": 4,
                        "name": "ds.0.itemids.2",
                        "value": 23257
                    },
                    {
                        "type": 1,
                        "name": "ds.0.color.2",
                        "value": "0040FF"
                    },
                    {
                      "type": 0,
                      "name": "ds.0.type.2",
                      "value": 0
                    },
                    {
                        "type": 1,
                        "name": "ds.1.hosts.0",
                        "value": "Zabbix server"
                    },
                    {
                        "type": 1,
```

```
                        "name": "ds.1.items.0",
                        "value": "*: Number of processed *values per second"
                    },
                    {
                        "type": 1,
                        "name": "ds.1.color",
                        "value": "000000"
                    },
                    {
                        "type": 0,
                        "name": "ds.1.aggregate_function",
                        "value": 3
                    },
                    {
                        "type": 1,
                        "name": "ds.1.data_set_label",
                        "value": "Number of processed values per second"
                    },
                    {
                        "type": 0,
                        "name": "draw_type",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "width",
                        "value": 30
                    },
                    {
                        "type": 0,
                        "name": "total_show",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "units_show",
                        "value": 1
                    },
                    {
                        "type": 0,
                        "name": "graph_time",
                        "value": 1
                    },
                    {
                        "type": 1,
                        "name": "time_period.from",
                        "value": "now-3h"
                    },
                    {
                        "type": 1,
                        "name": "time_period.to",
                        "value": "now"
                    },
                    {
                        "type": 0,
                        "name": "legend_lines",
                        "value": 4
                    }
                ]
            }
        ]
    }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

22 障害ホスト

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、`dashboard.create` メソッドおよび `dashboard.update` メソッドで障害中のホストウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。障害中のホストウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

問題のホストウィジェットでは、以下のパラメータがサポートされています。

パラメータ		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ		2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: `ABCDE._hostgroupids` の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメーターを"ABCDE に設定) を設定します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストグループを除外		2	exclude_groupids.0	ホストグループ ID 注: 複数のホストグループを除外するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストグループごとに作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
問題		1	問題	問題イベント名 (大文字と小文字は区別されません、完全な名前またはその一部)
重大度		0	重大度。0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効)。 注: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。

パラメータ	type	name	value
問題タグ			
評価タイプ	0	評価タイプ	0 - *(デフォルト)* And/Or 2 - Or
タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名内の数字は、タグ評価リスト内のタグ順序を表します。 パラメータの動作: - 問題タグを構成する場合は 必須
演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名内の数字は、タグ評価リスト内のタグ順序を表します。 パラメータの動作: - 問題タグを構成する場合は 必須
タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグ順序を表します。 パラメータの動作: - 問題タグを構成する場合は 必須
抑制された問題を表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効
問題のないグループを非表示	0	hide_empty_groups	0 - *(デフォルト)* 無効; 1 - 有効 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
問題表示	0	ext_ack	0 - *(デフォルト)* すべて; 1 - 未確認のみ; 2 - 分離済み

パラメータ	type	name	value
参照	1	reference	5 文字からなる任意の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、障害中のホストウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

障害中のホストウィジェットの設定

ホストグループ”2” と”4” に属する、名前に”CPU” という文字列が含まれ、深刻度が” 警告”、” 軽度の障害”、” 重度の障害”、” 致命的な障害” の障害が発生しているホストを表示する障害中のホストウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemhosts",
                        "name": "Problem hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 2
                            },
                            {
                                "type": 2,
                                "name": "groupids.1",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "problem",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
```

```
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

23 件の障害

説明

これらのパラメータおよび対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、dashboard.create および dashboard.update メソッドで*Problems* ウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成時または更新時に検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、一方でウィジェットを誤って作成または更新してしまうリスクも生じます。*Problems* ウィジェットの作成または更新を正常に行うには、以下の表に示されたパラメータの動作を参照してください。

パラメータ

以下のパラメータは、障害ウィジェットでサポートされています。

パラメータ		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
表示		0	show	1 - *(デフォルト)* 最近の障害; 2 - 履歴; 3 - 障害。
ホストグループ		2	groupids.0	ホストグループ ID。 注意: 複数のホストグループを設定するには、プロパティ名の番号を増やして、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID の代わりに: `ABCDE._hostgroupids` - 互換性のあるウィジェット（*Reference* パラメータが"ABCDE" に設定されているもの）を、ホストグループのデータソースとして設定します。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
除外するホストグループ		2	exclude_groupids.0	ホストグループ ID。 注意: 複数のホストグループを除外するには、プロパティ名の番号を増やして、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホスト ID。 注意: 複数のホストを設定するには、プロパティ名の番号を増やして、ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを指定する場合、ホストグループパラメータはまったく設定しないか、設定したホストが属するホストグループを少なくとも 1 つ指定して設定する必要があります。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。

パラメータ		type	name	value
	ホスト (ウィジェット/ダッシュボード)	1	hostids._reference	ホスト ID の代わりに: DASHBOARD.hostids - ホストセレクターをホストのデータソースとして設定します; ABCDE._hostids - 互換性のあるウィジェット（*Reference* パラメータが"ABCDE" に設定されているもの）を、ホストのデータソースとして設定します。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
障害		1	problem	障害のイベント名（大文字と小文字を区別しません。完全一致または部分一致）。
深刻度		0	severities.0	0 - 未分類; 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害。 デフォルト: 空欄（すべて有効）。 注意: 複数の値を設定するには、プロパティ名の番号を増やして、値ごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注意: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 障害タグを設定する場合は 必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注意: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 障害タグを設定する場合は 必須
	タグ値	1	tags.0.value	任意の文字列値。 注意: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 障害タグを設定する場合は 必須

パラメータ	type	name	value
タグの表示	0	show_tags	0 - *(デフォルト)* なし; 1 - 1; 2 - 2; 3 - 3。
タグ名(形式)	0	tag_name_format	0 - *(デフォルト)* 完全; 1 - 短縮; 2 - なし。 パラメータの動作: - タグの表示が”1”、”2”、または”3” に設定されている場合に サポート
タグ表示優先度	1	tag_priority	タグのカンマ区切りリスト。 パラメータの動作: - タグの表示が”1”、”2”、または”3” に設定されている場合に サポート
運用データの表示	0	show_opdata	0 - *(デフォルト)* なし; 1 - 別々に表示; 2 - 障害名とともに表示。
抑制された障害を表示	0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効。
確認ステータス	0	acknowledgement_status	0 - *(デフォルト)* すべて; 1 - 未確認; 2 - 確認済み。
自分によるもの	0	acknowledged_by_me	0 - *(デフォルト)* 無効; 1 - 有効。

パラメータ	type	name	value
エントリの並べ替え	0	sort_triggers	1 - 深刻度（降順）； 2 - ホスト（昇順）； 3 - 時刻（昇順）； 4 - *(デフォルト)* 時刻（降順）； 13 - 深刻度（昇順）； 14 - ホスト（降順）； 15 - 障害（昇順）； 16 - 障害（降順）。 ”時刻（降順）”および”時刻（昇順）”を除くすべての値では、タイムラインの表示パラメータを”無効”に設定する必要があります。 ”ホスト（昇順）”および”ホスト（降順）”の値は、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
タイムラインの表示	0	show_timeline	0 - 無効; 1 - *(デフォルト)* 有効。 パラメータの動作: - エントリの並べ替えが”時刻（降順）”または”時刻（昇順）”に設定されている場合に サポート
行全体を強調表示	0	highlight_row	0 - *(デフォルト)* 無効; 1 - 有効。 このパラメータは高コントラストテーマではサポートされません。 このオプションは Zabbix 7.4.3 以降でサポートされています。
表示行数	0	show_lines	指定可能な値は 1～100 です。 デフォルト: 25。
Reference	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、障害ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

障害ウィジェットの設定

ホストグループ”4”で、以下の条件を満たす障害を表示する障害ウィジェットを設定します:

- ”scope”という名前のタグが付いており、そのタグの値が”performance”、”availability”または”capacity”である障害。
- 深刻度が”警告”、”軽度の障害”、”重度の障害”、”致命的な障害”である障害。

さらに、タグと運用データを表示するようにウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problems",
                        "name": "Problems",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "performance"
                            },
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "availability"
                            },
                            {
                                "type": 1,
                                "name": "tags.2.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.2.operator",
                                "value": 0
                            },
```

```
                            {
                                "type": 1,
                                "name": "tags.2.value",
                                "value": "capacity"
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            },
                            {
                                "type": 0,
                                "name": "show_tags",
                                "value": 1
                            },
                            {
                                "type": 0,
                                "name": "show_opdata",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
```

```
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

24 深刻度ごとの障害

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで深刻度ごとの障害数ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。深刻度ごとの障害数ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

重大度別の問題ウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 テンプレートダッシュボード でウィジェットを構成する場合、このパラメータはサポートされません。

パラメータ		type	name	value
ホストグループを除外		2	exclude_groupids.0	ホストグループ ID 注: 複数のホストグループを除外するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストグループごとに作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
問題		1	問題	問題イベント名 (大文字と小文字は区別されません、完全な名前またはその一部)。
重大度		0	重大度 0	0 - 分類なし 1 - 情報 2 - 警告 3 - 平均 4 - 高 5 - 災害 デフォルト: 空 (すべて有効)。 注: 複数の値を構成するには、プロパティ名に増分番号を付けた各値のダッシュボードウィジェットフィールドオブジェクトを作成します。
問題タグ				
	評価タイプ	0	評価タイプ	0 - *(デフォルト)* And/Or 2 - Or
	タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名内の数字は、タグ評価リスト内のタグ順序を表します。 パラメータの動作: - 問題タグを構成する場合は 必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名内の数字は、タグ評価リスト内のタグ順序を表します。 パラメータの動作: - 問題タグを構成する場合は 必須

パラメータ		type	name	value
	タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名内の数字は、タグ評価リスト内のタグの順序を表します。 パラメータの動作: - 問題タグを設定する場合は 必須
表示		0	show_type	0 - *(デフォルト)* ホストグループ 1 - 合計。 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合はサポートされず、デフォルトでは” 合計」に設定されています。
レイアウト		0	layout	0 - *(デフォルト)* 水平; 1 - 垂直 パラメータの動作: - 表示が” 合計” に設定されている場合、サポートされます
運用データの表示		0	show_opdata	0 - *(デフォルト)* なし; 1 - 個別に; 2 - 問題名付き
抑制された問題の表示		0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効
問題のないグループを非表示		0	hide_empty_groups	0 - *(デフォルト)* 無効; 1 - 有効 パラメータの動作: - 表示が” ホストグループ」に設定されている場合に サポートされます。 このパラメータは、テンプレートダッシュボード でウィジェットを設定する場合、サポートされません。
問題の表示		0	ext_ack	0 - *(デフォルト)* すべて; 1 - 未確認のみ; 2 - 分離

パラメータ	type	name	value
タイムラインを表示	0	show_timeline	0 - 無効; 1 - *(デフォルト)* 有効
参照	1	参照	5 文字からなる任意の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、深刻度ごとの障害数ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

深刻度ごとの障害数ウィジェットの設定

すべてのホストグループの障害の合計数を表示する深刻度ごとの障害数ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "problemsbysv",
                        "name": "Problems by severity",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "show_type",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
```

```
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

25 SLA レポート

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでSLA レポートウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。*SLA* レポートウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

SLA レポートウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - *(デフォルト)* 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - 15 分
SLA	10	slaid.0	SLA ID。 パラメーターの動作: - 必須
サービス	9	serviceid.0	サービスID。
表示期間	0	show_periods	有効な値の範囲は 1～100 です。 デフォルト: 20

パラメーター	type	name	value
From	1	date_from	YYYY-MM-DD 形式の有効な日付文字列。 修飾子 d、w、M、y(例: now、now/d、now/w-1w など) を使用した相対日付がサポートされています。
To	1	date_to	YYYY-MM-DD 形式の有効な日付文字列。 修飾子 d、w、M、y(例: now、now/d、now/w-1w など) を使用した相対日付がサポートされています。

例

次の例は、*SLA* レポートウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

SLA レポートウィジェットの設定

SLA "4"、サービス"2" の過去 30 日間の SLA レポートを表示する *SLA* レポートウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "slareport",
                        "name": "SLA report",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 10,
                                "name": "slaid.0",
                                "value": 4
                            },
                            {
                                "type": 9,
                                "name": "serviceid.0",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "date_from",
                                "value": "now-30d"
                            },
                            {
                                "type": 1,
                                "name": "date_to",
                                "value": "now"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
```

```
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

26 システム情報

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでシステム情報ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。システム情報ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

システム情報ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - *(デフォルト)* 15 分
表示	0	info_type	0 - *(デフォルト)* システムステータス 1 - HA ノード

パラメーター	type	name	value
ソフトウェア更新チェックの詳細を表示	0	show_software_update_check_details	0 - (デフォルト) 無効 1 - 有効 パラメーターの動作: - Zabbix サーバー設定のAllowSoftwareUpdateCheckが有効で、表示が” システムステータス” に設定されている場合、サポート

例

次の例は、システム情報ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

システム情報ウィジェットの設定

10 分間隔で更新し、ソフトウェアアップデートのチェックを有効にした状態でシステム統計情報を表示するシステム情報ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "systeminfo",
                        "name": "System information",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 0,
                                "name": "rf_rate",
                                "value": 600
                            },
                            {
                                "type": 0,
                                "name": "show_software_update_check_details",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

27 上位ホスト

説明

これらのパラメータおよび各ダッシュボードウィジェットフィールドオブジェクトに対応するプロパティ値を使用すると、dashboard.create および dashboard.update メソッドで*Top hosts* ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。
これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、一方でウィジェットを誤って作成または更新してしまうリスクも生じます。
Top hosts ウィジェットを正常に作成または更新するには、以下の表に示されているパラメータの動作を参照してください。

Parameters

上位ホストウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメーター		type	name	value
	ホストグループ (ウィジェット)	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストタグ				
	評価タイプ	0	evaltype	0 - (デフォルト) And/Or; 2 - Or. このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	タグ名	1	tags.0.tag	任意の文字列値。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。

パラメーター		type	name	value
	タグ値	1	tags.0.value	任意の文字列値。 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - ホストタグを構成する場合は 必須。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
メンテナンス中のホストを表示		0	メンテナンス	0 - *(デフォルト)* 無効。 1 - 有効
列(下記参照)		0	列	構成された列の数値
並べ替え順序		0	列	構成された列の数値
順序		0	順序	2 - *(デフォルト)* 上位 N 件 3 - 下位 N 件
ホスト制限		0	show_lines	可能な値の範囲は 1 ～1000 です。 デフォルト: 10。 テンプレートダッシュボード でウィジェットを構成する場合、このパラメーターはサポートされません。

列

列には、データパラメーターの設定に応じて、共通パラメーターと追加パラメーターがあります。

Note:
列に関連するすべてのパラメーターにおいて、プロパティ名に含まれる番号 (例: columns.0.name) は、パラメーターが設定されている列を参照します。

以下のパラメーターはすべての列でサポートされています。

パラメーター	type	name	value
Name	1	columns.0.name	任意の文字列値 パラメーターの動作: - 必須
Data	0	columns.0.data	1 - 項目値 2 - ホスト名 3 - テキスト パラメーターの動作: - 必須
基本色	1	columns.0.base_color	16 進数カラーコード（例：FF0000） パラメータの動作: - 必須

アイテムの値

データが「アイテムの値」に設定されている場合、以下のパラメータがサポートされます。

Note:
しきい値プロパティ名の最初の数字（例: columnsthresholds.0.color.0）は、しきい値が設定されている列を参照し、2 番目の数字は昇順で並べ替えられたリスト内のしきい値の位置を参照します。ただし、しきい値が異なる順序で設定されている場合、Zabbix Web インターフェースでウィジェット設定を更新した後、値は昇順に並べ替えられます (例: `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`)。

Parameter	type	name	value
アイテム	1	columns.0.item	有効なアイテム名。 テンプレートダッシュボード でウィジェットを設定する場合は、テンプレート上で設定されたアイテムのみを指定してください。
アイテムの値の表示形式	0	columns.0.display_value_as	0 - (デフォルト) 数値; 1 - テキスト; 2 - バイナリ。
表示	0	columns.0.display	1 - (デフォルト) そのまま; 2 - バー; 3 - インジケーター; 6 - スパークライン。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定されている場合
最小	1	columns.0.min	任意の数値。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「バー」または「インジケーター」に設定されている場合
最大	1	columns.0.max	任意の数値。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「バー」または「インジケーター」に設定されている場合

Parameter	type	name	value
小数点以下の桁数スパークライン	0	columns.0.decimal_places	指定可能な値の範囲は 0～10 です。 デフォルト: 2。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定されている場合
幅	0	columns.0.sparkline.width	指定可能な値の範囲は 0～10 です。 デフォルト: 1。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合
塗りつぶし	0	columns.0.sparkline.fill	指定可能な値の範囲は 0～10 です。 デフォルト: 3。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合
色	1	columns.0.sparkline.color	16進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合
期間	1	columns.0.sparkline.time_period.from	DASHBOARD._timeperiod - 期間セレクター をデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および 終了パラメータでのみ期間を設定することもできます。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合
開始	1	columns.0.sparkline.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間 構文（now, now/d, now/w-1w など）による有効な時刻文字列。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合 - 必須 columns.0.sparkline.time_period.to が設定されている場合

Parameter	type	name	value
	終了	1	columns.0.sparkline.time_period.to 絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間 構文（now, now/d, now/w-1w など）による有効な時刻文字列。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合
	履歴データ	0	columns.0.sparkline.history (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定され、かつ表示が「スパークライン」に設定されている場合
しきい値	色	1	columnsthresholds.0.color 16進数のカラーコード（例: FF0000）。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定されている場合
	しきい値	1	columnsthresholds.0.threshold 任意の文字列値。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定されている場合
ハイライト	色	1	columns.0.highlights.0.color 16進数のカラーコード（例: FF0000）。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「テキスト」に設定されている場合
	パターン	1	columns.0.highlights.0.pattern 任意の文字列値。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「テキスト」に設定されている場合
サムネイルを表示		0	columns.0.show_thumbnail (デフォルト) 無効; 1 - 有効。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「バイナリ」に設定されている場合
集計関数		0	columns.0.aggregate_function (デフォルト) 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last。

Parameter	type	name	value
期間	1	columns.0.time_period.filter	DASHBOARD._timeperiod - 期間セレクター をデータソースとして設定します。 ABCDE._timeperiod - 互換性のあるウィジェット（reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および 終了パラメータでのみ期間を設定することもできます。 パラメータの動作: - サポート対象 集計関数が「min」、「max」、「avg」、「count」、「sum」、「first」、「last」に設定されている場合
開始	1	columns.0.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間 構文（now, now/d, now/w-1w など）による有効な時刻文字列。 パラメータの動作: - サポート対象 期間が設定されておらず、かつ 集計関数が「min」、「max」、「avg」、「count」、「sum」、「first」、「last」に設定されている場合 - 必須 columns.0.time_period.to が設定されている場合
終了	1	columns.0.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間 構文（now, now/d, now/w-1w など）による有効な時刻文字列。 パラメータの動作: - サポート対象 期間が設定されておらず、かつ 集計関数が「min」、「max」、「avg」、「count」、「sum」、「first」、「last」に設定されている場合 - 必須 columns.0.time_period.from が設定されている場合
履歴データ	0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - サポート対象 アイテムの値の表示形式が「数値」に設定されている場合
参照	1	reference	5 文字で構成される任意の文字列値（例: ABCDE または JBPNL）。この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

Text

データが” テキスト” に設定されている場合、以下のパラメーターがサポートされます。

パラメーター	type	name	value
テキスト	1	columns.0.text	マクロを含む任意の文字列値 サポートされているマクロ: {HOST.*}、{INVENTORY.*} パラメーターの動作: - データがテキスト” に設定されている場合、必須

例

次の例は、上位ホストウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

上位ホストウィジェットの設定

ホストグループ”4” で CPU 使用率上位のホストを表示する上位ホストウィジェットを構成します。さらに、次のカスタム列を構成します: ”Host name”, ”CPU utilization in %”, ”1m avg”, ”5m avg”, ”15m avg”, ”Processes”。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "tophosts",
                        "name": "Top hosts",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 1,
                                "name": "columns.0.name",
                                "value": "Host"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.data",
                                "value": 2
                            },
                            {
                                "type": 1,
                                "name": "columns.0.base_color",
                                "value": "FFFFFF"
                            },
                            {
                                "type": 1,
                                "name": "columns.1.name",
                                "value": "CPU utilization in %"
                            },
                            {
                                "type": 0,
                                "name": "columns.1.data",
                                "value": 1
                            },
                            {
                                "type": 1,
                                "name": "columns.1.base_color",
                                "value": "4CAF50"
                            },
                            {
                                "type": 1,
                                "name": "columns.1.item",
```

```
            "value": "CPU utilization"
        },
        {
            "type": 0,
            "name": "columns.1.display",
            "value": 3
        },
        {
            "type": 1,
            "name": "columns.1.min",
            "value": "0"
        },
        {
            "type": 1,
            "name": "columns.1.max",
            "value": "100"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.0",
            "value": "FFFF00"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.0",
            "value": "50"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.1",
            "value": "FF8000"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.1",
            "value": "80"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.color.2",
            "value": "FF4000"
        },
        {
            "type": 1,
            "name": "columnsthresholds.1.threshold.2",
            "value": "90"
        },
        {
            "type": 1,
            "name": "columns.2.name",
            "value": "1m avg"
        },
        {
            "type": 0,
            "name": "columns.2.data",
            "value": 1
        },
        {
            "type": 1,
            "name": "columns.2.base_color",
            "value": "FFFFFF"
        },
```

```
                {
                    "type": 1,
                    "name": "columns.2.item",
                    "value": "Load average (1m avg)"
                },
                {
                    "type": 1,
                    "name": "columns.3.name",
                    "value": "5m avg"
                },
                {
                    "type": 0,
                    "name": "columns.3.data",
                    "value": 1
                },
                {
                    "type": 1,
                    "name": "columns.3.base_color",
                    "value": "FFFFFF"
                },
                {
                    "type": 1,
                    "name": "columns.3.item",
                    "value": "Load average (5m avg)"
                },
                {
                    "type": 1,
                    "name": "columns.4.name",
                    "value": "15m avg"
                },
                {
                    "type": 0,
                    "name": "columns.4.data",
                    "value": 1
                },
                {
                    "type": 1,
                    "name": "columns.4.base_color",
                    "value": "FFFFFF"
                },
                {
                    "type": 1,
                    "name": "columns.4.item",
                    "value": "Load average (15m avg)"
                },
                {
                    "type": 1,
                    "name": "columns.5.name",
                    "value": "Processes"
                },
                {
                    "type": 0,
                    "name": "columns.5.data",
                    "value": 1
                },
                {
                    "type": 1,
                    "name": "columns.5.base_color",
                    "value": "FFFFFF"
                },
                {
                    "type": 1,
```

```
                                "name": "columns.5.item",
                                "value": "Number of processes"
                            },
                            {
                                "type": 0,
                                "name": "columns.5.decimal_places",
                                "value": 0
                            },
                            {
                                "type": 0,
                                "name": "column",
                                "value": 1
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

28 Top items

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで上位アイテムウィジェットを設定できます。

Attention:
ウィジェットの `fields` プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。上位アイテムウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

トップアイテムウィジェットでは、以下のパラメータがサポートされています。

パラメータ		type	name	value
更新間隔		0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ		2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: `ABCDE._hostgroupids` の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストタグ				
	評価タイプ	0	host_tags_evaltype	0 - *(デフォルト)* And/Or; 2 - Or. このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。

パラメータ		type	name	value
	タグ名	1	host_tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。
	演算子	0	host_tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
	タグ値	1	host_tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグ順序を参照します。 パラメーターの動作: - ホストタグを構成する場合は 必須 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
レイアウト		0	レイアウト	0 - *(デフォルト)* 水平 1 - 垂直
問題の表示項目		0	show_problems	0 - すべて 1 - *(デフォルト)* 抑制なし 2 - なし
(以下を参照)				

Advanced configuration

上位アイテムウィジェットでは、以下の詳細設定パラメータがサポートされています。

パラメータ		type	name	value
ホストの順序並べ替え		0	host_ordering_order_by	0 - *(default)* ホスト名; 3 - アイテムの値
	アイテムパターン	1	host_ordering_item.0	アイテム の名前またはパターン（例：”*: 1 秒あたりに処理される * 値の数”） 注: 複数のアイテムパターンを設定するには、プロパティ名に増分番号を含む各アイテムパターンのダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボード でウィジェットを設定する場合は、テンプレートで構成されているアイテムのパターンのみを設定する必要があります。 パラメータの動作: - *Order by* が”Item value” に設定されている場合、必須
Order		0	host_ordering_order	2 - *(デフォルト)* 上位 N 件; 3 - 下位 N 件
制限		0	host_ordering_limit	有効な値の範囲は 1～100 です。 デフォルト: 10
アイテムの順序順序付け		0	item_ordering_order_by	1 - ホスト 2 - アイテム名 3 - *(デフォルト)* アイテムの値
	ホストパターン	1	item_ordering_host.0	ホスト の名前またはパターン 注: 複数のホストパターンを設定するには、プロパティ名に増分番号を付けた、ホストパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 パラメータの動作: - 順序付けが”Host” に設定されている場合は 必須
順序制限		0	item_ordering_order	2 - *(デフォルト)* 上位 N 件; 3 - 下位 N 件
		0	item_ordering_limit	指定可能な値の範囲は 1 ～100 です。 デフォルト: 10 件

列

Note:
プロパティ名の最初の数字（例: columns.0.items.0、columns.0.item_tags_evaltype）は特定の列を表し、2 番目の数字が存在する場合は設定されたエンティティ（例: アイテムパターン、タグ）を表します。

パラメータ		type	name	value
アイテムパターン		1	columns.0.items.0	アイテム名またはパターン（例: ”*: 1 秒あたりに処理された * 値の数”）。 注意: 複数のアイテムパターンを設定するには、プロパティ名の 2 番目の数字を増やして、各アイテムパターンごとにダッシュボードウィジェットフィールドオブジェクトを作成します。 テンプレートダッシュボードでウィジェットを設定する場合は、テンプレート上で設定されたアイテムのパターンのみを設定してください。 パラメータの動作: - 必須
アイテムタグ				
	評価タイプ	0	columns.0.item_tags.0.evaltype	0 - (デフォルト) And/Or; 2 - Or.
	タグ名	1	columns.0.item_tags.0.tag	任意の文字列値。 注意: プロパティ名の 2 番目の数字は、タグ評価リスト内のタグ順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須
	演算子	0	columns.0.item_tags.0.operator	0 - (デフォルト) 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注意: プロパティ名の 2 番目の数字は、タグ評価リスト内のタグ順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須
	タグ値	1	columns.0.item_tags.0.value	任意の文字列値。 注意: プロパティ名の 2 番目の数字は、タグ評価リスト内のタグ順序を参照します。 パラメータの動作: - アイテムタグを設定する場合は必須
ベース色		1	columns.0.base_color	16 進数のカラーコード（例: FF0000）。
値の表示形式		0	columns.0.display_value_as	1 - (デフォルト) 数値; 2 - テキスト。

パラメータ		type	name	value
表示		0	columns.0.display	1 - *(デフォルト)* そのまま; 2 - バー; 3 - インジケーター; 6 - スパークライン。
最小		1	columns.0.min	任意の数値。接尾辞（例: "1d"、"2w"、"4K"、"8G"）をサポートします。 パラメータの動作: - 値の表示形式が" 数値" に設定され、表示が" バー" または" インジケーター" に設定されている場合にサポートされます
最大		1	columns.0.max	任意の数値。接尾辞（例: "1d"、"2w"、"4K"、"8G"）をサポートします。 パラメータの動作: - 値の表示形式が" 数値" に設定され、表示が" バー" または" インジケーター" に設定されている場合にサポートされます
スパークライン				
	幅	0	columns.0.sparkline.width	指定可能な値の範囲は 0～10 です。 デフォルト: 1。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます
	塗りつぶし	0	columns.0.sparkline.fill	指定可能な値の範囲は 0～10 です。 デフォルト: 3。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます
	色	1	columns.0.sparkline.color	16進数のカラーコード（例: FF0000）。 デフォルト: 42A5F5。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます
	期間	1	columns.0.sparkline.time_period	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（その reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメータでのみ期間を設定することもできます。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます

パラメータ		type	name	value
	開始	1	columns.0.sparkline.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます - columns.0.sparkline.time_period.to が設定されている場合は必須
	終了	1	columns.0.sparkline.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます - columns.0.sparkline.time_period.from が設定されている場合は必須
	履歴データ	0	columns.0.sparkline.history	0 - (デフォルト) 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - 表示が" スパークライン" に設定されている場合にサポートされます
しきい値	色	1	columns.0.thresholds.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - 値の表示形式が" 数値" に設定されている場合にサポートされます
	しきい値	1	columns.0.thresholds.value	任意の数値。 [接尾辞](#)（例: "1d"、"2w"、"4K"、"8G"）をサポートします。 パラメータの動作: - 値の表示形式が" 数値" に設定されている場合にサポートされます
ハイライト	ハイライト	1	columns.0.highlights.color	16 進数のカラーコード（例: FF0000）。 パラメータの動作: - 値の表示形式が" テキスト" に設定されている場合にサポートされます
	しきい値	1	columns.0.highlights.pattern	任意の正規表現。 パラメータの動作: - 値の表示形式が" テキスト" に設定されている場合にサポートされます
小数点以下桁数		0	columns.0.decimal_places	指定可能な値の範囲は 0～10 です。 デフォルト: 2。

パラメータ		type	name	value
高度な設定				
集計関数		0	columns.0.aggregate_function	値の表示形式が” 数値” に設定されている場合: 0 - *(デフォルト)* 使用しない; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last. 値の表示形式が” テキスト” に設定されている場合: 0 - *(デフォルト)* 使用しない; 4 - count; 6 - first; 7 - last.
期間		1	columns.0.time_period.reference	DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（その reference パラメータが ABCDE に等しいもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および終了パラメータでのみ期間を設定することもできます。 パラメータの動作: - 集計関数が”min”、”max”、”avg”、”count”、”sum”、”first”、”last” に設定されている場合にサポートされます
	開始	1	columns.0.time_period.from	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が”min”、”max”、”avg”、”count”、”sum”、”first”、”last” に設定されている場合にサポートされます - columns.0.time_period.to が設定されている場合は必須
	終了	1	columns.0.time_period.to	絶対時間（YYYY-MM-DD hh:mm:ss）または相対時間構文（now、now/d、now/w-1w など）の有効な時間文字列。 パラメータの動作: - 期間が設定されておらず、集計関数が”min”、”max”、”avg”、”count”、”sum”、”first”、”last” に設定されている場合にサポートされます - columns.0.time_period.from が設定されている場合は必須
履歴データ		0	columns.0.history	0 - *(デフォルト)* 自動; 1 - 履歴; 2 - トレンド。 パラメータの動作: - 値の表示形式が” 数値” に設定されている場合にサポートされます

例

次の例は、*Top items* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

トップアイテムウィジェットの設定

ホスト"10084" のデータと、名前が"component" のタグに値が"cpu" を含むアイテムのみを表示する トップアイテムウィジェットを設定します。さらに、ホストを上位に表示し、セルを表すために色付きのゲージバーを使用します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "topitems",
                        "name": "Top items",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 3,
                                "name": "hostids.0",
                                "value": 10084
                            },
                            {
                                "type": 1,
                                "name": "columns.0.items.0",
                                "value": "*"
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.tag",
                                "value": "component"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.item_tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "columns.0.item_tags.0.value",
                                "value": "cpu"
                            },
                            {
                                "type": 0,
                                "name": "columns.0.display",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "layout",
                                "value": 1
                            }
                        ]
```

```
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

29 上位トリガー

説明

これらのパラメータおよび対応するダッシュボードウィジェットフィールドオブジェクトで使用可能なプロパティ値により、`dashboard.create` および `dashboard.update` メソッドで*Top triggers* ウィジェットを設定できます。

Attention:
ダッシュボードの作成または更新時に、ウィジェットの `fields` プロパティは検証されません。これにより、ユーザーは組み込みウィジェット を変更したり、カスタムウィジェット を作成したりできますが、一方でウィジェットを誤って作成または更新してしまうリスクも生じます。*Top triggers* ウィジェットを正常に作成または更新するには、以下の表に示されたパラメータの動作を参照してください。

パラメータ

Top triggers ウィジェットでは、以下のパラメータがサポートされています。

パラメータ		type	name	value
更新間隔		0	rf_rate	0 - *(デフォルト)* 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分。
ホストグループ		2	groupids.0	ホストグループ ID。 注意: 複数のホストグループを設定するには、プロパティ名の番号を増やして、ホストグループごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
ホスト		3	hostids.0	ホスト ID。 注意: 複数のホストを設定するには、プロパティ名の番号を増やして、ホストごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。複数のホストを指定する場合、ホストグループパラメータはまったく設定しないか、設定したホストが属するホストグループを少なくとも 1 つ設定する必要があります。 このパラメータは、テンプレートダッシュボード 上でウィジェットを設定する場合はサポートされません。
障害		1	problem	障害のイベント名（大文字と小文字を区別しない、完全一致または部分一致）。
深刻度		0	severities.0	トリガーの深刻度。 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 致命的。 デフォルト: 空欄（すべて有効）。 注意: 複数の値を設定するには、プロパティ名の番号を増やして、値ごとにダッシュボードウィジェットフィールドオブジェクトを作成してください。
障害タグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or。
	タグ名	1	tags.0.tag	任意の文字列値。 注意: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。 パラメータの動作: - 障害タグを設定する場合は 必須

パラメータ		type	name	value
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。 注意: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。
	タグ値	1	tags.0.value	パラメータの動作: - 障害タグを設定する場合は 必須 任意の文字列値。 注意: プロパティ名の番号は、タグ評価リスト内のタグの順序を参照します。
期間		1	time_period._reference	パラメータの動作: - 障害タグを設定する場合は 必須 DASHBOARD._timeperiod - 期間セレクターをデータソースとして設定します; ABCDE._timeperiod - 互換性のあるウィジェット（その *Reference* パラメータが"ABCDE" に設定されているもの）をデータソースとして設定します。 デフォルト: DASHBOARD._timeperiod または、開始および 終了パラメータでのみ期間を設定することもできます。
	開始	1	time_period.from	絶対時間 (YYYY-MM-DD hh:mm:ss) または相対 時間構文 (now, now/d, now/w-1w など) の有効な時刻文字列。 パラメータの動作: - 期間が設定されていない場合に サポート - time_period.to が設定されている場合は 必須
	終了	1	time_period.to	絶対時間 (YYYY-MM-DD hh:mm:ss) または相対 時間構文 (now, now/d, now/w-1w など) の有効な時刻文字列。 パラメータの動作: - 期間が設定されていない場合に サポート - time_period.from が設定されている場合は 必須
トリガー表示数		0	show_lines	指定可能な値は 1～100 です。 デフォルト: 10。

例

次の例は、上位トリガーウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

Top triggers ウィジェットの設定

ホストグループ「4」の上位 5 件のトリガーと、各トリガーに対するすべての障害の件数を表示する *Top triggers* ウィジェットを設定します。

このウィジェットには、深刻度が「Warning」、「Average」、「High」、または「Disaster」のトリガーのみが表示され、タグ名「scope」を持ち、その値に「performance」、「availability」、または「capacity」を含むタグが付いた障害のみが表示されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "toptriggers",
                        "name": "Top triggers",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.0",
                                "value": 2
                            },
                            {
                                "type": 0,
                                "name": "severities.1",
                                "value": 3
                            },
                            {
                                "type": 0,
                                "name": "severities.2",
                                "value": 4
                            },
                            {
                                "type": 0,
                                "name": "severities.3",
                                "value": 5
                            },
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "performance"
                            },
```

```
                            {
                                "type": 1,
                                "name": "tags.1.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.1.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.1.value",
                                "value": "availability"
                            },
                            {
                                "type": 1,
                                "name": "tags.2.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.2.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.2.value",
                                "value": "capacity"
                            },
                            {
                                "type": 0,
                                "name": "show_lines",
                                "value": 5
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
```

```
            ]
        },
        "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

30 トリガーの概要

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドでトリガーの概要ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。トリガーの概要ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

Parameters

トリガー概要ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(デフォルト)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
表示	0	表示	1 - *(デフォルト)* 最近発生した問題; 2 - すべて; 3 - 問題
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストグループ *(ウィジェット)*	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合、サポートされません。

パラメーター		type	name	value
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
問題のあるタグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or
	タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 問題タグを構成する場合は 必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 問題のあるタグを構成する場合は必須
	タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - 問題のあるタグを構成する場合は必須
抑制された問題を表示		0	show_suppressed	0 - *(デフォルト)* 無効; 1 - 有効

パラメーター	type	name	value
レイアウト	0	layout	0 - *(デフォルト)* 水平; 1 - 垂直

例

次の例は、トリガーの概要ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

トリガーの概要ウィジェットの設定

”scope” という名前で値に”availability” を含むタグでトリガーされ、すべてのホストグループのトリガー状態を表示するトリガー概要ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Trigger overview",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "tags.0.tag",
                                "value": "scope"
                            },
                            {
                                "type": 0,
                                "name": "tags.0.operator",
                                "value": 0
                            },
                            {
                                "type": 1,
                                "name": "tags.0.value",
                                "value": "availability"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
```

```
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

31 URL

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで*URL*ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。*URL* ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメーター

URL ウィジェットでは、以下のパラメーターがサポートされています。

パラメーター	type	name	value
リフレッシュ間隔	0	rf_rate	0 - *(デフォルト)* 更新なし 10 - 10 秒 30 - 30 秒 60 - 1 分 120 - 2 分 600 - 10 分 900 - 15 分
URL	1	url	有効な URL 文字列。 パラメーターの動作: - 必須

パラメーター	type	name	value
ホストの上書き	1	override_hostid._reference	ABCDE._hostid - 互換性のあるウィジェット(その *Reference* パラメーターを"ABCDE" に設定) をホストのデータソースとして設定します。 DASHBOARD._hostid - ダッシュボードのホストセレクタをホストのデータソースとして設定します。 このパラメーターは、テンプレートダッシュボードでウィジェットを構成する場合にはサポートされません。

例

次の例は、*URL* ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、dashboard.createを参照してください。

URL ウィジェットの設定

Zabbix マニュアルのホームページを表示する *URL* ウィジェットを設定します。

Request:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "url",
                        "name": "URL",
                        "x": 0,
                        "y": 0,
                        "width": 36,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 1,
                                "name": "url",
                                "value": "https://www.zabbix.com/documentation/7.4/en"
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- dashboard.create
- dashboard.update

32 Web monitoring

説明

これらのパラメーターと、各ダッシュボードウィジェットフィールドオブジェクトの有効なプロパティ値を使用することで、dashboard.create メソッドおよび dashboard.update メソッドで*Web* 監視ウィジェットを設定できます。

Attention:
ウィジェットの fields プロパティは、ダッシュボードの作成または更新時に検証されません。これにより、ユーザーは組み込みウィジェットを変更したり、カスタムウィジェットを作成したりできますが、ウィジェットを誤って作成または更新してしまうリスクも生じます。*Web* 監視ウィジェットを正常に作成または更新するには、以下の表に記載されているパラメーターの動作を参照してください。

パラメータ

Web モニタリングウィジェットでは、以下のパラメータがサポートされています。

パラメータ	type	name	value
更新間隔	0	rf_rate	0 - 更新なし; 10 - 10 秒; 30 - 30 秒; 60 - *(*デフォルト*)* 1 分; 120 - 2 分; 600 - 10 分; 900 - 15 分
ホストグループ	2	groupids.0	ホストグループ ID 注: 複数のホストグループを構成するには、プロパティ名に増分番号を含む各ホストグループのダッシュボードウィジェットフィールドオブジェクトを作成します。 このパラメーターは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
ホストグループ *(*ウィジェット*)*	1	groupids._reference	ホストグループ ID: ABCDE._hostgroupids の代わりに、ホストグループのデータソースとして互換性のあるウィジェット (*Reference* パラメータを"ABCDE に設定) を設定します。 テンプレートダッシュボード でウィジェットを構成する場合、このパラメータはサポートされません。

パラメータ		type	name	value
ホストグループを除外		2	exclude_groupids.0	ホストグループ ID 注: 複数のホストグループを除外するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストグループごとに作成します。 このパラメーターは、テンプレートダッシュボードでウィジェットを構成する場合、サポートされません。
ホスト		3	hostids.0	ホスト ID 注: 複数のホストを構成するには、プロパティ名に増分番号を含むダッシュボードウィジェットフィールドオブジェクトをホストごとに作成します。複数のホストの場合、パラメータ ホストグループはまったく構成されていないか、構成されたホストが属する少なくとも 1 つのホストグループで構成されている必要があります。 このパラメータは、テンプレートダッシュボード でウィジェットを構成する場合はサポートされません。
シナリオタグ				
	評価タイプ	0	evaltype	0 - *(デフォルト)* And/Or; 2 - Or
	タグ名	1	tags.0.tag	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - シナリオタグを構成する場合は必須
	演算子	0	tags.0.operator	0 - 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - シナリオタグを構成する場合は 必須
	タグ値	1	tags.0.value	任意の文字列値 注: プロパティ名の数字は、タグ評価リスト内のタグの順序を示します。 パラメータの動作: - シナリオタグを構成する場合は 必須

パラメータ	type	name	value
メンテナンス中のホストを表示	0	メンテナンス	0 - 無効; 1 - *(デフォルト)* 有効
参照	1	参照	5 文字からなる任意の文字列値（例：ABCDE または JBPNL）この値は、ウィジェットが属するダッシュボード内で一意である必要があります。 パラメータの動作: - 必須

例

次の例は、*Web* 監視ウィジェットのダッシュボードウィジェットフィールドオブジェクトの設定を説明することだけを目的としています。ダッシュボードの設定の詳細については、`dashboard.create`を参照してください。

Web 監視ウィジェットの設定

ホストグループ"4" のアクティブな Web 監視シナリオのステータス概要を表示する *Web* 監視ウィジェットを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "web",
                        "name": "Web monitoring",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
```

```
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

参照

- Dashboard widget field
- `dashboard.create`
- `dashboard.update`

テンプレート

このクラスはテンプレートを操作するために設計されています。

オブジェクト参照:

- テンプレート
- テンプレートタグ

利用可能なメソッド:

- template.create - 新しいテンプレートを作成
- template.delete - テンプレートを削除
- template.get - テンプレートを取得
- template.massadd - 関連オブジェクトをテンプレートに追加
- template.massremove - 関連オブジェクトをテンプレートから削除
- template.massupdate - テンプレート内の関連オブジェクトを置換または削除
- template.update - テンプレートを更新

Template オブジェクト

次のオブジェクトは、`templateAPI` に直接関連しています。

テンプレート

テンプレートオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
templateid	ID	テンプレートの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
description	text	テンプレートの説明。
host	string	テンプレートの技術名。 プロパティの動作: - 作成操作では 必須
name	string	テンプレートの表示名。 デフォルト: `host` プロパティの値。
readme	text	ホストウィザード に表示する、テンプレート固有の設定手順です。 Markdown 形式をサポートします。
uuid	string	既存のテンプレートにインポートしたテンプレートを関連付けるために使用される、汎用一意識別子です。指定されていない場合は自動生成されます。
vendor_name	string	テンプレートのベンダー名。 作成操作では、`vendor_name` と `vendor_version` の両方を設定するか、両方とも空のままにする必要があります。更新操作では、データベースに値がある場合、`vendor_version` は空のままにできます。
vendor_version	string	テンプレートのベンダーバージョン。 作成操作では、`vendor_name` と `vendor_version` の両方を設定するか、両方とも空のままにする必要があります。更新操作では、データベースに値がある場合、`vendor_name` は空のままにできます。
wizard_ready	integer	テンプレートをホストウィザード で選択可能にするかどうか。 指定可能な値: 0 - *(デフォルト)* 利用不可。 1 - 利用可能。

テンプレートタグ

テンプレートタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	テンプレートタグ名。 プロパティの動作: - 必須
value	string	テンプレートタグの値。

template.create

説明

`object template.create(object/array templates)`

このメソッドでは、新しいテンプレートを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

`(object/array)` 作成するテンプレート。

標準のテンプレートプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
groups	object/array	テンプレートを追加するテンプレートグループ。 テンプレートグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
tags	object/array	テンプレートタグ。
templates	object/array	テンプレートにリンクするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。
macros	object/array	テンプレート用に作成するユーザーマクロ。

戻り値

(object) templateids プロパティの下で作成されたテンプレートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートの順序と一致します。

例

テンプレートの作成

タグ付きのテンプレートを作成し、このテンプレートに 2 つのテンプレートをリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.create",
    "params": {
        "host": "Linux template",
        "groups": {
            "groupid": 1
        },
        "templates": [
            {
                "templateid": "11115"
            },
            {
                "templateid": "11116"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::create() in *ui/include/classes/api/services/CTemplate.php*.

template.delete

説明

`object template.delete(array templateIds)`

このメソッドは、テンプレートを削除することができます。

テンプレートを削除すると、すべてのテンプレートエンティティ (アイテム、トリガー、グラフなど) が削除されます。テンプレートエンティティをホストに残しつつ、テンプレート自体を削除するにはtemplate.update、template.massupdate、host.update、host.massupdateのいずれかを使用して、ホストからテンプレートのリンクを解除します。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するテンプレートの ID。

戻り値

(object) templateids プロパティの下にある削除されたテンプレートの ID を含むオブジェクトを返します。

例

複数のテンプレートの削除

2 つのテンプレートを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::delete() in *ui/include/classes/api/services/CTemplate.php*.

template.get

説明

`integer/array template.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってテンプレートを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
templateids	ID/array	指定したテンプレート ID を持つテンプレートのみを返します。
groupids	ID/array	指定したテンプレートグループに属するテンプレートのみを返します。
parentTemplateids	ID/array	指定したテンプレートがリンクしているテンプレートのみを返します。
hostids	ID/array	指定したホスト/テンプレートにリンクしているテンプレートのみを返します。
graphids	ID/array	指定したグラフを含むテンプレートのみを返します。
itemids	ID/array	指定したアイテムを含むテンプレートのみを返します。
triggerids	ID/array	指定したトリガーを含むテンプレートのみを返します。
with_items	flag	アイテムを持つテンプレートのみを返します。
with_triggers	flag	トリガーを持つテンプレートのみを返します。
with_graphs	flag	グラフを持つテンプレートのみを返します。
with_httptests	flag	Web シナリオを持つテンプレートのみを返します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	object/array	指定したタグを持つテンプレートのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのテンプレートを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
selectTags	query	テンプレートタグを`tags` プロパティで返します。
selectDiscoveryRules	query	テンプレートの LLD ルールを含む`discoveryRules` プロパティを返します。 `count` をサポートします。
selectHosts	query	テンプレートにリンクしているホストを`hosts` プロパティで返します。 `count` をサポートします。
selectTemplateGroups	query	テンプレートが属するテンプレートグループを`templategroups` プロパティで返します。
selectTemplates	query	指定したテンプレートがリンクしているテンプレートを`templates` プロパティで返します。 `count` をサポートします。
selectParentTemplates	query	指定したテンプレートにリンクしているテンプレートを`parentTemplates` プロパティで返します。 `count` をサポートします。
selectHttpTests	query	テンプレートの Web シナリオを`httpTests` プロパティで返します。 `count` をサポートします。

Parameter	Type	Description
selectItems	query	テンプレートのアイテムを`items` プロパティで返します。 `count` をサポートします。
selectTriggers	query	テンプレートのトリガーを`triggers` プロパティで返します。 `count` をサポートします。
selectGraphs	query	テンプレートのグラフを`graphs` プロパティで返します。 `count` をサポートします。
selectMacros	query	テンプレートのマクロを`macros` プロパティで返します。
selectDashboards	query	テンプレートのダッシュボードを`dashboards` プロパティで返します。 `count` をサポートします。
selectValueMaps	query	テンプレートの値マップを含む`valuemaps` プロパティを返します。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectTemplates` - 結果は `name` でソートされます; `selectHosts` - `host` でソートされます; `selectParentTemplates` - `host` でソートされます; `selectItems` - `name` でソートされます; `selectDiscoveryRules` - `name` でソートされます; `selectTriggers` - `description` でソートされます; `selectGraphs` - `name` でソートされます; `selectDashboards` - `name` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `hostid`, `host`, `name`, `status`.
countOutput	boolean	これらのパラメータについては、リファレンス解説を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	テンプレートのローレベルディスカバリを `discoveries` プロパティで返します。 `count` をサポートします。 このクエリは非推奨です。代わりに `selectDiscoveryRules` を使用してください。

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

名前でテンプレートを取得する

”Linux” および”Windows” という名前の 2 つのテンプレートに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxyid": "0",
            "host": "Linux by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux by Zabbix agent",
            "flags": "0",
            "templateid": "10001",
            "description": " これは公式の Linux テンプレートです。Zabbix agent 7.4 以降が必要です。\r\n\r\nファイル
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "f8f7908280354f2abeed07dc788c3747",
            "vendor_name": "Zabbix",
            "vendor_version": "7.4-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## 概要\r\n\r\nこれは公式の Linux テンプレートです。Zabbix agent 7.4 以降が必要です。\r\n\r\
        },
        {
            "proxyid": "0",
            "host": "Windows by Zabbix agent",
            "status": "3",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
```

```
            "name": "Windows by Zabbix agent",
            "flags": "0",
            "templateid": "10081",
            "description": " これは公式の Windows テンプレートです。Zabbix agent 7.4 以降が必要です。\r\n\r\nこのテ
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "tls_psk_identity": "",
            "tls_psk": "",
            "custom_interfaces": "0",
            "uuid": "13b06904a6bf41cbb795e3193d896340",
            "vendor_name": "Zabbix",
            "vendor_version": "7.4-2",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "wizard_ready": "1",
            "readme": "## 概要\r\n\r\nこれは公式の Windows テンプレートです。Zabbix agent 7.4 以降が必要です。"
        }
    ],
    "id": 1
}
```


テンプレートグループの取得

テンプレート”Linux by Zabbix エージェント” が所属するテンプレートグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Response:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```


テンプレートによるホストの取得

”10001”(*Linux by Zabbix* エージェント) テンプレートがリンクされているホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートタグによる検索

タグ”host-name” が”{HOST.NAME}” に等しいテンプレートを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "hostid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- テンプレートグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース

ソース

CTemplate::get() in *ui/include/classes/api/services/CTemplate.php*.

template.massadd

説明

`object template.massadd(object parameters)`

このメソッドを使用すると、指定したテンプレートに複数の関連オブジェクトを同時に追加できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ使用できます。
このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object) 更新するテンプレートの ID と、テンプレートに追加するオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
templates	object/array	更新するテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	指定したテンプレートに追加するテンプレートグループ。 テンプレートグループでは、groupid プロパティのみを定義する必要があります。
macros	object/array	指定したテンプレート用に作成するユーザーマクロ。
templates_link	object/array	指定したテンプレートにリンクするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

グループをテンプレートにリンクする

テンプレートグループ「2」を 2 つのテンプレートに追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

2 つのテンプレートを 1 つのテンプレートにリンクする

テンプレート”10106” と”10104” をテンプレート”10073” にリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

関連項目

- template.update
- ホスト
- テンプレートグループ
- ユーザーマクロ

ソース

CTemplate::massAdd() in *ui/include/classes/api/services/CTemplate.php*.

template.massremove

説明

`object template.massremove(object parameters)`

このメソッドでは、複数のテンプレートから関連オブジェクトを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(object) 更新するテンプレートの ID と、削除するオブジェクトを含むパラメータ。

パラメータ	Type	説明
templateids	ID/array	更新するテンプレートの ID。 パラメータの動作: - 必須
groupids	ID/array	指定したテンプレートを削除するテンプレートグループの ID。
macros	string/array	指定したテンプレートから削除するユーザーマクロの ID。
templateids_clear	ID/array	指定したテンプレートからリンク解除してクリアするテンプレートの ID（上流）。
templateids_link	ID/array	指定したテンプレートからリンク解除するテンプレートの ID（上流）。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

グループからテンプレートを削除

グループ「2」から 2 つのテンプレートを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
```

```
        ],
        "groupids": "2"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

ホストからテンプレートのリンクを解除

テンプレート"10085" からテンプレート"10106" と"10104" のリンクを解除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

関連項目

- template.update
- ユーザーマクロ

ソース

CTemplate::massRemove() in *ui/include/classes/api/services/CTemplate.php*.

template.massupdate

説明

`object template.massupdate(object parameters)`

このメソッドでは、関連オブジェクトの一括置換または削除、および複数のテンプレートのプロパティ更新を同時に実行できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 更新するテンプレートの ID と、テンプレートに対して置き換えるオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
templates	object/array	更新するテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	テンプレートが所属する現在のテンプレートグループを置き換えるテンプレートグループ。 テンプレートグループでは、groupid プロパティのみを定義する必要があります。
macros	object/array	指定したテンプレート上の現在のすべてのユーザーマクロを置き換えるユーザーマクロ。
templates_clear	object/array	指定したテンプレートからリンク解除してクリアするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。
templates_link	object/array	現在リンクされているテンプレートを置き換えるテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。

戻り値

(object) templateids プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

テンプレートのリンク解除

指定されたテンプレートからテンプレート"10091" のリンクを解除してクリアします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massupdate",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "templates_clear": [
            {
                "templateid": "10091"
            }
        ]
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

ユーザーマクロの置換

複数のテンプレート上のすべてのユーザーマクロを、指定したユーザーマクロに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.massupdate",
    "params": {
        "templates": [
            {
                "templateid": "10074"
            },
            {
                "templateid": "10075"
            },
            {
                "templateid": "10076"
            },
            {
                "templateid": "10077"
            }
        ],
        "macros": [
            {
                "macro": "{$AGENT.TIMEOUT}",
                "value": "5m",
                "description": " エージェントが利用不可と見なされるまでのタイムアウト。Zabbix サーバー/プロキシから到達可
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10074",
            "10075",
            "10076",
            "10077"
        ]
    },
    "id": 1
}
```

関連項目

- template.update
- template.massadd
- テンプレートグループ
- ユーザーマクロ

ソース

CTemplate::massUpdate() in *ui/include/classes/api/services/CTemplate.php*.

template.update

説明

`object template.update(object/array templates)`

このメソッドでは、既存のテンプレートを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object/array`) 更新するテンプレートのプロパティ。

各テンプレートでは `templateid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、その他は変更されません。

標準のテンプレートプロパティ に加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	テンプレートが所属する現在のテンプレートグループを置き換えるためのテンプレートグループ。 テンプレートグループでは `groupid` プロパティのみを定義する必要があります。
tags	object/array	現在のテンプレートタグを置き換えるためのテンプレートタグ。
macros	object/array	指定したテンプレート上の現在のユーザーマクロを置き換えるためのユーザーマクロ。
templates	object/array	現在リンクされているテンプレートを置き換えるためのテンプレート。 渡されなかったテンプレートはリンク解除のみ行われます。 テンプレートでは `templateid` プロパティのみを定義する必要があります。
templates_clear	object/array	指定したテンプレートからリンク解除およびクリアを行うためのテンプレート。 テンプレートでは `templateid` プロパティのみを定義する必要があります。

戻り値

(`object`) `templateids` プロパティの下で更新されたテンプレートの ID を含むオブジェクトを返します。

例

標準テンプレートのプロパティの変更

テンプレートの技術名を「Linux by Zabbix agent Custom」に、表示名を「My template」に変更し、テンプレートの説明を更新します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "host": "Linux by Zabbix agent Custom",
        "name": "My template",
        "description": "This is a custom Linux template."
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートグループの更新

指定したテンプレートのすべてのテンプレートグループを、別のグループに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートタグの更新

すべてのテンプレートタグを別のタグに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
```

```
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートマクロの更新

すべてのテンプレートマクロを別のものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートにリンクされたテンプレートの更新

指定したテンプレートからすべてのテンプレートのリンクを解除し（クリアはしない）、別のテンプレートをそれにリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10087"
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

テンプレートにリンクされたテンプレートのクリア

指定したテンプレートについて、特定のリンクされたテンプレートのリンクを解除してクリアします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates_clear": [
            {
                "templateid": "10087"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

ソース

CTemplate::update() in *ui/include/classes/api/services/CTemplate.php*.

テンプレートグループ

このクラスは、テンプレートグループを操作するために設計されています。

オブジェクト参照:

- テンプレートグループ

利用可能なメソッド:

- templategroup.create - 新しいテンプレートグループを作成
- templategroup.delete - テンプレートグループを削除
- templategroup.get - テンプレートグループを取得
- templategroup.massadd - 関連オブジェクトをテンプレートグループに追加

- templategroup.massremove - 関連オブジェクトをテンプレートグループから削除
- templategroup.massupdate - テンプレートグループの関連オブジェクトを置換または削除
- templategroup.propagate - 権限をテンプレートグループのサブグループに伝播
- templategroup.update - テンプレートグループを更新

Template group オブジェクト

次のオブジェクトは、templategroupAPI に直接関連しています。

テンプレートグループ

テンプレートグループオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
groupid	ID	テンプレートグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
name	string	テンプレートグループの名前。 プロパティの動作: - 作成操作で 必須
uuid	string	既存のテンプレートグループにインポートしたテンプレートグループを関連付けるために使用される、汎用一意識別子。指定されていない場合は自動生成されます。

templategroup.create

説明

`object templategroup.create(object/array templateGroups)`

このメソッドは、新しいテンプレートグループを作成します。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(object/array) 作成するテンプレートグループ。このメソッドは、標準のテンプレートグループプロパティを持つテンプレートグループを受け付けます。

戻り値

(object) は、groupids プロパティに指定された、作成されたテンプレートグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートグループの順序と一致します。

例

テンプレートグループの作成

"Templates/Databases" という名前のテンプレートグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::create() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

説明

`object templategroup.delete(array templateGroupIds)`

このメソッドでは、テンプレートグループを削除できます。

テンプレートグループのみに属するテンプレートが含まれている場合、そのテンプレートグループは削除できません。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメーター

(array) 削除するテンプレートグループの ID

戻り値

(object) groupids プロパティの下に削除されたテンプレートグループの ID を含むオブジェクトを返します。

例

複数のテンプレートグループの削除

二つのテンプレートグループを削除する。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::delete() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

説明

`integer/array templategroup.get(object parameters)`

このメソッドは、指定されたパラメーターに従ってテンプレートグループを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
graphids	ID/array	指定したグラフを含むテンプレートを持つテンプレートグループのみを返します。
groupids	ID/array	指定したテンプレートグループ ID を持つテンプレートグループのみを返します。
templateids	ID/array	指定したテンプレートを含むテンプレートグループのみを返します。
triggerids	ID/array	指定したトリガーを含むテンプレートを持つテンプレートグループのみを返します。
with_graphs	boolean	グラフを持つテンプレートを含むテンプレートグループのみを返します。
with_graph_prototypes	boolean	グラフプロトタイプを持つテンプレートを含むテンプレートグループのみを返します。
with_httptests	boolean	Web 監視を持つテンプレートを含むテンプレートグループのみを返します。
with_items	boolean	アイテムを持つテンプレートを含むテンプレートグループのみを返します。 `with_simple_graph_items` パラメータを上書きします。
with_item_prototypes	boolean	アイテムプロトタイプを持つテンプレートを含むテンプレートグループのみを返します。 `with_simple_graph_item_prototypes` パラメータを上書きします。
with_simple_graph_item_prototypes	boolean	作成が有効で、かつ情報の型が数値であるアイテムプロトタイプを持つテンプレートを含むテンプレートグループのみを返します。
with_simple_graph_items	boolean	数値アイテムを持つテンプレートを含むテンプレートグループのみを返します。
with_templates	boolean	テンプレートを含むテンプレートグループのみを返します。
with_triggers	boolean	トリガーを持つテンプレートを含むテンプレートグループのみを返します。
selectTemplates	query	テンプレートグループに属するテンプレートを含む`templates` プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectTemplates` - 結果は `template` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `groupid`, `name`。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

Parameter	Type	Description
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクト数

例

名前によるデータの取得

"Templates/Databases" と"Templates/Modules" という名前の二つのテンプレートグループについて、全てのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

参照

- Template

ソース

CTemplateGroup::get() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

説明

`object templategroup.massadd(object parameters)`

このメソッドは、指定されたテンプレートグループすべてに複数の関連オブジェクトを同時に追加することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 更新するテンプレートグループの ID と、すべてのテンプレートグループに追加するオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
groups	object/array	更新するテンプレートグループ。 テンプレートグループでは、`groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
templates	object/array	すべてのテンプレートグループに追加するテンプレート。 テンプレートでは、`templateid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須

戻り値

(`object`) `groupids` プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループへテンプレートの追加

ID12 と 13 のテンプレートグループへ二つのテンプレートを追加します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

参照

- Template

ソース

CTemplateGroup::massAdd() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

説明

`object templategroup.massremove(object parameters)`

このメソッドは、関連するオブジェクトを複数のテンプレートグループから削除することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

更新するテンプレートグループの ID と、削除するオブジェクトを含む（`object`）パラメータ。

パラメータ	Type	説明
groupids	ID/array	更新するテンプレートグループの ID。 Parameter behavior: *- required*
templateids	ID/array	すべてのテンプレートグループから削除するテンプレートの ID。 Parameter behavior: *- required*

戻り値

（`object`）`groupids` プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループからテンプレートの削除

指定されたテンプレートグループから 2 つのテンプレートを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
```

```
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::massRemove() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massupdate

説明

`object templategroup.massupdate(object parameters)`

このメソッドは、複数のテンプレートグループの指定されたテンプレートを置き換えることができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 更新するテンプレートグループと更新対象のオブジェクトの ID を含むパラメータ

パラメータ	タイプ	説明
groups	object/array	更新するテンプレートグループ テンプレートグループには `groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
templates	object/array	テンプレートを使用して、指定されたテンプレートグループの現在のテンプレートを置き換えます。 指定されたテンプレート以外のすべてのテンプレートは、テンプレートグループから除外されます。 テンプレートには、`templateid` プロパティのみが定義されている必要があります。 パラメーターの動作: - 必須

戻り値

(object) groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループ内のテンプレートを置き換える

テンプレートグループ内のすべてのテンプレートを、指定のテンプレートに置き換えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massupdate",
    "params": {
        "groups": [
            {
                "groupid": "8"
            }
        ],
        "templates": [
            {
                "templateid": "40050"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
        ]
    },
    "id": 1
}
```

参照

- templategroup.update
- templategroup.massadd
- Template

ソース

CTemplateGroup::massUpdate() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

説明

`object templategroup.propagate(object parameters)`

このメソッドは、すべてのテンプレートグループのサブグループにアクセス許可を適用することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	Type	説明
groups	object/array	伝播するテンプレートグループ。 テンプレートグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
permissions	boolean	権限を伝播する必要がある場合は、true に設定します。 パラメータの動作: - 必須

戻り値

(object) groupids プロパティの下に伝搬されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループの権限をそのサブグループへ伝搬する

テンプレートグループの権限をそのサブグループへ伝搬します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

参照

- templategroup.update
- templategroup.massadd
- Template

ソース

CTemplateGroup::propagate() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.update

説明

`object templategroup.update(object/array templates)`

このメソッドは、既存のテンプレートグループを更新することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

更新する（object/array）のテンプレートグループのプロパティ。

各テンプレートグループでは、groupid プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、それ以外のプロパティは変更されません。

戻り値

（object）groupids プロパティの下に更新されたテンプレートグループの ID を含むオブジェクトを返します。

例

テンプレートグループの名前の変更

テンプレートグループの名前を”Templates/Databases” へ変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

ソース

CTemplateGroup::update() in *ui/include/classes/api/services/CTemplateGroup.php*.

テンプレートダッシュボード

このクラスは、テンプレートダッシュボードで動作するように設計されています。

オブジェクト参照:

- テンプレートダッシュボード
- テンプレートダッシュボードページ
    - テンプレートダッシュボードウィジェット
        * テンプレートダッシュボードウィジェットフィールド

利用可能なメソッド:

- templatedashboard.create - 新しいテンプレートダッシュボードを作成
- templatedashboard.delete - テンプレートダッシュボードを削除
- templatedashboard.get - テンプレートダッシュボードを取得
- templatedashboard.update - テンプレートダッシュボードを更新

Template dashboard オブジェクト

次のオブジェクトは、templatedashboardAPI に直接関連しています。

テンプレートダッシュボード

テンプレートダッシュボードオブジェクトには、以下のプロパティがあります。

Property	Type	Description
dashboardid	ID	テンプレートダッシュボードの ID。 Property behavior: - *read-only* - 更新操作で *required*
name	string	テンプレートダッシュボードの名前。 Property behavior: - 作成操作で *required*
templateid	ID	このダッシュボードが属するテンプレートの ID。 Property behavior: - *constant* - 作成操作で *required*
display_period	integer	デフォルトのページ表示期間（秒単位）。 指定可能な値: 10、30、60、120、600、1800、3600。 デフォルト: 30。
auto_start	integer	スライドショーを自動開始します。 指定可能な値: 0 - スライドショーを自動開始しない; 1 - （デフォルト）スライドショーを自動開始する。
uuid	string	インポートされたテンプレートダッシュボードを既存のものに関連付けるために使用される、汎用一意識別子です。指定されていない場合は自動生成されます。

テンプレートダッシュボードページ

テンプレートダッシュボードページオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
dashboard_pageid	ID	ダッシュボードページの ID。 プロパティの動作: - 読み取り専用
name	string	ダッシュボードページ名。 デフォルト: 空文字列。
display_period	integer	ダッシュボードページの表示期間（秒単位）。 指定可能な値: 0、10、30、60、120、600、1800、3600。 デフォルト: 0（デフォルトのページ表示期間が使用されます）。
widgets	array	テンプレートダッシュボードウィジェットオブジェクトの配列。

テンプレートダッシュボードウィジェット

テンプレートダッシュボードウィジェットオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
widgetid	ID	ダッシュボードウィジェットの ID。 プロパティの動作: - 読み取り専用
type	string	ダッシュボードウィジェットのタイプ。 使用可能な値: actionlog - アクションログ; clock - 時計; discovery - ディスカバリのステータス; favgraphs - お気に入りのグラフ; favmaps - お気に入りのマップ; gauge - ゲージ; graph - グラフ（クラシック）; graphprototype - グラフプロトタイプ; honeycomb - ハニカム; hostavail - ホストの可用性; hostcard - ホストカード; hostnavigator - ホストナビゲーター; itemcard - アイテムカード; itemnavigator - アイテムナビゲーター; item - アイテムの値; map - マップ; navtree - マップナビゲーションツリー; piechart - 円グラフ; plaintext - プレーンテキスト; problemhosts - 障害ホスト; problems - 障害; problemsbysv - 深刻度別の障害; slareport - SLA レポート; svggraph - グラフ; systeminfo - システム情報; tophosts - 上位ホスト; topitems - 上位アイテム; toptriggers - 上位トリガー; trigover - トリガーの概要; url - URL; web - Web 監視。 プロパティの動作: - 必須
name	string	カスタムウィジェット名。
x	integer	ダッシュボードの左端からの水平方向の位置。 指定可能な値の範囲は 0 から 71 です。
y	integer	ダッシュボードの上端からの垂直方向の位置。 指定可能な値の範囲は 0 から 63 です。
width	integer	ウィジェットの幅。 指定可能な値の範囲は 1 から 72 です。
height	integer	ウィジェットの高さ。 指定可能な値の範囲は 1 から 64 です。
view_mode	integer	ウィジェットの表示モード。 使用可能な値: 0 - *(デフォルト)* デフォルトのウィジェット表示; 1 - ヘッダーを非表示;
fields	array	テンプレートダッシュボードウィジェットフィールド オブジェクトの配列。

テンプレートダッシュボードウィジェットフィールド

テンプレートダッシュボードウィジェットフィールドオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
type	integer	ウィジェットフィールドのタイプ。 使用可能な値: 0 - 整数; 1 - 文字列; 4 - アイテム; 5 - アイテムプロトタイプ; 6 - グラフ; 7 - グラフプロトタイプ; 8 - マップ; 9 - サービス; 10 - SLA; 11 - ユーザー; 12 - アクション; 13 - メディアタイプ。 Property behavior: - *required*
name	string	ウィジェットフィールド名。 使用可能な値: Dashboard widget fields を参照してください。なお、一部のホスト関連パラメータ（例: *Problems* ウィジェットの *Host groups*、*Exclude host groups*、*Hosts*、*Host availability* ウィジェットの *Host groups* など）は、テンプレートダッシュボード上でウィジェットを設定する際には使用できません。これは、テンプレートダッシュボードが、テンプレートがリンクされているホストのデータのみを表示するためです。 Property behavior: - *required*
value	mixed	タイプに応じたウィジェットフィールドの値。 使用可能な値: Dashboard widget fields を参照してください。なお、一部のホスト関連パラメータ（例: *Problems* ウィジェットの *Host groups*、*Exclude host groups*、*Hosts*、*Host availability* ウィジェットの *Host groups* など）は、テンプレートダッシュボード上でウィジェットを設定する際には使用できません。これは、テンプレートダッシュボードが、テンプレートがリンクされているホストのデータのみを表示するためです。 Property behavior: - *required*

templatedashboard.create

説明

`object templatedashboard.create(object/array templateDashboards)`

このメソッドでは、新しいテンプレートダッシュボードを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object/array) 作成するテンプレートダッシュボード。

このメソッドは、標準のテンプレートダッシュボードプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
pages	array	ダッシュボード用に作成するテンプレートダッシュボードページ。ダッシュボードページは、指定された順序と同じ順序で並べられます。 パラメータの動作: - 必須

戻り値

(object) dashboardids プロパティの下に作成されたテンプレートダッシュボードの ID を含むオブジェクトを返します。返される ID の順序は、渡されたテンプレートダッシュボードの順序と一致します。

例

テンプレートダッシュボードの作成

単一のダッシュボードページに 1 つのグラフウィジェットを配置した、“Graphs” という名前のテンプレートダッシュボードを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.create",
    "params": {
        "templateid": "10318",
        "name": "Graphs",
        "pages": [
            {
                "widgets": [
                    {
                        "type": "graph",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 5,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 6,
                                "name": "graphid",
                                "value": "1123"
                            }
                        ]
                    }
                ]

            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

関連項目

- テンプレートダッシュボードページ
- テンプレートダッシュボードウィジェット
- テンプレートダッシュボードウィジェットフィールド

ソース

CTemplateDashboard::create() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

説明

`object templatedashboard.delete(array templateDashboardIds)`

このメソッドは、テンプレートダッシュボードを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するテンプレートダッシュボードの ID。

戻り値

(object) dashboardids プロパティの下にある削除されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

複数のテンプレートダッシュボードの削除

2 つのテンプレートダッシュボードを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

ソース

CTemplateDashboard::delete() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

説明

`integer/array templatedashboard.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってテンプレートダッシュボードを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。
メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dashboardids	ID/array	指定した ID を持つテンプレートダッシュボードのみを返します。
templateids	ID/array	指定したテンプレートに属するテンプレートダッシュボードのみを返します。
selectPages	query	テンプレートダッシュボードのページを正しい順序で含むpages プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: dashboardid, name。
countOutput	boolean	これらのパラメータについては、リファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

テンプレートダッシュボードの取得

指定したテンプレートについて、ウィジェットを含むすべてのテンプレートダッシュボードを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
        "dashboardid": "23",
        "name": "Docker overview",
        "templateid": "10001",
        "display_period": "30",
        "auto_start": "1",
        "uuid": "6dfcbe0bc5ad400ea9c1c2dd7649282f",
        "pages": [
            {
                "dashboard_pageid": "1",
                "name": "",
                "display_period": "0",
                "widgets": [
                    {
                        "widgetid": "220",
                        "type": "graph",
                        "name": "",
                        "x": "0",
                        "y": "0",
                        "width": "36",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "6",
                                "name": "graphid",
                                "value": "1125"
                            }
                        ]
                    },
                    {
                        "widgetid": "221",
                        "type": "graph",
                        "name": "",
                        "x": "12",
                        "y": "0",
                        "width": "36",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "6",
                                "name": "graphid",
                                "value": "1129"
                            }
                        ]
                    },
                    {
                        "widgetid": "222",
                        "type": "graph",
                        "name": "",
                        "x": "0",
                        "y": "5",
                        "width": "36",
                        "height": "5",
                        "view_mode": "0",
                        "fields": [
                            {
                                "type": "6",
                                "name": "graphid",
                                "value": "1128"
                            }
                        ]
```

```
                        },
                        {
                            "widgetid": "223",
                            "type": "graph",
                            "name": "",
                            "x": "12",
                            "y": "5",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1126"
                                }
                            ]
                        },
                        {
                            "widgetid": "224",
                            "type": "graph",
                            "name": "",
                            "x": "0",
                            "y": "10",
                            "width": "36",
                            "height": "5",
                            "view_mode": "0",
                            "fields": [
                                {
                                    "type": "6",
                                    "name": "graphid",
                                    "value": "1127"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- テンプレートダッシュボードページ
- テンプレートダッシュボードウィジェット
- テンプレートダッシュボードウィジェットフィールド

ソース

CTemplateDashboard::get() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

説明

`object templatedashboard.update(object/array templateDashboards)`

このメソッドでは、既存のテンプレートダッシュボードを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object/array) 更新するテンプレートダッシュボードのプロパティ。

各ダッシュボードには dashboardid プロパティを指定する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新されます。

また、このメソッドは標準のテンプレートダッシュボードのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
pages	array	既存のダッシュボードページを置き換えるテンプレートダッシュボードページ。 ダッシュボードページは dashboard_pageid プロパティによって更新されます。dashboard_pageid プロパティを持たないオブジェクトについては、新しいダッシュボードページが作成され、再利用されなかった既存のダッシュボードページは削除されます。ダッシュボードページは、指定された順序と同じ順序で並べられます。ダッシュボードページの指定されたプロパティのみが更新されます。pages プロパティには少なくとも 1 つのダッシュボードページオブジェクトが必要です。

戻り値

(object) dashboardids プロパティの下で更新されたテンプレートダッシュボードの ID を含むオブジェクトを返します。

例

テンプレートダッシュボードの名前変更

テンプレートダッシュボードの名前を「Performance graphs」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

テンプレートダッシュボードのページの更新

最初のダッシュボードページの名前を変更し、2 番目のダッシュボードページのウィジェットを置き換え、3 番目のページとして新しいページを追加します。その他のダッシュボードページはすべて削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
```

```
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```


関連項目

- テンプレートダッシュボードウィジェット
- テンプレートダッシュボードウィジェットフィールド

ソース

CTemplateDashboard::update() in *ui/include/classes/api/services/CTemplateDashboard.php*.

ディスカバリチェック

このクラスはディスカバリチェックで動作するように設計されています。

オブジェクト参照:

- ディスカバリチェック

利用可能なメソッド:

- dcheck.get - ディスカバリチェックを取得します

Discovery check オブジェクト

次のオブジェクトは、dcheckAPI に直接関連するものです。

ディスカバリチェック

ディスカバリチェックオブジェクトは、ネットワークディスカバリルールによって実行される特定のチェックを定義します。以下のプロパティがあります。

プロパティ	型	説明
dcheckid	ID	ディスカバリチェックの ID。
druleid	ID	このチェックが属するディスカバリルールの ID。
key_	string	アイテムキー（type が”Zabbix agent” に設定されている場合）または SNMP OID（type が”SNMPv1 agent”、”SNMPv2 agent”、または”SNMPv3 agent” に設定されている場合）。 プロパティの動作: - type が”Zabbix agent”、”SNMPv1 agent”、”SNMPv2 agent”、または”SNMPv3 agent” に設定されている場合は 必須
ports	string	チェックする 1 つまたは複数のポート範囲。カンマで区切ります。 デフォルト: 0。
snmp_community	string	プロパティの動作: - type が”SSH” (0)、”LDAP” (1)、”SMTP” (2)、”FTP” (3)、”HTTP” (4)、”POP” (5)、”NNTP” (6)、”IMAP” (7)、”TCP” (8)、”Zabbix agent” (9)、”SNMPv1 agent” (10)、”SNMPv2 agent” (11)、”SNMPv3 agent” (13)、”HTTPS” (14)、または”Telnet” (15) に設定されている場合に サポート SNMP コミュニティ。
snmpv3_authpassphrase	string	プロパティの動作: - type が”SNMPv1 agent” または”SNMPv2 agent” に設定されている場合は 必須 認証パスフレーズ。
snmpv3_authprotocol	integer	プロパティの動作: - type が”SNMPv3 agent” に設定され、snmpv3_securitylevel が”authNoPriv” または”authPriv” に設定されている場合に サポート 認証プロトコル。 使用可能な値: 0 - *(デフォルト)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512。
snmpv3_contextname	string	プロパティの動作: - type が”SNMPv3 agent” に設定され、snmpv3_securitylevel が”authNoPriv” または”authPriv” に設定されている場合に サポート SNMPv3 コンテキスト名。
snmpv3_privpassphrase	string	プロパティの動作: - type が”SNMPv3 agent” に設定されている場合に サポート プライバシーパスフレーズ。 プロパティの動作: - type が”SNMPv3 agent” に設定され、snmpv3_securitylevel が”authPriv” に設定されている場合に サポート

プロパティ	型	説明
snmpv3_privprotocol	integer	プライバシープロトコル。 使用可能な値: 0 - *(デフォルト)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C。 プロパティの動作: - type が”SNMPv3 agent” に設定され、snmpv3_securitylevel が”authPriv” に設定されている場合に サポート
snmpv3_securitylevel	string	セキュリティレベル。 使用可能な値: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv。 プロパティの動作: - type が”SNMPv3 agent” に設定されている場合に サポート
snmpv3_securityname	string	セキュリティ名。 プロパティの動作: - type が”SNMPv3 agent” に設定されている場合に サポート
type	integer	チェックのタイプ。 使用可能な値: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix agent; 10 - SNMPv1 agent; 11 - SNMPv2 agent; 12 - ICMP ping; 13 - SNMPv3 agent; 14 - HTTPS; 15 - Telnet。 プロパティの動作: - *必須*
uniq	integer	このチェックをデバイス一意性の判定基準として使用するかどうか。ディスカバリルールごとに設定できる一意チェックは 1 つだけです。 使用可能な値: 0 - *(デフォルト)* このチェックを一意性の判定基準として使用しない; 1 - このチェックを一意性の判定基準として使用する。 プロパティの動作: - type が”Zabbix agent”、”SNMPv1 agent”、”SNMPv2 agent”、または”SNMPv3 agent” に設定されている場合に サポート

プロパティ	型	説明
host_source	integer	ホスト名のソース。 使用可能な値: 1 - *(デフォルト)* DNS; 2 - IP; 3 - このチェックのディスカバリ値。
name_source	integer	表示名のソース。 使用可能な値: 0 - *(デフォルト)* 指定なし; 1 - DNS; 2 - IP; 3 - このチェックのディスカバリ値。
allow_redirect	integer	ICMP ping の対象が別の IP アドレスから応答する状況を許可します。 使用可能な値: 0 - *(デフォルト)* リダイレクトされた応答を、対象ホストがダウンしている場合として扱う（失敗）; 1 - リダイレクトされた応答を、対象ホストが稼働している場合として扱う（成功）。 プロパティの動作: - `type` が”ICMP ping” に設定されている場合に サポート

dcheck.get

説明

`integer/array dcheck.get(object parameters)`

このメソッドは、指定したパラメーターに応じたディスカバリチェックを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

`(object)` 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dcheckids	ID/array	指定した ID のディスカバリチェックのみを返します。
druleids	ID/array	指定したディスカバリルールに属するディスカバリチェックのみを返します。
dserviceids	ID/array	指定したディスカバリ済みサービスを検出したディスカバリチェックのみを返します。
selectDRules	query	ディスカバリチェックに関連するディスカバリルールを返します。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `dcheckid`, `druleid`。
countOutput	boolean	これらのパラメータについては、リファレンスコメントを参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

パラメータ	型	説明
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ディスカバリルールのディスカバリチェックを取得

ディスカバリルール"6" によって使用されるすべてのディスカバリチェックを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dcheckid": "6",
            "druleid": "4",
            "type": "3",
            "key_": "",
            "snmp_community": "",
            "ports": "21",
            "snmpv3_securityname": "",
            "snmpv3_securitylevel": "0",
            "snmpv3_authpassphrase": "",
            "snmpv3_privpassphrase": "",
            "uniq": "0",
            "snmpv3_authprotocol": "0",
            "snmpv3_privprotocol": "0",
            "snmpv3_contextname": "",
            "host_source": "1",
            "name_source": "0",
            "allow_redirect": "0"
        }
    ],
    "id": 1
}
```

ソース

CDCheck::get() in *ui/include/classes/api/services/CDCheck.php*.

ディスカバリルール

このクラスは、ネットワークディスカバリルールを操作するために設計されています。

Note:
この API は、ネットワークディスカバリルールを操作するためのものです。ローレベルディスカバリルールについては、LLD ルール APIを参照してください。

オブジェクト参照:

- ディスカバリルール

利用可能なメソッド:

- drule.create - 新しいネットワークディスカバリルールを作成
- drule.delete - ネットワークディスカバリルールを削除
- drule.get - ネットワークディスカバリルールを取得
- drule.update - ネットワークディスカバリルールを更新

Discovery rule オブジェクト

次のオブジェクトは、druleAPI に直接関連しています。

ディスカバリルール

ディスカバリルールオブジェクトは、ネットワークディスカバリルールを定義します。以下のプロパティがあります。

プロパティ	型	説明
druleid	ID	ディスカバリルールの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
iprange	string	確認する 1 つまたは複数の IP 範囲。カンマで区切ります。 IP 範囲のサポートされる形式の詳細については、ネットワークディスカバリ設定 セクションを参照してください。 プロパティの動作: - 作成操作で 必須
name	string	ディスカバリルールの名前。 プロパティの動作: - 作成操作で 必須
delay	string	ディスカバリルールの実行間隔。 秒、またはサフィックス付きの時間単位（例: 30s、1m、2h、1d）、あるいはユーザーマクロを受け付けます。 デフォルト: 1h。
proxyid	ID	ディスカバリに使用されるプロキシの ID。
status	integer	ディスカバリルールが有効かどうか。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
concurrency_max	integer	ディスカバリルールごとの同時チェックの最大数。 指定可能な値: 0 - *(デフォルト)* チェック数は無制限; 1 - 1 チェック; 2-999 - カスタムのチェック数。
error	string	ディスカバリルールの実行時に問題が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用

drule.create

説明

`object drule.create(object/array discoveryRules)`

このメソッドは、新しいディスカバリルールを作成することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成するディスカバリルール。

このメソッドは、標準のディスカバリルールのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
dchecks	array	ディスカバリルール用に作成するディスカバリチェック。 パラメータの動作: - 必須

戻り値

(object) druleids プロパティの下に作成されたディスカバリルールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたディスカバリルールの順序と一致します。

例

ディスカバリルールの作成

ローカルネットワークで Zabbix エージェントを実行しているマシンを見つけるためのディスカバリルールを作成します。ルールはポート 10050 で単一の Zabbix エージェントチェックを使用する必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.create",
    "params": {
        "name": "Zabbix agent discovery",
        "iprange": "192.168.1.1-255",
        "concurrency_max": "10",
        "dchecks": [
            {
                "type": "9",
                "key_": "system.uname",
                "ports": "10050",
                "uniq": "0"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ディスカバリチェック

ソース

CDRule::create() in *ui/include/classes/api/services/CDRule.php*.

drule.delete

説明

`object drule.delete(array discoveryRuleIds)`

このメソッドは、ディスカバリルールを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するディスカバリルールの ID。

戻り値

(object) druleids プロパティの下にある削除されたディスカバリルールの ID を含むオブジェクトを返します。

例

複数のディスカバリルールの削除

2 つのディスカバリルールを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CDRule::delete() in *ui/include/classes/api/services/CDRule.php*.

drule.get

説明

`integer/array drule.get(object parameters)`

このメソッドは、指定したパラメーターに応じたディスカバリールールを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dhostids	ID/array	指定した検出済みホストを作成したディスカバリルールのみを返します。
druleids	ID/array	指定した ID を持つディスカバリルールのみを返します。
dserviceids	ID/array	指定した検出済みサービスを作成したディスカバリルールのみを返します。
selectDChecks	query	ディスカバリルールで使用されるディスカバリチェックを含むdchecksプロパティを返します。 count をサポートします。
selectDHosts	query	ディスカバリルールによって作成された検出済みホストを含むdhostsプロパティを返します。 count をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectDChecks - 結果は dcheckid でソートされます。 selectDHosts - 結果は dhostsid でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: druleid, name。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

すべてのディスカバリルールの取得

設定されたすべてのディスカバリルールと、それらが使用するディスカバリチェックを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
```

```
        "selectDChecks": "extend"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "druleid": "2",
            "proxyid": "0",
            "name": "Local network",
            "iprange": "192.168.3.1-255",
            "delay": "5s",
            "status": "0",
            "concurrency_max": "0",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "7",
                    "druleid": "2",
                    "type": "3",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "21",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "1",
                    "name_source": "0",
                    "allow_redirect": "0"
                },
                {
                    "dcheckid": "8",
                    "druleid": "2",
                    "type": "4",
                    "key_": "",
                    "snmp_community": "",
                    "ports": "80",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "1",
                    "name_source": "0",
                    "allow_redirect": "0"
                }
            ]
        },
        {
            "druleid": "6",
            "proxyid": "0",
```

```
            "name": "Zabbix agent discovery",
            "iprange": "192.168.1.1-255",
            "delay": "1h",
            "status": "0",
            "concurrency_max": "10",
            "error": "",
            "dchecks": [
                {
                    "dcheckid": "10",
                    "druleid": "6",
                    "type": "9",
                    "key_": "system.uname",
                    "snmp_community": "",
                    "ports": "10050",
                    "snmpv3_securityname": "",
                    "snmpv3_securitylevel": "0",
                    "snmpv3_authpassphrase": "",
                    "snmpv3_privpassphrase": "",
                    "uniq": "0",
                    "snmpv3_authprotocol": "0",
                    "snmpv3_privprotocol": "0",
                    "snmpv3_contextname": "",
                    "host_source": "2",
                    "name_source": "3",
                    "allow_redirect": "0"
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- 検出されたホスト
- ディスカバリチェック

ソース

CDRule::get() in *ui/include/classes/api/services/CDRule.php*.

drule.update

説明

`object drule.update(object/array discoveryRules)`

このメソッドは、既存のディスカバリールールを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するディスカバリルールのプロパティ。

各ディスカバリルールでは `druleid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のディスカバリルールのプロパティに加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
dchecks	array	既存のチェックを置き換えるディスカバリチェック。

戻り値

(object) druleids プロパティの下で更新されたディスカバリルール ID を含むオブジェクトを返します。

例

ディスカバリルールの IP アドレス範囲の変更

ディスカバリルールの IP アドレス範囲を”192.168.2.1-255” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ディスカバリチェック

ソース

CDRule::update() in *ui/include/classes/api/services/CDRule.php*.

トリガー

このクラスはトリガーを操作するために設計されています。

オブジェクト参照:

- トリガー
- トリガータグ

利用可能なメソッド:

- trigger.create - 新しいトリガーを作成
- trigger.delete - トリガーを削除
- trigger.get - トリガーを取得
- trigger.update - トリガーを更新

Trigger オブジェクト

次のオブジェクトは、triggerAPI に直接関連しています。

トリガー

トリガーオブジェクトには、以下のプロパティがあります。

Property	Type	Description
triggerid	ID	トリガーの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
description	string	トリガーの名前。 プロパティの動作: - 作成操作で 必須
expression	string	トリガー条件式。 `value_type` が 5（バイナリ）に設定されたアイテムは、トリガー条件式では使用できません。 プロパティの動作: - 作成操作で 必須
event_name	string	トリガーによって生成されるイベント名。
opdata	string	運用データ。
comments	string	トリガーの追加説明。
error	string	トリガーの状態更新時に問題が発生した場合のエラーテキスト。 プロパティの動作: - 読み取り専用
flags	integer	トリガーの生成元。 使用可能な値: 0 - *(デフォルト)* 通常のトリガー; 4 - プロトタイプから変換されたトリガー。 プロパティの動作: - 読み取り専用
lastchange	timestamp	トリガーの状態が最後に変更された時刻。 プロパティの動作: - 読み取り専用
priority	integer	トリガーの深刻度。 使用可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 軽度の障害; 4 - 重度の障害; 5 - 致命的な障害。
state	integer	トリガーの状態。 使用可能な値: 0 - *(デフォルト)* トリガーの状態は最新です; 1 - 現在のトリガーの状態は不明です。 プロパティの動作: - 読み取り専用
status	integer	トリガーが有効か無効か。 使用可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
templateid	ID	親テンプレートトリガーの ID。 プロパティの動作: - 読み取り専用

Property	Type	Description
type	integer	トリガーが複数の障害イベントを生成できるかどうか。 使用可能な値: 0 - *(デフォルト)* 複数イベントを生成しない; 1 - 複数イベントを生成する。
url	string	トリガーに関連付けられた URL。
url_name	string	トリガーに関連付けられた URL のラベル。
value	integer	トリガーが OK 状態か障害状態か。 使用可能な値: 0 - *(デフォルト)* OK; 1 - 障害。 プロパティの動作: - 読み取り専用
recovery_mode	integer	OK イベント生成モード。 使用可能な値: 0 - *(デフォルト)* 条件式; 1 - 復旧条件式; 2 - なし。
recovery_expression	string	トリガー復旧条件式。 `value_type` が 5（バイナリ）に設定されたアイテムは、トリガー復旧条件式では使用できません。
correlation_mode	integer	OK イベントでクローズする対象。 使用可能な値: 0 - *(デフォルト)* すべての障害; 1 - タグ値が一致する場合はすべての障害。
correlation_tag	string	照合用のタグ。
manual_close	integer	手動クローズを許可。 使用可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。
uuid	string	インポートしたトリガーを既存のトリガーにリンクするために使用される汎用一意識別子。指定しない場合は自動生成されます。 プロパティの動作: - トリガーがテンプレートに属している場合に サポート

トリガータグ

トリガータグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	トリガータグ名。 プロパティの動作: - 必須
value	string	トリガータグの値。

trigger.create

説明

`object trigger.create(object/array triggers)`

このメソッドでは、新しいトリガーを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object/array) 作成するトリガー。

このメソッドは、標準のトリガープロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
dependencies	array	トリガーが依存するトリガー。 これらのトリガーでは、triggerid プロパティのみを定義する必要があります。
tags	array	トリガータグ。

戻り値

(object) triggerids プロパティの下で作成されたトリガーの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガーの順序と一致します。

例

トリガーの作成

2 つのトリガーを作成し、それぞれを別の 1 つのトリガーに依存させます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "{HOST.NAME} のプロセッサー負荷が高すぎます",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": " サービスが停止しました",
            "expression": "length(last(/Linux server/log[/var/log/system,Service .* has stopped]))<>0",
            "dependencies": [
                {
                    "triggerid": "17368"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::create() in *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

説明

`object trigger.delete(array triggerIds)`

このメソッドは、トリガーを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するトリガーの ID。

戻り値

(object) triggerids プロパティの下にある削除されたトリガーの ID を含むオブジェクトを返します。

例

複数のトリガーの削除

2 つのトリガーを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
```

```
    "id": 1
}
```

ソース

CTrigger::delete() in *ui/include/classes/api/services/CTrigger.php*.

trigger.get

説明

`integer/array trigger.get(object parameters)`

このメソッドは、指定したパラメーターに応じたトリガーを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

Parameter	Type	Description
triggerids	ID/array	指定した ID のトリガーのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートのトリガーのみを返します。
templateids	ID/array	指定したテンプレートに属するトリガーのみを返します。
hostids	ID/array	指定したホストに属するトリガーのみを返します。
itemids	ID/array	指定したアイテムを含むトリガーのみを返します。
functions	string/array	指定した関数を使用するトリガーのみを返します。 サポートされている関数の一覧については、supported function ページを参照してください。
group	string	指定した名前のホストグループまたはテンプレートグループに属するホストまたはテンプレートのトリガーのみを返します。
host	string	指定した技術名のホストに属するトリガーのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承されたトリガーのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属するトリガーのみを返します。
dependent	boolean	`true` に設定した場合、依存関係を持つトリガーのみを返します。`false` に設定した場合、依存関係を持たないトリガーのみを返します。
monitored	flag	監視中のホストに属し、有効なアイテムのみを含む有効なトリガーのみを返します。
active	flag	監視中のホストに属する有効なトリガーのみを返します。
maintenance	boolean	`true` に設定した場合、メンテナンス中のホストに属する有効なトリガーのみを返します。
withUnacknowledgedEvents	flag	未確認のイベントを持つトリガーのみを返します。
withAcknowledgedEvents	flag	すべてのイベントが確認済みのトリガーのみを返します。
withLastEventUnacknowledged	flag	最後のイベントが未確認のトリガーのみを返します。
skipDependent	flag	他のトリガーに依存している障害状態のトリガーをスキップします。なお、他のトリガーが無効である場合、無効なアイテムを持つ場合、または無効なアイテムのホストに属する場合、それらのトリガーは無視されます。
lastChangeSince	timestamp	指定した時刻以降に状態が変化したトリガーのみを返します。
lastChangeTill	timestamp	指定した時刻以前に状態が変化したトリガーのみを返します。
only_true	flag	最近障害状態になったトリガーのみを返します。
min_severity	integer	指定した深刻度以上のトリガーのみを返します。

Parameter	Type	Description
evaltype	integer	タグのevaluation method。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or.
tags	array	指定したタグを持つトリガーのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのトリガーを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist.
expandComment	flag	トリガーの説明内のマクロを展開します。
expandDescription	flag	トリガー名内のマクロを展開します。
expandExpression	flag	トリガー条件式内の関数およびマクロを展開します。
selectHostGroups	query	トリガーが属するホストグループを`hostgroups` プロパティで返します。
selectHosts	query	トリガーが属するホストを`hosts` プロパティで返します。
selectItems	query	トリガーに含まれるアイテムを`items` プロパティで返します。
selectFunctions	query	トリガーで使用されている関数を `functions` プロパティで返します。 関数オブジェクトはトリガー条件式で使用される関数を表し、以下のプロパティを持ちます: `functionid` - (ID) 関数の ID; `itemid` - (ID) 関数で使用されるアイテムの ID; `function` - (string) 関数名; `parameter` - (string) 関数に渡されるパラメータ。返される文字列では、クエリパラメータは $ 記号に置き換えられます。
selectDependencies	query	トリガーが依存しているトリガーを `dependencies` プロパティで返します。
selectDiscoveryData	query	トリガーのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。トリガーのディスカバリオブジェクトは、ディスカバリされたトリガーを、そのトリガーの元となったトリガープロトタイプに関連付けます。 以下のプロパティを持ちます: `parent_triggerid` - (ID) トリガーの作成元となったトリガープロトタイプの ID; `status` - (int) トリガーのディスカバリステータス: 0 - *(デフォルト)* トリガーはディスカバリされています; 1 - トリガーはすでにディスカバリされていません; `ts_delete` - (timestamp) すでにディスカバリされていないトリガーが削除される時刻; `ts_disable` - (timestamp) すでにディスカバリされていないトリガーが無効化される時刻; `disable_source` - (int) トリガーが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(デフォルト)* 自動的に無効化; 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	トリガーを作成したローレベルディスカバリルールを `discoveryRule` プロパティで返します。
selectLastEvent	query	最後の重要なトリガーイベントを`lastEvent` プロパティで返します。
selectTags	query	トリガーのタグを`tags` プロパティで返します。
selectTemplateGroups	query	トリガーが属するテンプレートグループを`templategroups` プロパティで返します。

Parameter	Type	Description
filter	object	指定したフィルターに完全一致する結果のみを返します。 プロパティ名をキーとし、値を一致対象の単一の値または値の配列とするオブジェクトを受け付けます。 text data type のプロパティはサポートされません。 追加のプロパティをサポートします: host - トリガーが属するホストの技術名; hostid - トリガーが属するホストの ID。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectHosts - 結果は host でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: triggerid, description, status, priority, lastchange, hostname。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	トリガーのディスカバリオブジェクトを triggerDiscovery プロパティで返します。トリガーのディスカバリオブジェクトは、トリガーをその作成元となったトリガープロトタイプに関連付けます。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

トリガー ID によるデータの取得

トリガー"14062" で使用されるすべてのデータと関数を取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "14062",
            "expression": "{13513}<10m",
            "description": "{HOST.NAME} has been restarted (uptime < 10m)",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "The host uptime is less than 10 minutes",
            "error": "",
            "templateid": "10016",
            "type": "0",
            "state": "0",
            "flags": "0",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "event_name": "",
            "uuid": "",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13513",
                    "itemid": "24350",
                    "triggerid": "14062",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


障害状態のトリガーの取得

障害状態にあるすべてのトリガーの ID、名前、深刻度を取得し、深刻度の降順で並べ替えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix self-monitoring processes < 100% busy",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix discoverer processes more than 75% busy",
            "priority": "3"
        }
    ],
    "id": 1
}
```


タグ付きの特定のトリガーの取得

タグ付きの特定のトリガーを取得します。

リクエスト：


```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17370",
            "description": "Service status",
            "tags": [
                {
                    "tag": "service",
                    "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
                },
                {
                    "tag": "error",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- ディスカバリルール

- アイテム
- ホスト
- ホストグループ
- テンプレートグループ

ソース

CTrigger::get() in *ui/include/classes/api/services/CTrigger.php*.

trigger.update

説明

`object trigger.update(object/array triggers)`

このメソッドでは、既存のトリガーを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object/array) 更新するトリガーのプロパティ。

各トリガーでは `triggerid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のトリガープロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
dependencies	array	トリガーが依存しているトリガー。
tags	array	トリガーでは `triggerid` プロパティのみを定義する必要があります。 トリガータグ。

戻り値

(object) `triggerids` プロパティの下で更新されたトリガーの ID を含むオブジェクトを返します。

例

トリガーの有効化

トリガーを有効化します。つまり、ステータスを”0” に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
```

```
    "id": 1
}
```

トリガーのタグの置換

トリガーのタグを置換します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "tags": [
            {
                "tag": "service",
                "value": "{{ITEM.VALUE}.regsub(\"Service (.*) has stopped\", \"\\1\")}"
            },
            {
                "tag": "error",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

依存関係の置き換え

トリガーの依存関係を置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

ソース

CTrigger::update() in *ui/include/classes/api/services/CTrigger.php*.

トリガープロトタイプ

このクラスは、トリガープロトタイプを操作するために設計されています。

オブジェクト参照:

- トリガープロトタイプ
- トリガープロトタイプタグ

利用可能なメソッド:

- triggerprototype.create - 新しいトリガープロトタイプを作成
- triggerprototype.delete - トリガープロトタイプを削除
- triggerprototype.get - トリガープロトタイプを取得
- triggerprototype.update - トリガープロトタイプを更新

Trigger prototype オブジェクト

次のオブジェクトは、`triggerprototype`API に直接関連しています。

トリガープロトタイプ

トリガープロトタイプオブジェクトには、以下のプロパティがあります。

Property	Type	Description
triggerid	ID	トリガープロトタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
description	string	トリガープロトタイプの名前。 プロパティの動作: - 作成操作では 必須
expression	string	トリガー条件式。 少なくとも 1 つのアイテムプロトタイプを含める必要があります。`value_type` が 5（バイナリ）に設定されているアイテムプロトタイプは、トリガー条件式では使用できません。 プロパティの動作: - 作成操作では 必須
event_name	string	トリガーによって生成されるイベント名。
opdata	string	運用データ。
comments	string	トリガープロトタイプに対する追加コメント。
priority	integer	トリガープロトタイプの深刻度。 指定可能な値: 0 - *(デフォルト)* 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 障害。

Property	Type	Description
status	integer	トリガープロトタイプが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
flags	integer	トリガープロトタイプの生成元。 指定可能な値: 2 - トリガープロトタイプ; 6 - ディスカバリされたトリガープロトタイプ。 プロパティの動作: - 読み取り専用
templateid	ID	親テンプレートのトリガープロトタイプの ID。 プロパティの動作: - 読み取り専用
type	integer	トリガープロトタイプが複数の障害イベントを生成できるかどうか。 指定可能な値: 0 - *(デフォルト)* 複数イベントを生成しない; 1 - 複数イベントを生成する。
url	string	トリガープロトタイプに関連付けられた URL。
url_name	string	トリガープロトタイプに関連付けられた URL のラベル。
recovery_mode	integer	OK イベントの生成モード。 指定可能な値: 0 - *(デフォルト)* 条件式; 1 - 復旧条件式; 2 - なし。
recovery_expression	string	トリガー復旧条件式。 少なくとも 1 つのアイテムプロトタイプを含める必要があります。 `value_type` が 5（バイナリ）に設定されているアイテムプロトタイプは、トリガー復旧条件式では使用できません。
correlation_mode	integer	OK イベントによるクローズ方法。 指定可能な値: 0 - *(デフォルト)* すべての障害; 1 - タグの値が一致する場合はすべての障害。
correlation_tag	string	一致判定に使用するタグ。
manual_close	integer	手動クローズを許可するかどうか。 指定可能な値: 0 - *(デフォルト)* いいえ; 1 - はい。
discover	integer	トリガープロトタイプのディスカバリ状態。 指定可能な値: 0 - *(デフォルト)* 新しいトリガーがディスカバリされる; 1 - 新しいトリガーはディスカバリされず、既存のトリガーは消失としてマークされる。
uuid	string	インポートされたトリガープロトタイプを既存のものにリンクするために使用される汎用一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - トリガープロトタイプがテンプレートに属している場合に サポート

トリガープロトタイプタグ

トリガープロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	トリガープロトタイプタグ名。 Property behavior: - *required*
value	string	トリガープロトタイプタグの値。

triggerprototype.create

説明

`object triggerprototype.create(object/array triggerPrototypes)`

このメソッドでは、新しいトリガープロトタイプを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するトリガープロトタイプ。

このメソッドは、標準のトリガープロトタイププロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
dependencies	array	トリガープロトタイプが依存するトリガーおよびトリガープロトタイプ。 トリガーでは、`triggerid` プロパティのみを定義する必要があります。
tags	array	トリガープロトタイプタグ。

戻り値

(`object`) `triggerids` プロパティの下に作成されたトリガープロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたトリガープロトタイプの順序と一致します。

例

トリガープロトタイプの作成

ファイルシステムの空きディスク容量が 20% 未満になったことを検出するトリガープロトタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": " ボリューム {#FSNAME} の空きディスク容量が 20% 未満です",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::create() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

説明

`object triggerprototype.delete(array triggerPrototypeIds)`

このメソッドは、トリガーのプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するトリガープロトタイプの ID。

戻り値

(object) triggerids プロパティの下にある削除されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

複数のトリガープロトタイプの削除

2 つのトリガープロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

説明

`integer/array triggerprototype.get(object parameters)`

このメソッドは、指定したパラメーターに応じたトリガーのプロトタイプを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
active	flag	監視中のホストに属する、有効なトリガープロトタイプのみを返します。
discoveryids	ID/array	指定した LLD ルールに属するトリガープロトタイプのみを返します。
functions	string/array	指定した関数を使用するトリガーのみを返します。 サポートされている関数の一覧は、Supported functionsページを参照してください。
group	string	指定した名前のホストグループまたはテンプレートグループに属するホストまたはテンプレートに属するトリガープロトタイプのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートに属するトリガープロトタイプのみを返します。
host	string	指定した名前のホストに属するトリガープロトタイプのみを返します。
hostids	ID/array	指定したホストに属するトリガープロトタイプのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承されたトリガープロトタイプのみを返します。
maintenance	boolean	`true` に設定した場合、メンテナンス中のホストに属する、有効なトリガープロトタイプのみを返します。
min_severity	integer	指定した深刻度以上のトリガープロトタイプのみを返します。
monitored	flag	監視中のホストに属し、かつ有効なアイテムのみを含む、有効なトリガープロトタイプのみを返します。
templated	boolean	`true` に設定した場合、テンプレートに属するトリガープロトタイプのみを返します。
templateids	ID/array	指定したテンプレートに属するトリガープロトタイプのみを返します。
triggerids	ID/array	指定した ID を持つトリガープロトタイプのみを返します。
expandExpression	flag	トリガー式内の関数およびマクロを展開します。
selectDependencies	query	トリガープロトタイプが依存しているトリガープロトタイプおよびトリガーを、`dependencies` プロパティで返します。

Parameter	Type	Description
selectDiscoveryData	query	トリガープロトタイプのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。トリガープロトタイプのディスカバリオブジェクトは、検出されたトリガープロトタイプを、その元となったトリガープロトタイプに関連付けます。 このオブジェクトは以下のプロパティを持ちます。 `parent_triggerid` - (ID) トリガープロトタイプの作成元となったトリガープロトタイプの ID。 `status` - (int) トリガープロトタイプのディスカバリステータス: 0 - *(default)* トリガープロトタイプは検出されています。 1 - トリガープロトタイプはすでに検出されていません。 `ts_delete` - (timestamp) すでに検出されていないトリガープロトタイプが削除される時刻。 `ts_disable` - (timestamp) すでに検出されていないトリガープロトタイプが無効化される時刻。 `disable_source` - (int) トリガープロトタイプが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(default)* 自動的に無効化。 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	トリガープロトタイプが属するLLD ruleを `discoveryRule` プロパティで返します。
selectDiscoveryRulePrototype	query	トリガープロトタイプが属する親 LLD ルールプロトタイプを含む`discoveryRulePrototype` プロパティを返します。
selectFunctions	query	トリガープロトタイプで使用されている関数を `functions` プロパティで返します。 関数オブジェクトはトリガー式で使用される関数を表し、以下のプロパティを持ちます。 `functionid` - (ID) 関数の ID。 `itemid` - (ID) 関数で使用されるアイテムの ID。 `function` - (string) 関数名。 `parameter` - (string) 関数に渡されるパラメータ。クエリパラメータは、返される文字列では $ 記号に置き換えられます。
selectHostGroups	query	トリガープロトタイプが属するホストグループを`hostgroups` プロパティで返します。
selectHosts	query	トリガープロトタイプが属するホストを`hosts` プロパティで返します。
selectItems	query	トリガープロトタイプで使用されるアイテムおよびアイテムプロトタイプを`items` プロパティで返します。
selectTags	query	トリガープロトタイプのタグを`tags` プロパティで返します。
selectTemplateGroups	query	トリガープロトタイプが属するテンプレートグループを`templategroups` プロパティで返します。
filter	object	指定したフィルターに完全一致する結果のみを返します。 プロパティ名をキーとし、照合する値または値の配列を値とするオブジェクトを受け付けます。 `text` data type のプロパティはサポートしていません。 追加のプロパティをサポートしています。 `host` - トリガープロトタイプが属するホストの技術名。 `hostid` - トリガープロトタイプが属するホストの ID。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます。 `selectHosts` - 結果は `host` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	これらのパラメータはreference commentaryで説明されています。
editable	boolean	

Parameter	Type	Description
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクトの数

例

LLD ルールからトリガープロトタイプの取得

すべてのトリガープロトタイプとその関数を LLD ルールから取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Free inodes is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "6ce467d05e8745409a177799bed34bb3",
```

```
            "url_name": "",
            "functions": [
                {
                    "functionid": "12598",
                    "itemid": "22454",
                    "triggerid": "13272",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
                    "parameter": "$",
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```


タグで特定のトリガープロトタイプを取得

リクエスト：


```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
```

```
            "17373"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Free disk space is less than 20% on volume {#FSNAME}",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- ディスカバリルール
- アイテム
- ホスト
- ホストグループ
- テンプレートグループ

ソース

ui/include/classes/api/services/CTriggerPrototype.php 内の CTriggerPrototype::get()。

triggerprototype.update

説明

`object triggerprototype.update(object/array triggerPrototypes)`

このメソッドは、既存のトリガープロトタイプを更新するためのものです。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 更新するトリガープロトタイプのプロパティ。

各トリガープロトタイプでは `triggerid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のトリガープロトタイプのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
dependencies	array	トリガープロトタイプが依存するトリガーおよびトリガープロトタイプ。 トリガーでは `triggerid` プロパティのみを定義する必要があります。

パラメータ	型	説明
tags	array	トリガープロトタイプタグ。

戻り値

(object) triggerids プロパティの下で更新されたトリガープロトタイプの ID を含むオブジェクトを返します。

例

トリガープロトタイプの有効化

トリガープロトタイプを有効化します。つまり、そのステータスを「0」に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

トリガープロトタイプのタグの置き換え

1 つのトリガープロトタイプのタグを置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
```

```
            "17373"
        ]
    },
    "id": 1
}
```

ソース

CTriggerPrototype::update() in *ui/include/classes/api/services/CTriggerPrototype.php*.

トレンド

このクラスはトレンドデータを扱うために設計されています。

オブジェクト参照:

- 浮動小数点トレンド
- 整数トレンド

利用可能なメソッド:

- trend.get - トレンドを取得します

Trend オブジェクト

次のオブジェクトは、trendAPI に直接関連しています。

> **Note:**
> トレンドオブジェクトは、アイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API によって変更することはできません。

浮動小数点トレンド

浮動小数点トレンドオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	値が計算された時間のタイムスタンプ。たとえば、”04:00:00” のタイムスタンプは、”04:00:00-04:59:59” の期間について計算された値を意味します。
itemid	ID	関連するアイテムの ID。
num	integer	その時間に利用可能だった値の数。
value_min	float	1 時間あたりの最小値。
value_avg	float	1 時間あたりの平均値。
value_max	float	1 時間あたりの最大値。

整数トレンド

整数トレンドオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	値が計算された 1 時間のタイムスタンプ。例えば、”04:00:00” のタイムスタンプは、”04:00:00-04:59:59” の期間について計算された値を意味します。
itemid	ID	関連するアイテムの ID。
num	integer	その 1 時間で利用可能だった値の数。
value_min	integer	1 時間の最小値。
value_avg	integer	1 時間の平均値。
value_max	integer	1 時間の最大値。

trend.get

説明

`integer/array trend.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってトレンドデータを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。
> このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。
> 詳細は、ユーザーロールを参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
itemids	ID/array	指定したアイテム ID を持つトレンドのみを返します。
time_from	timestamp	指定した時刻以降（その時刻を含む）に収集された値のみを返します。
time_till	timestamp	指定した時刻以前（その時刻を含む）に収集された値のみを返します。
countOutput	boolean	取得したオブジェクト数をカウントします。
limit	integer	取得するオブジェクト数を制限します。
output	query	返されるTrend object のプロパティを設定します。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

アイテムのトレンドデータの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
```

```
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```


ソース

CTrend::get() in *ui/include/classes/api/services/CTrend.php*.

プロキシグループ

このクラスは、プロキシグループを操作するために設計されています。

オブジェクト参照:

- プロキシグループ

利用可能なメソッド:

- proxygroup.create - 新しいプロキシグループを作成
- proxygroup.delete - プロキシグループを削除
- proxygroup.get - プロキシグループを取得
- proxygroup.update - プロキシグループを更新

Proxy group オブジェクト

次のオブジェクトは、proxygroupAPI に直接関連しています。

プロキシグループ

プロキシグループオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
proxy_groupid	ID	プロキシグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
name	string	プロキシグループの名前。 プロパティの動作: - 作成操作では 必須
description	text	プロキシグループの説明。
failover_delay	string	プロキシグループ内のプロキシがオンラインと見なされるために、Zabbix server と通信していなければならない期間。 30s、1m などの時間サフィックスをサポートしています。 ユーザーマクロをサポートしています。 指定可能な値: 10s-15m。 デフォルト: 1m。
min_online	string	プロキシグループをオンライン状態に維持するために必要な、オンラインのプロキシの最小数。 ユーザーマクロをサポートしています。 指定可能な値の範囲: 1-1000。 デフォルト: 1。

プロパティ	型	説明
state	integer	プロキシグループの状態。 指定可能な値: 0 - 不明; 1 - オフライン; 2 - 復旧中; 3 - オンライン; 4 - 劣化中。 プロパティの動作: - 読み取り専用

proxygroup.create

説明

`object proxygroup.create(object/array proxyGroups)`

このメソッドは、新しいプロキシグループを作成します。

Note:
このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳しくは、ユーザーの役割 をご覧ください。

パラメータ

(object/array) 作成するプロキシグループ。

このメソッドは、標準のプロキシグループプロパティを持つプロキシグループを受け付けます。

戻り値

(object) は、proxy_groupids プロパティに作成されたプロキシグループの ID を含むオブジェクトを返します。返される ID の順序は、渡されたプロキシグループの順序と一致します。

例

プロキシグループを作成する

カスタム設定でプロキシグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::create() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

説明

`object proxygroup.delete(array proxyGroupIds)`

このメソッドではプロキシグループを削除できます。

> **Note:**
> このメソッドは *Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細についてはユーザーの役割を参照してください。

パラメータ

(array) 消去するプロキシグループの ID

戻り値

(object) proxy_groupids プロパティの下にある、削除されたプロキシグループの ID を含むオブジェクトを返す。

例

複数のプロキシグループを削除する

2 つのプロキシグループを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::delete() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

説明

`integer/array proxygroup.get(object parameters)`

このメソッドは、指定したパラメーターに応じたプロキシグループを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
proxy_groupids	ID/array	指定した ID を持つプロキシグループのみを返します。
proxyids	ID/array	指定したプロキシを含むプロキシグループのみを返します。
selectProxies	query	プロキシグループに属するプロキシを含むproxies プロパティを返します。 count をサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: proxy_groupid, name。
countOutput	boolean	これらのパラメータについては、リファレンスコメント を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

全てのプロキシグループの取得

すべての構成済みのプロキシとプロキシグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "proxy_groupid": "1",
```

```
            "name": "Proxy group 1",
            "failover_delay": "1m",
            "min_online": "3",
            "description": "",
            "state": "1",
            "proxies": [
                {
                    "proxyid": "1",
                    "name": "proxy 1"
                },
                {
                    "proxyid": "2",
                    "name": "proxy 2"
                }
            ]
        },
        {
            "proxy_groupid": "2",
            "name": "Proxy group 2",
            "failover_delay": "10m",
            "min_online": "3",
            "description": "",
            "state": "3",
            "proxies": [
                {
                    "proxyid": "3",
                    "name": "proxy 3"
                },
                {
                    "proxyid": "4",
                    "name": "proxy 4"
                },
                {
                    "proxyid": "5",
                    "name": "proxy 5"
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- Proxy

ソース

CProxyGroup::get() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

説明

`object proxygroup.update(object/array proxies)`

このメソッドは、既存のプロキシグループを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するプロキシグループのプロパティ。

各プロキシグループでは proxy_groupid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準のプロキシグループプロパティ を持つプロキシグループを受け付けます。

戻り値

(object) proxy_groupids プロパティの下にある、更新されたプロキシグループの ID を含むオブジェクトを返す。

例

オンラインプロキシの最小数を変更する

グループがオンラインになるために必要なオンラインプロキシの最小数を変更します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CProxyGroup::update() in *ui/include/classes/api/services/CProxyGroup.php*.

ホスト

このクラスはホストを操作するために設計されています。

オブジェクト参照:

- ホスト
- ホストインベントリ
- ホストタグ

利用可能なメソッド:

- host.create - 新しいホストを作成
- host.delete - ホストを削除
- host.get - ホストを取得
- host.massadd - 関連オブジェクトをホストに追加
- host.massremove - 関連オブジェクトをホストから削除
- host.massupdate - ホストの関連オブジェクトを置換または削除
- host.update - ホストを更新

Host オブジェクト

次のオブジェクトは、hostAPI に直接関連しています。

ホスト

ホストオブジェクトには以下のプロパティがあります。

Property	Type	Description
hostid	ID	ホストの ID。 Property behavior: - *read-only* - 更新操作で *required*
host	string	ホストの技術名。 Property behavior: - 作成操作で *required*
description	text	ホストの説明。
flags	integer	ホストの生成元。 指定可能な値: 0 - 通常のホスト; 4 - プロトタイプから変換されたホスト。 Property behavior: - *read-only*
inventory_mode	integer	ホストインベントリの投入モード。 指定可能な値: -1 - *(default)* 無効; 0 - 手動; 1 - 自動。
ipmi_authtype	integer	IPMI 認証アルゴリズム。 指定可能な値: -1 - *(default)* デフォルト; 0 - なし; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+。
ipmi_password	string	IPMI パスワード。
ipmi_privilege	integer	IPMI 権限レベル。 指定可能な値: 1 - callback; 2 - *(default)* user; 3 - operator; 4 - admin; 5 - OEM。
ipmi_username	string	IPMI ユーザー名。
maintenance_from	timestamp	有効なメンテナンスの開始時刻。 Property behavior: - *read-only*
maintenance_status	integer	有効なメンテナンスのステータス。 指定可能な値: 0 - *(default)* メンテナンスなし; 1 - メンテナンス有効。 Property behavior: - *read-only*

Property	Type	Description
maintenance_type	integer	有効なメンテナンスタイプ。 指定可能な値: 0 - *(default)* データ収集ありのメンテナンス; 1 - データ収集なしのメンテナンス。 Property behavior: - *read-only*
maintenanceid	ID	現在ホストに対して有効なメンテナンスの ID。 Property behavior: - *read-only*
name	string	ホストの表示名。 デフォルト: `host` プロパティの値。
monitored_by	integer	ホストの監視に使用されるソース。 指定可能な値: 0 - *(default)* Zabbix server; 1 - プロキシ; 2 - プロキシグループ。
proxyid	ID	ホストの監視に使用されるプロキシの ID。 Property behavior: - `monitored_by` が”Proxy” に設定されている場合は *required*
proxy_groupid	ID	ホストの監視に使用されるプロキシグループの ID。 Property behavior: - `monitored_by` が”Proxy group” に設定されている場合は *required*
status	integer	ホストのステータスと機能。 指定可能な値: 0 - *(default)* 有効; 1 - 無効。
tls_connect	integer	ホストへの接続。 指定可能な値: 1 - *(default)* 暗号化なし; 2 - PSK; 4 - 証明書。
tls_accept	integer	ホストからの接続。 指定可能なビットマップ値: 1 - *(default)* 暗号化なし; 2 - PSK; 4 - 証明書。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、PSK と証明書の場合は 6）。
tls_issuer	string	証明書の発行者。
tls_subject	string	証明書のサブジェクト。
tls_psk_identity	string	PSK ID。1 つの PSK に対して 1 つだけ対応している必要があります (autoregistration、hosts、proxies 全体で一意)。 受信側に使用する PSK を知らせるために、PSK ID はネットワーク上で暗号化されずに送信されます。そのため、PSK ID には機密情報を含めないでください。 Property behavior: - *write-only* - `tls_connect` が”PSK” に設定されている場合、または `tls_accept` に”PSK” ビットが含まれている場合は *required*

Property	Type	Description
tls_psk	string	事前共有鍵 (PSK)。32 桁以上の 16 進数である必要があります。 Property behavior: - *write-only* - `tls_connect` が”PSK” に設定されている場合、または `tls_accept` に”PSK” ビットが含まれている場合は *required*
active_available	integer	ホストのアクティブインターフェースの可用性ステータス。 指定可能な値: 0 - インターフェースのステータスは不明; 1 - インターフェースは利用可能; 2 - インターフェースは利用不可。 Property behavior: - *read-only*
assigned_proxyid	ID	ホストがプロキシグループによって監視されている場合に、Zabbix server によって割り当てられたプロキシの ID。 Property behavior: - *read-only*

ホストインベントリ

ホストインベントリオブジェクトには、以下のプロパティがあります。

Note:
各プロパティには固有の ID 番号があり、これはホストインベントリフィールドをアイテムに関連付けるために使用されます。

ID	プロパティ	Type	説明	最大長
4	alias	string	別名。	128 文字
11	asset_tag	string	資産タグ。	64 文字
28	chassis	string	シャーシ。	64 文字
23	contact	string	連絡先担当者。	SQL データベースでは 65535 文字
32	contract_number	string	契約番号。	64 文字
47	date_hw_decomm	string	ハードウェア廃止日。	64 文字
46	date_hw_expiry	string	ハードウェア保守期限日。	64 文字
45	date_hw_install	string	ハードウェア設置日。	64 文字
44	date_hw_purchase	string	ハードウェア購入日。	64 文字
34	deployment_status	string	展開ステータス。	64 文字
14	hardware	string	ハードウェア。	255 文字
15	hardware_full	string	ハードウェア詳細。	SQL データベースでは 65535 文字
39	host_netmask	string	ホストのサブネットマスク。	39 文字
38	host_networks	string	ホストネットワーク。	SQL データベースでは 65535 文字
40	host_router	string	ホストルーター。	39 文字
30	hw_arch	string	ハードウェアアーキテクチャ。	32 文字
33	installer_name	string	設置担当者名。	64 文字
24	location	string	場所。	SQL データベースでは 65535 文字
25	location_lat	string	場所の緯度。	16 文字
26	location_lon	string	場所の経度。	16 文字
12	macaddress_a	string	MAC アドレス A。	64 文字
13	macaddress_b	string	MAC アドレス B。	64 文字
29	model	string	モデル。	64 文字
3	name	string	名前。	128 文字
27	notes	string	メモ。	SQL データベースでは 65535 文字
41	oob_ip	string	OOB IP アドレス。	39 文字
42	oob_netmask	string	OOB ホストのサブネットマスク。	39 文字
43	oob_router	string	OOB ルーター。	39 文字
5	os	string	OS 名。	128 文字
6	os_full	string	詳細 OS 名。	255 文字
7	os_short	string	短縮 OS 名。	128 文字

ID	プロパティ	Type	説明	最大長
61	poc_1_cell	string	主要 POC の携帯電話番号。	64 文字
58	poc_1_email	string	主要メールアドレス。	128 文字
57	poc_1_name	string	主要 POC 名。	128 文字
63	poc_1_notes	string	主要 POC メモ。	SQL データベースでは 65535 文字
59	poc_1_phone_a	string	主要 POC 電話番号 A。	64 文字
60	poc_1_phone_b	string	主要 POC 電話番号 B。	64 文字
62	poc_1_screen	string	主要 POC スクリーン名。	64 文字
68	poc_2_cell	string	副次 POC の携帯電話番号。	64 文字
65	poc_2_email	string	副次メールアドレス。	128 文字
64	poc_2_name	string	副次 POC 名。	128 文字
70	poc_2_notes	string	副次 POC メモ。	SQL データベースでは 65535 文字
66	poc_2_phone_a	string	副次 POC 電話番号 A。	64 文字
67	poc_2_phone_b	string	副次 POC 電話番号 B。	64 文字
69	poc_2_screen	string	副次 POC スクリーン名。	64 文字
8	serialno_a	string	シリアル番号 A。	64 文字
9	serialno_b	string	シリアル番号 B。	64 文字
48	site_address_a	string	サイト住所 A。	128 文字
49	site_address_b	string	サイト住所 B。	128 文字
50	site_address_c	string	サイト住所 C。	128 文字
51	site_city	string	サイト市区町村。	128 文字
53	site_country	string	サイト国名。	64 文字
56	site_notes	string	サイトメモ。	SQL データベースでは 65535 文字
55	site_rack	string	サイトのラック位置。	128 文字
52	site_state	string	サイト都道府県/州。	64 文字
54	site_zip	string	サイト ZIP/郵便番号。	64 文字
16	software	string	ソフトウェア。	255 文字
18	software_app_a	string	ソフトウェアアプリケーション A。	64 文字
19	software_app_b	string	ソフトウェアアプリケーション B。	64 文字
20	software_app_c	string	ソフトウェアアプリケーション C。	64 文字
21	software_app_d	string	ソフトウェアアプリケーション D。	64 文字
22	software_app_e	string	ソフトウェアアプリケーション E。	64 文字
17	software_full	string	ソフトウェア詳細。	SQL データベースでは 65535 文字
10	tag	string	タグ。	64 文字
1	type	string	タイプ。	64 文字
2	type_full	string	タイプ詳細。	64 文字
35	url_a	string	URL A。	2048 文字
36	url_b	string	URL B。	2048 文字
37	url_c	string	URL C。	2048 文字
31	vendor	string	ベンダー。	64 文字

ホストタグ

ホストタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	ホストタグ名。 プロパティの動作: - 必須
value	string	ホストタグの値。
automatic	integer	ホストタグのタイプ。 使用可能な値: 0 - *(デフォルト)* 手動（ユーザーが作成したタグ）; 1 - 自動（ローレベルディスカバリによって作成されたタグ） プロパティの動作: - 読み取り専用

host.create

説明

`object host.create(object/array hosts)`

このメソッドでは、新しいホストを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(`object/array`) 作成するホスト。

このメソッドは、標準のホストプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	ホストを追加するホストグループ。 ホストグループでは、`groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
interfaces	object/array	ホスト用に作成されるインターフェース。
tags	object/array	ホストタグ。
templates	object/array	ホストにリンクするテンプレート。 テンプレートでは、`templateid` プロパティのみを定義する必要があります。
macros	object/array	ホスト用に作成されるユーザーマクロ。
inventory	object	ホストインベントリのプロパティ。

戻り値

(`object`) `hostids` プロパティの下に生成されたホストの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストの順序と一致します。

例

ホストの作成

「Linux server」という名前のホストを IP インターフェースとタグ付きで作成し、グループに追加し、テンプレートをリンクして、ホストインベントリに MAC アドレスを設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Linux server",
        "interfaces": [
            {
                "type": 1,
                "main": 1,
                "useip": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050"
            }
        ],
        "groups": [
            {
```

```
                "groupid": "50"
            }
        ],
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server"
            }
        ],
        "templates": [
            {
                "templateid": "20045"
            }
        ],
        "macros": [
            {
                "macro": "{$USER_ID}",
                "value": "123321"
            },
            {
                "macro": "{$USER_LOCATION}",
                "value": "0:0:0",
                "description": "latitude, longitude and altitude coordinates"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

SNMP インターフェースを持つホストの作成

詳細を指定した SNMPv3 インターフェースを持つ「SNMP host」という名前のホストを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "SNMP host",
        "interfaces": [
            {
                "type": 2,
                "main": 1,
                "useip": 1,
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
```

```
                "details": {
                    "version": 3,
                    "bulk": 0,
                    "securityname": "mysecurityname",
                    "contextname": "",
                    "securitylevel": 1
                }
            }
        ],
        "groups": [
            {
                "groupid": "4"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

PSK 暗号化を使用するホストの作成

PSK 暗号化を設定した「PSK host」という名前のホストを作成します。
このホストは、PSK を使用するよう事前設定されている必要があることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "PSK host",
        "interfaces": [
            {
                "type": 1,
                "ip": "192.168.3.1",
                "dns": "",
                "port": "10050",
                "useip": 1,
                "main": 1
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "tls_accept": 2,
        "tls_connect": 2,
        "tls_psk_identity": "PSK 001",
        "tls_psk": "1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

プロキシで監視するホスト情報の作成

ID "1" のプロキシで監視するホスト情報を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

プロキシグループで監視するホストの作成

ID "1" のプロキシグループで監視するホストを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

```
{
"jsonrpc": "2.0",
"method": "host.create",
"params": {
"host": " プロキシグループで監視するホスト",
"groups": [
{
"groupid": "2"
}
],
"monitored_by": 2,
"proxy_groupid": 1
},
"id": 1
}
```

レスポンス:

```
{
"jsonrpc": "2.0",
"result": {
"hostids": [
"10592"
]
},
"id": 1
}
```

関連項目

- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース
- ホストインベントリ
- ホストタグ
- プロキシ
- プロキシグループ

ソース

CHost::create() in *ui/include/classes/api/services/CHost.php*.

host.delete

説明

`object host.delete(array hosts)`

このメソッドは、ホストを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストの ID。

戻り値

(object) hostids プロパティの下にある削除されたホストの ID を含むオブジェクトを返します。

例

複数のホストの削除

2 つのホストを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

ソース

CHost::delete() in *ui/include/classes/api/services/CHost.php*.

host.get

説明

`integer/array host.get(object parameters)`

このメソッドは、与えられたパラメータにしたがってホストを取得することができます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
groupids	ID/array	指定したグループに属するホストのみを返します。
dserviceids	ID/array	指定したディスカバリサービスに関連するホストのみを返します。
graphids	ID/array	指定したグラフを持つホストのみを返します。
hostids	ID/array	指定したホスト ID を持つホストのみを返します。
httptestids	ID/array	指定した Web 監視を持つホストのみを返します。
interfaceids	ID/array	指定したインターフェースを使用するホストのみを返します。
itemids	ID/array	指定したアイテムを持つホストのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるホストのみを返します。
monitored_hosts	flag	監視対象のホストのみを返します。

パラメータ	型	説明
proxyids	ID/array	指定したプロキシによって監視されているホストのみを返します。
proxy_groupids	ID/array	指定したプロキシグループによって監視されているホストのみを返します。
templated_hosts	flag	ホストとテンプレートの両方を返します。
templateids	ID/array	指定したテンプレートにリンクされているホストのみを返します。
triggerids	ID/array	指定したトリガーを持つホストのみを返します。
with_items	flag	アイテムを持つホストのみを返します。 `with_monitored_items` および `with_simple_graph_items` パラメータを上書きします。
with_item_prototypes	flag	アイテムのプロトタイプを持つホストのみを返します。 `with_simple_graph_item_prototypes` パラメータを上書きします。
with_simple_graph_item_prototypes	flag	作成が有効で、かつ数値型の情報タイプを持つアイテムのプロトタイプを持つホストのみを返します。
with_graphs	flag	グラフを持つホストのみを返します。
with_graph_prototypes	flag	グラフのプロトタイプを持つホストのみを返します。
with_httptests	flag	Web 監視を持つホストのみを返します。 `with_monitored_httptests` パラメータを上書きします。
with_monitored_httptests	flag	有効な Web 監視を持つホストのみを返します。
with_monitored_items	flag	有効なアイテムを持つホストのみを返します。 `with_simple_graph_items` パラメータを上書きします。
with_monitored_triggers	flag	有効なトリガーを持つホストのみを返します。トリガーで使用されるすべてのアイテムも有効である必要があります。
with_simple_graph_items	flag	数値型の情報タイプを持つアイテムを持つホストのみを返します。
with_triggers	flag	トリガーを持つホストのみを返します。 `with_monitored_triggers` パラメータを上書きします。
withProblemsSuppressed	boolean	`true` に設定すると、抑制された障害を持つホストのみを返します。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
severities	integer/array	指定した深刻度の障害のみを持つホストを返します。problem オブジェクトがトリガーである場合にのみ適用されます。
tags	object/array	指定したタグを持つホストのみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべてのホストを返します。 指定可能なoperatorの値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
inheritedTags	boolean	指定した `tags` を、リンクされているすべてのテンプレートにも持つホストを返します。 指定可能な値: `true` - リンクされたテンプレートも指定した `tags` を持っている必要があります; `false` - *(デフォルト)* リンクされたテンプレートのタグは無視されます。

パラメータ	型	説明
selectDiscoveryData	query	ホストディスカバリオブジェクトのデータを含む discoveryData プロパティを返します。ホストディスカバリオブジェクトは、ディスカバリされたホストを、そのホストの検出元となったホストのプロトタイプに関連付けます。 以下のプロパティを持ちます: host - (string) ホストの ID; parent_hostid - (string) ホストの作成元となったホストのプロトタイプの ID; status - (int) ホストディスカバリのステータス: 0 - *(デフォルト)* ホストはディスカバリされている, 1 - ホストはすでにディスカバリされていない; ts_delete - (timestamp) すでにディスカバリされていないホストが削除される時刻; ts_disable - (timestamp) すでにディスカバリされていないホストが無効化される時刻; disable_source - (int) ホストが LLD ルールによって無効化されたか、手動で無効化されたかを示す指標: 0 - *(デフォルト)* 自動的に無効化, 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	ホストを作成したローレベルディスカバリルール（VMware 監視ではホストのプロトタイプから作成）を含むdiscoveryRule プロパティを返します。
selectDiscoveryRules	query	ホストの LLD ルールを含むdiscoveryRules プロパティを返します。 count をサポートします。
selectGraphs	query	ホストのグラフを含むgraphs プロパティを返します。 count をサポートします。
selectHostGroups	query	ホストが属するホストグループのデータを含むhostgroups プロパティを返します。
selectHttpTests	query	ホストの Web シナリオを含むhttpTests プロパティを返します。 count をサポートします。
selectInterfaces	query	ホストインターフェースを含むinterfaces プロパティを返します。 count をサポートします。
selectInventory	query	ホストインベントリデータを含むinventory プロパティを返します。
selectItems	query	ホストのアイテムを含むitems プロパティを返します。 count をサポートします。
selectMacros	query	ホストマクロを含むmacros プロパティを返します。
selectParentTemplates	query	ホストにリンクされているテンプレートを含む parentTemplates プロパティを返します。 テンプレートオブジェクトのフィールドに加えて、link_type - (integer) テンプレートがホストにリンクされる方法、も含まれます。 指定可能な値: 0 - *(デフォルト)* 手動でリンク; 1 - LLD によって自動的にリンク。 count をサポートします。
selectDashboards	query	dashboards プロパティを返します。 count をサポートします。
selectTags	query	ホストタグを含むtags プロパティを返します。
selectInheritedTags	query	ホストにリンクされているすべてのテンプレート上のタグを含むinheritedTags プロパティを返します。
selectTriggers	query	ホストのトリガーを含むtriggers プロパティを返します。 count をサポートします。

パラメータ	型	説明
selectValueMaps	query	ホストの値マップを含む`valuemaps` プロパティを返します。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型 のプロパティはサポートしません。 追加のプロパティをサポートします: ホストインターフェース のプロパティ。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectParentTemplates` - 結果は `host` でソートされます; `selectInterfaces`; `selectItems` - `name` でソートされます; `selectDiscoveryRules` - `name` でソートされます; `selectTriggers` - `description` でソートされます; `selectGraphs` - `name` でソートされます; `selectDashboards` - `name` でソートされます。
search	object	指定したパターンに一致する結果を返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索が実行されます。 `string` および `text` データ型 のプロパティのみをサポートします。 追加のプロパティをサポートします: ホストインターフェース のプロパティ。
searchInventory	object	指定したパターンに一致するインベントリデータを持つホストを返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索が実行されます。 `string` および `text` データ型 のプロパティのみをサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `hostid`, `host`, `name`, `status`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	ホストのローレベルディスカバリルールを含む`discoveries` プロパティを返します。 `count` をサポートします。 このクエリは非推奨です。代わりに `selectDiscoveryRules` を使用してください。

パラメータ	型	説明
selectHostDiscovery	query	ホストディスカバリオブジェクトのデータを含む hostDiscovery プロパティを返します。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

名前によるデータの取得

"Zabbix server" と"Linux server" という名前の 2 つのホストに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "proxyid": "0",
            "host": "Zabbix server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Zabbix server",
            "flags": "0",
            "description": "The Zabbix monitoring server.",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
```

```
        },
        {
            "hostid": "10167",
            "proxyid": "0",
            "host": "Linux server",
            "status": "0",
            "ipmi_authtype": "-1",
            "ipmi_privilege": "2",
            "ipmi_username": "",
            "ipmi_password": "",
            "maintenanceid": "0",
            "maintenance_status": "0",
            "maintenance_type": "0",
            "maintenance_from": "0",
            "name": "Linux server",
            "flags": "0",
            "description": "",
            "tls_connect": "1",
            "tls_accept": "1",
            "tls_issuer": "",
            "tls_subject": "",
            "proxy_groupid": "0",
            "monitored_by": "0",
            "inventory_mode": "1",
            "active_available": "1",
            "assigned_proxyid": "0"
        }
    ],
    "id": 1
}
```

ホストグループの取得

ホスト"Zabbix server" が所属するホストグループ取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "hostgroups": [
                {
                    "groupid": "2",
                    "name": "Linux servers",
                    "flags": "0",
                    "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
```

```
                },
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ]
        }
    ],
    "id": 1
}
```

リンク先テンプレートの取得

ホスト”10084” にリンクしているテンプレートの ID および名前を取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

テンプレートでホストの取得

テンプレート”10001”(*Linux by Zabbix agent*) がリンクされているホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

ホストインベントリのデータで検索

ホストインベントリの"OS" フィールドに"Linux" が含まれるホストを検索します

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
```

```
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

ホストタグによる検索

タグ”host-name” が”linux-server” に等しいホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

ホストレベルだけでなく、リンクされた親テンプレートにもこれらのタグを持つホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
```

```
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

タグ”component” を持つホストを除外して取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "tags": [
                {
                    "tag": "class",
                    "value": "os",
                    "automatic": "0"
                },
                {
```

```
                    "tag": "target",
                    "value": "linux",
                    "automatic": "0"
                }
            ]
        },
        {
            "hostid": "10629",
            "tags": [
                {
                    "tag": "class",
                    "value": "application",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "browser",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

タグとテンプレートタグでホストの検索

タグと親テンプレートにリンクされているすべてのタグでホストを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10502",
            "name": "Desktop",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ],
            "inheritedTags": [
                {
                    "tag": "class",
                    "value": "os"
```

```
                },
                {
                    "tag": "target",
                    "value": "linux"
                },
                {
                    "tag": "os",
                    "value": "rhel-7"
                }
            ]
        }
    ],
    "id": 1
}
```

障害の深刻度でホストの検索

”Disaster” の障害を持つホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

”Average” と ”High” の障害を持つホストを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
```

```
        {
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

ホストのアイテムの取得

"Zabbix server" ホストのアイテムを取得します。リクエストは、アイテムのプロパティ itemid、name、status に限定されています。レスポンスサイズが大きいため、この例ではアイテムの一部のみを示しています。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Host name of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix agent ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Maximum number of open file descriptors",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maximum number of processes",
          "status": "0"
        },
        {
          "itemid": "69869",
```

```
            "name": "Interface enp0s3: Inbound packets discarded",
            "status": "0"
          },
          {
            "itemid": "69870",
            "name": "Interface enp0s3: Inbound packets with errors",
            "status": "0"
          },
          {
            "itemid": "69871",
            "name": "Interface enp0s3: Bits received",
            "status": "0"
          },
          {
            "itemid": "69872",
            "name": "Interface enp0s3: Outbound packets discarded",
            "status": "0"
          },
          {
            "itemid": "69873",
            "name": "Interface enp0s3: Outbound packets with errors",
            "status": "0"
          },
          {
            "itemid": "69874",
            "name": "Interface enp0s3: Bits sent",
            "status": "0"
          },
          {
            "itemid": "42253",
            "name": "Number of processes",
            "status": "0"
          },
          {
            "itemid": "42254",
            "name": "Number of running processes",
            "status": "0"
          },
          {
            "itemid": "42255",
            "name": "System boot time",
            "status": "0"
          }
        ]
      }
    ],
    "id": 1
}
```


関連項目

- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース
- プロキシ
- プロキシグループ

ソース

CHost::get() in *ui/include/classes/api/services/CHost.php*.

host.massadd

説明

`object host.massadd(object parameters)`

このメソッドを使用すると、指定したすべてのホストに複数の関連オブジェクトを同時に追加できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 更新するホストの ID と、すべてのホストに追加するオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
hosts	object/array	更新するホスト。 ホストでは、hostid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
groups	object/array	指定したホストに追加するホストグループ。 ホストグループでは、groupid プロパティのみを定義する必要があります。
interfaces	object/array	指定したホスト用に作成するホストインターフェース。
macros	object/array	指定したホスト用に作成するユーザーマクロ。
templates	object/array	指定したホストにリンクするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

マクロの追加

2 つのホストに 2 つの新しいマクロを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
```

```
                "description": " テスト用の説明"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

関連項目

- host.update
- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース

ソース

CHost::massAdd() in *ui/include/classes/api/services/CHost.php*.

host.massremove

説明

`object host.massremove(object parameters)`

このメソッドでは、複数のホストから関連オブジェクトを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新するホストの ID と、削除する必要があるオブジェクトを含むパラメータです。

パラメータ	Type	説明
hostids	ID/array	更新するホストの ID です。 パラメータの動作: - 必須
groupids	ID/array	指定したホストを削除するホストグループの ID です。
interfaces	object/array	指定したホストから削除するホストインターフェースです。 ホストインターフェースオブジェクトでは、ip、dns、port プロパティのみを定義する必要があります。
macros	string/array	指定したホストから削除するユーザーマクロです。
templateids	ID/array	指定したホストからリンク解除するテンプレートの ID です。
templateids_clear	ID/array	指定したホストからリンク解除してクリアするテンプレートの ID です。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

テンプレートのリンク解除

2 つのホストからテンプレートのリンクを解除し、テンプレート化されたすべてのエンティティを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

関連項目

- host.update
- ユーザーマクロ
- ホストインターフェース

ソース

CHost::massRemove() in *ui/include/classes/api/services/CHost.php*.

host.massupdate

説明

`object host.massupdate(object parameters)`

このメソッドでは、関連オブジェクトの一括置換または削除と、複数のホストのプロパティ更新を同時に実行できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
> このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(`object`) 更新するホストの ID と、更新するプロパティを含むパラメータです。

このメソッドは、標準のホストプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
hosts	object/array	更新するホストです。 ホストには `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須

パラメータ	型	説明
groups	object/array	ホストが所属する現在のホストグループを置き換えるホストグループです。 ホストグループには groupid プロパティのみを定義する必要があります。
interfaces	object/array	指定したホスト上の現在のホストインターフェースを置き換えるホストインターフェースです。
inventory	object	ホストインベントリのプロパティです。 ホストインベントリモードは inventory パラメータでは更新できません。代わりに inventory_mode を使用してください。
macros	object/array	指定したホスト上の現在のユーザーマクロを置き換えるユーザーマクロです。
templates	object/array	指定したホストに現在リンクされているテンプレートを置き換えるテンプレートです。 テンプレートには templateid プロパティのみを定義する必要があります。
templates_clear	object/array	指定したホストからリンク解除してクリアするテンプレートです。 テンプレートには templateid プロパティのみを定義する必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

複数のホストの有効化

2 つのホストの監視を有効にします。つまり、それらのステータスを「0」に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.massupdate",
    "params": {
        "hosts": [
            {
                "hostid": "69665"
            },
            {
                "hostid": "69666"
            }
        ],
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

参照

- host.update
- host.massadd
- host.massremove
- Host group
- Template
- User macro
- Host interface

ソース

CHost::massUpdate() in *ui/include/classes/api/services/CHost.php*.

host.update

説明

`object host.update(object/array hosts)`

このメソッドは、既存のホストを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 更新するホストのプロパティ。

各ホストについて `hostid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。更新されるのは指定されたプロパティのみで、それ以外は変更されません。

ただし、ホストの技術名を更新すると、その値でホストの表示名も更新されます（表示名が別途指定されていない場合）。

このメソッドは、標準のホストプロパティ に加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
groups	object/array	ホストが所属する現在のホストグループを置き換えるためのホストグループ。 リクエストに記載されていないホストグループはすべてリンク解除されます。 ホストグループでは `groupid` プロパティのみを定義する必要があります。
interfaces	object/array	現在のホストインターフェースを置き換えるためのホストインターフェース。 リクエストに記載されていないインターフェースはすべて削除されます。
tags	object/array	現在のホストタグを置き換えるためのホストタグ。 リクエストに記載されていないタグはすべて削除されます。
inventory	object	ホストインベントリ のプロパティ。
macros	object/array	現在のユーザーマクロを置き換えるためのユーザーマクロ。 リクエストに記載されていないマクロはすべて削除されます。
templates	object/array	現在リンクされているテンプレートを置き換えるためのテンプレート。 リクエストに記載されていないテンプレートはリンク解除のみ行われます。 テンプレートでは `templateid` プロパティのみを定義する必要があります。
templates_clear	object/array	ホストからリンク解除してクリアするためのテンプレート。 テンプレートでは `templateid` プロパティのみを定義する必要があります。

戻り値

(object) hostids プロパティの下で更新されたホストの ID を含むオブジェクトを返します。

例

ホストの有効化

ホストの監視を有効にする、つまりステータスを 0 にします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

テンプレートのリンク削除

ホストから 2 つのテンプレートのリンクを削除してクリアします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

ホストマクロの更新

すべてのホストマクロを新しい 2 つのマクロに置き換えます。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Updated description"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

ホストインベントリの更新

インベントリモードの変更とロケーションの追加をします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

ホストタグの更新

すべてのホストタグを新しいものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

検出されたホストマクロの更新

作成した検出ルールの” 自動” マクロを” 手動” に変換し、その値を”new-value” に変更します。

リクエスト :

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

ホスト暗号化の更新

ホスト”10590” を更新して、ホストから Zabbix サーバーへの接続にのみ PSK 暗号化を使用するようにし、PSK ID と PSK キーを変更します。

ホストはPSK を使用するように事前設定する必要があることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10590",
        "tls_connect": 1,
        "tls_accept": 2,
        "tls_psk_identity": "PSK 002",
        "tls_psk": "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

関連項目

- host.massadd
- host.massupdate
- host.massremove
- ホストグループ
- テンプレート
- ユーザーマクロ
- ホストインターフェース
- ホストインベントリ
- ホストタグ
- プロキシ
- プロキシグループ

ソース

CHost::update() in *ui/include/classes/api/services/CHost.php*.

ホストインターフェース

このクラスはホストインターフェースを操作するために設計されています。

オブジェクト参照:

- ホストインターフェース
  - 詳細

利用可能なメソッド:

- hostinterface.create - 新しいホストインターフェースを作成
- hostinterface.delete - ホストインターフェースを削除
- hostinterface.get - ホストインターフェースを取得
- hostinterface.massadd - ホストにホストインターフェースを追加
- hostinterface.massremove - ホストからホストインターフェースを削除
- hostinterface.replacehostinterfaces - ホスト上のホストインターフェースを置き換え
- hostinterface.update - ホストインターフェースを更新

Host interface オブジェクト

次のオブジェクトは、hostinterfaceAPI に直接関連しています。

ホストインターフェース

ホストインターフェースオブジェクトには、以下のプロパティがあります。

Attention:
作成操作では、`ip` と `dns` の両方のプロパティが必須であることに注意してください。DNS を使用しない場合は、空文字列に設定してください。

Property	Type	Description
interfaceid	ID	インターフェースの ID。 Property behavior: - *read-only* - 更新操作で *required*
available	integer	ホストインターフェースの可用性。 指定可能な値: 0 - *(default)* 不明; 1 - 利用可能; 2 - 利用不可。 Property behavior: - *read-only*
hostid	ID	このインターフェースが属するホストの ID。 Property behavior: - *constant* - 作成操作で *required*
type	integer	インターフェースのタイプ。 指定可能な値: 1 - エージェント; 2 - SNMP; 3 - IPMI; 4 - JMX。 Property behavior: - 作成操作で *required*
ip	string	インターフェースで使用される IP アドレス。 DNS 経由で接続する場合は空にできます。 Property behavior: - 作成操作で *required*
dns	string	インターフェースで使用される DNS 名。 IP 経由で接続する場合は空にできます。 Property behavior: - 作成操作で *required*
port	string	インターフェースで使用されるポート番号。 ユーザーマクロを含めることができます。 Property behavior: - 作成操作で *required*

Property	Type	Description
useip	integer	IP 経由で接続するかどうか。 指定可能な値: 0 - ホストの DNS 名を使用して接続; 1 - ホストの IP アドレスを使用して接続。
main	integer	Property behavior: - 作成操作で *required* このインターフェースをホスト上のデフォルトとして使用するかどうか。 ホスト上では、特定のタイプごとに 1 つのインターフェースのみをデフォルトとして設定できます。 指定可能な値: 0 - デフォルトではない; 1 - デフォルト。
details	object	Property behavior: - 作成操作で *required* インターフェース用の追加のdetailsオブジェクト。
disable_until	timestamp	Property behavior: - `type` が”SNMP” に設定されている場合は *required* 利用不可のホストインターフェースに対する次回のポーリング時刻。
error	string	Property behavior: - *read-only* ホストインターフェースが利用不可の場合のエラーテキスト。
errors_from	timestamp	Property behavior: - *read-only* ホストインターフェースが利用不可になった時刻。
		Property behavior: - *read-only*

詳細

details オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
version	integer	SNMP インターフェースのバージョン。 指定可能な値: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3.
bulk	integer	プロパティの動作: - 必須 bulk SNMP リクエストを使用するかどうか。 指定可能な値: 0 - bulk リクエストを使用しない; 1 - （デフォルト）- bulk リクエストを使用する。
community	string	SNMP コミュニティ。SNMPv1 および SNMPv2 インターフェースでのみ使用されます。 プロパティの動作: - `version` が”SNMPv1” または”SNMPv2c” に設定されている場合は 必須

プロパティ	型	説明
max_repetitions	integer	ネイティブ SNMP bulk リクエスト（GetBulkRequest-PDUs）の最大繰り返し値。 SNMPv2 および v3 の `discovery[]` アイテムと `walk[]` アイテムでのみ使用されます。 デフォルト: 10。
securityname	string	SNMPv3 セキュリティ名。SNMPv3 インターフェースでのみ使用されます。
securitylevel	integer	SNMPv3 セキュリティレベル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - （デフォルト）- noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	SNMPv3 認証パスフレーズ。SNMPv3 インターフェースでのみ使用されます。
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。SNMPv3 インターフェースでのみ使用されます。
authprotocol	integer	SNMPv3 認証プロトコル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - （デフォルト）- MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - （デフォルト）- DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	SNMPv3 コンテキスト名。SNMPv3 インターフェースでのみ使用されます。

hostinterface.create

説明

`object hostinterface.create(object/array hostInterfaces)`

このメソッドでは、新しいホストインターフェースを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 作成するホストインターフェース。

このメソッドは、標準のホストインターフェースプロパティを持つホストインターフェースを受け付けます。

戻り値

(object) interfaceids プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストインターフェイスの順序と一致します。

例

新しいインターフェースを作成する

ホスト「30052」にセカンダリ IP エージェントインターフェースを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10050"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

SNMP の詳細を指定してインターフェースを作成する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
```

```
            "30063"
        ]
    },
    "id": 1
}
```

参照

- hostinterface.massadd
- host.massadd

ソース

CHostInterface::create() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

説明

`object hostinterface.delete(array hostInterfaceIds)`

このメソッドは、ホストインターフェースを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストインターフェースの ID。

戻り値

(object) interfaceids プロパティの下にある削除されたホストインターフェースの ID を含むオブジェクトを返します。

例

ホストインターフェースの削除

ID 30062 のホストインターフェースを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

参照

- hostinterface.massremove
- host.massremove

ソース

CHostInterface::delete() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

説明

`integer/array hostinterface.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってホストインターフェースを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
hostids	ID/array	指定したホストで使用されているホストインターフェースのみを返します。
interfaceids	ID/array	指定した ID を持つホストインターフェースのみを返します。
itemids	ID/array	指定したアイテムで使用されているホストインターフェースのみを返します。
triggerids	ID/array	指定したトリガー内のアイテムで使用されているホストインターフェースのみを返します。
selectItems	query	インターフェースを使用しているアイテムを含む`items` プロパティを返します。 `count` をサポートします。
selectHosts	query	インターフェースを使用しているホストの配列を含む`hosts` プロパティを返します。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectItems`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `interfaceid`, `dns`, `ip`。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストインターフェースの取得

ホスト「30057」で使用されているインターフェースに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "interfaceid": "50039",
            "hostid": "30057",
            "main": "1",
            "type": "1",
            "useip": "0",
            "ip": "",
            "dns": "localhost",
            "port": "10050",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": []
        },
        {
            "interfaceid": "55082",
            "hostid": "30057",
            "main": "1",
            "type": "2",
            "useip": "1",
            "ip": "127.0.0.1",
            "dns": "",
            "port": "161",
            "available": "0",
            "error": "",
            "errors_from": "0",
            "disable_until": "0",
            "details": {
                "version": "2",
                "bulk": "0",
                "community": "{$SNMP_COMMUNITY}",
                "max_repetitions": "10"
            }
        }
    ],
    "id": 1
}
```

参照

- Host
- Item

ソース

CHostInterface::get() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massadd

説明

`object hostinterface.massadd(object parameters)`

このメソッドを使用すると、複数のホストにホストインターフェースを同時に追加できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ使用できます。
> このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

指定されたホスト上に作成するホストインターフェースを含む（`object`）パラメータです。

このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
interfaces	object/array	指定されたホスト上に作成するホストインターフェース。 パラメータの動作: - 必須
hosts	object/array	更新対象のホスト。 ホストでは `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須

戻り値

（`object`）`interfaceids` プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。

例

インターフェースの作成

2 つのホストにインターフェースを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

参照

- hostinterface.create
- host.massadd
- Host

ソース

CHostInterface::massAdd() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

説明

`object hostinterface.massremove(object parameters)`

このメソッドでは、指定されたホストからホストインターフェースを削除できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

更新するホストの ID と、削除するインターフェースを含む（`object`）パラメータです。

パラメータ	Type	説明
interfaces	object/array	指定したホストから削除するホストインターフェース。 ホストインターフェースオブジェクトでは、`ip`、`dns`、`port` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
hostids	ID/array	更新するホストの ID。 パラメータの動作: - 必須

戻り値

（`object`）`interfaceids` プロパティの下で削除されたホストインターフェイスの ID を含むオブジェクトを返します。

例

インターフェースの削除

2 つのホストから”127.0.0.1” の SNMP インターフェースを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

参照

- hostinterface.delete
- host.massremove

ソース

CHostInterface::massRemove() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.replacehostinterfaces

説明

`object hostinterface.replacehostinterfaces(object parameters)`

このメソッドでは、指定したホスト上のすべてのホストインターフェースを置き換えることができます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object) 更新するホストの ID と新しいホストインターフェースを含むパラメータ。

パラメータ	Type	説明
interfaces	object/array	現在のホストインターフェースを置き換えるホストインターフェース。 パラメータの動作: - 必須
hostid	ID	更新するホストの ID。 パラメータの動作: - 必須

戻り値

(object) interfaceids プロパティの下で作成されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェースの置換

すべてのホストインターフェースを単一のエージェントインターフェースに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.replacehostinterfaces",
    "params": {
        "hostid": "30052",
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 1,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30081"
        ]
    },
    "id": 1
}
```

参照

- host.update
- host.massupdate

ソース

CHostInterface::replaceHostInterfaces() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

説明

`object hostinterface.update(object/array hostInterfaces)`

このメソッドは、既存のホストインターフェースを更新することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

更新する (object/array) のホストインターフェースのプロパティ。

各ホストインターフェースでは、interfaceid プロパティを必ず定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、それ以外は変更されません。

戻り値

(object) interfaceids プロパティの下で更新されたホストインターフェイスの ID を含むオブジェクトを返します。

例

ホストインターフェースポートの変更

ホストインターフェースのポートを変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "30050"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

ソース

CHostInterface::update() in *ui/include/classes/api/services/CHostInterface.php*.

ホストグループ

このクラスはホストグループを操作するために設計されています。

オブジェクト参照:

- ホストグループ

利用可能なメソッド:

- hostgroup.create - 新しいホストグループを作成
- hostgroup.delete - ホストグループを削除
- hostgroup.get - ホストグループを取得
- hostgroup.massadd - 関連オブジェクトをホストグループに追加
- hostgroup.massremove - 関連オブジェクトをホストグループから削除
- hostgroup.massupdate - ホストグループ内の関連オブジェクトを置換または削除
- hostgroup.propagate - 権限とタグフィルターをホストグループのサブグループに伝播
- hostgroup.update - ホストグループを更新

Host group オブジェクト

次のオブジェクトは、hostgroupAPI に直接関連しています。

ホストグループ

ホストグループオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
groupid	ID	ホストグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
name	string	ホストグループの名前。 プロパティの動作: - 作成操作では 必須
flags	integer	ホストグループの生成元。 指定可能な値: 0 - 通常のホストグループ; 4 - ディスカバリされたホストグループ。 プロパティの動作: - 読み取り専用
uuid	string	既存のホストグループにインポートしたホストグループを関連付けるために使用される汎用一意識別子。指定しない場合は自動生成されます。

hostgroup.create

説明

`object hostgroup.create(object/array hostGroups)`

このメソッドでは、新しいホストグループを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 作成するホストグループ。

このメソッドは、標準のホストグループプロパティを持つホストグループを受け付けます。

戻り値

(`object`) `groupids` プロパティの下に作成されたホストグループの ID を含むオブジェクトを返します。
 返される ID の順序は、渡されたホストグループの順序と一致します。

例

ホストグループの作成

「Linux servers」という名前のホストグループを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
```

```
            "107819"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::create() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

説明

`object hostgroup.delete(array hostGroupIds)`

このメソッドは、ホストグループを削除することができます。

以下の場合、ホストグループを削除することはできません。

- このグループのみに属するホストが含まれている。
- 内部としてマークされている。
- ホストプロトタイプで使用されている。
- グローバルスクリプトで使用されている。
- 相関関係で使用されている。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストグループの ID。

戻り値

(object) groupids プロパティの下にある削除されたホストグループの ID を含むオブジェクトを返します。

例

複数のホストグループの削除

2 つのホストグループを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::delete() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

説明

`integer/array hostgroup.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってホストグループを取得できます。

Note:
このメソッドは、すべてのユーザータイプで利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

`(object)` 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

Parameter	Type	Description
graphids	ID/array	指定したグラフを持つホストを含むホストグループのみを返します。
groupids	ID/array	指定したホストグループ ID を持つホストグループのみを返します。
hostids	ID/array	指定したホストを含むホストグループのみを返します。
maintenanceids	ID/array	指定したメンテナンスの影響を受けるホストグループのみを返します。
triggerids	ID/array	指定したトリガーを持つホストを含むホストグループのみを返します。
with_graphs	boolean	グラフを持つホストを含むホストグループのみを返します。
with_graph_prototypes	boolean	グラフプロトタイプを持つホストを含むホストグループのみを返します。
with_hosts	boolean	ホストを含むホストグループのみを返します。
with_httptests	boolean	Web 監視を持つホストを含むホストグループのみを返します。 `with_monitored_httptests` パラメータを上書きします。
with_items	boolean	アイテムを持つホストを含むホストグループのみを返します。 `with_monitored_items` および `with_simple_graph_items` パラメータを上書きします。
with_item_prototypes	boolean	アイテムプロトタイプを持つホストを含むホストグループのみを返します。 `with_simple_graph_item_prototypes` パラメータを上書きします。
with_simple_graph_item_prototypes	boolean	作成が有効で、かつ数値型の情報タイプを持つアイテムプロトタイプを含むホストを持つホストグループのみを返します。
with_monitored_httptests	boolean	有効な Web 監視を持つホストを含むホストグループのみを返します。
with_monitored_hosts	boolean	監視対象のホストを含むホストグループのみを返します。
with_monitored_items	boolean	有効なアイテムを持つホストを含むホストグループのみを返します。 `with_simple_graph_items` パラメータを上書きします。
with_monitored_triggers	boolean	有効なトリガーを持つホストを含むホストグループのみを返します。トリガーで使用されるすべてのアイテムも有効である必要があります。
with_simple_graph_items	boolean	数値型のアイテムを持つホストを含むホストグループのみを返します。
with_triggers	boolean	トリガーを持つホストを含むホストグループのみを返します。 `with_monitored_triggers` パラメータを上書きします。
selectDiscoveryRules	query	ホストグループを検出した LLD ルールを含む`discoveryRules` プロパティを返します。

Parameter	Type	Description
selectDiscoveryData	query	ホストグループ検出オブジェクトを含む discoveryData プロパティを返します。 各ホストグループ検出オブジェクトは、検出されたホストグループにリンクされたホストグループプロトタイプであり、以下のプロパティを持ちます。 parent_group_prototypeid - (ID) ホストグループの検出元となったホストグループプロトタイプの ID; name - (string) ホストグループプロトタイプの名前; ts_delete - (timestamp) すでに検出されなくなったホストグループが削除される時刻; status - (int) ホストグループの検出ステータス: 0 - *(default)* ホストグループは検出されている; 1 - ホストグループはすでに検出されていない。
selectHostPrototypes	query	このホストグループを検出したホストプロトタイプを含むhostPrototypes プロパティを返します。
selectHosts	query	ホストグループに属するホストを含むhosts プロパティを返します。 count をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: selectHosts - 結果は host でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: groupid, name。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroupDiscoveries	query	ホストグループ検出オブジェクトを含む groupDiscoveries プロパティを返します。 各ホストグループ検出オブジェクトは、検出されたホストグループにリンクされたホストグループプロトタイプです。 このクエリは非推奨です。代わりに selectDiscoveryData を使用してください。

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクト数

例

名前によるデータの取得

"Zabbix servers" および"Linux servers" という名前の 2 つのホストグループに関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

関連項目

- ホスト

ソース

CHostGroup::get() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

説明

`object hostgroup.massadd(object parameters)`

このメソッドでは、指定したすべてのホストグループに対して、複数の関連オブジェクトを同時に追加できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object) 更新するホストグループの ID と、すべてのホストグループに追加するオブジェクトを含むパラメータです。

このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
groups	object/array	更新するホストグループ。 ホストグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - 必須
hosts	object/array	すべてのホストグループに追加するホスト。 ホストでは、hostid プロパティのみを定義する必要があります。

戻り値

(object) groupids プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストグループへのホストの追加

ID 5 および 6 のホストグループに 2 つのホストを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

関連項目

- ホスト

ソース

CHostGroup::massAdd() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

説明

`object hostgroup.massremove(object parameters)`

このメソッドでは、複数のホストグループから関連オブジェクトを削除できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
> このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 更新するホストグループの ID と、削除するオブジェクトを含むパラメータ。

パラメータ	Type	説明
groupids	ID/array	更新するホストグループの ID。 パラメータの動作: - 必須
hostids	ID/array	すべてのホストグループから削除するホストの ID。

戻り値

(object) groupids プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストグループからのホストの削除

指定されたホストグループから 2 つのホストを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::massRemove() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massupdate

説明

`object hostgroup.massupdate(object parameters)`

このメソッドでは、複数のホストグループ内のホストおよびテンプレートを、指定したものに置き換えることができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 更新するホストグループの ID と、更新対象のオブジェクトを含むパラメータ。

パラメータ	Type	説明
groups	object/array	更新するホストグループ。 ホストグループでは、`groupid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須
hosts	object/array	指定したホストグループ上の現在のホストを置き換えるホスト。 記載されたものを除く他のすべてのホストは、ホストグループから除外されます。 ディスカバリされたホストは影響を受けません。 ホストでは、`hostid` プロパティのみを定義する必要があります。 パラメータの動作: - 必須

戻り値

(`object`) `groupids` プロパティの下で更新された host group の ID を含むオブジェクトを返します。

例

ホストグループ内のホストの置き換え

ホストグループ内のすべてのホストを、指定したホストに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massupdate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            }
        ]
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

関連項目

- hostgroup.update
- hostgroup.massadd
- ホスト

ソース

CHostGroup::massUpdate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

説明

`object hostgroup.propagate(object parameters)`

このメソッドは、ホストグループのすべてのサブグループに権限とタグフィルターを適用することができます。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロール をご覧ください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

| パラメータ |Type| 説明 |

|--|--|------|

|groups|object/array| 伝播するホストグループ。

 ホストグループでは groupid プロパティが定義されている必要があります。

パラメータの動作:
- 必須 |

|permissions|boolean| 権限を伝播するには”true” に設定します。

パラメータの動作:
- tag_filters が設定されていない場合は 必須 |

|tag_filters|boolean| タグフィルターを伝播するには”true” に設定します。

パラメータの動作:
- permissions が設定されていない場合は 必須 |

戻り値

(object) は、groupids プロパティの下で、伝播されたホストグループの ID を含むオブジェクトを返します。

例

ホストグループの権限とタグフィルターをそのサブグループに伝達します。

ホストグループの権限とタグフィルターをそのサブグループに伝達します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
```

```
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

参照

- hostgroup.update
- hostgroup.massadd
- Host

ソース

CHostGroup::propagate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

説明

`object hostgroup.update(object/array hostGroups)`

このメソッドは、既存のホストグループを更新することができます。

Note:
このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 更新するホストグループのプロパティ。

各ホストグループでは groupid プロパティを定義する必要があり、その他のプロパティはすべて任意です。指定されたプロパティのみが更新され、それ以外のプロパティは変更されません。

戻り値

(object) groupids プロパティの下で更新された host gtoup の ID を含むオブジェクトを返します。

例

ホストグループの名前の変更

ホストグループの名前を”Linux hosts” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

ソース

CHostGroup::update() in *ui/include/classes/api/services/CHostGroup.php*.

ホストプロトタイプ

このクラスはホストプロトタイプを操作するために設計されています。

オブジェクト参照:

- ホストプロトタイプ
- グループリンク
- グループプロトタイプ
- ホストプロトタイプタグ
- カスタムインターフェース
    - カスタムインターフェースの詳細

利用可能なメソッド:

- hostprototype.create - 新しいホストプロトタイプを作成
- hostprototype.delete - ホストプロトタイプを削除
- hostprototype.get - ホストプロトタイプを取得
- hostprototype.update - ホストプロトタイプを更新

Host prototype オブジェクト

次のオブジェクトは、`hostprototype`API に直接関連しています。

ホストプロトタイプ

ホストプロトタイプオブジェクトには、以下のプロパティがあります。

Property	Type	Description
hostid	ID	ホストプロトタイプの ID。 Property behavior: - 読み取り専用 - 更新操作では 必須
host	string	ホストプロトタイプの技術名。 Property behavior: - 作成操作では 必須 - 継承されたオブジェクトでは 読み取り専用
name	string	ホストプロトタイプの表示名。 デフォルト: `host` プロパティの値。 Property behavior: - 継承されたオブジェクトでは 読み取り専用

Property	Type	Description
status	integer	ホストプロトタイプのステータス。 使用可能な値: 0 - *(デフォルト)* 監視対象ホスト; 1 - 非監視ホスト。
flags	integer	ホストプロトタイプの生成元。 使用可能な値: 2 - ホストプロトタイプ; 6 - ディスカバリされたホストプロトタイプ Property behavior: - 読み取り専用
inventory_mode	integer	ホストインベントリの入力モード。 使用可能な値: -1 - *(デフォルト)* 無効; 0 - 手動; 1 - 自動。
templateid	ID	親テンプレートホストプロトタイプの ID。 Property behavior: - 読み取り専用
discover	integer	ホストプロトタイプのディスカバリステータス。 使用可能な値: 0 - *(デフォルト)* 新しいホストがディスカバリされる; 1 - 新しいホストはディスカバリされず、既存のホストは失われたものとしてマークされる。
custom_interfaces	integer	ホストプロトタイプによって作成されるホストのカスタムインターフェースのソース。 使用可能な値: 0 - *(デフォルト)* 親ホストからインターフェースを継承する; 1 - ホストプロトタイプのカスタムインターフェースを使用する。 Property behavior: - 継承されたオブジェクトでは 読み取り専用
uuid	string	インポートされたホストプロトタイプを、すでに存在するホストプロトタイプにリンクするために使用される汎用一意識別子。指定されていない場合は自動生成されます。 Property behavior: - ホストプロトタイプがテンプレートに属している場合に サポート

グループリンク

グループリンクオブジェクトは、ホストプロトタイプをホストグループに関連付けます。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
groupid	ID	ホストグループの ID。 Property behavior: *- required*

グループプロトタイプ

グループプロトタイプオブジェクトは、ディスカバリされたホストに対して作成されるグループを定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
group_prototypeid	ID	グループプロトタイプの ID。 Property behavior: - *read-only*
name	string	グループプロトタイプの名前。 Property behavior: - create 操作で *required*

ホストプロトタイプタグ

ホストプロトタイプタグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	ホストプロトタイプタグ名。 プロパティの動作: - 必須
value	string	ホストプロトタイプタグの値。

カスタムインターフェース

ホストプロトタイプオブジェクト の `custom_interfaces` が「ホストプロトタイプのカスタムインターフェースを使用」に設定されている場合、カスタムインターフェースがサポートされます。カスタムインターフェースオブジェクトには、以下のプロパティがあります。

Property	Type	Description
type	integer	インターフェースのタイプ。 指定可能な値: 1 - エージェント; 2 - SNMP; 3 - IPMI; 4 - JMX. プロパティの動作: - 必須
useip	integer	IP 経由で接続するかどうか。 指定可能な値: 0 - ホストの DNS 名を使用して接続; 1 - ホストの IP アドレスを使用して接続。 プロパティの動作: - 必須
ip	string	インターフェースで使用される IP アドレス。 マクロを含めることができます。 プロパティの動作: - `useip` が「ホストの IP アドレスを使用して接続」に設定されている場合は 必須
dns	string	インターフェースで使用される DNS 名。 マクロを含めることができます。 プロパティの動作: - `useip` が「ホストの DNS 名を使用して接続」に設定されている場合は 必須

Property	Type	Description
port	string	インターフェースで使用されるポート番号。 ユーザーマクロおよび LLD マクロを含めることができます。 プロパティの動作: - 必須
main	integer	そのインターフェースをホスト上のデフォルトとして使用するかどうか。 ホスト上では、あるタイプのインターフェースのうち 1 つだけをデフォルトとして設定できます。 指定可能な値: 0 - デフォルトではない; 1 - デフォルト。 プロパティの動作: - 必須
details	object	カスタムインターフェースの詳細 用の追加オブジェクト。 プロパティの動作: - `type` が「SNMP」に設定されている場合は 必須

カスタムインターフェースの詳細

details オブジェクトには、以下のプロパティがあります。

Property	Type	Description
version	integer	SNMP インターフェースのバージョン。 指定可能な値: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. プロパティの動作: - 必須
bulk	integer	bulk SNMP リクエストを使用するかどうか。 指定可能な値: 0 - bulk リクエストを使用しない; 1 - *(デフォルト)* - bulk リクエストを使用する。
community	string	SNMP コミュニティ。 プロパティの動作: - `version` が”SNMPv1” または”SNMPv2c” に設定されている場合は 必須
max_repetitions	integer	ネイティブ SNMP bulk リクエスト (GetBulkRequest-PDUs) の最大繰り返し値。 SNMPv2 および v3 の `discovery[]` と `walk[]` アイテムでのみ使用されます。 デフォルト: 10。
securityname	string	SNMPv3 セキュリティ名。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合に サポート

Property	Type	Description
securitylevel	integer	SNMPv3 セキュリティレベル。 指定可能な値: 0 - *(デフォルト)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. プロパティの動作: - `version` が”SNMPv3” に設定されている場合に サポート
authpassphrase	string	SNMPv3 認証パスフレーズ。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authNoPriv” または”authPriv” に設定されている場合に サポート
privpassphrase	string	SNMPv3 プライバシーパスフレーズ。 プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authPriv” に設定されている場合に サポート
authprotocol	integer	SNMPv3 認証プロトコル。 指定可能な値: 0 - *(デフォルト)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authNoPriv” または”authPriv” に設定されている場合に サポート
privprotocol	integer	SNMPv3 プライバシープロトコル。SNMPv3 インターフェースでのみ使用されます。 指定可能な値: 0 - *(デフォルト)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. プロパティの動作: - `version` が”SNMPv3” に設定され、`securitylevel` が”authPriv” に設定されている場合に サポート
contextname	string	SNMPv3 コンテキスト名。 プロパティの動作: - `version` が”SNMPv3” に設定されている場合に サポート

hostprototype.create

説明

`object hostprototype.create(object/array hostPrototypes)`

このメソッドでは、新しいホストプロトタイプを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object/array) 作成するホストプロトタイプ。

標準のホストプロトタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
groupLinks	array	ホストプロトタイプ用に作成するグループリンク。 パラメータの動作: - 必須
ruleid	ID	このホストプロトタイプが属する LLD ルールの ID。 パラメータの動作: - 必須
groupPrototypes	array	ホストプロトタイプ用に作成するグループプロトタイプ。
macros	object/array	ホストプロトタイプ用に作成するユーザーマクロ。
tags	object/array	ホストプロトタイプタグ。
interfaces	object/array	ホストプロトタイプのカスタムインターフェース。
templates	object/array	ホストプロトタイプにリンクするテンプレート。 テンプレートでは、templateid プロパティのみを定義する必要があります。

戻り値

(object) hostids プロパティの下で作成されたホストプロトタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストプロトタイプの順序と一致します。

例

ホストプロトタイプの作成

LLD ルール「23542」に、グループプロトタイプ「{#HV.NAME}」、タグの組「datacenter」: 「{#DATACENTER.NAME}」、およびコミュニティ {$SNMP_COMMUNITY} を使用するカスタム SNMPv2 インターフェース 127.0.0.1:161 を持つホストプロトタイプ「{#VM.NAME}」を作成します。これをホストグループ「2」にリンクします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.create",
    "params": {
        "host": "{#VM.NAME}",
        "ruleid": "23542",
        "custom_interfaces": "1",
        "groupLinks": [
            {
                "groupid": "2"
            }
        ],
        "groupPrototypes": [
            {
                "name": "{#HV.NAME}"
            }
        ],
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            }
```

```
        ],
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

関連項目

- グループリンク
- グループプロトタイプ
- ホストプロトタイプタグ
- カスタムインターフェース
- ユーザーマクロ

ソース

CHostPrototype::create() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

説明

`object hostprototype.delete(array hostPrototypeIds)`

このメソッドは、ホストのプロトタイプを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストのプロトタイプの ID。

戻り値

(object) hostids プロパティの下にある削除されたホストのプロトタイプの ID を含むオブジェクトを返します。

例

複数のホストのプロトタイプの削除

2 つのホストのプロトタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

ソース

CHostPrototype::delete() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

説明

`integer/array hostprototype.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってホストプロトタイプを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
hostids	ID/array	指定した ID を持つホストプロトタイプのみを返します。
discoveryids	ID/array	指定した LLD ルールに属するホストプロトタイプのみを返します。
inherited	boolean	`true` に設定すると、テンプレートから継承されたアイテムのみを返します。

パラメータ	Type	説明
selectDiscoveryData	query	ホストプロトタイプのディスカバリオブジェクトデータを含む `discoveryData` プロパティを返します。ホストプロトタイプのディスカバリオブジェクトは、検出されたホストプロトタイプを、それが検出された元のホストプロトタイプに関連付けます。 以下のプロパティを持ちます。 `host` - (`string`) ホストの ID。 `parent_hostid` - (`string`) ホストプロトタイプの作成元となったホストプロトタイプの ID。 `status` - (`int`) ホストプロトタイプのディスカバリステータス。 0 - *(default)* ホストプロトタイプは検出されています。 1 - ホストプロトタイプはすでに検出されていません。 `ts_delete` - (`timestamp`) すでに検出されていないホストプロトタイプが削除される時刻。 `ts_disable` - (`timestamp`) すでに検出されていないホストプロトタイプが無効化される時刻。 `disable_source` - (`int`) ホストプロトタイプが LLD ルールによって無効化されたか、手動で無効化されたかを示すインジケーター。 0 - *(default)* 自動的に無効化。 1 - LLD ルールによって無効化。
selectDiscoveryRule	query	ホストプロトタイプが属する LLD ルールを含む`discoveryRule` プロパティを返します。
selectDiscoveryRulePrototype	query	ホストプロトタイプが属する親 LLD ルールプロトタイプを含む`discoveryRulePrototype` プロパティを返します。
selectInterfaces	query	ホストプロトタイプのカスタムインターフェースを含む`interfaces` プロパティを返します。
selectGroupLinks	query	ホストプロトタイプのグループリンクを含む`groupLinks` プロパティを返します。
selectGroupPrototypes	query	ホストプロトタイプのグループプロトタイプを含む`groupPrototypes` プロパティを返します。
selectMacros	query	ホストプロトタイプのマクロを含む`macros` プロパティを返します。
selectParentHost	query	ホストプロトタイプが属するホストを含む`parentHost` プロパティを返します。
selectTags	query	ホストプロトタイプのタグを含む`tags` プロパティを返します。
selectTemplates	query	ホストプロトタイプにリンクされたテンプレートを含む`templates` プロパティを返します。 `count` をサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `hostid`, `host`, `name`, `status`, `discovered`。
countOutput	boolean	これらのパラメータはリファレンスコメントで説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

LLD ルールからホストプロトタイプを取得する

LLD ルールから、すべてのホストプロトタイプ、そのグループリンク、グループプロトタイプ、およびタグを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10092",
            "host": "{#HV.UUID}",
            "name": "{#HV.UUID}",
            "status": "0",
            "templateid": "0",
            "flags": "2",
            "discover": "0",
            "custom_interfaces": "1",
            "inventory_mode": "-1",
            "groupLinks": [
                {
                    "group_prototypeid": "4",
                    "hostid": "10092",
                    "groupid": "7",
                    "templateid": "0"
                }
            ],
            "groupPrototypes": [
                {
                    "group_prototypeid": "7",
                    "hostid": "10092",
                    "name": "{#CLUSTER.NAME}",
                    "templateid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "datacenter",
                    "value": "{#DATACENTER.NAME}"
                },
                {
                    "tag": "instance-type",
                    "value": "{#INSTANCE_TYPE}"
                }
            ],
            "interfaces": [
                {
                    "main": "1",
                    "type": "2",
```

```
                    "useip": "1",
                    "ip": "127.0.0.1",
                    "dns": "",
                    "port": "161",
                    "details": {
                        "version": "2",
                        "bulk": "1",
                        "community": "{$SNMP_COMMUNITY}",
                        "max_repetitions": "10"
                    }
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- グループリンク
- グループプロトタイプ
- ユーザーマクロ

ソース

CHostPrototype::get() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

説明

`object hostprototype.update(object/array hostPrototypes)`

このメソッドでは、既存のホストプロトタイプを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object/array`) 更新するホストプロトタイプのプロパティ。

各ホストプロトタイプでは `hostid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準のホストプロトタイププロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
groupLinks	array	ホストプロトタイプ上の現在のグループリンクを置き換えるためのグループリンク。
groupPrototypes	array	パラメータの動作: - 継承されたオブジェクトでは *read-only* ホストプロトタイプ上の既存のグループプロトタイプを置き換えるためのグループプロトタイプ。 リクエストに含まれていないすべてのグループプロトタイプは削除されます。
macros	object/array	パラメータの動作: - 継承されたオブジェクトでは *read-only* 現在のユーザーマクロを置き換えるためのユーザーマクロ。 リクエストに含まれていないすべてのマクロは削除されます。

パラメータ	型	説明
tags	object/array	現在のタグを置き換えるためのホストプロトタイプタグ。 リクエストに含まれていないすべてのタグは削除されます。 パラメータの動作: - 継承されたオブジェクトでは *read-only*
interfaces	object/array	現在のインターフェースを置き換えるための、ホストプロトタイプのカスタムインターフェース。 カスタムインターフェースオブジェクトには、そのすべてのパラメータを含める必要があります。 リクエストに含まれていないすべてのインターフェースは削除されます。 パラメータの動作: - Host prototype object の `custom_interfaces` が"use host prototypes custom interfaces" に設定されている場合に *supported* - 継承されたオブジェクトでは *read-only*
templates	object/array	現在リンクされているテンプレートを置き換えるためのテンプレート。 テンプレートでは `templateid` プロパティのみを定義する必要があります。

戻り値

(`object`) `hostids` プロパティの下で更新されたホストプロトタイプの ID を含むオブジェクトを返します。

例

ホストプロトタイプの無効化

ホストプロトタイプを無効化します。つまり、そのステータスを「1」に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

ホストプロトタイプのタグの更新

ホストプロトタイプのタグを新しいものに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
```

```
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

ホストプロトタイプのカスタムインターフェースの更新

継承されたインターフェースを、ホストプロトタイプのカスタムインターフェースに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "custom_interfaces": "1",
        "interfaces": [
            {
                "main": "1",
                "type": "2",
                "useip": "1",
                "ip": "127.0.0.1",
                "dns": "",
                "port": "161",
                "details": {
                    "version": "2",
                    "bulk": "1",
                    "community": "{$SNMP_COMMUNITY}"
                }
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
```

```
}
```

関連項目

- グループリンク
- グループプロトタイプ
- ホストプロトタイプタグ
- カスタムインターフェース
- ユーザーマクロ

ソース

CHostPrototype::update() in *ui/include/classes/api/services/CHostPrototype.php*.

マップ

このクラスは、マップを操作するために設計されています。

オブジェクト参照:

- マップ
- マップ要素
  - マップ要素ホスト
  - マップ要素ホストグループ
  - マップ要素マップ
  - マップ要素トリガー
  - マップ要素タグ
  - マップ要素 URL
- マップリンク
  - マップリンクトリガー
  - マップリンクインジケーター
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

利用可能なメソッド:

- map.create - 新しいマップを作成
- map.delete - マップを削除
- map.get - マップを取得
- map.update - マップを更新

Map オブジェクト

次のオブジェクトは、mapAPI に直接関連しています。

マップ

マップオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
sysmapid	ID	マップの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
height	integer	マップの高さ（ピクセル単位）。 プロパティの動作: - 作成操作で 必須
name	string	マップ名。 プロパティの動作: - 作成操作で 必須

プロパティ	型	説明
width	integer	マップの幅（ピクセル単位）。 プロパティの動作: - 作成操作で 必須
backgroundid	ID	マップの背景として使用される画像の ID。
background_scale	integer	背景画像の縦横比を維持したスケーリングを有効にするかどうか。 指定可能な値: 0 - スケーリングを無効にする; 1 - *(デフォルト)* 画像のスケーリングを有効にする。
expand_macros	integer	マップの設定時にラベル内のマクロを展開するかどうか。 指定可能な値: 0 - *(デフォルト)* マクロを展開しない; 1 - マクロを展開する。
expandproblem	integer	問題が 1 件のみの要素について、障害トリガーを表示するかどうか。 指定可能な値: 0 - 常に障害数を表示する; 1 - *(デフォルト)* 障害が 1 件のみの場合は障害トリガーを表示する。
grid_align	integer	グリッドへの整列を有効にするかどうか。 指定可能な値: 0 - グリッドへの整列を無効にする; 1 - *(デフォルト)* グリッドへの整列を有効にする。
grid_show	integer	マップ上にグリッドを表示するかどうか。 指定可能な値: 0 - グリッドを表示しない; 1 - *(デフォルト)* グリッドを表示する。
grid_size	integer	マップグリッドのサイズ（ピクセル単位）。 サポートされる値: 20、40、50、75、100。 デフォルト: 50。
highlight	integer	アイコンのハイライトを有効にするかどうか。 指定可能な値: 0 - ハイライト無効; 1 - *(デフォルト)* ハイライト有効。
iconmapid	ID	マップで使用されるアイコンマップの ID。
label_format	integer	高度なラベルを有効にするかどうか。 指定可能な値: 0 - *(デフォルト)* 高度なラベルを無効にする; 1 - 高度なラベルを有効にする。
label_location	integer	マップ要素ラベルの位置。 指定可能な値: 0 - *(デフォルト)* 下; 1 - 左; 2 - 右; 3 - 上。
label_string_host	string	ホスト要素のカスタムラベル。 プロパティの動作: - `label_type_host` が”custom” に設定されている場合は 必須
label_string_hostgroup	string	ホストグループ要素のカスタムラベル。 プロパティの動作: - `label_type_hostgroup` が”custom” に設定されている場合は 必須

プロパティ	型	説明
label_string_image	string	画像要素のカスタムラベル。 プロパティの動作: - `label_type_image` が”custom” に設定されている場合は 必須
label_string_map	string	マップ要素のカスタムラベル。 プロパティの動作: - `label_type_map` が”custom” に設定されている場合は 必須
label_string_trigger	string	トリガー要素のカスタムラベル。 プロパティの動作: - `label_type_trigger` が”custom” に設定されている場合は 必須
label_type	integer	マップ要素ラベルのタイプ。 指定可能な値: 0 - ラベル; 1 - IP アドレス; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし。
label_type_host	integer	ホスト要素のラベルタイプ。 指定可能な値: 0 - ラベル; 1 - IP アドレス; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。
label_type_hostgroup	integer	ホストグループ要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。
label_type_image	integer	ホストグループ要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 4 - なし; 5 - カスタム。
label_type_map	integer	マップ要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。
label_type_trigger	integer	トリガー要素のラベルタイプ。 指定可能な値: 0 - ラベル; 2 - *(デフォルト)* 要素名; 3 - ステータスのみ; 4 - なし; 5 - カスタム。

プロパティ	型	説明
markelements	integer	最近ステータスが変化したマップ要素をハイライトするかどうか。 指定可能な値: 0 - *(デフォルト)* 要素をハイライトしない; 1 - 要素をハイライトする。
severity_min	integer	マップに表示されるトリガーの最小深刻度。 サポートされるトリガー深刻度の一覧については、trigger `severity` プロパティを参照してください。
show_element_label	integer	要素ラベルのデフォルトの表示方法。 指定可能な値: 0 - 常に表示; 1 - *(デフォルト)* 自動非表示。
show_link_label	integer	リンクラベルのデフォルトの表示方法。 指定可能な値: 0 - 常に表示; 1 - *(デフォルト)* 自動非表示。
show_unack	integer	障害の表示方法。 指定可能な値: 0 - *(デフォルト)* すべての障害数を表示する; 1 - 未確認の障害数のみを表示する; 2 - 確認済みと未確認の障害数を別々に表示する。
userid	ID	マップの所有者であるユーザーの ID。
private	integer	マップ共有のタイプ。 指定可能な値: 0 - 公開マップ; 1 - *(デフォルト)* 非公開マップ。
show_suppressed	integer	抑制された障害を表示するかどうか。 指定可能な値: 0 - *(デフォルト)* 抑制された障害を非表示にする; 1 - 抑制された障害を表示する。

マップ要素

マップ要素オブジェクトは、マップ上に表示されるオブジェクトを定義します。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
selementid	ID	マップ要素の ID。 プロパティの動作: - *read-only*
elements	array	要素データオブジェクト。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合は *required*

Property	Type	Description
elementtype	integer	マップ要素のタイプ。 指定可能な値: 0 - ホスト; 1 - マップ; 2 - トリガー; 3 - ホストグループ; 4 - 画像。 プロパティの動作: - *required*
iconid_off	ID	デフォルト状態で要素を表示するために使用される画像の ID。 プロパティの動作: - *required*
areatype	integer	ホストグループ内の各ホストをどのように表示するか。 指定可能な値: 0 - *(default)* ホストグループ要素がマップ全体を占有する; 1 - ホストグループ要素が固定サイズになる。
elementsubtype	integer	ホストグループ要素をマップ上でどのように表示するか。 指定可能な値: 0 - *(default)* ホストグループを単一の要素として表示する; 1 - グループ内の各ホストを個別に表示する。
evaltype	integer	マップ要素のタグフィルタリング条件の評価方法。 指定可能な値: 0 - *(default)* And/Or; 2 - Or。
height	integer	固定サイズのホストグループ要素の高さ（ピクセル単位）。 デフォルト: 200。
iconid_disabled	ID	無効なマップ要素を表示するために使用される画像の ID。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合に *supported*
iconid_maintenance	ID	メンテナンス中のマップ要素を表示するために使用される画像の ID。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合に *supported*
iconid_on	ID	障害のあるマップ要素を表示するために使用される画像の ID。 プロパティの動作: - `elementtype` が”host”、”map”、”trigger”、または”host group” に設定されている場合に *supported*
label	string	要素のラベル。
label_location	integer	マップ要素ラベルの位置。 指定可能な値: -1 - *(default)* デフォルトの位置; 0 - 下; 1 - 左; 2 - 右; 3 - 上。

Property	Type	Description
permission	integer	権限レベルのタイプ。 指定可能な値: -1 - なし; 2 - 読み取り専用; 3 - 読み書き。
show_label	integer	要素ラベルの表示方法。 指定可能な値: -1 - *(default)* マップのデフォルト; 0 - 常に表示; 1 - 自動的に非表示。
sysmapid	ID	この要素が属するマップの ID。 プロパティの動作: - *read-only*
urls	array	マップ要素の URL。 マップ要素 URL オブジェクトについては、以下で詳しく説明します。
use_iconmap	integer	ホスト要素に対してアイコンマッピングを使用するかどうか。 指定可能な値: 0 - アイコンマッピングを使用しない; 1 - *(default)* アイコンマッピングを使用する。
viewtype	integer	ホストグループ要素の配置アルゴリズム。 指定可能な値: 0 - *(default)* グリッド。
width	integer	固定サイズのホストグループ要素の幅（ピクセル単位）。 デフォルト: 200。
x	integer	要素の X 座標（ピクセル単位）。 デフォルト: 0。
y	integer	要素の Y 座標（ピクセル単位）。 デフォルト: 0。
zindex	integer	マップ要素の順序付けに使用される値（z-index）。 デフォルト: 0。

マップ要素ホスト

マップ要素のホストオブジェクトは、1 つのホスト要素を定義します。

プロパティ	Type	説明
hostid	ID	ホストの ID。

マップ要素ホストグループ

マップ要素のホストグループオブジェクトは、1 つのホストグループ要素を定義します。

プロパティ	Type	説明
groupid	ID	ホストグループの ID。

マップ要素 Map

マップ要素の Map オブジェクトは、1 つのマップ要素を定義します。

プロパティ	Type	説明
sysmapid	ID	マップの ID。

マップ要素トリガー

マップ要素のトリガーオブジェクトは、1 つ以上のトリガー要素を定義します。

プロパティ	Type	説明
triggerid	ID	トリガーの ID。

マップ要素タグ

マップ要素のタグオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	マップ要素のタグ名。 プロパティの動作: - 必須
operator	integer	マップ要素のタグ条件の演算子。 使用可能な値: 0 - *(デフォルト)* 含む; 1 - 等しい; 2 - 含まない; 3 - 等しくない; 4 - 存在する; 5 - 存在しない。
value	string	マップ要素のタグ値。

マップ要素 URL

マップ要素 URL オブジェクトは、特定のマップ要素で利用可能なクリック可能なリンクを定義します。以下のプロパティがあります。

プロパティ	型	説明
sysmapelementurlid	ID	マップ要素 URL の ID。 プロパティの動作: - 読み取り専用
name	string	リンクのキャプション。 プロパティの動作: - 必須
url	string	リンク URL。 プロパティの動作: - 必須
selementid	ID	この URL が属するマップ要素の ID。

マップリンク

マップリンクオブジェクトは、2 つのマップ要素間のリンクを定義します。以下のプロパティがあります。

プロパティ	型	説明
linkid	ID	マップリンクの ID。 プロパティの動作: - 読み取り専用
sysmapid	ID	リンクが属するマップの ID。

プロパティ	型	説明
selementid1	ID	一方の端でリンクされる最初のマップ要素の ID。 プロパティの動作: - 必須
selementid2	ID	もう一方の端でリンクされる最初のマップ要素の ID。 プロパティの動作: - 必須
drawtype	integer	リンク線の描画スタイル。 指定可能な値: 0 - *(デフォルト)* 実線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	16 進数カラーコードで表した線の色。 デフォルト: 000000。
label	string	リンクラベル。
show_label	integer	リンクラベルの表示方法。 指定可能な値: -1 - *(デフォルト)* マップのデフォルト; 0 - 常に表示; 1 - 自動的に非表示。
indicator_type	integer	リンクインジケーターのタイプを選択します。 指定可能な値: 0 - *(デフォルト)* 静的リンク; 1 - トリガー; 2 - アイテムの値。
linktriggers	array	リンクステータスインジケーターとして使用するマップリンクトリガー。 マップリンクトリガーオブジェクトの詳細は以下で説明しています。 プロパティの動作: - `indicator_type` が”trigger” (1) に設定されている場合は 必須。
itemid	ID	アイテムの ID。 プロパティの動作: - `indicator_type` が”item value” (2) に設定されている場合は 必須。
highlights	array	リンクステータスインジケーターとして使用するマップリンクハイライト。 マップリンクインジケーターオブジェクトの詳細は以下で説明しています。 プロパティの動作: - `indicator_type` が”item value” (2) に設定されている場合は 必須。
thresholds	array	リンクステータスインジケーターとして使用するマップリンクしきい値。 マップリンクインジケーターオブジェクトの詳細は以下で説明しています。 プロパティの動作: - `indicator_type` が”item value” (2) に設定されている場合は 必須。

プロパティ	型	説明
permission	integer	権限レベルのタイプ。 指定可能な値: -1 - なし; 2 - 読み取り専用; 3 - 読み書き。

マップリンクトリガー

マップリンクトリガーオブジェクトは、トリガーの状態に基づくマップリンクのステータスインジケーターを定義します。次のプロパティがあります。

プロパティ	型	説明
triggerid	ID	リンクインジケーターとして使用されるトリガーの ID。 プロパティの動作: - 必須
color	string	16 進数のカラーコードによるインジケーターの色。 デフォルト: DD0000。
drawtype	integer	インジケーターの描画スタイル。 使用可能な値: 0 - *(デフォルト)* 線; 2 - 太線; 3 - 点線; 4 - 破線。

マップリンクインジケーター

マップリンクインジケーターオブジェクトは、アイテムの値に基づくマップリンクのステータスインジケーターを定義します。このオブジェクトには、次のプロパティがあります。

プロパティ	Type	説明
pattern	string	比較対象の正規表現です。ハイライトでのみ使用できます。
sortorder	integer	ハイライトのソート順を定義するために使用されます。ハイライトでのみ使用できます。
threshold	string	比較対象の数値です。しきい値でのみ使用できます。
drawtype	integer	インジケーターの描画スタイルです。 使用可能な値: 0 - *(デフォルト)* 実線; 2 - 太線; 3 - 点線; 4 - 破線。
color	string	16 進数のカラーコードで指定するインジケーターの色です。 デフォルト: DD0000。 プロパティの動作: - 必須

マップ URL

マップ URL オブジェクトは、マップ上の特定のタイプのすべての要素で利用可能な、クリック可能なリンクを定義します。以下のプロパティがあります。

プロパティ	Type	説明
sysmapurlid	ID	マップ URL の ID。 Property behavior: - *read-only*
name	string	リンクのキャプション。 Property behavior: - *required*
url	string	リンク URL。 Property behavior: - *required*
elementtype	integer	URL を利用可能にするマップ要素のタイプ。 サポートされているタイプの一覧については、マップ要素の `type` プロパティを参照してください。 デフォルト: 0。
sysmapid	ID	この URL が属するマップの ID。

マップユーザー

ユーザーに基づくマップ権限の一覧です。以下のプロパティがあります。

プロパティ	型	説明
sysmapuserid	ID	マップユーザーの ID。 プロパティの動作: - 読み取り専用
userid	ID	ユーザーの ID。 プロパティの動作: - 必須
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き可能。 プロパティの動作: - 必須

マップユーザーグループ

ユーザーグループに基づくマップ権限の一覧です。以下のプロパティがあります。

プロパティ	型	説明
sysmapusrgrpid	ID	マップユーザーグループの ID。 プロパティの動作: - 読み取り専用
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 必須

プロパティ	型	説明
permission	integer	権限レベルの種類。 指定可能な値: 2 - 読み取り専用; 3 - 読み書き。 プロパティの動作: - 必須

マップ図形

マップ図形オブジェクトは、マップ上に表示される幾何学図形（テキストありまたはなし）を定義します。このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
sysmap_shapeid	ID	マップ図形要素の ID。 プロパティの動作: - 読み取り専用
type	integer	マップ図形要素のタイプ。 使用可能な値: 0 - 長方形; 1 - 楕円。 新しい図形を作成する場合、このプロパティは必須です。 プロパティの動作: - 必須
x	integer	図形の X 座標（ピクセル単位）。 デフォルト: 0。
y	integer	図形の Y 座標（ピクセル単位）。 デフォルト: 0。
width	integer	図形の幅（ピクセル単位）。 デフォルト: 200。
height	integer	図形の高さ（ピクセル単位）。 デフォルト: 200。
text	string	図形のテキスト。
font	integer	図形内のテキストのフォント。 使用可能な値: 0 - Georgia, serif 1 - “Palatino Linotype”, “Book Antiqua”, Palatino, serif 2 - “Times New Roman”, Times, serif 3 - Arial, Helvetica, sans-serif 4 - “Arial Black”, Gadget, sans-serif 5 - “Comic Sans MS”, cursive, sans-serif 6 - Impact, Charcoal, sans-serif 7 - “Lucida Sans Unicode”, “Lucida Grande”, sans-serif 8 - Tahoma, Geneva, sans-serif 9 - “Trebuchet MS”, Helvetica, sans-serif 10 - Verdana, Geneva, sans-serif 11 - “Courier New”, Courier, monospace 12 - “Lucida Console”, Monaco, monospace デフォルト: 9。

Property	Type	Description
font_size	integer	フォントサイズ（ピクセル単位）。 デフォルト: 11。
font_color	string	フォントの色。 デフォルト: 000000。
text_halign	integer	テキストの水平方向の配置。 使用可能な値: 0 - 中央; 1 - 左; 2 - 右。 デフォルト: 0。
text_valign	integer	テキストの垂直方向の配置。 使用可能な値: 0 - 中央; 1 - 上; 2 - 下。 デフォルト: 0。
border_type	integer	枠線のタイプ。 使用可能な値: 0 - なし; 1 - ──────; 2 - ┅; 3 - – – –。 デフォルト: 0。
border_width	integer	枠線の幅（ピクセル単位）。 デフォルト: 0。
border_color	string	枠線の色。 デフォルト: 000000。
background_color	string	背景色（塗りつぶし色）。 デフォルト:（空）。
zindex	integer	すべての図形および線の順序付けに使用される値（z-index）。 デフォルト: 0。

マップの線

マップの線オブジェクトは、マップ上に表示される線を定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
sysmap_shapeid	ID	マップ図形要素の ID。 プロパティの動作: - 読み取り専用
x1	integer	線の点 1 の X 座標（ピクセル単位）。 デフォルト: 0。
y1	integer	線の点 1 の Y 座標（ピクセル単位）。 デフォルト: 0。

プロパティ	型	説明
x2	integer	線の点 2 の X 座標（ピクセル単位）。 デフォルト: 200。
y2	integer	線の点 2 の Y 座標（ピクセル単位）。 デフォルト: 200。
line_type	integer	線の種類。 使用可能な値: 0 - なし; 1 - ————; 2 - ┅; 3 - – – –. デフォルト: 0。
line_width	integer	線の幅（ピクセル単位）。 デフォルト: 0。
line_color	string	線の色。 デフォルト: 000000。
zindex	integer	すべての図形および線の順序付けに使用される値（z-index）。 デフォルト: 0。

map.create

説明

`object map.create(object/array maps)`

このメソッドでは、新しいマップを作成できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するマップの対応付けです。

このメソッドは、標準のマッププロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
links	array	マップ上に作成するマップリンク。
selements	array	マップ上に作成するマップ要素。
urls	array	マップ上に作成するマップ URL。
users	array	マップ上に作成するマップユーザー共有。
userGroups	array	マップ上に作成するマップユーザーグループ共有。
shapes	array	マップ上に作成するマップ図形。
lines	array	マップ上に作成するマップ線。

> **Note:**
> マップリンクを作成するには、マップ要素の `selementid` に任意の値を設定し、その後この値を使用して、リンクの `selementid1` または `selementid2` プロパティでこの要素を参照する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。例を参照してください。

戻り値

(`object`) `sysmapids` プロパティの下で作成されたマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたマップの順序と一致します。

例

空のマップを作成する

要素のないマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

ホストマップの作成

2 つのホスト要素と、それらの間のリンクを持つマップを作成します。マップリンクオブジェクトでマップ要素を参照するために、一時的な”selementid1” および”selementid2” の値が使用されていることに注意してください。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "selementid": "1",
                "elements": [
                    {"hostid": "1033"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            },

            {
                "selementid": "2",
                "elements": [
                    {"hostid": "1037"}
                ],
                "elementtype": 0,
                "iconid_off": "2"
            }
        ],
        "links": [
            {
                "selementid1": "1",
                "selementid2": "2"
```

```
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

トリガーマップの作成

2 つのトリガーを含むトリガー要素を持つマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Trigger map",
        "width": 600,
        "height": 600,
        "selements": [
            {
                "elements": [
                    {"triggerid": "12345"},
                    {"triggerid": "67890"}
                ],
                "elementtype": 2,
                "iconid_off": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

マップの共有

2 種類の共有（ユーザーおよびユーザーグループ）を持つマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": " マップの共有",
```

```
        "width": 600,
        "height": 600,
        "users": [
            {
                "userid": "4",
                "permission": "3"
            }
        ],
        "userGroups": [
            {
                "usrgrpid": "7",
                "permission": "2"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

マップ図形

マップ名のタイトルを持つマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Host map",
        "width": 600,
        "height": 600,
        "shapes": [
            {
                "type": 0,
                "x": 0,
                "y": 0,
                "width": 600,
                "height": 11,
                "text": "{MAP.NAME}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
```

```
    "id": 1
}
```

マップの線

マップの線を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map API lines",
        "width": 500,
        "height": 500,
        "lines": [
            {
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

関連項目

- マップ要素
- マップリンク
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

ソース

CMap::create() in *ui/include/classes/api/services/CMap.php*.

map.delete

説明

`object map.delete(array mapIds)`

このメソッドは、マップを削除することができます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するマップの ID。

戻り値

(object) sysmapids プロパティの下にある削除されたマップの ID を含むオブジェクトを返します。

例

複数のマップの削除

2 つのマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

ソース

CMap::delete() in *ui/include/classes/api/services/CMap.php*.

map.get

説明

`integer/array map.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってマップを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロールを参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
sysmapids	ID/array	指定した ID のマップのみを返します。
userids	ID/array	指定したユーザー ID に属するマップのみを返します。

パラメータ	型	説明
expandUrls	flag	対応するマップ要素にグローバルマップ URL を追加し、すべてのマップ要素 URL 内のマクロを展開します。
selectIconMap	query	マップで使用されているアイコンマップを含む`iconmap` プロパティを返します。
selectLinks	query	要素間のマップリンクを含む`links` プロパティを返します。
selectSelements	query	マップ要素を含む`selements` プロパティを返します。
selectUrls	query	マップ URL を含む`urls` プロパティを返します。
selectUsers	query	マップが共有されているユーザーを含む`users` プロパティを返します。
selectUserGroups	query	マップが共有されているユーザーグループを含む`userGroups` プロパティを返します。
selectShapes	query	マップ図形を含む`shapes` プロパティを返します。
selectLines	query	マップ線を含む`lines` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `name`, `width`, `height`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

マップを取得

マップ「3」に関するすべてのデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.get",
    "params": {
        "output": "extend",
        "selectSelements": "extend",
        "selectLinks": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "selectShapes": "extend",
        "selectLines": "extend",
        "sysmapids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix server",
                    "label_location": "3",
                    "x": "11",
                    "y": "141",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "-1",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": [
                        {
                            "tag": "service",
                            "value": "mysqld",
                            "operator": "0"
                        }
                    ]
                },
                {
                    "selementid": "11",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Web server",
                    "label_location": "3",
                    "x": "211",
                    "y": "191",
                    "iconid_disabled": "0",
                    "iconid_maintenance": "0",
                    "elementsubtype": "0",
                    "areatype": "0",
                    "width": "200",
                    "height": "200",
                    "viewtype": "0",
                    "use_iconmap": "1",
                    "show_label": "0",
                    "zindex":"0",
                    "elements": [],
                    "urls": [],
                    "tags": []
                },
                {
                    "selementid": "12",
                    "sysmapid": "3",
                    "elementtype": "0",
```

```
                "evaltype": "0",
                "iconid_off": "185",
                "iconid_on": "0",
                "label": "{HOST.NAME}\r\n{HOST.CONN}",
                "label_location": "0",
                "x": "111",
                "y": "61",
                "iconid_disabled": "0",
                "iconid_maintenance": "0",
                "elementsubtype": "0",
                "areatype": "0",
                "width": "200",
                "height": "200",
                "viewtype": "0",
                "use_iconmap": "0",
                "show_label": "1",
                "zindex":"0",
                "elements": [
                    {
                        "hostid": "10084"
                    }
                ],
                "urls": [],
                "tags": []
            }
        ],
        "links": [
            {
                "linkid": "23",
                "sysmapid": "3",
                "selementid1": "10",
                "selementid2": "11",
                "drawtype": "0",
                "color": "00CC00",
                "label": "",
                "show_label": "1",
                "indicator_type": "0",
                "itemid": "0",
                "linktriggers": [],
                "thresholds": [],
                "highlights": []
            }
        ],
        "users": [
            {
                "sysmapuserid": "1",
                "userid": "2",
                "permission": "2"
            }
        ],
        "userGroups": [
            {
                "sysmapusrgrpid": "1",
                "usrgrpid": "7",
                "permission": "2"
            }
        ],
        "shapes":[
            {
                "sysmap_shapeid":"1",
                "type":"0",
                "x":"0",
```

```
                "y":"0",
                "width":"680",
                "height":"15",
                "text":"{MAP.NAME}",
                "font":"9",
                "font_size":"11",
                "font_color":"000000",
                "text_halign":"0",
                "text_valign":"0",
                "border_type":"0",
                "border_width":"0",
                "border_color":"000000",
                "background_color":"",
                "zindex":"0"
            }
        ],
        "lines":[
            {
                "sysmap_shapeid":"2",
                "x1": 30,
                "y1": 10,
                "x2": 100,
                "y2": 50,
                "line_type": 1,
                "line_width": 10,
                "line_color": "009900",
                "zindex":"1"
            }
        ],
        "sysmapid": "3",
        "name": "Local network",
        "width": "400",
        "height": "400",
        "backgroundid": "0",
        "background_scale": "1",
        "label_type": "2",
        "label_location": "3",
        "show_element_label": "0",
        "show_link_label": "1",
        "highlight": "1",
        "expandproblem": "1",
        "markelements": "0",
        "show_unack": "0",
        "grid_size": "50",
        "grid_show": "1",
        "grid_align": "1",
        "label_format": "0",
        "label_type_host": "2",
        "label_type_hostgroup": "2",
        "label_type_trigger": "2",
        "label_type_map": "2",
        "label_type_image": "2",
        "label_string_host": "",
        "label_string_hostgroup": "",
        "label_string_trigger": "",
        "label_string_map": "",
        "label_string_image": "",
        "iconmapid": "0",
        "expand_macros": "0",
        "severity_min": "0",
        "userid": "1",
        "private": "1",
```

```
            "show_suppressed": "1"
        }
    ],
    "id": 1
}
```

関連項目

- アイコンマップ
- マップ要素
- マップリンク
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

ソース

CMap::get() in *ui/include/classes/api/services/CMap.php*.

map.update

説明

`object map.update(object/array maps)`

このメソッドでは、既存のマップを更新できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object/array) 更新するマップのプロパティ。

各マップでは mapid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のマッププロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
links	array	既存のリンクを置き換えるマップリンク。
selements	array	既存の要素を置き換えるマップ要素。
urls	array	既存の URL を置き換えるマップ URL。
users	array	既存の要素を置き換えるマップユーザー の共有設定。
userGroups	array	既存の要素を置き換えるマップユーザーグループ の共有設定。
shapes	array	既存の図形を置き換えるマップ図形。
lines	array	既存の線を置き換えるマップ線。

> **Note:**
> 新しいマップ要素間にマップリンクを作成するには、要素の selementid に任意の値を設定し、その値を使用してリンクの selementid1 または selementid2 プロパティでこの要素を参照する必要があります。要素が作成されると、この値は Zabbix によって生成された正しい ID に置き換えられます。map.create の例を参照してください。

戻り値

(object) sysmapids プロパティの下で更新されたマップの ID を含むオブジェクトを返します。

例

マップのサイズ変更

マップのサイズを 1200x1200 ピクセルに変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

マップの所有者を変更

管理者および特権管理者のみ利用できます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

関連項目

- マップ要素
- マップリンク
- マップ URL
- マップユーザー
- マップユーザーグループ
- マップ図形
- マップ線

ソース

CMap::update() in *ui/include/classes/api/services/CMap.php*.

メディアタイプ

このクラスはメディアタイプを操作するために設計されています。

オブジェクト参照:

- メディアタイプ
  - Webhook パラメータ
  - スクリプトパラメータ
- メッセージテンプレート

利用可能なメソッド:

- mediatype.create - 新しいメディアタイプを作成
- mediatype.delete - メディアタイプを削除
- mediatype.get - メディアタイプを取得
- mediatype.update - メディアタイプを更新

Media type オブジェクト

次のオブジェクトは、`mediatype`API に直接関連しています。

メディアタイプ

メディアタイプオブジェクトには、以下のプロパティがあります。

Property	Type	Description
mediatypeid	ID	メディアタイプの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
name	string	メディアタイプの名前。 プロパティの動作: - 作成操作では 必須
type	integer	メディアタイプで使用される転送方式。 指定可能な値: 0 - Email; 1 - Script; 2 - SMS; 4 - Webhook.
exec_path	string	プロパティの動作: - 作成操作では 必須 AlertScriptsPath サーバー設定パラメータで指定されたディレクトリに配置されているスクリプトファイル名（例: notification.sh）。
gsm_modem	string	プロパティの動作: - `type` が”Script” に設定されている場合は 必須 GSM モデムのシリアルデバイス名。
passwd	string	プロパティの動作: - `type` が”SMS” に設定されている場合は 必須 認証パスワード。
		プロパティの動作: - `smtp_authentication` が”Normal password” に設定されている場合に サポート

Property	Type	Description
provider	integer	メールプロバイダー。 指定可能な値: 0 - *(デフォルト)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.
smtp_email	string	通知の送信元となるメールアドレス。 プロパティの動作: - `type` が”Email” に設定されている場合は 必須
smtp_helo	string	SMTP HELO。 プロパティの動作: - `type` が”Email” に設定されている場合に サポート
smtp_server	string	SMTP サーバー。 プロパティの動作: - `type` が”Email” に設定されている場合は 必須
smtp_port	integer	接続先の SMTP サーバーポート。 デフォルト: 25。 プロパティの動作: - `type` が”Email” に設定されている場合に サポート
smtp_security	integer	使用する SMTP 接続のセキュリティレベル。 指定可能な値: 0 - *(デフォルト)* None; 1 - STARTTLS; 2 - SSL/TLS. プロパティの動作: - `type` が”Email” に設定されている場合に サポート
smtp_verify_host	integer	SMTP 用の SSL ホスト検証。 指定可能な値: 0 - *(デフォルト)* No; 1 - Yes. プロパティの動作: - `smtp_security` が”STARTTLS” または”SSL/TLS” に設定されている場合に サポート
smtp_verify_peer	integer	SMTP 用の SSL ピア検証。 指定可能な値: 0 - *(デフォルト)* No; 1 - Yes. プロパティの動作: - `smtp_security` が”STARTTLS” または”SSL/TLS” に設定されている場合に サポート

Property	Type	Description
smtp_authentication	integer	使用する SMTP 認証方式。 指定可能な値: 0 - *(デフォルト)* None; 1 - Normal password; 2 - OAuth token. *Office365 relay* メールプロバイダーでは OAuth 認証は使用できません。 プロパティの動作: - `type` が”Email” に設定されている場合に サポート
redirection_url	string	OAuth 認可を戻すための Zabbix Web インターフェース URL。 デフォルト: API settings プロパティ `url` の値に `zabbix.php?action=oauth.authorize` の部分を付加したもの プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須
client_id	string	OAuth 認可サーバー内に登録されたクライアント識別子。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須
client_secret	string	OAuth 認可サーバー内に登録されたクライアントシークレット。Super Admin タイプのユーザーのみアクセスできます。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須
authorization_url	string	アクセストークンおよびリフレッシュトークンを取得するための、パラメータ付き OAuth URL。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須
token_url	string	認可トークンをアクセストークンおよびリフレッシュトークンに交換するための OAuth URL。 この URL は、無効なアクセストークンを更新するためにサーバーでも使用されます。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須
tokens_status	integer	トークンの状態を表すビットマスク。 指定可能な値: 0 - (default) 両方のトークンに無効な値が含まれる 1 - アクセストークンに有効な値が含まれる 2 - リフレッシュトークンに有効な値が含まれる 3 - 両方のトークンに有効な値が含まれる。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合に サポート
access_token	string	OAuth アクセストークンの値。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須

Property	Type	Description
access_token_updated	timestamp	`refresh_token` による更新時にサーバーが、またはトークン変更時に API が `access_token` を最後に変更した時刻のタイムスタンプ。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合に サポート
access_expires_in	integer	`access_token` が期限切れとなり、`refresh_url` へのリクエストが必要になるまでの秒数。 Zabbix サーバーが `access_token` を更新した際、または API がトークン変更時に設定します。 タイムスタンプは `access_token_updated` の値を加算して計算されます。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合に サポート
refresh_token	string	OAuth リフレッシュトークンの値。 プロパティの動作: - `smtp_authentication` が”OAuth token” に設定されている場合は 必須
status	integer	メディアタイプが有効かどうか。 指定可能な値: 0 - *(デフォルト)* Enabled; 1 - Disabled.
username	string	ユーザー名。 プロパティの動作: - `smtp_authentication` が”Normal password” に設定されている場合に サポート
maxsessions	integer	並列処理できるアラートの最大数。 `type` が”SMS” に設定されている場合の指定可能な値: 1。 `type` が”Email”、”Script”、または”Webhook” に設定されている場合の指定可能な値: 0-100。 デフォルト: 1。
maxattempts	integer	アラート送信の最大試行回数。 指定可能な値: 1-100。 デフォルト: 3。
attempt_interval	string	再試行の間隔。 秒、およびサフィックス付きの時間単位を受け付けます。 指定可能な値: 0-1h。 デフォルト: 10s。
message_format	integer	メッセージ形式。 指定可能な値: 0 - Plain text; 1 - *(デフォルト)* HTML. プロパティの動作: - `type` が”Email” に設定されている場合に サポート

Property	Type	Description
script	text	webhook スクリプト本文（JavaScript）。 プロパティの動作: - `type` が”Webhook” に設定されている場合は 必須
timeout	string	webhook スクリプトのタイムアウト。 秒、およびサフィックス付きの時間単位を受け付けます。 指定可能な値: 1-60s。 デフォルト: 30s。
process_tags	integer	プロパティの動作: - `type` が”Webhook” に設定されている場合に サポート Webhook スクリプト応答内の JSON プロパティ値をタグとして処理します。これらのタグは既存の障害タグに追加されます。 指定可能な値: 0 - *(デフォルト)* webhook スクリプト応答を無視; 1 - webhook スクリプト応答をタグとして処理。
show_event_menu	integer	プロパティの動作: - `type` が”Webhook” に設定されている場合に サポート カスタム URL にリンクするエントリをイベントメニュー に含めます。 また、problem.get およびevent.get の出力に `urls` プロパティも追加します。 指定可能な値: 0 - *(デフォルト)* イベントメニューエントリおよび `urls` プロパティを含めない; 1 - イベントメニューエントリおよび `urls` プロパティを含める。
event_menu_url	string	プロパティの動作: - `type` が”Webhook” に設定されている場合に サポート イベントメニュー のエントリ、およびproblem.get とevent.get が返す `urls` プロパティで使用される URL。
event_menu_name	string	プロパティの動作: - `show_event_menu` が”Include event menu entry and `urls` property” に設定されている場合に サポート イベントメニュー のエントリ、およびproblem.get とevent.get が返す `urls` プロパティで使用される名前。
parameters	array	プロパティの動作: - `show_event_menu` が”Include event menu entry and `urls` property” に設定されている場合に サポート Webhook またはスクリプト のパラメータ。
description	text	プロパティの動作: - `type` が”Webhook” または”Script” に設定されている場合に サポート メディアタイプの説明。

webhook パラメータ

webhook パラメータには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	パラメータ名。 プロパティの動作: - 必須

プロパティ	Type	説明
value	string	パラメータ値。マクロをサポートします。 サポートされるマクロについては、サポートされるマクロページで説明されています。

スクリプトパラメータ

スクリプトパラメータには、以下のプロパティがあります。

プロパティ	Type	説明
sortorder	integer	パラメータ値がコマンドライン引数としてスクリプトに渡される順序です。最初の引数は 0 から始まります。 プロパティの動作: - 必須
value	string	パラメータ値です。マクロをサポートします。 サポートされるマクロについては、サポートされるマクロ ページを参照してください。

メッセージテンプレート

メッセージテンプレートオブジェクトは、通知を送信するためのアクションの実行時にデフォルトメッセージとして使用されるテンプレートを定義します。次のプロパティがあります。

プロパティ	Type	説明
eventsource	integer	イベントソース。 使用可能な値: 0 - トリガー; 1 - ディスカバリ; 2 - 自動登録; 3 - 内部; 4 - サービス。 プロパティの動作: - 必須
recovery	integer	実行モード。 使用可能な値: 0 - 実行; 1 - 復旧時の実行; 2 - 更新時の実行。 プロパティの動作: - 必須
subject	string	メッセージの件名。
message	string	メッセージ本文。

mediatype.create

説明

`object mediatype.create(object/array mediaTypes)`

このメソッドでは、新しいメディアタイプを作成できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) 作成するメディアタイプ。

標準のメディアタイプのプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
message_templates	array	メディアタイプ用に作成するメッセージテンプレート。

戻り値

(object) mediatypeids プロパティの下に作成されたメディアタイプの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメディアタイプの順序と一致します。

例

メールメディアタイプの作成

カスタム SMTP ポートとメッセージテンプレートを使用して、新しいメールメディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "0",
        "name": "Email",
        "smtp_server": "mail.example.com",
        "smtp_helo": "example.com",
        "smtp_email": "zabbix@example.com",
        "smtp_port": "587",
        "message_format": "1",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "Problem: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" started at {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Resolved in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" on host \"{HOST.NAME}\" has been resolved at {EVENT.R
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Updated problem in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem \"{EVENT.NAME}\" on host \"{HOST
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
```

```
    "id": 1
}
```

スクリプトメディアタイプの作成

試行回数と試行間隔にカスタム値を指定して、新しいスクリプトメディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push notifications",
        "exec_path": "push-notification.sh",
        "maxattempts": "5",
        "attempt_interval": "11s",
        "parameters": [
            {
                "sortorder": "0",
                "value": "{ALERT.SENDTO}"
            },
            {
                "sortorder": "1",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "sortorder": "2",
                "value": "{ALERT.MESSAGE}"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

webhook メディアタイプの作成

新しい webhook メディアタイプを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "4",
        "name": "Webhook",
        "script": "var Webhook = {\r\n    token: null,\r\n    to: null,\r\n    subject: null,\r\n    messa
        "parameters": [
            {
                "name": "Message",
                "value": "{ALERT.MESSAGE}"
            },
            {
```

```
                "name": "Subject",
                "value": "{ALERT.SUBJECT}"
            },
            {
                "name": "To",
                "value": "{ALERT.SENDTO}"
            },
            {
                "name": "Token",
                "value": "<Token>"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::create() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

説明

`object mediatype.delete(array mediaTypeIds)`

このメソッドは、メディアタイプを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するメディアタイプの ID。

戻り値

(object) mediatypeids プロパティの下で削除されたメディアタイプの ID を含むオブジェクトを返します。

例

複数のメディアタイプの削除

2 つのメディアタイプを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::delete() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

説明

`integer/array mediatype.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってメディアタイプを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

Note:
メディアタイプのユーザー関連情報を要求する場合、*Admin* および *User* タイプのユーザーは、自分自身のユーザーに関するデータのみ取得できます。例については、Retrieving media types as *Admin* を参照してください。

パラメータ	Type	説明
mediatypeids	ID/array	指定した ID を持つメディアタイプのみを返します。
mediaids	ID/array	指定したmediaで使用されているメディアタイプのみを返します。
userids	ID/array	指定したユーザーによって使用されているメディアタイプのみを返します。
selectActions	query	メディアタイプを使用しているアクションを含む`actions` プロパティを返します。
selectMessageTemplates	query	メディアタイプメッセージの配列を含む`message_templates` プロパティを返します。 Parameter behavior: - *Super admin* タイプのユーザーで *supported*
selectUsers	query	メディアタイプを使用しているユーザーを含む`users` プロパティを返します。 ユーザータイプに基づく制限については`user.get` を参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `mediatypeid`。

パラメータ	Type	説明
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が照合対象の単一の値または値の配列であるオブジェクトを受け付けます。 `text` data type のプロパティはサポートしていません。 *Admin* および *User* タイプのユーザーで指定可能なMedia type object プロパティ: `mediatypeid`, `name`, `type`, `status`, `maxattempts`。
output	query	返されるMedia type object のプロパティです。 *Admin* および *User* タイプのユーザーは、次のプロパティのみ取得できます: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`。例については、Retrieving media types as *Admin* を参照してください。 デフォルト: `extend`。
search	object	指定したパターンに一致する結果を返します（大文字・小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索を実行します。 `string` および `text` data type のプロパティのみをサポートします。 *Admin* および *User* タイプのユーザーで指定可能なMedia type object プロパティ: `name`, `description`。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

メディアタイプの取得

設定済みのすべてのメディアタイプを取得します。次の例では、2 つのメディアタイプを返します。

- メールメディアタイプ
- SMS メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "smtp_server": "mail.example.com",
            "smtp_helo": "example.com",
            "smtp_email": "zabbix@example.com",
            "exec_path": "",
            "gsm_modem": "",
            "username": "",
            "passwd": "",
            "status": "0",
            "smtp_port": "25",
            "smtp_security": "0",
            "smtp_verify_peer": "0",
            "smtp_verify_host": "0",
            "smtp_authentication": "0",
            "maxsessions": "1",
            "maxattempts": "3",
            "attempt_interval": "10s",
            "message_format": "0",
            "script": "",
            "timeout": "30s",
            "process_tags": "0",
            "show_event_menu": "1",
            "event_menu_url": "",
            "event_menu_name": "",
            "description": "",
            "provider": "0",
            "message_templates": [
                {
                    "eventsource": "0",
                    "recovery": "0",
                    "subject": "Problem: {EVENT.NAME}",
                    "message": "Problem started at {EVENT.TIME} on {EVENT.DATE}\r\nProblem name: {EVENT.NA
                },
                {
                    "eventsource": "0",
                    "recovery": "1",
                    "subject": "Resolved: {EVENT.NAME}",
                    "message": "Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE
                },
                {
                    "eventsource": "0",
                    "recovery": "2",
                    "subject": "Updated problem: {EVENT.NAME}",
                    "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
                },
                {
                    "eventsource": "1",
                    "recovery": "0",
                    "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                    "message": "Discovery rule: {DISCOVERY.RULE.NAME}\r\n\r\nDevice IP: {DISCOVERY.DEVICE.
                },
                {
                    "eventsource": "2",
                    "recovery": "0",
                    "subject": "Autoregistration: {HOST.HOST}",
```

```
                "message": "Host name: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.PORT}"
            }
        ],
        "parameters": []
    },
    {
        "mediatypeid": "3",
        "type": "2",
        "name": "SMS",
        "smtp_server": "",
        "smtp_helo": "",
        "smtp_email": "",
        "exec_path": "",
        "gsm_modem": "/dev/ttyS0",
        "username": "",
        "passwd": "",
        "status": "0",
        "smtp_port": "25",
        "smtp_security": "0",
        "smtp_verify_peer": "0",
        "smtp_verify_host": "0",
        "smtp_authentication": "0",
        "maxsessions": "1",
        "maxattempts": "3",
        "attempt_interval": "10s",
        "message_format": "1",
        "script": "",
        "timeout": "30s",
        "process_tags": "0",
        "show_event_menu": "1",
        "event_menu_url": "",
        "event_menu_name": "",
        "description": "",
        "provider": "0",
        "message_templates": [
            {
                "eventsource": "0",
                "recovery": "0",
                "subject": "",
                "message": "{EVENT.SEVERITY}: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "",
                "message": "RESOLVED: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVEN
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "",
                "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
            },
            {
                "eventsource": "2",
                "recovery": "0",
```

```
                    "subject": "",
                    "message": "Autoregistration: {HOST.HOST}\r\nHost IP: {HOST.IP}\r\nAgent port: {HOST.P
                }
            ],
            "parameters": []
        }
    ],
    "id": 1
}
```

Admin としてメディアタイプを取得

Admin タイプのユーザーとして、有効になっているすべてのメディアタイプと、それらのメディアタイプを使用するユーザーを取得します。次の例では、2 つのメディアタイプが返されます。

- ユーザーが 1 人いるメールメディアタイプ（*Admin* タイプのユーザー自身のユーザーのみ）
- ユーザーがいない SMS メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "1",
            "type": "0",
            "name": "Email",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": [
                {
                    "userid": "3",
                    "username": "database-admin",
                    "name": "John",
                    "surname": "Doe",
                    "url": "",
                    "autologin": "0",
                    "autologout": "0",
                    "lang": "default",
                    "refresh": "30s",
                    "theme": "default",
                    "attempt_failed": "0",
                    "attempt_ip": "",
                    "attempt_clock": "0",
                    "rows_per_page": "50",
                    "timezone": "default",
                    "roleid": "2",
                    "provisioned": "0"
                }
            ]
```

```
        },
        {
            "mediatypeid": "3",
            "type": "2",
            "name": "SMS",
            "status": "0",
            "description": "",
            "maxattempts": "3",
            "users": []
        }
    ],
    "id": 1
}
```

スクリプトと Webhook のメディアタイプ取得

次の例では、3 つのメディアタイプが返されます。

- パラメータ付きのスクリプトメディアタイプ
- パラメータなしのスクリプトメディアタイプ
- パラメータ付きの Webhook メディアタイプ

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Script with parameters",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix alert"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Script without parameters",
```

```
            "parameters": []
        },
        {
            "mediatypeid": "11",
            "name": "Webhook",
            "parameters": [
                {
                    "name": "alert_message",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "name": "event_update_message",
                    "value": "{EVENT.UPDATE.MESSAGE}"
                },
                {
                    "name": "host_name",
                    "value": "{HOST.NAME}"
                },
                {
                    "name": "trigger_description",
                    "value": "{TRIGGER.DESCRIPTION}"
                },
                {
                    "name": "trigger_id",
                    "value": "{TRIGGER.ID}"
                },
                {
                    "name": "alert_source",
                    "value": "Zabbix"
                }
            ]
        }
    ],
    "id": 1
}
```


参照

- User

ソース

CMediaType::get() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

説明

`object mediatype.update(object/array mediaTypes)`

このメソッドでは、既存のメディアタイプを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) 更新するメディアタイプのプロパティ。

各メディアタイプに対して `mediatypeid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のメディアタイププロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
message_templates	array	現在のメッセージテンプレートを置き換えるメッセージテンプレート。

戻り値

(object) mediatypeids プロパティの下で更新されたメディアタイプの ID を含むオブジェクトを返します。

例

メディアタイプの有効化

メディアタイプを有効化します。つまり、ステータスを"0" に設定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

ソース

CMediaType::update() in *ui/include/classes/api/services/CMediaType.php*.

メンテナンス

このクラスはメンテナンスを扱うために設計されています。

オブジェクト参照:

- メンテナンス
- 期間
- 障害タグ

利用可能なメソッド:

- maintenance.create - 新しいメンテナンスを作成
- maintenance.delete - メンテナンスを削除
- maintenance.get - メンテナンスを取得
- maintenance.update - メンテナンスを更新

Maintenance オブジェクト

次のオブジェクトは、maintenanceAPI に直接関連しています。

メンテナンス

メンテナンスオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
maintenanceid	ID	メンテナンスの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
name	string	メンテナンスの名前。 プロパティの動作: - 作成操作で 必須
active_since	timestamp	メンテナンスが有効になる時刻（この時刻を含む）。 指定された値は分単位に切り捨てられます。 プロパティの動作: - 作成操作で 必須
active_till	timestamp	メンテナンスが有効でなくなる時刻（この時刻を含まない）。 指定された値は分単位に切り捨てられます。 プロパティの動作: - 作成操作で 必須
description	string	メンテナンスの説明。
maintenance_type	integer	メンテナンスのタイプ。 使用可能な値: 0 - *(デフォルト)* データ収集あり; 1 - データ収集なし。
tags_evaltype	integer	障害タグの評価方法。 使用可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。

時間帯

時間帯オブジェクトは、メンテナンスを有効にする期間を定義するために使用されます。以下のプロパティがあります。

プロパティ	型	説明
period	integer	メンテナンス期間の長さ（秒単位）。 指定した値は分単位に切り捨てられます。 指定可能な値の範囲: 300 - 86399940。 デフォルト: 3600。
timeperiod_type	integer	時間帯のタイプ。 指定可能な値: 0 - *(デフォルト)* 1 回のみ; 2 - 毎日; 3 - 毎週; 4 - 毎月。
start_date	timestamp	メンテナンス期間を有効にする日時。 指定した値は分単位に切り捨てられます。 デフォルト: 現在の日付。 プロパティの動作: - `timeperiod_type` が「1 回のみ」に設定されている場合に サポート

プロパティ	型	説明
start_time	integer	メンテナンス開始時刻の時刻部分（秒単位）。 指定した値は分単位に切り捨てられます。 デフォルト: 0。 プロパティの動作: - timeperiod_type が「毎日」、「毎週」、または「毎月」に設定されている場合に サポート
every	integer	毎日および毎週の期間では、every はメンテナンスを有効にする日または週の間隔を定義します。 timeperiod_type が「毎日」または「毎週」に設定されている場合のデフォルト値: 1。 毎月の期間で day が設定されている場合、every プロパティはメンテナンスを有効にする月の日を定義します。 timeperiod_type が「毎月」に設定され、day が設定されている場合のデフォルト値: 1。 毎月の期間で dayofweek が設定されている場合、every プロパティはメンテナンスを有効にする月の週を定義します。 timeperiod_type が「毎月」に設定され、dayofweek が設定されている場合の指定可能な値: 1 - *(デフォルト)* 第 1 週; 2 - 第 2 週; 3 - 第 3 週; 4 - 第 4 週; 5 - 最終週。 プロパティの動作: - timeperiod_type が「毎日」、「毎週」、または「毎月」に設定されている場合に サポート
dayofweek	integer	メンテナンスを有効にする曜日。 指定可能なビットマップ値: 1 - 月曜日; 2 - 火曜日; 4 - 水曜日; 8 - 木曜日; 16 - 金曜日; 32 - 土曜日; 64 - 日曜日。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、月曜日・水曜日・金曜日の場合は 21）。 プロパティの動作: - timeperiod_type が「毎週」に設定されている場合、または timeperiod_type が「毎月」に設定されていて day が設定されていない場合に 必須
day	integer	メンテナンスを有効にする月の日。 プロパティの動作: - timeperiod_type が「毎月」に設定されていて dayofweek が設定されていない場合に 必須

プロパティ	型	説明
month	integer	メンテナンスを有効にする月。 指定可能なビットマップ値: 1 - 1 月; 2 - 2 月; 4 - 3 月; 8 - 4 月; 16 - 5 月; 32 - 6 月; 64 - 7 月; 128 - 8 月; 256 - 9 月; 512 - 10 月; 1024 - 11 月; 2048 - 12 月。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、1 月・4 月・7 月・10 月の場合は 585）。 プロパティの動作: - `timeperiod_type` が「毎月」に設定されている場合に 必須

障害タグ

障害タグオブジェクトは、メンテナンスが有効になった際にどの障害を抑制する必要があるかを定義するために使用されます。
タグは、メンテナンスオブジェクト の `maintenance_type` が「データ収集あり」に設定されている場合にのみ指定できます。
このオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
tag	string	障害タグ名。 プロパティの動作: - 必須
operator	integer	条件の演算子。 使用可能な値: 0 - 等しい; 2 - (デフォルト) 含む。
value	string	障害タグの値。

maintenance.create

説明

`object maintenance.create(object/array maintenances)`

このメソッドでは、新しいメンテナンスを作成できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

`(object/array)` 作成するメンテナンス。

標準のメンテナンスプロパティに加えて、このメソッドは以下のパラメータを受け付けます。

Parameter	Type	Description
groups	object/array	メンテナンス対象となるホストグループ。 ホストグループでは、groupid プロパティのみを定義する必要があります。 パラメータの動作: - hosts が設定されていない場合は 必須
hosts	object/array	メンテナンス対象となるホスト。 ホストでは、hostid プロパティのみを定義する必要があります。 パラメータの動作: - groups が設定されていない場合は 必須
timeperiods	object/array	メンテナンスの期間。 パラメータの動作: - 必須
tags	object/array	障害タグ。 どの障害を抑制するかを定義します。 タグが指定されていない場合、アクティブなメンテナンス中のホストのすべての障害が抑制されます。 パラメータの動作: - メンテナンスオブジェクトの maintenance_type が「データ収集あり」に設定されている場合に サポート

戻り値

(object) maintenanceids プロパティの下で作成されたメンテナンスの ID を含むオブジェクトを返します。返される ID の順序は、渡されたメンテナンスの順序と一致します。

例

メンテナンスの作成

ID "2" のホストグループに対し、データ収集を伴うメンテナンスを作成します。問題タグ **service:mysqld** および **error** を対象とします。メンテナンスは 17.03.2026 から 17.03.2027 まで有効で、毎週日曜日の 18:00 に開始し、1 時間継続します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "2"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
```

```
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

関連項目

- 期間

ソース

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

説明

`object maintenance.delete(array maintenanceIds)`

このメソッドは、メンテナンス期間を削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除するメンテナンス期間の ID。

戻り値

(`object`) `maintenanceids` プロパティの下にある削除されたメンテナンス期間の ID を含むオブジェクトを返します。

例

複数のメンテナンス期間の削除

2 つのメンテナンス期間を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

ソース

CMaintenance::delete() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

説明

`integer/array maintenance.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってメンテナンスを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
groupids	ID/array	指定したホストグループに割り当てられているメンテナンスのみを返します。
hostids	ID/array	指定したホストに割り当てられているメンテナンスのみを返します。
maintenanceids	ID/array	指定した ID のメンテナンスのみを返します。
selectHostGroups	query	メンテナンスに割り当てられているホストグループを含む`hostgroups` プロパティを返します。
selectHosts	query	メンテナンスに割り当てられているホストを含む`hosts` プロパティを返します。
selectTags	query	メンテナンスの障害タグを含む`tags` プロパティを返します。
selectTimeperiods	query	メンテナンスの期間を含む`timeperiods` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

パラメータ	Type	説明
startSearch	boolean	

戻り値

(integer/array) 以下のいずれかを返します。

- オブジェクトの配列
- countOutput パラメータが使用されている場合は、取得したオブジェクト数

例

メンテナンスの取得

設定済みのすべてのメンテナンスと、割り当てられているホストグループ、定義された期間、および障害タグに関するデータを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
            "tags": [
                {
```

```
                    "tag": "service",
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```


関連項目

- ホスト
- ホストグループ
- 期間

ソース

ui/include/classes/api/services/CMaintenance.php 内の CMaintenance::get()。

maintenance.update

説明

`object maintenance.update(object/array maintenances)`

このメソッドでは、既存のメンテナンスを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object/array`) 更新するメンテナンスのプロパティ。

各メンテナンスでは `maintenanceid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準のメンテナンスプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
groups	object/array	現在のグループを置き換えるホストグループ。 ホストグループでは `groupid` プロパティのみを定義する必要があります。 パラメータの動作: - `hosts` が設定されていない場合は 必須
hosts	object/array	現在のホストを置き換えるホスト。 ホストでは `hostid` プロパティのみを定義する必要があります。 パラメータの動作: - `groups` が設定されていない場合は 必須
timeperiods	object/array	現在の期間を置き換えるメンテナンスの期間。

パラメータ	型	説明
tags	object/array	現在のタグを置き換える障害タグ。 パラメータの動作: - メンテナンスオブジェクト の maintenance_type が”with data collection” に設定されている場合に サポート

戻り値

(object) maintenanceids プロパティの下で更新されたメンテナンスの ID を含むオブジェクトを返します。

例

異なるホストの割り当て

現在メンテナンスに割り当てられているホストを、別の 2 つのホストに置き換えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

関連項目

- 期間

ソース

CMaintenance::update() in *ui/include/classes/api/services/CMaintenance.php*.

ユーザーグループ

このクラスはユーザーグループを操作するために設計されています。

オブジェクト参照:

- ユーザーグループ
- 権限
- タグベースの権限

利用可能なメソッド:

- usergroup.create - 新しいユーザーグループを作成
- usergroup.delete - ユーザーグループを削除
- usergroup.get - ユーザーグループを取得
- usergroup.update - ユーザーグループを更新

User group オブジェクト

次のオブジェクトは、`usergroup`API に直接関連しています。

ユーザーグループ

ユーザーグループオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
usrgrpid	ID	ユーザーグループの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須
name	string	ユーザーグループ名。 プロパティの動作: - 作成操作では 必須
debug_mode	integer	デバッグモードが有効か無効か。 指定可能な値: 0 - *(デフォルト)* 無効; 1 - 有効。
gui_access	integer	グループ内のユーザーの Web インターフェース認証方式。 指定可能な値: 0 - *(デフォルト)* システムのデフォルト認証方式を使用; 1 - 内部認証を使用; 2 - LDAP 認証を使用; 3 - Web インターフェースへのアクセスを無効化。
mfa_status	integer	グループ内のユーザーに対して MFA が有効か無効か。 指定可能な値: 0 - 無効（設定済みのすべての MFA 方式に対して）; 1 - 有効（設定済みのすべての MFA 方式に対して）。
mfaid	ID	グループ内のユーザーに使用されるMFA 方式。 プロパティの動作: - *Super admin* タイプのユーザーで サポート - Authentication object の `mfa_status` が”Enabled” に設定されている場合に サポート
users_status	integer	ユーザーグループが有効か無効か。 deprovisioned ユーザーについては、ユーザーグループを有効にできません。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
userdirectoryid	ID	認証に使用されるユーザーディレクトリの ID。 プロパティの動作: - *Super admin* タイプのユーザーで サポート - `gui_access` が”use the system default authentication method” または”use LDAP authentication” に設定されている場合に サポート

権限

permission オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
id	ID	権限を追加するホストグループまたはテンプレートグループの ID。 プロパティの動作: - 必須
permission	integer	ホストグループまたはテンプレートグループに対するアクセスレベル。 指定可能な値: 0 - アクセス拒否; 2 - 読み取り専用アクセス; 3 - 読み書きアクセス。 プロパティの動作: - 作成操作では 必須

タグベースの権限

タグベースの権限オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
groupid	ID	権限を追加するホストグループの ID。 プロパティの動作: - 必須
tag	string	タグ名。
value	string	タグの値。

usergroup.create

説明

`object usergroup.create(object/array userGroups)`

このメソッドでは、新しいユーザーグループを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するユーザーグループ。

このメソッドは、標準のユーザーグループプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
hostgroup_rights	object/array	ユーザーグループに割り当てるホストグループの権限。
templategroup_rights	object/array	ユーザーグループに割り当てるテンプレートグループの権限。
tag_filters	array	ユーザーグループに割り当てるタグベースの権限。
users	object/array	ユーザーグループに追加するユーザー。 ユーザーには `userid` プロパティのみが定義されている必要があります。

戻り値

(`object`) `usrgrpids` プロパティの下に、作成されたユーザーグループの ID を含むオブジェクトを返します。
返される ID の順序は、渡されたユーザーグループの順序と一致します。

例

ユーザーグループの作成

ホストグループ「2」へのアクセスを拒否するユーザーグループ *Operation managers* を作成し、そこにユーザーを追加します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

参照

- Permission

ソース

ui/include/classes/api/services/CUserGroup.php 内の CUserGroup::create()。

usergroup.delete

説明

`object usergroup.delete(array userGroupIds)`

このメソッドは、ユーザーグループを削除することができます。

Attention:
プロビジョニング解除されたユーザーグループ (認証の `disabled_usrgrpid` に指定されたユーザーグループ) は削除できません。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(`array`) 削除するユーザーグループの ID。

戻り値

(`object`) `usrgrpids` プロパティの下にある削除されたユーザーグループの ID を含むオブジェクトを返します。

例

複数のユーザーグループの削除

2 つのユーザーグループを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

ソース

CUserGroup::delete() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

説明

`integer/array usergroup.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってユーザーグループを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
mfaids	ID/array	指定した MFA 方式を持つユーザーグループのみを返します。 パラメータの動作: - *Super admin* タイプのユーザーで *supported*
mfa_status	integer	指定した MFA ステータスを持つユーザーグループのみを返します。 サポートされているステータスの一覧は、ユーザーグループページを参照してください。
status	integer	指定したステータスを持つユーザーグループのみを返します。 サポートされているステータスの一覧は、ユーザーグループページを参照してください。
userids	ID/array	指定したユーザーを含むユーザーグループのみを返します。
usrgrpids	ID/array	指定した ID を持つユーザーグループのみを返します。
selectTagFilters	query	ユーザーグループのタグベースの権限を`tag_filters` プロパティで返します。

パラメータ	型	説明
selectUsers	query	ユーザーグループのユーザーを`users` プロパティで返します。 ユーザータイプに基づく制限については、`user.get`を参照してください。
selectHostGroupRights	query	ユーザーグループのホストグループ権限を `hostgroup_rights` プロパティで返します。 ホストグループへのアクセスレベルの一覧は、ユーザーグループページを参照してください。
selectTemplateGroupRights	query	ユーザーグループのテンプレートグループ権限を `templategroup_rights` プロパティで返します。 テンプレートグループへのアクセスレベルの一覧は、ユーザーグループページを参照してください。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。
output	query	返されるユーザーグループオブジェクトのプロパティ。 *Admin* および *User* タイプのユーザーは、`usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status` の各プロパティのみ取得できます。 デフォルト: `extend`。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `usrgrpid`, `name`。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

有効なユーザーグループの取得

有効なすべてのユーザーグループを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix administrators",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Guests",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Enabled debug mode",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "No access to the Web インターフェース",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Read only",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Deny",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
```

```
        }
    ],
    "id": 1
}
```

参照

- User

ソース

CUserGroup::get() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

説明

`object usergroup.update(object/array userGroups)`

このメソッドでは、既存のユーザーグループを更新できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(object/array) 更新するユーザーグループのプロパティ。

各ユーザーグループでは usrgrpid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

このメソッドは、標準のユーザーグループプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
hostgroup_rights	object/array	ユーザーグループに現在割り当てられている権限を置き換える、ホストグループの権限。
templategroup_rights	object/array	ユーザーグループに現在割り当てられている権限を置き換える、テンプレートグループの権限。
tag_filters	array	ユーザーグループに現在割り当てられている権限を置き換える、タグベースの権限。
users	object/array	ユーザーグループに現在割り当てられているユーザーを置き換える、ユーザー。 ユーザーでは userid プロパティのみを定義する必要があります。

戻り値

(object) usrgrpids プロパティの下で更新されたユーザーグループの ID を含むオブジェクトを返します。

例

ユーザーグループの有効化と権限の更新

ユーザーグループを有効化し、ホストグループ「2」と「4」に対する読み書き権限を付与します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
```

```
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

参照

- Permission

ソース

CUserGroup::update() in *ui/include/classes/api/services/CUserGroup.php*.

ユーザーディレクトリ

このクラスは、ユーザーディレクトリを操作するために設計されています。

オブジェクト参照:

- ユーザーディレクトリ
  - メディアタイプのマッピング
  - プロビジョニンググループのマッピング

利用可能なメソッド:

- userdirectory.create - 新しいユーザーディレクトリを作成
- userdirectory.delete - ユーザーディレクトリを削除
- userdirectory.get - ユーザーディレクトリを取得
- userdirectory.update - ユーザーディレクトリを更新
- userdirectory.test - ユーザーディレクトリ接続をテスト

User directory オブジェクト

次のオブジェクトは、`userdirectory`API に直接関連しています。

ユーザーディレクトリ

ユーザーディレクトリオブジェクトには、以下のプロパティがあります。

Property	Type	Description
userdirectoryid	ID	ユーザーディレクトリの ID。 ユーザーディレクトリが削除されると、削除されたユーザーディレクトリにリンクされているすべてのユーザーについて、User object のプロパティ `userdirectoryid` の値”0” に設定されます。 Property behavior: - *read-only* - 更新操作では *required*
idp_type	integer	ユーザーディレクトリに対して ID プロバイダーが使用する認証プロトコルの種類。 SAML 型のユーザーディレクトリは 1 つだけ存在できます。 指定可能な値: 1 - LDAP 型のユーザーディレクトリ; 2 - SAML 型のユーザーディレクトリ。
group_name	string	Property behavior: - 作成操作では *required* LDAP/SAML ユーザーディレクトリ属性で、LDAP/SAML ユーザーディレクトリと Zabbix の間でグループをマッピングするために使用されるグループ名を含むもの。 例: *cn*
user_username	string	Property behavior: - `provision_status` が”Enabled” に設定され、かつAuthentication object の `saml_jit_status` が”Enabled for configured SAML IdPs” に設定されている場合は *required* LDAP/SAML ユーザーディレクトリ属性（`scim_status` が”SCIM provisioning is enabled” に設定されている場合は SCIM 属性でも可）で、ユーザーの名前を含み、ユーザーがプロビジョニングされる際にUser object のプロパティ `name` の値として使用されるもの。 例: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	LDAP/SAML ユーザーディレクトリ属性（`scim_status` が”SCIM provisioning is enabled” に設定されている場合は SCIM 属性でも可）で、ユーザーの姓を含み、ユーザーがプロビジョニングされる際にUser object のプロパティ `surname` の値として使用されるもの。 例: *sn*, *surname*, *lastName*
provision_status	integer	ユーザーディレクトリのプロビジョニング状態。 指定可能な値: 0 - *(default)* Disabled（このユーザーディレクトリによって作成されたユーザーのプロビジョニングは無効）; 1 - Enabled（このユーザーディレクトリによって作成されたユーザーのプロビジョニングは有効。さらに、LDAP または SAML のプロビジョニング状態（Authentication object の `ldap_jit_status` または `saml_jit_status`）も有効である必要があります）。
provision_groups	array	LDAP/SAML ユーザーグループパターンを Zabbix ユーザーグループおよびユーザーロールにマッピングするための、provisioning groups mappings オブジェクトの配列。 Property behavior: - `provision_status` が”Enabled” に設定されている場合は *required*
provision_media	array	ユーザーの LDAP/SAML メディア属性（例: email）を、通知送信用の Zabbix ユーザーメディアにマッピングするためのmedia type mappings オブジェクトの配列。
LDAP-specific properties:		

Property	Type	Description
name	string	ユーザーディレクトリの一意な名前。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *required*
host	string	LDAP サーバーのホスト名、IP、または URI。 URI にはスキーマ（`ldap://` または `ldaps://`）、ホスト、およびポート（任意）を含める必要があります。 例: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *required*
port	integer	LDAP サーバーのポート。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *required*
base_dn	string	LDAP ユーザーディレクトリにおけるユーザーアカウントへのベースパス。 例: *ou=Users,dc=example,dc=org* *ou=Users,ou=system*（OpenLDAP の場合） *DC=company,DC=com*（Microsoft Active Directory の場合） *uid=%{user},dc=example,dc=com*（ユーザーへの直接バインドの場合。プレースホルダー”*%{user}*” は必須） Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *required*
search_attribute	string	ログイン要求で提供された情報からユーザーアカウントを識別するために使用する LDAP ユーザーディレクトリ属性。 例: *uid*（OpenLDAP の場合） *sAMAccountName*（Microsoft Active Directory の場合） Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *required*
bind_dn	string	LDAP サーバーに対してバインドおよび検索を行うための LDAP サーバーアカウント。 ユーザーへの直接バインドおよび匿名バインドでは、`bind_dn` は空でなければなりません。 例: *uid=ldap_search,ou=system*（OpenLDAP の場合） *CN=ldap_search,OU=user_group,DC=company,DC=com*（Microsoft Active Directory の場合） *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*

Property	Type	Description
bind_password	string	LDAP サーバーに対してバインドおよび検索を行うアカウントの LDAP パスワード。 ユーザーへの直接バインドおよび匿名バインドでは、`bind_password` は空でなければなりません。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
description	string	ユーザーディレクトリの説明。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
group_basedn	string	グループへの LDAP ユーザーディレクトリベースパス。LDAP ユーザーディレクトリでユーザーメンバーシップチェックを設定するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 例: *ou=Groups,dc=example,dc=com* Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
group_filter	string	ユーザーがメンバーである LDAP ユーザーディレクトリグループを取得するためのフィルター文字列。LDAP ユーザーディレクトリでユーザーメンバーシップチェックを設定するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 サポートされる `group_filter` プレースホルダー: *%{attr}* - 検索属性（`search_attribute` プロパティ値に置き換えられます）; *%{groupattr}* - グループ属性（`group_member` プロパティ値に置き換えられます）; *%{host}* - LDAP サーバーのホスト名、IP、または URI（`host` プロパティ値に置き換えられます）; *%{user}* - Zabbix ユーザー名。 デフォルト: *(%{groupattr}=%{user})* 例: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* は、LDAP グループオブジェクトに値”*uid=User1,ou=Users,dc=example,dc=com*” を持つ”*member*” 属性が含まれている場合に”User1” に一致し、”User1” がメンバーであるグループを返します; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* は、LDAP グループオブジェクトに `group_member` プロパティで指定された属性が含まれ、その値が”*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*” である場合に”User1” に一致し、”User1” がメンバーであるグループを返します。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*

Property	Type	Description
group_member	string	グループメンバーに関する情報を含む LDAP ユーザーディレクトリ属性。LDAP ユーザーディレクトリでユーザーメンバーシップチェックを設定するために使用されます。 `group_membership` が設定されている場合、ユーザーのプロビジョニング時には無視されます。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
group_membership	string	ユーザーが所属するグループに関する情報を含む LDAP ユーザーディレクトリ属性。 例: *memberOf* Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
search_filter	string	ログイン要求で提供された情報に基づいて、LDAP ユーザーディレクトリ内のユーザーを検索および認証するために使用するカスタムフィルター文字列。 サポートされる `search_filter` プレースホルダー: *%{attr}* - 検索属性名（例: *uid*, *sAMAccountName*）; *%{user}* - Zabbix ユーザー名。 デフォルト: *(%{attr}=%{user})* Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
start_tls	integer	LDAP サーバーとの通信を Transport Layer Security (TLS) を使用して保護できるようにする LDAP サーバー設定オプション。 `ldaps://` プロトコルを使用するホストでは、`start_tls` を”Disabled” に設定する必要があります。 指定可能な値: 0 - *(default)* Disabled; 1 - Enabled。 Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
user_ref_attr	string	ユーザーオブジェクトを参照するために使用される LDAP ユーザーディレクトリ属性。`user_ref_attr` の値は、ユーザーディレクトリ内の指定属性から値を取得し、それを `group_filter` 文字列内の *%{ref}* プレースホルダーの代わりに配置するために使用されます。 例: *cn*, *uid*, *member*, *uniqueMember* Property behavior: - `idp_type` が”User directory of type LDAP” に設定されている場合は *supported*
SAML-specific properties:		

Property	Type	Description
idp_entityid	string	ID プロバイダーを識別する URI で、SAML メッセージ内で ID プロバイダーと通信するために使用されます。 例: *https://idp.example.com/idp* Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *required*
sp_entityid	string	ID プロバイダーのサービスプロバイダーを識別する URL または任意の文字列。 例: *https://idp.example.com/sp* *zabbix* Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *required*
username_attribute	string	SAML ユーザーディレクトリ属性（`scim_status` が”SCIM provisioning is enabled” に設定されている場合は SCIM 属性でも可）で、認証時にUser object のプロパティ `username` の値と比較されるユーザー名を含むもの。 例: *uid*, *userprincipalname*, *samaccountname*, *username*, *userusername*, *urn:oid:0.9.2342.19200300.100.1.1*, *urn:oid:1.3.6.1.4.1.5923.1.1.1.13*, *urn:oid:0.9.2342.19200300.100.1.44* Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *required*
sso_url	string	ID プロバイダーの SAML シングルサインオンサービスの URL。Zabbix はこの URL に SAML 認証要求を送信します。 例: *http://idp.example.com/idp/sso/saml* Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *required*
slo_url	string	ID プロバイダーの SAML シングルログアウトサービスの URL。Zabbix はこの URL に SAML ログアウト要求を送信します。 例: *https://idp.example.com/idp/slo/saml* Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*
encrypt_nameid	integer	SAML name ID を暗号化するかどうか。 指定可能な値: 0 - *(default)* name ID を暗号化しない; 1 - name ID を暗号化する。 Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*

Property	Type	Description
encrypt_assertions	integer	SAML アサーションを暗号化するかどうか。 指定可能な値: 0 - *(default)* アサーションを暗号化しない; 1 - アサーションを暗号化する。 Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*
nameid_format	string	SAML ID プロバイダーのサービスプロバイダーにおける Name ID 形式。 例: *urn:oasis:names:tc:SAML:2.0:nameid-format:persistent* *urn:oasis:names:tc:SAML:2.0:nameid-format:transient* *urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos* *urn:oasis:names:tc:SAML:2.0:nameid-format:entity* Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*
scim_status	integer	SAML の SCIM プロビジョニングを有効にするか無効にするか。 指定可能な値: 0 - *(default)* SCIM プロビジョニングは無効; 1 - SCIM プロビジョニングは有効。 Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*
sign_assertions	integer	SAML アサーションに SAML 署名を付与するかどうか。 指定可能な値: 0 - *(default)* アサーションに署名しない; 1 - アサーションに署名する。 Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*
sign_authn_requests	integer	SAML AuthN 要求に SAML 署名を付与するかどうか。 指定可能な値: 0 - *(default)* AuthN 要求に署名しない; 1 - AuthN 要求に署名する。 Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*
sign_messages	integer	SAML メッセージに SAML 署名を付与するかどうか。 指定可能な値: 0 - *(default)* メッセージに署名しない; 1 - メッセージに署名する。 Property behavior: - `idp_type` が”User directory of type SAML” に設定されている場合は *supported*

Property	Type	Description
sign_logout_requests	integer	SAML ログアウト要求に SAML 署名を付与するかどうか。 指定可能な値: 0 - *(default)* ログアウト要求に署名しない; 1 - ログアウト要求に署名する。 Property behavior: - `idp_type` が"User directory of type SAML" に設定されている場合は *supported*
sign_logout_responses	integer	SAML ログアウト応答に SAML 署名を付与するかどうか。 指定可能な値: 0 - *(default)* ログアウト応答に署名しない; 1 - ログアウト応答に署名する。 Property behavior: - `idp_type` が"User directory of type SAML" に設定されている場合は *supported*

メディアタイプのマッピング

メディアタイプのマッピングオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
userdirectory_mediaid	ID	メディアタイプのマッピング ID。 プロパティの動作: - 読み取り専用
name	string	メディアタイプのマッピング一覧に表示される名前。 プロパティの動作: - 必須
mediatypeid	ID	作成されるメディアタイプの ID。Media object のプロパティ `mediatypeid` の値として使用されます。 プロパティの動作: - 必須
attribute	string	ユーザーのメディア（例: *user@example.com*）を含む LDAP/SAML ユーザーディレクトリ属性（`scim_status` が"SCIM provisioning is enabled" に設定されている場合は SCIM 属性でも可）。この値はMedia object のプロパティ `sendto` の値として使用されます。 LDAP/SAML アイデンティティプロバイダーから受信したデータ内にこの属性が存在し、かつ値が空でない場合、プロビジョニングされたユーザーのメディア作成がトリガーされます。 プロパティの動作: - 必須
active	integer	プロビジョニングされたユーザーに対してメディアが作成される際の、ユーザーメディア `active` プロパティの値。 指定可能な値: 0 - *(デフォルト)* 有効; 1 - 無効。
severity	integer	プロビジョニングされたユーザーに対してメディアが作成される際の、ユーザーメディア `severity` プロパティの値。 デフォルト: 63。
period	string	プロビジョニングされたユーザーに対してメディアが作成される際の、ユーザーメディア `period` プロパティの値。 デフォルト: 1-7,00:00-24:00。

プロビジョニンググループのマッピング

プロビジョニンググループのマッピングには、以下のプロパティがあります。

プロパティ	型	説明
name	string	LDAP/SAML ユーザーディレクトリ内のグループのフルネーム（例: *Zabbix administrators*）です（`scim_status` が”SCIM provisioning is enabled” に設定されている場合は SCIM も含みます）。 ワイルドカード文字”*” をサポートします。 すべてのプロビジョニンググループのマッピング内で一意である必要があります。 プロパティの動作: - 必須
roleid	ID	ユーザーに割り当てるユーザーロールの ID です。 複数のプロビジョニンググループのマッピングが一致した場合、最も高いユーザータイプ（*User*、*Admin*、または *Super admin*）のロールがユーザーに割り当てられます。同じユーザータイプのロールが複数ある場合は、最初のロール（アルファベット順でソート）がユーザーに割り当てられます。 プロパティの動作: - 必須
user_groups	array	Zabbix ユーザーグループ ID オブジェクトの配列です。各オブジェクトには以下のプロパティがあります: `usrgrpid` - （ID）ユーザーに割り当てる Zabbix ユーザーグループの ID。 複数のプロビジョニンググループのマッピングが一致した場合、一致したすべてのマッピングの Zabbix ユーザーグループがユーザーに割り当てられます。 プロパティの動作: - 必須

userdirectory.create

説明

`object userdirectory.create(object/array userDirectory)`

このメソッドは、新しいユーザーディレクトリを作成します。

> **Note:**
> このメソッドは、*Super admin* のユーザーのみ利用可能です。

パラメータ

`(object/array)` 作成するユーザーディレクトリ。

このメソッドは、標準ユーザーディレクトリプロパティを持つユーザーディレクトリを受け入れます。

戻り値

`(object)` は、`userdirectoryids` プロパティに作成されたユーザーディレクトリの ID を含むオブジェクトを返します。返される ID の順序は、渡されたユーザーディレクトリの順序と一致します。

例

ユーザーディレクトリの作成

StartTLS over LDAP を使用してユーザーを認証するためのユーザーディレクトリを作成します。LDAP 経由でユーザーを認証するには、LDAP 認証 を有効にする必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
        "idp_type": "1",
```

```
        "name": "LDAP API server #1",
        "host": "ldap://local.ldap",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "ldapsecretpassword",
        "search_attribute": "uid",
        "start_tls": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

ユーザーディレクトリの作成（JIT プロビジョニング有効）

LDAP 経由でユーザーを認証するためのユーザーディレクトリを作成します。（JIT プロビジョニング有効）LDAP 経由でユーザーを認証するには、LDAP 認証 が有効になっている必要があります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.create",
    "params": {
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
```

```
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
    "id": 1
}
```


Response:


```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```


ソース

CUserDirectory::create() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

説明

`object userdirectory.delete(array userDirectoryIds)`

このメソッドを使用すると、ユーザーディレクトリを削除できます。ユーザーディレクトリが 1 つでもユーザーグループに直接使用されている場合は、削除できません。
`authentication.ldap_configured` が 1 に設定されている場合、またはユーザーディレクトリがまだ残っている場合、デフォルトの LDAP ユーザーディレクトリは削除できません。

> **Note:**
> このメソッドが使用できるのは、*Super admin* ユーザータイプのみです。

パラメータ

(`array`) 削除するユーザーディレクトリの ID。

戻り値

(`object`) `userdirectoryids` プロパティの下にある削除されたユーザーディレクトリの ID を含むオブジェクトを返します。

例

複数のユーザーディレクトリの削除

2 つのユーザーディレクトリを削除します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

ソース

CUserDirectory::delete() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

説明

`integer/array userdirectory.get(object parameters)`

このメソッドは、指定したパラメーターに応じたユーザーディレクトリを取得できます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。

パラメータ

(`object`) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
userdirectoryids	ID/array	指定した ID を持つユーザーディレクトリのみを返します。
selectUsrgrps	query	ユーザーディレクトリに関連付けられたユーザーグループ を含む `usrgrps` プロパティを返します。 `count` をサポートします。
selectProvisionMedia	query	ユーザーディレクトリに関連付けられたメディアタイプマッピング を含む `provision_media` プロパティを返します。
selectProvisionGroups	query	ユーザーディレクトリに関連付けられたプロビジョニンググループマッピング を含む `provision_groups` プロパティを返します。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `name`。
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が単一の値または値の配列であるオブジェクトを受け付けます。 サポートされるプロパティ: `userdirectoryid`, `idp_type`, `provision_status`。

パラメータ	型	説明
search	object	指定したパターンに一致する結果を返します（大文字と小文字を区別しません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索を実行します。 サポートされるプロパティ: name, description。 SAML タイプのユーザーディレクトリでは、name と description の両方のプロパティは空の値になります。両方のプロパティは userdirectory.update 操作で変更できます。
countOutput	boolean	これらのパラメータはreference commentary で説明されています。
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ユーザーディレクトリの取得

各ユーザーディレクトリに関連付けられたメディアタイプマッピングとプロビジョニンググループマッピングを表示する追加プロパティを含むすべてのユーザーディレクトリを取得します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
       {
            "userdirectoryid": "1",
            "idp_type": "2",
            "name": "",
            "provision_status": "1",
            "description": "",
            "group_name": "groups",
            "user_username": "",
            "user_lastname": "",
            "idp_entityid": "http://example.com/simplesaml/saml2/idp/metadata.php",
            "sso_url": "http://example.com/simplesaml/saml2/idp/SSOService.php",
```

```
            "slo_url": "",
            "username_attribute": "uid",
            "sp_entityid": "zabbix",
            "nameid_format": "",
            "sign_messages": "0",
            "sign_assertions": "0",
            "sign_authn_requests": "0",
            "sign_logout_requests": "0",
            "sign_logout_responses": "0",
            "encrypt_nameid": "0",
            "encrypt_assertions": "0",
            "scim_status": "1",
            "provision_media": [
                {
                    "userdirectory_mediaid": "1",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "1",
                    "user_groups": [
                        {
                            "usrgrpid": "13"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "2",
            "idp_type": "1",
            "name": "AD server",
            "provision_status": "1",
            "description": "",
            "host": "host.example.com",
            "port": "389",
            "base_dn": "DC=zbx,DC=local",
            "search_attribute": "sAMAccountName",
            "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
            "start_tls": "0",
            "search_filter": "",
            "group_basedn": "OU=Zabbix,DC=zbx,DC=local",
            "group_name": "CN",
            "group_member": "member",
            "group_filter": "(%{groupattr}=CN=%{ref},OU=Users,OU=Zabbix,DC=zbx,DC=local)",
            "group_membership": "",
            "user_username": "givenName",
            "user_lastname": "sn",
            "user_ref_attr": "CN",
            "provision_media": [
                {
                    "userdirectory_mediaid": "2",
                    "name": "example.com",
                    "mediatypeid": "1",
                    "attribute": "user@example.com",
```

```
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00"
                }
            ],
            "provision_groups": [
                {
                    "name": "*",
                    "roleid": "4",
                    "user_groups": [
                        {
                            "usrgrpid": "8"
                        }
                    ]
                },
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API server #1",
            "provision_status": "0",
            "description": "",
            "host": "ldap://local.ldap",
            "port": "389",
            "base_dn": "ou=Users,dc=example,dc=org",
            "search_attribute": "uid",
            "bind_dn": "cn=ldap_search,dc=example,dc=org",
            "start_tls": "1",
            "search_filter": "",
            "group_basedn": "",
            "group_name": "",
            "group_member": "",
            "group_filter": "",
            "group_membership": "",
            "user_username": "",
            "user_lastname": "",
            "user_ref_attr": "",
            "provision_media": [],
            "provision_groups": []
        }
    ],
    "id": 1
}
```


参照

- User group

ソース

CUserDirectory::get() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

説明

`object userdirectory.test(array userDirectory)`

このメソッドは、ユーザーディレクトリの接続設定をテストします。

> **Note:**
> このメソッドは、ユーザープロビジョニングのユーザーディレクトリ設定と一致する設定データ（例：ユーザーに割り当てられるユーザーロール、ユーザーグループ、ユーザーメディア）をテストすることもできます。このタイプのテストでは、`provision_status` が有効に設定されているユーザーディレクトリ に対して API リクエストを送信する必要があります。

> **Note:**
> このメソッドは、スーパー管理者ユーザータイプでのみ使用できます。

パラメータ

(`object`) ユーザーディレクトリのプロパティ。

`userdirectory.get` API は `bind_password` フィールドを返さないため、`userdirectoryid` および/または `bind_password` を指定する必要があります。
このメソッドは、標準のユーザーディレクトリプロパティ に加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
test_username	string	ユーザーディレクトリでテストするユーザー名。
test_password	string	ユーザーディレクトリでテストするユーザー名に対応するパスワード。

戻り値

(`bool`) 成功した場合は true を返します。

例

既存のユーザーのユーザーディレクトリのテスト

"user1" のユーザーディレクトリ"3" をテストします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "bind_password": "password",
        "test_username": "user1",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

存在しないユーザーのユーザーディレクトリのテスト

存在しない”user2” でユーザーディレクトリ”3” のテストをします。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "3",
        "host": "127.0.0.1",
        "port": "389",
        "base_dn": "ou=Users,dc=example,dc=org",
        "search_attribute": "uid",
        "bind_dn": "cn=ldap_search,dc=example,dc=org",
        "test_username": "user2",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

ユーザープロビジョニングのユーザーディレクトリのテスト

”user3” プロビジョニングのユーザーディレクトリ設定と一致する構成されたデータのユーザーディレクトリ”3” をテストします (例えば、何かユーザーの役割、ユーザーグループ、ユーザーメディアがユーザーに割り当てられます)。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.test",
    "params": {
        "userdirectoryid": "2",
        "host": "host.example.com",
        "port": "389",
        "base_dn": "DC=zbx,DC=local",
        "search_attribute": "sAMAccountName",
        "bind_dn": "CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local",
        "test_username": "user3",
        "test_password": "password"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
```

```
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

ソース

CUserDirectory::test() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

説明

`object userdirectory.update(object/array userDirectory)`

このメソッドは、既存のユーザーディレクトリを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。

パラメーター

(object/array) 更新するユーザーディレクトリのプロパティ。

userdirectoryid プロパティは各ユーザーディレクトリに対して定義する必要があり、その他のプロパティはオプションです。渡されたプロパティのみが更新され、他は一切変更されません。

戻り値

(object) userdirectoryids プロパティの下にある更新されたユーザーディレクトリの ID を含むオブジェクトを返します。

例

ユーザーディレクトリのバインドパスワードの更新

ユーザーディレクトリの新しいバインドパスワードを設定します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

ユーザーディレクトリのマッピングの更新

ユーザーディレクトリ"2" のプロビジョニンググループマッピングとメディアタイプマッピングを更新します。

リクエスト：

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "2",
        "provision_media": [
                {
                    "userdirectory_mediaid": "2"
                }
            ],
            "provision_groups": [
                {
                    "name": "Zabbix administrators",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        },
                        {
                            "usrgrpid": "11"
                        }
                    ]
                }
            ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CUserDirectory::update() in *ui/include/classes/api/services/CUserDirectory.php*.

ユーザーマクロ

このクラスは、ホストレベルおよびグローバルのユーザーマクロを扱うために設計されています。

オブジェクト参照:

- グローバルマクロ
- ホストマクロ
    - マクロ設定

利用可能なメソッド:

- usermacro.create - 新しいホストマクロを作成
- usermacro.createglobal - 新しいグローバルマクロを作成
- usermacro.delete - ホストマクロを削除
- usermacro.deleteglobal - グローバルマクロを削除
- usermacro.get - ホストマクロおよびグローバルマクロを取得
- usermacro.update - ホストマクロを更新

- usermacro.updateglobal - グローバルマクロを更新

User macro オブジェクト

次のオブジェクトは、`usermacro`API に直接関連しています。

グローバルマクロ

グローバルマクロオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
globalmacroid	ID	グローバルマクロの ID。 Property behavior: - *read-only* - 更新操作では *required*
description	string	マクロの説明。
macro	string	マクロ文字列。 Property behavior: - 作成操作では *required*
type	integer	マクロのタイプ。 使用可能な値: 0 - *(デフォルト)* テキストマクロ; 1 - シークレットマクロ; 2 - Vault シークレット。
value	string	マクロの値。 Property behavior: - type が”Secret macro” に設定されている場合は *write-only* - 作成操作では *required*

ホストマクロ

ホストマクロオブジェクトは、ホスト、ホストプロトタイプ、またはテンプレートで利用可能なマクロを定義します。以下のプロパティがあります。

Property	Type	Description
hostmacroid	ID	マクロの ID。 Property behavior: - *read-only* - 更新操作で *required*
automatic	integer	マクロがディスカバリルールによって制御されるかどうかを定義します。 使用可能な値: 0 - *(default)* マクロはユーザーによって管理されます。 1 - マクロはディスカバリルールによって管理されます。 ユーザーは自動マクロを作成できません。 自動マクロを更新するには、これを手動に変換する必要があります。
config	object/array	Host Wizard でマクロをどのように表示するかを制御するマクロ設定。
description	string	マクロの説明。
hostid	ID	このマクロが属するホスト、ホストプロトタイプ、またはテンプレートの ID。 Property behavior: - *constant* - 作成操作で *required*

Property	Type	Description
macro	string	マクロ文字列。 Property behavior: - 作成操作で *required*
type	integer	マクロのタイプ。 使用可能な値: 0 - *(default)* テキストマクロ。 1 - シークレットマクロ。 2 - Vault シークレット。
value	string	マクロの値。 Property behavior: - `type` が”Secret macro” に設定されている場合は *write-only* - 作成操作で *required*

マクロ設定

マクロ設定オブジェクトは、ホストウィザードでマクロがどのように表示されるかを定義します。

プロパティ	型	説明
type	integer	マクロ入力フィールドのタイプ。 使用可能な値: 0 - マクロはホストウィザードで使用されない; 1 - テキストボックス; 2 - リスト; 3 - チェックボックス。 プロパティの動作: - *required*
label	string	マクロ入力フィールドのラベル。 プロパティの動作: - `type` が”Textbox”、”List”、または”Checkbox” に設定されている場合は *required*
description	text	マクロ入力フィールドの横に表示されるヘルプテキスト。Markdown 形式をサポートします。
priority	integer	マクロリスト内でのマクロ入力フィールドの位置。 `priority` が設定されていない場合、マクロはグループ化されていないマクロリストの末尾に追加されます。
required	integer	マクロを必須としてマークします。 使用可能な値: 0 - 必須ではない; 1 - 必須。 プロパティの動作: - `type` が”Textbox” または”List” に設定されている場合は *supported*
regex	string	テキストボックスフィールド内のユーザー入力を検証するための正規表現。 プロパティの動作: - `type` が”Textbox” に設定されている場合は *supported*
section_name	string	マクロがグループ化される折りたたみ可能なセクションのラベル。 `section_name` が設定されていない場合、マクロはグループ化されません。グループ化されていないマクロが最初に表示され、グループ化されたマクロはその下に、各折りたたみ可能セクション内で `priority` の順に表示されます。

プロパティ	型	説明
options	text	リスト項目またはチェックボックスの値を定義する JSON 文字列。 リストの場合: value および text プロパティを持つオブジェクトの配列。 例: [{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}] チェックボックスの場合: checked および unchecked プロパティを持つオブジェクト。 例: {"checked": true, "unchecked": false} プロパティの動作: - type が”List” または”Checkbox” に設定されている場合は *required*

usermacro.create

説明

`object usermacro.create(object/array hostMacros)`

このメソッドは、新しいホストマクロを作成することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成するホストマクロ。

このメソッドは、標準のホストマクロプロパティを持つホストマクロを受け付けます。

戻り値

(object) hostmacroids プロパティの下で作成されたホストマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたホストマクロの順序と一致します。

例

ホストマクロの作成

ホスト”10198” で値”public” を持つホストマクロ”{$SNMP_COMMUNITY}” を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
```

```
    "id": 1
}
```

ソース

CUserMacro::create() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

説明

`object usermacro.createglobal(object/array globalMacros)`

このメソッドは、新しいグローバルマクロを作成することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 作成するグローバルマクロ。

このメソッドは、標準のグローバルマクロのプロパティを持つグローバルマクロを受け付けます。

戻り値

(`object`) `globalmacroids` プロパティの下で作成されたグローバルマクロの ID を含むオブジェクトを返します。返される ID の順序は、渡されたグローバルマクロの順序と一致します。

例

グローバルマクロの作成

値が”public” のグローバルマクロ”{$SNMP_COMMUNITY}” を作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::createGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

説明

`object usermacro.delete(array hostMacroIds)`

このメソッドは、ホストマクロを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するホストマクロの ID。

戻り値

(object) hostmacroids プロパティの下にある削除されたホストマクロの ID を含むオブジェクトを返します。

例

複数のホストマクロの削除

2 つのホストマクロを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::delete() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

説明

`object usermacro.deleteglobal(array globalMacroIds)`

このメソッドは、グローバルマクロを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するグローバルマクロの ID。

戻り値

(object) globalmacroids プロパティの下にある削除されたグローバルマクロの ID を含むオブジェクトを返します。

例

複数のグローバルマクロの削除

2 つのグローバルマクロを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::deleteGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

説明

`integer/array usermacro.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってホストマクロおよびグローバルマクロを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
globalmacro	flag	ホストマクロではなくグローバルマクロを返します。
globalmacroids	ID/array	指定した ID を持つグローバルマクロのみを返します。
groupids	ID/array	指定したホストグループまたはテンプレートグループに属するホストまたはテンプレートのホストマクロのみを返します。
hostids	ID/array	指定したホストまたはテンプレートに属するマクロのみを返します。
hostmacroids	ID/array	指定した ID を持つホストマクロのみを返します。
inherited	boolean	`true` に設定した場合、テンプレートから継承されたホストプロトタイプのユーザーマクロのみを返します。
selectHostGroups	query	ホストマクロが属するホストグループを`hostgroups` プロパティで返します。 ホストマクロを取得する場合にのみ使用されます。
selectHosts	query	ホストマクロが属するホストを`hosts` プロパティで返します。 ホストマクロを取得する場合にのみ使用されます。

パラメータ	型	説明
selectTemplateGroups	query	テンプレートマクロが属するテンプレートグループを`templategroups` プロパティで返します。
selectTemplates	query	テンプレートマクロを取得する場合にのみ使用されます。 ホストマクロが属するテンプレートを`templates` プロパティで返します。
sortfield	string/array	ホストマクロを取得する場合にのみ使用されます。 指定したプロパティで結果をソートします。
countOutput	boolean	指定可能な値: `macro`。 これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合、取得されたオブジェクトの数

例

ホストのホストマクロを取得する

ホスト「10198」に定義されているすべてのホストマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "9",
            "hostid": "10198",
            "macro": "{$INTERFACE}",
            "value": "eth0",
            "description": "",
            "type": "0",
            "automatic": "0"
        },
        {
            "hostmacroid": "11",
```

```
            "hostid": "10198",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0",
            "automatic": "0"
        }
    ],
    "id": 1
}
```

テンプレートのホストマクロを取得しています

テンプレート”10265” に定義されている、マクロ名に”STATUS” を含むホストマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "The hostname or IP address of the Apache status page host.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "1",
                "section_name": "",
                "label": "Apache status host",
                "description": "Hostname or IP address of the Apache status page host.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "The URL path of the Apache status page.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
```

```
                "section_name": "",
                "label": "Apache status page path",
                "description": "URL path of the Apache status page.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "The port of the Apache status page.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "Apache status page port",
                "description": "In the range from 1 to 65535 inclusive.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "The request scheme, which may be either HTTP or HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Request scheme",
                "description": "Request scheme, which may be either HTTP or HTTPS.",
                "required": "0",
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```


グローバルマクロの取得

すべてのグローバルマクロを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
    ],
    "id": 1
}
```

ソース

CUserMacro::get() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

説明

`object usermacro.update(object/array hostMacros)`

このメソッドでは、既存のホストマクロを更新できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

更新する（object/array）ホストマクロのプロパティ。

各ホストマクロでは hostmacroid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

戻り値

（object）hostmacroids プロパティの下で更新されたホストマクロの ID を含むオブジェクトを返します。

例

ホストマクロの値の変更

ホストマクロの値を”public” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

検出ルールによって作成されたマクロの値を変更します。

検出ルールによって作成された" 自動" マクロを" 手動" マクロに変換し、その値を" 新しい値" に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::update() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

説明

`object usermacro.updateglobal(object/array globalMacros)`

このメソッドは、既存のグローバルマクロを更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

更新する (`object/array`) グローバルマクロのプロパティ。

各グローバルマクロには `globalmacroid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

戻り値

(object) globalmacroids プロパティの下で更新されたグローバルマクロの ID を含むオブジェクトを返します。

例

グローバルマクロの値の変更

グローバルマクロの値を”public” に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CUserMacro::updateGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

レポート

このクラスは、スケジュールされたレポートを扱うために設計されています。

オブジェクト参照:

- レポート
- ユーザー
- ユーザーグループ

利用可能なメソッド:

- report.create - 新しいスケジュールされたレポートを作成
- report.delete - スケジュールされたレポートを削除
- report.get - スケジュールされたレポートを取得
- report.update - スケジュールされたレポートを更新

Report オブジェクト

次のオブジェクトは、reportAPI に直接関連しています。

レポート

レポートオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
reportid	ID	レポートの ID。 プロパティの動作: - 読み取り専用 - 更新操作では 必須

プロパティ	型	説明
userid	ID	レポートを作成したユーザーの ID。 プロパティの動作: - 作成操作では 必須
name	string	レポートの一意な名前。 プロパティの動作: - 作成操作では 必須
dashboardid	ID	レポートのベースとなるダッシュボードの ID。 プロパティの動作: - 作成操作では 必須
period	integer	レポートを作成する対象期間。 指定可能な値: 0 - *(デフォルト)* 前日; 1 - 前週; 2 - 前月; 3 - 前年。
cycle	integer	期間の繰り返しスケジュール。 指定可能な値: 0 - *(デフォルト)* 毎日; 1 - 毎週; 2 - 毎月; 3 - 毎年。
start_time	integer	レポートを送信するために作成する時刻を、その日の開始からの秒数で表したもの。 デフォルト: 0。
weekdays	integer	レポートを送信する曜日。 指定可能なビットマップ値: 1 - 月曜日; 2 - 火曜日; 4 - 水曜日; 8 - 木曜日; 16 - 金曜日; 32 - 土曜日; 64 - 日曜日。 これはビットマスクフィールドです。指定可能なビットマップ値の任意の合計を指定できます（たとえば、月曜日・水曜日・金曜日の場合は 21）。 デフォルト: 0。
active_since	string	プロパティの動作: - `cycle` が”weekly” に設定されている場合は 必須 開始日。 指定可能な値: 空文字列 - *(デフォルト)* 未指定（0 として保存）; YYYY-MM-DD 形式の特定の日付（その日の開始時刻 (00:00:00) のタイムスタンプとして保存）。
active_till	string	終了日。 指定可能な値: 空文字列 - *(デフォルト)* 未指定（0 として保存）; YYYY-MM-DD 形式の特定の日付（その日の終了時刻 (23:59:59) のタイムスタンプとして保存）。
subject	string	レポートメッセージの件名。

プロパティ	型	説明
message	string	レポートメッセージの本文。
status	integer	レポートが有効か無効か。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
description	text	レポートの説明。
state	integer	レポートの状態。 指定可能な値: 0 - *(デフォルト)* レポートはまだ処理されていない; 1 - レポートが生成され、すべての受信者への送信に成功した; 2 - レポートの生成に失敗した; "info" にエラー情報が含まれる; 3 - レポートは生成されたが、一部（またはすべて）の受信者への送信に失敗した; "info" にエラー情報が含まれる。 プロパティの動作: - 読み取り専用
lastsent	timestamp	最後に正常に送信されたレポートの Unix タイムスタンプ。 プロパティの動作: - 読み取り専用
info	string	エラーの説明または追加情報。 プロパティの動作: - 読み取り専用

ユーザー

users オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
userid	ID	レポートの送信先ユーザーの ID。 プロパティの動作: - 必須
access_userid	ID	レポートの生成時に代理として使用するユーザーの ID。 0 - *(デフォルト)* 受信者としてレポートを生成。
exclude	integer	ユーザーをメーリングリストから除外するかどうか。 使用可能な値: 0 - *(デフォルト)* 含める; 1 - 除外。

ユーザーグループ

ユーザーグループオブジェクトには、次のプロパティがあります。

プロパティ	Type	説明
usrgrpid	ID	レポートの送信先となるユーザーグループの ID。 プロパティの動作: - 必須
access_userid	ID	レポートの生成時に代理として使用されるユーザーの ID。 0 - *(デフォルト)* 受信者としてレポートを生成。

report.create

説明

`object report.create(object/array reports)`

このメソッドでは、新しい定期レポートを作成できます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* ユーザータイプでのみ利用できます。
> メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object/array`) 作成するスケジュールレポート。

このメソッドは、標準のスケジュールレポートのプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
users	object/array	レポートの送信先となるユーザー。 パラメータの動作: - `user_groups` が設定されていない場合は必須
user_groups	object/array	レポートの送信先となるユーザーグループ。 パラメータの動作: - `users` が設定されていない場合は必須

戻り値

(`object`) `reportids` プロパティの下で作成されたスケジュールされたレポートの ID を含むオブジェクトを返します。返される ID の順序は、渡されたスケジュール済みレポートの順序と一致します。

例

スケジュールされたレポートの作成

2021-04-01 から 2021-08-31 までの期間、毎週月曜日から金曜日の 12:00 に、前週分の週次レポートを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.create",
    "params": {
        "userid": "1",
        "name": " 週次レポート",
        "dashboardid": "1",
        "period": "1",
        "cycle": "1",
        "start_time": "43200",
        "weekdays": "31",
        "active_since": "2021-04-01",
        "active_till": "2021-08-31",
        "subject": " 週次レポート",
        "message": " レポートの添付テキスト",
        "status": "1",
        "description": " レポートの説明",
        "users": [
            {
                "userid": "1",
                "access_userid": "1",
                "exclude": "0"
            },
            {
                "userid": "2",
```

```
                "access_userid": "0",
                "exclude": "1"
            }
        ],
        "user_groups": [
            {
                "usrgrpid": "7",
                "access_userid": "0"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- ユーザー
- ユーザーグループ

ソース

CReport::create() in *ui/include/classes/api/services/CReport.php*.

report.delete

説明

`object report.delete(array reportids)`

このメソッドは、定期レポートを削除することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する定期レポートの ID。

戻り値

(object) reportids プロパティの下にある削除された定期レポートの ID を含むオブジェクトを返します。

例

複数の定期レポートの削除

2 つの定期レポートの削除

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CReport::delete() in *ui/include/classes/api/services/CReport.php*.

report.get

説明

`integer/array report.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってスケジュール済みレポートを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。
メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
reportids	ID/array	指定したレポート ID を持つ定期レポートのみを返します。
expired	boolean	`true` に設定した場合、期限切れの定期レポートのみを返します。
selectUsers	query	レポートの送信先として設定されているユーザーを含む`users` プロパティを返します。
selectUserGroups	query	レポートの送信先として設定されているユーザーグループを含む`user_groups` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `reportid`, `name`, `status`。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

レポートデータの取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": [
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "Weekly report",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Weekly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "Report description",
            "state": "1",
            "lastsent": "1613563219",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                },
                {
                    "userid": "2",
                    "access_userid": "0",
                    "exclude": "1"
                }
            ],
            "user_groups": [
                {
                    "usrgrpid": "7",
                    "access_userid": "0"
                }
            ]
        },
        {
            "reportid": "2",
            "userid": "1",
```

```
            "name": "Monthly report",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Monthly report",
            "message": "Report accompanying text",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```


関連項目

- ユーザー
- ユーザーグループ

ソース

CReport::get() in *ui/include/classes/api/services/CReport.php*.

report.update

説明

`object report.update(object/array reports)`

このメソッドでは、既存の定期レポートを更新できます。

Note:
このメソッドは、*Admin* および *Super admin* のユーザータイプでのみ利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 更新する定期レポートのプロパティ。

各定期レポートでは `reportid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外はすべて変更されません。

このメソッドは、標準の定期レポートのプロパティに加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
users	object/array	定期レポートに現在割り当てられているユーザーを置き換えるためのユーザー。 パラメータの動作: - `user_groups` が設定されていない場合は 必須

Parameter	Type	Description
user_groups	object/array	定期レポートに現在割り当てられているユーザーグループを置き換えるためのユーザーグループ。 パラメータの動作: - `users` が設定されていない場合は 必須

戻り値

(`object`) `reportids` プロパティの下で更新されたスケジュールされたレポートの ID を含むオブジェクトを返します。

例

スケジュールレポートの無効化

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- ユーザー
- ユーザーグループ

ソース

CReport::update() in *ui/include/classes/api/services/CReport.php*.

ロール

このクラスはユーザーロールで動作するように設計されています。

オブジェクト参照:

- ロール
- ロールルール
  - UI 要素
  - サービス
  - サービスタグ
  - モジュール
  - アクション

利用可能なメソッド:

- role.create - 新しいユーザーロールを作成
- role.delete - ユーザーロールを削除
- role.get - ユーザーロールを取得
- role.update - ユーザーロールを更新

Role オブジェクト

次のオブジェクトは、`role`API に直接関連しています。

ロール

ロールオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
roleid	ID	ロールの ID。 Property behavior: - *read-only* - 更新操作で *required*
name	string	ロール名。 Property behavior: - 作成操作で *required*
type	integer	ユーザータイプ。 指定可能な値: 1 - *(default)* ユーザー; 2 - 管理者; 3 - 特権管理者。 Property behavior: - 作成操作で *required*
readonly	integer	ロールが読み取り専用かどうか。 指定可能な値: 0 - *(default)* いいえ; 1 - はい。 Property behavior: - *read-only*

ロールルール

ロールルールオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
ui	array	UI 要素オブジェクトの配列。
ui.default_access	integer	新しい UI 要素へのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
services.read.mode	integer	サービスへの読み取り専用アクセス。 指定可能な値: 0 - `services.read.list` で指定された、または `services.read.tag` プロパティに一致したサービスへの読み取り専用アクセス; 1 - *(デフォルト)* すべてのサービスへの読み取り専用アクセス。
services.read.list	array	Serviceオブジェクトの配列。 指定したサービスおよびその子サービスに対して、ユーザーロールに読み取り専用アクセスが付与されます。読み取り専用アクセスは、サービスへの読み書きアクセスを上書きしません。 プロパティの動作: - `services.read.mode` が”0” に設定されている場合に サポート

プロパティ	型	説明
services.read.tag	object	Service tagオブジェクトの配列。 タグに一致したサービスおよびその子サービスに対して、ユーザーロールに読み取り専用アクセスが付与されます。読み取り専用アクセスは、サービスへの読み書きアクセスを上書きしません。 プロパティの動作: - `services.read.mode` が"0" に設定されている場合に サポート
services.write.mode	integer	サービスへの読み書きアクセス。 指定可能な値: 0 - *(デフォルト)* `services.write.list` で指定された、または `services.write.tag` プロパティに一致したサービスへの読み書きアクセス; 1 - すべてのサービスへの読み書きアクセス。
services.write.list	array	Serviceオブジェクトの配列。 指定したサービスおよびその子サービスに対して、ユーザーロールに読み書きアクセスが付与されます。読み書きアクセスは、サービスへの読み取り専用アクセスを上書きします。 プロパティの動作: - `services.write.mode` が"0" に設定されている場合に サポート
services.write.tag	object	Service tagオブジェクトの配列。 タグに一致したサービスおよびその子サービスに対して、ユーザーロールに読み書きアクセスが付与されます。読み書きアクセスは、サービスへの読み取り専用アクセスを上書きします。 プロパティの動作: - `services.write.mode` が"0" に設定されている場合に サポート
modules	array	moduleオブジェクトの配列。
modules.default_access	integer	新しいモジュールへのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
api.access	integer	API へのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。
api.mode	integer	`api` プロパティに一覧表示された API メソッドの扱い方を指定するモード。 指定可能な値: 0 - *(デフォルト)* 拒否リスト; 1 - 許可リスト。
api	array	API メソッドの配列。
actions	array	actionオブジェクトの配列。
actions.default_access	integer	新しいアクションへのアクセスを有効にするかどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

UI 要素

UI 要素オブジェクトには、次のプロパティがあります。

プロパティ	型	説明
name	string	UI 要素の名前。 Role オブジェクトの type が"User"、"Admin"、または"Super admin" に設定されている場合の指定可能な値: monitoring.dashboard - ダッシュボード; monitoring.problems - 監視 → 障害; monitoring.hosts - 監視 → ホスト; monitoring.latest_data - 監視 → 最新データ; monitoring.maps - 監視 → マップ; services.services - サービス → サービス; services.sla_report - サービス → *SLA* レポート; inventory.overview - インベントリ → 概要; inventory.hosts - インベントリ → ホスト; reports.availability_report - レポート → 可用性レポート; reports.top_triggers - レポート → トリガー *top 100*。 Role オブジェクトの type が"Admin" または"Super admin" に設定されている場合の指定可能な値: monitoring.discovery - 監視 → ディスカバリ; services.sla - サービス → *SLA*; reports.scheduled_reports - レポート → 定期レポート; reports.notifications - レポート → 通知; configuration.template_groups - データ収集 → テンプレートグループ; configuration.host_groups - データ収集 → ホストグループ; configuration.templates - データ収集 → テンプレート; configuration.hosts - データ収集 → ホスト; configuration.maintenance - データ収集 → メンテナンス; configuration.discovery - データ収集 → ディスカバリ; configuration.trigger_actions - アラート → アクション → トリガーアクション; configuration.service_actions - アラート → アクション → サービスアクション; configuration.discovery_actions - アラート → アクション → ディスカバリアクション; configuration.autoregistration_actions - アラート → アクション → 自動登録アクション; configuration.internal_actions - アラート → アクション → 内部アクション。 Role オブジェクトの type が"Super admin" に設定されている場合の指定可能な値: reports.system_info - レポート → システム情報; reports.audit - レポート → 監査ログ; reports.action_log - レポート → アクションログ; configuration.event_correlation - データ収集 → イベント相関; administration.media_types - アラート → メディアタイプ; administration.scripts - アラート → スクリプト; administration.user_groups - ユーザー → ユーザーグループ; administration.user_roles - ユーザー → ユーザーロール; administration.users - ユーザー → ユーザー; administration.api_tokens - ユーザー → *API* トークン; administration.authentication - ユーザー → 認証; administration.general - 管理 → 一般; administration.audit_log - 管理 → 監査ログ; administration.housekeeping - 管理 → ハウスキーピング; administration.proxy_groups - 管理 → プロキシグループ; administration.proxies - 管理 → プロキシ; administration.macros - 管理 → マクロ; administration.queue - 管理 → キュー。 プロパティの動作: - 必須

プロパティ	型	説明
status	integer	UI 要素へのアクセスが有効かどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

サービス

プロパティ	Type	説明
serviceid	ID	サービスの ID。 プロパティの動作: - 必須

サービスタグ

プロパティ	Type	説明
tag	string	タグ名。 空文字列が指定された場合、サービスタグはサービスのマッチングに使用されません。 プロパティの動作: - 必須
value	string	タグ値。 値が指定されていない場合、または空文字列が指定された場合、サービスのマッチングにはタグ名のみが使用されます。

モジュール

モジュールオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
moduleid	ID	モジュールの ID。 プロパティの動作: - 必須
status	integer	モジュールへのアクセスが有効かどうか。 指定可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

アクション

アクションオブジェクトには以下のプロパティがあります。

プロパティ	型	説明
name	string	アクションの名前。 Role オブジェクトの type が”User”、”Admin”、または”Super admin” に設定されている場合の使用可能な値: edit_dashboards - ダッシュボードの作成と編集; edit_maps - マップの作成と編集; add_problem_comments - 障害コメントの追加; change_severity - 障害の深刻度の変更; acknowledge_problems - 障害の確認; suppress_problems - 障害の抑制; close_problems - 障害のクローズ; execute_scripts - スクリプトの実行; manage_api_tokens - API トークンの管理; change_problem_ranking - 障害の順位付けを原因から症状へ、またはその逆に変更; edit_own_media - 自身のメディアの作成/編集を許可。 Role オブジェクトの type が”Admin” または”Super admin” に設定されている場合の使用可能な値: edit_maintenance - メンテナンスの作成と編集; manage_scheduled_reports - スケジュールレポートの管理, manage_sla - SLA の管理。 Role オブジェクトの type が”User” または”Admin” に設定されている場合の使用可能な値: invoke_execute_now - ホストに対して読み取り権限のみを持つユーザーによるアイテムチェックの実行を許可。 Role オブジェクトの type が”Super admin” に設定されている場合の使用可能な値: edit_user_media - ユーザーのメディアの作成/編集を許可。 プロパティの動作: - 必須
status	integer	アクションの実行アクセスが有効かどうか。 使用可能な値: 0 - 無効; 1 - *(デフォルト)* 有効。

role.create

説明

`object role.create(object/array roles)`

このメソッドでは、新しいロールを作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

`(object/array)` 作成するロール。

このメソッドは、標準のロールプロパティに加えて、以下のパラメータを受け付けます。

パラメータ	Type	説明
rules	array	ロール用に作成するロールルール。

戻り値

(object) roleids プロパティの下に作成されたロールの ID を含むオブジェクトを返します。返される ID の順序は、渡されたロールの順序と一致します。

例

ロールの作成

タイプが「User」で、2 つの UI 要素へのアクセスが拒否されたロールを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.create",
    "params": {
        "name": "Operator",
        "type": "1",
        "rules": {
            "ui": [
                {
                    "name": "monitoring.hosts",
                    "status": "0"
                },
                {
                    "name": "monitoring.maps",
                    "status": "0"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

関連項目

- ロールルール
- UI 要素
- モジュール
- アクション

ソース

CRole::create() in *ui/include/classes/api/services/CRole.php*.

role.delete

説明

`object role.delete(array roleids)`

このメソッドは、ユーザーの役割を削除することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するユーザーの役割の ID。

戻り値

(object) roleids プロパティの下にある削除されたユーザーの役割の ID を含むオブジェクトを返します。

例

複数のユーザーの役割の削除

2 つのユーザーの役割を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

ソース

CRole::delete() in *ui/include/classes/api/services/CRole.php*.

role.get

説明

`integer/array role.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってロールを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(object) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
roleids	ID/array	指定した ID を持つロールのみを返します。
selectRules	query	ロールルールを含む`rules` プロパティを返します。

パラメータ	Type	説明
selectUsers	query	ロールが割り当てられているユーザーを含むusers プロパティを返します。 ユーザータイプに基づく制限については、user.get を参照してください。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: roleid, name。
countOutput	boolean	これらのパラメータについては、reference commentary で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

ロールデータの取得

「Super admin role」ロールのデータとそのアクセスルールを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
```

```
            "name": "monitoring.problems",
            "status": "1"
        },
        {
            "name": "monitoring.hosts",
            "status": "1"
        },
        {
            "name": "monitoring.latest_data",
            "status": "1"
        },
        {
            "name": "monitoring.maps",
            "status": "1"
        },
        {
            "name": "services.services",
            "status": "1"
        },
        {
            "name": "services.sla_report",
            "status": "1"
        },
        {
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
```

```
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
        },
        {
            "name": "configuration.service_actions",
            "status": "1"
        },
        {
            "name": "configuration.discovery_actions",
            "status": "1"
        },
        {
            "name": "configuration.autoregistration_actions",
            "status": "1"
        },
        {
            "name": "configuration.internal_actions",
            "status": "1"
        },
        {
            "name": "reports.system_info",
            "status": "1"
        },
        {
            "name": "reports.audit",
            "status": "1"
        },
        {
            "name": "reports.action_log",
            "status": "1"
        },
        {
            "name": "configuration.event_correlation",
            "status": "1"
        },
        {
            "name": "administration.media_types",
            "status": "1"
        },
        {
            "name": "administration.scripts",
            "status": "1"
        },
        {
```

```
            "name": "administration.user_groups",
            "status": "1"
        },
        {
            "name": "administration.user_roles",
            "status": "1"
        },
        {
            "name": "administration.users",
            "status": "1"
        },
        {
            "name": "administration.api_tokens",
            "status": "1"
        },
        {
            "name": "administration.authentication",
            "status": "1"
        },
        {
            "name": "administration.general",
            "status": "1"
        },
        {
            "name": "administration.audit_log",
            "status": "1"
        },
        {
            "name": "administration.housekeeping",
            "status": "1"
        },
        {
            "name": "administration.proxy_groups",
            "status": "1"
        },
        {
            "name": "administration.proxies",
            "status": "1"
        },
        {
            "name": "administration.macros",
            "status": "1"
        },
        {
            "name": "administration.queue",
            "status": "1"
        }
    ],
    "ui.default_access": "1",
    "services.read.mode": "1",
    "services.read.list": [],
    "services.read.tag": {
        "tag": "",
        "value": ""
    },
    "services.write.mode": "1",
    "services.write.list": [],
    "services.write.tag": {
        "tag": "",
        "value": ""
    },
    "modules": [
```

```
                {
                    "moduleid": 1,
                    "status": "1"
                },
                {
                    "moduleid": 2,
                    "status": "1"
                },
                {
                    "moduleid": 3,
                    "status": "1"
                },
                {
                    "moduleid": 4,
                    "status": "1"
                },
                {
                    "moduleid": 5,
                    "status": "1"
                },
                {
                    "moduleid": 6,
                    "status": "1"
                },
                {
                    "moduleid": 7,
                    "status": "1"
                },
                {
                    "moduleid": 8,
                    "status": "1"
                },
                {
                    "moduleid": 9,
                    "status": "1"
                },
                {
                    "moduleid": 10,
                    "status": "1"
                },
                {
                    "moduleid": 11,
                    "status": "1"
                },
                {
                    "moduleid": 12,
                    "status": "1"
                },
                {
                    "moduleid": 13,
                    "status": "1"
                },
                {
                    "moduleid": 14,
                    "status": "1"
                },
                {
                    "moduleid": 15,
                    "status": "1"
                },
                {
                    "moduleid": 16,
```

```
            "status": "1"
        },
        {
            "moduleid": 17,
            "status": "1"
        },
        {
            "moduleid": 18,
            "status": "1"
        },
        {
            "moduleid": 19,
            "status": "1"
        },
        {
            "moduleid": 20,
            "status": "1"
        },
        {
            "moduleid": 21,
            "status": "1"
        },
        {
            "moduleid": 22,
            "status": "1"
        },
        {
            "moduleid": 23,
            "status": "1"
        },
        {
            "moduleid": 24,
            "status": "1"
        },
        {
            "moduleid": 25,
            "status": "1"
        },
        {
            "moduleid": 26,
            "status": "1"
        },
        {
            "moduleid": 27,
            "status": "1"
        },
        {
            "moduleid": 28,
            "status": "1"
        },
        {
            "moduleid": 29,
            "status": "1"
        },
        {
            "moduleid": 30,
            "status": "1"
        },
        {
            "moduleid": 31,
            "status": "1"
        },
```

```
            {
                "moduleid": 32,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
            {
                "name": "edit_maps",
                "status": "1"
            },
            {
                "name": "acknowledge_problems",
                "status": "1"
            },
            {
                "name": "suppress_problems",
                "status": "1"
            },
            {
                "name": "close_problems",
                "status": "1"
            },
            {
                "name": "change_severity",
                "status": "1"
            },
            {
                "name": "add_problem_comments",
                "status": "1"
            },
            {
                "name": "execute_scripts",
                "status": "1"
            },
            {
                "name": "manage_api_tokens",
                "status": "1"
            },
            {
                "name": "edit_maintenance",
                "status": "1"
            },
            {
                "name": "manage_scheduled_reports",
                "status": "1"
            },
            {
                "name": "manage_sla",
                "status": "1"
            },
            {
                "name": "invoke_execute_now",
                "status": "1"
            },
```

```
                    {
                        "name": "change_problem_ranking",
                        "status": "1"
                    },
                    {
                        "name": "edit_own_media",
                        "status": "1"
                    },
                    {
                        "name": "edit_user_media",
                        "status": "1"
                    }
                ],
                "actions.default_access": "1"
            }
        }
    ],
    "id": 1
}
```


関連項目

- ロールルール
- ユーザー

ソース

CRole::get() in *ui/include/classes/api/services/CRole.php*.

role.update

説明

`object role.update(object/array roles)`

このメソッドでは、既存のロールを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(`object/array`) 更新するロールのプロパティ。

各ロールに対して `roleid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他は変更されません。

このメソッドは、標準のロールプロパティ に加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
rules	array	ロールに対して更新するアクセスrules。

戻り値

(`object`) `roleids` プロパティの下で更新されたロールの ID を含むオブジェクトを返します。

例

スクリプト実行機能の無効化

ID が"5" のロールを更新し、スクリプトを実行する機能を無効にします。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

API へのアクセス制限

ID「5」のロールを更新し、任意の「create」、「update」、または「delete」メソッドの呼び出しを拒否します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

ソース

CRole::update() in *ui/include/classes/api/services/CRole.php*.

値マップ

このクラスは値マップを操作するために設計されています。

オブジェクト参照:

- 値マップ
  - 値のマッピング

利用可能なメソッド:

- valuemap.create - 新しい値マップを作成
- valuemap.delete - 値マップを削除
- valuemap.get - 値マップを取得
- valuemap.update - 値マップを更新

Value map オブジェクト

次のオブジェクトは、`valuemap`API に直接関連しています。

値のマッピング

値のマッピングオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
valuemapid	ID	値のマッピングの ID。 プロパティの動作: - 読み取り専用 - 更新操作で 必須
hostid	ID	この値のマッピングが属するホストまたはテンプレートの ID。 プロパティの動作: - 定数 - 作成操作で 必須
name	string	値のマッピングの名前。 プロパティの動作: - 作成操作で 必須
mappings	array	現在の値のマッピングの値マッピング。マッピングオブジェクトの詳細は以下を参照してください。 プロパティの動作: - 作成操作で 必須
uuid	string	インポートした値のマッピングを既存のものに関連付けるために使用される汎用一意識別子。指定されていない場合は自動生成されます。 プロパティの動作: - 値のマッピングがテンプレートに属する場合に サポート

値のマッピング

値のマッピングオブジェクトは、値のマップの値のマッピングを定義します。
このオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
type	integer	マッピングの一致タイプ。 指定可能な値: 0 - *(デフォルト)* 値が等しい場合にマッピングが適用されます。 1 - 値が以上の場合にマッピングが適用されます[1]。 2 - 値が以下の場合にマッピングが適用されます[1]。 3 - 値が範囲内の場合にマッピングが適用されます（範囲には端点を含みます。カンマ文字で区切って複数の範囲を定義できます）[1]。 4 - 値が正規表現に一致する場合にマッピングが適用されます[2]。 5 - 一致が見つからない場合、マッピングは適用されず、デフォルト値が使用されます。 type が”0”、”1”、”2”、”3”、”4” に設定されている場合、value は空にできません。
value	string	type が”5” に設定されている場合、value は空でなければなりません。 元の値。 Property behavior: - type が”1”、”2”、”3”、”4” に設定されている場合は *required* - type が”5” に設定されている場合は *supported*
newvalue	string	元の値がマッピングされる先の値。 Property behavior: - *required*

[1] 値の型が”numeric unsigned”、”numeric float” のアイテムでのみサポートされます。
[2] 値の型が”character” のアイテムでのみサポートされます。

valuemap.create

説明

`object valuemap.create(object/array valuemaps)`

このメソッドは、新しいバリューマップを作成することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 作成する値マップ。

このメソッドは、標準の値マッププロパティを持つ値マップを受け付けます。

戻り値

(object) valuemapids プロパティの下に作成されたバリューマップの ID を含むオブジェクトを返します。返される ID の順序は、渡されたバリューマップの順序と一致します。

例

バリューマップの作成

2 つのマッピングを使用して 1 つのバリューマップを作成します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
        "hostid": "50009",
```

```
        "name": "Service state",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Up"
            },
            {
                "type": "5",
                "newvalue": "Down"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::create() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

説明

`object valuemap.delete(array valuemapids)`

このメソッドは、バリューマップを削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除するバリューマップの ID。

戻り値

(object) valuemapids プロパティの下にある削除されたバリューマップの ID を含むオブジェクトを返します。

例

複数のバリューマップの削除

2 つのバリューマップを削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
    "params": [
        "1",
        "2"
    ],
```

```
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::delete() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

説明

`integer/array valuemap.get(object parameters)`

このメソッドでは、指定されたパラメータに従って値マップを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロールの設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
valuemapids	ID/array	指定した ID の値マップのみを返します。
selectMappings	query	現在の値マップの値マッピングを`mappings` プロパティで返します。 `count` をサポートします。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `valuemapid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列

- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

値マップの取得

設定されているすべての値マップを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status"
        },
        {
            "valuemapid": "5",
            "name": "APC Battery Status"
        },
        {
            "valuemapid": "7",
            "name": "Dell Open Manage System Status"
        }
    ],
    "id": 1
}
```

マッピングを含む 1 つの値マップを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "valuemapid": "4",
            "name": "APC Battery Replacement Status",
            "mappings": [
                {
                    "type": "0",
                    "value": "1",
                    "newvalue": "unknown"
```

```
                },
                {
                    "type": "0",
                    "value": "2",
                    "newvalue": "notInstalled"
                },
                {
                    "type": "0",
                    "value": "3",
                    "newvalue": "ok"
                },
                {
                    "type": "0",
                    "value": "4",
                    "newvalue": "failed"
                },
                {
                    "type": "0",
                    "value": "5",
                    "newvalue": "highTemperature"
                },
                {
                    "type": "0",
                    "value": "6",
                    "newvalue": "replaceImmediately"
                },
                {
                    "type": "0",
                    "value": "7",
                    "newvalue": "lowCapacity"
                }
            ]
        }
    ],
    "id": 1
}
```


ソース

CValueMap::get() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

説明

`object valuemap.update(object/array valuemaps)`

このメソッドでは、既存の値マップを更新できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロール を参照してください。

パラメータ

(`object/array`) 更新する値マップのプロパティ。

各値マップでは `valuemapid` プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、それ以外は変更されません。

戻り値

(`object`) `valuemapids` プロパティの下にある更新されたバリューマップの ID を含むオブジェクトを返します。

例

値マップ名の変更

値マップ名を「Device status」に変更します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

1 つの値マップのマッピングを変更する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

ソース

CValueMap::update() in *ui/include/classes/api/services/CValueMap.php*.

履歴

このクラスは履歴データを扱うために設計されています。

オブジェクト参照:

- 浮動小数点履歴
- 整数履歴
- 文字列履歴
- テキスト履歴
- ログ履歴

利用可能なメソッド:

- history.clear - 履歴データをクリア
- history.get - 履歴データを取得
- history.push - 履歴データを Zabbix サーバーに送信

History オブジェクト

次のオブジェクトは、`history`API に直接関連しています。

> **Note:**
> History オブジェクトはアイテムの情報の種類によって異なります。これらは Zabbix サーバーによって作成され、API から変更することはできません。

浮動小数点数の履歴

浮動小数点数の履歴オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	float	受信した値。

整数の履歴

整数の履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	integer	受信した値。

文字列の履歴

文字列履歴オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	string	受信した値。

テキスト履歴

テキスト履歴オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
id	ID	履歴エントリの ID。
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	text	受信した値。

ログ履歴

ログ履歴オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
id	ID	履歴エントリの ID。
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
logeventid	integer	Windows イベントログエントリの ID。
ns	integer	値を受信した時点のナノ秒。
severity	integer	Windows イベントログエントリのレベル。
source	string	Windows イベントログエントリのソース。
timestamp	timestamp	Windows イベントログエントリの時刻。
value	text	受信した値。

バイナリ履歴

バイナリ履歴オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
clock	timestamp	その値を受信した時刻。
itemid	ID	関連するアイテムの ID。
ns	integer	値を受信した時点のナノ秒。
value	text	受信した値。

history.clear

説明

`object history.clear(array itemids)`

このメソッドは、アイテムのヒストリを消去することができます。

> **Note:**
> このメソッドは、*Admin* および *Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) ヒストリを消去するアイテムの ID。

戻り値

(object) itemids プロパティの下にあるヒストリが消去されたアイテムの ID を含むオブジェクトを返します。

例

ヒストリの消去

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
```

```
        "13205"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

ソース

CHistory::clear() in *ui/include/classes/api/services/CHistory.php*.

history.get

説明

`integer/array history.get(object parameters)`

このメソッドでは、指定されたパラメータに従って履歴データを取得できます。

Attention:
このメソッドは、housekeeper によってまだ削除されていない場合、削除済みエンティティの履歴データを返すことがあります。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
history	integer	返す履歴オブジェクトのタイプ。 指定可能な値: 0 - 数値浮動小数点; 1 - 文字; 2 - ログ; 3 - *(デフォルト)* 符号なし数値; 4 - テキスト; 5 - バイナリ。
hostids	ID/array	指定したホストの履歴のみを返します。
itemids	ID/array	指定したアイテムの履歴のみを返します。
time_from	timestamp	指定した時刻以降に受信した値のみを返します。
time_till	timestamp	指定した時刻以前に受信した値のみを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `itemid`, `clock`, `ns`。

パラメータ	型	説明
search	object	指定したパターンに一致する結果を返します（大文字と小文字は区別されません）。 キーがプロパティ名、値が検索する文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、LIKE "%…%" 検索が実行されます。 history が 6 (JSON) に設定されている場合、value による検索はサポートされません。
countOutput	boolean	これらのパラメータについては、リファレンスコメントを参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

アイテムの履歴データの取得

numeric(float) アイテムが受信した最新の値を 10 件返します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23296",
            "clock": "1351090996",
            "value": "0.085",
            "ns": "563157632"
        },
        {
            "itemid": "23296",
            "clock": "1351090936",
```

```
            "value": "0.16",
            "ns": "549216402"
        },
        {
            "itemid": "23296",
            "clock": "1351090876",
            "value": "0.18",
            "ns": "537418114"
        },
        {
            "itemid": "23296",
            "clock": "1351090816",
            "value": "0.21",
            "ns": "522659528"
        },
        {
            "itemid": "23296",
            "clock": "1351090756",
            "value": "0.215",
            "ns": "507809457"
        },
        {
            "itemid": "23296",
            "clock": "1351090696",
            "value": "0.255",
            "ns": "495509699"
        },
        {
            "itemid": "23296",
            "clock": "1351090636",
            "value": "0.36",
            "ns": "477708209"
        },
        {
            "itemid": "23296",
            "clock": "1351090576",
            "value": "0.375",
            "ns": "463251343"
        },
        {
            "itemid": "23296",
            "clock": "1351090516",
            "value": "0.315",
            "ns": "447947017"
        },
        {
            "itemid": "23296",
            "clock": "1351090456",
            "value": "0.275",
            "ns": "435307141"
        }
    ],
    "id": 1
}
```


ソース

CHistory::get() in *ui/include/classes/api/services/CHistory.php*.

history.push

説明

`object history.push(object/array itemHistoryData)`

このメソッドは、アイテムヒストリデータを Zabbix サーバーに送信できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 送信するアイテムの履歴データ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
itemid	ID	関連するアイテムの ID。 パラメータの動作: - host および key が設定されていない場合は 必須
host	string	ホストの技術名。 パラメータの動作: - itemid が設定されていない場合は 必須
key	string	アイテムキー。 パラメータの動作: - itemid が設定されていない場合は 必須
value	mixed	アイテムの値。 パラメータの動作: - 必須
clock	timestamp	値を受信した時刻。
ns	integer	値を受信した時点のナノ秒。

戻り値

(object) データ送信操作の結果を返します。

例

アイテムヒストリデータの送信

Zabbix サーバーへ、アイテム"10600"、"10601"、"999999" のヒストリデータを送信します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "history.push",
    "params": [
        {
            "itemid": 10600,
            "value": 0.5,
            "clock": 1690891294,
            "ns": 45440940
        },
        {
            "itemid": 10600,
            "value": 0.6,
            "clock": 1690891295,
            "ns": 312431
        },
        {
            "itemid": 10601,
            "value": "[Tue Aug 01 15:01:35 2023] [error] [client 1.2.3.4] File does not exist: /var/www/ht
        },
        {
            "itemid": 999999,
            "value": 123
        }
```

```
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Item is disabled."
            },
            {
                "error": "No permissions to referred object or it does not exist."
            }
        ]
    },
    "id": 1
}
```

参照

- Trapper items
- HTTP agent items
- Host
- Item

ソース

CHistory::push() in *ui/include/classes/api/services/CHistory.php*.

検出されたサービス

このクラスは、検出されたサービスを操作するために設計されています。

オブジェクト参照:

- 検出されたサービス

利用可能なメソッド:

- dservice.get - 検出されたサービスを取得します

Discovered service オブジェクト

次のオブジェクトは、dserviceAPI に直接関連しています。

検出されたサービス

> **Note:**
> 検出されたサービスは Zabbix サーバーによって作成され、API 経由で変更することはできません。

検出されたサービスオブジェクトには、ホスト上のネットワークディスカバリルールによって検出されたサービスに関する情報が含まれます。以下のプロパティがあります。

プロパティ	型	説明
dserviceid	ID	検出されたサービスの ID。
dcheckid	ID	サービスの検出に使用されたディスカバリチェックの ID。
dhostid	ID	サービスを実行している検出されたホストの ID。
dns	string	サービスを実行しているホストの DNS。
ip	string	サービスを実行しているホストの IP アドレス。
lastdown	timestamp	検出されたサービスが最後に停止した時刻。
lastup	timestamp	検出されたサービスが最後に稼働状態になった時刻。
port	integer	サービスポート番号。
status	integer	サービスのステータス。 使用可能な値: 0 - サービス稼働; 1 - サービス停止。
value	string	Zabbix エージェント、SNMPv1、SNMPv2、または SNMPv3 のディスカバリチェックの実行時にサービスから返される値。

dservice.get

説明

`integer/array dservice.get(object parameters)`

このメソッドは、指定したパラメーターに応じた検出されたサービスを取得できます。

Note:
このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dserviceids	ID/array	指定した ID を持つ検出済みサービスのみを返します。
dhostids	ID/array	指定した検出済みホストに属する検出済みサービスのみを返します。
dcheckids	ID/array	指定したディスカバリチェックによって検出された検出済みサービスのみを返します。
druleids	ID/array	指定したディスカバリルールによって検出された検出済みサービスのみを返します。
selectDRules	query	サービスを検出したディスカバリルールの配列を含む`drules` プロパティを返します。
selectDHosts	query	サービスが属する検出済みホストの配列を含む`dhosts` プロパティを返します。
selectHosts	query	サービスと同じ IP アドレスおよびプロキシを持つホストを含む`hosts` プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectHosts` - 結果は `hostid` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dserviceid`, `dhostid`, `ip`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	

パラメータ	型	説明
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

ホストで検出されたサービスの取得

検出されたホスト"11" で検出されたすべてサービスを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "dserviceid": "12",
            "dhostid": "11",
            "value": "",
            "port": "80",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650607",
            "dcheckid": "5",
            "ip": "192.168.1.134",
            "dns": "john.local"
        },
        {
            "dserviceid": "13",
            "dhostid": "11",
            "value": "",
            "port": "21",
            "status": "1",
            "lastup": "0",
            "lastdown": "1348650610",
            "dcheckid": "6",
            "ip": "192.168.1.134",
            "dns": "john.local"
        }
    ],
```

```
    "id": 1
}
```

関連項目

- 検出されたホスト
- ディスカバリチェック
- ホスト

ソース

CDService::get() in *ui/include/classes/api/services/CDService.php*.

検出されたホスト

このクラスは、検出されたホストを操作するために設計されています。

オブジェクト参照:

- 検出されたホスト

利用可能なメソッド:

- dhost.get - 検出されたホストを取得します

Discovered host オブジェクト

次のオブジェクトは、dhostAPI に直接関連しています。

検出されたホスト

> **Note:**
> 検出されたホストは Zabbix サーバーによって作成され、API 経由で変更することはできません。

検出されたホストオブジェクトには、ネットワークディスカバリルールによって検出されたホストに関する情報が含まれます。次のプロパティがあります。

Property	Type	Description
dhostid	ID	検出されたホストの ID。
druleid	ID	ホストを検出したディスカバリルールの ID。
lastdown	timestamp	検出されたホストが最後にダウンした時刻。
lastup	timestamp	検出されたホストが最後にアップした時刻。
status	integer	検出されたホストがアップしているかダウンしているか。ホストは、少なくとも 1 つのアクティブな検出済みサービスがある場合にアップと見なされます。 取り得る値: 0 - ホストはアップ; 1 - ホストはダウン。

dhost.get

説明

`integer/array dhost.get(object parameters)`

このメソッドは、指定したパラメーターに応じた検出されたホストを取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
dhostids	ID/array	指定した ID を持つディスカバリされたホストのみを返します。
druleids	ID/array	指定したディスカバリルールによって作成されたディスカバリされたホストのみを返します。
dserviceids	ID/array	指定したサービスを実行しているディスカバリされたホストのみを返します。
selectDRules	query	ホストを検出したディスカバリルールの配列を含む`drules` プロパティを返します。
selectDServices	query	ホスト上で実行されているディスカバリされたサービスを含む`dservices` プロパティを返します。 `count` をサポートします。
limitSelects	integer	サブセレクトで返されるレコード数を制限します。 以下のサブセレクトに適用されます: `selectDServices` - 結果は `dserviceid` でソートされます。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `dhostid`, `druleid`。
countOutput	boolean	これらのパラメータはリファレンスコメント で説明されています。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ディスカバリルールによって検出されたホストの取得

ディスカバリルール"4" によって検出されたすべてのホストと実行中の発見されたサービスを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "dservices": [
                {
                    "dserviceid": "1",
                    "dhostid": "1",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697227",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.1",
                    "dns": "station.company.lan"
                }
            ],
            "dhostid": "1",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697227",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "2",
                    "dhostid": "2",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.4",
                    "dns": "john.company.lan"
                }
            ],
            "dhostid": "2",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "3",
                    "dhostid": "3",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.26",
                    "dns": "printer.company.lan"
                }
            ],
```

```
            "dhostid": "3",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        },
        {
            "dservices": [
                {
                    "dserviceid": "4",
                    "dhostid": "4",
                    "type": "4",
                    "key_": "",
                    "value": "",
                    "port": "80",
                    "status": "0",
                    "lastup": "1337697234",
                    "lastdown": "0",
                    "dcheckid": "5",
                    "ip": "192.168.1.7",
                    "dns": "mail.company.lan"
                }
            ],
            "dhostid": "4",
            "druleid": "4",
            "status": "0",
            "lastup": "1337697234",
            "lastdown": "0"
        }
    ],
    "id": 1
}
```


関連項目

- 検出されたサービス
- ディスカバリルール

ソース

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

正規表現

このクラスは、グローバル正規表現を扱うために設計されています。

オブジェクト参照:

- 正規表現
- 式

利用可能なメソッド:

- regexp.create - 新しい正規表現を作成
- regexp.delete - 正規表現を削除
- regexp.get - 正規表現を取得
- regexp.update - 正規表現を更新

Regular expression オブジェクト

次のオブジェクトは、regexpAPI に直接関連しています。

正規表現

グローバル正規表現オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
regexpid	ID	正規表現の ID。 Property behavior: - *read-only* - 更新操作で必須
name	string	正規表現の名前。 Property behavior: - 作成操作で必須
test_string	string	テスト文字列。

式

expressions オブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
expression	string	正規表現。 プロパティの動作: - 必須
expression_type	integer	正規表現のタイプ。 使用可能な値: 0 - 文字列を含む; 1 - 任意の文字列を含む; 2 - 文字列を含まない; 3 - 結果が TRUE; 4 - 結果が FALSE。 プロパティの動作: - 必須
exp_delimiter	string	式の区切り文字。 デフォルト値: ","。 使用可能な値: ","、"."、または "/"。 プロパティの動作: - expression_type が「任意の文字列を含む」に設定されている場合にサポート
case_sensitive	integer	大文字と小文字の区別。 デフォルト値: 0。 使用可能な値: 0 - 区別しない; 1 - 区別する。

regexp.create

説明

`object regexp.create(object/array regularExpressions)`

このメソッドでは、新しいグローバル正規表現を作成できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、ユーザーロール を参照してください。

パラメータ

(object/array) 作成する正規表現。

標準プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
expressions	array	Expressions オプション。 パラメータの動作: - 必須

戻り値

(object) regexpids プロパティの下で作成された正規表現の ID を含むオブジェクトを返します。

例

新しいグローバル正規表現の作成。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
      "name": "SNMP ディスカバリ用のストレージデバイス",
      "test_string": "/boot",
      "expressions": [
        {
          "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$",
          "expression_type": "4",
          "case_sensitive": "1"
        }
      ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

説明

`object regexp.delete(array regexpids)`

このメソッドは、グローバル正規表現を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する正規表現の ID。

戻り値

(object) regexpids プロパティの下にある削除された正規表現の ID を含むオブジェクトを返します。

例

複数のグローバル正規表現の削除

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::delete() in *ui/include/classes/api/services/CRegexp.php*.

regexp.get

説明

`integer/array regexp.get(object parameters)`

このメソッドでは、指定されたパラメータに従ってグローバル正規表現を取得できます。

> **Note:**
> このメソッドは *Super Admin* のみが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はUser roles を参照してください。

パラメータ

(object) 必要な出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
regexids	ID/array	指定した ID を持つ正規表現のみを返します。
selectExpressions	query	`expressions` プロパティを返します。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `regexpid`, `name`。
countOutput	boolean	これらのパラメータについては、リファレンスコメント を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

パラメータ	型	説明
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

グローバル正規表現の取得

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid", "name"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Network interfaces for discovery",
      "expressions": [
        {
          "expression": "^Software Loopback Interface",
          "expression_type": "4"
        },
        {
          "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
          "expression_type": "4"
        },
        {
          "expression": "^NULL[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ll]o[0-9.]*$",
          "expression_type": "4"
        },
```

```
        {
          "expression": "^[Ss]ystem$",
          "expression_type": "4"
        },
        {
          "expression": "^Nu[0-9.]*$",
          "expression_type": "4"
        }
      ]
    }
  },
  "id": 1
}
```

ソース

CRegexp::get() in *ui/include/classes/api/services/CRegexp.php*.

regexp.update

説明

`object regexp.update(object/array regularExpressions)`

このメソッドでは、既存のグローバル正規表現を更新できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ使用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロール を参照してください。

パラメータ

(object/array) 更新する正規表現のプロパティ。

各オブジェクトには regexpid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

標準プロパティに加えて、このメソッドは以下のパラメータを受け付けます。

パラメータ	Type	説明
expressions	array	式のオプション。

戻り値

(object) regexpids プロパティの下にある更新された正規表現の ID を含むオブジェクトを返します。

例

ファイルシステムディスカバリ用のグローバル正規表現の更新。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
```

```
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        }
      ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

ソース

CRegexp::update() in *ui/include/classes/api/services/CRegexp.php*.

監査ログ

このクラスは監査ログを操作するために設計されています。

オブジェクト参照:

- 監査ログ

利用可能なメソッド:

- auditlog.get - 監査ログのレコードを取得します

Audit log オブジェクト

次のオブジェクトは、auditlogAPI に直接関連しています。

監査ログ

監査ログオブジェクトには、ユーザーの操作に関する情報が含まれます。
このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
auditid	ID	監査ログエントリの ID。CUID アルゴリズムを使用して生成されます。
userid	ID	監査ログエントリ作成者の userid。
username	string	監査ログエントリ作成者のユーザー名。
clock	timestamp	監査ログエントリの作成タイムスタンプ。
ip	string	監査ログエントリ作成者の IP アドレス。

Property	Type	Description
action	integer	監査ログエントリの操作。 指定可能な値: 0 - 追加; 1 - 更新; 2 - 削除; 4 - ログアウト; 7 - 実行; 8 - ログイン; 9 - ログイン失敗; 10 - 履歴のクリア; 11 - 設定のリフレッシュ; 12 - プッシュ。

Property	Type	Description
resourcetype	integer	監査ログエントリのリソースタイプ。 指定可能な値: 0 - ユーザー; 3 - メディアタイプ; 4 - ホスト; 5 - アクション; 6 - グラフ; 11 - ユーザーグループ; 13 - トリガー; 14 - ホストグループ; 15 - アイテム; 16 - 画像; 17 - 値のマッピング; 18 - サービス; 19 - マップ; 22 - Web シナリオ; 23 - ディスカバリルール; 25 - スクリプト; 26 - プロキシ; 27 - メンテナンス; 28 - 正規表現; 29 - マクロ; 30 - テンプレート; 31 - トリガープロトタイプ; 32 - アイコンのマッピング; 33 - ダッシュボード; 34 - イベント相関関係; 35 - グラフプロトタイプ; 36 - アイテムプロトタイプ; 37 - ホストプロトタイプ; 38 - 自動登録; 39 - モジュール; 40 - 設定; 41 - ハウスキーピング; 42 - 認証; 43 - テンプレートダッシュボード; 44 - ユーザーロール; 45 - API トークン; 46 - 定期レポート; 47 - 高可用性ノード; 48 - SLA; 49 - ユーザーディレクトリ; 50 - テンプレートグループ; 51 - コネクタ; 52 - LLD ルール; 53 - 履歴; 54 - 多要素認証; 55 - プロキシグループ; 56 - LLD ルールプロトタイプ。
resourceid	ID	監査ログエントリのリソース識別子。
resource_cuid	ID	CUID アルゴリズムを使用して生成された、監査ログエントリのリソース一意識別子。
resourcename	string	監査ログエントリのリソースの人間が判読可能な名前。
recordsetid	ID	監査ログエントリのレコードセット ID。同じ操作中に作成された監査ログレコードは、同じレコードセット ID を持ちます。CUID アルゴリズムを使用して生成されます。

Property	Type	Description
details	text	監査ログエントリの詳細。詳細は JSON オブジェクトとして保存され、各プロパティ名は変更が発生したプロパティまたはネストされたオブジェクトへのパスを表し、各値にはそのプロパティまたはネストされたオブジェクトにおける変更内容のデータが（配列形式で）含まれます。 指定可能な値の形式: ["add"] - ネストされたオブジェクトが追加された; ["add", "<value>"] - 追加されたオブジェクトのプロパティが <value> に等しい; ["update"] - ネストされたオブジェクトが更新された; ["update", "<new value>", "<old value>"] - 更新されたオブジェクトのプロパティが <old value> から <new value> に変更された; ["delete"] - ネストされたオブジェクトが削除された。

auditlog.get

説明

`integer/array auditlog.get(object parameters)`

このメソッドでは、指定されたパラメータに従って監査ログのレコードを取得できます。

Note:
このメソッドは、*Super admin* ユーザータイプでのみ利用できます。このメソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細はユーザーロールを参照してください。

パラメータ

(`object`) 必要な出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	Type	説明
auditids	ID/array	指定した ID の監査ログのみを返します。
userids	ID/array	指定したユーザーによって作成された監査ログのみを返します。
time_from	timestamp	指定した時刻以降、またはその時刻に作成された監査ログエントリのみを返します。
time_till	timestamp	指定した時刻以前、またはその時刻に作成された監査ログエントリのみを返します。
sortfield	string/array	指定したプロパティで結果を並べ替えます。 指定可能な値: `auditid`, `userid`, `clock`。 これらのパラメータについては、リファレンス解説を参照してください。
countOutput	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(`integer/array`) 以下のいずれかを返します。

- オブジェクトの配列
- `countOutput` パラメータが使用されている場合は、取得したオブジェクト数

例

監査ログの取得

最新の監査ログレコードを 2 件取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "auditid": "cksstgfam0001yhdcc41y20q2",
            "userid": "1",
            "username": "Admin",
            "clock": "1629975715",
            "ip": "127.0.0.1",
            "action": "1",
            "resourcetype": "0",
            "resourceid": "0",
            "resource_cuid": "0",
            "resourcename": "Jim",
            "recordsetid": "cksstgfal0000yhdcso67ondl",
            "details": "{\"user.name\":[\"update\",\"Jim\",\"\"],\"user.medias[37]\":[\"add\"],\"user.medi
        },
        {
            "auditid": "ckssofl0p0001yhdcqxclsg8r",
            "userid": "1",
            "username": "Admin",
            "clock": "1629967278",
            "ip": "127.0.0.1",
            "action": "0",
            "resourcetype": "0",
            "resourceid": "20",
            "resource_cuid": "0",
            "resourcename": "John",
            "recordsetid": "ckssofl0p0000yhdcpxyo1jgo",
            "details": "{\"user.username\":[\"add\",\"John\"], \"user.userid:\":[\"add\",\"20\"],\"user.us
        }
    ],
    "id": 1
}
```

参照

- Audit log object

ソース

CAuditLog::get() in *ui/include/classes/api/services/CAuditLog.php*.

相関関係

このクラスは、相関関係を扱うために設計されています。

オブジェクト参照:

- 相関関係
- 相関関係の操作
- 相関関係フィルター
  - 相関関係フィルター条件

利用可能なメソッド:

- correlation.create - 新しい相関関係を作成
- correlation.delete - 相関関係を削除
- correlation.get - 相関関係を取得
- correlation.update - 相関関係を更新

Correlation オブジェクト

次のオブジェクトは、`correlation`API に直接関連しています。

相関関係

相関関係オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
correlationid	ID	相関関係の ID。 Property behavior: - *read-only* - 更新操作で *required*
name	string	相関関係の名前。 Property behavior: - 作成操作で *required*
description	string	相関関係の説明。
status	integer	相関関係が有効か無効か。 使用可能な値: 0 - *(default)* 有効; 1 - 無効。

相関関係の実行内容

相関関係の実行内容オブジェクトは、相関関係が実行されたときに実行される操作を定義します。以下のプロパティがあります。

Property	Type	Description
type	integer	操作の種類。 使用可能な値: 0 - 古いイベントをクローズ; 1 - 新しいイベントをクローズ。 プロパティの動作: - 必須

相関フィルター

相関フィルターオブジェクトは、設定された相関操作を実行するために満たす必要がある一連の条件を定義します。このオブジェクトには、以下のプロパティがあります。

Property	Type	Description
conditions	array	結果をフィルタリングするために使用するフィルター条件のセット。条件は、formula 内で配置された順序で並べ替えられます。 Property behavior: - 必須
evaltype	integer	フィルター条件の評価方法。 指定可能な値: 0 - And/Or; 1 - And; 2 - Or; 3 - カスタム式。 Property behavior: - 必須
eval_formula	string	フィルター条件の評価に使用される生成済みの式。この式には、各フィルター条件をその `formulaid` で参照する ID が含まれます。`eval_formula` の値は、カスタム式を持つフィルターでは `formula` の値と同じです。 Property behavior: - 読み取り専用
formula	string	カスタム式を持つフィルターの条件評価に使用する、ユーザー定義の式。この式には、各フィルター条件をその `formulaid` で参照する ID を含める必要があります。式で使用する ID は、フィルター条件で定義されたものと完全に一致している必要があります。未使用または省略された条件があってはなりません。 Property behavior: - `evaltype` が”custom expression” に設定されている場合は 必須

相関フィルター条件

相関フィルター条件オブジェクトは、相関操作を実行する前に確認する必要がある特定の条件を定義します。

Property	Type	Description
type	integer	条件のタイプ。 使用可能な値: 0 - 古いイベントタグ; 1 - 新しいイベントタグ; 2 - 新しいイベントのホストグループ; 3 - イベントタグのペア; 4 - 古いイベントタグ値; 5 - 新しいイベントタグ値。 Property behavior: - 必須
tag	string	イベントタグ（古いまたは新しい）。 Property behavior: - `type` が「古いイベントタグ」、「新しいイベントタグ」、「古いイベントタグ値」、または「新しいイベントタグ値」に設定されている場合は 必須
groupid	ID	ホストグループの ID。 Property behavior: - `type` が「新しいイベントのホストグループ」に設定されている場合は 必須

Property	Type	Description
oldtag	string	古いイベントタグ。 Property behavior: - type が「イベントタグのペア」に設定されている場合は 必須
newtag	string	古いイベントタグ。 Property behavior: - type が「イベントタグのペア」に設定されている場合は 必須
value	string	イベントタグ（古いまたは新しい）の値。 Property behavior: - type が「古いイベントタグ値」または「新しいイベントタグ値」に設定されている場合は 必須
formulaid	string	カスタム式から条件を参照するために使用される任意の一意な ID。使用できるのは大文字のみです。フィルター条件を変更する際には、この ID はユーザーが定義する必要がありますが、その後に条件を要求すると新たに生成されます。
operator	integer	条件演算子。 Property behavior: - type が「新しいイベントのホストグループ」、「古いイベントタグ値」、または「新しいイベントタグ値」に設定されている場合は 必須

Note:
さまざまな種類の式でフィルターを使用する方法をよりよく理解するには、correlation.get およびcorrelation.create メソッドのページにある例を参照してください。

以下の演算子と値が、各条件タイプでサポートされています。

Condition	Condition name	Supported operators	Expected value
2	ホストグループ	=, <>	ホストグループ ID。
4	古いイベントタグ値	=, <>, like, not like	string
5	新しいイベントタグ値	=, <>, like, not like	string

correlation.create

説明

`object correlation.create(object/array correlations)`

このメソッドは、新しい相関関係を作成することができます。

Note:
このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(`object/array`) 作成する相関関係。

このメソッドは、標準の相関関係プロパティに加えて、以下のパラメータを受け付けます。

パラメータ	型	説明
operations	array	相関関係に対して作成する相関関係の操作。 パラメータの動作: - 必須

パラメータ	型	説明
filter	object	相関関係用の相関関係フィルターオブジェクト。 パラメータの動作: - 必須

戻り値

(object) correlationids プロパティの下に作成された相関関係の ID を含むオブジェクトを返します。返す ID の順序は、渡された相関関係の順序と一致します。

例

新しいイベントタグ相関の作成

1 つの条件と 1 つの演算による評価方法 AND/OR を使用して相関関係を作成します。デフォルトでは相関関係は有効になります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new event tag correlation",
        "filter": {
            "evaltype": 0,
            "conditions": [
                {
                    "type": 1,
                    "tag": "ok"
                }
            ]
        },
        "operations": [
            {
                "type": 0
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

カスタム式フィルターの使用

カスタムフィルター条件を使用する相関関係を作成します。式 ID、”A” または”B” は任意に選択されています。条件タイプは、演算子”<>” を使用した” ホストグループ” になります。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.create",
    "params": {
        "name": "new host group correlation",
        "description": "a custom description",
        "status": 0,
```

```
        "filter": {
            "evaltype": 3,
            "formula": "A or B",
            "conditions": [
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "A"
                },
                {
                    "type": 2,
                    "operator": 1,
                    "formulaid": "B"
                }
            ]
        },
        "operations": [
            {
                "type": 1
            }
        ]
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

関連項目

- 相関フィルター
- 相関操作

ソース

CCorrelation::create() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

説明

`object correlation.delete(array correlationids)`

このメソッドは、相関関係を削除することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメーター

(array) 削除する相関関係の ID。

戻り値

(object) correlationids プロパティの下にある削除された相関関係の ID を含むオブジェクトを返します。

例

複数の相関関係の削除

2 つの相関関係を削除します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

ソース

CCorrelation::delete() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

説明

`integer/array correlation.get(object parameters)`

このメソッドは、指定したパラメーターに応じた相関関係を取得できます。

> **Note:**
> このメソッドは、すべてのタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object) 目的の出力を定義するパラメータです。

このメソッドは以下のパラメータをサポートしています。

パラメータ	型	説明
correlationids	ID/array	指定した ID を持つ相関関係のみを返します。
selectFilter	query	相関条件を含む`filter` プロパティを返します。
selectOperations	query	相関操作を含む`operations` プロパティを返します。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: `correlationid`, `name`, `status`。
countOutput	boolean	これらのパラメータについては、リファレンス解説 を参照してください。
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

パラメータ	型	説明
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメーターが使用されている場合、取得したオブジェクトの数

例

相関関係の取得

設定されたすべての相関を、相関条件および操作とともに取得します。フィルターは"and/or" 評価タイプを使用するため、formula プロパティは空で、eval_formula が自動的に生成されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "correlationid": "1",
            "name": "Correlation 1",
            "description": "",
            "status": "0",
            "filter": {
                "evaltype": "0",
                "formula": "",
                "conditions": [
                    {
                        "type": "3",
                        "oldtag": "error",
                        "newtag": "ok",
                        "formulaid": "A"
                    }
                ],
                "eval_formula": "A"
            },
            "operations": [
                {
                    "type": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

関連項目

- 相関フィルター
- 相関操作

ソース

CCorrelation::get() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

説明

`object correlation.update(object/array correlations)`

このメソッドは、既存の相関関係を更新することができます。

> **Note:**
> このメソッドは、*Super admin* タイプのユーザーのみ利用可能です。メソッドを呼び出す権限は、ユーザーの役割の設定で取り消すことができます。詳細はユーザーの役割を参照してください。

パラメータ

(object/array) 更新する相関関係のプロパティ。

各相関関係について correlationid プロパティを定義する必要があり、その他のプロパティはすべて任意です。渡されたプロパティのみが更新され、その他はすべて変更されません。

このメソッドは、標準の相関関係プロパティに加えて、以下のパラメータを受け付けます。

Parameter	Type	Description
filter	object	現在のフィルターを置き換えるための相関関係フィルターオブジェクト。
operations	array	既存の操作を置き換えるための相関関係操作。

戻り値

(object) correlationids'プロパティの下で更新された相関関係 ID を含むオブジェクトを返します。

例

相関関係の無効化

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

評価方法を維持したままで条件の置き換え

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

関連項目

- 相関フィルター
- 相関操作

ソース

CCorrelation::update() in *ui/include/classes/api/services/CCorrelation.php*.

障害

このクラスは障害を扱うために設計されています。

オブジェクト参照:

- Problem
  - メディアタイプ URL
- Problem tag

利用可能なメソッド:

- problem.get - 障害を取得します

Problem オブジェクト

次のオブジェクトは、`problem` API に直接関連しています。

障害

> **Note:**
> 障害は Zabbix サーバーによって作成され、API 経由では変更できません。

障害オブジェクトには、以下のプロパティがあります。

Property	Type	Description
eventid	ID	障害イベントの ID。
source	integer	障害イベントの種類。 使用可能な値: 0 - トリガーによって作成されたイベント; 3 - 内部イベント; 4 - サービスステータス更新時に作成されたイベント。
object	integer	障害イベントに関連するオブジェクトの種類。 `source` が「トリガーによって作成されたイベント」に設定されている場合の使用可能な値: 0 - トリガー。 `source` が「内部イベント」に設定されている場合の使用可能な値: 0 - トリガー; 4 - アイテム; 5 - LLD ルール。 `source` が「サービスステータス更新時に作成されたイベント」に設定されている場合の使用可能な値: 6 - サービス。
objectid	ID	関連オブジェクトの ID。
clock	timestamp	障害イベントが作成された時刻。
ns	integer	障害イベントが作成された時点のナノ秒。
r_eventid	ID	復旧イベントの ID。
r_clock	timestamp	復旧イベントが作成された時刻。
r_ns	integer	復旧イベントが作成された時点のナノ秒。
cause_eventid	ID	原因イベントの ID。
correlationid	ID	このイベントがグローバル相関ルールによって復旧された場合の、相関ルールの ID。
userid	ID	障害をクローズしたユーザーの ID（障害が手動でクローズされた場合）。
name	string	解決済みの障害名。
acknowledged	integer	障害の確認状態。 使用可能な値: 0 - 未確認; 1 - 確認済み。
severity	integer	障害の現在の深刻度。 使用可能な値: 0 - 未分類; 1 - 情報; 2 - 警告; 3 - 平均; 4 - 高; 5 - 致命的。
suppressed	integer	障害が抑制されているかどうか。 使用可能な値: 0 - 障害は通常状態; 1 - 障害は抑制されている。
opdata	string	展開済みマクロを含む運用データ。
urls	array	アクティブなメディアタイプ URL。

メディアタイプ URL

メディアタイプ URL オブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
name	string	メディアタイプで定義された URL 名。

プロパティ	Type	説明
url	string	メディアタイプで定義された URL 値。

結果には、有効なイベントメニューエントリを持つアクティブなメディアタイプのエントリのみが含まれます。プロパティで使用されているマクロは展開されますが、いずれかのプロパティに未展開のマクロが含まれている場合、両方のプロパティが結果から除外されます。サポートされているマクロについては、サポートされているマクロを参照してください。

障害タグ

障害タグオブジェクトには、以下のプロパティがあります。

プロパティ	Type	説明
tag	string	障害タグ名。
value	string	障害タグの値。

problem.get

説明

`integer/array problem.get(object parameters)`

このメソッドでは、指定されたパラメータに従って障害を取得できます。

このメソッドは、未解決の障害を取得するためのものです。指定した場合は、最近解決された障害を追加で取得することもできます。「最近」がどの程度前までを指すかを決定する期間は、*Administration* → *General* で定義されます。その期間より前に解決された障害は problem テーブルに保持されません。さらに過去に解決された障害を取得するには、`event.get` メソッドを使用してください。

Attention:
このメソッドは、これらの障害がまだハウスキーパーによって削除されていない場合、削除済みエンティティの障害を返すことがあります。

Note:
このメソッドは、あらゆるタイプのユーザーが利用できます。メソッドを呼び出す権限は、ユーザーロール設定で取り消すことができます。詳細については、User roles を参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートしています。

パラメータ	Type	説明
eventids	ID/array	指定した ID の障害のみを返します。
groupids	ID/array	指定したホストグループに属するオブジェクトによって作成された障害のみを返します。
hostids	ID/array	指定したホストに属するオブジェクトによって作成された障害のみを返します。
objectids	ID/array	指定したオブジェクトによって作成された障害のみを返します。
source	integer	指定したタイプの障害のみを返します。 サポートされているイベントタイプの一覧は、障害イベントオブジェクトのページを参照してください。 デフォルト: 0 - トリガーによって作成された障害。
object	integer	指定したタイプのオブジェクトによって作成された障害のみを返します。 サポートされているオブジェクトタイプの一覧は、障害イベントオブジェクトのページを参照してください。 デフォルト: 0 - トリガー。
acknowledged	boolean	`true` に設定した場合、確認済みの障害のみを返します。

パラメータ	Type	説明
action	integer	指定したイベント更新アクションが実行された障害のみを返します。複数のアクションを指定する場合は、許容されるビットマップ値の合計をビットマスクとして使用します（たとえば、確認とイベント抑制の場合は 34）。
action_userids	ID/array	障害イベント更新アクションを実行したユーザーの指定した ID に一致する障害のみを返します。
suppressed	boolean	`true` に設定した場合、抑制された障害のみを返します。
symptom	boolean	`true` に設定した場合、症状の障害イベントのみを返します。
severities	integer/array	指定したイベント深刻度の障害のみを返します。object が trigger の場合にのみ適用されます。
evaltype	integer	タグの評価方法。 指定可能な値: 0 - *(デフォルト)* And/Or; 2 - Or。
tags	array	指定したタグを持つ障害のみを返します。 形式: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`。 空の配列を指定すると、すべての障害を返します。 指定可能なoperatorの値: 0 - *(デフォルト)* Contains; 1 - Equals; 2 - Does not contain; 3 - Does not equal; 4 - Exists; 5 - Does not exist。
recent	boolean	`true` に設定した場合、最近復旧した障害のみを返します（`ok_period` に依存します）。
eventid_from	string	指定した ID 以上の ID を持つ障害のみを返します。
eventid_till	string	指定した ID 以下の ID を持つ障害のみを返します。
time_from	timestamp	指定した時刻以降に作成された障害のみを返します。
time_till	timestamp	指定した時刻以前に作成された障害のみを返します。
selectAcknowledges	query	障害更新を含む `acknowledges` プロパティを返します。障害更新は新しい順にソートされます。 障害更新オブジェクトは以下のプロパティを持ちます: `acknowledgeid` - (ID) 更新の ID; `userid` - (ID) イベントを更新したユーザーの ID; `eventid` - (ID) 更新されたイベントの ID; `clock` - (timestamp) イベントが更新された時刻; `message` - (string) メッセージのテキスト; `action` - (integer) 更新アクションのタイプ (`event.acknowledge`を参照); `old_severity` - (integer) この更新アクション前のイベント深刻度; `new_severity` - (integer) この更新アクション後のイベント深刻度; `suppress_until` - (timestamp) イベントが抑制される期限の時刻; `taskid` - (ID) 現在のイベントがランク変更中の場合のタスク ID。 `count` をサポートします。
selectTags	query	障害タグを含む`tags` プロパティを返します。出力形式: `[{"tag": "<tag>", "value": "<value>"}, ...]`。
selectSuppressionData	query	有効なメンテナンスおよび手動抑制の一覧を含む `suppression_data` プロパティを返します: `maintenanceid` - (ID) メンテナンスの ID; `userid` - (ID) 障害を抑制したユーザーの ID; `suppress_until` - (integer) 障害が抑制される期限の時刻。

パラメータ	Type	説明
filter	object	指定したフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 text データ型 のプロパティはサポートされません。
sortfield	string/array	指定したプロパティで結果をソートします。 指定可能な値: eventid。
countOutput	boolean	これらのパラメータはリファレンス解説で説明されています。
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

戻り値

(integer/array) 次のいずれかを返します:

- オブジェクトの配列
- countOutput パラメータが使用されている場合、取得されたオブジェクトの数

例

トリガーの障害イベントの取得

トリガー「15112」の最近のイベントを取得します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectTags": "extend",
        "selectSuppressionData": "extend",
        "objectids": "15112",
        "recent": true,
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
```

```
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "localhost 上の Zabbix エージェントに 5 分間到達できません",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0"
                }
            ],
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "id": 1
}
```


指定されたユーザーが確認済みの問題を取得しています

ID=10 のユーザーが確認済みの問題を取得しています

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "id": 1
}
```


関連項目

- アラート
- アイテム
- ホスト
- LLD ルール
- トリガー

ソース

CEvent::get() in *ui/include/classes/api/services/CProblem.php*.

高可用性ノード

このクラスは、高可用性クラスターの一部であるサーバーノード、またはスタンドアロンサーバーインスタンスで動作するように設計されています。

オブジェクト参照:

- 高可用性ノード

利用可能なメソッド:

- hanode.get - ノードを取得します

High availability node オブジェクト

次のオブジェクトは、Zabbix サーバーの HA (高可用性) クラスタの運用に関連しています。

高可用性ノード

> **Note:**
> ノードは Zabbix サーバーによって作成され、API 経由で変更することはできません。

高可用性ノードオブジェクトには、以下のプロパティがあります。

プロパティ	型	説明
ha_nodeid	ID	ノードの ID。
name	string	zabbix_server.conf の HANodeName 設定項目を使用してノードに割り当てられた名前。スタンドアロンモードで動作しているサーバーでは空です。
address	string	ノードの接続元となる IP または DNS 名。
port	integer	ノードが動作しているポート。
lastaccess	integer	ハートビート時刻、つまりノードから最後に更新があった時刻。UTC タイムスタンプ。
status	integer	ノードの状態。 使用可能な値: 0 - standby; 1 - 手動停止; 2 - 利用不可; 3 - active。

hanode.get

説明

`integer/array hanode.get(object parameters)`

このメソッドでは、指定されたパラメータに従って高可用性クラスターノードの一覧を取得できます。

> **Note:**
> このメソッドは、*Super admin* ユーザータイプでのみ利用できます。詳細は、ユーザーロールを参照してください。

パラメータ

(`object`) 目的の出力を定義するパラメータ。

このメソッドは以下のパラメータをサポートします。

パラメータ	Type	説明
ha_nodeids	ID/array	指定されたノード ID を持つノードのみを返します。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名、値が一致対象の単一の値または値の配列であるオブジェクトを受け付けます。 サポートされるプロパティ: `name`, `address`, `status`。
sortfield	string/array	指定されたプロパティで結果をソートします。 指定可能な値: `name`, `lastaccess`, `status`。
countOutput	flag	これらのパラメータについては、リファレンス解説で説明されています。
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

戻り値

(`integer/array`) 次のいずれかを返します:

- オブジェクトの配列
- `countOutput` パラメーターが使用されている場合、取得したオブジェクトの数

例

ステータス順に並べられたノードの一覧を取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

レスポンス:

```
{
  "jsonrpc": "2.0",
  "result": {
    "ckuo7i1nw000h0sajj3l3hh8u": {
      "ha_nodeid": "ckuo7i1nw000h0sajj3l3hh8u",
      "name": "node-active",
      "address": "192.168.1.13",
      "port": "10051",
      "lastaccess": "1635335704",
      "status": "3"
    },
    "ckuo7i1nw000e0sajwfttc1mp": {
      "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
      "name": "node6",
      "address": "192.168.1.10",
      "port": "10053",
      "lastaccess": "1635332902",
      "status": "2"
    },
    "ckuo7i1nv000c0sajz85xcrtt": {
      "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
      "name": "node4",
      "address": "192.168.1.8",
      "port": "10052",
      "lastaccess": "1635334214",
      "status": "1"
    },
    "ckuo7i1nv000a0saj1fcdkeu4": {
      "ha_nodeid": "ckuo7i1nv000a0saj1fcdkeu4",
      "name": "node2",
      "address": "192.168.1.6",
      "port": "10051",
      "lastaccess": "1635335705",
      "status": "0"
    }
  },
  "id": 1
}
```

ID を指定して特定のノードのリストを取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "name": "node4",
            "address": "192.168.1.8",
            "port": "10052",
            "lastaccess": "1635334214",
            "status": "1"
        },
        {
            "ha_nodeid": "ckuo7i1nw000e0sajwfttc1mp",
            "name": "node6",
            "address": "192.168.1.10",
            "port": "10053",
            "lastaccess": "1635332902",
            "status": "2"
        }
    ],
    "id": 1
}
```


停止したノードの一覧を取得する

リクエスト:


```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```


レスポンス:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "ha_nodeid": "ckuo7i1nw000g0sajjsjre7e3",
            "address": "192.168.1.12",
            "port": "10051"
        },
        {
            "ha_nodeid": "ckuo7i1nv000c0sajz85xcrtt",
            "address": "192.168.1.8",
            "port": "10052"
        },
        {
            "ha_nodeid": "ckuo7i1nv000d0sajd95y1b6x",
            "address": "192.168.1.9",
            "port": "10053"
        }
    ],
    "id": 1
}
```


スタンバイノードの数を取得する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

特定の IP アドレスにあるノードのステータスを確認する

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

ソース

CHaNode::get() in *ui/include/classes/api/services/CHaNode.php*.

付録 **1.** 参考文献の解説

表記法　データ型

Zabbix API は、以下のデータ型を入力としてサポートしています。

型	説明
ID	エンティティを参照するために使用される一意の識別子。
boolean	ブール値 (`true` または `false`)。
flag	渡された値は、`null` と等しくない場合に `true` とみなされ、そうでない場合は `false` とみなされます。
integer	整数。
float	浮動小数点数。
string	テキスト文字列。
text	より長いテキスト文字列。
timestamp	Unix タイムスタンプ。
array	順序付けられた値のシーケンス (単純な配列)。
object	連想配列。
query	返されるデータを定義する値。(特定のプロパティのみを返す) プロパティ名の配列として定義するか、定義済みの値のいずれかとして定義できます。 `extend` - すべてのオブジェクトプロパティを返します。 `count` - 取得したレコードの数を返します。特定のサブセレクトでのみサポートされます。

Attention:
Zabbix API は常に文字列または配列としてのみ値を返します。

プロパティの動作

一部のオブジェクトプロパティには、その動作を説明する短いラベルが付けられています。使用されるラベルは次のとおりです:

- 読み取り専用 - プロパティの値は自動的に設定され、特定の状況下でもユーザーが定義または変更することはできません (例: 継承オブジェクトまたは検出されたオブジェクトの場合は読み取り専用)。
- 書き込み専用 - プロパティの値は設定できますが、作成後にアクセスすることはできません。
- 定数 - プロパティの値はオブジェクトの作成時に設定できますが、作成後は変更できません。
- サポート - プロパティの値の設定は必須ではありませんが、特定の状況下では設定が許可されます (例: `type` が" シンプルチェック"、" 外部チェック"、"SSH エージェント"、"TELNET エージェント"、または"HTTP エージェント" に設定されている場合、サポート)。ただし、サポートのプロパティは、状況に関係なくデフォルト値に設定される場合があることに注意してください。
- 必須 - プロパティの値は、すべての操作 (取得操作を除く) または特定の条件 (例: 作成操作の場合は必須、`operationtype` が" グローバルスクリプト" に設定され、`opcommand_hst` が設定されていない場合は必須) で設定する必要があります。

Note:
更新操作の場合、プロパティは更新操作中に設定されると" 設定済み" とみなされます。

ラベルが付いていないプロパティはオプションです。

パラメーターの動作

一部の操作パラメーターには、操作における動作を説明する短いラベルが付けられています。使用されるラベルは次のとおりです:

- 読み取り専用 - パラメーターの値は自動的に設定され、特定の条件下でもユーザーが定義または変更することはできません (例: 継承されたオブジェクトまたは検出されたオブジェクトの場合は読み取り専用)。
- 書き込み専用 - パラメーターの値は設定できますが、設定後はアクセスできません。
- サポート - パラメーターの値の設定は必須ではありませんが、特定の条件下では設定が許可されます (例: Proxy オブジェクトの `operating_mode` が" パッシブプロキシ" に設定されている場合はサポート)。ただし、サポートされたパラメーターは、条件に関わらずデフォルト値に設定される場合があります。
- 必須 - パラメーターの値の設定は必須です。

ラベルが付いていないパラメーターはオプションです。

予約 **ID** 値**"0"**　予約 ID 値"0" は、要素をフィルタリングしたり、参照されているオブジェクトを削除したりするために使用できます。たとえば、ホストから参照されているプロキシを削除するには、proxyid を 0 に設定する必要があります ("proxyid": "0")。また、サーバーオプション proxyids によって監視されるホストをフィルタリングするには、proxyids を 0 に設定する必要があります ("proxyids": "0")。

一般的な**"get"** メソッドのパラメータ　以下のパラメータは、すべての get メソッドでサポートされています。

パラメータ	型	説明
countOutput	boolean	実際のデータの代わりに、結果内のレコード数を返します。

パラメータ	型	説明
editable	boolean	`true` に設定すると、ユーザーが書き込み権限を持つオブジェクトのみを返します。 デフォルト: `false`。
excludeSearch	boolean	`search` パラメータで指定された条件に一致しない結果を返します。
filter	object	指定されたフィルターに完全に一致する結果のみを返します。 キーがプロパティ名（例: `host.get` のホストオブジェクトのプロパティ、`item.get` のアイテムオブジェクトのプロパティなど）、値が一致対象となる単一の値または値の配列であるオブジェクトを受け付けます。 `text` データ型のプロパティはサポートしていません。 一部のメソッドでは、このパラメータに対して固有の機能があり、メソッドページで説明されています（例: host.get の `filter` パラメータはホストインターフェースのプロパティもサポートしています）。
limit	integer	返されるレコード数を制限します。
output	query	返されるオブジェクトのプロパティ。 オブジェクト ID（つまり `hostid`、`itemid` など）は、`output` パラメータで指定されていない場合でも、常にレスポンスに含まれることに注意してください。 デフォルト: `extend`。
preservekeys	boolean	結果の配列でキーとして ID を使用します。
search	object	指定されたパターンに一致する結果を返します（大文字と小文字を区別しません）。 キーがプロパティ名（例: `host.get` のホストオブジェクトのプロパティ、`item.get` のアイテムオブジェクトのプロパティなど）、値が検索対象の文字列であるオブジェクトを受け付けます。追加オプションが指定されていない場合、`LIKE "%…%"` 検索が実行されます。 `string` および `text` データ型のプロパティのみをサポートします。 一部のメソッドでは、このパラメータに対して固有の機能があり、メソッドページで説明されています（例: host.get の `search` パラメータはホストインターフェースのプロパティもサポートしています）。
searchByAny	boolean	`true` に設定すると、`filter` または `search` パラメータで指定されたすべての条件ではなく、いずれかの条件に一致する結果を返します。 デフォルト: `false`。
searchWildcardsEnabled	boolean	`true` に設定すると、`search` パラメータでワイルドカード文字として"*" を使用できるようになります。 デフォルト: `false`。
sortfield	string/array	指定されたプロパティで結果をソートします。ソートに使用できるプロパティの一覧については、各 API get メソッドの説明を参照してください。マクロはソート前に展開されません。 値が指定されていない場合、データはソートされずに返されます。
sortorder	string/array	ソート順。配列が渡された場合、各値は `sortfield` パラメータで指定された対応するプロパティに適用されます。 指定可能な値: ASC - *(デフォルト)* 昇順; DESC - 降順。
startSearch	boolean	`search` パラメータはフィールドの先頭を比較し、代わりに `LIKE "…%"` 検索を実行します。 `searchWildcardsEnabled` が `true` に設定されている場合は無視されます。

エンティティの生成元フラグ　get メソッドは、ローレベルディスカバリ（LLD ルール/LLD ルールのプロトタイプ、アイテム/アイテムのプロトタイプなど）に関連するエンティティに対して `flags` プロパティを返します。このプロパティは、検出済みエンティティでは編集に制限があるため、そのエンティティが検出されたものかどうかを示すのに役立ちます。

`flags` プロパティは、以下の値の組み合わせ（「+」演算）に基づく結果を返します。

Value	Description
0	基本エンティティ（アイテム、トリガー、グラフ、ホスト）
1	ローレベルディスカバリルール
2	任意のプロトタイプ（アイテムのプロトタイプ、トリガーのプロトタイプ、LLD ルールのプロトタイプなど）
4	検出済みエンティティ（検出されたアイテム、トリガー、グラフ、ホスト、LLD ルール）

`flags` プロパティが返す組み合わせ済みの値は、次のいずれかになります。

Value	Combination of	Description
0	0	通常のエンティティ（アイテム、トリガー、グラフ、ホスト）。
2	2	エンティティのプロトタイプ（アイテムのプロトタイプ、トリガーのプロトタイプなど）。
6	2+4	検出されたアイテム、トリガー、グラフ、ホスト（プロトタイプから変換されたもの）。
1	1	ローレベルディスカバリルール。
3	1+2	ローレベルディスカバリルールのプロトタイプ。
5	1+4	検出されたローレベルディスカバリルール（プロトタイプから変換されたもの）。
7	1+2+4	検出されたローレベルディスカバリルールのプロトタイプ。

例　ユーザー権限のチェック

ユーザーは、名前が”MySQL” または”Linux” で始まるホストに書き込む権限を持っていますか？

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
”result” が”0” の場合は、読み取り/書き込み権限を持つホストが存在しないことを意味します。

不一致を数える

名前に部分文字列”ubuntu” が含まれていないホストの数を数えます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
```

```
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

ワイルドカードを使用したホストの検索

名前に”server” という単語が含まれ、インターフェースポートが”10050” または”10071” であるホストを検索します。結果をホスト名で降順に並べ替え、5 つに限定します。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "50004",
            "host": "WebServer-Nginx",
```

```
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    ],
    "id": 1
}
```

”preservekeys” を含むワイルドカードを使用したホストの検索

前のリクエストにパラメーター”preservekeys” を追加すると、結果はオブジェクトの ID をキーとする連想配列として返されます。

リクエスト:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "host"],
        "selectInterfaces": ["port"],
        "filter": {
            "port": ["10050", "10071"]
        },
        "search": {
            "host": "*server*"
        },
        "searchWildcardsEnabled": true,
        "searchByAny": true,
        "sortfield": "host",
        "sortorder": "DESC",
        "limit": 5,
        "preservekeys": true
    },
    "id": 1
}
```

レスポンス:

```
{
    "jsonrpc": "2.0",
    "result": {
        "50003": {
            "hostid": "50003",
            "host": "WebServer-Tomcat02",
            "interfaces": [
```

```
                {
                    "port": "10071"
                }
            ]
        },
        "50005": {
            "hostid": "50005",
            "host": "WebServer-Tomcat01",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "50004": {
            "hostid": "50004",
            "host": "WebServer-Nginx",
            "interfaces": [
                {
                    "port": "10071"
                }
            ]
        },
        "99032": {
            "hostid": "99032",
            "host": "MySQL server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        },
        "99061": {
            "hostid": "99061",
            "host": "Linux server 01",
            "interfaces": [
                {
                    "port": "10050"
                }
            ]
        }
    },
    "id": 1
}
```


付録 **2. 7.2** から **7.4** での変更点

後方互換性のない変更点　マップ

ZBXNEXT-8784 マップリンクトリガーオブジェクトから、プロパティ `linktriggerid` および `linkid` を削除しました。

メディアタイプ

ZBX-26258 `mediatype.get`: `selectUsers` パラメータの厳密なバリデーションを追加しました。

user

ZBX-26258 `user.get`: メソッドパラメータの厳密な検証を追加しました。
ZBX-26258 `user.get`: プロパティ `userid` は、明示的に要求された場合、または `output` が"extend" に設定されている場合にのみ返されるようになりました。

ユーザーグループ

ZBX-26258 usergroup.get: mfaids、selectUsers パラメータの厳密な検証を追加しました。

その他の変更点およびバグフィックス　ダッシュボード

ZBXNEXT-8919 svggraph およびpiechart ウィジェットに、新しいダッシュボードウィジェットフィールド ds.0.color_palette を追加しました。
ZBXNEXT-9087 新しいダッシュボードウィジェットタイプitemcard を追加しました。

discoveryruleprototype

ZBXNEXT-1527 新しい LLD ルールプロトタイプ API が追加され、discoveryruleprototype.create、discoveryruleprototype.update、discoveryruleprototype.get、discoveryruleprototype.delete メソッドが利用可能になりました。

グラフ

ZBXNEXT-1527 graph.get メソッドに新しいパラメータ selectDiscoveryData を追加しました。selectGraphDiscovery パラメータは非推奨になりました。

graphprototype

ZBXNEXT-1527 graphprototype.get メソッドに、新しいパラメータ selectDiscoveryRulePrototype および selectDiscoveryData を追加しました。

ホスト

ZBXNEXT-1527 host.get メソッドに新しいパラメータ selectDiscoveryData および selectDiscoveryRules を追加しました。selectHostDiscovery および selectDiscoveries パラメータは非推奨になりました。

hostgroup

ZBXNEXT-1527 hostgroup.get メソッドに新しいパラメータ selectDiscoveryData を追加し、selectGroupDiscoveries パラメータは非推奨となりました。

hostprototype

ZBXNEXT-1527 hostprototype.get メソッドに新しいパラメータ selectDiscoveryRulePrototype および selectDiscoveryData を追加しました。

アイテム

ZBXNEXT-1527 item.get メソッドに新しいパラメータ selectDiscoveryData を追加しました。selectItemDiscovery パラメータは非推奨になりました。

アイテムプロトタイプ

ZBXNEXT-1527 itemprototype.get メソッドに新しいパラメータ selectDiscoveryRulePrototype および selectDiscoveryData を追加しました。

マップ

ZBXNEXT-875 マップ要素 オブジェクトにプロパティ zindex を追加しました。
ZBXNEXT-2122 ロール API のaction オブジェクトにプロパティ edit_own_media および edit_user_media を追加しました。
ZBXNEXT-8784 マップ オブジェクトにプロパティ background_scale、show_element_label、show_link_label を追加しました。
ZBXNEXT-8784 マップ要素 オブジェクトにプロパティ show_label を追加しました。
ZBXNEXT-8784 マップリンク オブジェクトにプロパティ show_label、indicator_type、itemid、thresholds、highlights を追加しました。
ZBXNEXT-8784 マップリンクインジケーター オブジェクトを追加しました。

mediatype

ZBXNEXT-9805 mediatype オブジェクトの smtp_authentication プロパティに、オプション 2（「OAuth token」）を追加しました。
ZBXNEXT-9805 mediatype オブジェクトに、プロパティ redirection_url、client_id、client_secret、authorization_url、token_url、tokens_status、access_token、access_token_updated、access_expires_in、refresh_token を追加しました。

テンプレート

ZBXNEXT-9872 新しいプロパティ readme および wizard_ready を追加しました。
ZBXNEXT-1527 template.get メソッドに新しいパラメータ selectDiscoveryRules を追加しました。selectDiscoveries パラメータは非推奨になりました。

templatedashboard

ZBXNEXT-9087 新しいテンプレートダッシュボードウィジェットタイプitemcardを追加しました。

トリガー

ZBXNEXT-1527 trigger.get メソッドに新しいパラメータ selectDiscoveryData を追加し、selectTriggerDiscovery パラメータは非推奨になりました。

triggerprototype

ZBXNEXT-1527 triggerprototype.get メソッドに新しいパラメータ selectDiscoveryRulePrototype および selectDiscoveryData を追加しました。

ユーザーマクロ

ZBXNEXT-9872 Host macro object に新しいプロパティ config を追加しました。

付録 **3. 7.4** での変更点

7.4.3 ダッシュボード

ZBXNEXT-10143 problems ウィジェットに、新しいダッシュボードウィジェットフィールド highlight_row を追加しました。

タスク

ZBX-26202 task.create: *Admin* および *User* タイプのユーザーが、' 今すぐ実行' タスクを作成できるようになりました。

21. 拡張機能

概要　Zabbix は多くの機能を提供していますが、追加機能の余地は常にあります。
拡張機能は、ソースコードを変更することなく、Zabbix の監視機能を変更および強化するための便利な方法です。

組み込みの拡張オプション（トラッパーアイテム、ユーザーパラメータなど）を使用するか、カスタム拡張機能（ロード可能モジュール、プラグインなど）を使用または作成することで、Zabbix の機能を拡張できます。

このセクションでは、Zabbix を拡張するためのすべてのオプションへの参照を含む概要を説明します。

カスタムコマンドによるデータ収集　トラッパーアイテム

トラッパーアイテム は、データを問い合わせて取得するのではなく、受信したデータを受け付けるアイテムです。
トラッパーアイテムは、特定のデータを Zabbix サーバーまたはプロキシに送信する場合に便利です。たとえば、長時間実行されるユーザースクリプトにおける定期的な可用性データやパフォーマンスデータの送信に使用できます。
Zabbix サーバーまたはプロキシへのデータ送信は、Zabbix sender ユーティリティまたは Zabbix sender のプロトコルを使用して可能です。
Zabbix サーバーへのデータ送信は、history.push API メソッドを使用して行うこともできます。

外部チェック

外部チェック は、実行可能ファイルを実行してチェックを行うためのアイテムです。たとえば、シェルスクリプト やバイナリを実行します。

外部チェックは Zabbix サーバーまたはプロキシ（ホストがプロキシによって監視されている場合）によって実行され、監視対象のホスト上で動作するエージェントは必要ありません。

ユーザーパラメータ

ユーザーパラメータ はユーザー定義キーに関連付けられたユーザー定義コマンドで、実行されると、Zabbix エージェントが稼働しているホストから必要なデータを取得できます。ユーザーパラメータは、Zabbix であらかじめ定義されていないエージェントまたはエージェント 2 のアイテムを設定する際に役立ちます。

system.run[] Zabbix エージェントアイテム

system.run[] Zabbix エージェントアイテム は、ユーザー定義コマンド用のアイテムです（事前定義されたキー system.run[] に関連付けられています。たとえば system.run[myscript.sh]）。このコマンドは、Zabbix エージェントが実行されているホスト上で実行できます。

注: system.run[] アイテムはデフォルトで無効になっており、使用する場合は、有効化（許可）し、Zabbix エージェントまたは agent 2 の設定ファイル（AllowKey 設定パラメータ）で定義する必要があります。

Attention:
外部チェック、ユーザーパラメータ、system.run[] Zabbix エージェントアイテムなどのアイテム内のユーザー定義コマンドは、Zabbix コンポーネントの実行に使用される OS ユーザーによって実行されます。これらのコマンドを実行するには、このユーザーに必要な権限が付与されている必要があります。

HTTP エージェントアイテム

HTTP エージェントアイテムは、HTTP/HTTPS 経由でデータリクエストを実行するためのアイテムです。
HTTP エージェントアイテムは、HTTP エンドポイントにリクエストを送信して、*Elasticsearch* や *OpenWeatherMap* などのサービスからデータを取得したり、Zabbix API のステータスや Apache または Nginx Web サーバーのステータスを確認したりするのに役立ちます。
HTTP エージェントアイテム（トラップを有効にした場合）は、トラッパーアイテムとして機能することもできます。

スクリプトアイテム

スクリプトアイテム は、HTTP/HTTPS 経由でデータを取得するユーザー定義の JavaScript コードを実行するためのアイテムです。スクリプトアイテムは、HTTP エージェントアイテムで提供される機能だけでは不十分な場合に役立ちます。たとえば、複数の手順や複雑なロジックを必要とする高度なデータ収集シナリオでは、スクリプトアイテムを設定して HTTP 呼び出しを行い、受信したデータを処理し、その後、変換した値を 2 回目の HTTP 呼び出しに渡すことができます。

Note:
HTTP エージェントアイテムおよびスクリプトアイテムは、Zabbix サーバーおよびプロキシでサポートされており、監視対象のホスト上でエージェントを実行する必要はありません。

高度な拡張機能　ロード可能モジュール

C で記述されたロード可能モジュールは、UNIX プラットフォーム上の Zabbix コンポーネント（サーバー、プロキシ、エージェント）の機能を拡張するための、柔軟性が高く高性能な選択肢です。ロード可能モジュールは基本的に、Zabbix デーモンによって使用され、起動時に読み込まれる共有ライブラリです。ライブラリには特定の関数が含まれている必要があり、それにより Zabbix プロセスは、そのファイルが実際に読み込み可能で利用できるモジュールであることを判別できます。

ロード可能モジュールには多くの利点があります。たとえば、新しいメトリクスを追加したり、その他の任意のロジックを実装したりできること（例: Zabbix の履歴データのエクスポート）、高いパフォーマンス、さらにそれらが提供する機能を開発、利用、共有できることです。これにより、問題のない保守に貢献し、Zabbix コアコードベースとは独立して、新機能をより容易に提供できるようになります。

ロード可能モジュールは、特に複雑な監視構成で有用です。組み込みシステムを監視する場合、監視対象パラメータの数が多い場合、または複雑なロジックや長い起動時間を伴う重いスクリプトを使用する場合、ユーザーパラメータ、system.run[] Zabbix エージェントのアイテム、外部チェックなどの拡張機能は、パフォーマンスに影響を与えます。ロード可能モジュールは、パフォーマンスを犠牲にすることなく Zabbix の機能を拡張する方法を提供します。

プラグイン

プラグインは、Zabbix agent 2 の監視機能を拡張します。
Go プログラミング言語で記述されており、ロード可能モジュール（C で記述）に代わる選択肢を提供します。

プラグインは、構造を定義し、1 つまたは複数のプラグインインターフェース（*Exporter*、*Collector*、*Configurator*、*Runner*、*Watcher*）を実装する Go パッケージです。
Zabbix agent 2 では、次の 2 種類のプラグインがサポートされています。

- 組み込みプラグイン（Zabbix 4.4.0 以降でサポート）
- ロード可能プラグイン（Zabbix 6.0.0 以降でサポート）

Zabbix が提供するプラグインの一覧を参照してください。

独自のプラグインを作成するための手順やチュートリアルについては、Developer centerを参照してください。

アラートのカスタマイズ　Webhook

webhook は Zabbix のメディアタイプの 1 つであり、ヘルプデスクシステム、チャット、メッセンジャーなどの外部ソフトウェアへ Zabbix のアラート機能を拡張するための手段を提供します。スクリプトアイテムと同様に、webhook はカスタム JavaScript コードを使用して HTTP 呼び出しを行うのに便利で、たとえば Microsoft Teams、Discord、Jira などのさまざまなプラットフォームに通知を送信できます。また、何らかのデータ（たとえば、作成されたヘルプデスクチケットに関する情報）を返し、それを Zabbix 内に表示することも可能です。

既存の webhook は、Zabbix のGit リポジトリ で利用できます。カスタム webhook の開発については、Webhook development guidelinesを参照してください。

アラートスクリプト

アラートスクリプト は、Zabbix のメディアタイプ の一種であり、Zabbix アラートを処理するための代替手段（スクリプト）を作成するオプションを提供します。アラートスクリプトは、Zabbix でアラートを送信するための既存のメディアタイプに満足できない場合に役立ちます。

Web インターフェースのカスタマイズ　カスタムテーマ

カスタムテーマを使用して、Zabbix Web インターフェースの外観を変更できます。独自のテーマの作成と適用については、手順を参照してください。

Web インターフェースモジュール

Web インターフェースモジュール では、サードパーティ製モジュールを追加したり、独自に開発したりすることで、Zabbix Web インターフェースの機能を拡張できます。Web インターフェースモジュールを使用すると、新しいメニュー項目や、それに対応するビュー、アクションなどを追加できます。

グローバルスクリプト　グローバルスクリプト は、設定されたスコープとユーザー権限に応じて、監視対象上で（shell (/bin/sh) インタープリターにより）実行できる、ユーザー定義のコマンドセットです。グローバルスクリプトは、以下のアクション向けに設定できます。

- アクション実行内容
- 手動ホストアクション
- 手動イベントアクション

グローバルスクリプトは、多くの場面で役立ちます。例えば、アクションの実行内容または手動ホストアクション用に設定した場合、グローバルスクリプトを使用して、アプリケーション（Web サーバー、ミドルウェア、CRM など）の再起動や、ディスク容量の解放（古いファイルの削除、`/tmp` のクリーンアップなど）といったリモートコマンド を自動または手動で実行できます。また別の例として、手動イベントアクション用に設定した場合は、グローバルスクリプトを使用して外部システム内の障害チケットを管理できます。

グローバルスクリプトは、Zabbix サーバー、プロキシ、またはエージェントによって実行できます。

> **Attention:**
> ユーザー定義コマンドは、Zabbix コンポーネントの実行に使用される OS ユーザーとして実行されます。これらのコマンドを実行するには、このユーザーに必要な権限が付与されている必要があります。

Zabbix API　Zabbix API は、Zabbix Web インターフェースの一部である HTTP ベースの API です。Zabbix API を使用すると、次のいずれかの操作を実行できます。

- Zabbix の設定をプログラムから取得および変更する。
- Zabbix の設定をインポートおよびエクスポートする。
- Zabbix の履歴データおよびトレンドデータにアクセスする。
- アプリケーションが Zabbix と連携して動作するように設定する。
- Zabbix をサードパーティ製ソフトウェアと統合する。
- 定型作業を自動化する。

Zabbix API は多数のメソッドで構成されており、それらは名目上、個別の API にグループ化されています。各メソッドは特定のタスクを実行します。利用可能なメソッド、および Zabbix API が提供する機能の概要については、Zabbix API のメソッドリファレンスを参照してください。

1 ローダブルモジュール

概要

ロード可能モジュールは、Zabbix の機能を拡張するための、パフォーマンスを重視した選択肢を提供します。

Zabbix の機能は、たとえばuser parameters、external checks、およびsystem.run Zabbix エージェントのアイテムなど、さまざまな方法で拡張できます。これらは非常に有効に機能しますが、大きな欠点が 1 つあります。つまり、fork() です。Zabbix はユーザーメトリクスを処理するたびに新しいプロセスを fork する必要があり、これはパフォーマンスの観点から好ましくありません。通常は大きな問題ではありませんが、組み込みシステムの監視、大量の監視パラメータの利用、または複雑なロジックや長い起動時間を持つ重いスクリプトを使用する場合には、深刻な問題になる可能性があります。

ロード可能モジュールのサポートにより、パフォーマンスを損なうことなく、Zabbix エージェント、サーバー、プロキシを拡張する方法が提供されます。

ロード可能モジュールは基本的に、Zabbix デーモンによって使用され、起動時に読み込まれる共有ライブラリです。ライブラリには特定の関数が含まれている必要があり、それによって Zabbix プロセスは、そのファイルが実際に読み込み可能で利用可能なモジュールであることを検出できます。

ロード可能モジュールには多くの利点があります。優れたパフォーマンスと任意のロジックを実装できる能力は非常に重要ですが、おそらく最も重要な利点は、Zabbix モジュールを開発、利用、共有できることです。これにより、問題のない保守に貢献し、新しい機能をより容易に、かつ Zabbix コアコードベースとは独立して提供するのに役立ちます。

モジュールのライセンスおよびバイナリ形式での配布は、AGPL-3.0 ライセンスによって規定されます（モジュールは実行時に Zabbix とリンクし、Zabbix ヘッダを使用します。Zabbix 7.0 以降、Zabbix コード全体は AGPL-3.0 ライセンスの下で提供されています）。Zabbix はバイナリ互換性を保証しません。

モジュール API の安定性は、1 つの Zabbix LTS（Long Term Support）releaseサイクルの間は保証されます。Zabbix API の安定性は保証されません（技術的には、モジュールから Zabbix 内部関数を呼び出すことは可能ですが、そのようなモジュールが動作する保証はありません）。

モジュール API

共有ライブラリが Zabbix モジュールとして扱われるためには、いくつかの関数を実装してエクスポートする必要があります。現在、Zabbix モジュール API には 6 つの関数があり、そのうち 1 つだけが必須で、残りの 5 つは任意です。

必須インターフェース

唯一の必須関数は **zbx_module_api_version()** です。

```
int zbx_module_api_version(void);
```

この関数は、このモジュールで実装されている API バージョンを返す必要があります。モジュールをロードするには、このバージョンが Zabbix でサポートされているモジュール API バージョンと一致していなければなりません。
Zabbix でサポートされているモジュール API のバージョンは ZBX_MODULE_API_VERSION です。
したがって、この関数はこの定数を返す必要があります。
この目的で使用されていた古い定数 ZBX_MODULE_API_VERSION_ONE は、ソース互換性を維持するために現在は ZBX_MODULE_API_VERSION と等しいものとして定義されていますが、使用は推奨されません。

オプションのインターフェース

オプションの関数は **zbx_module_init()**、**zbx_module_item_list()**、**zbx_module_item_timeout()**、**zbx_module_history_write_cbs()**、**zbx_module_uninit()** です。

```
int zbx_module_init(void);
```

この関数は、モジュールに必要な初期化処理（存在する場合）を実行する必要があります。
成功した場合は、ZBX_MODULE_OK を返す必要があります。
それ以外の場合は、ZBX_MODULE_FAIL を返す必要があります。
後者の場合、Zabbix は起動しません。

```
ZBX_METRIC  *zbx_module_item_list(void);
```

この関数は、モジュールがサポートするアイテムの一覧を返す必要があります。
各アイテムは ZBX_METRIC 構造体で定義されます。詳細は以下のセクションを参照してください。
一覧は、”key” フィールドが NULL の ZBX_METRIC 構造体で終端されます。

```
void    zbx_module_item_timeout(int timeout);
```

モジュールが **zbx_module_item_list()** をエクスポートしている場合、この関数は、モジュールによって実装されたアイテムチェックが従うべき Zabbix 設定ファイル内のタイムアウト設定を指定するために Zabbix によって使用されます。
ここで、”timeout” パラメータは秒単位です。

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

この関数は、異なるデータ型の履歴をエクスポートするために Zabbix server が使用するコールバック関数を返す必要があります。
コールバック関数は ZBX_HISTORY_WRITE_CBS 構造体のフィールドとして提供され、モジュールが特定の型の履歴に関心がない場合、フィールドは NULL にできます。

```
int zbx_module_uninit(void);
```

この関数は、割り当てられたリソースの解放、ファイルディスクリプタのクローズなど、必要な終了処理（存在する場合）を実行する必要があります。

すべての関数は、モジュールがロードされる際に Zabbix の起動時に 1 回呼び出されます。ただし、zbx_module_uninit() は例外で、モジュールがアンロードされる際に Zabbix の停止時に 1 回呼び出されます。

アイテムの定義

各アイテムは、ZBX_METRIC 構造体で定義されます。

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

ここで、**key** はアイテムキー（例: "dummy.random"）、**flags** は CF_HAVEPARAMS または 0 のいずれかです（アイテムがパラメータを受け付けるかどうかによります）。**function** はアイテムを実装する C 関数（例: "zbx_module_dummy_random"）で、**test_param** は Zabbix エージェントが"-p" フラグ付きで起動されたときに使用されるパラメータリストです（例: "1,1000"、NULL も可）。定義の例は次のようになります。

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

アイテムを実装する各関数は、2 つのポインタパラメータを受け取る必要があります。1 つ目は AGENT_REQUEST 型、2 つ目は AGENT_RESULT 型です。

```
int zbx_module_dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    ...

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}
```

これらの関数は、アイテムの値を正常に取得できた場合は SYSINFO_RET_OK を返す必要があります。それ以外の場合は、SYSINFO_RET_FAIL を返す必要があります。AGENT_REQUEST から情報を取得する方法、および AGENT_RESULT に情報を設定する方法の詳細については、以下の"dummy" モジュールの例を参照してください。

履歴エクスポートコールバックの提供

Attention:
モジュール経由の履歴エクスポートは、Zabbix プロキシではサポートされなくなりました。

モジュールでは、型ごとに履歴データをエクスポートする関数を指定できます: Numeric (float)、Numeric (unsigned)、Character、Text、Log:

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CBS;
```

それぞれの関数は、引数として"history_num" 要素の"history" 配列を受け取る必要があります。エクスポートする履歴データの型に応じて、"history" はそれぞれ次の構造体の配列になります:

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    double      value;
}
ZBX_HISTORY_FLOAT;
```

```
typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    zbx_uint64_t    value;
}
ZBX_HISTORY_INTEGER;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_STRING;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
}
ZBX_HISTORY_TEXT;

typedef struct
{
    zbx_uint64_t    itemid;
    int     clock;
    int     ns;
    const char  *value;
    const char  *source;
    int     timestamp;
    int     logeventid;
    int     severity;
}
ZBX_HISTORY_LOG;
```

コールバックは、データが Zabbix データベースに書き込まれ、値キャッシュに保存された後、履歴同期処理の最後に、Zabbix サーバーの履歴同期プロセスによって使用されます。

Attention:
履歴エクスポートモジュールで内部エラーが発生した場合は、復旧するまで監視全体をブロックするのではなく、データを破棄して Zabbix サーバーが動作を継続できるようにモジュールを実装することを推奨します。

モジュールのビルド

現在、モジュールは Zabbix のソースツリー内でビルドすることを想定しています。これは、モジュール API が Zabbix のヘッダーで定義されている一部のデータ構造に依存しているためです。

ロード可能なモジュールにとって最も重要なヘッダーは **include/module.h** で、これらのデータ構造を定義しています。**include/module.h** が正しく動作するために必要なその他のシステムヘッダーは **stdlib.h** と **stdint.h** です。

この情報を踏まえると、モジュールをビルドする準備はすべて整っています。モジュールには **stdlib.h**、**stdint.h**、**module.h** をインクルードする必要があり、ビルドスクリプトではこれらのファイルがインクルードパスに含まれていることを確認する必要があります。詳細については、以下のサンプル「dummy」モジュールを参照してください。

もう 1 つ便利なヘッダーとして **include/zbxcommon.h** があります。これは **zabbix_log()** 関数を定義しており、ロギングやデバッグに使用できます。

設定パラメータ

Zabbix エージェント、サーバー、およびプロキシは、モジュールを扱うための 2 つのパラメータをサポートしています。

- LoadModulePath – ロード可能なモジュールの配置場所へのフルパス
- LoadModule – 起動時にロードするモジュール。モジュールは、LoadModulePath で指定されたディレクトリに配置されている必要があります。あるいは、モジュール名の前にパスを指定する必要があります。前に付けるパスが絶対パス（'/' で始まる）の場合、LoadModulePath は無視されます。複数の LoadModule パラメータを含めることができます。

たとえば、Zabbix エージェントを拡張するには、次のパラメータを追加できます。

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

エージェントの起動時に、mariadb.so、apache.so、kernel.so モジュールは /usr/local/lib/zabbix/agent ディレクトリからロードされ、dummy.so は /usr/local/lib/zabbix からロードされます。モジュールが存在しない場合、権限が正しくない場合、または共有ライブラリが Zabbix モジュールではない場合、エージェントは起動に失敗します。

Web インターフェースの設定

ロード可能なモジュールは、Zabbix エージェント、サーバー、プロキシでサポートされています。したがって、Zabbix Web インターフェースでのアイテムタイプは、モジュールがどこにロードされるかによって異なります。モジュールがエージェントにロードされる場合、アイテムタイプは「Zabbix agent」または「Zabbix agent (active)」である必要があります。モジュールがサーバーまたはプロキシにロードされる場合、アイテムタイプは「Simple check」である必要があります。

Zabbix モジュールを介した履歴のエクスポートには、Web インターフェースでの設定は必要ありません。モジュールがサーバーによって正常にロードされ、少なくとも 1 つの非 NULL コールバック関数を返す **zbx_module_history_write_cbs()** 関数を提供している場合、履歴のエクスポートは自動的に有効になります。

ダミーモジュール

Zabbix には、C 言語で記述されたサンプルモジュールが含まれています。このモジュールは src/modules/dummy にあります。

```
alex@alex:~trunk/src/modules/dummy$ ls -l
-rw-rw-r-- 1 alex alex 9019 Apr 24 17:54 dummy.c
-rw-rw-r-- 1 alex alex   67 Apr 24 17:54 Makefile
-rw-rw-r-- 1 alex alex  245 Apr 24 17:54 README
```

このモジュールには十分なドキュメントがあり、独自のモジュールを作成する際のテンプレートとして使用できます。

上記の説明どおり、Zabbix ソースツリーのルートで./configure を実行した後、**dummy.so** をビルドするには **make** を実行するだけです。

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"
```

```c
/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}
```

```c
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
```

```
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
```

```
 *             dummy_history_string_cb                                        *
 *             dummy_history_text_cb                                          *
 *             dummy_history_log_cb                                           *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
```

```
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

このモジュールは 3 つの新しいアイテムをエクスポートします。

- dummy.ping - 常に'1' を返します
- dummy.echo[param1] - 最初のパラメータをそのまま返します。たとえば、dummy.echo[ABC] は ABC を返します
- dummy.random[param1, param2] - param1-param2 の範囲内のランダムな数値を返します。たとえば、dummy.random[1,1000000]

制限事項

ロード可能モジュールのサポートは、Unix プラットフォームでのみ実装されています。
これは、Windows エージェントでは動作しないことを意味します。

場合によっては、モジュールが *zabbix_agentd.conf* からモジュール関連の設定パラメータを読み取る必要があることがあります。
現在、これはサポートされていません。
モジュールで何らかの設定パラメータを使用する必要がある場合は、モジュール固有の設定ファイルの解析を実装する必要があります。

2 プラグイン

概要

プラグインは、Zabbix エージェント 2 の監視機能を拡張します。
Go プログラミング言語で記述されており、ロード可能モジュール（C で記述）や、Zabbix の機能を拡張するためのその他の方法に代わる選択肢を提供します。

以下の機能は、エージェント 2 とそのプラグインに固有のものです。

- パッシブチェックとアクティブチェックの両方に対する、スケジュール済み間隔および柔軟な間隔
- スケジュールおよびタスクの同時実行性を考慮したタスクキュー管理
- プラグインレベルのタイムアウト
- 起動時の Zabbix エージェント 2 とそのプラグインの互換性チェック

Zabbix エージェント 2 がサポートするすべてのメトリクスは、プラグインによって収集されます。
このページでは、ユーザーの観点から見たプラグイン設定の基本原則を説明し、Zabbix が提供する組み込みプラグインとロード可能プラグインの両方を一覧表示します。

> **Note:**
> Zabbix 6.0 以降、プラグインをエージェント 2 に組み込む必要はなくなりました。
> ロード可能プラグインとして追加できるため、新しいメトリクス用プラグインの作成が簡単になります。
> 独自のプラグインを作成する手順については、Developer center および Example plugin for Zabbix agent 2 を参照してください。

プラグインの設定

このセクションでは、プラグイン設定の一般的な原則とベストプラクティスについて説明します。

すべてのプラグインは *Plugins.** パラメータを使用して設定します。これは、Zabbix agent 2 の設定ファイルの一部として指定することも、プラグイン独自の設定ファイルで指定することもできます。プラグインが個別の設定ファイルを使用する場合、このファイルへのパスは Zabbix agent 2 設定ファイルの Include パラメータで指定する必要があります。

一般的なプラグインパラメータは、次の構造になります。

Plugins.<PluginName>.<Parameter>=<Value>

さらに、2 つの特定のパラメータグループがあります。

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* - デフォルトのパラメータ値を定義するために使用します。
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* - 名前付きセッションを介して、異なる監視対象ごとに個別のパラメータセットを定義するために使用します。

すべてのパラメータ名は、次の要件に従う必要があります。

- プラグイン名は大文字で始めることを推奨します
- パラメータ名は大文字で始める必要があります
- 特殊文字は使用できません
- ネストの最大レベルに制限はありません
- パラメータ数に制限はありません

たとえば、Uptime プラグインに対してのみ、*Scheduling* 更新間隔を持たないアクティブチェックをエージェント再起動直後に実行するには、設定ファイルで `Plugins.Uptime.System.ForceActiveChecksOnStart=1` を設定します。同様に、CPU プラグインの同時チェックのカスタム上限を設定するには、設定ファイルで `Plugins.CPU.System.Capacity=N` を設定します。

デフォルト値

設定ファイルでは、接続関連パラメータ（URI、ユーザー名、パスワードなど）のデフォルト値を、次の形式で設定できます。

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

例えば、`Plugins.Mysql.Default.Username=zabbix`、`Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017` などです。

このようなパラメータの値がアイテムキーまたは名前付きセッション のパラメータで指定されていない場合、プラグインはデフォルト値を使用します。デフォルトパラメータも未定義の場合は、ハードコードされたデフォルト値が使用されます。

> **Note:**
> アイテムキーにパラメータがまったくない場合、Zabbix エージェント 2 は、デフォルトパラメータセクションで定義された値を使用してメトリクスの収集を試みます。

名前付きセッション

名前付きセッションは、プラグインパラメータの追加レベルを表し、監視対象の各インスタンスごとに個別の認証パラメータセットを指定するために使用できます。
各名前付きセッションパラメータは、次の構造にする必要があります。

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

セッション名は、URI、ユーザー名、パスワードを個別に指定する代わりに、connString アイテムキーのパラメータとして使用できます。

アイテムキーでは、最初のパラメータとして connString または URI のいずれかを指定できます。
最初のキーのパラメータがどのセッション名にも一致しない場合、それは URI として扱われます。
なお、アイテムキーで URI に埋め込まれた認証情報を渡すことはサポートされていません。代わりに、名前付きセッションパラメータを使用してください。

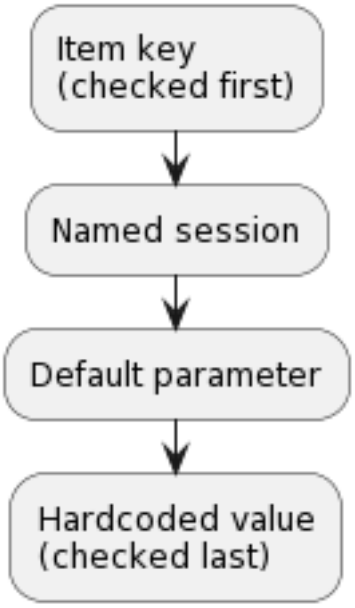
利用可能な名前付きセッションパラメータの一覧は、プラグインによって異なります。

アイテムキーのパラメータで新しい値を指定することにより、セッションパラメータを上書きできます（例を参照）。

名前付きセッションにパラメータが定義されていない場合、Zabbix エージェント 2 はデフォルトのプラグインパラメータで定義された値を使用します。

パラメータの優先順位

Zabbix エージェント 2 のプラグインは、接続関連のパラメータ値を次の順序で検索します。



1. 最初のアイテムキーのパラメータは、セッション名と比較されます。一致が見つからない場合、そのパラメータは実際の値として扱われます。この場合、手順 3 はスキップされます。一致が見つかった場合、パラメータ値（通常は URI）は、その名前付きセッションで定義されている必要があります。
2. 他のパラメータは、定義されていればアイテムキーから取得されます。
3. アイテムキーのパラメータ（たとえば password）が空の場合、プラグインは対応する名前付きセッションのパラメータを探します。
4. セッションパラメータも指定されていない場合は、対応するデフォルトパラメータで定義された値が使用されます。
5. それでも見つからない場合、プラグインはハードコードされたデフォルト値を使用します。

例 1

2 つのインスタンス「MySQL1」と「MySQL2」の監視。

設定パラメータ:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

この設定の結果、各セッション名はアイテムキー の connString として使用できます。例えば、`mysql.ping[MySQL1]` または `mysql.ping[MySQL2]` です。

例 2

アイテムキーで一部のパラメータを指定します。

設定パラメータ:

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

アイテムキー: `pgsql.ping[session1,new_username,,postgres]`

この設定の結果、エージェントは以下のパラメータを使用して PostgreSQL に接続します。

- セッションパラメータの URI: `192.0.2.234:5432`
- アイテムキーのユーザー名: `new_username`
- セッションパラメータのパスワード（アイテムキーで省略されているため）: `session_password`
- アイテムキーのデータベース名: `postgres`

例 3

デフォルトの設定パラメータを使用してメトリクスを収集します。

設定パラメータ:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

アイテムキー: `pgsql.ping[,,,postgres]`

この設定の結果、エージェントは次のパラメータを使用して PostgreSQL に接続します。

- デフォルト URI: `192.0.2.234:5432`
- デフォルトのユーザー名: `zabbix`
- デフォルトのパスワード: `password`
- アイテムキーのデータベース名: `postgres`

接続

一部のプラグインは、複数のインスタンスから同時にメトリクスを収集することをサポートしています。ローカルおよびリモートのインスタンスの両方を監視できます。TCP および Unix ソケット接続がサポートされています。

プラグインは、インスタンスへの接続を開いた状態に維持するよう設定することを推奨します。これにより、接続数が少なくなるため、ネットワーク輻輳、レイテンシー、CPU およびメモリ使用量を削減できます。これはクライアントライブラリによって処理されます。

Note:
未使用の接続を開いたままにしておく期間は、*Plugins.<PluginName>.KeepAlive* パラメータで指定できます。例: *Plugins.Memcached.KeepAlive*

組み込みプラグイン

Zabbix エージェント 2 の組み込みプラグインは、追加設定なしですぐに利用できます。追加情報については、プラグイン名をクリックしてプラグインリポジトリを参照してください。

プラグイン名	説明	サポートされているアイテムキー	コメント
Agent	使用中の Zabbix エージェントのメトリクス。	agent.hostname, agent.ping, agent.version	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Ceph	Ceph の監視。	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	
CPU	システム CPU の監視 (CPU/CPU コア数、検出された CPU、使用率)。	system.cpu.discovery, system.cpu.num, system.cpu.util	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Docker	Docker コンテナの監視。	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	以下も参照してください: 設定パラメータ
File	ファイルメトリクスの収集。	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Kernel	カーネルの監視。	kernel.maxfiles, kernel.maxproc	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。

プラグイン名	説明	サポートされているアイテムキー	コメント
Log	ログファイルの監視。	log, log.count, logrt, logrt.count	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。 以下も参照してください: プラグイン設定パラメータ（Unix/Windows）
Memcached	Memcached サーバーの監視。	memcached.ping, memcached.stats	
Modbus	Modbus データを読み取ります。	modbus.get	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
MQTT	MQTT トピックの公開値を受信します。	mqtt.get	MQTT ブローカーへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定してください。現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。
MySQL	MySQL およびその派生製品の監視。	mysql.custom.query, mysql.db.discovery, mysql.db.size, mysql.get_status_variables, mysql.ping, mysql.replication.discovery, mysql.replication.get_slave_status, mysql.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定してください。現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。
NetIf	ネットワークインターフェースの監視。	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Oracle	Oracle Database の監視。	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	プラグインを使用する前に、Oracle Instant Client をインストールしてください。
Proc	プロセス CPU 使用率。	proc.cpu.util	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。

プラグイン名	説明	サポートされているアイテムキー	コメント
Redis	Redis サーバーの監視。	redis.config, redis.info, redis.ping, redis.slowlog.count	Redis への暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッションまたはデフォルトパラメータとして指定してください。TLS パラメータはアイテムキーのパラメータとして渡すことはできません。不正または無効な TLS 設定により Zabbix エージェント 2 が起動できなくなる可能性があるため、TLS を有効にする前に証明書ファイル、権限、およびパスを確認してください。

Smart	S.M.A.R.T. の監視。	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	必要な smartctl の最小バージョンは 7.1 です。Zabbix エージェント 2 を実行するユーザーには、smartctl に対する sudo/root アクセス権が必要です。プラグインは次のコマンドのみを使用します:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			サポートされているkeysは、Linux/Windows 上の Zabbix エージェント 2 でのみ、パッシブチェックとアクティブチェックの両方で使用できます。
			以下も参照してください:
			設定パラメータ

SW	インストール済みパッケージの一覧表示。	system.sw.packages, system.sw.packages.get	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
Swap	スワップ領域のサイズ (バイト/パーセンテージ)。	system.swap.size	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
SystemRun	指定したコマンドを実行します。	system.run	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。 以下も参照してください: プラグイン設定パラメータ（Unix/Windows）
Systemd	systemd サービスの監視。	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP 接続の可用性チェック。	net.tcp.port	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
UDP	UDP サービスの可用性およびパフォーマンスの監視。	net.udp.service, net.udp.service.perf	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Uname	システムに関する情報の取得。	system.hostname, system.sw.arch, system.uname	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
Uptime	システム稼働時間メトリクスの収集。	system.uptime	サポートされているキーは、Zabbix エージェントのkeyと同じパラメータを持ちます。
VFSDev	VFS メトリクスの収集。	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
WebCertificate	TLS/SSL Web サイト証明書の監視。	web.certificate.get	
WebPage	Web ページの監視。	web.page.get, web.page.perf, web.page.regexp	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。

プラグイン名	説明	サポートされているアイテムキー	コメント
ZabbixAsync	非同期メトリクスの収集。	net.tcp.listen, net.udp.listen, sensor, system.boottime, system.cpu.intr, system.cpu.load, system.cpu.switches, system.hw.cpu, system.hw.macaddr, system.localtime, system.sw.os, system.swap.in, system.swap.out, vfs.fs.discovery	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
ZabbixStats	Zabbix サーバー/プロキシの内部メトリクス、またはキュー内の遅延アイテム数。	zabbix.stats	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。
ZabbixSync	同期メトリクスの収集。	net.dns, net.dns.record, net.tcp.service, net.tcp.service.perf, proc.mem, proc.num, system.hw.chassis, system.hw.devices, system.sw.packages, system.users.num, vfs.dir.count, vfs.dir.size, vfs.fs.get, vfs.fs.inode, vfs.fs.size, vm.memory.size.	サポートされているキーは、Zabbix エージェントのkeysと同じパラメータを持ちます。

ロード可能なプラグイン

Zabbix agent 2 のロード可能なプラグインは、すぐに利用できる状態では提供されておらず、別途インストールする必要があります。

- Linux では、パッケージを使用するか（例: Ubuntu で Ember+ を `apt install zabbix-agent2-plugin-ember-plus` でインストール）、プラグインをビルドできます
- Windows では、MSI からインストールするか、プラグインをビルドできます

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

追加情報を含むプラグイン README ファイルがあるプラグインリポジトリに移動するには、プラグイン名をクリックしてください。

Plugin name	Description	Supported item keys	Comments
Ember+	Ember+ の監視。	ember.get	Ember+ プラグイン設定パラメータも参照してください。

Plugin name	Description	Supported item keys	Comments
MongoDB	MongoDB サーバーおよびクラスター（ドキュメント指向の分散データベース）の監視。	mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッション パラメータとして指定してください。 現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。 MongoDB プラグイン設定パラメータも参照してください。
MSSQL	MSSQL データベースの監視。	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッション またはデフォルト パラメータとして指定してください。現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。 MSSQL プラグイン設定パラメータも参照してください。
NVIDIA GPU	NVIDIA GPU の監視。	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	NVIDIA GPU プラグイン設定パラメータも参照してください。

Plugin name	Description	Supported item keys	Comments
PostgreSQL	PostgreSQL およびその派生製品の監視。	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	データベースへの暗号化接続を設定するには、エージェント設定ファイルで TLS パラメータを名前付きセッション またはデフォルト パラメータとして指定してください。 現在、TLS パラメータはアイテムキーのパラメータとして渡すことはできません。 PostgreSQL プラグイン設定パラメータも参照してください。

Note:
ロード可能なプラグインは、次のオプションで起動すると:
 - `-V --version` - プラグインのバージョンおよびライセンス情報を表示します。
 - `-h --help` - ヘルプ情報を表示します。

3 フロントエンドモジュール

概要

サードパーティ製モジュールを追加したり独自のモジュールを開発したりすることで、Zabbix のソースコードを変更することなく、Zabbix Web インターフェースの機能を拡張できます。

モジュールのコードは、Zabbix のソースコードと同じ権限で実行されることに注意してください。これは次のことを意味します。

- サードパーティ製モジュールは有害である可能性があります。インストールするモジュールは信頼できるものでなければなりません。
- サードパーティ製モジュールのコード内のエラーにより、Web インターフェースがクラッシュする可能性があります。このような場合は、Web インターフェースからモジュールのコードを削除してください。Zabbix Web インターフェースを再読み込みすると、一部のモジュールが存在しないことを示す通知が表示されます。モジュール管理（管理 → 一般 → モジュール）に移動し、ディレクトリのスキャンを再度クリックして、存在しないモジュールをデータベースから削除してください。

インストール

必ず対象のモジュールのインストールマニュアルをお読みください。障害を容易に特定できるよう、新しいモジュールは 1 つずつインストールすることを推奨します。

モジュールをインストールする直前に、次の点を確認してください。

- 信頼できる提供元からモジュールをダウンロードしていることを確認してください。悪意のあるコードをインストールすると、データ損失などの結果を招く可能性があります
- 同じモジュールの異なるバージョン（同じ ID）は並行してインストールできますが、一度に有効化できるのは 1 つのバージョンのみです

モジュールをインストールする手順:

- Zabbix の Web インターフェースの `modules` フォルダ内で、モジュールを専用のフォルダに展開します

- モジュールのフォルダに、少なくとも manifest.json ファイルが含まれていることを確認します
- モジュール管理 に移動し、ディレクトリをスキャンボタンをクリックします
- 新しいモジュールが、バージョン、作成者、説明、ステータスとともに一覧に表示されます
- ステータスをクリックしてモジュールを有効化します

トラブルシューティング:

Problem	Solution
モジュールが一覧に表示されない	モジュールが一覧に表示されない場合は、modules/your-module/ に manifest.json が存在することを確認してください。存在しない場合は、誤ったディレクトリにモジュールを展開した可能性があります。存在する場合は、そのモジュールがお使いの Zabbix バージョンと互換性がない可能性があります。また、Web サーバーのユーザーが、モジュールフォルダおよびすべてのサブディレクトリに対して少なくとも読み取りおよび検索権限 (r-x) を持ち、内部のすべてのファイルに対して読み取り権限 (r--) を持っていることも確認してください。
Web インターフェースがクラッシュした	モジュールのコードが現在の Zabbix バージョンまたはサーバー設定と互換性がありません。モジュールファイルを削除し、Web インターフェースを再読み込みしてください。一部のモジュールが存在しないという通知が表示されます。モジュール管理 に移動し、再度 ディレクトリをスキャンをクリックして、存在しないモジュールをデータベースから削除してください。
同一の名前空間、*ID*、またはアクションに関するエラーメッセージが表示される	新しいモジュールが、すでに他の有効なモジュールによって登録されている名前空間、ID、またはアクションを登録しようとしました。新しいモジュールを有効化する前に、競合しているモジュール（エラーメッセージに記載）を無効化してください。
技術的なエラーメッセージが表示される	エラーをモジュールの開発者に報告してください。

モジュールの開発

カスタムモジュールの開発の詳細については、開発者センターを参照してください。

22. 付録

サイドバーから、参照したい項目を選択してください。

1 インストールとセットアップ

サイドバーから、参照したい項目を選択してください。

1 データベースの作成

概要

Zabbix データベースは、Zabbix サーバーまたはプロキシのインストール時に作成する必要があります。

このセクションでは、Zabbix データベースを作成するための手順を説明します。サポートされている各データベースごとに、個別の手順が用意されています。

Note:
最小限の権限を持つデータベースロール/ユーザーを作成してデータベースのセキュリティを向上させるには、サポートされている各データベース向けのデータベース作成のベストプラクティスを参照してください:

- MySQL/MariaDB
- PostgreSQL/TimescaleDB

安全な TLS 接続を設定するには、データベースへの安全な接続を参照してください。

UTF-8 は、Zabbix でサポートされている唯一のエンコーディングです。これは、セキュリティ上の欠陥なく動作することが確認されています。他のエンコーディングの一部を使用する場合、既知のセキュリティ上の問題があることに注意してください。UTF-8 への切り替えについては、Zabbix データベースの文字セットと照合順序の修復を参照してください。あわせて、utf8mb4 照合順序でのフィルタリングの制限も参照してください。

> **Note:**
> Zabbix Git repository からインストールする場合は、次の手順に進む前に以下のコマンドを実行する必要があります:
>

 `make dbschema`

MySQL/MariaDB

Zabbix サーバー/プロキシが MySQL データベースで正しく動作するために、文字セット utf8（別名 utf8mb3）および utf8mb4 がサポートされています（それぞれ utf8_bin および utf8mb4_bin 照合順序を使用）。新規インストールでは utf8mb4 の使用を推奨します。

決定論的トリガーは、スキーマのインポート時に作成する必要があります。MySQL および MariaDB では、バイナリログが有効で、スーパーユーザー権限がなく、かつ MySQL 設定ファイルで `log_bin_trust_function_creators = 1` が設定されていない場合、これには `GLOBAL log_bin_trust_function_creators = 1` の設定が必要です。

Zabbix パッケージからインストールする場合は、ご利用のプラットフォーム向けの手順に進んでください。

ソースから Zabbix をインストールする場合:

- データベースとユーザーを作成して設定します。

```
mysql -uroot -p<password>

mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user 'zabbix'@'localhost' identified by '<password>';
mysql> grant all privileges on zabbix.* to 'zabbix'@'localhost';
mysql> SET GLOBAL log_bin_trust_function_creators = 1;
mysql> quit;
```

- データをデータベースにインポートし、utf8mb4 文字セットをデフォルトとして設定します（Zabbix ソースのルートディレクトリにいることを前提とします）。Zabbix プロキシのデータベースでは、`schema.sql` のみをインポートしてください（images.sql および data.sql は不要です）。

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### Zabbix プロキシ用のデータベースを作成している場合はここで停止
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

スキーマのインポートが正常に完了した後は、`log_bin_trust_function_creators` を無効にできます。

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

データベースオブジェクトを作成する権限を持つデータベースユーザーが必要です。

Zabbix パッケージからインストールする場合は、ご利用のプラットフォーム向けの手順に進んでください。

ソースから Zabbix をインストールする場合:

- データベースユーザーを作成します。

次のシェルコマンドはユーザー zabbix を作成します。プロンプトが表示されたらパスワードを指定し、同じパスワードを再入力してください（なお、最初に sudo のパスワードを求められる場合があります）。

```
sudo -u postgres createuser --pwprompt zabbix
```

- データベースを作成します。

次のシェルコマンドは、先ほど作成したユーザーを所有者 (`-O zabbix`) として、データベース zabbix（最後のパラメータ）を作成します。

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- 初期スキーマとデータをインポートします（Zabbix ソースのルートディレクトリにいることを前提としています）。Zabbix プロキシのデータベースの場合は、schema.sql のみをインポートしてください（images.sql と data.sql は不要です）。

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### stop here if you are creating database for Zabbix proxy
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
上記のコマンドは、ほとんどの GNU/Linux インストール環境で動作する例として示しています。システム/データベースの設定方法によっては、たとえば次のように別のコマンドを使用できます:

 psql -U <username>

 データベースのセットアップで問題が発生した場合は、データベース管理者に相談してください。

TimescaleDB

TimescaleDB の作成および設定手順は、別のセクションで説明されています。

SQLite

SQLite の使用は、**Zabbix** プロキシでのみサポートされています！

データベースは、存在しない場合は自動的に作成されます。

インストールセクションに戻ります。

2 Zabbix データベースのキャラクターセットと照合の修正

MySQL/MariaDB

これまで、MySQL およびその派生製品では、標準 UTF8 の 4 バイト実装に対して、MySQL 独自の 3 バイト実装である utf8mb3 の別名として「utf8」が使用されていました。MySQL 8.0.28 および MariaDB 10.6.1 以降では、文字セット「utf8mb3」は非推奨となっており、将来的にはサポートが廃止される予定です。一方で、「utf8」は「utf8mb4」を参照するようになります。Zabbix 6.0 以降では、「utf8mb4」がサポートされています。将来的な問題を避けるため、「utf8mb4」を使用することを強く推奨します。「utf8mb4」へ切り替えるもう 1 つの利点は、Unicode の補助文字をサポートできることです。

Warning:
Zabbix 6.0 より前のバージョンは utf8mb4 を認識しないため、utf8mb4 への変換を実行する前に、必ず Zabbix サーバーおよび DB スキーマを先に 6.0.x 以降へアップグレードしてください。

1. データベースの文字セットと照合順序を確認します。

例:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2 _general_ci   |
+--------------------------+----------------------+
```

または:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

ご覧のとおり、ここでは文字セットが「utf8mb4」ではなく、照合順序も「utf8mb4_bin」ではないため、修正する必要があります。

2. Zabbix を停止します。

3. データベースのバックアップコピーを作成します。

4. データベースレベルで文字セットと照合順序を修正します。

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

修正後の値:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. テーブルおよびカラムレベルで文字セットと照合順序を修正するため、script を読み込みます。

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. スクリプトを実行します。

```
                SET @ZABBIX_DATABASE = '<your DB name>';
If MariaDB →    set innodb_strict_mode = OFF;
                CALL zbx_convert_utf8();
If MariaDB →    set innodb_strict_mode = ON;
                drop procedure zbx_convert_utf8;
```

「utf8mb4」は、ディスク容量をわずかに多く消費することが想定される点に注意してください。

7. エラーがなければ、修正済みのデータベースのバックアップコピーを作成することをお勧めします。

8. Zabbix を起動します。

3 データベースのアップグレードを主キーへ

概要

このセクションでは、既存のインストール環境において、テーブルを手動で主キー対応にアップグレードするための手順を説明します。

主キーへのアップグレードにより、データのインデックス作成およびアクセス方法が最適化され、クエリの高速化や容量の節約につながる場合があります。また、クラスタ構成におけるデータ管理と同期も改善されるため、スケーリングが容易になり、一部のサーバーに障害が発生した場合でもシステムの信頼性を維持しやすくなります。

> **Attention:**
> このページで提供される手順は上級ユーザー向けに設計されており、環境固有の設定に応じて調整が必要になる場合があります。主キーへのアップグレードは、時間がかかり、多くのリソースを消費する可能性があります。十分な空きディスク容量があることを確認してください。データベースのサイズや保存されているデータ量によっては、この処理に履歴テーブルで現在使用している容量の最大 2.5 倍の空き領域が必要になる場合があります。

Zabbix 6.0 以降、新規インストールではすべてのテーブルで主キーが使用されています。

データベースを主キー対応に自動アップグレードする機能はありませんが、既存のインストール環境では、Zabbix サーバーを 6.0 以降にアップグレードした後に手動でアップグレードできます。

> **Attention:**
> Zabbix 7.0 以降では、テーブルを主キー対応にアップグレードすると、テーブルは倍精度浮動小数点データ型を使用するようにもアップグレードされます。

 Zabbix 7.0（またはそれ以降）を使用している場合、テーブルはすでに倍精度浮動小数点を使用しています。ただし、このページの手順に従って、すでに倍精度浮動小数点を使用しているテーブルに影響を与えることなく、主キー対応へのアップグレードを実行できます。

 Zabbix 6.4（またはそれ以前）を使用している場合は、まずテーブルを倍精度浮動小数点対応にアップグレードすることを検討してください。詳細については、Zabbix 7.0 ドキュメントの拡張範囲の数値へのアップグレードを参照してください。

以下の手順を利用できます。

- MySQL
- PostgreSQL
- TimescaleDB

重要な注意事項

データベースのアップグレードを実行するには、次の手順に従ってください。

1. Zabbix サーバーを停止します。

アップグレード中は Zabbix サーバーを停止することを強く推奨します。ただし、どうしても必要な場合は、サーバーの稼働中にアップグレードを実行することもできます（MySQL、MariaDB、および TimescaleDB を使用しない PostgreSQL の場合のみ）。

2. データベースをバックアップします。
3. ご利用の Zabbix バージョンと互換性のある最新の zabbix-sql-scripts パッケージをインストールします（例: RHEL の場合: `dnf install zabbix-sql-scripts`）。
4. 使用しているデータベース用のスクリプトを実行します。
5. Zabbix サーバーを起動します。

Warning:
スクリプトはサーバーデータベースに対してのみ実行してください。このアップグレードによるメリットはプロキシにはありません。

データベースでパーティションを使用している場合は、DB 管理者または Zabbix サポートに問い合わせてください。

プライマリキーへのアップグレードが正常に完了した後、CSV ファイルは削除できます。

必要に応じて、Zabbix Web インターフェースをメンテナンスモードに切り替えることができます。

MySQL

セッションがドロップされないように、エクスポートとインポートは tmux/screen で実行する必要があります。

参照: 重要事項

mysqlsh を使用した MySQL 8.0+

この方法は稼働中の Zabbix サーバーでも使用できますが、アップグレード中はサーバーを停止することを推奨します。
MySQL Shell (*mysqlsh*) はインストールされており、DB に接続できる必要があります。

- root として（推奨）、または FILE 権限を持つ任意のユーザーとして MySQL コンソールにログインします。
- local_infile 変数を有効にして MySQL を起動します。
- `history_upgrade_prepare.sql` を実行して、古いテーブルの名前を変更し、新しいテーブルを作成します。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- データをエクスポートしてインポートします。

mysqlsh 経由で接続します。ソケット接続を使用する場合は、パスの指定が必要になることがあります。

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

次のコマンドで JavaScript モードに切り替えます。

```
\js
```

次に、以下のコードを実行します（CSVPATH は必要に応じて変更できます）。

```
CSVPATH="/var/lib/mysql-files";

util.exportTable("history_old", CSVPATH + "/history.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history.csv", {"dialect": "csv", "table": "history" });

util.exportTable("history_uint_old", CSVPATH + "/history_uint.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_uint.csv", {"dialect": "csv", "table": "history_uint" });

util.exportTable("history_str_old", CSVPATH + "/history_str.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_str.csv", {"dialect": "csv", "table": "history_str" });

util.exportTable("history_log_old", CSVPATH + "/history_log.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_log.csv", {"dialect": "csv", "table": "history_log" });

util.exportTable("history_text_old", CSVPATH + "/history_text.csv", { dialect: "csv" });
util.importTable(CSVPATH + "/history_text.csv", {"dialect": "csv", "table": "history_text" });
```

”JavaScript is not supported” というメッセージが表示される場合は、MySQL Shell のインストールに JS サポートが含まれていません。
その場合は、JavaScript モードが有効になるように、Oracle の公式 MySQL Shell package をインストールするか（またはソースからビルドしてください）。

- 古いテーブルを削除するには、移行後の手順 に従ってください。

mysqlsh を使用しない MariaDB/MySQL 8.0+

このアップグレード方法はより時間がかかるため、*mysqlsh* を使用したアップグレードが不可能な場合にのみ使用してください。

テーブルのアップグレード

- root（推奨）または FILE 権限を持つ任意のユーザーとして MySQL コンソールにログインします。
- 実行中の Zabbix サーバーで移行を実施する場合は、local_infile 変数を有効にして MySQL を起動します。
- history_upgrade_prepare.sql を実行して、古いテーブルの名前を変更し、新しいテーブルを作成します。

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

サーバー停止中の移行

移行中のタイムアウトを回避するため、データを移行する前に max_execution_time（MySQL の場合）または max_statement_time（MariaDB の場合）を無効にする必要があります。

MySQL の場合:

```
SET @@max_execution_time=0;
```

MariaDB の場合:

```
SET @@max_statement_time=0;
```

```
INSERT IGNORE INTO history SELECT * FROM history_old;
INSERT IGNORE INTO history_uint SELECT * FROM history_uint_old;
INSERT IGNORE INTO history_str SELECT * FROM history_str_old;
INSERT IGNORE INTO history_log SELECT * FROM history_log_old;
INSERT IGNORE INTO history_text SELECT * FROM history_text_old;
```

古いテーブルを削除するには、移行後の手順 に従ってください。

サーバー実行中の移行

インポート/エクスポートがどのパスで有効になっているかを確認します。

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

secure_file_priv の値がディレクトリへのパスである場合、そのディレクトリ内のファイルに対してエクスポート/インポートが実行されます。
この場合は、クエリ内のファイルパスをそれに応じて編集するか、アップグレード中のみ *secure_file_priv* の値を空文字列に設定してください。

secure_file_priv の値が空の場合、任意の場所からエクスポート/インポートを実行できます。

secure_file_priv の値が NULL の場合は、エクスポートされたテーブルデータを含むパスに設定してください（上記の例では'/var/lib/mysql-files/'）。

詳細については、MySQL documentation または MariaDB documentation を参照してください。

エクスポート中のタイムアウトを回避するため、データをエクスポートする前に max_execution_time（MySQL の場合）または max_statement_time（MariaDB の場合）を無効にする必要があります。

MySQL の場合:

```
SET @@max_execution_time=0;
```

MariaDB の場合:

```
SET @@max_statement_time=0;
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES TER
LOAD DATA INFILE '/var/lib/mysql-files/history.csv' IGNORE INTO TABLE history FIELDS TERMINATED BY ',' ESC

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_uint.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_uint.csv' IGNORE INTO TABLE history_uint FIELDS TERMINATED
```

```
SELECT * INTO OUTFILE '/var/lib/mysql-files/history_str.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_str.csv' IGNORE INTO TABLE history_str FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_log.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINES
LOAD DATA INFILE '/var/lib/mysql-files/history_log.csv' IGNORE INTO TABLE history_log FIELDS TERMINATED BY

SELECT * INTO OUTFILE '/var/lib/mysql-files/history_text.csv' FIELDS TERMINATED BY ',' ESCAPED BY '"' LINE
LOAD DATA INFILE '/var/lib/mysql-files/history_text.csv' IGNORE INTO TABLE history_text FIELDS TERMINATED
```

古いテーブルを削除するには、移行後の手順 に従ってください。

PostgreSQL

セッションが切断されないようにするため、エクスポートおよびインポートは tmux/screen 上で実行する必要があります。
TimescaleDB を使用するインストールでは、このセクションをスキップしてPostgreSQL + TimescaleDB に進んでください。

参照: 重要な注意事項

テーブルのアップグレード

- history_upgrade_prepare.sql を使用してテーブル名を変更します:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

サーバー停止中の移行

- 現在の履歴をエクスポートし、一時テーブルにインポートしてから、重複を無視しながら新しいテーブルにデータを挿入します:

```
INSERT INTO history SELECT * FROM history_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_uint SELECT * FROM history_uint_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_str SELECT * FROM history_str_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_log SELECT * FROM history_log_old ON CONFLICT (itemid,clock,ns) DO NOTHING;

INSERT INTO history_text SELECT * FROM history_text_old ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

INSERT のパフォーマンス向上に関するヒントも参照してください: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- 古いテーブルを削除するには、移行後の手順 に従ってください。

実行中のサーバーでの移行

- 現在の履歴をエクスポートし、一時テーブルにインポートしてから、重複を無視して新しいテーブルにデータを挿入します:

```
\copy history_old TO '/tmp/history.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    DOUBLE PRECISION DEFAULT '0.0000'          NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history FROM '/tmp/history.csv' DELIMITER ',' CSV
INSERT INTO history SELECT * FROM temp_history ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_uint_old TO '/tmp/history_uint.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_uint (
    itemid                   bigint                                    NOT NULL,
    clock                    integer         DEFAULT '0'               NOT NULL,
    value                    numeric(20)     DEFAULT '0'               NOT NULL,
    ns                       integer         DEFAULT '0'               NOT NULL
);
\copy temp_history_uint FROM '/tmp/history_uint.csv' DELIMITER ',' CSV
INSERT INTO history_uint SELECT * FROM temp_history_uint ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_str_old TO '/tmp/history_str.csv' DELIMITER ',' CSV
```

```
CREATE TEMP TABLE temp_history_str (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    value                   varchar(255)    DEFAULT ''              NOT NULL,
    ns                      integer         DEFAULT '0'             NOT NULL
);
\copy temp_history_str FROM '/tmp/history_str.csv' DELIMITER ',' CSV
INSERT INTO history_str (itemid,clock,value,ns) SELECT * FROM temp_history_str ON CONFLICT (itemid,clock,n

\copy history_log_old TO '/tmp/history_log.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_log (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    timestamp               integer         DEFAULT '0'             NOT NULL,
    source                  varchar(64)     DEFAULT ''              NOT NULL,
    severity                integer         DEFAULT '0'             NOT NULL,
    value                   text            DEFAULT ''              NOT NULL,
    logeventid              integer         DEFAULT '0'             NOT NULL,
    ns                      integer         DEFAULT '0'             NOT NULL
);
\copy temp_history_log FROM '/tmp/history_log.csv' DELIMITER ',' CSV
INSERT INTO history_log SELECT * FROM temp_history_log ON CONFLICT (itemid,clock,ns) DO NOTHING;

\copy history_text_old TO '/tmp/history_text.csv' DELIMITER ',' CSV
CREATE TEMP TABLE temp_history_text (
    itemid                  bigint                                  NOT NULL,
    clock                   integer         DEFAULT '0'             NOT NULL,
    value                   text            DEFAULT ''              NOT NULL,
    ns                      integer         DEFAULT '0'             NOT NULL
);
\copy temp_history_text FROM '/tmp/history_text.csv' DELIMITER ',' CSV
INSERT INTO history_text SELECT * FROM temp_history_text ON CONFLICT (itemid,clock,ns) DO NOTHING;
```

- 古いテーブルを削除するには、移行後の手順 に従ってください。

PostgreSQL + TimescaleDB

セッションが切断されないようにするため、エクスポートおよびインポートは tmux/screen で実行する必要があります。アップグレード中は Zabbix サーバーを停止しておく必要があります。

参照: 重要な注意事項

- history_upgrade_prepare.sql を使用してテーブル名を変更します。
  - 圧縮が有効な場合(デフォルトのインストール)、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-pa のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - 圧縮が無効な場合は、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compre のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- 圧縮設定に応じて TimescaleDB の hypertable 移行スクリプトを実行します:
  - 圧縮が有効な場合(デフォルトのインストール)、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-pa のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/trends_u
    ```

  - 圧縮が無効な場合は、/usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compre のスクリプトを実行します:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/trend
```

参照: INSERT パフォーマンスを改善するための ヒント。

- 古いテーブルを削除するには、移行後の手順 に従ってください。

移行後

すべてのデータベースについて、移行が完了したら次の作業を行ってください。

- すべてが想定どおりに動作することを確認します。
- 古いテーブルを削除します。

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- TimescaleDB の場合は、次の古いテーブルも削除します。

```
DROP TABLE trends_old;
```

参照

- パーティショニングのための監査ログテーブルの準備

4 監査ログテーブルのパーティション化の準備

概要

一部のデータベース（MySQL など）では、パーティション列をテーブルの一意制約の一部にする必要があります。したがって、auditlog テーブルを時間でパーティション分割するには、主キーを auditid から複合キー auditid + clock に変更する必要があります。

このセクションでは、auditlog テーブルの主キーを変更する手順について説明します。

このページに記載されている手順は、上級ユーザー向けに設計されています。これらの手順は、お客様の構成に合わせて調整する必要がある場合があります。主キーの変更は、将来のアップグレードパッチと互換性がなくなる可能性があるため、手動でのアップグレードが必要になる場合があります。

 主キーの変更は、auditlog テーブルのサイズによっては、多くのリソースを消費し、多くの時間を要する操作となる場合があります。変更中は、Zabbix サーバーを停止し、Zabbix フロントエンドをメンテナンスモード に切り替えることをお勧めします。ただし、どうしても必要な場合は、ダウンタイムなしで主キーを変更する方法があります (下記参照)

auditlog テーブルをパーティション分割すると、例えば大規模なシステムにおけるハウスキーピング機能の向上につながります。Zabbix のhousekeeping は現在、パーティション分割されたテーブルを利用できません（TimescaleDB を除く）が、スクリプトを使用して Zabbix ハウスキーピングを無効化し、パーティションを削除することができます。

Zabbix 7.0 以降、TimescaleDB の auditlog テーブルはハイパーテーブルに変換され、ハウスキーパーがデータをチャンク単位で削除できるようになりました。既存の auditlog テーブルをハイパーテーブルにアップグレードするには、TimescaleDB スキーマのアップグレード を参照してください。

MySQL

インデックス再構築に関する重要な注意事項

MySQL は、ALTER TABLE 操作中に主キーのインデックスを自動的に再構築します。ただし、データベースの最適なパフォーマンスを確保するために、OPTIMIZE TABLE 文を使用してインデックスも手動で再構築することを強く推奨します。

インデックスの再構築では、一時的にテーブル自体が使用している容量と同程度の追加ディスク領域が必要になる場合があります。現在のデータおよびインデックスのサイズを取得するには、次の文を実行できます。

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

使用可能なディスク領域が懸念される場合は、ダウンタイムなしでの主キー変更 の手順に従ってください。そのほかにも、次のような方法があります。

- sort_buffer_size MySQL パラメータを増やすことで、インデックスを手動で再構築する際のディスク使用量を削減できる場合があります。ただし、この変数を変更すると、データベース全体のメモリ使用量に影響する可能性があります。
- 不要な可能性のあるデータを削除して、空き領域を確保することを検討してください。
- housekeeper を実行する前に、housekeeper の データ保存期間パラメータを短くすることを検討してください。

ダウンタイムを伴う主キーの変更

1. 現在の auditlog テーブルの主キーを削除し、新しい主キーを追加します。

```
ALTER TABLE Auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. インデックスを再構築します（オプション設定ですが、強く推奨します。インデックスの再構築に関する重要な注意事項を参照してください）

```
OPTIMIZE TABLE Auditlog;
```

ダウンタイムなしでの主キーの変更

主キーを手動で変更する方法をここで説明します。
または、Percona の pt-online-schema-change ツールキットを使用することもできます。
このツールキットは、auditlog テーブルの変更に使用する領域を最小限に抑えながら、以下の操作を自動的に実行します。

1. 新しい主キーを持つ新しいテーブルを作成し、インデックスを作成します。

```
CREATE TABLE `auditlog_new` (
  `auditid`             varchar(25)                             NOT NULL,
  `userid`              bigint unsigned                         NULL,
  `username`            varchar(100)    DEFAULT ''              NOT NULL,
  `clock`               integer         DEFAULT '0'             NOT NULL,
  `ip`                  varchar(39)     DEFAULT ''              NOT NULL,
  `action`              integer         DEFAULT '0'             NOT NULL,
  `resourcetype`        integer         DEFAULT '0'             NOT NULL,
  `resourceid`          bigint unsigned                         NULL,
  `resource_cuid`       varchar(25)                             NULL,
  `resourcename`        varchar(255)    DEFAULT ''              NOT NULL,
  `recordsetid`         varchar(25)                             NOT NULL,
  `details`             longtext                                NOT NULL,
  PRIMARY KEY (auditid,clock)
) ENGINE=InnoDB;
CREATE INDEX `auditlog_1` ON `auditlog_new` (`userid`,`clock`);
CREATE INDEX `auditlog_2` ON `auditlog_new` (`clock`);
CREATE INDEX `auditlog_3` ON `auditlog_new` (`resourcetype`,`resourceid`);
```

2. テーブルを入れ替えます。

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. 古いテーブルから新しいテーブルへデータをコピーします。

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

過剰なリソース使用を避けるため、これはチャンク単位（必要に応じて WHERE clock 句を含む複数の INSERT INTO 文）で実行できます。

4. 古いテーブルを削除します。

```
DROP TABLE auditlog_old;
```

PostgreSQL

インデックス再構築に関する重要な注意事項

PostgreSQL は、ALTER TABLE 操作中に主キーのインデックスを自動的に再構築します。ただし、データベースのパフォーマンスを最適な状態に保つため、REINDEX TABLE CONCURRENTLY 文を使用してインデックスも手動で再構築することを強く推奨します。

インデックスの再構築では、一時的に現在インデックスが使用しているディスク容量の最大 3 倍が必要になる場合があります。現在のインデックスサイズを確認するには、次のクエリを実行できます。

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

使用可能なディスク容量が懸念される場合は、ダウンタイムなしで主キーを変更する の手順に従ってください。そのほかにも、次のような方法があります。

- PostgreSQL パラメータ maintenance_work_mem を増やすことで、インデックスを手動で再構築する際のディスク使用量を削減できる場合があります。ただし、この変数を変更すると、データベース全体のメモリ使用量に影響する可能性があります。
- より多くの空き容量がある別のディスクまたはテーブルスペースがある場合は、インデックス再構築用の一時保存場所を変更することを検討してください。一時オブジェクト用に別のテーブルスペースを指定するには、PostgreSQL パラメータ temp_tablespaces を設定できます。
- 不要な可能性のあるデータを削除して、空き容量を確保することを検討してください。
- housekeeper を実行する前に、housekeeper の データ保存期間パラメータを短くすることを検討してください。

ダウンタイムを伴う主キーの変更

1. 現在の auditlog テーブルの主キーを削除し、新しい主キーを追加します。

```
ALTER TABLE Auditlog DROP CONSTRAINT Auditlog_pkey;
ALTER TABLE Auditlog ADD PRIMARY KEY (auditid,clock);
```

2. インデックスを再構築します（オプション作業ですが、強く推奨します。インデックスの再構築に関する重要な注意事項を参照してください）

```
REINDEX TABLE CONCURRENTLY Auditlog;
```

ダウンタイムなしでの主キーの変更

主キーを手動で変更する方法をここで説明します。または、新しいテーブルの作成、データのコピー、テーブルの入れ替えのために、pg_repack 拡張機能を検討することもできます。

1. 新しい主キーを持つ新しいテーブルを作成し、インデックスを作成します。

```
CREATE TABLE auditlog_new (
  auditid             varchar(25)                             NOT NULL,
  userid              bigint                                  NULL,
  username            varchar(100)    DEFAULT ''              NOT NULL,
  clock               integer         DEFAULT '0'             NOT NULL,
  ip                  varchar(39)     DEFAULT ''              NOT NULL,
  action              integer         DEFAULT '0'             NOT NULL,
  resourcetype        integer         DEFAULT '0'             NOT NULL,
  resourceid          bigint                                  NULL,
  resource_cuid       varchar(25)                             NULL,
  resourcename        varchar(255)    DEFAULT ''              NOT NULL,
  recordsetid         varchar(25)                             NOT NULL,
  details             text            DEFAULT ''              NOT NULL,
  PRIMARY KEY (auditid,clock)
);
CREATE INDEX auditlog_new_1 ON auditlog_new (userid,clock);
CREATE INDEX auditlog_new_2 ON auditlog_new (clock);
CREATE INDEX auditlog_new_3 ON auditlog_new (resourcetype,resourceid);
```

2. テーブルを入れ替えます。

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. 古いテーブルから新しいテーブルへデータをコピーします。

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

過剰なリソース使用を避けるため、これはチャンク単位で実行できます（必要に応じて WHERE clock 句を付けた複数の INSERT INTO 文）。

4. 古いテーブルを削除します。

```
DROP TABLE auditlog_old;
```

関連項目

- プライマリキーへのデータベースアップグレード

5 データベースへの安全な接続

概要

このセクションでは、以下の間で安全な TLS 接続を確立するための Zabbix のセットアップ手順と設定例を説明します。

データベース	Zabbix コンポーネント
MySQL	Zabbix Web インターフェース、Zabbix サーバー、Zabbix プロキシ
PostgreSQL	Zabbix Web インターフェース、Zabbix サーバー、Zabbix プロキシ

DBMS 内で接続の暗号化を設定するには、詳細について各ベンダーの公式ドキュメントを参照してください。

- MySQL: ソースおよびレプリカのレプリケーション用データベースサーバー。
- MySQL: グループレプリケーションなどのデータベースサーバー。
- PostgreSQL 暗号化オプション。

すべての例は、CentOS 8 で公式リポジトリから利用可能な MySQL CE (8.0) および PgSQL (13) の GA リリースに基づいています。

要件

暗号化を設定するには、次が必要です。

- デベロッパーサポートされている OS と OpenSSL1.1.X 以上または OpenSSL の代用品。

Note:
特に新規インストールの場合は、サポート終了した OS は避けることをお勧めします。

- デベロッパーが提供する公式リポジトリからインストールおよび保守されるデータベースエンジン（RDBMS）やオペレーティングシステムには、暗号化サポートが実装されていない古いバージョンのデータベースソフトウェアが付属していることがよくあります。たとえば、RHEL7 ベースのシステムでの PostgreSQL9.2 や暗号化サポートのない MariaDB5.5 などです。

用語

このオプションを設定すると、Zabbix サーバー/プロキシおよび Web インターフェースからデータベースへの接続に TLS 接続の使用が強制されます。

- `required` - ID の検証を行わず、転送モードとして TLS を使用して接続します
- `verify_ca` - TLS を使用して接続し、証明書を検証します
- `verify_full` - TLS を使用して接続し、証明書を検証し、さらに DBHost で指定されたデータベースの ID（CN）がその証明書と一致することを検証します

Zabbix の設定

Web インターフェースからデータベース

Web インターフェースのインストール中に、データベースへの安全な接続を設定できます。

- トランスポート暗号化を有効にするにはデータベース接続設定ステップでデータベースの *TLS* 暗号化チェックボックスをオンにします。
- 証明書による暗号化を有効にするには、*TLS* 暗号化フィールドがオンになっている場合に表示されるデータベースの証明書確認チェックボックスをオンにします。

Note:
MySQL の場合、データベースホストが localhost に設定されていると、ソケットファイル（Unix の場合）または共有メモリ（Windows の場合）を使用する接続を暗号化できないため、データベースの *TLS* 暗号化チェックボックスが無効になります。
PostgreSQL の場合、データベースホストフィールドの値がスラッシュで始まるか、フィールドが空の場合、*TLS* 暗号化チェックボックスは無効になります。

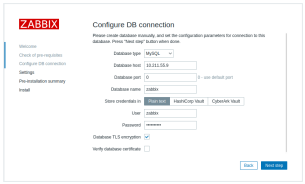
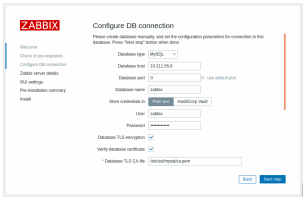
次のパラメータは、証明書モードの TLS 暗号化で使用可能になります。(両方のチェックボックスがマークされている場合)

パラメータ	説明
データベースの *TLS CA* ファイル	有効な TLS 認証局（CA）ファイルへのフルパスを指定します。
データベースの *TLS* キーファイル	有効な TLS キーファイルへのフルパスを指定します。
データベースの *TLS* 証明書ファイル	有効な TLS 証明書ファイルへのフルパスを指定します。
データベースのホスト検証	このチェックボックスをオンにすると、ホスト検証がアクティブになります。 PHP MySQL ライブラリではピア証明書の検証手順をスキップできないため、MySQL では無効になっています。
データベースの *TLS* 暗号化方式リスト	有効な暗号のカスタムリストを指定します。暗号リストの形式は、OpenSSL 標準に準拠している必要があります。 MySQL でのみ使用できます。

Attention:
TLS パラメータは有効なファイルを指している必要があります。存在しないファイルまたは無効なファイルを指している場合は、認証エラーが発生します。証明書ファイルが書き込み可能である場合、Web インターフェースはシステム情報レポートに「データベースの TLS 証明書ファイルは読み込み専用にしてください」という警告を表示します。(PHP ユーザーが証明書の所有者である場合にのみ表示されます)。
パスワードで保護された証明書はサポートされていません。

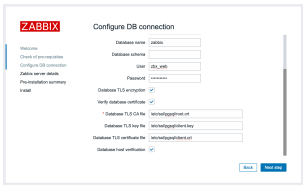
使用例

Zabbix の Web インターフェースでは、GUI インターフェースを使用して、利用可能なオプション `required`、`verify_ca`、`verify_full` を定義します。必要なオプションは、インストールウィザードの *Configure DB connections* ステップで指定します。これらのオプションは、次のように設定ファイル (zabbix.conf.php) にマッピングされます。

GUI 設定	設定ファイル	説明	結果
	... // TLS 接続に使用。 $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	*Database TLS encryption* をチェックする *Verify database certificate* は未チェックのままにする	`required` モードを有効化します。
	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem'; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	1. *Database TLS encryption* と *Verify database certificate* をチェックする 2. *Database TLS CA file* のパスを指定する	`verify_ca` モードを有効化します。

GUI 設定	設定ファイル	説明	結果

	...
	// 厳密に定義された Cipher リストを使用する TLS 接続に使用。
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '<cipher_list>';
	...
	
	または:
	
	...
	// Cipher リストを定義しない TLS 接続に使用 - MySQL サーバーによって選択される
	$DB['ENCRYPTION'] = true;
	$DB['KEY_FILE'] = '<key_file_path>';
	$DB['CERT_FILE'] = '<key_file_path>';
	$DB['CA_FILE'] = '<key_file_path>';
	$DB['VERIFY_HOST'] = true;
	$DB['CIPHER_LIST'] = '';
	...
		1. *Database TLS encryption* と *Verify database certificate* をチェックする
		2. *Database TLS key file* のパスを指定する
		3. *Database TLS CA file* のパスを指定する
		4. *Database TLS certificate file* のパスを指定する
		5. *Database TLS cipher list* を指定する（任意）
			MySQL で `verify_full` モードを有効化します。

	... $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = '<key_file_path>'; $DB['CERT_FILE'] = '<key_file_path>'; $DB['CA_FILE'] = '<key_file_path>'; $DB['VERIFY_HOST'] = true; $DB['CIPHER_LIST'] = ' '; ...	1. *Database TLS encryption* と *Verify database certificate* をチェックする 2. *Database TLS key file* のパスを指定する 3. *Database TLS CA file* のパスを指定する 4. *Database TLS certificate file* のパスを指定する 5. *Database host verification* をチェックする	PostgreSQL で `verify_full` モードを有効化します。

関連情報: MySQL の暗号化設定例, PostgreSQL の暗号化設定例。

Zabbix サーバー/プロキシの設定

データベースへのセキュアな接続は、Zabbix のserverおよび/またはproxy設定ファイル内のそれぞれのパラメータで設定できます。

Configuration	Result
なし	暗号化なしでデータベースに接続します。
1. DBTLSConnect=required を設定	サーバー/プロキシは、データベースへの TLS 接続を確立します。暗号化されていない接続は許可されません。
1. DBTLSConnect=verify_ca を設定 2. DBTLSCAFile を設定 - TLS 認証局ファイルを指定	サーバー/プロキシは、データベース証明書を検証した後、データベースへの TLS 接続を確立します。
1. DBTLSConnect=verify_full を設定 2. DBTLSCAFile を設定 - TLS 認証局ファイルを指定	サーバー/プロキシは、データベース証明書とデータベースホストの識別情報を検証した後、データベースへの TLS 接続を確立します。
1. DBTLSCAFile を設定 - TLS 認証局ファイルを指定 2. DBTLSCertFile を設定 - クライアント公開鍵証明書ファイルを指定 3. DBTLSKeyFile を設定 - クライアント秘密鍵ファイルを指定	サーバー/プロキシは、データベースへの接続時にクライアント証明書を提供します。
1. DBTLSCipher を設定 - クライアントが TLS 1.2 までの TLS プロトコルを使用する接続で許可する暗号スイートの一覧 または DBTLSCipher13 - クライアントが TLS 1.3 プロトコルを使用する接続で許可する暗号スイートの一覧	(MySQL) 提供された一覧の暗号スイートを使用して TLS 接続が確立されます。 (PostgreSQL) このオプションを設定するとエラーと見なされます。

1 MySQL の暗号化設定

概要

このセクションは CentOS8.2 および MySQL8.0.21 の暗号化設定の例をいくつか示し、データベースへの接続を暗号化するためのクイックスタートガイドとして使用できます。

Attention:
MySQL ホストが localhost に設定されている場合、暗号化オプションは使用できません。この場合、Zabbix フロントエンドとデータベース間の接続はソケットファイル（Unix の場合）または共有メモリ（Windows の場合）を使用し、暗号化できません。

Note:
暗号化の組み合わせのリストは、このページにリストされているものに限定されません。利用できる組み合わせはもっとたくさんあります。

前提条件

公式リポジトリから MySQL データベースをインストールします。

MySQL リポジトリの使用方法詳細についてはMySQL のドキュメントを参照してください。

MySQL サーバーは、自己署名証明書を使用して安全な接続を受け入れる準備ができています。

暗号化された接続を使用しているユーザーを確認するには、次のクエリを実行します（パフォーマンススキーマをオンにする必要があります）：

```
mysql> SELECT sbt.variable_value AS tls_version, t2.variable_value AS cipher, processlist_user AS user, pr
            FROM performance_schema.status_by_thread  AS sbt
            JOIN performance_schema.threads AS t ON t.thread_id = sbt.thread_id
            JOIN performance_schema.status_by_thread AS t2 ON t2.thread_id = t.thread_id
            WHERE sbt.variable_name = 'Ssl_version' and t2.variable_name = 'Ssl_cipher'
            ORDER BY tls_version;
```

転送時のみの暗号化

MySQL の設定

最近のバージョンのデータベースは、required 暗号化モード に対して、そのままで使用できる状態になっています。初期設定と起動後に、サーバー側の証明書が作成されます。

主要コンポーネント用のユーザーとロールを作成します。

Attention:
MySQL バージョン 8.4 以降では、mysql_native_password の代わりに caching_sha2_password を使用する必要があります。

```
mysql> CREATE USER
 'zbx_srv'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>',
 'zbx_web'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>'
 REQUIRE SSL
 PASSWORD HISTORY 5;

mysql> CREATE ROLE 'zbx_srv_role', 'zbx_web_role';

mysql> GRANT SELECT, UPDATE, DELETE, INSERT, CREATE, DROP, ALTER, INDEX, REFERENCES ON zabbix.* TO 'zbx_sr
mysql> GRANT SELECT, UPDATE, DELETE, INSERT ON zabbix.* TO 'zbx_web_role';

mysql> GRANT 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> GRANT 'zbx_web_role' TO 'zbx_web'@'%';

mysql> SET DEFAULT ROLE 'zbx_srv_role' TO 'zbx_srv'@'%';
mysql> SET DEFAULT ROLE 'zbx_web_role' TO 'zbx_web'@'%';
```

X.509 プロトコルは本人確認には使用されませんが、ユーザーは暗号化された接続のみを使用するよう設定されていることに注意してください。ユーザー設定の詳細については、MySQL documentation を参照してください。

接続を確認するには次を実行します（セキュア接続のテストにはソケット接続は使用できません）。

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

現在の状態と利用可能な暗号スイートを確認します。

```
mysql> status
--------------
mysql Ver 8.0.21 for Linux on x86_64 (MySQL Community Server - GPL)

Connection id: 62
Current database:
Current user: zbx_srv@bfdb.local
SSL: Cipher in use is TLS_AES_256_GCM_SHA384

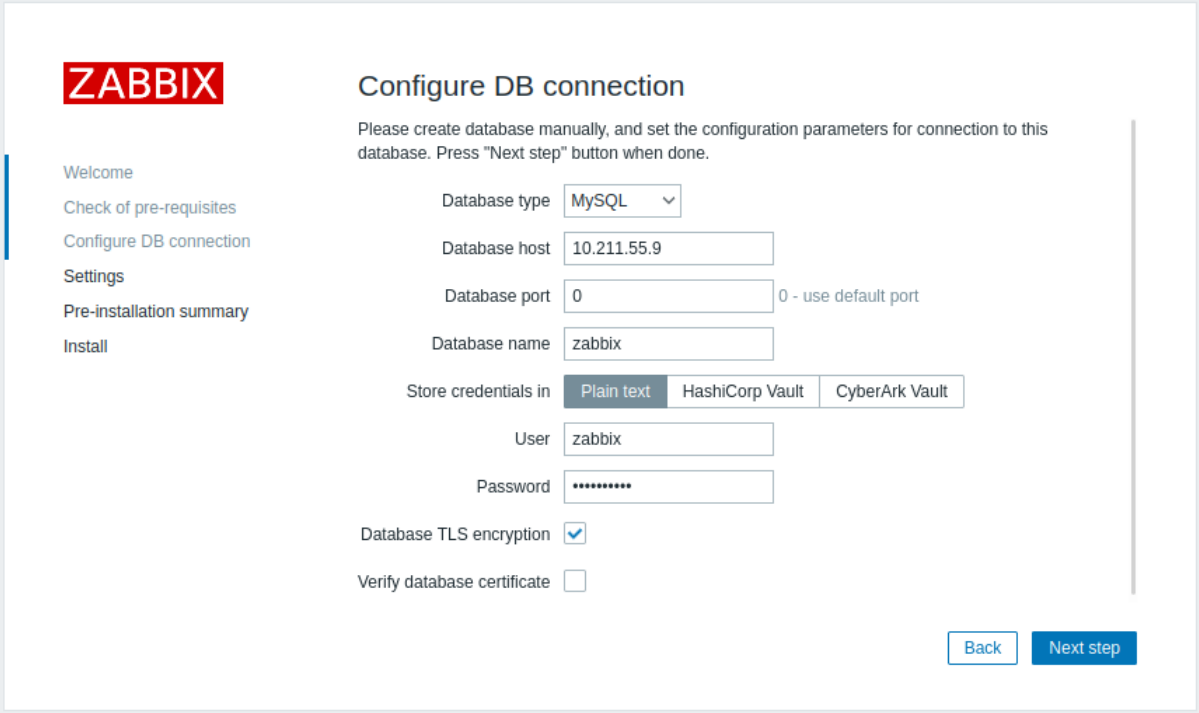

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

フロントエンド

Zabbix フロントエンドとデータベース間の接続でトランスポートのみの暗号化を有効にするには：

- データベースの *TLS* 暗号化をチェック
- データベース証明書の確認をチェックしない



サーバ

サーバーとデータベース間の接続でトランスポートのみの暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

認証局検証による暗号化

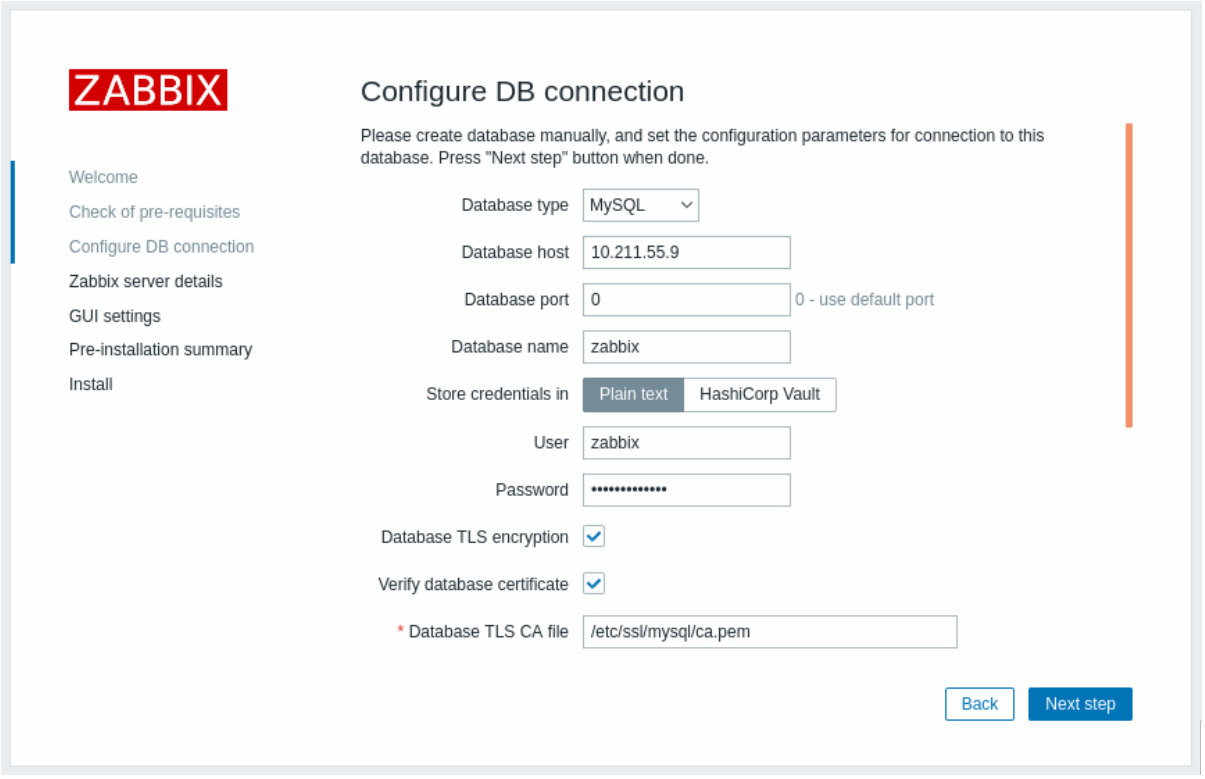
必要な MySQL CA を Zabbix Web インターフェースサーバーにコピーし、Web サーバーがこのファイルを読み取れるように適切な権限を設定します。

Note:
このモードは、MySQL ライブラリが古いため、RHEL 7 では動作しません。

Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で、証明書検証付きの暗号化を有効にするには、次のようにします。

- *Database TLS encryption* と *Verify database certificate* をチェックします
- Database TLS CA ファイルへのパスを指定します



または、*/etc/zabbix/web/zabbix.conf.php* で設定することもできます。

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

必要なユーザーで接続可能かどうかを確認するには、コマンドラインツールを使用してトラブルシュートします。

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

サーバ

Zabbix サーバーとデータベース間の証明書検証による暗号化接続を有効にするには*/etc/zabbix/zabbix_server.conf* を設定します：

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

完全検証による暗号化

MySQL の設定

MySQL CE サーバーの設定オプション（*/etc/my.cnf.d/server-tls.cnf*）を次のように設定します。

```
[mysqld]
...
##### この例では、鍵は MySQL CE の datadir ディレクトリに配置されています
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
```

```
...
```

MySQL CE サーバーおよびクライアント（Zabbix Web インターフェース）用の鍵は、MySQL CE のドキュメントに従って手動で作成する必要があります: MySQL を使用した SSL および RSA 証明書と鍵の作成 または openssl を使用した SSL 証明書と鍵の作成

Attention:
MySQL サーバー証明書には、Common Name フィールドに FQDN 名を設定する必要があります。これは、Zabbix Web インターフェースがデータベースとの通信に DNS 名またはデータベースホストの IP アドレスを使用するためです。

MySQL ユーザーを作成します。

Attention:
MySQL バージョン 8.4 以降では、`mysql_native_password` の代わりに `caching_sha2_password` を使用する必要があります。

```
mysql> CREATE USER
  'zbx_srv'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>',
  'zbx_web'@'%' IDENTIFIED WITH mysql_native_password BY '<strong_password>'
  REQUIRE X509
  PASSWORD HISTORY 5;
```

そのユーザーでログインできるか確認します。

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

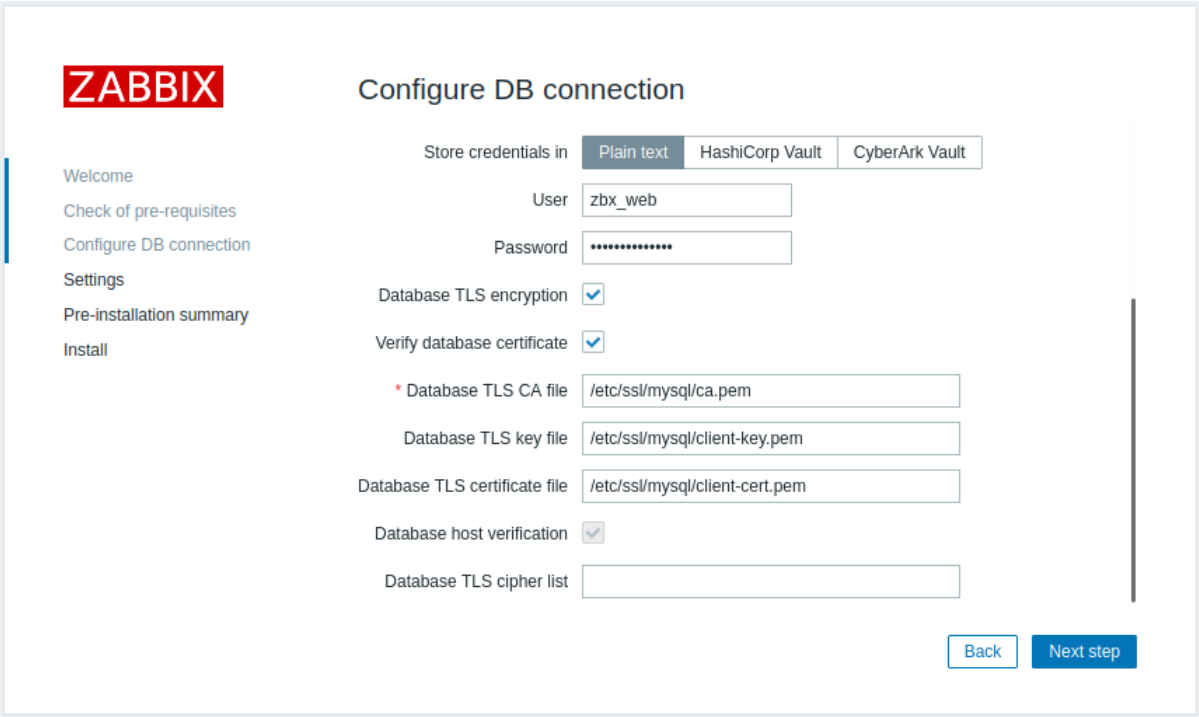
Web インターフェース

Zabbix Web インターフェースとデータベース間の接続で完全検証による暗号化を有効にするには、次の設定を行います。

- *Database TLS encryption* と *Verify database certificate* をチェックします
- *Database TLS key file* のパスを指定します
- *Database TLS CA file* のパスを指定します
- *Database TLS certificate file* のパスを指定します

Database host verification はチェック済みでグレーアウトしていることに注意してください。この手順は MySQL では省略できません。

Warning:
Database TLS cipher list フィールドを空のままにすると、Web インターフェース（クライアント）とサーバーの両方で許可されている共通の暗号スイートが有効になります。または、暗号設定要件に従って、暗号スイートを明示的に設定することもできます。



または、*/etc/zabbix/web/zabbix.conf.php* で次のように設定することもできます。

```
...
// 厳密に定義された Cipher リストを使用する TLS 接続用
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
...
// または
...
// Cipher リストを定義しない TLS 接続用 - MySQL サーバーによって選択
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続を完全に検証して暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します：

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

2 PostgreSQL の暗号化設定

概要

このセクションでは、CentOS8.2 および PostgreSQL13 の暗号化設定例をいくつか示します。

> **Note:**
> *Database host* フィールドの値がスラッシュで始まるか、空の場合、Zabbix フロントエンドと PostgreSQL 間の接続を暗号化できません（GUI のパラメーターが無効になっています）

前提条件

公式リポジトリを使用して、PostgreSQL データベースをインストールします。

PostgreSQL は、初期状態では TLS 接続を受け付けるように設定されていません。postgresql.conf を使用した証明書の準備および ph_hba.conf によるユーザーアクセス制御については、PostgreSQL のドキュメントの手順に従ってください。

デフォルトでは、PostgreSQL ソケットは localhost にバインドされています。ネットワーク経由のリモート接続を許可するには、実際のネットワークインターフェースで待ち受けるように設定してください。

すべてのmodesに対する PostgreSQL の設定は、次のようになります。

/var/lib/pgsql/13/data/postgresql.conf:

```
...
ssl = on
ssl_ca_file = 'root.crt'
ssl_cert_file = 'server.crt'
ssl_key_file = 'server.key'
ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'
ssl_prefer_server_ciphers = on
```

```
ssl_min_protocol_version = 'TLSv1.3'
...
```

アクセス制御については、*/var/lib/pgsql/13/data/pg_hba.conf* を調整します。

```
...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

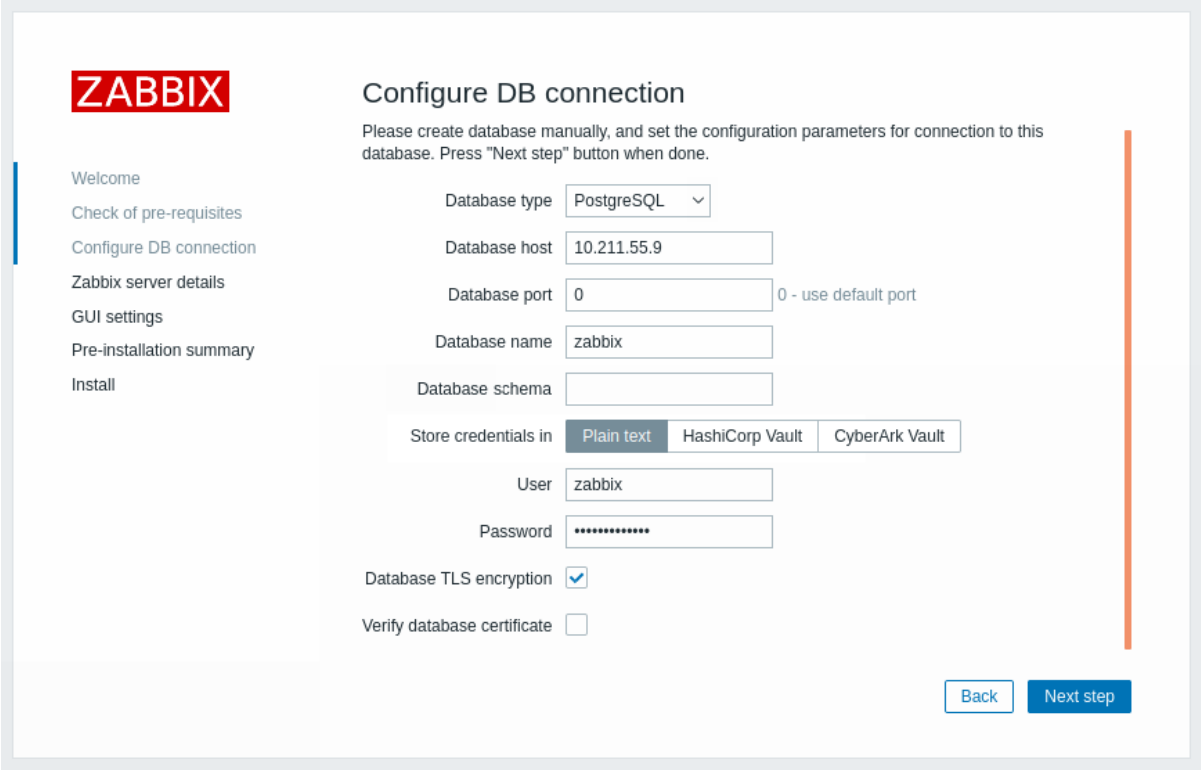
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

トランスポートのみの暗号化

フロントエンド

Zabbix フロントエンドとデータベース間の接続でトランスポートのみの暗号化を有効にするには:

- データベースの *TLS* 暗号化をチェックします
- データベース証明書の確認はチェックしません



サーバー

サーバーとデータベース間の接続に対してトランスポートのみの暗号化を有効にするには*/etc/zabbix/zabbix_server.conf* を設定します:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

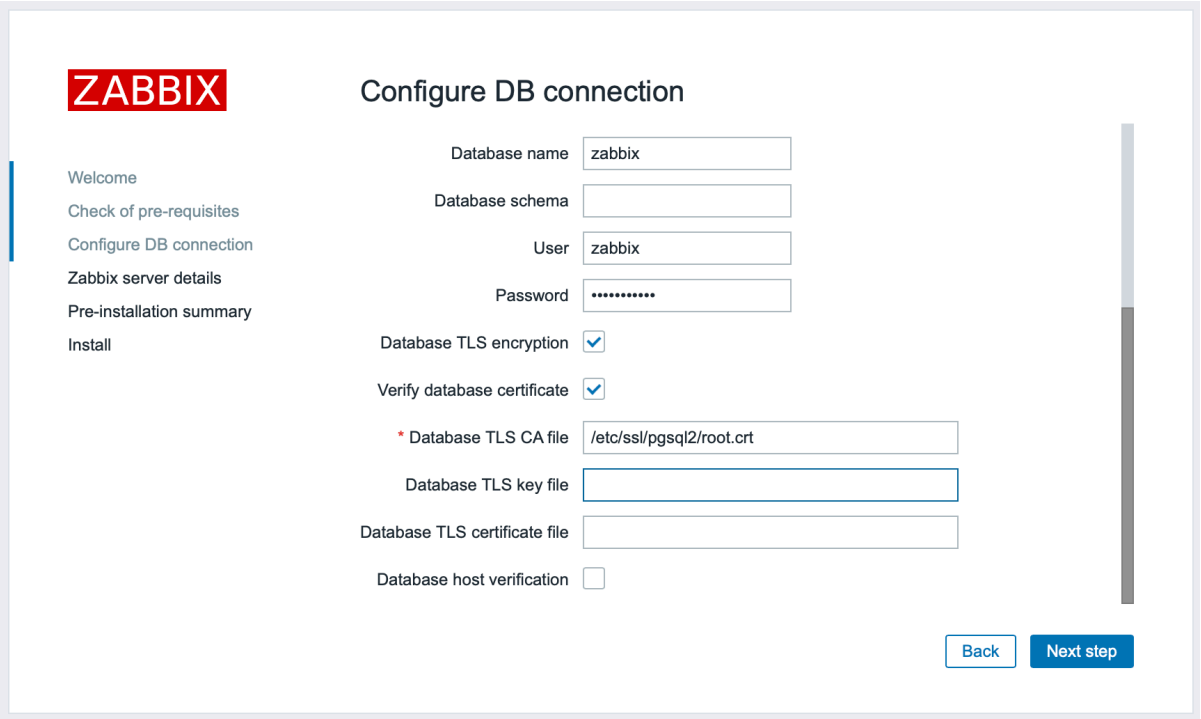
認証局検証による暗号化

フロントエンド

Zabbix フロントエンドとデータベース間の接続の認証局検証による暗号化を有効にするには:

- データベース *TLS* 暗号化とデータベース証明書の確認をチェック

- データベース *TLS CA* ファイルへのパスを指定



または、これは*/etc/zabbix/web/zabbix.conf.php:* で設定できます。

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続の証明書検証による暗号化を有効にするには、*/etc/zabbix/zabbix_server.conf* を設定します。

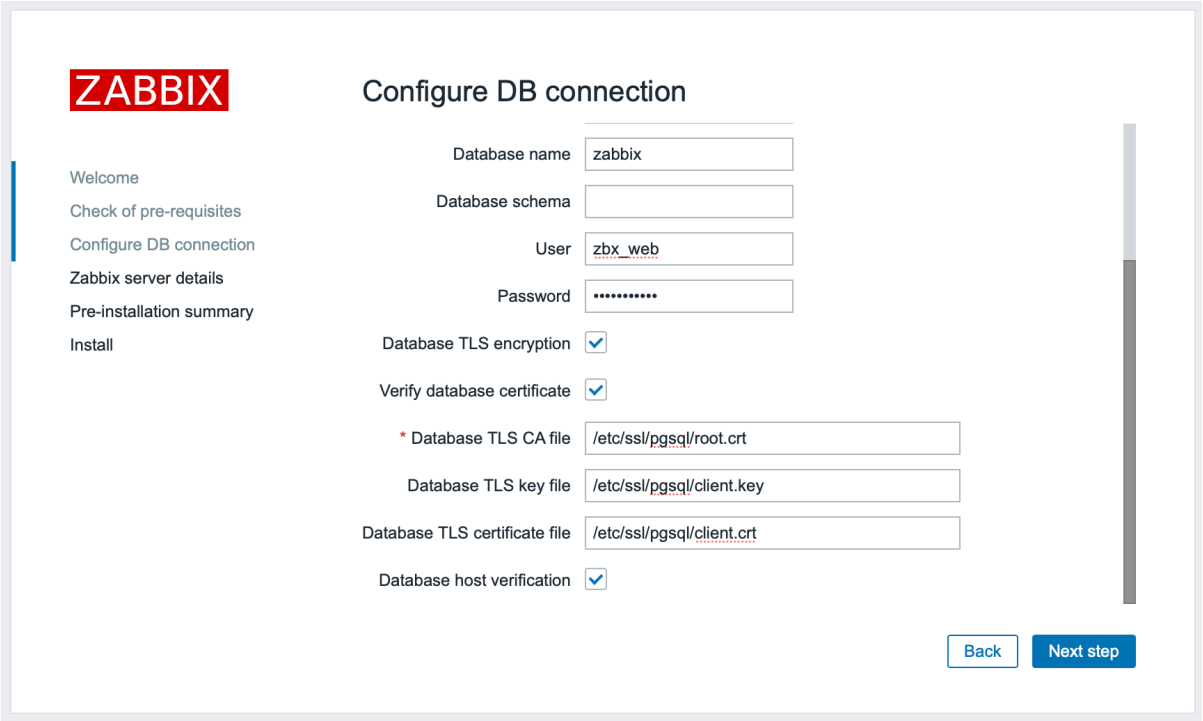
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

完全な検証を伴う暗号化

フロントエンド

Zabbix フロントエンドとデータベース間接続の証明書とデータベースホスト ID 検証による暗号化を有効にするには：

- データベース *TLS* 暗号化とデータベース証明書の確認をチェック
- データベース *TLS* キーファイルへのパスを指定
- データベース *TLSCA* ファイルへのパスを指定
- データベース *TLS* 証明書ファイルへのパスを指定
- データベースホストの検証をチェック



または*/etc/zabbix/web/zabbix.conf.php* で設定：

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

サーバー

Zabbix サーバーとデータベース間の接続の証明書とデータベースホスト ID 検証による暗号化を有効にするには*/etc/zabbix/zabbix_server.conf* を設定します。

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

6 フロントエンドへの安全な接続

概要

このセクションでは、Zabbix フロントエンドと Zabbix サーバー間のセキュアな TLS 接続を実現するための Zabbix のセットアップ手順と設定例について説明します。

設定

デフォルトでは、Zabbix Web インターフェースと Zabbix サーバー間の通信は暗号化されていません。セキュリティを向上させるには、両側で TLS を有効にしてください。以下は、そのための最も簡単な方法の例です。

1. 証明書と鍵を生成します。

作業ディレクトリを作成します。

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

CA 証明書を作成します（MyZabbixCA の値は実際のコモンネームに合わせて調整してください）。

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Zabbix サーバー用の秘密鍵と証明書を生成します（zabbix-server.example.com の値は実際のコモンネームに合わせて調整してください）。

```
sudo openssl genrsa -out server.key 2048
sudo openssl req -new -key server.key -out server.csr -subj "/CN=zabbix-server.example.com/"
sudo openssl x509 -req -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out serv
```

Zabbix Web インターフェース用の秘密鍵と証明書を生成します（zabbix-frontend-node の値は実際のコモンネームに合わせて調整してください）。

```
sudo openssl genrsa -out frontend.key 2048
sudo openssl req -new -key frontend.key -out frontend.csr -subj "/CN=zabbix-frontend-node/"
sudo openssl x509 -req -days 365 -in frontend.csr -CA ca.crt -CAkey ca.key -CAcreateserial -sha256 -out fr
```

2. 適切な権限を設定します。

Zabbix サーバー用（お使いのディストリビューションの Zabbix サーバーデーモンユーザーに応じて所有者/グループを調整してください）。

```
sudo chown root:zabbix /etc/zabbix/ssl/server.{crt,key} /etc/zabbix/ssl/ca.crt
sudo chmod 640 /etc/zabbix/ssl/server.key
sudo chmod 644 /etc/zabbix/ssl/server.crt /etc/zabbix/ssl/ca.crt
```

Web インターフェース用（お使いのディストリビューションの Web サーバーユーザーに応じて所有者/グループを調整してください）。

```
sudo chown root:www-data /etc/zabbix/ssl/frontend.{crt,key}
sudo chmod 640 /etc/zabbix/ssl/frontend.key
sudo chmod 644 /etc/zabbix/ssl/frontend.crt
```

3. Zabbix サーバーを設定します。

zabbix_server.conf に以下を追加します。

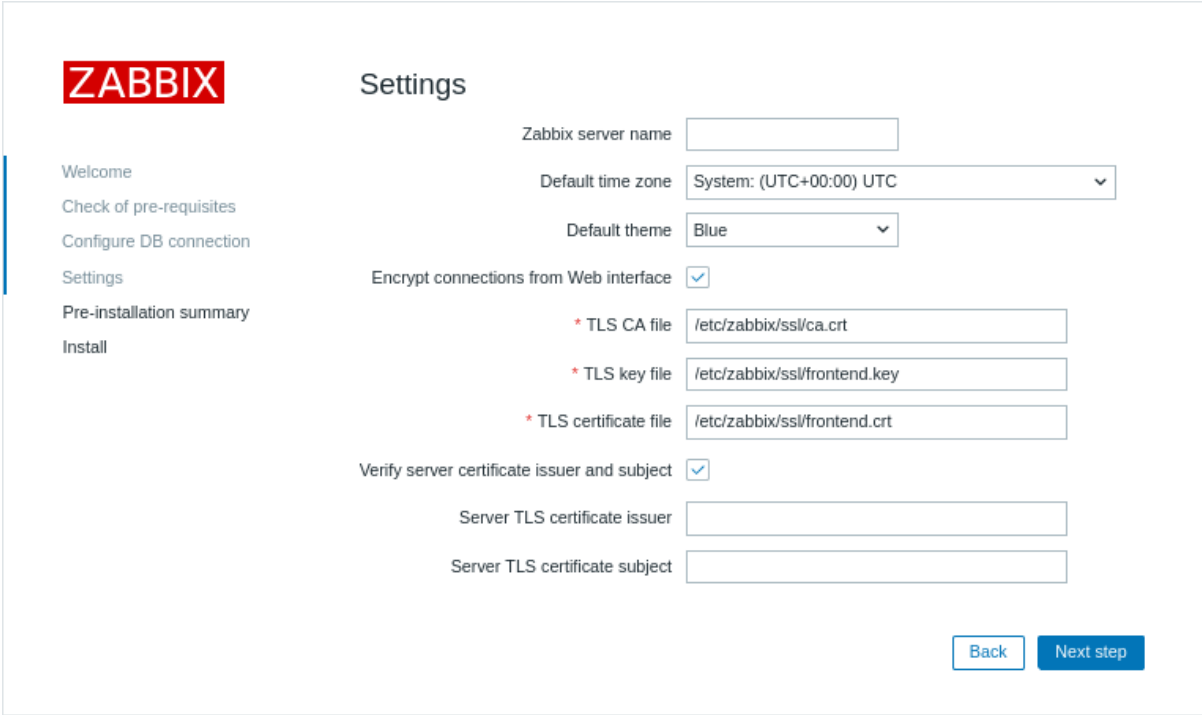
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Optionally:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

その後、サーバーを再起動します。

```
sudo systemctl restart zabbix-server
```

4. Zabbix Web インターフェースを設定します。

Web インターフェースのインストール時に、*Encrypt connections from Web interface* オプション（必要に応じて *Verify server certificate issuer and subject* オプションも）を有効にし、*TLS CA file*、*TLS key file*、*TLS certificate file* の各フィールド（必要に応じて *Server TLS certificate issuer* および *Server TLS certificate subject* フィールドも）に入力します。



Parameter	Description
TLS CA file	サーバー証明書の検証に使用する認証局（CA）証明書ファイルのフルパスを指定します。
TLS key file	クライアント証明書に対応するクライアント秘密鍵ファイルのフルパスを指定します。
TLS certificate file	相互 TLS 認証が必要な場合に、クライアント証明書ファイルのフルパスを指定します。
Server TLS certificate issuer	サーバー証明書と照合する発行者の識別名（DN）を指定します。
Server TLS certificate subject	サーバー証明書と照合するサブジェクトの識別名（DN）を指定します。

既存のインストールでは、`zabbix.conf.php` 内の以下のフィールドを編集します。

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Optionally:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = '/CN=MyZabbixCA/';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = '/CN=zabbix-server.example.com/';
```

5. Zabbix Web インターフェースまたは Zabbix サーバーのログファイルにエラーメッセージがないことを確認し、暗号化を検証します。

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 TimescaleDB のセットアップ

概要

Zabbix は TimescaleDB をサポートしています。TimescaleDB は PostgreSQL ベースのデータベースソリューションであり、データを時間ベースのチャンクに自動的に分割することで、大規模環境での高速なパフォーマンスを実現します。

Warning:
現在、TimescaleDB は Zabbix プロキシではサポートされていません。

このページの手順は、以下のシナリオで使用できます。

- TimescaleDB データベースを作成する場合、または既存の PostgreSQL テーブルから TimescaleDB へ移行する場合（設定を参照）。
- Zabbix のアップグレード時に既存の TimescaleDB データベーススキーマをアップグレードする場合（TimescaleDB スキーマのアップグレードを参照）。

設定

前提条件: データベースサーバーに、サポート対象バージョンの TimescaleDB 拡張機能がインストールされていること。インストール手順については、TimescaleDB documentationを参照してください。

> **Warning:**
> TimescaleDB をインストールする前に、official PostgreSQLリポジトリからサポート対象の PostgreSQL リリースをインストールしてください。

次を実行して、特定のデータベースで TimescaleDB 拡張機能を有効にします。

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

このコマンドの実行には、データベース管理者権限が必要です。

> **Note:**
> 'public' 以外のデータベーススキーマを使用している場合は、上記のコマンドに SCHEMA 句を追加する必要があります。例: `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

次に、`postgresql/timescaledb/schema.sql` スクリプトを実行します。新規インストールの場合、このスクリプトは通常の PostgreSQL データベースが初期スキーマ/データ付きで作成された後に実行する必要があります（database creationを参照）。

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> TimescaleDB バージョン 2.9.0 以降で `schema.sql` スクリプトを実行する際に、ベストプラクティスに従っていないことを示す警告メッセージが表示されても無視してください。この警告にかかわらず、設定は正常に完了します。

既存の履歴、トレンド、および監査ログデータの移行には、かなりの時間がかかる場合があります。移行期間中は、Zabbix サーバーおよび Web インターフェースを停止しておく必要があります。

`schema.sql` スクリプトは、次のハウスキーピングパラメータを設定します。

- アイテムの履歴保存期間を上書き
- アイテムのトレンド保存期間を上書き

履歴およびトレンドに対してパーティション化されたハウスキーピングを使用するには、これら両方のオプションを有効にする必要があります。また、履歴のみ、またはトレンドのみのいずれか一方について個別に上書きを有効にすることも可能です。

PostgreSQL および TimescaleDB では、`postgresql/timescaledb/schema.sql` スクリプトにより、追加で次の 2 つのパラメータが設定されます。

- 圧縮を有効化
- 7 日より古いレコードを圧縮

housekeeper が圧縮データを正常に削除するには、*Override item history period* オプションと *Override item trend period* オプションの両方を有効にする必要があります。
override が無効で、テーブルに圧縮チャンクがある場合、housekeeper はこれらのテーブルからデータを削除せず、設定が正しくないことを示す警告が*Housekeeping* セクションおよび*System information* セクションに表示されます。

これらのパラメータはすべて、インストール後に *Administration* > *Housekeeping* で変更できます。

> **Note:**
> `postgresql.conf` 内の PostgreSQL 設定パラメータを最適化するために、TimescaleDB が提供する timescaledb-tune ツールを実行することをお勧めします。

TimescaleDB スキーマのアップグレード

新しい TimescaleDB ハイパーテーブルを含むバージョンに Zabbix をアップグレードする場合、Zabbix サーバーはそれらのハイパーテーブルを自動的には設定しません（例えば、Zabbix 6.4 から 7.4 へアップグレードする場合、バージョン 7.0.0 および 7.0.2 で新しいハイパーテーブルが導入されています）。

新しい TimescaleDB ハイパーテーブルを設定するには、次の手順に従ってください。

1. Zabbix サーバーを起動します。これにより既存のデータベースがアップグレードされます。

2. データベースのアップグレードが完了したことをサーバーのログファイルで確認し、完了後に Zabbix サーバーを停止します。テーブルがハイパーテーブルではない場合に圧縮を有効化しようとすると、サーバーは警告をログに記録することに注意してください。
3. `postgresql/timescaledb/schema.sql` スクリプトを実行します。これにより、新しい TimescaleDB ハイパーテーブルが設定されます。なお、Zabbix 7.0.0 以降、スクリプトの場所と名前は `postgresql/timescaledb.sql` から `postgresql/timescaledb/schema.sql` に変更されています。

Attention:
TimescaleDB バージョン 2.9.0 以降で `schema.sql` スクリプトを実行する際に、ベストプラクティスに従っていないことを示す警告メッセージは無視してください。この警告にかかわらず、設定は正常に完了します。

TimescaleDB 圧縮

TimescaleDB ハイパーテーブルであるすべての Zabbix テーブルで、TimescaleDB ネイティブ圧縮がサポートされています。TimescaleDB へのアップグレードまたは移行時には、大きなテーブルの初回圧縮にかなり時間がかかる場合があります。

圧縮は"timescale" Timescale Community ライセンスでサポートされており、"apache" Apache 2.0 ライセンスではサポートされていないことに注意してください。Zabbix が圧縮をサポートしていないことを検出した場合、警告メッセージが Zabbix サーバーログに書き込まれ、ユーザーは Web インターフェースで圧縮を有効にできません。

Note:
圧縮を使用する前に、TimescaleDB documentation で圧縮について理解しておくことをお勧めします。

圧縮には、特に次のようないくつかの制限があることに注意してください。

- 圧縮済みチャンクの変更（挿入、削除、更新）はできません
- 圧縮済みテーブルのスキーマ変更はできません

圧縮設定は、Zabbix Web インターフェースの *Administration* > *Housekeeping* セクションにある *History, trends and audit log compression* ブロックで変更できます。

Parameter	Default	Comments
Enable compression	Enabled	チェックボックスをオンまたはオフにしても、圧縮はすぐには有効化/無効化されません。圧縮は Housekeeper によって処理されるため、変更が反映されるまで最大で `HousekeepingFrequency` 時間の 2 倍かかります（zabbix_server.conf で設定） 圧縮を無効にした後、圧縮期間に該当する新しいチャンクは圧縮されません。ただし、以前に圧縮されたすべてのデータは圧縮されたまま残ります。以前に圧縮されたチャンクを非圧縮に戻すには、TimescaleDB documentation の手順に従ってください。
Compress records older than	7d	TimescaleDB サポート付きの古いバージョンの Zabbix からアップグレードする場合、圧縮はデフォルトでは有効になりません。 このパラメータは 7 日未満にはできません。 圧縮済みチャンクは不変であるため、この値より古い遅延データ（例: プロキシによって遅延したデータ）はすべて破棄されます。

トレンド更新のパフォーマンスを向上させるには、`trends` および `trends_uint` テーブルの"chunk_time_interval" を 30 日から 7 日以下に下げることを検討してください。値は、トレンドを使用するアイテム数に応じて調整します。この設定の目的は、TimescaleDB のベストプラクティスに従い、チャンクサイズがシステムで利用可能なリソース内に収まるようにすることです。

8 Elasticsearch のセットアップ

Attention:
Elasticsearch のサポートは実験的機能です！

Zabbix は、データベースの代わりに Elasticsearch を使用して履歴データを保存することをサポートしています。ユーザーは、履歴データの保存先として、互換性のあるデータベースと Elasticsearch のいずれかを選択できます。このセクションで説明するセットアップ手順は、Elasticsearch バージョン 7.X に適用されます。これより前または後のバージョンの Elasticsearch を使用する場合、一部の機能が意図したとおりに動作しない可能性があります。

Warning:
1. すべての履歴データが Elasticsearch に保存される場合、トレンドは計算もデータベースへの保存もされません。トレンドが計算・保存されない場合、履歴の保存期間を延長する必要があることがあります。
 2. Elasticsearch を使用する場合、データベースから値を取得する範囲クエリは、データ保存期間のタイムスタンプによって制限されます。

設定

関連するすべての要素間の適切な通信を確保するには、サーバー構成ファイルとフロントエンド構成ファイルのパラメーターが適切に構成されていることを確認してください。

Zabbix サーバーとフロントエンド

更新するパラメーターを含む Zabbix サーバー構成ファイルの草案：

```
### Option: HistoryStorageURL
# History storage HTTP[S] URL.
#
# Mandatory: no
# Default:
# HistoryStorageURL=
### Option: HistoryStorageTypes
# Comma separated list of value types to be sent to the history storage.
#
# Mandatory: no
# Default:
# HistoryStorageTypes=uint,dbl,str,log,text
```

Zabbix サーバー構成ファイルに入力するパラメーター値の例：

```
HistoryStorageURL=http://test.elasticsearch.lan:9200
HistoryStorageTypes=str,log,text
```

この設定により、Zabbix サーバーは数値タイプの履歴値を対応するデータベースに保存し、テキスト履歴データを Elasticsearch に保存します。

Elasticsearch は次のアイテムタイプをサポートしています：

```
uint,dbl,str,log,text
```

サポートされているアイテムタイプの説明：

アイテム値のタイプ	データベーステーブル	**Elasticsearch** でのタイプ
符号なし数値型	history_uint	uint
浮動小数点数型	history	dbl
固定長文字列	history_str	str
ログ	history_log	log
テキスト	history_text	text

更新するパラメータを含む Zabbix フロントエンド設定ファイル（conf/zabbix.conf.php）の草案：

```
// Elasticsearch url (can be string if same url is used for all types).
$HISTORY['url']   = [
      'uint' => 'http://localhost:9200',
      'text' => 'http://localhost:9200'
];
// Value types stored in Elasticsearch.
$HISTORY['types'] = ['uint', 'text'];
```

Zabbix フロントエンド設定ファイルに入力するパラメータ値の例：

```
$HISTORY['url']   = 'http://test.elasticsearch.lan:9200';
$HISTORY['types'] = ['str', 'text', 'log'];
```

この設定により、Elasticsearch に Text、Character、および Log の履歴値が強制的に保存されます。

また、すべてが正しく機能するように conf/zabbix.conf.php で $HISTORY をグローバルにする必要があります。(方法については conf/zabbix.conf.php.example を参照してください)

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;
```

Elasticsearch のインストールとマッピングの作成

動作させるための最後の 2 つの手順は、Elasticsearch 自体をインストールし、マッピング処理を作成することです。

Elasticsearch をインストールするには、Elasticsearch installation guideを参照してください。

> **Note:**
> マッピングは Elasticsearch 内のデータ構造です（データベースのテーブルに似ています）。すべての履歴データ型のマッピングは database/elasticsearch/elasticsearch.map で利用できます。

> **Warning:**
> マッピングの作成は必須です。手順に従ってマッピングが作成されていない場合、一部の機能は正常に動作しません。

text 型のマッピングを作成するには、Elasticsearch に対して次のリクエストを送信します。

```
curl -X PUT \
 http://your-elasticsearch.here:9200/text \
 -H 'content-type:application/json' \
 -d '{
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "fields": {
               "analyzed": {
                  "index": true,
                  "type": "text",
                  "analyzer": "standard"
               }
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

同様のリクエストを実行して、対応する型を修正した Character および Log 履歴値のマッピングも作成する必要があります。

> **Note:**
> Elasticsearch を使用するには、追加情報についてRequirementsを参照してください。

> **Note:**
> Housekeeper は Elasticsearch からデータを削除しません。

複数の日付ベースのインデックスに履歴データを保存する

このセクションでは、パイプラインと取り込みノードを操作するために必要な追加の手順について説明します。

まず、インデックスのテンプレートを作成する必要があります。

次の例は、uint テンプレートを作成するためのリクエストです：

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/uint_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [
      "uint*"
   ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "type": "long"
         }
      }
   }
}'
```

他のテンプレートを作成するには、URL を変更し（最後の部分はテンプレートの名前です）、「index_patterns」フィールドをインデックス名と一致するように変更して、データベース/elasticsearch/elasticsearch.map から取得できる有効なマッピングを設定する必要があります。

たとえば、次のコマンドを使用して、テキストインデックスのテンプレートを作成できます。

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_template/text_template \
 -H 'content-type:application/json' \
 -d '{
   "index_patterns": [
      "text*"
   ],
   "settings": {
      "index": {
         "number_of_replicas": 1,
         "number_of_shards": 5
      }
   },
   "mappings": {
      "properties": {
         "itemid": {
            "type": "long"
         },
         "clock": {
            "format": "epoch_second",
            "type": "date"
         },
         "value": {
            "fields": {
               "analyzed": {
                  "index": true,
                  "type": "text",
                  "analyzer": "standard"
               }
```

```
            },
            "index": false,
            "type": "text"
         }
      }
   }
}'
```

これは、Elasticsearch が自動的に作成されたインデックスに有効なマッピングを設定できるようにするために必要です。次に、パイプライン定義を作成する必要があります。パイプラインは、データをインデックスに入れる前の、ある種のデータの前処理です。次のコマンドを使用して作成できます uint インデックスのパイプライン：

```
curl -X PUT \
 http://your-elasticsearch.here:9200/_ingest/pipeline/uint-pipeline \
 -H 'content-type:application/json' \
 -d '{
   "description": "daily uint index naming",
   "processors": [
      {
         "date_index_name": {
            "field": "clock",
            "date_formats": [
               "UNIX"
            ],
            "index_name_prefix": "uint-",
            "date_rounding": "d"
         }
      }
   ]
}'
```

ユーザーは、丸めパラメーター (”date_rounding”) を変更して、特定のインデックスローテーション期間を設定できます。他のパイプラインを作成するには、URL（最後の部分はパイプラインの名前）を変更し”index_name_prefix” フィールドをインデックス名と一致するように変更する必要があります。

Elasticsearch ドキュメントも参照してください

さらに、履歴データを複数の日付ベースのインデックスに保存することも、Zabbix サーバー構成の新しいパラメーターで有効にする必要があります。

```
### Option: HistoryStorageDateIndex
# Enable preprocessing of history values in history storage to store values in different indices based on
# 0 - disable
# 1 - enable
#
# Mandatory: no
# Default:
# HistoryStorageDateIndex=0
```

トラブルシューティング

次の手順は Elasticsearch のセットアップに関する問題のトラブルシューティングに役立つ場合があります。

1. マッピングが正しいかどうかを確認します (`http://localhost:9 00/uint` などの必要なインデックス URL への GET リクエスト)。
2. シャードが失敗状態になっていないかどうかを確認します (Elasticsearch の再起動が役立つはずです)。
3. Elasticsearch の設定を確認します。Zabbix フロントエンドホストと Zabbix サーバーホストからのアクセスを許可する必要があります。
4. Elasticsearch ログを確認します。
5. `LogSlowQueries`を使用して、ElasticSearch データベースで遅いクエリを確認できます。

それでもインストールで問題が発生する場合は、このリストのすべての情報 (マッピング、エラーログ、構成、バージョンなど) を含むバグレポートを作成してください。

9 Zabbix 向け Nginx セットアップに関するディストリビューション固有の注意事項

RHEL

RHEL 8、9、10 では、Nginx は AppStream に含まれているため、追加のリポジトリは不要です。以下を実行して有効化し、インストールするだけです。

```
dnf module list nginx
dnf -y install nginx
```

RHEL 7 では、Nginx は CentOS 7 Extras リポジトリ（CentOS を使用している場合）から、以下のようにインストールできます。

```
yum -y install nginx
```

最新のアップストリームビルドを使用したい場合は、公式の NGINX リポジトリを /etc/yum.repos.d/nginx.repo に設定します。

```
cat > /etc/yum.repos.d/nginx.repo << 'EOF'
[nginx-stable]
name=nginx stable repo
baseurl=https://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
EOF
dnf -y install nginx
```

SLES 15

SUSE Linux Enterprise Server 15 では、php-fpm を設定する必要があります（設定ファイルへのパスは、サービスパックによって若干異なる場合があります）。

```
cp /etc/php8/fpm/php-fpm.conf{.default,}
cp /etc/php8/fpm/php-fpm.d/www.conf{.default,}
sed -i 's/user = nobody/user = wwwrun/; s/group = nobody/group = www/' /etc/php8/fpm/php-fpm.d/www.conf
```

10 エージェントを **root** として実行する

Zabbix **5.0.0** 以降、公式パッケージの Zabbix エージェント用 systemd サービスファイルには、User および Group のディレクティブが明示的に含まれています。これらは両方とも zabbix に設定されています。

zabbix_agentd.conf ファイルを介して Zabbix エージェントをどのユーザーで実行するかを設定することは、現在ではできません。これは、エージェントがこの設定を無視し、systemd サービスファイルで指定されたユーザーとして実行されるためです。Zabbix エージェントを root として実行するには、以下で説明する変更を行う必要があります。

Zabbix エージェント

Zabbix エージェントのデフォルトのユーザーとグループをオーバーライドするには、次のコマンドを実行します。

```
systemctl edit zabbix-agent
```

次に以下のコンテンツを追加します。

```
[Service]
User=root
Group=root
```

デーモンをリロードし、zabbix-agent サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Zabbix エージェントの場合、これにより、zabbix_agentd.conf ファイルでユーザーを設定する機能が再度有効になります。次に、エージェントの設定ファイルで User=root および AllowRoot=1 設定パラメーターを設定する必要があります。

Zabbix エージェント 2

Zabbix エージェント 2 のデフォルトのユーザーおよびグループを上書きするには、次を実行します。

```
systemctl edit zabbix-agent2
```

次に、以下の内容を追加します。

```
[Service]
User=root
Group=root
```

デーモンを再読み込みし、zabbix-agent2 サービスを再起動します。

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Zabbix agent2 では、これによって実行ユーザーが完全に決定されます。
追加の変更は必要ありません。

11 Microsoft Windows 上の Zabbix エージェント

エージェントの設定

両世代の Zabbix エージェントは、Windows サービスとして実行されます。Zabbix agent 2 の場合は、以下の手順内の *agentd* を *agent2* に置き換えてください。

Microsoft Windows ホストでは、Zabbix エージェントの単一インスタンスまたは複数インスタンスを実行できます。単一インスタンスでは、次のいずれかを使用できます。

- エージェントのバイナリと同じディレクトリにあるデフォルトの設定ファイル
- コマンドラインで指定した設定ファイル

複数インスタンスの場合、各エージェントインスタンスはそれぞれ専用の設定ファイルを持つ必要があります（いずれか 1 つのインスタンスはデフォルトの設定ファイルを使用できます）。

設定ファイルの例は、Zabbix ソースアーカイブ内で次の場所にあります。

- Zabbix agent の場合は `conf/zabbix_agentd.conf`
- Zabbix agent2 の場合は `conf/zabbix_agent2.conf`

設定ファイルを明示的に指定せずに、アーカイブから Windows 用の Zabbix エージェント/Zabbix agent 2 をサービスとしてインストールする場合は、エージェントをインストールする前に次の作業を行ってください。

- `conf/zabbix_agentd.conf` を、zabbix_agentd.exe がインストールされるディレクトリに手動でコピーする必要があります。
- `conf/zabbix_agent2.conf` および `conf/zabbix_agent2.d` ディレクトリを、zabbix_agent2.exe がインストールされるディレクトリに手動でコピーする必要があります。

Zabbix Windows エージェントの設定の詳細については、設定ファイルのオプションを参照してください。

Hostname パラメータ

ホストでアクティブチェックを実行するには、Zabbix エージェントでホスト名を定義しておく必要があります。さらに、エージェント側で設定するホスト名の値は、Web インターフェースでそのホストに設定された「ホスト名」と完全に一致している必要があります。

エージェント側のホスト名の値は、エージェントの設定ファイル内の **Hostname** または **HostnameItem** パラメータで定義できます。これらのパラメータのいずれかが指定されていない場合は、デフォルト値が使用されます。

HostnameItem パラメータのデフォルト値は、”system.hostname” エージェントキーが返す値です。Windows では、これは gethostname() 関数の結果を返します。この関数は、ローカルホスト名を判定するために名前空間プロバイダーへ問い合わせを行います。どの名前空間プロバイダーからも応答がない場合は、NetBIOS 名が返されます。

Hostname のデフォルト値は、HostnameItem パラメータが返す値です。したがって、これら両方のパラメータが未指定の場合、実際のホスト名はホストの NetBIOS 名になります。Zabbix エージェントは、NetBIOS ホスト名を使用して Zabbix サーバーからアクティブチェックの一覧を取得し、その結果を送信します。

”system.hostname” キーは、*type* と *transform* という 2 つのオプションパラメータをサポートしています。

Type は、アイテムが返す名前の種類を決定します。

- *netbios* (デフォルト) - NetBIOS ホスト名を返します。これは 15 文字までに制限され、英大文字のみです。
- *host* - 大文字と小文字を区別し、完全な実際の Windows ホスト名（ドメインなし）を返します。
- *shorthost* - 最初のドットより前のホスト名部分を返します。名前にドットが含まれない場合は、文字列全体を返します。
- *fqdn* - 完全修飾ドメイン名（末尾のドットなし）を返します。

Transform では、ホスト名に対する追加の変換ルールを指定できます。

- *none* (デフォルト) - 元の大文字・小文字をそのまま使用します。
- *lower* - テキストを小文字に変換します。

そのため、zabbix_agentd.conf ファイルの設定を簡素化し、統一するには、次の 3 つの異なる方法を使用できます。

1. **Hostname** または **HostnameItem** パラメータを未定義のままにし、Zabbix エージェントに NetBIOS ホスト名をホスト名として使用させる。

2. **Hostname** パラメータを未定義のままにし、**HostnameItem** を次のように定義する:
   HostnameItem=system.hostname[host] - Zabbix エージェントに完全な実際の Windows ホスト名（大文字・小文字を区別）をホスト名として使用させます
   HostnameItem=system.hostname[shorthost,lower] - Zabbix エージェントに最初のドットより前のホスト名部分のみを、小文字に変換してホスト名として使用させます。
   HostnameItem=system.hostname[fqdn] - Zabbix エージェントに完全修飾ドメイン名をホスト名として使用させます。

ホスト名は、Windows サービスのインストール、起動、停止、アンインストールに使用される Windows サービス名の一部としても使用されます。たとえば、Zabbix エージェントの設定ファイルに `Hostname=Windows_db_server` が指定されている場合、エージェントは"`Zabbix Agent [Windows_db_server]`" という Windows サービスとしてインストールされます。したがって、各 Zabbix エージェントインスタンスに異なる Windows サービス名を持たせるには、各インスタンスで異なるホスト名を使用する必要があります。

エージェントを Windows サービスとしてインストールする

エージェントをインストールする前に、`conf/zabbix_agentd.conf` を手動で
zabbix_agentd.exe をインストールするディレクトリにコピーしてください。

デフォルトの設定ファイルで Zabbix エージェントの単一インスタンスをインストールするには、次のようにします。

```
zabbix_agentd.exe --install
```

Attention:
64 ビットシステムでは、実行中の 64 ビットプロセスに関連するすべてのチェックを正しく動作させるために、64 ビット版の Zabbix エージェントが必要です。

デフォルト以外の設定ファイルを使用する場合は、サービスのインストールに次のコマンドを使用してください。

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

設定ファイルにはフルパスを指定する必要があります。

Zabbix エージェントの複数インスタンスは、次のようにサービスとしてインストールできます。

```
  zabbix_agentd.exe --config <configuration_file_for_instance_1> --install --multiple-agents
  zabbix_agentd.exe --config <configuration_file_for_instance_2> --install --multiple-agents
  ...
  zabbix_agentd.exe --config <configuration_file_for_instance_N> --install --multiple-agents
```

インストールされたサービスは、コントロールパネルに表示されるようになります。

エージェントの起動

エージェントサービスを開始するには、コントロールパネルを使用するか、コマンドラインから実行します。

デフォルトの設定ファイルで Zabbix エージェントの単一インスタンスを起動するには：

```
 zabbix_agentd.exe --start
```

Zabbix エージェントの単一インスタンスを別の設定ファイルで起動するには：

```
 zabbix_agentd.exe --config <your_configuration_file> --start
```

Zabbix エージェントの複数インスタンスのうち 1 つを開始するには：

```
 zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

エージェントの停止

エージェントサービスを停止するには、コントロールパネルを使用するか、コマンドラインからコマンドを実行します。

デフォルトの設定ファイルで開始された Zabbix エージェントの単一インスタンスを停止するには

```
 zabbix_agentd.exe --stop
```

別の設定ファイルで開始された Zabbix エージェントの単一インスタンスを停止するには

```
 zabbix_agentd.exe --config <your_configuration_file> --stop
```

Zabbix エージェントの複数インスタンスのうち 1 つを停止するには

```
 zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Windows サービスとしてのエージェントのアンインストール

デフォルトの設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには

```
zabbix_agentd.exe --uninstall
```

デフォルト以外の設定ファイルを使用して Zabbix エージェントの単一インスタンスをアンインストールするには

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Windows サービスから Zabbix エージェントの複数のインスタンスをアンインストールするには

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

制限

Windows 用 Zabbix エージェントは、CPU が NUMA ノード全体に不均一に分散されている非標準の Windows 構成をサポートしていません。論理 CPU が不均一に分散されている場合、一部の CPU では CPU パフォーマンスメトリックが使用できない場合があります。たとえば、2 つの NUMA ノードを持つ 72 個の論理 CPU がある場合、両方のノードにそれぞれ 36 個の CPU が必要です。

12 Microsoft Entra ID での SAML 設定

概要

このセクションでは、SAML 2.0 認証を使用して Microsoft Entra ID（旧称 Microsoft Azure Active Directory）から Zabbix へのシングルサインオンとユーザープロビジョニングを構成するためのガイドラインを示します。

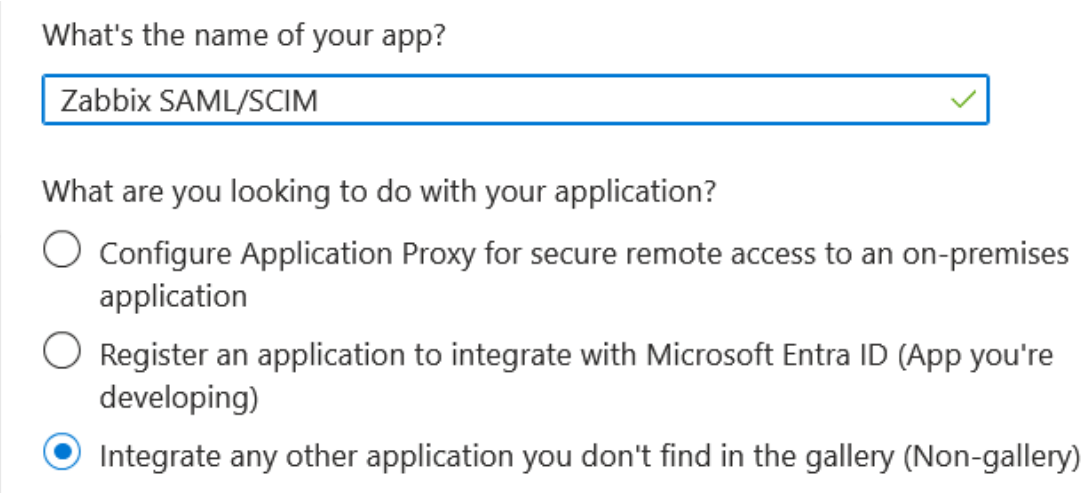
Microsoft Entra ID の設定

アプリケーションの作成

1. Microsoft Entra ID で Microsoft Entra 管理センターにログインします。テスト目的の場合は、Microsoft Entra ID で無料試用アカウントを作成できます。

2. Microsoft Entra 管理センターで、*Applications -> Enterprise applications -> New application -> Create your own application* を選択します。

3. アプリの名前を追加し、*Integrate any other application...* オプションを選択します。その後、*Create* をクリックします。

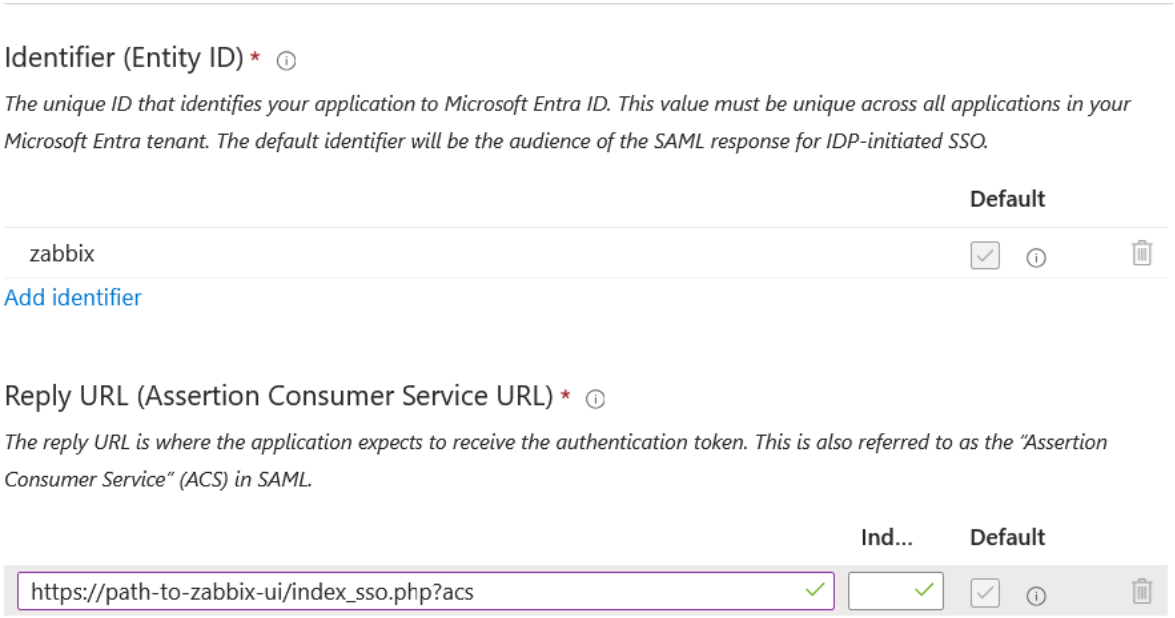


シングルサインオンの設定

1. アプリケーションページで、*Set up single sign on* に移動して *Get started* をクリックします。次に、*SAML* を選択します。

2. *Basic SAML Configuration* を編集します。

- *Identifier (Entity ID)* には、Microsoft Entra ID に対してアプリを識別するための一意の名前を設定します。例えば `zabbix` です。
- *Reply URL (Assertion Consumer Service URL)* には、Zabbix のシングルサインオンエンドポイント `https://<path-to-zabbix-ui>/ir` を設定します。



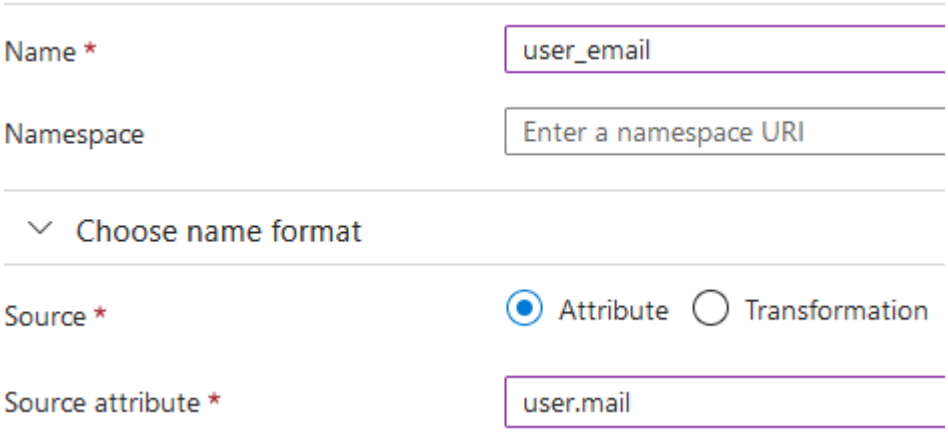
"HTTPS" が必要であることに注意してください。これを Zabbix で機能させるには、`conf/zabbix.conf.php` に次の行を追加する必要があります。

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

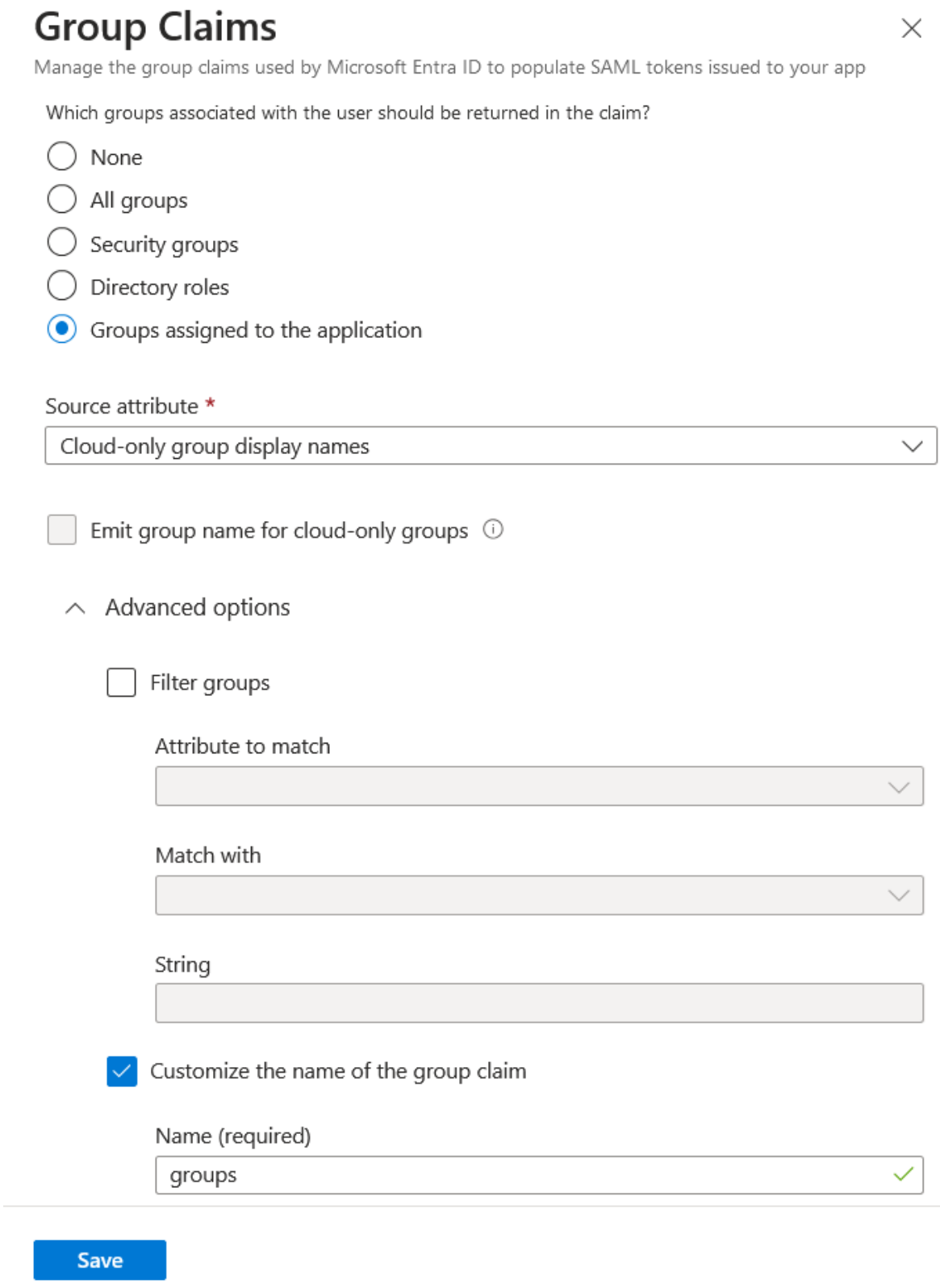
3. *Attributes & Claims* を編集します。Zabbix に渡したいすべての属性（user_name、user_lastname、user_email、user_mobile、groups）を追加する必要があります。

属性名は任意です。異なる属性名を使用することもできますが、Zabbix の SAML 設定内の対応するフィールド値と一致している必要があります。

- 属性を追加するには、*Add new claim* をクリックします。



- Zabbix にグループを渡すための属性を追加するには、*Add a group claim* をクリックします。



この claim では、選択した *Source attribute* によって、グループ ID ではなくグループ名が Zabbix に渡されることが重要です。そうでない場合、JIT ユーザープロビジョニングは正しく機能しません。

4. *SAML Certificates* で、Entra ID が提供する Base64 証明書をダウンロードし、Zabbix Web インターフェースのインストール先にある `conf/certs` に配置します。

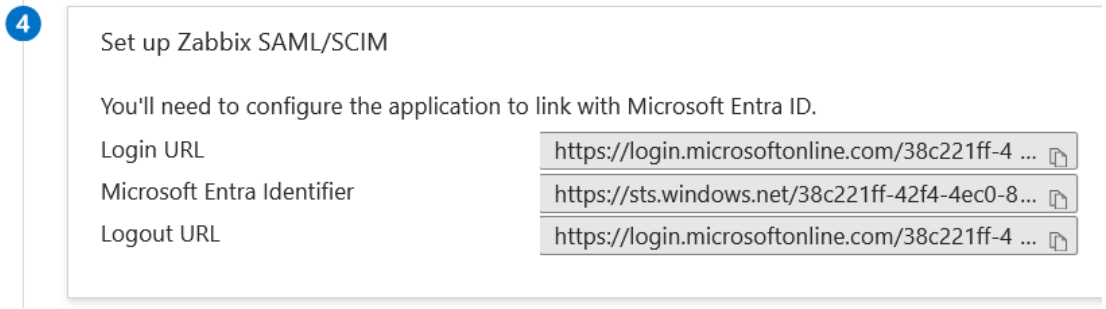
次を実行して、これに 644 の権限を設定します。

```
chmod 644 entra.cer
```

`conf/zabbix.conf.php` に次の行が含まれていることを確認してください。

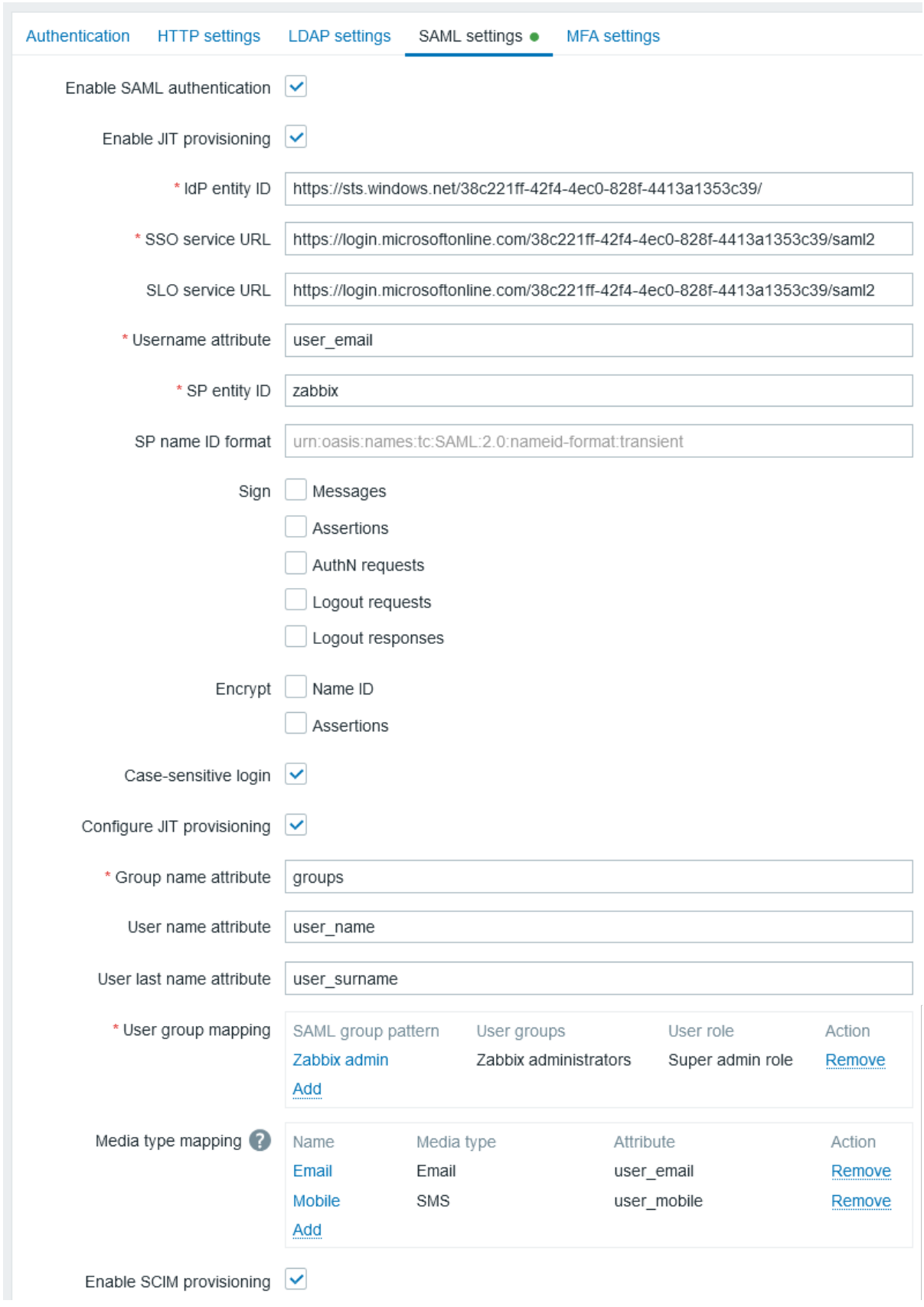
```
$SSO['IDP_CERT'] = 'conf/certs/entra.cer';
```

5. Entra ID の *Set up <your app name>* の値を使用して、Zabbix の SAML 認証を設定します（次のセクションを参照）。



Zabbix の設定

1. Zabbix で、SAML 設定に移動し、Entra ID の設定に基づいて設定オプションを入力します。



Zabbix のフィールド	Entra ID の設定フィールド	サンプル値
IdP entity ID	Microsoft Entra identifier	
SSO service URL	Login URL	
SLO service URL	Logout URL	
SP entity ID	Identifier (Entity ID)	
Username attribute	Custom attribute (claim)	`user_email`
Group name attribute	Custom attribute (claim)	`groups`
User name attribute	Custom attribute (claim)	`user_name`

Zabbix のフィールド	Entra ID の設定フィールド	サンプル値
User last name attribute	Custom attribute (claim)	`user_lastname`

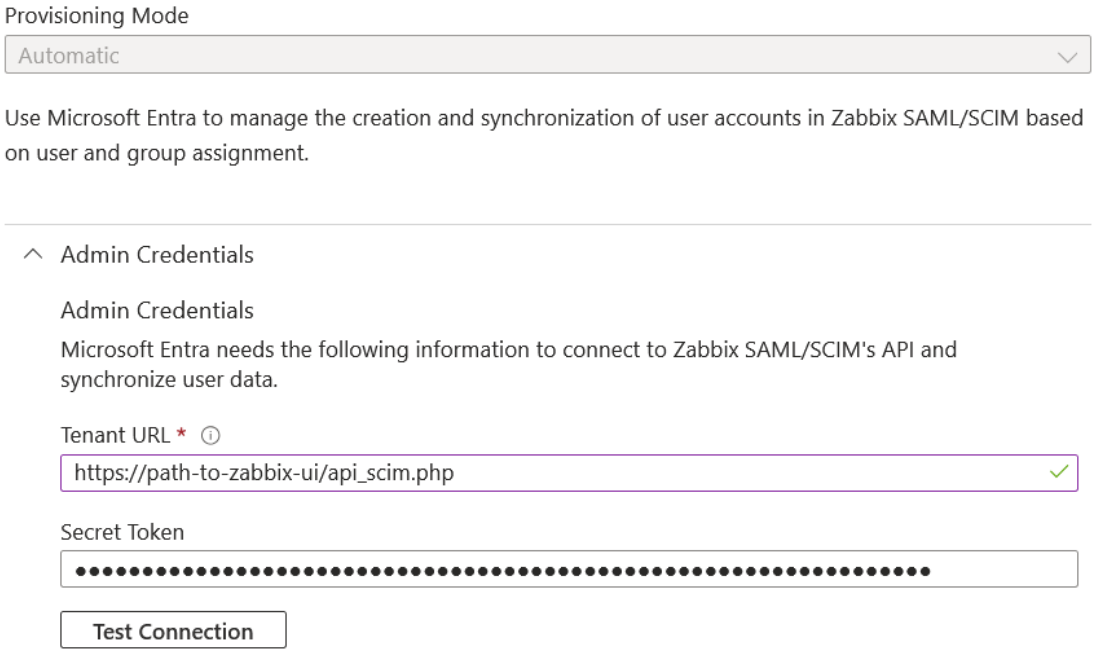
ユーザーグループのマッピングも設定する必要があります。メディアのマッピングは任意です。

これらの設定を保存するには、*Update* をクリックします。

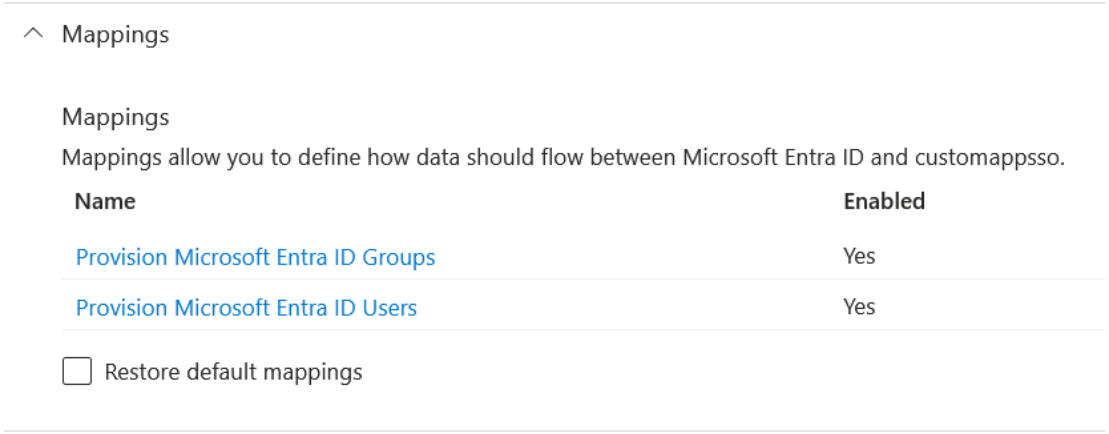
SCIM ユーザープロビジョニング

1. Entra ID アプリケーションのページで、メインメニューから *Provisioning* ページを開きます。*Get started* をクリックし、次に自動プロビジョニングモードを選択します。

- *Tenant URL* には、次の値を設定します: `https://<path-to-zabbix-ui>/api_scim.php`
- *Secret token* には、Super admin 権限を持つ Zabbix API トークンを入力します。
- *Test connection* をクリックして、接続が確立されているか確認します。

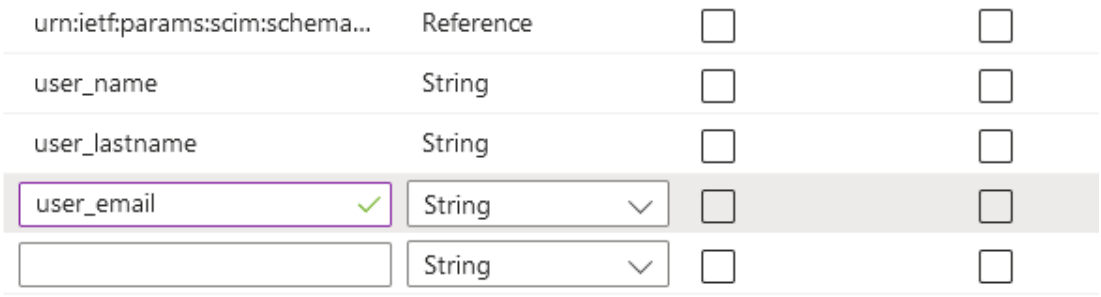


2. これで、SCIM で Zabbix に渡されるすべての属性を追加できます。これを行うには、*Mappings* をクリックし、次に *Provision Microsoft Entra ID Users* をクリックします。



Attribute Mapping リストの下部で、*Show advanced options* を有効にし、次に *Edit attribute list for customappsso* をクリックします。

属性リストの下部で、タイプ'String' の独自属性を追加します。



リストを保存します。

3. 次に、追加した属性のマッピングを追加できます。Attribute Mapping リストの下部で、*Add New Mapping* をクリックし、以下のようにマッピングを作成します。



すべてのマッピングを追加したら、マッピングのリストを保存します。



4. Zabbix へのユーザープロビジョニングの前提条件として、Entra ID でユーザーとグループを設定しておく必要があります。

これを行うには、*Microsoft Entra admin center* に移動し、それぞれの Users ページおよび Groups ページでユーザー/グループを追加します。

5. Entra ID でユーザーとグループを作成したら、アプリケーションの *Users and groups* メニューに移動し、それらをアプリに追加できます。

6. アプリの *Provisioning* メニューに移動し、*Start provisioning* をクリックして、ユーザーを Zabbix にプロビジョニングします。

Entra ID の Users PATCH リクエストでは、メディアの変更はサポートされていないことに注意してください。

認証リクエストの署名

Entra ID を設定して、署名付き認証リクエストの署名を検証できます。

これを機能させるには、公開鍵/秘密鍵を作成します。

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

権限を設定します。

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

以下を追加して、Zabbix Web インターフェースの設定を更新します。

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

トラブルシューティング

ユーザーが SAML 経由で Zabbix にログインしようとする際、MicroSoft Edge のプロファイルですでにログインしている場合、MicroSoft Edge ブラウザーで認証の問題が発生することがあります。この問題の兆候として、ユーザーは MicroSoft Edge のプライベートモードでは Zabbix にログインできる場合があります。

この場合の認証の問題を回避するには、Zabbix Web インターフェースの設定ファイル (*zabbix.conf.php*) で `requestedAuthnContext` を”false” に設定する必要がある場合があります。

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 Okta を使用した **SAML** の設定

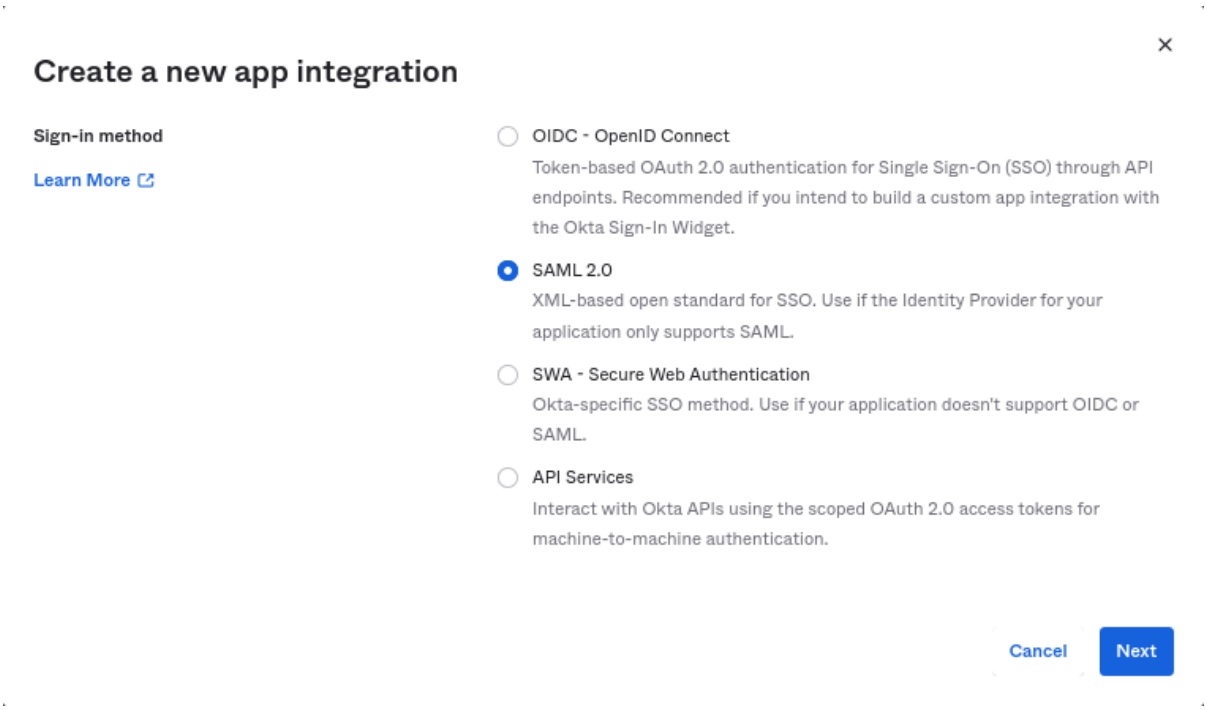
このセクションでは、Zabbix で SAML 2.0 認証およびユーザープロビジョニングを有効にするために、Okta を設定する際のガイドラインを説明します。

Okta の設定

1. https://developer.okta.com/signup/ にアクセスし、アカウントを登録またはサインインします。

2. Okta の Web インターフェースで、*Applications* → *Applications* に移動します。

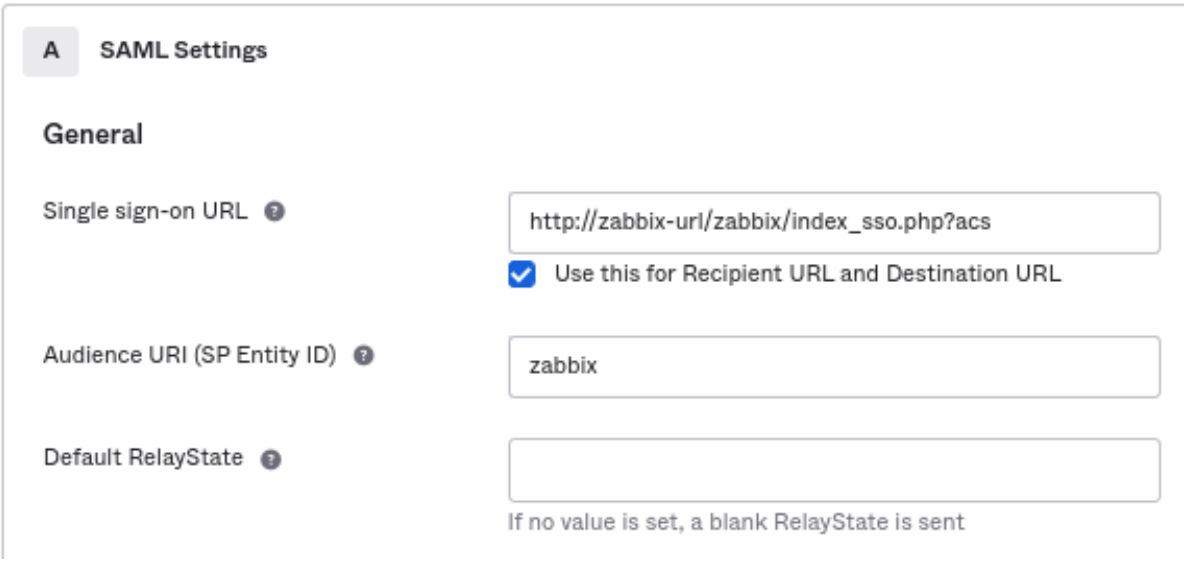
3. *Create App Integration* をクリックします。



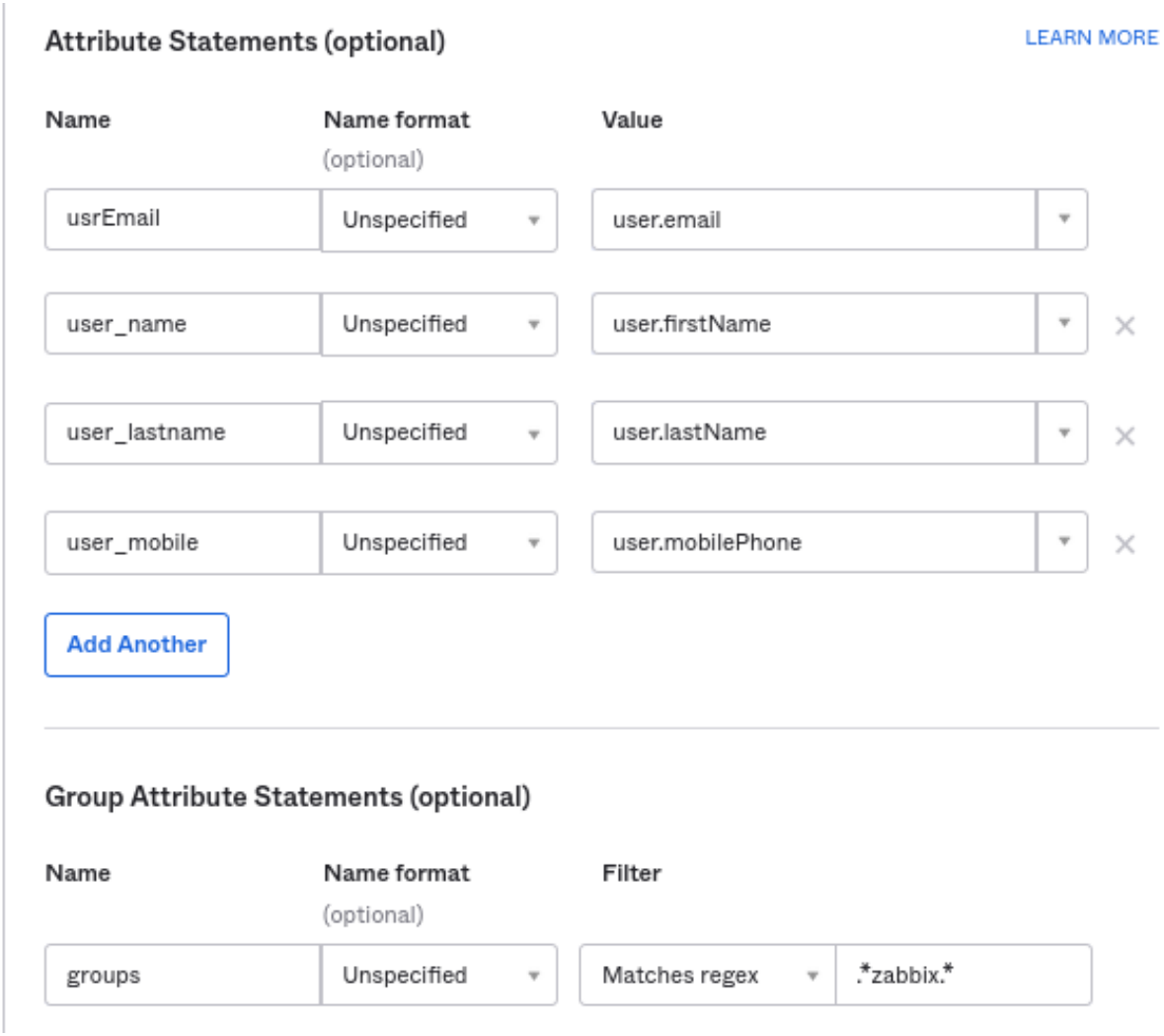
サインイン方法として「SAML 2.0」を選択し、*Next* をクリックします。

4. 一般設定でアプリ名を入力し、*Next* をクリックします。

5. SAML 設定で、以下の値を入力してから *Next* をクリックします。



- **General** で以下を追加します:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs`
    リクエスト内で `acs` パラメータが切り取られないよう、「https」ではなく「http」を使用している点に注意してください。さらに、*Use this for Recipient URL and Destination URL* チェックボックスもオンにする必要があります。
  - *Audience URI (SP Entity ID)*: `zabbix`
    この値は、SAML アサーション内で一意のサービスプロバイダー識別子として使用されます（一致しない場合、処理は拒否されます）。このフィールドには、URL または任意の文字列データを指定できます。
  - *Default RelayState*:
    このフィールドは空欄のままにしてください。カスタムリダイレクトが必要な場合は、Zabbix の *Users* → *Users* 設定で追加できます。
  - その他のフィールドは、必要に応じて設定してください。
- **Attribute Statements/Group Attribute Statements** で以下を追加します:



これらの属性ステートメントは、Zabbix と共有される SAML アサーションに挿入されます。

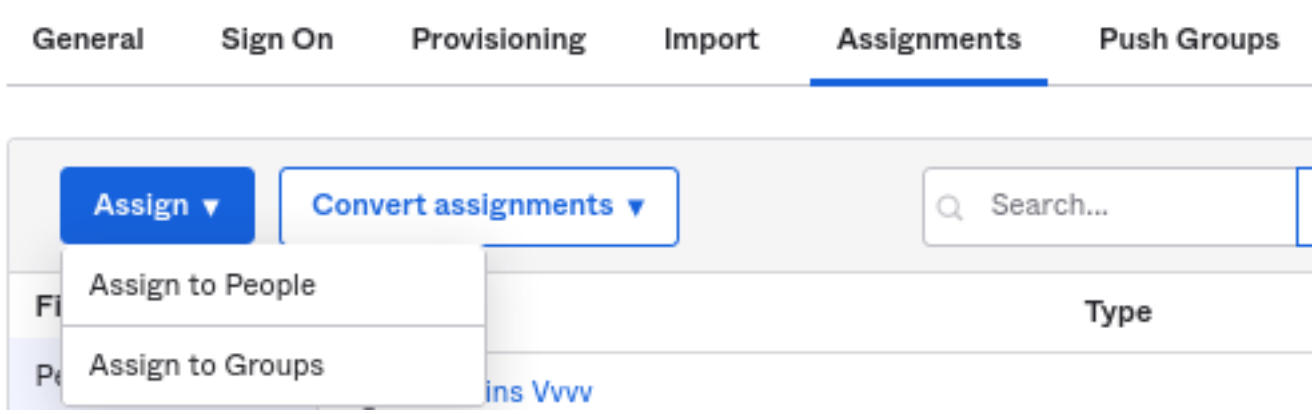
ここで使用している属性名は任意の例です。異なる属性名を使用しても構いませんが、Zabbix の SAML 設定内の対応するフィールド値と一致している必要があります。

JIT ユーザープロビジョニングを使用せずに Zabbix への SAML サインインを設定する場合は、email 属性のみが必要です。

Note:
暗号化接続を使用する予定がある場合は、秘密鍵および公開暗号化証明書を生成し、その後公開証明書を Okta にアップロードしてください。証明書アップロードフォームは、*Assertion Encryption* を「Encrypted」に設定すると表示されます（このパラメータを見つけるには *Show Advanced Settings* をクリックしてください）。

6. 次のタブで、「I'm a software vendor. I'd like to integrate my app with Okta」を選択し、「Finish」を押します。

7. 新しく作成したアプリケーションの「Assignments」タブに移動し、*Assign* ボタンをクリックして、ドロップダウンから「Assign to People」を選択します。



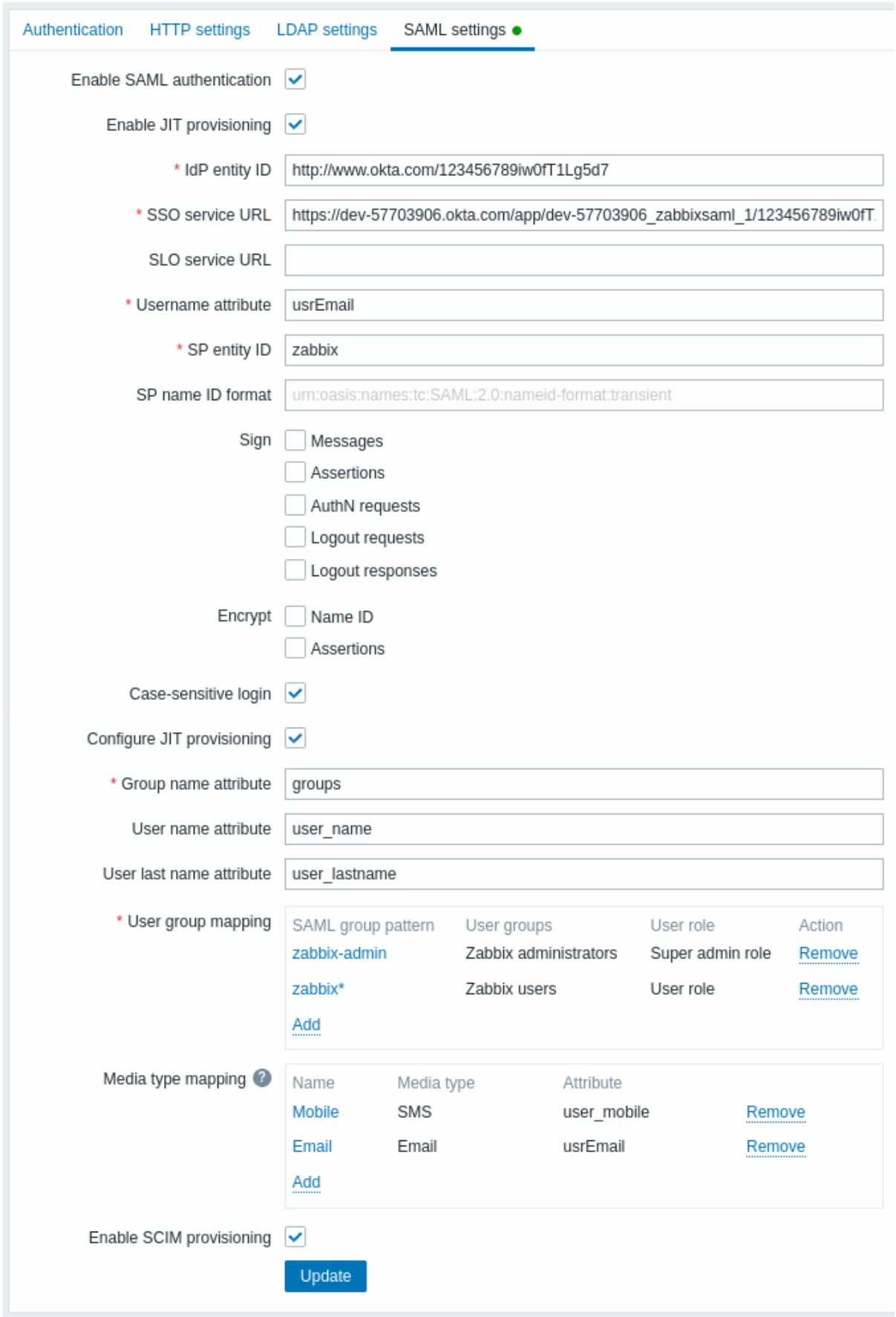
8. 表示されるポップアップで、Zabbix で SAML 2.0 を使用して認証するユーザーにアプリを割り当て、*Save and go back* をクリックします。

9. 「Sign On」タブに移動し、*View Setup Instructions* ボタンをクリックします。

設定手順が新しいタブで開きます。Zabbix の設定中は、このタブを開いたままにしておいてください。

Zabbix の設定

1. Zabbix で、SAML 設定に移動し、Okta のセットアップ手順に基づいて設定オプションを入力します。



Zabbix のフィールド	Okta の設定フィールド	サンプル値
IdP entity ID	Identity Provider Issuer	
SSO service URL	Identity Provider Single Sign-On URL	
Username attribute	Attribute name	`usrEmail`
SP entity ID	Audience URI	`zabbix`
Group name attribute	Attribute name	`groups`

Zabbix のフィールド	Okta の設定フィールド	サンプル値
User name attribute	Attribute name	`user_name`
User last name attribute	Attribute name	`user_lastname`

ユーザーグループおよびメディアのマッピングを設定することも必要です。

2. Okta の SAML セットアップ手順で提供される証明書を、*ui/conf/certs* フォルダに *idp.crt* としてダウンロードします。

次のコマンドを実行して、644 のパーミッションを設定します。

```
chmod 644 idp.crt
```

3. Okta で *Assertion Encryption* が「Encrypted」に設定されている場合は、Zabbix でも *Encrypt* パラメータの「Assertions」チェックボックスをオンにする必要があります。

4. これらの設定を保存するには、「Update」ボタンを押します。

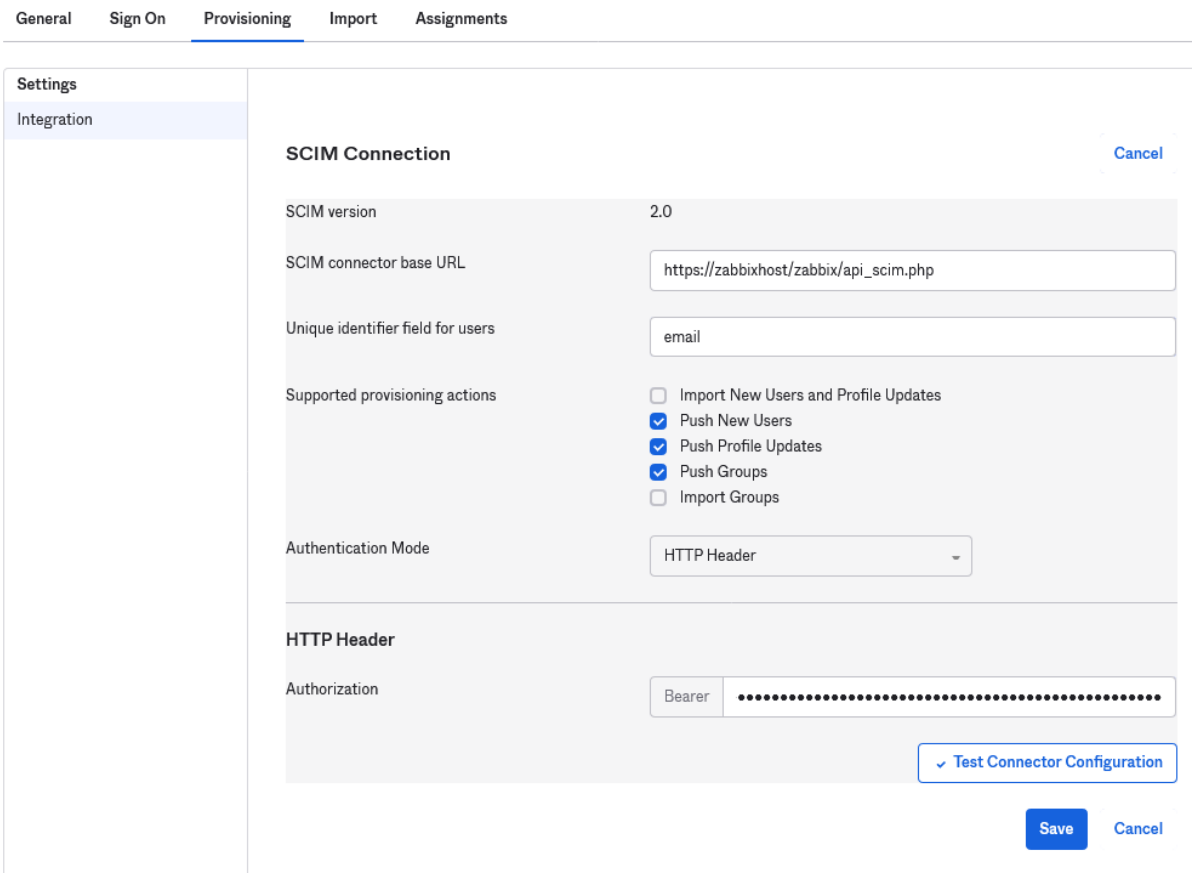
SCIM プロビジョニング

1. SCIM プロビジョニングを有効にするには、Okta でアプリケーションの「General」→「App Settings」に移動します。

Enable SCIM provisioning チェックボックスをオンにします。すると、新しい *Provisioning* タブが表示されます。

2. 「Provisioning」タブに移動して、SCIM 接続を設定します。

- *SCIM connector base URL* には、Zabbix Web インターフェースへのパスを指定し、その末尾に `api_scim.php` を追加します。
  例:
  `https://<your-zabbix-url>/zabbix/api_scim.php`
- *Unique identifier field for users*: `email`
- *Authentication mode*: `HTTP header`
- *Authorization* には、Super admin 権限を持つ有効な API トークンを入力します



Attention:
認証の問題が発生している場合は、Authorization header forwarding を参照してください。

3. *Test Connector Configuration* をクリックして接続をテストします。すべて正しく設定されていれば、成功メッセージが表示されます。

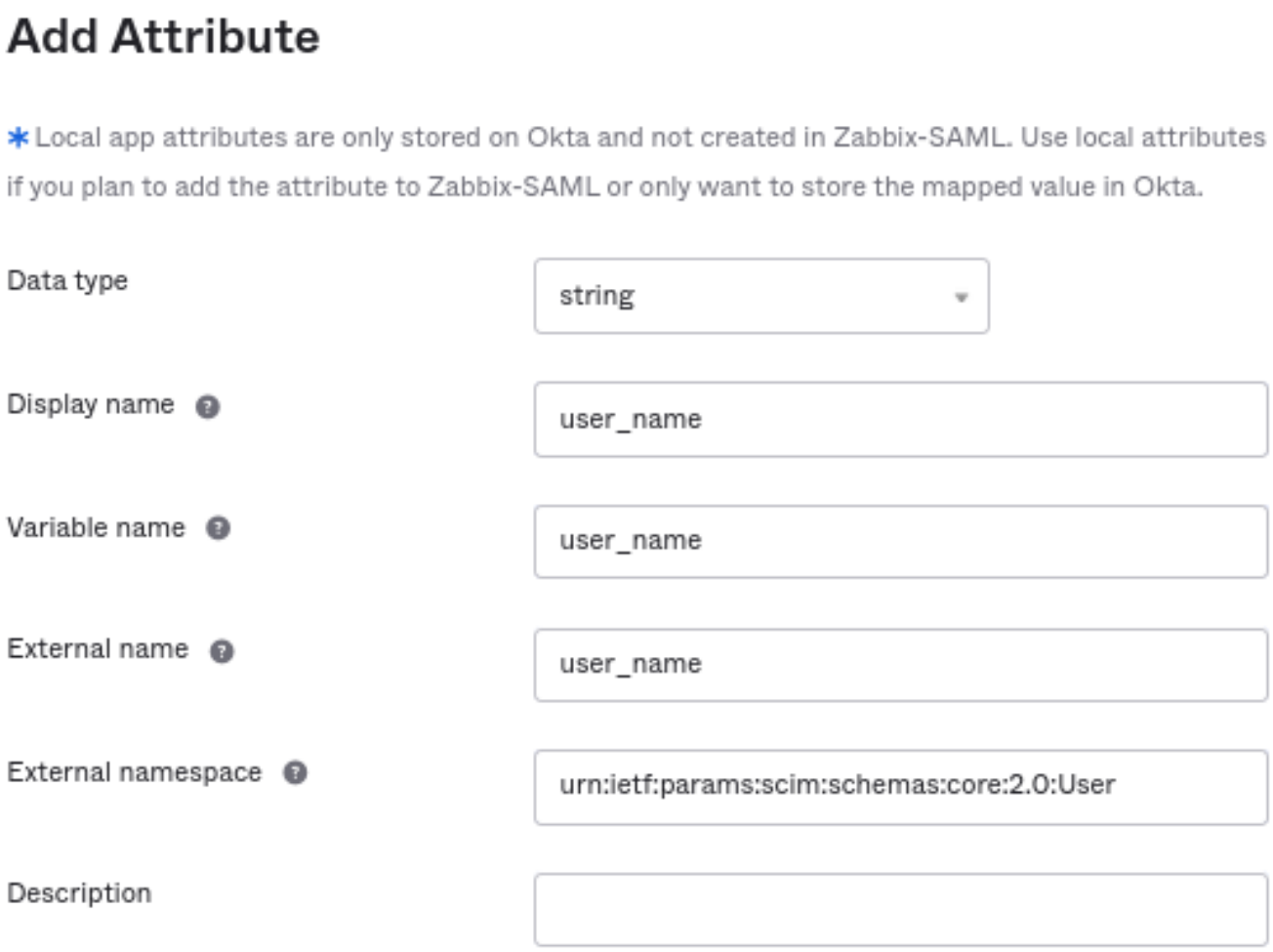
4. 「Provisioning」→「To App」で、以下のチェックボックスがオンになっていることを確認します。

- Create Users
- Update User Attributes
- Deactivate Users

これにより、これらのリクエストタイプが Zabbix に送信されるようになります。

5. SAML で定義されているすべての属性が SCIM でも定義されていることを確認してください。アプリのプロファイルエディターには、「Provisioning」→「To App」で *Go to Profile Editor* をクリックするとアクセスできます。

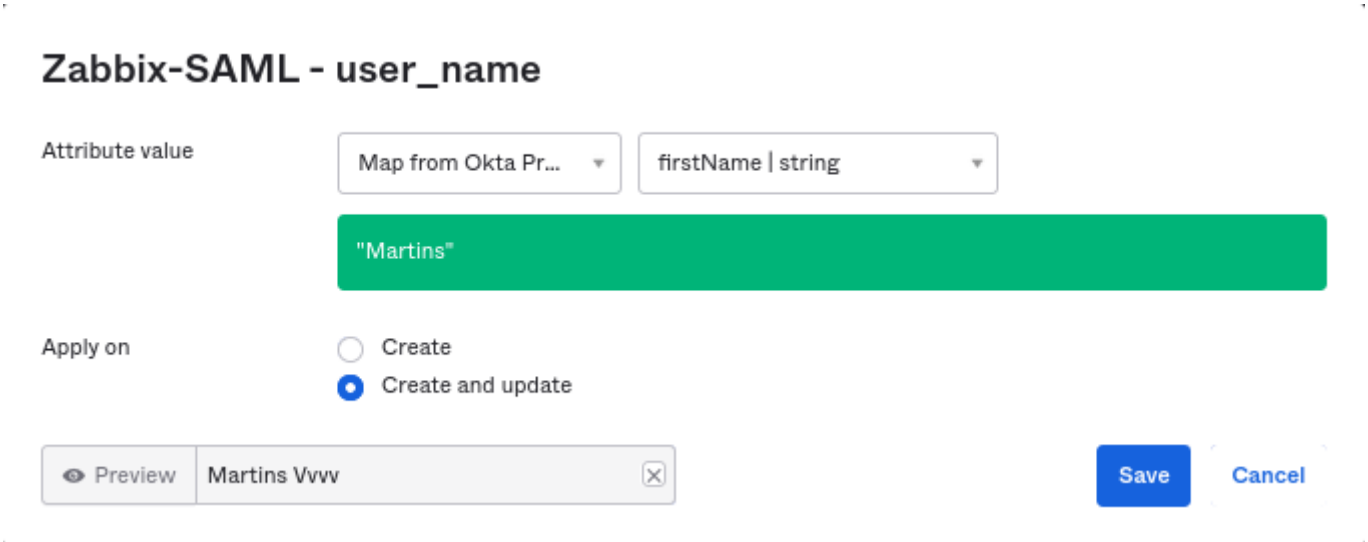
Add Attribute をクリックします。*Display name*、*Variable name*、*External name* には、たとえば `user_name` のように、SAML 属性名を入力します。



External namespace は、ユーザースキーマと同じである必要があります: `urn:ietf:params:scim:schemas:core:2.0:User`

6. アプリケーションの「Provisioning」→「To App」→「Attribute Mappings」に移動します。下部にある *Show Unmapped Attributes* をクリックします。新しく追加した属性が表示されます。

7. 追加した各属性をマッピングします。



8. 「Assignments」タブでユーザーを追加します。これらのユーザーは、あらかじめ *Directory → People* に追加されている必要があります。これらすべての割り当ては、リクエストとして Zabbix に送信されます。

9. 「Push Groups」タブでグループを追加します。Zabbix の SAML 設定におけるユーザーグループマッピングパターンは、ここで指定するグループと一致している必要があります。一致しない場合、ユーザーは Zabbix で作成できません。

グループメンバーに関する情報は、何らかの変更が行われるたびに送信されます。

14 OneLogin を使用した **SAML** の設定

概要

このセクションでは、SAML 2.0 認証を使用して OneLogin から Zabbix へのシングルサインオンとユーザープロビジョニングを構成するためのガイドラインを示します。

OneLogin の設定

アプリケーションの作成

1. OneLogin のアカウントにログインします。テスト目的で、OneLogin で無料の開発者アカウントを作成できます。

2. OneLogin の Web インターフェースで、”アプリケーション”→”アプリケーション” に移動します。

3. ”アプリを追加” をクリックし、適切なアプリを検索します。このページのガイドラインは、*SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)* アプリの例に基づいています。

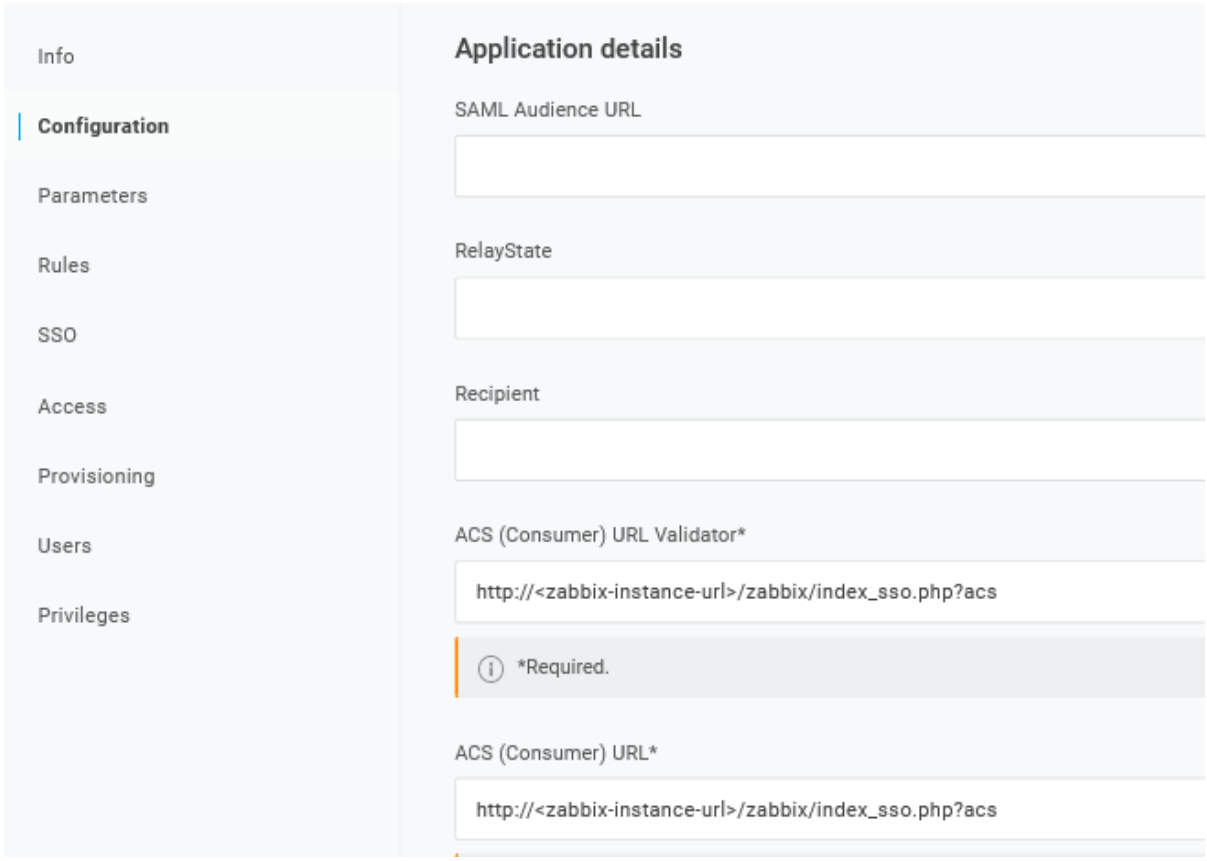
4. まず、アプリの表示名をカスタマイズします。アイコンやアプリの詳細を追加することもできます。その後、保存をクリックします。

SSO/SCIM プロビジョニングの設定

1. *Configuration -> Application details* で、これらのフィールドの値として、Zabbix シングルサインオンエンドポイント `http://<zabbix-instance-url>/zabbix/index_sso.php?acs` を設定します。

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

リクエスト内で acs パラメータが切り取られないようにするため、”https” ではなく ”http” を使用している点に注意してください。



”https” を使用することも可能です。これを Zabbix で動作させるには、`conf/zabbix.conf.php` に次の行を追加する必要があります。

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

その他のオプションはデフォルト値のままにしてください。

2. *Configuration -> API connection* で、次の値を設定します。

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: `user_name`、`user_lastname`、`user_email`、`user_mobile` など、SCIM 経由で Zabbix に渡したいすべてのカスタム属性を含める必要があります。

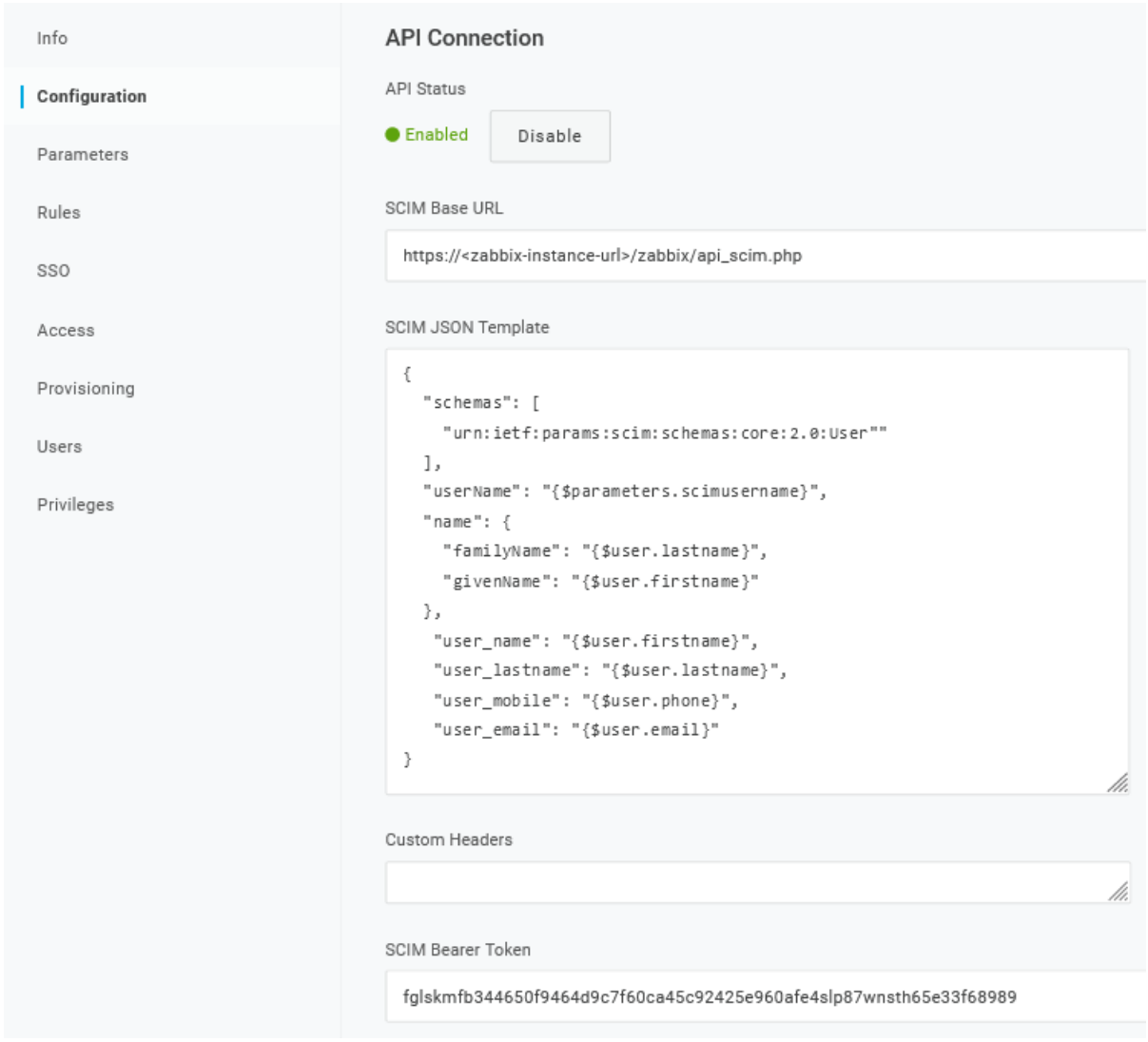
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

属性名は任意です。異なる属性名を使用することもできますが、Zabbix の SAML 設定内の対応するフィールド値と一致している必要があります。

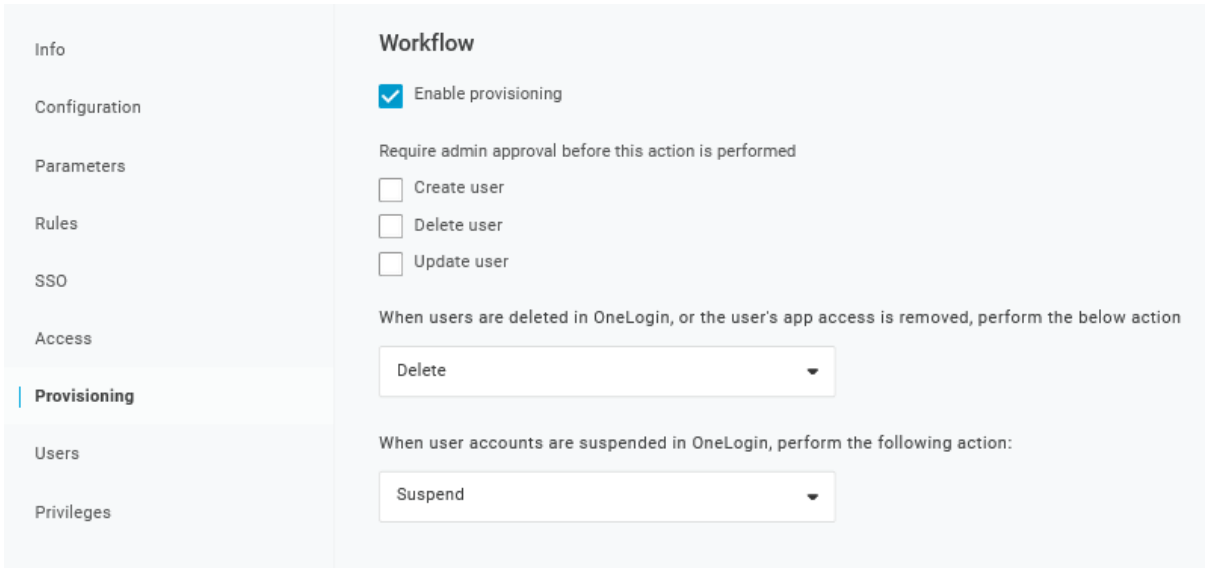
ユーザープロビジョニングを機能させるには、OneLogin がレスポンス内で `givenName` と `familyName` を含む `name` 属性を受け取る必要がある点に注意してください。これは、サービスプロバイダー側で必須でない場合でも必要です。そのため、アプリケーション設定部分のスキーマでこれを指定する必要があります。

- *SCIM Bearer Token*: Super admin 権限を持つ Zabbix API トークンを入力します。

接続を有効化するには、*Enable* をクリックします。

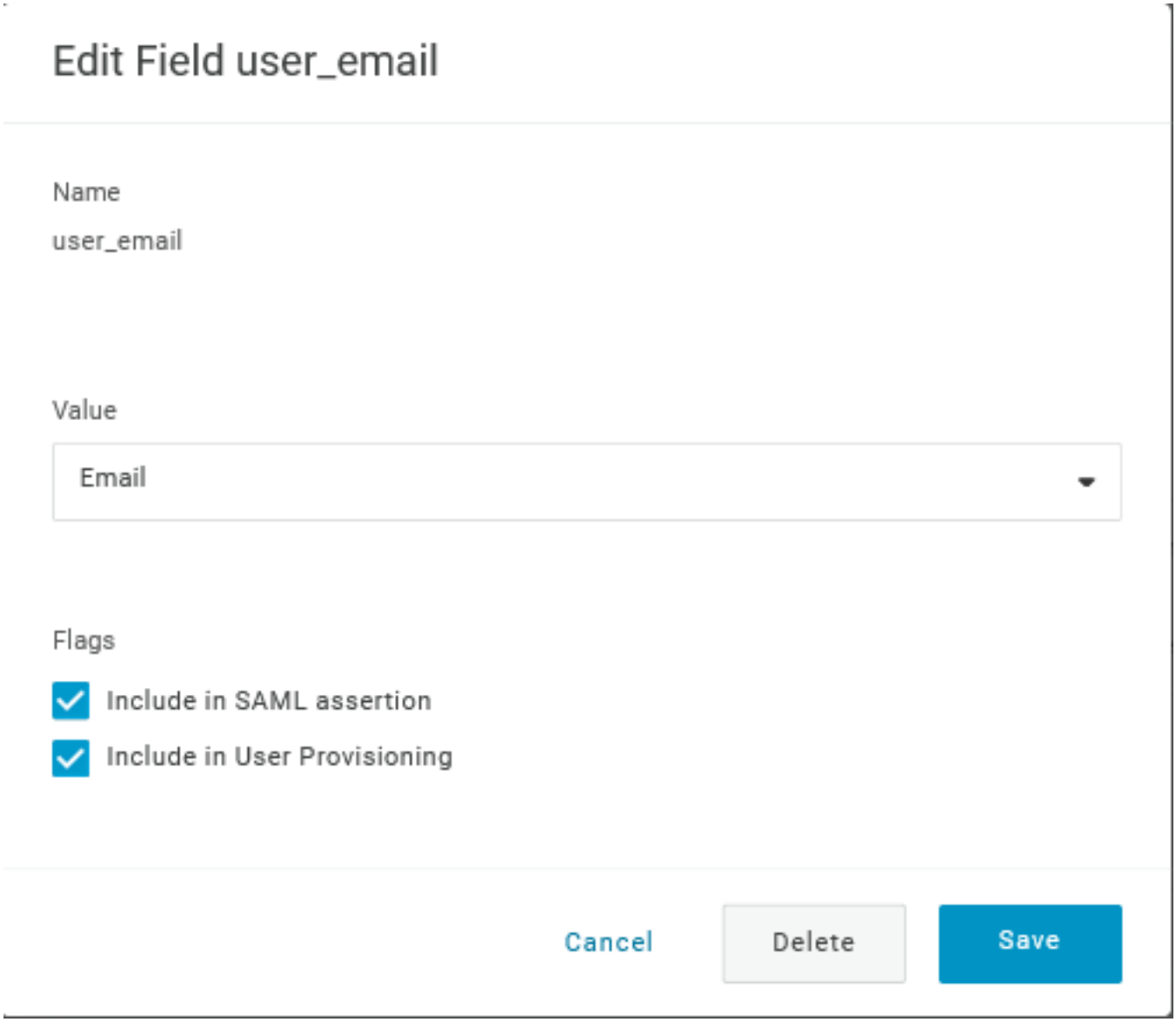


3. *Provisioning* ページで、Provisioning オプションを有効にします。



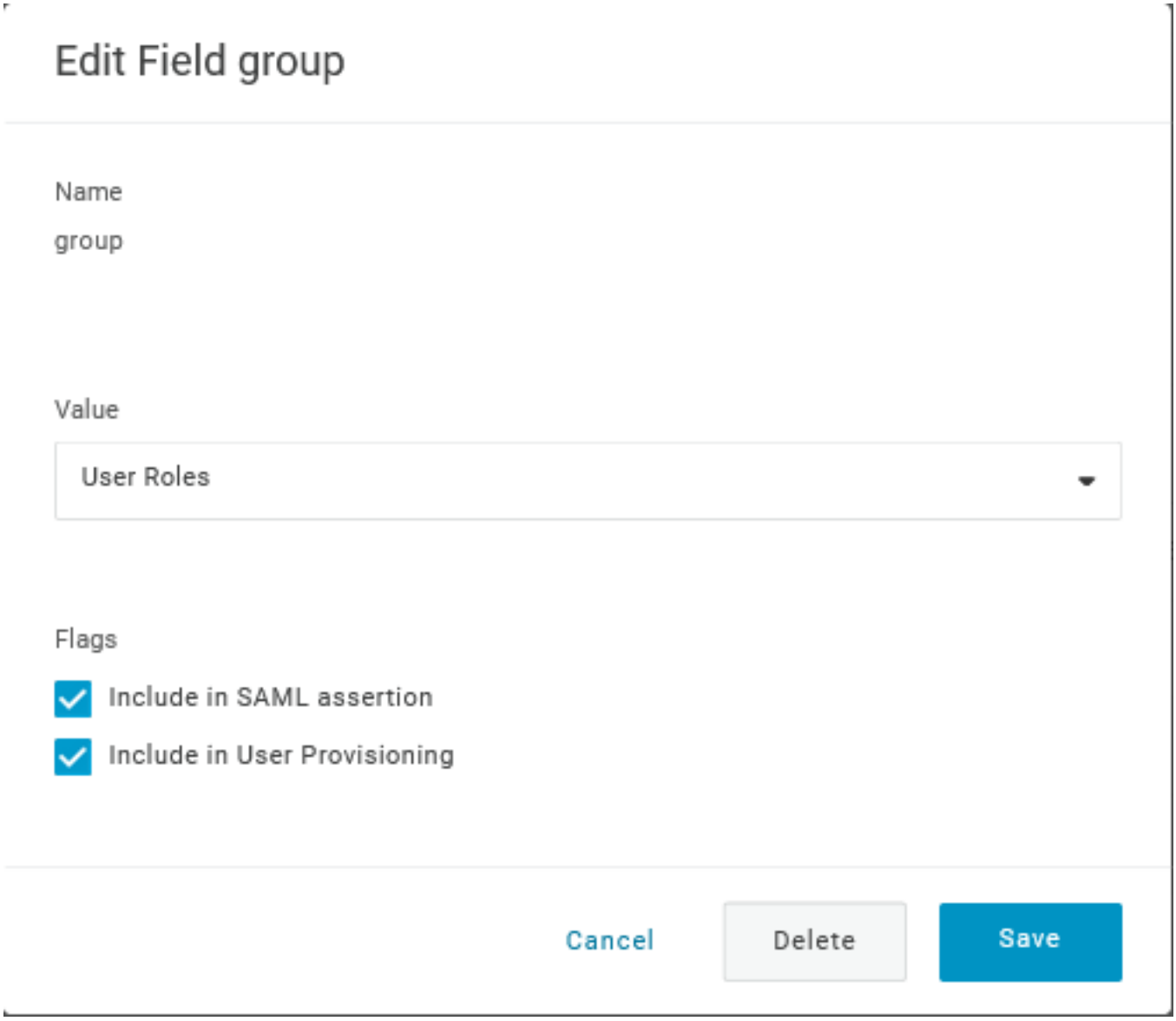
4. *Parameters* ページには、デフォルトパラメータの一覧が表示されます。

- 'scimusername' が OneLogin のユーザーログイン値（例: メールアドレス）と一致していることを確認してください。
- 'Groups' パラメータについて、*Include in User Provisioning* オプションを有効にしてください。
- "+" をクリックして、`user_name`、`user_lastname`、`user_email`、`user_mobile` など、SAML アサーションおよびユーザープロビジョニングに必要なカスタムパラメータを作成します。

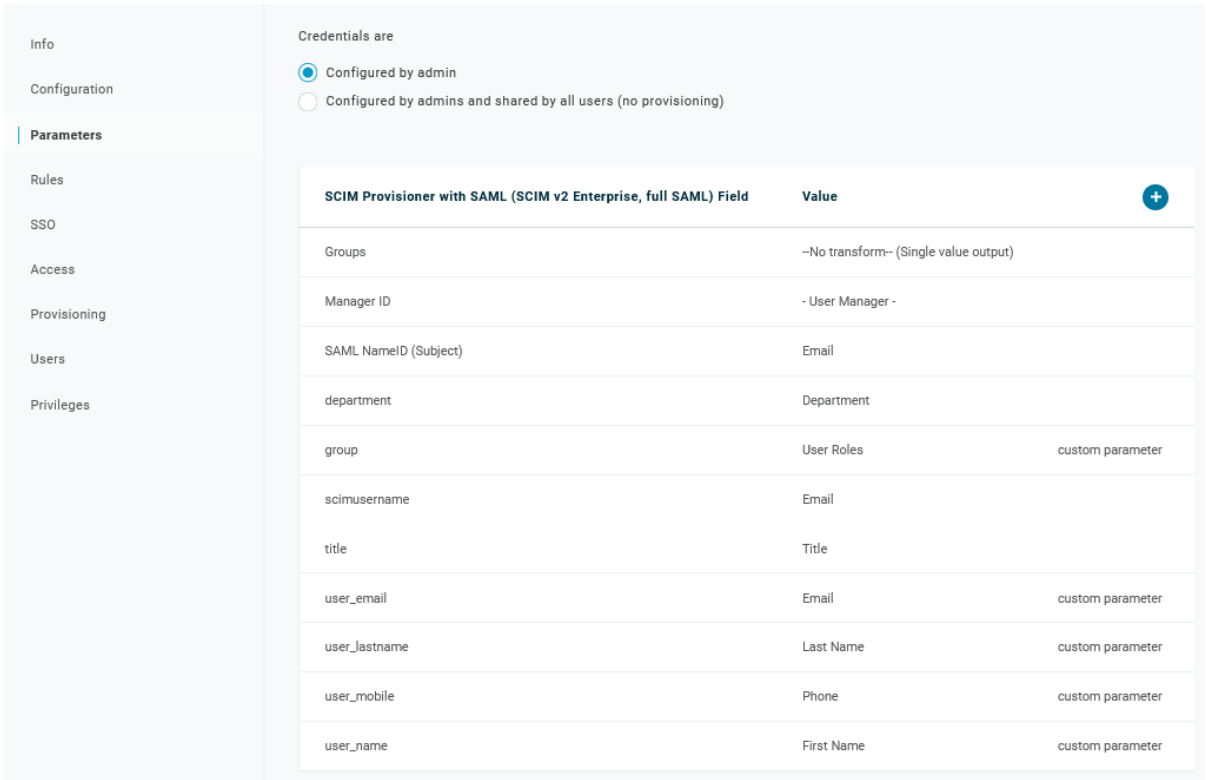


パラメータを追加する際は、*Include in SAML assertion* と *Include in User Provisioning* の両方のオプションを有効にしてください。

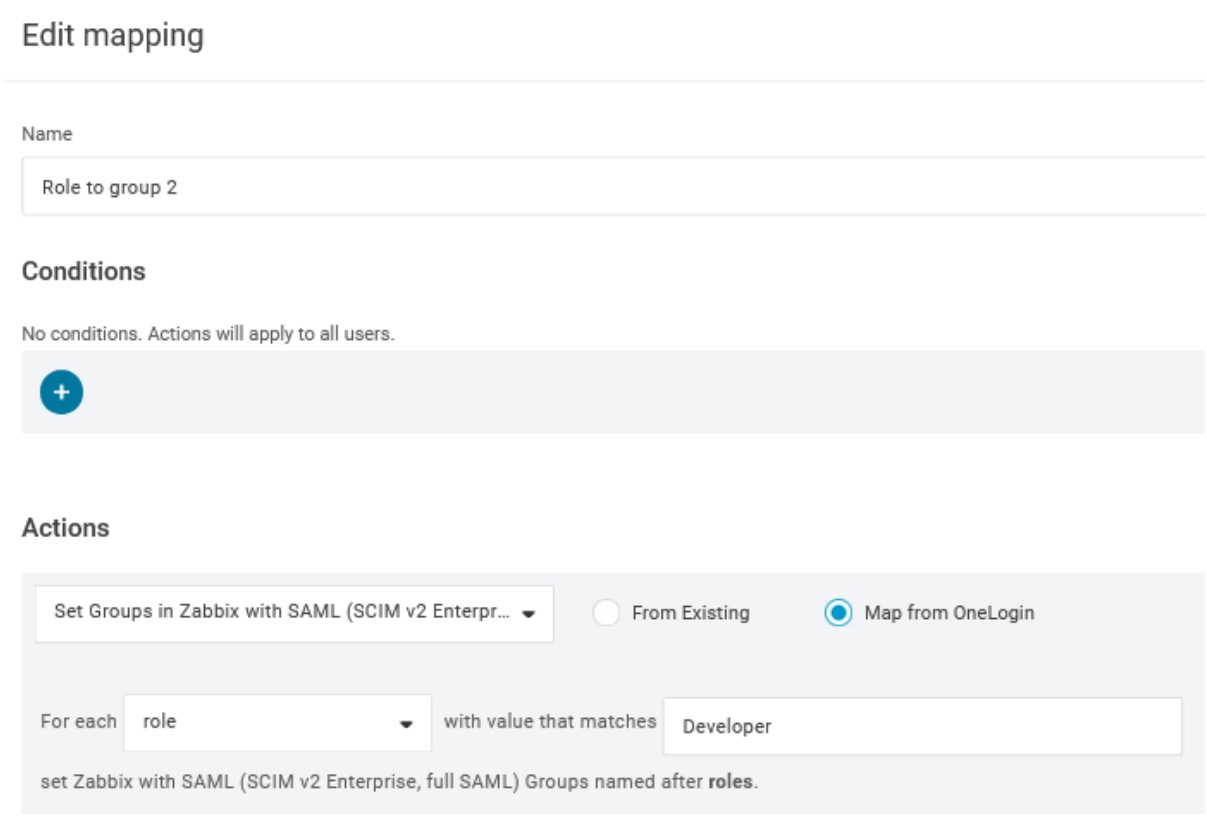
- OneLogin のユーザーロールに一致する'group' パラメータを追加します。ユーザーロールは、セミコロン ; 区切りの文字列として渡されます。OneLogin のユーザーロールは、Zabbix でユーザーグループを作成するために使用されます。



パラメーター一覧を確認してください。



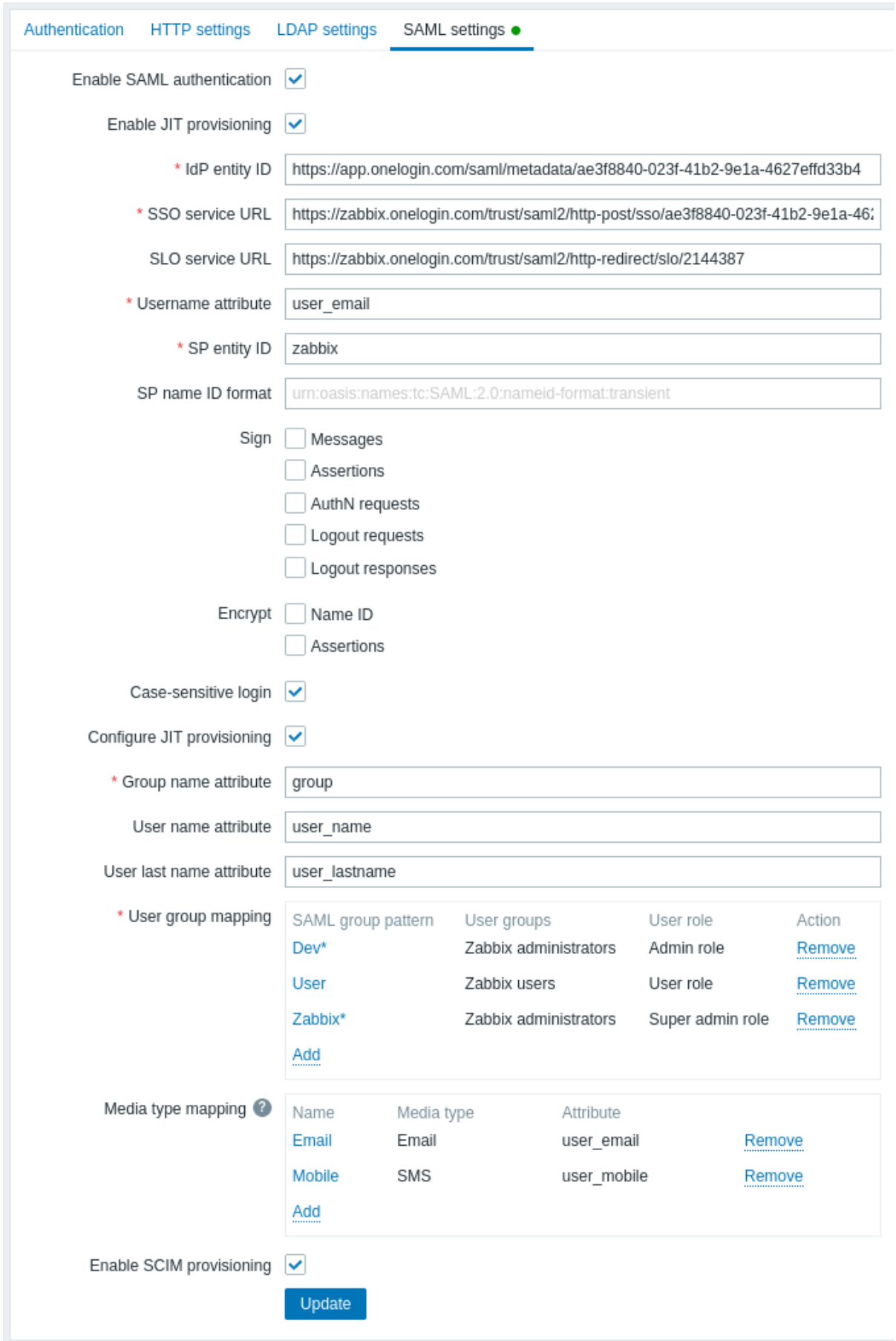
5. *Rules* ページで、デフォルトの Groups パラメータに対するユーザーロールマッピングを作成します。



正規表現を使用して、特定のロールをグループとして渡すことができます。OneLogin は複数のロールを持つ属性を送信する際に区切り文字として ; を使用するため、ロール名には ; を含めないでください。

Zabbix の設定

1. Zabbix で、SAML 設定に移動し、OneLogin の設定に基づいて設定オプションを入力します。



Zabbix field	OneLogin の設定フィールド	サンプル値
IdP entity ID	Issuer URL （OneLogin のアプリケーションの SSO タブを参照）	
SSO service URL	SAML 2.0 Endpoint (HTTP) （OneLogin のアプリケーションの SSO タブを参照）	

Zabbix field	OneLogin の設定フィールド	サンプル値
SLO service URL	SLO Endpoint (HTTP) （OneLogin のアプリケーションの SSO タブを参照）	
Username attribute	カスタムパラメータ	`user_email`
Group name attribute	カスタムパラメータ	`group`
User name attribute	カスタムパラメータ	`user_name`
User last name attribute	カスタムパラメータ	`user_lastname`

ユーザーグループのマッピングも設定する必要があります。メディアのマッピングは任意です。これらの設定を保存するには、*Update* をクリックします。

2. OneLogin が提供する証明書をダウンロードし、Zabbix の Web インターフェースのインストール先にある `conf/certs` に `idp.crt` として配置します。

次のコマンドを実行して、644 の権限を設定します。

```
chmod 644 idp.crt
```

OneLogin では、*Applications* -> *SSO* -> 現在の証明書の *View details* をクリックして、証明書のダウンロードにアクセスできます。

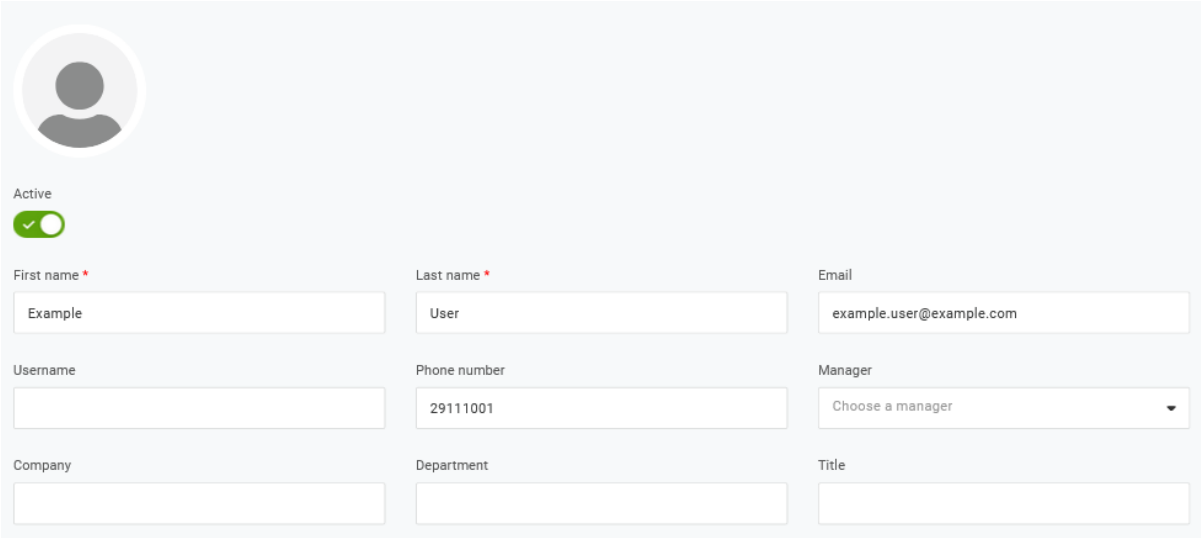
別の証明書名および保存場所を使用することも可能です。その場合は、`conf/zabbix.conf.php` に次の行を追加してください。

```
$SSO['IDP_CERT'] = 'path/to/certname.crt';
```

SCIM ユーザープロビジョニング

ユーザープロビジョニングを有効にすると、OneLogin でユーザーとそのロールを追加/更新し、Zabbix にすぐにプロビジョニングできるようになります。

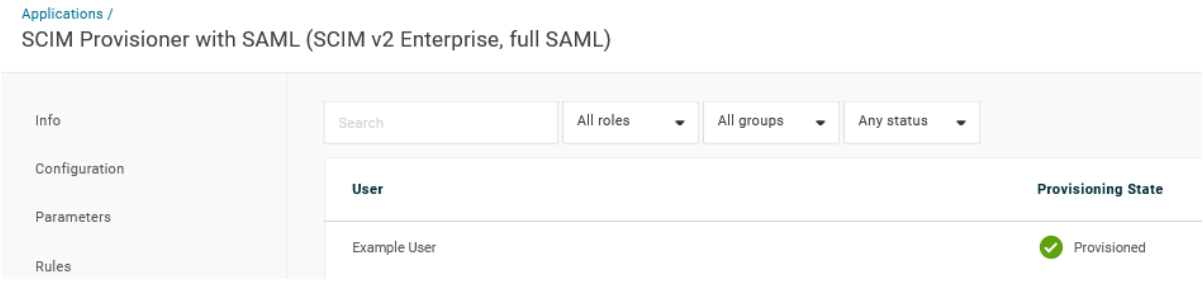
例えば、新しいユーザーを作成します。



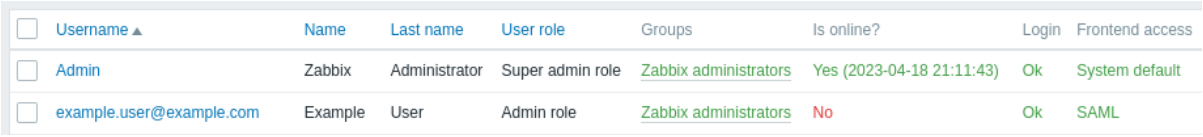
ユーザーロールと、ユーザーをプロビジョニングするアプリケーションを追加します。



ユーザーを保存すると、Zabbix にプロビジョニングされます。アプリケーション → ユーザーで、現在のアプリケーションユーザーのプロビジョニングステータスを確認できます。



プロビジョニングが正常に完了すると、Zabbix ユーザーリストにユーザーが表示されます。



Username ▲	Name	Last name	User role	Groups	Is online?	Login	Frontend access
Admin	Zabbix	Administrator	Super admin role	Zabbix administrators	Yes (2023-04-18 21:11:43)	Ok	System default
example.user@example.com	Example	User	Admin role	Zabbix administrators	No	Ok	SAML

15 スケジュールレポートの設定

概要

このセクションでは、Zabbix web service のインストール方法と、定期レポートの生成を有効にするための Zabbix の設定方法について説明します。

インストール

スケジュールされたレポートの生成を有効にするには、新しいZabbix web service プロセスとGoogle Chrome browser をインストールする必要があります。
web service は、Zabbix サーバーがインストールされている同じマシン、または別のマシンにインストールできます。
Google Chrome browser は、web service がインストールされている同じマシンにインストールする必要があります。

公式の zabbix-web-service パッケージは、Zabbix リポジトリで利用できます。
Google Chrome ブラウザはこれらのパッケージには含まれていないため、別途インストールする必要があります。

ソースから Zabbix web service をコンパイルするには、Installing Zabbix web serviceを参照してください。

インストール後、web service がインストールされているマシンで zabbix_web_service を実行します。

```
zabbix_web_service
```

設定

関連するすべての要素間で適切な通信を確保するため、サーバー設定ファイルおよび Web インターフェースの設定パラメータが正しく設定されていることを確認してください。

Zabbix サーバー

Zabbix サーバー設定ファイル内の以下のパラメータを更新する必要があります: *WebServiceURL* および *StartReportWriters*。

WebServiceURL

このパラメータは、Web サービスとの通信を有効にするために必要です。URL は `http[s]://host:port/report` の形式である必要があります。

- デフォルトでは、Web サービスはポート 10053 で待ち受けます。別のポートは、Web サービスの設定ファイルで指定できます。
- `/report` パスの指定は必須です（このパスはハードコードされており、変更できません）。

例:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
Zabbix サーバーと Zabbix Web サービスの間では、証明書を使用して暗号化を設定することを強く推奨します。デフォルトでは、Zabbix サーバーと Zabbix Web サービスの間で送信されるデータは暗号化されていないため、不正アクセスにつながる可能性があります。

StartReportWriters

このパラメータは、起動するレポートライタープロセスの数を決定します。設定されていない場合、または 0 に等しい場合、レポート生成は無効になります。必要なレポートの数と頻度に応じて、1 から 100 までのレポートライタープロセスを有効にできます。

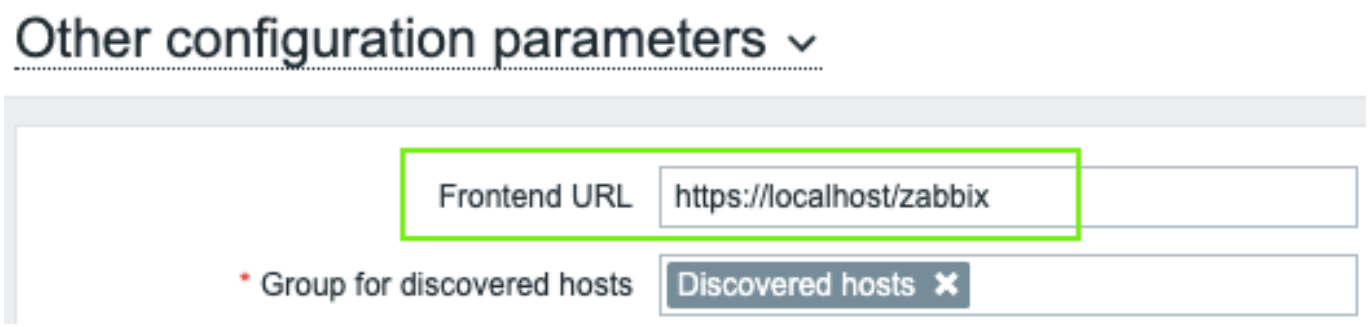
例:

```
StartReportWriters=3
```

Zabbix Web インターフェース

Zabbix Web インターフェースと Zabbix web サービス間の通信を有効にするには、*Frontend URL* パラメータを設定する必要があります。

- Web インターフェースのメニューセクション *Administration* > *General* > *Other* に移動します
- *Frontend URL* パラメータに、Zabbix Web インターフェースの完全な URL を指定します



Note:
セットアップ手順の完了後、すべてが正しく動作していることを確認するために、テストレポートを設定して送信するとよいでしょう。

16 追加の **Web** インターフェース言語

概要

Zabbix Web インターフェースで英語以外の言語を使用するには、そのロケールを Web サーバーにインストールする必要があります。さらに翻訳を機能させるには、PHP の gettext 拡張機能が必要です。

ロケールのインストール

インストールされているすべての言語のリストを表示するには、次のコマンドを実行します。

```
locale -a
```

必要な言語がリストにない場合は*/etc/locale.gen* ファイルを開き、必要なロケールのコメントを解除します。Zabbix は UTF-8 エンコーディングを使用するため、UTF-8 文字セットを使用してロケールを選択する必要があります。

次に以下を実行します。

```
locale-gen
```

Web サーバーを再起動してください。

これでロケールがインストールされました。新しい言語を表示するには、Ctrl+F5 でブラウザ上の Zabbix Web インターフェースページを再読み込みする必要がある場合があります。

Zabbix のインストール

Zabbix git repository から直接 Zabbix をインストールする場合、翻訳ファイルは手動で生成する必要があります。翻訳ファイルを生成するには、次を実行します:

```
make gettext
locale/make_mo.sh
```

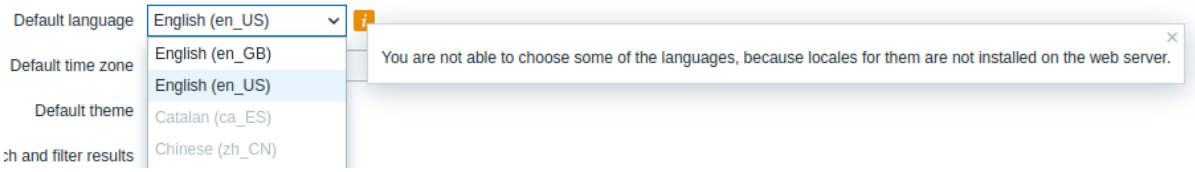
この手順は、パッケージまたはソース tar.gz ファイルから Zabbix をインストールする場合は不要です。

言語選択

Zabbix Web インターフェースで言語を選択する方法はいくつかあります。

- Web インターフェースをインストールする時 - フロントエンドインストールウィザードで選択した言語がシステムデフォルトとして設定されます。
- インストール後、システムのデフォルト言語は管理 → 一般 → *GUI* のメニューセクションで変更できます。
- 特定のユーザーの言語はユーザープロファイルで変更できます。

対象の言語ロケールがマシンにインストールされていない場合、その言語は Zabbix 言語セレクターでグレー表示されます。少なくとも 1 つのロケールが欠落している場合、言語セレクターの横に赤いアイコンが表示されます。このアイコンを押すと、次のメッセージが表示されます: ”Web サーバーにロケールがインストールされていないため、いくつかの言語は利用できません。”



2 プロセスの設定

サイドバーから、参照したい項目を選択してください。

1 Zabbix サーバー

概要

このセクションでは、Zabbix サーバー設定ファイル (zabbix_server.conf) でサポートされるパラメータを一覧表示しています。

パラメータは追加情報なしで一覧表示されています。
詳細を確認するには、パラメータをクリックしてください。

Parameter	Description
AlertScriptsPath	カスタムアラートスクリプトの場所。
AllowRoot	サーバーを’root’ として実行することを許可します。
AllowSoftwareUpdateCheck	Zabbix UI が zabbix.com からソフトウェア更新に関する情報を受信することを許可します。
AllowUnsupportedDBVersions	サポート対象外のデータベースバージョンでサーバーを動作させることを許可します。
CacheSize	設定キャッシュのサイズ。
CacheUpdateFrequency	Zabbix が設定キャッシュの更新を何秒ごとに実行するかを決定するパラメータです。
DBHost	データベースのホスト名。
DBName	データベース名。
DBPassword	データベースのパスワード。
DBPort	ローカルソケットを使用しない場合のデータベースポート。
DBSchema	データベーススキーマ名。PostgreSQL で使用されます。
DBSocket	MySQL ソケットファイルへのパス。
DBUser	データベースユーザー。
DBTLSConnect	このオプションを指定した値に設定すると、データベースへの TLS 接続の使用を強制します。
DBTLSCAFile	データベース証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。
DBTLSCertFile	データベースへの認証に使用する Zabbix サーバー証明書を含むファイルのフルパス名。
DBTLSKeyFile	データベースへの認証に使用する秘密鍵を含むファイルのフルパス名。
DBTLSCipher	TLS v1.2 までの TLS プロトコルで Zabbix サーバーが許可する暗号スイートの一覧。MySQL でのみサポートされます。
DBTLSCipher13	TLS v1.3 プロトコルで Zabbix サーバーが許可する暗号スイートの一覧。MySQL でのみ、バージョン 8.0.16 以降でサポートされます。
DebugLevel	デバッグレベルを指定します。
EnableGlobalScripts	Zabbix サーバーでグローバルスクリプトを有効にします。
ExportDir	イベント、履歴、トレンドを改行区切り JSON 形式でリアルタイムエクスポートするためのディレクトリ。設定すると、リアルタイムエクスポートが有効になります。
ExportFileSize	エクスポートファイル 1 つあたりの最大サイズ（バイト単位）。
ExportType	リアルタイムエクスポートのための、カンマ区切りのエンティティタイプ（events、history、trends）の一覧（デフォルトではすべてのタイプ）。
ExternalScripts	外部スクリプトの場所。
Fping6Location	fping6 の場所。
FpingLocation	fping の場所。
FrontendAllowedIP	Web インターフェースからの接続を許可する IP アドレスまたは CIDR 範囲のカンマ区切り一覧。
HANodeName	高可用性クラスターノード名。
HistoryCacheSize	履歴キャッシュのサイズ。
HistoryIndexCacheSize	履歴インデックスキャッシュのサイズ。
HistoryStorageDateIndex	履歴ストレージ内の履歴値の前処理を有効にし、日付に基づいて異なるインデックスに値を保存します。

Parameter	Description
HistoryStorageURL	履歴ストレージの HTTP[S] URL。
HistoryStorageTypes	履歴ストレージに送信する値タイプのカンマ区切り一覧。
HousekeepingFrequency	Zabbix がハウスキーピング処理を何時間ごとに実行するかを決定するパラメータです。
Include	設定ファイルに個別のファイル、またはディレクトリ内のすべてのファイルを含めることができます。
JavaGateway	Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。
JavaGatewayPort	Zabbix Java ゲートウェイが待ち受けるポート。
ListenBacklog	TCP キュー内の保留中接続の最大数。
ListenIP	トラッパーが待ち受ける IP アドレスのカンマ区切り一覧。
ListenPort	トラッパーの待ち受けポート。
LoadModule	サーバー起動時に読み込むモジュール。
LoadModulePath	サーバーモジュールの場所へのフルパス。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogSlowQueries	データベースクエリがログに記録されるまでに許容される時間をミリ秒単位で決定します。
LogType	ログ出力のタイプ。
MaxConcurrentChecksPerPoller	各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが一度に実行できる非同期チェックの最大数。
MaxHousekeeperDelete	1 回のハウスキーピングサイクル内の 1 タスクあたり、'MaxHousekeeperDelete' 行（[tablename]、[field]、value に対応）を超えて削除されることはありません。
NodeAddress	Web インターフェースがサーバーへ接続する方法を上書きするための、オプションのポート付き IP またはホスト名。
PidFile	PID ファイル名。
ProblemHousekeepingFrequency	削除されたトリガーに対する障害を Zabbix がどのくらいの頻度で削除するかを決定します。
ProxyConfigFrequency	Zabbix サーバーが Zabbix プロキシに設定データを送信する頻度を決定します。
ProxyDataFrequency	Zabbix サーバーが Zabbix プロキシから履歴データを要求する頻度を決定します。
ServiceManagerSyncFrequency	Zabbix がサービスマネージャーの設定を同期する頻度を決定します。
SMSDevices	Zabbix サーバーでの使用を許可するモデムファイルのカンマ区切り一覧。
SNMPTrapperFile	SNMP トラップデーモンからサーバーへデータを渡すために使用される一時ファイル。
SocketDir	内部 Zabbix サービスで使用される IPC ソケットを保存するディレクトリ。
SourceIP	送信元 IP アドレス。
SSHKeyLocation	SSH チェックおよびアクション用の公開鍵と秘密鍵の場所。
SSLCertLocation	クライアント認証用の SSL クライアント証明書ファイルの場所。
SSLKeyLocation	クライアント認証用の SSL 秘密鍵ファイルの場所。
SSLCALocation	SSL サーバー証明書検証用の認証局 (CA) ファイルの場所を上書きします。
StartAgentPollers	非同期 Zabbix エージェントポーラーの事前 fork されたインスタンス数。
StartAlerters	アラート送信プロセスの事前 fork されたインスタンス数。
StartBrowserPollers	ブラウザアイテムポーラーの事前 fork されたインスタンス数。
StartConnectors	コネクターワーカーの事前 fork されたインスタンス数。
StartDBSyncers	履歴同期プロセスの事前 fork されたインスタンス数。
StartDiscoverers	ディスカバリワーカーの事前 fork されたインスタンス数。
StartEscalators	エスカレーターの事前 fork されたインスタンス数。
StartHistoryPollers	履歴ポーラーの事前 fork されたインスタンス数。
StartHTTPAgentPollers	非同期 HTTP エージェントポーラーの事前 fork されたインスタンス数。
StartHTTPPollers	HTTP ポーラーの事前 fork されたインスタンス数。
StartIPMIPollers	IPMI ポーラーの事前 fork されたインスタンス数。
StartJavaPollers	Java ポーラーの事前 fork されたインスタンス数。
StartLLDProcessors	ローレベルディスカバリ (LLD) ワーカーの事前 fork されたインスタンス数。
StartODBCPollers	ODBC ポーラーの事前 fork されたインスタンス数。
StartPingers	ICMP pinger の事前 fork されたインスタンス数。
StartPollersUnreachable	到達不能なホスト用ポーラー（IPMI および Java を含む）の事前 fork されたインスタンス数。
StartPollers	ポーラーの事前 fork されたインスタンス数。
StartPreprocessors	前処理ワーカーの事前起動されるインスタンス数。
StartProxyPollers	パッシブプロキシ用ポーラーの事前 fork されたインスタンス数。
StartReportWriters	レポートライターの事前 fork されたインスタンス数。
StartSNMPPollers	非同期 SNMP ポーラーの事前 fork されたインスタンス数。
StartSNMPTrapper	1 に設定すると、SNMP トラッパープロセスが起動されます。
StartTimers	タイマーの事前 fork されたインスタンス数。
StartTrappers	トラッパーの事前 fork されたインスタンス数。
StartVMwareCollectors	VMware コレクターの事前 fork されたインスタンス数。
StatsAllowedIP	外部 Zabbix インスタンスの IP アドレス（CIDR 表記も可）または DNS 名のカンマ区切り一覧。統計リクエストは、ここに一覧表示されたアドレスからのみ受け付けられます。

Parameter	Description
Timeout	Zabbix プロキシ、エージェント、Web サービスとの接続確立およびデータ交換、ならびに SNMP チェック（SNMP `walk[OID]` および `get[OID]` アイテムを除く）に対して待機する時間（秒）を指定します。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	サーバー証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	GnuTLS 優先順位文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書ベースおよび PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。証明書ベースおよび PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS 優先順位文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。証明書ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS 優先順位文字列または OpenSSL (TLS 1.2) 暗号文字列。PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSFrontendAccept	Web インターフェースから受け入れる受信接続の種類。
TLSFrontendCertIssuer	許可される Web インターフェース証明書の発行者。
TLSFrontendCertSubject	許可される Web インターフェース証明書のサブジェクト。
TLSKeyFile	サーバー秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSListen	トラッパーソケットでの TLS を制御します。
TmpDir	一時ディレクトリ。
TrapperTimeout	以下に対するタイムアウトを秒単位で指定します: - Zabbix プロキシからの履歴データ取得; - Zabbix プロキシへの設定データ送信; - Zabbix サーバー上でのグローバルスクリプト実行またはリモートコマンド実行。
TrendCacheSize	トレンドキャッシュのサイズ。
TrendFunctionCacheSize	トレンド関数キャッシュのサイズ。
UnavailableDelay	利用不可期間中にホストの可用性を確認する頻度を決定します。
UnreachableDelay	到達不能期間中にホストの可用性を確認する頻度を決定します。
UnreachablePeriod	ホストを利用不可として扱うまでの到達不能時間（秒）を決定します。
User	システム上の特定の既存ユーザーに権限を落とします。
ValueCacheSize	履歴値キャッシュのサイズ。
Vault	vault プロバイダーを指定します。
VaultDBPath	キーによってデータベース認証情報を取得する場所を指定します。
VaultPrefix	vault パスまたはクエリ用のカスタムプレフィックス。
VaultTLSCertFile	クライアント認証に使用する SSL 証明書ファイル名。
VaultTLSKeyFile	クライアント認証に使用する SSL 秘密鍵ファイル名。
VaultToken	HashiCorp vault 認証トークン。
VaultURL	vault サーバーの HTTP[S] URL。
VMwareCacheSize	VMware データ保存用の共有メモリサイズ。
VMwareFrequency	単一の VMware サービスからデータを収集する間隔（秒）。
VMwarePerfFrequency	単一の VMware サービスからパフォーマンスカウンター統計を取得する間隔（秒）。
VMwareTimeout	vmware collector が VMware サービスからの応答を待機する最大秒数。
WebDriverURL	WebDriver インターフェースの HTTP[S] URL。
WebServiceURL	<host:port>/report 形式の Zabbix web service への HTTP[S] URL。

すべてのパラメータは、明示的に必須と記載されていない限り、必須ではありません。

注意:

- デフォルト値はデーモンのデフォルトを反映しており、同梱される設定ファイル内の値ではありません。
- 値ではenvironment variables をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

AlertScriptsPath

カスタムアラートスクリプトの場所（コンパイル時のインストール変数 *datadir* に依存します）

デフォルト: `/usr/local/share/zabbix/alertscripts`

AllowRoot

サーバーを'root' として実行できるようにします。
無効で、かつサーバーが'root' によって起動された場合、サーバーは代わりに'zabbix' ユーザーへ切り替えようとします。
通常のユーザーで起動された場合は、効果はありません。

デフォルト: 0
 値: 0 - 許可しない; 1 - 許可する

AllowSoftwareUpdateCheck

Zabbix UI が zabbix.com からソフトウェアアップデートに関する情報を受信することを許可します。

デフォルト: 1
 値: 0 - 許可しない、1 - 許可する

AllowUnsupportedDBVersions

サーバーがサポートされていないデータベースバージョンで動作することを許可します。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

CacheSize

設定キャッシュのサイズ（バイト単位）。ホスト、アイテム、トリガーのデータを保存するための共有メモリのサイズです。

デフォルト: 32M
 範囲: 128K-64G

CacheUpdateFrequency

このパラメータは、Zabbix が設定キャッシュの更新を何秒ごとに実行するかを決定します。runtime control オプションも参照してください。

デフォルト: 10
 範囲: 1-3600

DBHost

データベースのホスト名。
MySQL では、`localhost` または空文字列を指定するとソケットが使用されます。PostgreSQL では、空文字列に設定するとデフォルトの Unix ドメインソケットが使用されます。パス（例: /var/run/pgbouncer）に設定すると、指定した Unix ドメインソケットが使用されます。

デフォルト: `localhost`

DBName

データベース名

必須: 必須

DBPassword

データベースのパスワード。

パスワードを使用しない場合は、この行をコメントアウトしてください。

DBPort

デフォルトの Unix ドメインソケットを使用しない場合のデータベースポート。[3]

MySQL のデフォルト: 3306

PostgreSQL のデフォルト: 5432

範囲: 1024-65535

DBSchema

データベーススキーマ名。

PostgreSQL で使用されます。

DBSocket

MySQL ソケットファイルへのパス。[3]

DBUser

データベースユーザー

DBTLSConnect

このオプションを以下の値に設定すると、データベースへの TLS 接続が強制的に使用されます。
required - TLS を使用して接続します。
verify_ca - TLS を使用して接続し、証明書を検証します。
verify_full - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID がその証明書と一致することを確認します。

MySQL 5.7.11 以降および PostgreSQL では、以下の値がサポートされています。`required`、`verify_ca`、`verify_full`。
MariaDB バージョン 10.2.6 以降では、`required` および `verify_full` の値がサポートされています。
 デフォルトではどのオプションも設定されておらず、動作はデータベース構成によって異なります。

DBTLSCAFile

データベース証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

必須: いいえ（DBTLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須）

DBTLSCertFile

データベースへの認証に使用する Zabbix サーバー証明書を含むファイルのフルパス名

DBTLSKeyFile

データベースへの認証に使用する秘密鍵を含むファイルのフルパス名

DBTLSCipher

Zabbix サーバーが TLS v1.2 までの TLS プロトコルで許可する暗号化暗号スイートの一覧です。
MySQL でのみサポートされています。

DBTLSCipher13

Zabbix サーバーが TLS v1.3 プロトコルで許可する暗号スイートの一覧です。
MySQL でのみサポートされ、バージョン 8.0.16 以降で利用できます。

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）
ランタイム制御オプションも参照してください。

デフォルト: 3
 範囲: 0-5

EnableGlobalScripts

Zabbix サーバーでグローバルスクリプトを有効にします。
 注意: グローバルスクリプトの実行はデフォルトで有効です。ただし、Zabbix 7.0 以降の新規インストールでは、EnableGlobalScripts は明示的に 0（無効）に設定されます。

デフォルト: 1
 値: 0 - 無効; 1 - 有効

ExportDir

イベント、履歴、トレンドを改行区切りの JSON 形式でリアルタイムエクスポートするためのディレクトリです。
設定すると、リアルタイムエクスポートが有効になります。

ExportFileSize

エクスポートファイル 1 つあたりの最大サイズ（バイト単位）。`ExportDir` が設定されている場合、ローテーションに使用されます。

デフォルト: 1G
 範囲: 1M-1G

ExportType

リアルタイムエクスポート 用の、カンマ区切りのエンティティタイプ（events、history、trends）のリストです（デフォルトではすべてのタイプ）。ExportDir が設定されている場合にのみ有効です。
ExportType が指定されていても ExportDir が設定されていない場合、これは設定エラーとなり、サーバーは起動しません。

history と trends のエクスポートの例:

```
ExportType=history,trends
```

events のみをエクスポートする例:

```
ExportType=events
```

外部スクリプト

外部スクリプトの場所（コンパイル時のインストール変数 `datadir` に依存します）

デフォルト: /usr/local/share/zabbix/externalscripts

Fping6Location

fping6 の場所です。
fping6 バイナリの所有者が root であり、SUID フラグが設定されていることを確認してください。
使用している fping ユーティリティが IPv6 アドレスを処理できる場合は、空にしてください（”Fping6Location=”）。

デフォルト: /usr/sbin/fping6

FpingLocation

fping の場所です。
fping バイナリの所有者が root であり、SUID フラグが設定されていることを確認してください。

デフォルト: /usr/sbin/fping

FrontendAllowedIP

Web インターフェースからの接続を許可する IP アドレスまたは CIDR 範囲のカンマ区切りリストです。
このパラメータが設定されている場合、Web インターフェースからの接続はここに記載されたアドレスからのみ受け付けられます。
デフォルトでは、Web インターフェースへのリクエストに対してすべての接続が受け付けられます。
IPv6 サポートが有効な場合、’127.0.0.1’、’::127.0.0.1’、’::ffff:127.0.0.1’ は同等に扱われ、’::/0’ は任意の IPv4 または IPv6 アドレスを許可します。
’0.0.0.0/0’ は任意の IPv4 アドレスを許可するために使用できます。

例:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

高可用性クラスターノード名です。
空の場合、サーバーはスタンドアロンモードで動作し、名前が空のノードが作成されます。

HistoryCacheSize

履歴キャッシュのサイズ（バイト単位）。履歴データを保存するための共有メモリのサイズです。

デフォルト: 16M
 範囲: 128K-16G

HistoryIndexCacheSize

履歴インデックスキャッシュのサイズ（バイト単位）。履歴キャッシュに保存された履歴データのインデックス作成に使用する共有メモリのサイズです。1 つのアイテムをキャッシュするには、インデックスキャッシュサイズとしておよそ 100 バイトが必要です。

デフォルト: 4M
 範囲: 128K-16G

HistoryStorageDateIndex

履歴ストレージ内の履歴値の前処理を有効にし、日付に基づいて異なるインデックスに値を保存できるようにします。

デフォルト: 0
 値: 0 - 無効、1 - 有効

HistoryStorageURL

履歴ストレージの HTTP[S] URL です。
このパラメータは、Elasticsearch の設定に使用されます。

HistoryStorageTypes

履歴ストレージに送信する値のタイプのカンマ区切りリストです。
このパラメータは、Elasticsearch の設定に使用されます。

デフォルト: uint,dbl,str,log,text

HousekeepingFrequency

このパラメータは、Zabbix がハウスキーピング処理を何時間ごとに実行するかを決定します。ハウスキーピングとは、データベースから古い情報を削除することです。
 注: ハウスキーパーの過負荷を防ぐため（たとえば、履歴およびトレンドの保存期間が大幅に短縮された場合）、1 回のハウスキーピングサイクルで削除される古い情報は、各アイテムについて HousekeepingFrequency 時間の 4 倍までに制限されます。したがって、HousekeepingFrequency が 1 の場合、1 サイクルあたり削除される古い情報は最大 4 時間分（最も古いエントリから開始）です。
 注: サーバー起動時の負荷を下げるため、ハウスキーピングはサーバー起動後 30 分間延期されます。したがって、HousekeepingFrequency が 1 の場合、サーバー起動後の最初のハウスキーピング処理は 30 分後に実行され、その後は 1 時間ごとに繰り返されます。
HousekeepingFrequency を 0 に設定することで、自動ハウスキーピングを無効にできます。この場合、ハウスキーピング処理は *housekeeper_execute* ランタイム制御オプションによってのみ開始でき、1 回のハ

ウスキーピングサイクルで削除される古い情報の期間は、前回のハウスキーピングサイクル以降の期間の 4 倍になります。ただし、4 時間未満にはならず、4 日を超えることもありません。
 あわせてruntime control オプションも参照してください。

デフォルト: 1
 範囲: 0-24

Include

設定ファイルには、個別のファイル、またはディレクトリ内のすべてのファイルを含めることができます。
指定したディレクトリ内で関連するファイルのみを含めるには、パターンマッチング用のアスタリスクワイルドカード文字を使用できます。
制限事項については、特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。

Java poller が起動している場合にのみ必要です。

JavaGatewayPort

Zabbix Java ゲートウェイが listen するポート

デフォルト: 10052
 範囲: 1024-32767

ListenBacklog

TCP キュー内の保留中の接続の最大数
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' に自動的に切り捨てられる場合があります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

トラッパーがリッスンする IP アドレスのカンマ区切りリスト
 このパラメータが指定されていない場合、トラッパーはすべてのネットワークインターフェースをリッスンします。

デフォルト: 0.0.0.0

ListenPort

トラッパーのリッスンポート

デフォルト: 10051
 範囲: 1024-32767

LoadModule

サーバーの起動時に読み込むモジュールです。
モジュールは、サーバーの機能を拡張するために使用されます。
モジュールは、LoadModulePath で指定されたディレクトリに配置するか、モジュール名の前にパスを指定する必要があります。
前に付けるパスが絶対パス('/' で始まる)の場合､LoadModulePath は無視されます。
 形式:
LoadModule=<module.so>
LoadModule
複数の LoadModule パラメータを含めることができます。

LoadModulePath

サーバーモジュールの配置場所へのフルパスです。
デフォルト値はコンパイルオプションによって異なります。

LogFile

ログファイル名

必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションに失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024
 必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogSlowQueries

データベースクエリがログに記録されるまでの所要時間をミリ秒単位で指定
0 - スロークエリをログに記録しません。
 このオプションは、DebugLevel=3 以降で有効になります。

デフォルト: 0
 範囲: 0-3600000

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: `file`

MaxConcurrentChecksPerPoller

各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが同時に実行できる非同期チェックの最大数です。StartHTTPAgentPollers、StartAgentPollers、およびStartSNMPPollers を参照してください。

デフォルト: 1000
 範囲: 1-1000

MaxHousekeeperDelete

1 回のハウスキーピングサイクル内の 1 つのタスクで削除される行数は、'MaxHousekeeperDelete' 行（[tablename]、[field]、[value] に対応）を超えません。
0 に設定すると、制限はまったく適用されません。この場合、データベースに過負荷をかけないよう、十分に理解したうえで設定する必要があります。[2]
 このパラメータは、すでに削除されたアイテムによって残されたデータの削除にのみ適用されます。

デフォルト: 5000
 範囲: 0-1000000

NodeAddress

Web インターフェースがサーバーに接続する方法を上書きするための、オプションのポートを含む IP アドレスまたはホスト名です。
 形式: <address>[:<port>]

IP アドレスまたはホスト名が設定されていない場合は、ListenIP の値が使用されます。ListenIP が設定されていない場合は、`localhost` が使用されます。
 ポートが設定されていない場合は、ListenPort の値が使用されます。ListenPort が設定されていない場合は、10051 が使用されます。

 このオプションは、Web インターフェース設定で指定されたアドレスによって上書きできます。

 参照: HANodeName パラメータ、高可用性の有効化。

デフォルト: `localhost:10051`

PidFile

PID ファイルの名前

デフォルト: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

削除されたトリガーの問題を Zabbix が削除する頻度（秒単位）

デフォルト: 60
 範囲: 1-3600

ProxyConfigFrequency

Zabbix サーバーが設定データを Zabbix プロキシに送信する頻度を秒単位で決定します。パッシブモードのプロキシでのみ使用されます。

デフォルト: 10
 範囲: 1-604800

ProxyDataFrequency

Zabbix サーバーが Zabbix プロキシに履歴データを要求する頻度を秒単位で指定します。
パッシブモードのプロキシでのみ使用されます。

デフォルト: 1
 範囲: 1-3600

ServiceManagerSyncFrequency

Zabbix がサービスマネージャーの設定を同期する頻度（秒単位）

デフォルト: 60
 範囲: 1-3600

SMSDevices

Zabbix サーバーでの使用が許可される、カンマ区切りのモデムファイルのリストです。
 このパラメータが設定されていない場合、SMS の送信はできません。

例:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

SNMP トラップデーモンからサーバーへデータを渡すために使用される一時ファイル
zabbix_trap_receiver.pl または SNMPTT 設定ファイルと同じである必要があります。

デフォルト: `/tmp/zabbix_traps.tmp`

SocketDir

Zabbix 内部サービスで使用される IPC ソケットを保存するディレクトリ

デフォルト: `/tmp`

SourceIP

以下の送信元 IP アドレス：

- Zabbix プロキシおよび Zabbix エージェントへの送信接続
- エージェントレス接続（VMware、SSH、JMX、SNMP、Telnet、および簡易チェック）
- HTTP エージェント接続
- スクリプトアイテムの JavaScript HTTP リクエスト
- JavaScript HTTP リクエストの前処理
- 通知メールの送信（SMTP サーバーへの接続）
- Webhook 通知（JavaScript HTTP 接続）
- Vault への接続

SSHKeyLocation

SSH チェックおよびアクションで使用する公開鍵と秘密鍵の場所です。

SSLCertLocation

クライアント認証用の SSL クライアント証明書ファイルの場所
 このパラメータは Web モニタリングでのみ使用されます。

SSLKeyLocation

クライアント認証用の SSL 秘密鍵ファイルの場所
 このパラメータは Web モニタリングでのみ使用されます。

SSLCALocation

SSL サーバー証明書の検証用に、認証局（CA）ファイルの場所を上書きします。
設定されていない場合は、システム全体のディレクトリが使用されます。
 このパラメータの値は、libcurl オプション CURLOPT_CAPATH として設定されることに注意してください。
7.42.0 より前の libcurl バージョンでは、libcurl が OpenSSL を使用するようにコンパイルされている場合にのみ有効です。
詳細については、cURL web page を参照してください。
 このパラメータは、Web 監視および SMTP 認証で使用されます。

StartAgentPollers

事前に fork された Zabbix エージェントのpollerインスタンス数です。
MaxConcurrentChecksPerPollerを参照してください。

デフォルト: 1
 範囲: 0-1000

StartAlerters

alerters のプリフォークされたインスタンスの数

デフォルト: 3
 範囲: 1-100

StartBrowserPollers

ブラウザアイテムポーラー のプリフォークされたインスタンスの数

デフォルト: 1
 範囲: 0-1000

StartConnectors

コネクタワーカー の事前 fork 済みインスタンス数。コネクタワーカーが起動されると、コネクタマネージャープロセスは自動的に起動されます。

デフォルト: 0
 範囲: 0-1000

StartDBSyncers

事前に fork された履歴同期プロセスのインスタンス数です。
 注意: この値を変更する際は注意してください。増やすことで、かえって悪影響を及ぼす場合があります。
おおよそ、デフォルト値で最大 4000 NVPS まで処理できます。

デフォルト: 4
 範囲: 1-100

StartDiscoverers

ディスカバリワーカー[1] のプリフォークされたインスタンスの数

デフォルト: 5
 範囲: 0-1000

StartEscalators

エスカレーター のプリフォークされたインスタンスの数

デフォルト: 1
 範囲: 1-100

StartHistoryPollers

履歴ポーラーのプリフォークされたインスタンスの数
 計算チェックの場合のみ必須です。

デフォルト: 5
 範囲: 0-1000

StartHTTPAgentPollers

HTTP エージェントのpollerの事前 fork インスタンス数。MaxConcurrentChecksPerPollerを参照してください。

デフォルト: 1
 範囲: 0-1000

StartHTTPPollers

HTTP ポーラー[1] のプリフォークされたインスタンスの数

デフォルト: 1
 範囲: 0-1000

StartIPMIPollers

IPMI ポーラーのプリフォークされたインスタンスの数

デフォルト: 0
 範囲: 0-1000

StartJavaPollers

Java ポーラー[1] のプリフォークされたインスタンスの数

デフォルト: 0
 範囲: 0-1000

StartLLDProcessors

低レベルディスカバリ（LLD）のワーカー[1] の事前 fork インスタンス数。
LLD ワーカーが起動されると、LLD マネージャープロセスは自動的に起動されます。

デフォルト: 2
 範囲: 1-100

StartODBCPollers

ODBC ポーラー[1] のプリフォークされたインスタンスの数

デフォルト: 1
 範囲: 0-1000

StartPingers

ICMP pingers[1] のプリフォークされたインスタンスの数

デフォルト: 1
 範囲: 0-1000

StartPollersUnreachable

到達不能ホスト用ポーラー（IPMI および Java を含む）のプリフォークされたインスタンスの数[1]
 通常のポーラー、IPMI ポーラー、または Java ポーラーを起動する場合、少なくとも 1 つの到達不能ホスト用ポーラーが実行中である必要があります。

デフォルト: 1
 範囲: 0-1000

StartPollers

ポーラー[1] のプリフォークされたインスタンスの数

デフォルト: 5
 範囲: 0-1000

StartPreprocessors

前処理worker[1] スレッドの事前起動インスタンス数は、利用可能な CPU コア数以上に設定する必要があります。
前処理が CPU 負荷に制約されず、多数のネットワークリクエストがある場合は、worker をさらに多く設定する必要があります。

デフォルト: 16
 範囲: 1-1000

StartProxyPollers

パッシブプロキシ用ポーラー[1] のプリフォークされたインスタンスの数

デフォルト: 1
 範囲: 0-250

StartReportWriters

レポートライターのプリフォークされたインスタンスの数
0 に設定すると、スケジュールされたレポート生成は無効になります。
 レポートライターが起動すると、レポートマネージャプロセスが自動的に開始されます。

デフォルト: 0
 範囲: 0-100

StartSNMPPollers

SNMP ポーラー の事前 fork インスタンス数。MaxConcurrentChecksPerPoller を参照してください。

デフォルト: 1
 範囲: 0-1000

StartSNMPTrapper

1 に設定すると、SNMP トラッパー プロセスが起動します。

デフォルト: 0
 範囲: 0-1

StartTimers

タイマー のプリフォークされたインスタンスの数
 タイマープロセスのメンテナンス期間

デフォルト: 1
 範囲: 1-1000

StartTrappers

trappers[1] の事前に fork されたインスタンス数です。
 トラッパーは、Zabbix sender、アクティブエージェント、およびアクティブプロキシからの受信接続を受け付けます。

デフォルト: 5
 範囲: 0-1000

StartVMwareCollectors

プリフォークされたVMware Collector インスタンスの数

デフォルト: 0
 範囲: 0-250

StatsAllowedIP

外部 Zabbix インスタンスの IP アドレス（CIDR 表記も可）または DNS 名を、カンマ区切りで指定したリストです。統計情報のリクエストは、ここに一覧されたアドレスからのみ受け付けられます。このパラメータが設定されていない場合、統計情報のリクエストは一切受け付けられません。
IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同一のものとして扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可するために使用できます。

例:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Timeout

Zabbix プロキシ、エージェント、Web サービス、およびレガシー SNMP チェック（単一の OID 番号または文字列）との接続確立およびデータ交換を行う際の待機時間（秒）を指定します。

このパラメータは、さまざまな通信処理の継続時間を定義します。

- Zabbix エージェントでのリモートコマンド実行
- SSH/Telnet コマンド実行
- Zabbix Web サービスへのリクエスト
- メディアタイプのテストリクエストおよび `history.push` メソッドの通信タイムアウト
- IPMI インターフェースが利用不可になった場合のアイテムの再スケジュール
- 権限または暗号化の問題によりデータ交換に失敗した際に、Zabbix プロキシへ応答を送信する処理
- IPC 非同期ソケットおよびランタイム制御オプションの期限
- JMX 接続
- リモートの Zabbix プロキシまたはサーバーからの統計情報の取得
- Zabbix Web インターフェースへの応答送信
- 非同期ポーラーの DNS リクエスト
- アクティブチェックのハートビートへの応答
- アクティブエージェントからの Zabbix エージェントデータ（値）の取得
- Zabbix sender からのデータ取得
- trapper がリクエストの処理に失敗した場合の応答
- Zabbix エージェントへのアクティブチェックリストの送信

このタイムアウトは、Web インターフェースで柔軟なタイムアウト設定が構成されているチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。
たとえば、SNMP `walk[OID]` および `get[OID]` アイテムは Web インターフェースで設定されたタイムアウトを使用しますが、レガシー SNMP チェックは引き続きサーバーのタイムアウト値を使用します。

デフォルト: 3
 範囲: 1-30

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネント間の暗号化通信に使用されるサーバー証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

GnuTLS の優先順位文字列、または OpenSSL（TLS 1.2）の暗号スイート文字列です。
証明書ベースおよび PSK ベースの暗号化に対する、デフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

TLS 1.3 における OpenSSL 1.1.1 以降用の暗号スイート文字列です。
証明書ベースおよび PSK ベースの暗号化に対する、デフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS の優先順位文字列、または OpenSSL（TLS 1.2）の暗号文字列です。
証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列です。
証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先順位文字列、または OpenSSL（TLS 1.2）の暗号文字列です。
PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

OpenSSL の例:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列です。
PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。

例:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

失効した証明書を含むファイルの完全なパス名です。
このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSFrontendAccept

フロントエンドから受け入れる接続の種類

複数の値をカンマで区切って指定できます。

- unencrypted - 暗号化せずに接続を受け入れます。
- cert - TLS と証明書で保護された接続を受け入れます。

デフォルト: `unencrypted`

TLSFrontendCertIssuer

許可されたフロントエンド証明書の発行者

TLSFrontendCertSubject

許可されたフロントエンド証明書のサブジェクト

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用される、サーバー秘密鍵を含むファイルのフルパス名

TLSListen

トラッパーソケットの TLS を制御します。

サポートされる値:

- 必須 - TLS 接続のみを受け入れる

TmpDir

一時ディレクトリ

デフォルト: `/tmp`

TrapperTimeout

以下のタイムアウトを秒単位で指定します。

- Zabbix プロキシからの履歴データの取得
- Zabbix プロキシへの設定データの送信
- Zabbix サーバー上でのグローバルスクリプト実行またはリモートコマンド実行

デフォルト: 300
 範囲: 1-300

TrendCacheSize

トレンドキャッシュのサイズ（バイト単位）
 トレンドデータを保存するための共有メモリのサイズ

デフォルト: `4M`
 範囲: 128K-16G

TrendFunctionCacheSize

トレンド関数キャッシュのサイズ（バイト単位）
 計算されたトレンド関数データをキャッシュするための共有メモリのサイズ

デフォルト: `4M`
 範囲: 128K-2G

UnavailableDelay

利用不可期間中にホストの可用性をチェックする頻度 (秒単位)

デフォルト: 60
 範囲: 1-3600

UnreachableDelay

到達不能期間中にホストの可用性をチェックする頻度 (秒単位)

デフォルト: 15
 範囲: 1-3600

UnreachablePeriod

到達不能状態が何秒続いた場合にホストを利用不可と判断するかを指定します。

デフォルト: 45
 範囲: 1-3600

ユーザー

システム上の特定の既存ユーザーの権限を削除します。
'root' として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: zabbix

ValueCacheSize

履歴値キャッシュのサイズ（バイト単位）
 アイテム履歴データリクエストをキャッシュするための共有メモリのサイズ。
0 に設定すると、値キャッシュは無効になります（非推奨）
 値キャッシュの共有メモリが不足すると、5 分ごとにサーバーログに警告メッセージが書き込まれます。

デフォルト: 8M
 範囲: 0,128K-64G

Vault

Vault プロバイダーを指定します。
HashiCorp - HashiCorp KV Secrets Engine バージョン 2
CyberArk - CyberArk Central 認証情報プロバイダー
 フロントエンドで設定された Vault プロバイダーと一致する必要があります。

デフォルト: HashiCorp

VaultDBPath

データベースの認証情報をキーで取得する元となる、Vault に応じた Vault パスまたはクエリです。

HashiCorp で使用されるキーは、'password' および'username' です。

VaultPrefix=/v1/secret/data/zabbix/ を使用する場合のパスの例:

```
database
```

VaultPrefix を使用しない場合のパスの例:

```
secret/zabbix/database
```

CyberArk で使用されるキーは、'Content' および'UserName' です。

例:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

このオプションは、DBUser および DBPassword が指定されていない場合にのみ使用できます。

VaultPrefix

Vault に応じて、Vault パスまたはクエリに使用するカスタムプレフィックスです。
指定しない場合は、最適なデフォルト値が使用されます。
HashiCorp では、VaultPrefix が指定されていない場合、マウントポイントの後に data が自動的に追加されることに注意してください。

HashiCorp のプレフィックス例:

```
v1/secret/data/zabbix/
```

Cyberark のプレフィックス例:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

クライアント認証に使用する SSL 証明書ファイルの名前です。
 証明書ファイルは PEM1 形式である必要があります。
 証明書ファイルに秘密鍵も含まれている場合は、SSL キーファイルのフィールドを空のままにしてください。
 このファイルを含むディレクトリは、設定パラメータ SSLCertLocation で指定します。
 このオプションは省略できますが、CyberArkCCP vault では推奨されます。

VaultTLSKeyFile

クライアント認証に使用される SSL 秘密鍵ファイルの名前。
 秘密鍵ファイルは PEM1 形式である必要があります。
 このファイルを含むディレクトリは、設定パラメータ SSLKeyLocation で指定します。
 このオプションは省略できますが、CyberArkCCP vault では推奨されます。

VaultToken

HashiCorp Vault 認証トークン。Zabbix サーバー専用に生成され、Vault マクロで指定されたパスへの読み取り専用権限と、オプションの VaultDBPath 設定パラメータで指定されたパスへの読み取り専用権限が付与されます。
VaultToken と VAULT_TOKEN 環境変数が同時に定義されている場合はエラーになります。

必須: Vault が *HashiCorp* に設定されている場合、必須。それ以外の場合は必須ではありません。

VaultURL

vault サーバーの HTTP[S] URL です。
SSLCALocation が指定されていない場合は、システム全体の CA 証明書ディレクトリが使用されます。

デフォルト: `https://127.0.0.1:8200`

VMwareCacheSize

VMware データを保存するための共有メモリのサイズ
VMware 内部チェック zabbix[vmware,buffer,...] を使用して、VMware キャッシュの使用状況を監視できます (内部チェック を参照)
VMware Collector インスタンスが起動設定されていない場合、共有メモリは割り当てられませんのでご注意ください。

デフォルト: 8M 範囲: 256K-2G

VMwareFrequency

単一の VMware サービスからのデータ収集間隔（秒単位）
 この間隔は、VMware 監視項目の中で最も短い更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwarePerfFrequency

単一の VMware サービスからパフォーマンスカウンター統計を取得する間隔（秒）です。
この間隔は、VMware パフォーマンスカウンターを使用するすべての VMware 監視アイテムのうち、最小の更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwareTimeout

VMware コレクターが VMware サービス（vCenter または ESX ハイパーバイザー）からの応答を待機する最大秒数

デフォルト: 10
 範囲: 1-300

WebServiceURL

`http[s]://host:port/report` 形式の Zabbix Web サービスへの HTTP[S] URL です。

例:

```
WebServiceURL=http://localhost:10053/report
```

注: スキーム（`http://`）は、非 TLS（HTTP）接続の場合にのみ省略できます。TLS が設定されている場合は、`https://` を使用する必要があります。

WebDriverURL

WebDriver インターフェースの HTTP[S] URL。

例（Selenium WebDriver スタンドアロンサーバーで使用）:

```
WebDriverURL=http://localhost:4444
```

脚注

[1] 前処理マネージャーでは、IPMI マネージャー、SNMP トラッパー、前処理ワーカー、ディスカバリワーカーに加えて、あまりにも多くのデータ収集プロセス（poller、unreachable poller、ODBC poller、HTTP poller、Java poller、pinger、trapper、proxypoller）があると、プロセスごとのファイルディスクリプター上限を使い果たす可能性があることに注意してください。

> **Warning:**
> ファイルディスクリプター上限を使い果たすと、Zabbix サーバーは停止します。通常は起動直後に発生しますが、場合によってはもう少し時間がかかることもあります。このような問題を回避するには、同時チェック数とプロセス数を最適化するために、Zabbix サーバー設定ファイルを確認してください。また、必要に応じて、システムの上限を確認および調整し、ファイルディスクリプター上限が十分に高く設定されていることを確認してください。

[2] 大量のアイテムが削除されると、データベースへの負荷が増加します。これは、ハウスキーパーがそれらのアイテムに関連するすべての履歴データを削除する必要があるためです。たとえば、テンプレートから削除する必要があるのがアイテムプロトタイプ 1 つだけであっても、そのテンプレートが 50 台のホストにリンクされており、各ホストでそのプロトタイプが 100 個の実際のアイテムに展開される場合、合計 5000 個のアイテムを削除する必要があります（1*50*100）。MaxHousekeeperDelete に 500 が設定されている場合（MaxHousekeeperDelete=500）、ハウスキーパープロセスは 1 サイクルで、削除されたアイテムに対して履歴テーブルおよびトレンドテーブルから最大 2500000 個の値（5000*500）を削除する必要があります。

[3] DBSocket と DBPort はサーバー設定では相互排他的です。どちらか一方のみを指定するか、両方とも未定義のままにしてください。

2 Zabbix プロキシ

概要

このセクションでは、Zabbix プロキシ設定ファイル（zabbix_proxy.conf）でサポートされるパラメータを一覧表示しています。

パラメータは追加情報なしで一覧表示されています。
詳細を確認するには、各パラメータをクリックしてください。

Parameter	Description
AllowRoot	プロキシを'root' として実行できるようにします。
AllowUnsupportedDBVersions	サポート対象外のデータベースバージョンでプロキシを動作させることを許可します。
CacheSize	設定キャッシュのサイズ。
ConfigFrequency	このパラメータは非推奨です（代わりに ProxyConfigFrequency を使用してください）。 プロキシが Zabbix サーバーから設定データを取得する間隔（秒）。
DataSenderFrequency	プロキシは収集したデータを N 秒ごとにサーバーへ送信します。
DBHost	データベースのホスト名。
DBName	データベース名、または SQLite3 の場合はデータベースファイルへのパス。
DBPassword	データベースのパスワード。
DBPort	ローカルソケットを使用しない場合のデータベースポート。
DBSchema	データベーススキーマ名。PostgreSQL で使用されます。
DBSocket	MySQL ソケットファイルへのパス。
DBUser	データベースユーザー。
DBTLSConnect	このオプションを指定した値に設定すると、データベースへの TLS 接続の使用を強制します。
DBTLSCAFile	データベース証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。
DBTLSCertFile	データベース認証に使用する Zabbix プロキシ証明書を含むファイルのフルパス名。
DBTLSKeyFile	データベース認証に使用する秘密鍵を含むファイルのフルパス名。
DBTLSCipher	TLS v1.2 までの TLS プロトコルで Zabbix プロキシが許可する暗号スイートの一覧。MySQL でのみサポートされます。
DBTLSCipher13	TLS v1.3 プロトコルで Zabbix プロキシが許可する暗号スイートの一覧。MySQL でのみ、バージョン 8.0.16 以降でサポートされます。
DebugLevel	デバッグレベル。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可するかどうか。
ExternalScripts	外部スクリプトの場所。
Fping6Location	fping6 の場所。
FpingLocation	fping の場所。
HistoryCacheSize	履歴キャッシュのサイズ。
HistoryIndexCacheSize	履歴インデックスキャッシュのサイズ。
Hostname	一意で大文字と小文字を区別するプロキシ名。
HostnameItem	Hostname が未定義の場合に設定するために使用されるアイテム。
HousekeepingFrequency	Zabbix がハウスキーピング処理を実行する間隔（時間）。
Include	設定ファイルには、個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
JavaGateway	Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。
JavaGatewayPort	Zabbix Java ゲートウェイが待ち受けるポート。
ListenBacklog	TCP キュー内の保留中接続の最大数。
ListenIP	トラッパーが待ち受ける、カンマ区切りの IP アドレス一覧。
ListenPort	トラッパーの待ち受けポート。
LoadModule	プロキシ起動時に読み込むモジュール。
LoadModulePath	プロキシモジュールの場所へのフルパス。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogRemoteCommands	実行されたシェルコマンドの警告としてのログ記録を有効にします。
LogSlowQueries	ログに記録されるまでにデータベースクエリが要してよい時間。
LogType	ログ出力の種類。
MaxConcurrentChecksPerPoller	各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーが一度に実行できる非同期チェックの最大数。
PidFile	PID ファイル名。
ProxyBufferMode	履歴、ディスカバリ、自動登録データの保存メカニズム（disk/memory/hybrid）を指定します。
ProxyConfigFrequency	プロキシが Zabbix サーバーから設定データを取得する間隔（秒）。
ProxyLocalBuffer	データがすでにサーバーと同期されている場合でも、プロキシはデータをローカルに N 時間保持します。
ProxyMemoryBufferAge	プロキシメモリバッファ内のデータの最大保持時間（秒）。

Parameter	Description
ProxyMemoryBufferSize	収集した履歴、ディスカバリ、自動登録データ用の共有メモリキャッシュのサイズ。
ProxyMode	プロキシの動作モード（active/passive）。
ProxyOfflineBuffer	Zabbix サーバーとの接続がない場合に備えて、プロキシはデータを N 時間保持します。
Server	ProxyMode が active モードに設定されている場合: 設定データの取得元およびデータの送信先となる Zabbix サーバーの IP アドレスまたは DNS 名（address:port）、あるいはクラスター（address:port;address2:port）。 ProxyMode が passive モードに設定されている場合: Zabbix サーバーの、CIDR 表記を含めることもできるカンマ区切りの IP アドレス一覧、または DNS 名。
SNMPTrapperFile	SNMP トラップデーモンからプロキシへデータを渡すために使用される一時ファイル。
SocketDir	内部 Zabbix サービスで使用される IPC ソケットを保存するディレクトリ。
SourceIP	送信元 IP アドレス。
SSHKeyLocation	SSH チェックおよびアクション用の公開鍵と秘密鍵の場所。
SSLCertLocation	クライアント認証用 SSL クライアント証明書ファイルの場所。
SSLKeyLocation	クライアント認証用 SSL 秘密鍵ファイルの場所。
SSLCALocation	SSL サーバー証明書の検証用に、認証局（CA）ファイルの場所を上書きします。
StartAgentPollers	非同期 Zabbix エージェントポーラーの事前 fork されたインスタンス数。
StartBrowserPollers	ブラウザアイテムポーラーの事前 fork されたインスタンス数。
StartDBSyncers	履歴シンカーの事前 fork されたインスタンス数。
StartDiscoverers	ディスカバリワーカーの事前 fork されたインスタンス数。
StartHTTPAgentPollers	非同期 HTTP エージェントポーラーの事前 fork されたインスタンス数。
StartHTTPPollers	HTTP ポーラーの事前 fork されたインスタンス数。
StartIPMIPollers	IPMI ポーラーの事前 fork されたインスタンス数。
StartJavaPollers	Java ポーラーの事前 fork されたインスタンス数。
StartODBCPollers	ODBC ポーラーの事前 fork されたインスタンス数。
StartPingers	ICMP pinger の事前 fork されたインスタンス数。
StartPollersUnreachable	到達不能なホスト用ポーラー（IPMI および Java を含む）の事前 fork されたインスタンス数。
StartPollers	ポーラーの事前 fork されたインスタンス数。
StartPreprocessors	前処理ワーカーの事前起動されるインスタンス数。
StartSNMPPollers	非同期 SNMP ポーラーの事前 fork されたインスタンス数。
StartSNMPTrapper	1 に設定すると、SNMP トラッパープロセスが起動されます。
StartTrappers	トラッパーの事前 fork されたインスタンス数。
StartVMwareCollectors	VMware コレクターの事前 fork されたインスタンス数。
StatsAllowedIP	外部 Zabbix インスタンスの、CIDR 表記を含めることもできるカンマ区切りの IP アドレス一覧、または DNS 名。統計リクエストは、ここに一覧表示されたアドレスからのみ受け付けられます。
Timeout	Zabbix プロキシ、エージェント、Web サービスとの接続確立およびデータ交換、ならびに SNMP チェック（SNMP `walk[OID]` および `get[OID]` アイテムを除く）を待機する時間（秒）を指定します。
TLSAccept	Zabbix サーバーから受け入れる受信接続の種類。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	サーバー証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	GnuTLS 優先順位文字列、または OpenSSL（TLS 1.2）暗号文字列。証明書ベースおよび PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。証明書ベースおよび PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS 優先順位文字列、または OpenSSL（TLS 1.2）暗号文字列。証明書ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。証明書ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS 優先順位文字列、または OpenSSL（TLS 1.2）暗号文字列。PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSConnect	プロキシが Zabbix サーバーに接続する方法。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	プロキシ秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSListen	トラッパーソケット上の TLS を制御します。
TLSPSKFile	プロキシ事前共有鍵を含むファイルのフルパス名。Zabbix サーバーとの暗号化通信に使用されます。

Parameter	Description
TLSPSKIdentity	事前共有鍵の識別文字列。Zabbix サーバーとの暗号化通信に使用されます。
TLSServerCertIssuer	許可されるサーバー証明書の発行者。
TLSServerCertSubject	許可されるサーバー証明書のサブジェクト。
TmpDir	一時ディレクトリ。
TrapperTimeout	以下に対するタイムアウトを秒単位で指定します: - Zabbix サーバーからの設定データ取得; - グローバルスクリプト実行またはリモートコマンド実行。
UnavailableDelay	利用不可期間中にホストの可用性を確認する間隔。
UnreachableDelay	到達不能期間中にホストの可用性を確認する間隔。
UnreachablePeriod	ホストを利用不可として扱うまでの到達不能秒数。
User	システム上の特定の既存ユーザーに権限を落とします。
Vault	vault プロバイダー。
VaultDBPath	キーによってデータベース認証情報を取得する場所。
VaultPrefix	vault パスまたはクエリ用のカスタムプレフィックス。
VaultTLSCertFile	クライアント認証に使用する SSL 証明書ファイル名。
VaultTLSKeyFile	クライアント認証に使用する SSL 秘密鍵ファイル名。
VaultToken	HashiCorp vault 認証トークン。
VaultURL	vault サーバーの HTTP[S] URL。
VMwareCacheSize	VMware データ保存用の共有メモリサイズ。
VMwareFrequency	単一の VMware サービスからデータを収集する間隔（秒）。
VMwarePerfFrequency	単一の VMware サービスからパフォーマンスカウンター統計を取得する間隔（秒）。
VMwareTimeout	vmware collector が VMware サービスからの応答を待機する最大秒数。
WebDriverURL	WebDriver インターフェースの HTTP[S] URL。

すべてのパラメータは、明示的に必須と記載されていない限り任意です。

注意:

- デフォルト値はデーモンのデフォルト値を反映しており、同梱の設定ファイル内の値ではありません。
- 値ではenvironment variables をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

AllowRoot

プロキシを’root’ ユーザーとして実行できるようにします。無効にした場合、プロキシが’root’ ユーザーで起動されると、プロキシは’zabbix’ ユーザーへの切り替えを試みます。一般ユーザーで起動した場合は効果がありません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

AllowUnsupportedDBVersions

サポートされていないデータベースバージョンでプロキシが動作することを許可します。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

CacheSize

設定キャッシュのサイズ（バイト単位）。ホストとアイテムのデータを保存するための共有メモリのサイズです。

デフォルト: 32MB
 範囲: 128KB-64GB

ConfigFrequency

このパラメータは非推奨です。（代わりに ProxyConfigFrequency を使用してください）
 プロキシが Zabbix サーバーから設定データを取得する頻度（秒単位）
 アクティブプロキシパラメータ。パッシブプロキシの場合は無視されます。（ProxyMode パラメータを参照）

デフォルト: 3600
 範囲: 1-604800

DataSenderFrequency

プロキシは収集したデータを N 秒ごとにサーバーに送信します。アクティブプロキシは、リモートコマンドタスクを実行するために Zabbix サーバーを 1 秒ごとにポーリングすることに注意してください。
 アクティブプロキシパラメータ。パッシブプロキシの場合は無視されます（ProxyMode パラメータを参照）。

デフォルト: 1
 範囲: 1-3600

DBHost

データベースのホスト名
MySQL では `localhost` または空文字列を指定するとソケットが使用されます。PostgreSQL では空文字列のみを指定するとソケットが使用されます。

デフォルト: `localhost`

DBName

SQLite3 のデータベース名またはデータベースファイルへのパス（Zabbix のマルチプロセスアーキテクチャでは、インメモリデータベース は使用できません。例: `:memory:`、`file::memory:?cache=shared`、`file:memdb1?mode=memory&cache=shared`）
 警告: Zabbix サーバーが使用しているデータベースと同じデータベースを使用しないでください。

必須: はい

DBPassword

データベースのパスワード。パスワードを使用しない場合は、この行をコメントアウトしてください。SQLite の場合は無視されます。

DBPort

ローカルソケットを使用しない場合のデータベースポート。SQLite では無視されます。

範囲: 1024-65535

DBSchema

データベーススキーマ名。PostgreSQL で使用されます。

DBSocket

MySQL ソケットファイルへのパス。
 ローカルソケットを使用しない場合のデータベースポート。SQLite の場合は無視されます。

デフォルト: 3306

DBUser

データベースユーザー。SQLite では無視されます。

DBTLSConnect

このオプションを設定すると、強制的にデータベースへ TLS を使用して接続します。
required - TLS を使用して接続します。
verify_ca - TLS を使用して接続し、証明書を検証します。
verify_full - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID がその証明書と一致することを確認します。
MySQL 5.7.11 以降および PostgreSQL では、次の値がサポートされています。”required”、”verify”、”verify_full”
MariaDB バージョン 10.2.6 以降では、”required” および”verify_full” がサポートされています。
 デフォルトではどのオプションも設定されておらず、動作はデータベースの設定によって異なります。

DBTLSCAFile

データベース証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。

必須: いいえ（DBTLSConnect が *verify_ca* または *verify_full* に設定されている場合は必須）

DBTLSCertFile

データベースへの認証に使用する Zabbix プロキシ証明書を含むファイルのフルパス名。

DBTLSKeyFile

データベースへの認証に使用する秘密鍵を格納しているファイルのフルパス名。

DBTLSCipher

Zabbix プロキシが TLS v1.2 までの TLS プロトコルで許可する暗号化方式のリスト。MySQL でのみサポートされます。

DBTLSCipher13

Zabbix プロキシが TLS v1.3 プロトコルで許可する暗号化暗号スイートのリスト。MySQL のバージョン 8.0.16 以降でのみサポートされます。

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可するかどうか。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

外部スクリプト

外部スクリプトの場所（コンパイル時のインストール変数 `datadir` に依存します）

デフォルト: `/usr/local/share/zabbix/externalscripts`

Fping6Location

fping6 のパス。fping6 バイナリが root 権限を持ち、SUID フラグが設定されていることを確認してください。fping ユーティリティが IPv6 アドレスを処理できる場合は、この行を空 (”Fping6Location=”) にしてください。

デフォルト: `/usr/sbin/fping6`

FpingLocation

fping のパス。fping バイナリに root 権限があり、SUID フラグが設定されていることを確認してください。

デフォルト: `/usr/sbin/fping`

HistoryCacheSize

履歴キャッシュのサイズ（バイト単位）。履歴データを保存するための共有メモリのサイズです。

デフォルト: 16M
 範囲: 128K-16G

HistoryIndexCacheSize

履歴インデックスキャッシュのサイズ（バイト単位）。履歴キャッシュに保存された履歴データのインデックス作成に使用する共有メモリのサイズです。1 つのアイテムをキャッシュするには、インデックスキャッシュサイズとしておよそ 100 バイトが必要です。

デフォルト: 4M
 範囲: 128K-16G

ホスト名

一意のプロキシ名（大文字と小文字が区別されます）。このプロキシ名がサーバーに認識されていることを確認してください。
 使用可能な文字：英数字、’.’、’ ’、’_’、’-’ 最大文字数：128 文字

デフォルト：HostnameItem によって設定

HostnameItem

Hostname が未定義の場合に、Hostname を設定するために使用する項目です。（エージェントと同様にプロキシでも実行されます）Hostname が設定されている場合は無視されます。
UserParameters、パフォーマンスカウンタ、エイリアスはサポートしていませんが、system.run[] はサポートしています。

デフォルト: system.hostname

HousekeepingFrequency

Zabbix がハウスキーピング処理を実行する頻度（時間単位）です。
ハウスキーピングとは、データベースから古い情報を削除することです。
 注意: プロキシ起動時の負荷を下げるため、ハウスキーピングはプロキシ起動後 30 分間延期されます。
したがって、HousekeepingFrequency が 1 の場合、プロキシ起動後の最初のハウスキーピング処理は 30 分後に実行され、その後は 1 時間ごとに繰り返されます。
HousekeepingFrequency を 0 に設定することで、自動ハウスキーピングを無効にできます。
この場合、ハウスキーピング処理は *housekeeper_execute* ランタイム制御オプションによってのみ開始できます。
 あわせてランタイム制御 オプションも参照してください。

デフォルト: 1
 範囲: 0-24

Include

設定ファイルには、ディレクトリ内の個々のファイルまたはすべてのファイルをインクルードできます。
 指定したディレクトリ内の関連ファイルのみをインクルードするには、パターンマッチングでアスタリスクワイルドカード文字を使用できます。
 制限事項については、特記事項を参照してください。

例:

Include=/absolute/path/to/config/files/*.conf

JavaGateway

Zabbix Java ゲートウェイの IP アドレス（またはホスト名）。Java ポーラーが起動されている場合にのみ必要です。

JavaGatewayPort

Zabbix Java ゲートウェイが listen するポート

デフォルト: 10052
 範囲: 1024-32767

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' に自動的に切り捨てられる場合があります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

トラッパーがリッスンする IP アドレスのカンマ区切りリスト
 このパラメータが指定されていない場合、トラッパーはすべてのネットワークインターフェースをリッスンします。

デフォルト: 0.0.0.0

ListenPort

トラッパーのリッスンポート

デフォルト: 10051
 範囲: 1024-32767

LoadModule

プロキシ起動時にロードするモジュール。モジュールはプロキシの機能を拡張するために使用されます。モジュールは LoadModulePath で指定されたディレクトリに配置するか、モジュール名の前にパスを指定する必要があります。パスが絶対パス (「/」で始まる) の場合、LoadModulePath は無視されます。
 形式:
LoadModule=<module.so>
LoadModule=<path/module.so>
LoadModule=
複数の LoadModule パラメータを指定できます。

LoadModulePath

プロキシモジュールの場所へのフルパス。デフォルトはコンパイルオプションによって異なります。

LogFile

ログファイル名

必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ。(MB)
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録するかどうか。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: file

LogSlowQueries

データベースクエリがログに記録されるまでの所要時間。(ミリ秒単位)
0 - スロークエリをログに記録しません。
 このオプションは、DebugLevel=3 以降で有効になります。

デフォルト: 0
 範囲: 0-3600000

MaxConcurrentChecksPerPoller

各 HTTP エージェントポーラー、エージェントポーラー、または SNMP ポーラーで同時に実行できる非同期チェックの最大数。StartHTTPAgentPollers、StartAgentPollers、StartSNMPPollersを参照してください。

デフォルト: 1000
 範囲: 1-1000

PidFile

PID ファイルの名前

デフォルト: /tmp/zabbix_proxy.pid

ProxyBufferMode

履歴、ネットワークディスカバリ、および自動登録データの保存メカニズムを指定します。*disk* - データはデータベースに保存され、データベースからアップロードされます。*memory* - データはメモリに保存され、メモリからアップロードされます。バッファのメモリが不足すると、古いデータは破棄されます。シャットダウン時には、バッファは破棄されます。*hybrid* - プロキシバッファは通常、メモリモードと同様に動作しますが、メモリが不足するか、最も古いレコードが設定された保存期間を超えるまでその状態が続きます。その場合、バッファはデータベースにフラッシュされ、すべてのデータがアップロードされるまでディスクモードのように動作し、その後再びメモリで動作を開始します。シャットダウン時には、メモリバッファはデータベースにフラッシュされます。

参照: プロキシメモリバッファ。

デフォルト: `disk`
 値: disk; memory; hybrid

ProxyConfigFrequency

プロキシが Zabbix サーバーから設定データを取得する頻度。(秒単位）
 アクティブプロキシパラメータ。パッシブプロキシの場合は無視されます。(ProxyMode パラメータを参照）

デフォルト: 10
 範囲: 1-604800

ProxyLocalBuffer

データがサーバーと同期済みであっても、プロキシはデータをローカルに N 時間保持します。
 このパラメータは、ローカルデータがサードパーティ製アプリケーションで使用される場合に使用できます。

デフォルト: 0
 範囲: 0-720

ProxyMemoryBufferAge

プロキシメモリバッファ内のデータの最大保存期間（秒）有効（0 以外）の場合、プロキシメモリバッファ内のレコードが古い場合、すべてのレコードがサーバーにアップロードされるまで、プロキシバッファを強制的にデータベースモードに切り替えます。このパラメータは、ProxyOfflineBuffer パラメータ以下である必要があります。

デフォルト: 0
 範囲: 0;600-864000

ProxyMemoryBufferSize

収集された履歴、検出、および自動登録データを格納する共有メモリキャッシュのサイズ（バイト単位）ProxyMemoryBufferAge 有効（0 以外）の場合、キャッシュがいっぱいになるか、保存されたレコードが ProxyMemoryBufferAge で定義された値よりも古い場合を除き、プロキシは履歴検出および自動登録データをメモリに保持します。

このパラメータは ProxyLocalBuffer パラメータと同時に使用できません。

デフォルト: 0
 範囲: 0;128K-2G

ProxyMode

プロキシの動作モード。
0 - アクティブモードでのプロキシ
1 - パッシブモードでのプロキシ
 注意アクティブプロキシを使用している場合、Zabbix サーバーの trapper ポートにアクセスできる第三者が、(機密性の高い）プロキシ設定データにアクセスできるようになる可能性があります。これは、誰でもアクティブプロキシを装って設定データを要求できることによるものであり、認証は行われません。

デフォルト: 0
 範囲: 0-1

ProxyOfflineBuffer

Zabbix サーバーとの接続が切断された場合、プロキシは N 時間データを保持します。
 古いデータは失われます。

デフォルト: 1
 範囲: 1-720

サーバー

ProxyMode が アクティブモードに設定されている場合:
 設定データの取得元および送信先となる Zabbix サーバーの IP アドレスまたは DNS での名前 (アドレス: ポート)、またはクラスタ (アドレス: ポート; アドレス 2: ポート) を指定します。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
 クラスタノードはセミコロンで区切る必要があります。

ProxyMode が パッシブモードに設定されている場合:
Zabbix サーバーの IP アドレス (CIDR 表記も可能) または DNS 名をカンマ区切りで指定します。ここで指定したアドレスからの受信接続のみが許可されます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' はいずれも同等に扱われます。
'::/0' は、任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' を使用すると、任意の IPv4 アドレスを許可できます。

例:

Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

必須: yes

SNMPTrapperFile

SNMP トラップデーモンからプロキシへデータを渡すために使用される一時ファイル
zabbix_trap_receiver.pl または SNMPTT 設定ファイルと同じである必要があります。

デフォルト: `/tmp/zabbix_traps.tmp`

SocketDir

Zabbix 内部サービスで使用される IPC ソケットを格納するディレクトリ

デフォルト: `/tmp`

SourceIP

以下の送信元 IP アドレス：
- Zabbix サーバーへの送信接続
- エージェントレス接続（VMware、SSH、JMX、SNMP、Telnet、および簡易チェック）
- HTTP エージェント接続
- スクリプトアイテムの JavaScript HTTP リクエスト
- JavaScript HTTP リクエストの前処理
- Vault への接続

SSHKeyLocation

SSH チェックおよびアクションで使用する公開鍵と秘密鍵の格納場所

SSLCertLocation

クライアント認証用の SSL クライアント証明書ファイルの格納場所
 このパラメータは Web モニタリングでのみ使用されます。

SSLKeyLocation

クライアント認証用の SSL 秘密鍵ファイルの格納場所
 このパラメータは Web モニタリングでのみ使用されます。

SSLCALocation

SSL サーバ証明書の検証に使用する証明機関（CA）ファイルの格納場所
 このパラメータの値は、libcurl オプションの CURLOPT_CAPATH として設定されます。libcurl バージョン 7.42.0 より前のバージョンでは、libcurl が OpenSSL を使用するようにコンパイルされている場合にのみ有効です。詳細については、cURL ウェブページを参照してください。
 このパラメータは、Web 監視および SMTP 認証で使用されます。

StartAgentPollers

Zabbix エージェントの pre-forked インスタンスの数。（ポーラー）MaxConcurrentChecksPerPollerを参照してください。

デフォルト: 1
 範囲: 0-1000

StartBrowserPollers

ブラウザアイテムポーラー の pre-forked インスタンスの数

デフォルト: 1
 範囲: 0-1000

StartDBSyncers

history syncersの pre-forked インスタンスの数。
 注: この値を変更する際は注意してください。値を大きくすると、メリットよりもデメリットが大きくなる可能性があります。

デフォルト: 4
 範囲: 1-100

StartDiscoverers

検出ワーカーの pre-forked インスタンスの数

デフォルト: 5
 範囲: 0-1000

StartHTTPAgentPollers

HTTP エージェントポーラーの pre-forked インスタンスの数。MaxConcurrentChecksPerPollerを参照してください。

デフォルト: 1
 範囲: 0-1000

StartHTTPPollers

HTTP ポーラーの pre-forked インスタンスの数

デフォルト: 1 | 範囲: 0-1000

StartIPMIPollers

IPMI ポーラーの pre-forked インスタンスの数

デフォルト: 0
 範囲: 0-1000

StartJavaPollers

Java ポーラーの pre-forked インスタンスの数

デフォルト: 0
 範囲: 0-1000

StartODBCPollers

ODBC ポーラーの pre-forked インスタンスの数

デフォルト: 1
 範囲: 0-1000

StartPingers

ICMP pingers の pre-forked インスタンス数

デフォルト: 1
 範囲: 0-1000

StartPollersUnreachable

到達不能ホスト用ポーラー（IPMI および Java を含む）の pre-forked インスタンスの数。通常のポーラー、IPMI ポーラー、または Java ポーラーを起動する場合、少なくとも 1 つの到達不能ホスト用ポーラーが実行中である必要があります。

デフォルト: 1
 範囲: 0-1000

StartPollers

ポーラー の pre-forked インスタンスの数

デフォルト: 5
 範囲: 0-1000

StartPreprocessors

前処理ワーカー スレッドの事前起動インスタンス数は、利用可能な CPU コア数以上に設定する必要があります。前処理が CPU に依存せず、ネットワークリクエストが多数発生する場合は、ワーカー数を増やす必要があります。

デフォルト: 16
 範囲: 1-1000

StartSNMPPollers

SNMP ポーラー の pre-forked インスタンスの数。MaxConcurrentChecksPerPoller を参照してください。

デフォルト: 1
 範囲: 0-1000

StartSNMPTrapper

1 に設定すると、SNMP トラッパープロセスが起動します。

デフォルト: 0
 範囲: 0-1

StartTrappers

トラッパー の pre-forked インスタンスの数。
 トラッパーは、Zabbix sender とアクティブエージェントからの受信接続を受け入れます。

デフォルト: 5
 範囲: 0-1000

StartVMwareCollectors

VMware Collector pre-forked インスタンスの数。

デフォルト: 0
 範囲: 0-250

StatsAllowedIP

外部 Zabbix インスタンスの IP アドレス（CIDR 表記も可）、または DNS 名をカンマ区切りで指定します。統計情報リクエストは、ここにリストされたアドレスからのみ受け付けられます。このパラメータが設定されていない場合、統計情報リクエストは受け付けられません。
IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' はすべての IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' はすべての IPv4 アドレスを許可します。

例：

StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

タイムアウト

Zabbix サーバー、エージェント、Web サービス、およびレガシー SNMP チェック（単一の OID 番号または文字列）との接続確立およびデータ交換を待機する時間（秒）を指定します。

このパラメータは、さまざまな通信処理の継続時間を定義します。

- Zabbix エージェントでのリモートコマンド実行
- SSH / Telnet コマンドの実行

- IPMI インターフェースが利用不可になった場合のアイテムの再スケジュール
- 権限または暗号化の問題によりデータ交換に失敗した際に、Zabbix サーバーへ応答を送信する処理
- IPC 非同期ソケットおよびランタイム制御オプションの制限時間
- 非同期ポーラーによる DNS リクエスト
- アクティブチェックのハートビートへの応答
- アクティブエージェントからの Zabbix エージェントデータ（値）の取得
- Zabbix sender からのデータ取得
- Zabbix エージェントへのアクティブチェックリストの送信

このタイムアウトは、Web インターフェースで柔軟なタイムアウト設定が構成されているチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。
たとえば、SNMP `walk[OID]` および `get[OID]` アイテムは Web インターフェースで設定されたタイムアウトを使用します。一方、レガシー SNMP チェックは引き続きサーバーのタイムアウト値を使用します。

デフォルト: 3
 範囲: 1-30

TLSAccept

Zabbix サーバーからの受信接続の受け入れ方法。パッシブプロキシの場合は使用し、アクティブプロキシの場合は無視されます。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続を受け入れます。（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用した接続を受け入れます。
cert - TLS と証明書を使用した接続を受け入れます。

必須：パッシブプロキシの場合、TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用されるピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネント間の暗号化通信に使用されるプロキシ証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

GnuTLS の優先度文字列または OpenSSL (TLS 1.2) の暗号文字列。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSCipherAll13

TLS 1.3 における OpenSSL 1.1.1 以降の暗号文字列。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL::+SIGN-ALL:+CTYPE-X.509

OpenSSL の例:

EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

TLSCipherCert

GnuTLS の優先度文字列または OpenSSL (TLS 1.2) の暗号文字列。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

OpenSSL の例:

EECDH+aRSA+AES128:RSA+aRSA+AES128

TLSCipherCert13

TLS 1.3 における OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先度文字列または OpenSSL (TLS 1.2) の暗号文字列。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL

OpenSSL の例:

kECDHEPSK+AES128:kPSK+AES128

TLSCipherPSK13

TLS 1.3 における OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSConnect

プロキシが Zabbix サーバーに接続する方法。アクティブプロキシの場合は使用され、パッシブプロキシの場合は無視されます。指定できる値は 1 つだけです。
unencrypted - 暗号化なしで接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須：アクティブプロキシの場合、TLS 証明書または PSK パラメータが定義されている場合は「yes」（*uncrypted* 接続の場合でも）、それ以外の場合は「no」

TLSCRLFile

失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるプロキシ秘密鍵を格納するファイルのフルパス名

TLSListen

トラッパーソケットの TLS を制御します。

サポートされる値:

- 必須 - TLS 接続のみを受け入れる

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるプロキシ事前共有キーを格納するファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー証明書の発行者

TLSServerCertSubject

許可されたサーバー証明書のサブジェクト

TmpDir

テンポラリディレクトリ

デフォルト: `/tmp`

TrapperTimeout

以下のタイムアウトを秒単位で指定します。
 - Zabbix サーバーからの設定データの取得
 - グローバルスクリプトの実行またはリモートコマンドの実行

デフォルト: 300
 範囲: 1-300

UnavailableDelay

非可用性 期間中にホストへの可用性をチェックする頻度（秒単位）

デフォルト: 60
 範囲: 1-3600

UnreachableDelay

到達不能 期間中にホストへの可用性をチェックする頻度（秒単位）

デフォルト: 15
 範囲: 1-3600

UnreachablePeriod

到達不能が何秒間続いた後にホストを利用不可とみなすかの指定

デフォルト: 45
 範囲: 1-3600

ユーザー

システム上の特定の既存ユーザーの権限を削除します。
'root' として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: `zabbix`

Vault

Vault プロバイダー:
HashiCorp - HashiCorp KV Secrets Engine バージョン 2
CyberArk - CyberArk Central 認証情報プロバイダー
 フロントエンドで設定された Vault プロバイダーと一致する必要があります。

デフォルト: `HashiCorp`

VaultDBPath

データベースの認証情報がキーによって取得される、Vault パスまたはクエリ（Vault によって異なります）。

HashiCorp で使用されるキーは、'password' と 'username' です。

VaultPrefix=/v1/secret/data/zabbix/ の場合のパスの例：

`database`

VaultPrefix がない場合のパスの例：

secret/zabbix/database

CyberArk で使用されるキーは、'Content' と 'UserName' です。

例：

`AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database`

このオプションは、DBUser と DBPassword が指定されていない場合にのみ使用できます。

VaultPrefix

Vault パスまたはクエリのカスタムプレフィックス（Vault によって異なります）指定されていない場合は、最適なデフォルトが使用されます。
HashiCorp の場合、VaultPrefix が指定されていない場合は、マウントポイントの後に'data' が自動的に追加されることに注意してください。
HashiCorp の場合、VaultPrefix が指定されていない場合は、マウントポイントの後に'data' が自動的に追加されることに注意してください。

Hashicorp のプレフィックスの例：

```
v1/secret/data/zabbix/
```

Cyberark のプレフィックスの例：

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

クライアント認証に使用する SSL 証明書ファイルの名前。証明書ファイルは PEM1 形式である必要があります。
 証明書ファイルに秘密鍵も含まれている場合は、SSL 鍵ファイルのフィールドを空のままにしてください。
 このファイルを含むディレクトリは、SSLCertLocation 構成パラメータで指定します。
 このオプションは省略可能ですが、CyberArkCCP Vault の場合は推奨されます。

VaultTLSKeyFile

クライアント認証に使用する SSL 秘密鍵ファイルの名前。秘密鍵ファイルは PEM1 形式である必要があります。
 このファイルを含むディレクトリは、SSLKeyLocation 構成パラメータで指定します。
 このオプションは省略可能ですが、CyberArkCCP Vault の場合は推奨されます。

VaultToken

HashiCorp Vault 認証トークン。Zabbix プロキシ専用に生成され、オプションの VaultDBPath 設定パラメータで指定されたパスへの読み取り専用権限を持ちます。
VaultToken と VAULT_TOKEN 環境変数が同時に定義されている場合はエラーになります。

必須: Vault が *HashiCorp* に設定されている場合は必須、それ以外の場合は必須ではありません。

VaultURL

Vault サーバーの HTTP[S] URL。SSLCALocation が指定されていない場合は、システム全体の CA 証明書ディレクトリが使用されます。

デフォルト: `https://127.0.0.1:8200`

VMwareCacheSize

VMware データを保存するための共有メモリのサイズ
VMware 内部チェック zabbix[vmware,buffer,...] を使用して、VMware キャッシュの使用状況を監視できます (内部チェック を参照)
 起動するように構成された VMware Collector インスタンスがない場合、共有メモリは割り当てられません。

デフォルト: 8M
 範囲: 256K-2G

VMwareFrequency

単一の VMware サービスからのデータ収集間隔（秒単位）
 この間隔は、VMware 監視項目の中で最も短い更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwarePerfFrequency

単一の VMware サービスからパフォーマンスカウンタ統計情報を取得する間隔（秒単位）
 この間隔は、VMware パフォーマンスカウンタを使用する VMware 監視item のうち、最も短い更新間隔に設定する必要があります。

デフォルト: 60
 範囲: 10-86400

VMwareTimeout

VMware コレクターが VMware サービス（vCenter または ESX ハイパーバイザー）からの応答を待機する最大秒数

デフォルト: 10
 範囲: 1-300

WebDriverURL

WebDriver インターフェースの HTTP[S] URL

例 (Selenium WebDriver スタンドアロンサーバーで使用):

```
WebDriverURL=http://localhost:4444
```

脚注

[1] DBSocket と DBPort は、サーバー設定において相互に排他的な設定です。どちらか一方のみを指定するか、両方を未定義のままにしてください。

3 Zabbix エージェント (UNIX)

概要

このセクションでは、Zabbix エージェント設定ファイル (zabbix_agentd.conf) でサポートされているパラメータを一覧表示しています。

パラメータは追加情報なしで一覧表示されています。
各パラメータをクリックすると、詳細を確認できます。

Parameter	Description
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
AllowRoot	エージェントを'root' として実行することを許可します。
BufferSend	データをバッファ内に N 秒を超えて保持しません。
BufferSize	メモリバッファ内の値の最大数。
DebugLevel	デバッグレベル。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnableRemoteCommands	Zabbix server からのリモートコマンドを許可するかどうか。
HeartbeatFrequency	ハートビートメッセージの送信間隔（秒）。
HostInterface	ホストインターフェースを定義するオプションのパラメータ。
HostInterfaceItem	ホストインターフェースの取得に使用するアイテムを定義するオプションのパラメータ。
HostMetadata	ホストメタデータを定義するオプションのパラメータ。

Parameter	Description
HostMetadataItem	ホストメタデータの取得に使用する Zabbix エージェントアイテムを定義するオプションのパラメータ。
Hostname	ホスト名を定義するオプションのパラメータ。
HostnameItem	ホスト名の取得に使用する Zabbix エージェントアイテムを定義するオプションのパラメータ。
Include	設定ファイルには、個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenBacklog	TCP キュー内の保留中接続の最大数。
ListenIP	エージェントが待ち受ける、カンマ区切りの IP アドレス一覧。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LoadModule	エージェント起動時に読み込むモジュール。
LoadModulePath	エージェントモジュールの配置場所へのフルパス。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogRemoteCommands	実行されたシェルコマンドの警告としてのログ記録を有効にします。
LogType	ログ出力の種類。
MaxLinesPerSecond	'log' および'logrt' のアクティブチェックを処理する際に、エージェントが 1 秒あたりに Zabbix server またはプロキシへ送信する新規行の最大数。
PidFile	PID ファイル名。
RefreshActiveChecks	アクティブチェックの一覧を更新する頻度。
Server	CIDR 表記を任意で含むカンマ区切りの IP アドレス、または Zabbix server と Zabbix プロキシの DNS 名の一覧。
ServerActive	アクティブチェックの取得元となる Zabbix server/プロキシのアドレス、またはクラスタ設定。
SourceIP	送信元 IP アドレス。
StartAgents	パッシブチェックを処理する、事前 fork された zabbix_agentd インスタンス数。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。
TLSAccept	受け入れる受信接続の種類。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	GnuTLS 優先順位文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherAll13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert	GnuTLS 優先順位文字列または OpenSSL (TLS 1.2) 暗号文字列。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK	GnuTLS 優先順位文字列または OpenSSL (TLS 1.2) 暗号文字列。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	TLS 1.3 における OpenSSL 1.1.1 以降用の暗号文字列。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	エージェントが Zabbix server またはプロキシにどのように接続するか。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェント秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	エージェント事前共有鍵を含むファイルのフルパス名。Zabbix server との暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列。Zabbix server との暗号化通信に使用されます。
TLSServerCertIssuer	許可されるサーバー（プロキシ）証明書の発行者。
TLSServerCertSubject	許可されるサーバー（プロキシ）証明書のサブジェクト。
UnsafeUserParameters	ユーザー定義パラメータへの引数として、すべての文字を渡すことを許可します。
User	システム上の特定の既存ユーザーに権限を落とします。
UserParameter	監視するユーザー定義パラメータ。
UserParameterDir	UserParameter コマンドのデフォルト検索パス。

明示的に必須と記載されていない限り、すべてのパラメータは必須ではありません。

注意:

- デフォルト値は、配布される設定ファイル内の値ではなく、デーモンのデフォルト値を反映しています。
- 値ではenvironment variables をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

エイリアス

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、より短くてシンプルなものに置き換えることができます。
 Alias パラメーターは複数指定できます。同じ *Alias* キーを持つ複数のパラメーターは許可されません。
 異なる *Alias* キーが同じアイテムキーを参照する場合があります。
 エイリアスは *HostMetadataItem* では使用できますが、*HostnameItem* パラメーターでは使用できません。

例 1: ユーザー’zabbix’ の ID を取得します。

Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,”^zabbix:.:([0-9]+)”„„\1]

これで、**zabbix.userid** という短縮キーを使用してデータを取得できるようになりました。

例 2: デフォルトおよびカスタムパラメーターを使用した CPU 使用率の取得

エイリアス =cpu.util:system.cpu.util エイリアス =cpu.util[*]:system.cpu.util[*]

これにより、**cpu.util** キーを使用してデフォルトのパラメータで CPU 使用率を取得できるだけでなく、**cpu.util[all, idle, avg15]** を使用して CPU 使用率に関する特定のデータを取得できます。

例 3: 同じ検出項目を処理する複数の低レベル検出 ルールを実行する。

エイリアス =vfs.fs.discovery[*]:vfs.fs.discovery

これで、**vfs.fs.discovery** を使用して、ルールごとに異なるパラメータを持つ複数の検出ルールを設定できるようになりました。(例: **vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など)

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

AllowRoot

エージェントを’root’ で実行できるようにします。無効にした場合、エージェントが’root’ で起動されると、代わりに’zabbix’ ユーザーへの切り替えを試みます。一般ユーザーで起動した場合は効果がありません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

BufferSend

N 秒を超えてバッファ内にデータを保持しません。

デフォルト: 5
 範囲: 1-3600

バッファサイズ

メモリバッファ内の値の最大数。バッファがいっぱいになった場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。

デフォルト: 100
 範囲: 2-65535

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
AllowKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可するかどうか。このパラメータは非推奨です。代わりに AllowKey=system.run[*] または DenyKey=system.run[*] を使用してください。
 これは、値に応じて AllowKey/DenyKey パラメータの内部エイリアスです。
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

デフォルト: 0
 値: 0 - 許可しない 1 - 許可する

HeartbeatFrequency

ハートビートメッセージの送信頻度（秒単位）アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージは無効です。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストインターフェースを定義するオプションのパラメーターです。ホストインターフェースは、ホストの自動登録プロセスで使用されます。定義されていない場合は、HostInterfaceItem から値が取得されます。
 値が 255 文字の制限を超える場合、エージェントはエラーを発行し、起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストインターフェースの取得に使用する項目を定義するオプションパラメータです。
 ホストインターフェースは、ホストの自動登録プロセスで使用されます。
 自動登録リクエスト中に、指定された項目によって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。
system.run[]項目は、AllowKey/DenyKey の値に関係なくサポートされます。
 このオプションは、HostInterface が定義されていない場合にのみ使用されます。

HostMetadata

ホストメタデータを定義するオプションパラメータです。ホストメタデータは、ホスト自動登録プロセス（アクティブエージェント）でのみ使用されます。定義されていない場合は、HostMetadataItem から値が取得されます。
 指定された値が 2034 バイトの制限を超える場合、または UTF-8 以外の文字列の場合、エージェントはエラーを発行し、起動しません。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータの取得に使用する Zabbix エージェントアイテムを定義するオプションパラメータです。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。ユーザーパラメータとエイリアスがサポートされています。system.run[]アイテムは、AllowKey/DenyKey の値に関係なくサポートされます。
HostMetadataItem の値は、自動登録の試行ごとに取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。
 自動登録リクエスト中、指定されたアイテムによって返される値が UTF-8 コードポイントの制限である 65535 を超える場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列である必要があります。そうでない場合は無視されます。

Hostname

カンマ区切りの、大文字と小文字を区別する一意のホスト名のリストです。アクティブチェックに必須で、サーバーに設定されているホスト名と一致する必要があります。値が未定義の場合は、HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、-'。最大文字数: ホスト名ごとに 128 文字、行全体で 2048 文字。

デフォルト: HostnameItem によって設定

HostnameItem

ホスト名の取得に使用する Zabbix エージェントアイテムを定義する追加パラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータやエイリアスはサポートされていませんが、system.run[]アイテムは、AllowKey/DenyKey の値に関わらずサポートされます。

デフォルト: `system.hostname`

Include

設定ファイルには、ディレクトリ内の個々のファイルまたはすべてのファイルをインクルードできます。指定したディレクトリ内の関連ファイルのみをインクルードするには、パターンマッチングでアスタリスクワイルドカード文字を使用できます。
 制限事項については、特記事項を参照してください。

例:

Include=/absolute/path/to/config/files/*.conf

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' で自動的に切り捨てられる場合があります。

デフォルト: `SOMAXCONN`
 範囲: 0 - INT_MAX

ListenIP

エージェントが待受する IP アドレスのカンマ区切りリスト

デフォルト: 0.0.0.0

ListenPort

エージェントは、サーバーからの接続をこのポートで待受します。

デフォルト: 10050
 範囲: 1024-32767

LoadModule

エージェント起動時にロードするモジュール。モジュールはエージェントの機能を拡張するために使用されます。モジュールは LoadModulePath で指定されたディレクトリに配置するか、モジュール名の前にパスを指定する必要があります。パスが絶対パス ('/' で始まる)の場合、LoadModulePath は無視されます。
 形式:
LoadModule=<module.so>
LoadModule=<path/module.so>
LoadModu
複数の LoadModule パラメータを指定できます。

LoadModulePath

エージェントモジュールの場所へのフルパス。デフォルトはコンパイルオプションによって異なります。

LogFile

ログファイル名

必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。コマンドはリモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: `file`

MaxLinesPerSecond

'log' および'logrt' アクティブチェックの処理時に、エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新規行の最大数。ここで指定した値は、'log' または'logrt' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新規行を処理します。

デフォルト: 20
 範囲: 1-1000

PidFile

PID ファイルの名前

デフォルト: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

アクティブチェックのリストを更新する頻度（秒単位）アクティブチェックのリストの更新に失敗した場合、次の更新は 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーおよび Zabbix プロキシの IP アドレス（CIDR 表記も可）または DNS 名をカンマ区切りで指定します。受信接続は、ここに指定されたホストからのみ受け付けられます。IPv6 サポートが有効になっている場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 アドレスまたは IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可します。"IPv4 互換 IPv6 アドレス"（プレフィックス 0000::/96）はサポートされていますが、RFC4291 では非推奨となっています。スペースの使用は可能です。

例:

Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

必須: StartAgents が明示的に 0 に設定されていない場合は必須

ServerActive

アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定です。
サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、必要に応じてコロンで区切ったポート番号です。
 クラスタ設定は、セミコロンで区切られた 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。
複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマで区切って指定できます。
プロキシグループを使用する場合を除き、各 Zabbix サーバー/クラスタごとに複数の Zabbix プロキシを指定してはいけません。
Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定してはいけません。
 複数の独立した Zabbix サーバーを並行して使用するために、カンマ区切りで複数のアドレスを指定できます。
スペースを含めることができます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでそのホストのポートを指定する場合、IPv6 アドレスは角括弧で囲む必要があります。
ポートを指定しない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用する高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下の送信元 IP アドレス：
- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 特定のアイテム（web.page.get、net.tcp.port など）の実行中に接続を確立する場合

StartAgents

パッシブチェックを処理する、Zabbix_agentd のプリフォークインスタンスの数。0 に設定するとパッシブチェックは無効化され、エージェントはどの TCP ポートも待受しません。

デフォルト: 10
 範囲: 0-100

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。

このパラメータは、以下を含むさまざまな通信処理の継続時間を定義します。

- Zabbix サーバーからの応答待機。
- アクティブチェック におけるアイテム設定要求およびアイテムデータを含む、Zabbix サーバーへのリクエスト送信。
- logfile を介したログデータの取得。
- ハートビートメッセージの送信。
- `vfs.*` チェックの最大継続時間。
- Zabbix エージェントモジュールによる使用。
- バージョン 7.0 より前のサーバーまたはプロキシがタイムアウトなしでチェックを送信する場合のフォールバックとしての使用。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる接続の種類。パッシブチェックに使用します。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続を受け入れます（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用した接続を受け入れます
cert - TLS と証明書を使用した接続を受け入れます

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

GnuTLS の優先度文字列または OpenSSL (TLS 1.2) の暗号文字列。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSCipherAll13

TLS 1.3 における OpenSSL 1.1.1 以降の暗号文字列。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL::+SIGN-ALL:+CTYPE-X.509

OpenSSL の例:

EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

TLSCipherCert

GnuTLS の優先度文字列または OpenSSL (TLS 1.2) の暗号文字列。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL:+CTYPE-X.509

OpenSSL の例:

EECDH+aRSA+AES128:RSA+aRSA+AES128

TLSCipherCert13

TLS 1.3 における OpenSSL 1.1.1 以降の暗号文字列。証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

TLSCipherPSK

GnuTLS の優先度文字列または OpenSSL (TLS 1.2) の暗号文字列。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

GnuTLS の例:

NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIGN-ALL

OpenSSL の例:

kECDHEPSK+AES128:kPSK+AES128

TLSCipherPSK13

TLS 1.3 における OpenSSL 1.1.1 以降の暗号文字列。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSConnect

エージェントが Zabbix サーバーまたはプロキシに接続する方法。アクティブチェックに使用されます。指定できる値は 1 つだけです。
unencrypted - 暗号化なしで接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCRLFile

失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータの引数として、すべての文字を渡すことを許可します。以下の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

ユーザー

システム上の特定の既存ユーザーの権限を削除します。
'root' として実行され、AllowRoot が無効になっている場合にのみ有効です。

デフォルト: zabbix

UserParameter

監視するユーザー定義パラメータ。複数のユーザー定義パラメータを指定できます。
 形式: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返さないでください。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスで指定できます。

例:

UserParameter=system.test,who|wc -l UserParameter=check_cpu,./custom_script.sh

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。このパスを指定すると、エージェントはコマンド実行前にここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドにはフルパスではなく相対パスの ./ プレフィックスを付けることができます。
 指定できるエントリは 1 つだけです。

例:

UserParameterDir=/opt/myscripts

参照

1. バージョン 2.0.0 以降のアクティブチェックとパッシブチェックの Zabbix エージェント構成の違い

4 Zabbix エージェント 2 (UNIX)

概要

Zabbix agent 2 は新世代の Zabbix エージェントであり、Zabbix エージェントの代わりに使用できます。

このセクションでは、Zabbix agent 2 の設定ファイル (zabbix_agent2.conf) でサポートされるパラメータを一覧表示します。

パラメータは追加情報なしで一覧表示されています。詳細については、パラメータをクリックしてください。

Parameter	Description
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。

Parameter	Description
BufferSend	データをバッファ内に N 秒を超えて保持しません。
BufferSize	メモリバッファ内の値の最大数。
ControlSocket	'-R' オプションで実行時コマンドを送信するために使用される制御ソケット。
DebugLevel	デバッグレベル。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnablePersistentBuffer	アクティブアイテムに対するローカル永続ストレージの使用を有効にします。
ForceActiveChecksOnStart	再起動後、最初に受信した設定に対して直ちにアクティブチェックを実行します。
HeartbeatFrequency	ハートビートメッセージの送信間隔（秒）。
HostInterface	ホストインターフェースを定義するオプションのパラメータ。
HostInterfaceItem	ホストインターフェースの取得に使用するアイテムを定義するオプションのパラメータ。
HostMetadata	ホストメタデータを定義するオプションのパラメータ。
HostMetadataItem	ホストメタデータの取得に使用する Zabbix エージェントアイテムを定義するオプションのパラメータ。
Hostname	ホスト名を定義するオプションのパラメータ。
HostnameItem	ホスト名の取得に使用する Zabbix エージェントアイテムを定義するオプションのパラメータ。
Include	設定ファイルには、個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenIP	エージェントが待ち受ける、カンマ区切りの IP アドレスのリスト。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力の種類。
PersistentBufferFile	Zabbix agent 2 が SQLite データベースを保持するファイル。
PersistentBufferPeriod	サーバーまたはプロキシへの接続がない場合にデータを保存する期間。
PidFile	PID ファイル名。
Plugins.<PluginName>.System.Capacity	同時に実行できるプラグインごとのチェック数の上限。
Plugins.Log.MaxLinesPerSecond	'log' および 'logrt' のアクティブチェックを処理する際に、エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行の最大数。
Plugins.SystemRun.LogRemoteCommands	実行されたシェルコマンドの警告としてのログ記録を有効にします。
PluginSocket	ロード可能プラグインとの通信用 UNIX ソケットへのパス。
PluginTimeout	ロード可能プラグインとの接続のタイムアウト（秒）。
RefreshActiveChecks	アクティブチェックの一覧を更新する頻度。
Server	CIDR 表記を任意で含む、カンマ区切りの IP アドレス、または Zabbix サーバーおよび Zabbix プロキシの DNS 名のリスト。
ServerActive	アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定。
SourceIP	送信元 IP アドレス。
StatusPort	設定した場合、エージェントはこのポートで HTTP ステータス要求 (http://localhost:<port>/status) を待ち受けます。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。
TLSAccept	受け入れる受信接続の種類。
TLSCAFile	ピア証明書の検証用トップレベル CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll	OpenSSL の暗号文字列 (TLS 1.2)。証明書ベースおよび PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherAll13	OpenSSL 1.1.1 以降向けの OpenSSL 暗号文字列 (TLS 1.3)。証明書ベースおよび PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert	OpenSSL の暗号文字列 (TLS 1.2)。証明書ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherCert13	OpenSSL 1.1.1 以降向けの OpenSSL 暗号文字列 (TLS 1.3)。証明書ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK	OpenSSL の暗号文字列 (TLS 1.2)。PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSCipherPSK13	OpenSSL 1.1.1 以降向けの OpenSSL 暗号文字列 (TLS 1.3)。PSK ベース暗号化のデフォルト暗号スイート選択基準を上書きします。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシに接続する方法。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェント秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。

Parameter	Description
TLSPSKFile	エージェント事前共有鍵を含むファイルのフルパス名。Zabbix サーバーとの暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列。Zabbix サーバーとの暗号化通信に使用されます。
TLSServerCertIssuer	許可されるサーバー（プロキシ）証明書の発行者。
TLSServerCertSubject	許可されるサーバー（プロキシ）証明書のサブジェクト。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡すことを許可します。
UserParameter	監視するユーザー定義パラメータ。
UserParameterDir	UserParameter コマンドのデフォルト検索パス。

すべてのパラメータは、明示的に必須と記載されていない限り、必須ではありません。

注意:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値ではenvironment variables をサポートしています。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

エイリアス

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、より短くてシンプルなものに置き換えることができます。
 Alias パラメーターは複数指定できます。同じ *Alias* キーを持つ複数のパラメーターは許可されません。
 異なる *Alias* キーが同じアイテムキーを参照する場合があります。
 エイリアスは *HostMetadataItem* では使用できますが、*HostnameItem* パラメーターでは使用できません。

例 1: ユーザー'zabbix' の ID を取得します。

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

これで、**zabbix.userid** という短縮キーを使用してデータを取得できるようになりました。

例 2: デフォルトおよびカスタムパラメーターを使用した CPU 使用率の取得

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

これにより、**cpu.util** キーを使用してデフォルトのパラメータで CPU 使用率を取得できるだけでなく、**cpu.util[all, idle, avg15]** を使用して CPU 使用率に関する特定のデータを取得できます。

例 3: 同じ検出項目を処理する複数の低レベル検出 ルールを実行する。

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

これで、**vfs.fs.discovery** を使用して、ルールごとに異なるパラメータを持つ複数の検出ルールを設定できるようになりました。(例: **vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など)

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

BufferSend

バッファから Zabbix サーバーへ値を送信する間隔（秒単位）です。バッファがいっぱいの場合は、データがより早く送信されます。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファ内の値の最大数。バッファがいっぱいになった場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。このパラメータは、永続バッファが無効になっている場合（*EnablePersistentBuffer=0*）のみ使用してください。

デフォルト: 1000
 範囲: 2-65535

ControlSocket

コントロールソケット、'-R' オプションを指定したランタイムコマンドの送信に使用します。

デフォルト: `/tmp/agent.sock`

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
AllowKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

EnablePersistentBuffer

アクティブアイテム用のローカル永続ストレージの使用を有効にします。永続ストレージが無効になっている場合は、メモリバッファが使用されます。

デフォルト: 0
 値: 0 - 無効、1 - 有効

ForceActiveChecksOnStart

再起動後、最初に受信した構成に対してアクティブチェックを直ちに実行します。プラグインごとの設定パラメータとしても利用可能です。例: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

デフォルト: 0
 値: 0 - 無効、1 - 有効

HeartbeatFrequency

ハートビートメッセージの送信頻度（秒単位）。アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージは無効です。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストインターフェースを定義する追加パラメーターです。ホストインターフェースは、ホストの自動登録プロセスで使用されます。定義されていない場合は、HostInterfaceItem から値が取得されます。
 値が 255 文字の制限を超える場合、エージェントはエラーを通知して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストインターフェースの取得に使用するアイテムを定義する追加パラメータです。
 ホストインターフェースは、ホストの自動登録プロセスで使用されます。このオプションは、HostInterface が定義されていない場合にのみ使用されます。
system.run[]アイテムは、AllowKey/DenyKey の値に関係なくサポートされます。
 自動ディスカバリリクエスト中に、指定された項目によって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。

HostMetadata

ホストメタデータを定義する追加パラメータです。ホストメタデータは、ホスト自動登録プロセスでのみ使用されます。定義されていない場合は、HostMetadataItem から値が取得されます。
 指定された値が 2034 バイトの制限を超える場合、または UTF-8 以外の文字列の場合、エージェントはエラーを通知して起動しません。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータの取得に使用する項目を定義する追加パラメータです。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。ユーザーパラメータとエイリアスがサポートされています。system.run[] 項目は、AllowKey/DenyKey の値に関係なくサポートされます。
HostMetadataItem の値は、自動登録の試行ごとに取得され、ホストの自動登録プロセスでのみ使用されます。
 自動登録リクエスト中、指定されたアイテムによって返される値が UTF-8 コードポイントの上限である 65535 を超える場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列である必要があります。そうでない場合は無視されます。

ホスト名

カンマ区切りの、大文字と小文字を区別する一意のホスト名のリストです。アクティブチェックに必須で、サーバーに設定されているホスト名と一致する必要があります。値が未定義の場合は、HostnameItem から取得されます。
 使用可能な文字: 英数字、’.’、”、’_’、’-’。最大文字数: ホスト名ごとに 128 文字、行全体で 2048 文字。

デフォルト: HostnameItem によって設定

HostnameItem

ホスト名の取得に使用する項目を定義する追加パラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータやエイリアスはサポートされていませんが、system.run[] アイテムは、AllowKey/DenyKey の値に関わらずサポートされます。

デフォルト: `system.hostname`

Include

設定ファイルには、個々のファイルまたはディレクトリ内のすべてのファイルを include できます。Zabbix はインストール時に、コンパイル時に変更されない限り、/usr/local/etc に include ディレクトリを作成します。パスは、*zabbix_agent2.conf* ファイルの場所からの相対パスで指定できます。
 指定されたディレクトリ内の関連ファイルのみを include するには、パターンマッチングでアスタリスクワイルドカード文字を使用できます。
 制限事項については特記事項を参照してください。

例:

Include=/absolute/path/to/config/files/*.conf

ListenIP

エージェントが Listen する IP アドレスをカンマ区切りで指定します。最初の IP アドレスは、Zabbix サーバーに接続している場合、アクティブなチェックリストを取得するために Zabbix サーバーに送信されます。

デフォルト: `0.0.0.0`

ListenPort

エージェントは、サーバーからの接続をこのポートで待ち受けします。

デフォルト: 10050
 範囲: 1024-32767

LogFile

ログファイル名

デフォルト: `/tmp/zabbix_agent2.log`
 必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: `file`

PersistentBufferFile

Zabbix エージェント 2 が SQLite データベースを保存するファイル。完全なファイル名を指定する必要があります。このパラメータは、永続バッファが有効になっている場合（*EnablePersistentBuffer=1*）のみ使用されます。

PersistentBufferPeriod

サーバーまたはプロキシへの接続がない場合にデータを保存する期間です。古いデータは失われます。ログデータは保持されます。このパラメータは、永続バッファが有効になっている場合（*EnablePersistentBuffer=1*）のみ使用されます。

デフォルト: 1h
 範囲: 1m～365d

PidFile

PID ファイルの名前

デフォルト: `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

`<PluginName>` プラグインごとに同時に実行できるチェックの上限。

デフォルト: 1000 範囲: 1-1000

Plugins.Log.MaxLinesPerSecond

'log' および'logrt' アクティブチェックの処理時に、エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新規行の最大数。ここで指定した値は、'log' および'logrt' アイテムキーで指定された'maxlines' パラメータによって上書きされます。

注: Zabbix は、ログアイテム内で必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新規行を処理します。

デフォルト: 20
 範囲: 1-1000

Plugins.SystemRun.LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。コマンドはリモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

PluginSocket

ロード可能なプラグイン通信用の UNIX ソケットへのパス

デフォルト: `/tmp/agent.plugin.sock`

PluginTimeout

読み込み可能なプラグインとの接続のタイムアウト（秒単位）

デフォルト: `Timeout`
 範囲: 1-30

RefreshActiveChecks

アクティブチェックのリストを更新する頻度（秒単位）。アクティブチェックのリストの更新に失敗した場合、次の更新は 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーまたは Zabbix プロキシの IP アドレス（CIDR 表記も可）または DNS 名をカンマ区切りで指定します。ここで指定したホストからの接続のみが受信されます。IPv6 サポートが有効な場合、'127.0.0.1'、'::127.0.0.1'、'::ffff:127.0.0.1' は同等に扱われ、'::/0' は任意の IPv4 または IPv6 アドレスを許可します。'0.0.0.0/0' は任意の IPv4 アドレスを許可します。スペースも使用できます。このパラメータを指定しない場合、パッシブチェックは無効になり、エージェントはどの TCP ポートも待受しません。

例：

Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

ServerActive

アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定を指定します。
サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、必要に応じてコロンで区切ったポート番号です。
 クラスタ設定は、1 つ以上のサーバーまたはプロキシグループメンバーのアドレスをセミコロンで区切って指定します。
複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマで区切って指定できます。
プロキシグループを使用しない限り、各 Zabbix サーバー/クラスタごとに複数の Zabbix プロキシを指定してはいけません。
Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定してはいけません。
 複数の独立した Zabbix サーバーを並行して使用するために、カンマ区切りで複数のアドレスを指定できます。
スペースを含めることができます。
 ポートを指定しない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでそのホストのポートを指定する場合、IPv6 アドレスは角括弧で囲む必要があります。
ポートを指定しない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータを指定しない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用した高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下の送信元 IP アドレス：
- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 特定のアイテム（web.page.get、net.tcp.port など）の実行中に接続を確立する場合

StatusPort

設定されている場合、エージェントはこのポートで HTTP ステータスリクエスト（http://localhost:<port>/status）を待受します。

範囲: 1024-32767

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。

このパラメータは、以下を含むさまざまな通信処理の継続時間を定義します。

- Zabbix サーバーからの応答待機時間。
- アクティブチェック におけるアイテム設定要求およびアイテムデータを含む、Zabbix サーバーへのリクエスト送信。
- logfile を介したログデータの取得。
- ハートビートメッセージの送信。
- `vfs.*` チェックの最大継続時間。
- バージョン 7.0 より前のサーバーまたはプロキシがタイムアウトなしでチェックを送信する場合のフォールバックとして機能すること。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる受信接続。パッシブチェックに使用します。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続を受け入れます（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用した接続を受け入れます
cert - TLS と証明書を使用した接続を受け入れます

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll

OpenSSL 暗号文字列 (TLS 1.2) 証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128

TLSCipherAll13

OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列 (TLS 1.3) 証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSCipherCert

OpenSSL 暗号文字列 (TLS 1.2) 証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

EECDH+aRSA+AES128:RSA+aRSA+AES128

このパラメータは `TLSAccept=cert,psk` と同時に使用できないことに注意してください。証明書接続 (`TLSConnect=cert`) の場合は、代わりに TLSCipherAll を使用してください。

TLSCipherCert13

OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列 (TLS 1.3) 証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

このパラメータは TLSAccept=cert,psk と同時に使用できないことに注意してください。証明書接続 (TLSConnect=cert) の場合は、代わりに TLSCipherAll13 を使用してください。

TLSCipherPSK

OpenSSL 暗号文字列 (TLS 1.2)PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

kECDHEPSK+AES128:kPSK+AES128

TLSCipherPSK13

OpenSSL 1.1.1 以降用の OpenSSL 暗号文字列 (TLS 1.3)PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSConnect

エージェントが Zabbix サーバーまたはプロキシに接続する方法。アクティブチェックに使用されます。指定できる値は 1 つだけです。
unencrypted - 暗号化なしで接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCRLFile

失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータの引数として、すべての文字を渡すことを許可します。以下の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータ。複数のユーザー定義パラメータを指定できます。
 形式: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返さないでください。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスで指定できます。

例:

UserParameter=system.test,who|wc -l UserParameter=check_cpu,./custom_script.sh

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。このパスを指定すると、エージェントはコマンド実行前にここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドにはフルパスではなく相対パスの ./ プレフィックスを付けることができます。
 指定できるエントリは 1 つだけです。

例:

UserParameterDir=/opt/myscripts

5 Zabbix エージェント (Windows)

概要

このセクションでは、Windows 用 Zabbix エージェント設定ファイル（zabbix_agentd.conf）でサポートされるパラメータを一覧表示しています。

パラメータは追加情報なしで一覧表示されています。
各パラメータをクリックすると、詳細を確認できます。

Parameter	Description
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
BufferSend	データをバッファ内に N 秒を超えて保持しません。
BufferSize	メモリバッファ内の値の最大数。
DebugLevel	デバッグレベル。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnableRemoteCommands	Zabbix サーバーからのリモートコマンドを許可するかどうか。
HeartbeatFrequency	ハートビートメッセージの送信間隔（秒）。
HostInterface	ホストインターフェースを定義する任意のパラメータ。
HostInterfaceItem	ホストインターフェースの取得に使用するアイテムを定義する任意のパラメータ。
HostMetadata	ホストメタデータを定義する任意のパラメータ。
HostMetadataItem	ホストメタデータの取得に使用する Zabbix エージェントのアイテムを定義する任意のパラメータ。
Hostname	ホスト名を定義する任意のパラメータ。
HostnameItem	ホスト名の取得に使用する Zabbix エージェントのアイテムを定義する任意のパラメータ。
Include	設定ファイルには、個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenBacklog	TCP キュー内の保留中接続の最大数。
ListenIP	エージェントが待ち受ける、カンマ区切りの IP アドレスのリスト。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogRemoteCommands	実行されたシェルコマンドの警告としてのログ記録を有効にします。
LogType	ログ出力の種類。
MaxLinesPerSecond	'log' および'logrt' のアクティブチェックを処理する際に、エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新しい行の最大数。
PerfCounter	指定した期間 <period>（秒）におけるシステムパフォーマンスカウンタ <perf_counter_path> の平均値となる新しいパラメータ <parameter_name> を定義します。
PerfCounterEn	指定した期間 <period>（秒）におけるシステムパフォーマンスカウンタ <perf_counter_path> の平均値となる新しいパラメータ <parameter_name> を定義します。PerfCounter と比較して、perfcounter パスは英語である必要があります。
RefreshActiveChecks	アクティブチェックのリストを更新する頻度。
Server	CIDR 表記を任意で使用できる、カンマ区切りの IP アドレス、または Zabbix サーバーおよび Zabbix プロキシの DNS 名のリスト。
ServerActive	アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定。
SourceIP	送信元 IP アドレス。
StartAgents	パッシブチェックを処理する、事前に fork された zabbix_agentd のインスタンス数。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。
TLSAccept	受け入れる受信接続の種類。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信で使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信で使用されます。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシにどのように接続するか。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信で使用されます。
TLSKeyFile	エージェント秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信で使用されます。
TLSPSKFile	エージェントの事前共有鍵を含むファイルのフルパス名。Zabbix サーバーとの暗号化通信で使用されます。
TLSPSKIdentity	Zabbix サーバーとの暗号化通信で使用される事前共有鍵の識別文字列。
TLSServerCertIssuer	許可されるサーバー（プロキシ）証明書の発行者。

Parameter	Description
TLSServerCertSubject	許可されるサーバー（プロキシ）証明書のサブジェクト。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡すことを許可します。
UserParameter	監視するユーザー定義パラメータ。
UserParameterDir	UserParameter コマンドのデフォルト検索パス。

すべてのパラメータは、明示的に必須と記載されていない限り、必須ではありません。

注意:

- デフォルト値は、配布される設定ファイル内の値ではなく、デーモンのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

エイリアス

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、より短くてシンプルなものに置き換えることができます。
 Alias パラメータは複数指定できます。同じ *Alias* キーを持つ複数のパラメータは許可されません。
 異なる *Alias* キーが同じアイテムキーを参照することは可能です。
 エイリアスは *HostMetadataItem* では使用できますが、*HostnameItem* や *PerfCounter* パラメータでは使用できません。

例 1: サーバーからページングファイルの使用率をパーセンテージで取得する。

Alias=pg_usage:perf_counter[\Paging File(_Total)% Usage]

これで、短縮キー **pg_usage** を使用してデータを取得できるようになりました。

例 2: デフォルトおよびカスタムパラメータを使用した CPU 負荷の取得

エイリアス =cpu.load:system.cpu.load エイリアス =cpu.load[*]:system.cpu.load[*]

これにより、**cpu.load** キーを使用してデフォルトのパラメータで CPU 負荷を取得し、**cpu.load[percpu,avg15]** を使用して CPU 負荷に関する特定のデータを取得できます。

例 3: 同じ検出項目を処理する複数の低レベル検出 ルールを実行する。

エイリアス =vfs.fs.discovery[*]:vfs.fs.discovery

これで、**vfs.fs.discovery** を使用して、ルールごとに異なるパラメータを持つ複数の検出ルールを設定できるようになりました。(例: **vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など)

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

BufferSend

N 秒を超えてバッファ内にデータを保持しません。

デフォルト: 5
 範囲: 1-3600

バッファサイズ

メモリバッファ内の値の最大数。バッファがいっぱいになった場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。

デフォルト: 100
 範囲: 2-65535

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
AllowKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

EnableRemoteCommands

Zabbix サーバーからのリモートコマンドを許可するかどうか。このパラメータは非推奨です。代わりに AllowKey=system.run[*] または DenyKey=system.run[*] を使用してください。
 これは、値に応じて AllowKey/DenyKey パラメータの内部エイリアスです。
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

HeartbeatFrequency

ハートビートメッセージの送信頻度（秒単位）アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージは無効です。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストインターフェースを定義するオプションのパラメーターです。ホストインターフェースは、ホストの自動登録プロセスで使用されます。定義されていない場合は、HostInterfaceItem から値が取得されます。
 値が 255 文字の制限を超える場合、エージェントはエラーを通知し、起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストインターフェースの取得に使用する項目を定義するオプションパラメータです。
 ホストインターフェースは、ホストの自動登録プロセスで使用されます。
 自動登録リクエスト中に、指定された項目によって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。
system.run[]項目は、AllowKey/DenyKey の値に関係なくサポートされます。
 このオプションは、HostInterface が定義されていない場合にのみ使用されます。

HostMetadata

ホストメタデータを定義するオプションパラメータです。ホストメタデータは、ホスト自動登録プロセス（アクティブエージェント）でのみ使用されます。定義されていない場合は、HostMetadataItem から値が取得されます。
 指定された値が 2034 バイトの制限を超える場合、または UTF-8 以外の文字列の場合、エージェントはエラーを通知し、起動しません。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータの取得に使用する Zabbix エージェントアイテムを定義するオプションパラメータです。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。ユーザーパラメータ、パフォーマンスカウンタ、およびエイリアスがサポートされています。system.run[]アイテムは、AllowKey/DenyKey の値に関係なくサポートされます。
HostMetadataItem の値は、自動登録の試行ごとに取得され、ホストの自動登録プロセス（アクティブエージェント）でのみ使用されます。
 自動登録リクエスト中、指定されたアイテムによって返される値が UTF-8 コードポイントの制限である 65535 を超える場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列である必要があります。そうでない場合は無視されます。

ホスト名

カンマ区切りの、大文字と小文字を区別する一意のホスト名のリストです。アクティブチェックに必須で、サーバーに設定されているホスト名と一致する必要があります。値が未定義の場合は、HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、' '、'_'、'-'。最大文字数: ホスト名ごとに 128 文字、行全体で 2048 文字

デフォルト: HostnameItem によって設定

HostnameItem

ホスト名の取得に使用する Zabbix エージェントアイテムを定義するオプションパラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータ、パフォーマンスカウンタ、またはエイリアスはサポートされていませんが、system.run[]アイテムは、AllowKey/DenyKey の値に関わらずサポートされます。
より詳細な説明も参照してください。

デフォルト: `system.hostname`

Include

設定ファイルには、個々のファイルまたはディレクトリ内のすべてのファイルを含めることができます（設定ファイルは、Zabbix エージェントを Windows MSI インストーラーパッケージでインストールした場合はデフォルトで `C:\Program Files\Zabbix Agent` に保存されます。Zabbix エージェントを zip アーカイブでインストールした場合は、インストール時に指定したフォルダに保存されます）。含めるファイルはすべて正しい構文で記述されている必要があります。正しく記述されていない場合、エージェントは起動しません。
 指定したディレクトリ内の関連ファイルのみを含めるには、パターンマッチングにアスタリスクワイルドカード文字を使用できます。
 制限事項については、特記事項を参照してください。

例:

Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d*.conf

ListenBacklog

TCP キュー内の保留中の接続の最大数。
 デフォルト値はシステムに依存するハードコードされた定数です。
 サポートされる最大値はシステムに依存し、値が大きすぎる場合は' 実装で指定された最大値' で自動的に切り捨てられる場合があります。

デフォルト: SOMAXCONN
 範囲: 0 - INT_MAX

ListenIP

エージェントが待受する IP アドレスのカンマ区切りリスト

デフォルト: 0.0.0.0

ListenPort

エージェントは、サーバーからの接続をこのポートで待受します。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイル名

デフォルト: C:\\zabbix_agentd.log
 必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではない

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。コマンドはリモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - Windows イベントログにログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: file

MaxLinesPerSecond

'log'、'logrt'、'eventlog' アクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新規行の最大数。ここで指定した値は、'log'、'logrt'、'eventlog' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新規行を処理します。

デフォルト: 20
 範囲: 1-1000

PerfCounter

システムパフォーマンスカウンタ <perf_counter_path> の、指定された期間 <period> (秒単位) における平均値を表す新しいパラメータ <parameter_name> を定義します。
 構文: <parameter_name>,"<perf_counter_path>",<period>

例えば、過去 1 分間の 1 秒あたりのプロセッサ割り込み回数の平均を取得したい場合は、次のように新しいパラメータ"interrupts" を定義します。

PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60

パフォーマンスカウンタのパスを二重引用符で囲んでいることに注意してください。パラメータ名 (interrupts) は、アイテム作成時のアイテムキーとして使用されます。平均値を計算するためのサンプルは 1 秒ごとに取得されます。
Windows で利用可能なすべてのパフォーマンスカウンタのリストを取得するには、"typeperf -qx" を実行してください。

PerfCounterEn

システムパフォーマンスカウンタ <perf_counter_path> の、指定された期間 <period> (秒単位) における平均値を表す新しいパラメータ <parameter_name> を定義します。PerfCounter とは異なり、perfcounter のパスは英語で指定する必要があります。**Windows Server 2008/Vista** 以降でのみサポートされます。
 構文: <parameter_name>,"<perf_counter_path>",<period>

例えば、過去 1 分間の 1 秒あたりのプロセッサ割り込み回数の平均を取得する場合、新しいパラメータ”interrupts” を次のように定義します。

PerfCounterEn = interrupts,”\Processor(0)\Interrupts/sec”,60

パフォーマンスカウンタのパスを二重引用符で囲んでいることに注意してください。パラメータ名 (interrupts) は、アイテム作成時にアイテムキーとして使用されます。平均値を計算するためのサンプルは 1 秒ごとに取得されます。
 英語の文字列のリストは、次のレジストリキーを表示すると見つかります: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`

RefreshActiveChecks

アクティブチェックのリストを更新する頻度（秒単位）アクティブチェックのリストの更新に失敗した場合、次の更新は 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

サーバー

Zabbix サーバーまたは Zabbix プロキシの IP アドレス（CIDR 表記も可）または DNS 名をカンマ区切りで指定します。受信接続は、ここに指定されたホストからのみ受け付けられます。IPv6 サポートが有効になっている場合、’127.0.0.1’、’::127.0.0.1’、’::ffff:127.0.0.1’ は同等に扱われ、’::/0’ は任意の IPv4 アドレスまたは IPv6 アドレスを許可します。’0.0.0.0/0’ は任意の IPv4 アドレスを許可します。’IPv4 互換 IPv6 アドレス’（プレフィックス 0000::/96）はサポートされていますが、RFC4291 では非推奨となっています。スペースは使用できます。

例:

Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

必須: StartAgents が明示的に 0 に設定されていない場合は必須

ServerActive

アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定。サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、必要に応じてコロンで区切られたポート番号です。
 クラスタ設定は、セミコロンで区切られた 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマで区切って指定できます。プロキシグループを使用する場合を除き、各 Zabbix サーバー/クラスタごとに複数の Zabbix プロキシを指定してはいけません。Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定してはいけません。
 複数の独立した Zabbix サーバーを並行して使用するために、カンマ区切りで複数のアドレスを指定できます。スペースを含めることができます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでそのホストのポートを指定する場合、アドレスは角括弧で囲む必要があります。ポートが指定されていない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーを使用する高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

範囲: (*)

SourceIP

以下の送信元 IP アドレス：
- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 特定のアイテム（web.page.get、net.tcp.port など）の実行中に接続を確立する場合

StartAgents

パッシブチェックを処理する、zabbix_agentd のプリフォークインスタンスの数。0 に設定するとパッシブチェックは無効化され、エージェントはどの TCP ポートも待受しません。

デフォルト: 10
 範囲: 0-100 (*)

Timeout

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。

このパラメータは、以下を含むさまざまな通信処理の継続時間を定義します。

- Zabbix サーバーからの応答待機;
- アクティブチェック におけるアイテム設定要求やアイテムデータを含む、Zabbix サーバーへのリクエスト送信;
- logfile または Windows イベントログ監視によるログデータの取得;
- ハートビートメッセージの送信;
- `vfs.*` チェックの最大継続時間;
- Zabbix エージェントモジュールでの使用;
- バージョン 7.0 より前のサーバーまたはプロキシがタイムアウトなしでチェックを送信する場合のフォールバックとして使用。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる受信接続。パッシブチェックに使用します。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続に応答します（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用した接続に応答します。
cert - TLS と証明書を使用した接続に応答します。

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではありません。

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSConnect

エージェントが Zabbix サーバーまたはプロキシに接続する方法。アクティブチェックに使用されます。指定できる値は 1 つだけです。
unencrypted - 暗号化なしで接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCRLFile

失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータの引数として、すべての文字を渡すことを許可します。以下の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータ。複数のユーザー定義パラメータを指定できます。
 形式: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返さないでください。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスで指定できます。

例:

UserParameter=system.test,dir /b | find /c /v ”” UserParameter=check_cpu,.\custom_script.bat

UserParameterDir

UserParameter コマンドのデフォルトの検索パスです。このパスを指定すると、エージェントはコマンド実行前に、ここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドにはフルパスではなく相対パスの ./ プレフィックスを付けることができます。指定できるエントリは 1 つだけです。

例:

UserParameterDir=C:\opt\myscripts

Note:
(*) ServerActive にリストされているアクティブサーバーの数と、StartAgents で指定されたパッシブチェック用の事前フォーク済みインスタンスの数の合計は、64 未満である必要があります。

6 Zabbix エージェント 2 (Windows)

概要

Zabbix エージェント 2 は新世代の Zabbix エージェントであり、Zabbix エージェントの代わりに使用できます。

Windows 用 Zabbix エージェント 2 の設定ファイル (zabbix_agent2.conf) でサポートされるパラメータを、このセクションに一覧表示します。

パラメータは追加情報なしで一覧表示されています。詳細については、パラメータをクリックしてください。

Parameter	Description
Alias	アイテムキーのエイリアスを設定します。
AllowKey	パターンに一致するアイテムキーの実行を許可します。
BufferSend	データをバッファに N 秒を超えて保持しません。
BufferSize	メモリバッファ内の値の最大数。
ControlSocket	’-R’ オプションで実行時コマンドを送信するために使用される制御ソケット。
DebugLevel	デバッグレベル。
DenyKey	パターンに一致するアイテムキーの実行を拒否します。
EnablePersistentBuffer	アクティブアイテムに対するローカル永続ストレージの使用を有効にします。
ForceActiveChecksOnStart	再起動後、最初に受信した設定に対して直ちにアクティブチェックを実行します。
HeartbeatFrequency	ハートビートメッセージの送信間隔（秒）。
HostInterface	ホストインターフェースを定義するオプションのパラメータ。
HostInterfaceItem	ホストインターフェースの取得に使用するアイテムを定義するオプションのパラメータ。
HostMetadata	ホストメタデータを定義するオプションのパラメータ。
HostMetadataItem	ホストメタデータの取得に使用する Zabbix エージェントアイテムを定義するオプションのパラメータ。
Hostname	ホスト名を定義するオプションのパラメータ。
HostnameItem	ホスト名の取得に使用する Zabbix エージェントアイテムを定義するオプションのパラメータ。
Include	設定ファイルには、個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenIP	エージェントが待ち受ける、カンマ区切りの IP アドレス一覧。
ListenPort	エージェントはこのポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力の種類。
PersistentBufferFile	Zabbix エージェント 2 が SQLite データベースを保持するファイル。
PersistentBufferPeriod	サーバーまたはプロキシへの接続がない場合にデータを保存する期間。
Plugins.<PluginName>.System.Capacity	同時に実行できるプラグインごとのチェック数の上限。
Plugins.Log.MaxLinesPerSecond	’log’ および’logrt’ のアクティブチェックを処理する際に、エージェントが 1 秒あたりに Zabbix サーバーまたはプロキシへ送信する新規行の最大数。
Plugins.SystemRun.LogRemoteCommands	実行されたシェルコマンドの警告としてのログ記録を有効にします。
PluginSocket	ロード可能プラグインとの通信用 UNIX ソケットへのパス。
PluginTimeout	ロード可能プラグインとの接続タイムアウト（秒）。
PerfCounter	指定した期間 <period>（秒）におけるシステムパフォーマンスカウンタ <perf_counter_path> の平均値となる新しいパラメータ <parameter_name> を定義します。

Parameter	Description
PerfCounterEn	指定した期間 <period>（秒）におけるシステムパフォーマンスカウンタ <perf_counter_path> の平均値となる新しいパラメータ <parameter_name> を定義します。PerfCounter と比較して、perfcounter パスは英語で指定する必要があります。
RefreshActiveChecks	アクティブチェックの一覧を更新する頻度。
Server	CIDR 表記を含めることもできる、カンマ区切りの IP アドレス、または Zabbix サーバーと Zabbix プロキシの DNS 名の一覧。
ServerActive	アクティブチェックの取得元となる Zabbix サーバー/プロキシのアドレス、またはクラスタ設定。
SourceIP	送信元 IP アドレス。
StatusPort	設定した場合、エージェントはこのポートで HTTP ステータスリクエスト (http://localhost:<port>/status) を待ち受けます。
Timeout	Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を待機する時間（秒）を指定します。
TLSAccept	受け入れる受信接続の種類。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCertFile	エージェント証明書または証明書チェーンを含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSCipherAll13	OpenSSL の暗号文字列 (TLS 1.3)。証明書ベースおよび PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherCert13	OpenSSL の暗号文字列 (TLS 1.3)。証明書ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSCipherPSK13	OpenSSL の暗号文字列 (TLS 1.3)。PSK ベースの暗号化に対するデフォルトの暗号スイート選択基準を上書きします。
TLSConnect	エージェントが Zabbix サーバーまたはプロキシに接続する方法。
TLSCRLFile	失効した証明書を含むファイルのフルパス名。このパラメータは Zabbix コンポーネント間の暗号化通信に使用されます。
TLSKeyFile	エージェント秘密鍵を含むファイルのフルパス名。Zabbix コンポーネント間の暗号化通信に使用されます。
TLSPSKFile	エージェント事前共有鍵を含むファイルのフルパス名。Zabbix サーバーとの暗号化通信に使用されます。
TLSPSKIdentity	事前共有鍵の識別文字列。Zabbix サーバーとの暗号化通信に使用されます。
TLSServerCertIssuer	許可されるサーバー（プロキシ）証明書の発行者。
TLSServerCertSubject	許可されるサーバー（プロキシ）証明書のサブジェクト。
UnsafeUserParameters	ユーザー定義パラメータの引数にすべての文字を渡すことを許可します。
UserParameter	監視するユーザー定義パラメータ。
UserParameterDir	UserParameter コマンドのデフォルト検索パス。

すべてのパラメータは、明示的に必須と記載されていない限り、必須ではありません。

注意:

- デフォルト値はプロセスのデフォルトを反映しており、配布される設定ファイル内の値ではありません。
- 値では環境変数 をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

エイリアス

アイテムキーのエイリアスを設定します。長くて複雑なアイテムキーを、より短くてシンプルなものに置き換えることができます。
 Alias パラメータは複数指定できます。同じ *Alias* キーを持つパラメータは複数指定できません。
 異なる *Alias* キーが同じアイテムキーを参照することは可能です。
 エイリアスは *HostMetadataItem* では使用できますが、*HostnameItem* パラメータでは使用できません。

例 1: サーバーからページングファイルの使用率をパーセンテージで取得する。

Alias=pg_usage:perf_counter[\Paging File(_Total)% Usage]

これで、短縮キー **pg_usage** を使用してデータを取得できるようになりました。

例 2: デフォルトおよびカスタムパラメータを使用した CPU 負荷の取得。

Alias=cpu.load:system.cpu.load Alias=cpu.load[*]:system.cpu.load[*]

これにより、**cpu.load** キーを使用してデフォルトのパラメータで CPU 負荷を取得し、**cpu.load[percpu,avg15]** を使用して CPU 負荷に関する特定のデータを取得できます。

例 3: 同じ検出項目を処理する複数の低レベル検出 ルールを実行する。

Alias=vfs.fs.discovery[*]:vfs.fs.discovery

これで、**vfs.fs.discovery** を使用して、ルールごとに異なるパラメータを持つ複数の検出ルールを設定できるようになりました。(例: **vfs.fs.discovery[foo]**、**vfs.fs.discovery[bar]** など)

AllowKey

パターンに一致するアイテムキーの実行を許可します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
DenyKey と組み合わせて、複数のキー一致ルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

BufferSend

バッファから Zabbix サーバーへ値を送信する間隔（秒単位）です。
 バッファがいっぱいの場合、データはより早く送信されます。

デフォルト: 5
 範囲: 1-3600

BufferSize

メモリバッファ内の値の最大数。バッファがいっぱいになった場合、エージェントは収集したすべてのデータを Zabbix サーバーまたはプロキシに送信します。
 このパラメータは、永続バッファが無効になっている場合（*EnablePersistentBuffer=0*）のみ使用してください。

デフォルト: 1000
 範囲: 2-65535

ControlSocket

’-R’ オプションを指定したランタイムコマンドの送信に使用する制御ソケット

デフォルト: `\\.\pipe\agent.sock`

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

DenyKey

パターンに一致するアイテムキーの実行を拒否します。キーパターンはワイルドカード表現で、”*” 文字を含む任意の数の文字に一致します。
AllowKey と組み合わせて、複数のキーマッチングルールを定義できます。パラメータは出現順序に従って 1 つずつ処理されます。エージェントチェックの制限も参照してください。

EnablePersistentBuffer

アクティブアイテム用のローカル永続ストレージの使用を有効にします。永続ストレージが無効になっている場合は、メモリバッファが使用されます。

デフォルト: 0
 値: 0 - 無効、1 - 有効

ForceActiveChecksOnStart

再起動後、最初に受信した構成に対してアクティブチェックを直ちに実行します。プラグインごとの設定パラメータとしても可能です。例: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

デフォルト: 0
 値: 0 - 無効、1 - 有効

HeartbeatFrequency

ハートビートメッセージの送信頻度。(秒単位）アクティブチェックの可用性を監視するために使用されます。
0 - ハートビートメッセージは無効です。

デフォルト: 60
 範囲: 0-3600

HostInterface

ホストインターフェースを定義する追加パラメーターです。ホストインターフェースは、ホストの自動登録プロセスで使用されます。定義されていない場合は、HostInterfaceItem から値が取得されます。
 値が 255 文字の制限を超える場合、エージェントはエラーを通知して起動しません。

範囲: 0-255 文字

HostInterfaceItem

ホストインターフェースの取得に使用するアイテムを定義する追加パラメータです。
 ホストインターフェースは、ホストの自動登録プロセスで使用されます。このオプションは、HostInterface が定義されていない場合にのみ使用されます。
system.run[]アイテムは、AllowKey/DenyKey の値に関係なくサポートされます。
 自動ディスカバリリクエスト中に、指定された項目によって返される値が 255 文字の制限を超えている場合、エージェントは警告メッセージをログに記録します。

HostMetadata

ホストメタデータを定義する追加パラメータです。ホストメタデータは、ホスト自動登録プロセス（アクティブエージェント）でのみ使用されます。定義されていない場合は、HostMetadataItem から値が取得されます。
 指定された値が 2034 バイトの制限を超える場合、または UTF-8 以外の文字列の場合、エージェントはエラーを通知して起動しません。

範囲: 0-2034 バイト

HostMetadataItem

ホストメタデータの取得に使用する項目を定義する追加パラメータです。このオプションは、HostMetadata が定義されていない場合にのみ使用されます。ユーザーパラメータとエイリアスがサポートされています。system.run[] 項目は、AllowKey/DenyKey の値に関係なくサポートされます。
HostMetadataItem の値は、自動登録の試行ごとに取得され、ホストの自動登録プロセスでのみ使用されます。
 自動登録リクエスト中、指定されたアイテムによって返される値が UTF-8 コードポイントの上限である 65535 を超える場合、エージェントは警告メッセージをログに記録します。アイテムによって返される値は UTF-8 文字列である必要があります。そうでない場合は無視されます。

ホスト名

カンマ区切りの、大文字と小文字を区別する一意のホスト名のリストです。アクティブチェックに必須で、サーバーに設定されているホスト名と一致する必要があります。値が未定義の場合は、HostnameItem から取得されます。
 使用可能な文字: 英数字、'.'、''、'_'、'-'。最大文字数: ホスト名ごとに 128 文字、行全体で 2048 文字。

デフォルト: HostnameItem によって設定

HostnameItem

ホスト名の取得に使用する項目を定義する追加パラメータです。このオプションは、Hostname が定義されていない場合にのみ使用されます。ユーザーパラメータやエイリアスはサポートされていませんが、system.run[] アイテムは、AllowKey/DenyKey の値に関わらずサポートされます。

デフォルト: `system.hostname`

Include

設定ファイルには、個々のファイルまたはディレクトリ内のすべてのファイルを含めることができます（設定ファイルは、Zabbix エージェントを Windows MSI インストーラーパッケージでインストールした場合にはデフォルトで `C:\Program Files\Zabbix Agent 2` に保存されます。Zabbix エージェントを zip アーカイブでインストールした場合には、インストール時に指定したフォルダに保存されます）含めるファイルはすべて正しい構文で記述する必要があります。正しく記述されていない場合、エージェントは起動しません。パスは、*zabbix_agent2.conf* ファイルの場所からの相対パスで指定できます。（例：`Include=.\zabbix_agent2.d\plugins.d\*.conf`）
 指定したディレクトリ内の関連ファイルのみを含めるには、パターンマッチングでアスタリスクワイルドカード文字を使用できます。
 制限事項については、特記事項を参照してください。

例：

Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d*.conf

ListenIP

エージェントが Listen する IP アドレスをカンマ区切りで指定します。最初の IP アドレスは、Zabbix サーバーに接続している場合、アクティブなチェックリストを取得するために Zabbix サーバーに送信されます。

デフォルト: `0.0.0.0`

ListenPort

エージェントは、サーバーからの接続をこのポートで待ち受けします。

デフォルト: 10050
 範囲: 1024-32767

LogFile

エージェントのログファイル名

デフォルト: `c:\\zabbix_agent2.log`
 必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません。

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
console - 標準出力にログを書き出します。

デフォルト: `file`

PersistentBufferFile

Zabbix エージェント 2 が SQLite データベースを保存するファイル。完全なファイル名を指定する必要があります。このパラメータは、永続バッファが有効になっている場合（*EnablePersistentBuffer=1*）のみ使用されます。

PersistentBufferPeriod

サーバーまたはプロキシへの接続がない場合にデータを保存する期間です。古いデータは失われます。ログデータは保持されます。このパラメータは、永続バッファが有効になっている場合（*EnablePersistentBuffer=1*）のみ使用されます。

デフォルト: 1h
 範囲: 1m-365d

Plugins.<PluginName>.System.Capacity

<PluginName> プラグインごとに同時に実行できるチェックの上限。

デフォルト: 1000 範囲: 1-1000

Plugins.Log.MaxLinesPerSecond

'log'、'logrt'、'eventlog' アクティブチェックを処理する際に、エージェントが Zabbix サーバーまたはプロキシに送信する 1 秒あたりの新規行の最大数。ここで指定した値は、'log'、'logrt'、'eventlog' アイテムキーで指定された'maxlines' パラメータによって上書きされます。
 注: Zabbix は、ログアイテム内で必要な文字列を検索するために、*MaxLinesPerSecond* で設定された値の 10 倍の新規行を処理します。

デフォルト: 20
 範囲: 1-1000

Plugins.SystemRun.LogRemoteCommands

実行されたシェルコマンドを警告としてログに記録します。コマンドはリモートで実行された場合にのみログに記録されます。system.run[] が HostMetadataItem、HostInterfaceItem、または HostnameItem パラメータによってローカルで起動された場合は、ログエントリは作成されません。

デフォルト: 0
 値: 0 - 無効、1 - 有効

PluginSocket

ロード可能なプラグイン通信用の UNIX ソケットへのパス

デフォルト: `\\.\pipe\agent.plugin.sock`

PluginTimeout

読み込み可能なプラグインとの接続のタイムアウト（秒単位）

デフォルト: `Timeout`
 範囲: 1-30

PerfCounter

システムパフォーマンスカウンタ <perf_counter_path> の、指定された期間 <period> (秒単位) における平均値を表す新しいパラメータ <parameter_name> を定義します。
 構文: <parameter_name>,"<perf_counter_path>",<period>

例えば、過去 1 分間の 1 秒あたりのプロセッサ割り込み回数の平均を取得したい場合は、次のように新しいパラメータ"interrupts"を定義します。

PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60

パフォーマンスカウンタのパスを二重引用符で囲んでいることに注意してください。パラメータ名 (interrupts) は、アイテム作成時のアイテムキーとして使用されます。平均値を計算するためのサンプルは 1 秒ごとに取得されます。
Windows で利用可能なすべてのパフォーマンスカウンタのリストを取得するには、"typeperf -qx" を実行してください。

PerfCounterEn

システムパフォーマンスカウンタ <perf_counter_path> の、指定された期間 <period> (秒単位) における平均値を表す新しいパラメータ <parameter_name> を定義します。PerfCounter とは異なり、perfcounter のパスは英語環境用で指定する必要があります。**Windows Server 2008/Vista** 以降でのみサポートされます。
 構文: <parameter_name>,"<perf_counter_path>",<period>

例えば、過去 1 分間の 1 秒あたりのプロセッサ割り込み回数の平均を取得する場合、新しいパラメータ”interrupts” を次のように定義します。

PerfCounterEn = interrupts,”\Processor(0)\Interrupts/sec”,60

パフォーマンスカウンタのパスを二重引用符で囲んでいることに注意してください。パラメータ名 (interrupts) は、アイテム作成時にアイテムキーとして使用されます。平均値を計算するためのサンプルは 1 秒ごとに取得されます。
 英語環境用の文字列のリストは、次のレジストリキーを表示すると確認出来ます。: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009

RefreshActiveChecks

アクティブチェックのリストを更新する頻度（秒単位）。アクティブチェックのリストの更新に失敗した場合、次の更新は 60 秒後に試行されます。

デフォルト: 5
 範囲: 1-86400

Server

Zabbix サーバーまたは Zabbix プロキシの IP アドレス（CIDR 表記も可）または DNS 名をカンマ区切りで指定します。ここで指定したホストからの接続のみが受信されます。IPv6 サポートが有効な場合、’127.0.0.1’、’::127.0.0.1’、’::ffff:127.0.0.1’ は同等に扱われ、’::/0’ は任意の IPv4 または IPv6 アドレスを許可します。’0.0.0.0/0’ は任意の IPv4 アドレスを許可します。スペースも使用できます。このパラメータを指定しない場合、パッシブチェックは無効になり、エージェントはどの TCP ポートも待受しません。

例：

Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com

ServerActive

アクティブチェックを取得するための Zabbix サーバー/プロキシのアドレス、またはクラスタ設定です。
サーバー/プロキシのアドレスは、IP アドレスまたは DNS 名と、必要に応じてコロンで区切られたポート番号で構成されます。
 クラスタ設定は、セミコロンで区切られた 1 つ以上のサーバーまたはプロキシグループメンバーのアドレスです。
複数の Zabbix サーバー/クラスタおよび Zabbix プロキシを、カンマで区切って指定できます。
プロキシグループを使用しない限り、各 Zabbix サーバー/クラスタごとに複数の Zabbix プロキシを指定してはいけません。
Zabbix プロキシを指定した場合、そのプロキシに対応する Zabbix サーバー/クラスタは指定してはいけません。
 複数の独立した Zabbix サーバーを並行して使用するために、カンマ区切りで複数のアドレスを指定できます。
スペースを含めることができます。
 ポートが指定されていない場合は、デフォルトのポートが使用されます。
IPv6 アドレスでそのホストのポートを指定する場合、IPv6 アドレスは角括弧で囲む必要があります。
ポートを指定しない場合、IPv6 アドレスの角括弧は省略可能です。
 このパラメータが指定されていない場合、アクティブチェックは無効になります。

Zabbix プロキシの例:

```
ServerActive=127.0.0.1:10051
```

Zabbix プロキシグループの例:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

複数サーバーの例:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

2 つのクラスタと 1 つのサーバーによる高可用性の例:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

以下の送信元 IP アドレス：
- Zabbix サーバーまたは Zabbix プロキシへの送信接続
- 特定のアイテム（web.page.get、net.tcp.port など）の実行中に接続を確立する場合

StatusPort

設定されている場合、エージェントはこのポートで HTTP ステータスリクエスト（http://localhost:<port>/status）を待受します。

範囲: 1024-32767

タイムアウト

Zabbix プロキシまたはサーバーとの接続確立およびデータ交換を行う際に、どのくらいの時間（秒）待機するかを指定します。

このパラメータは、以下を含むさまざまな通信処理の継続時間を定義します。

- Zabbix サーバーからの応答待機時間。
- アクティブチェック におけるアイテム設定要求およびアイテムデータを含む、Zabbix サーバーへのリクエスト送信時間。
- logfile または Windows イベントログ監視によるログデータ取得時間。
- ハートビートメッセージの送信時間。
- `vfs.*` チェックの最大継続時間。
- バージョン 7.0 より前のサーバーまたはプロキシがタイムアウトなしでチェックを送信する場合のフォールバックとして使用されます。

このタイムアウトは、Web インターフェースでタイムアウト設定を構成できるエージェントチェック（グローバル、プロキシ、またはアイテム単位）には使用されません。

デフォルト: 3
 範囲: 1-30

TLSAccept

受け入れる受信接続。パッシブチェックに使用します。複数の値をカンマで区切って指定できます。
unencrypted - 暗号化なしの接続に応答します（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用した接続に応答します。
cert - TLS と証明書を使用した接続に応答します。

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではありません。

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるエージェント証明書または証明書チェーンを含むファイルのフルパス名

TLSCipherAll13

OpenSSL 暗号文字列 (TLS 1.3)。証明書ベースおよび PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSCipherCert13

OpenSSL 暗号文字列 (TLS 1.3) 証明書ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

このパラメータは `TLSAccept=cert,psk` と同時に使用できないことに注意してください。証明書接続 (`TLSConnect=cert`) の場合は、代わりに TLSCipherAll13 を使用してください。

TLSCipherPSK13

OpenSSL 暗号文字列 (TLS 1.3) 。PSK ベースの暗号化におけるデフォルトの暗号スイート選択基準を上書きします。

例:

TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256

TLSConnect

エージェントが Zabbix サーバーまたはプロキシに接続する方式。アクティブチェックに使用されます。指定できる値は 1 つだけです。
unencrypted - 暗号化なしで接続（デフォルト）
psk - TLS と事前共有鍵（PSK）を使用して接続
cert - TLS と証明書を使用して接続

必須：TLS 証明書または PSK パラメータが定義されている場合は必須、そうでない場合は必須ではない。

TLSCRLFile

失効した証明書を含むファイルのフルパス名。このパラメータは、Zabbix コンポーネント間の暗号化通信に使用されます。

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるエージェント秘密鍵を含むファイルのフルパス名

TLSPSKFile

Zabbix サーバーとの暗号化通信に使用されるエージェントの事前共有キーを含むファイルのフルパス名

TLSPSKIdentity

Zabbix サーバーとの暗号化通信に使用される、事前共有キーの ID 文字列

TLSServerCertIssuer

許可されたサーバー（プロキシ）証明書の発行者

TLSServerCertSubject

許可されたサーバー（プロキシ）証明書のサブジェクト

UnsafeUserParameters

ユーザー定義パラメータの引数として、すべての文字を渡すことを許可します。以下の文字は許可されません: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @
 改行文字も許可されません。

デフォルト: 0
 値: 0 - 許可しない、1 - 許可する

UserParameter

監視するユーザー定義パラメータ。複数のユーザー定義パラメータを指定できます。
 形式: UserParameter=< キー >,< シェルコマンド >
 シェルコマンドは空文字列や EOL のみを返さないでください。UserParameterDir パラメータが指定されている場合、シェルコマンドは相対パスで指定できます。

例:

UserParameter=system.test,dir /b | find /c /v "" UserParameter=check_cpu,.\custom_script.bat

UserParameterDir

UserParameter コマンドのデフォルト検索パスです。これを指定すると、エージェントはコマンド実行前にここで指定されたディレクトリに作業ディレクトリを変更します。これにより、UserParameter コマンドにはフルパスではなく相対パス .\プレフィックスを付けることができます。
 指定できるエントリは 1 つだけです。

例:

UserParameterDir=C:\opt\myscripts

7 Zabbix エージェント 2 プラグイン

概要

このセクションには、Zabbix エージェント 2 プラグインの設定ファイルパラメータの説明が含まれています。特定のプラグインに関する情報にアクセスするには、サイドバーを使用してください。

1 Ceph プラグイン

概要

このセクションでは、Ceph Zabbix エージェント 2 のプラグイン設定ファイル（ceph.conf）でサポートされているパラメータを一覧表示します。

次の点に注意してください。

- デフォルト値は、同梱の設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- "#" で始まるコメントは、行頭でのみサポートされています。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.ApiKey				Ceph への接続に使用するデフォルトの API キーです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Ceph.Default.User				Ceph への接続に使用するデフォルトのユーザー名です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.Uri			https://localhost:8003	8003への接続に使用するデフォルトの URI です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（含まれていても無視されます）。 URI 形式に一致している必要があります。 サポートされるスキームは `https` のみです。スキームは省略できます。 ポートは省略できます（デフォルト =8003）。 例: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.InsecureSkipVerify	false / true		false	http クライアントがサーバーの証明書チェーンおよびホスト名を検証するかどうかを決定します。 *true* の場合、TLS はサーバーから提示された任意の証明書と、その証明書内の任意のホスト名を受け入れます。このモードでは、TLS は中間者攻撃を受けやすくなります（テスト用途でのみ使用してください）。
Plugins.Ceph.KeepAlive		60-900	300	未使用のプラグイン接続を閉じるまでの最大待機時間（秒）です。
Plugins.Ceph.Sessions.<SessionName>.ApiKey				名前付きセッションの API キーです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Ceph.Sessions.<SessionName>.User				名前付きセッションのユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Ceph.Sessions.<SessionName>.Uri				名前付きセッションの接続文字列です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（含まれていても無視されます）。 URI 形式に一致している必要があります。 サポートされるスキームは `https` のみです。スキームは省略できます。 ポートは省略できます（デフォルト =8003）。 例: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout		1-30	global timeout	リクエスト実行のタイムアウトです（リクエストを中断する前に、その完了を待機する時間（秒））。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

2 Docker プラグイン

概要

このセクションでは Docker Zabbix エージェント 2 プラグイン設定ファイル (docker.conf) でサポートされているパラメーターを一覧表示します。

注意点:

- デフォルト値は、出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- 値は環境変数をサポートします。
- Zabbix はBOMなしの UTF-8 エンコーディングでのみ設定ファイルをサポートします。
- 行の先頭が”＃”で始まった場合のみコメントとして認識されます。

パラメーター

パラメーター	必須	範囲	デフォルト	説明
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Docker デーモンの unix ソケットの場所。 スキームが含まれている必要があります (`unix://`のみサポート)。
Plugins.Docker.Timeout		1-30	global timeout	リクエストの実行タイムアウト (リクエストが完了するのを待ってからシャットダウンするまでの数秒の時間)。

参照:

- 一般的な Zabbix エージェント 2 の設定パラメーターの説明: Zabbix エージェント 2 (UNIX) / Zabbix エージェント 2 (Windows)
- プラグインの設定手順

3 Ember+ プラグイン

概要

このセクションでは、Ember+ Zabbix エージェント 2 プラグイン設定ファイル (ember.conf) でサポートされるパラメータの一覧を示します。

Ember+ プラグインはロード可能なプラグインであり、Ember+ プラグインリポジトリ で利用可能で、詳細な説明も記載されています。

このプラグインは現在、ソースコードからビルドする形式のみで利用可能です (Unix および Windows 対応)

以下の点にご注意ください。

- デフォルト値はプロセスのデフォルト値を反映しており、同梱の設定ファイルの値は反映されていません。
- 値は環境変数 をサポートしています。
- Zabbix は BOM なしの UTF-8 エンコードの設定ファイルのみをサポートしています。
- ”#” で始まるコメントは、行の先頭でのみサポートされます。

オプション

Parameter	Description
-V --version	プラグインのバージョンおよびライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（短縮形）。

パラメータ

パラメータ	必須	範囲	デフォルト	説明
Plugins.EmberPlus.Default.Uri			tcp://localhost:9092	接続するデフォルトの URI。サポートされているスキーマは `tcp://`のみです。スキーマは省略可能です。埋め込まれた認証情報は無視されます。
Plugins.EmberPlus.KeepAlive		60-900	300	未使用のプラグイン接続を閉じるまでの最大待機時間（秒）
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9092	名前付きセッションの接続先 URI。サポートされているスキーマは `tcp://`のみです。スキーマは省略可能です。埋め込まれた認証情報は無視されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.EmberPlus.System.Path				Ember+ プラグイン実行ファイルへのパス 使用例: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-`
Plugins.EmberPlus.Timeout		1-30	グローバルタイムアウト	セッション内での初回接続時および後続操作時にサーバーからの応答を待機する時間（秒）

参照:

- Zabbix エージェント 2 の一般的な設定パラメータの説明: Zabbix エージェント 2 (UNIX) / Zabbix エージェント 2 (Windows)
- プラグイン の設定手順

4 Memcached プラグイン

概要

このセクションでは、Memcached Zabbix エージェント 2 プラグイン設定ファイル（memcached.conf）でサポートされるパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、同梱の設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- "#" で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Memcached.Default.Password				Memcached への接続に使用するデフォルトのパスワードです。 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Memcached.Default.Uri			tcp://localhost:11211	Memcached への接続に使用するデフォルトの URI です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略できます。
				ポートは省略できます（デフォルト =11211）。
				例: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Default.User				Memcached への接続に使用するデフォルトのユーザー名です。 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Memcached.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）です。
Plugins.Memcached.Sessions.<SessionName>.Password				名前付きセッションのパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Plugins.Memcached.Sessions.<SessionName>.Uri				名前付きセッションの接続文字列です。
				<SessionName> - アイテムキーで使用するセッション名を定義します。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略できます。
				ポートは省略できます（デフォルト =11211）。
				例: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Sessions.<SessionName>.User				名前付きセッションのユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Memcached.Timeout		1-30	global timeout	リクエスト実行のタイムアウトです（リクエストを停止する前に、その完了を待機する時間（秒））。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

5 Modbus プラグイン

概要

このセクションでは、Modbus Zabbix エージェント 2 プラグインの設定ファイル（modbus.conf）でサポートされているパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされています。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint				Endpoint は、プロトコルスキーム、ホストアドレス、ポートまたはシリアルポート名と属性で構成される接続文字列です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Modbus.Sessions.<SessionName>.SlaveID				名前付きセッションのスレーブ ID。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 例: `Plugins.Modbus.Sessions.MB1.SlaveID=20` item key のスレーブ ID パラメータで指定された値が空の場合にのみ、この名前付きセッションパラメータがチェックされることに注意してください。
Plugins.Modbus.Sessions.<SessionName>.Timeout				名前付きセッションのタイムアウト（秒）。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 例: `Plugins.Modbus.Sessions.MB1.Timeout=2`

リクエスト実行タイムアウト（リクエストを停止する前に、完了を待機する時間）を設定する必要がある場合は、item configuration フォームを使用してください。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

6 MongoDB プラグイン

概要

このセクションでは、MongoDB Zabbix エージェント 2 プラグインの設定ファイル（mongo.conf）でサポートされるパラメータを一覧表示します。

MongoDB プラグインはロード可能なプラグインであり、MongoDB plugin repository で利用でき、詳細に説明されています。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートします。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

オプション

パラメータ	説明
-V --version	プラグインのバージョンとライセンス情報を出力します
-h --help	省略形のヘルプを表示します)

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Default.Password				MongoDB への接続に使用するデフォルトのパスワードです。 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MongoDB.Default.Uri				MongoDB への接続に使用するデフォルトの URI です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（含まれていても無視されます）。 URI 形式に一致している必要があります。 サポートされるのは tcp スキームのみです。スキームは省略できます。 ポートは省略できます（default=27017）。 例: `tcp://127.0.0.1:27017`, `tcp:localhost`, `localhost`
Plugins.MongoDB.Default.User				MongoDB への接続に使用するデフォルトのユーザー名です。 アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MongoDB.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）です。
Plugins.MongoDB.Sessions.<SessionName>.Password				名前付きセッションのパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	(yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full)			ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名です。Zabbix エージェント 2 と監視対象データベース間の暗号化通信で使用されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	(yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile is specified			エージェント証明書または証明書チェーンを含むファイルのフルパス名です。Zabbix エージェント 2 と監視対象データベース間の暗号化通信で使用されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect				Zabbix エージェント 2 と監視対象データベース間の通信に使用する暗号化タイプです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *required* - TLS 接続を必須にします。 *verify_ca* - 証明書を検証します。 *verify_full* - 証明書と IP アドレスを検証します。 プラグインバージョン 1.2.1 以降でサポートされています。
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	(yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile is specified			監視対象データベースとの暗号化通信に使用するデータベース秘密鍵を含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.Sessions.<SessionName>.Uri				名前付きセッションの接続文字列です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（含まれていても無視されます）。 URI 形式に一致している必要があります。 サポートされるのは tcp スキームのみです。スキームは省略できます。 ポートは省略できます（default=27017）。 例: `tcp://127.0.0.1:27017`, `tcp:localhost`, `localhost`

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.User				名前付きセッションのユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MongoDB.System.Path				MongoDB プラグイン実行ファイルへのパスです。 使用例: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout		1-30	global timeout	リクエスト実行のタイムアウトです（リクエストが完了するまで待機する秒数で、この時間を超えるとリクエストは停止されます）。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

7 MQTT プラグイン

概要

このセクションでは、MQTT Zabbix エージェント 2 プラグイン設定ファイル（mqtt.conf）でサポートされるパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.Password				MQTT への接続に使用するデフォルトのパスワードです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.Default.TLSCAFile				Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信において、ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名です。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.Default.TLSCertFile				Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルのフルパス名です。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.Default.TLSKeyFile				Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用する、MQTT 秘密鍵を含むファイルのフルパス名です。名前付きセッションで値が指定されていない場合に使用されます。

Plugins.MQTT.Default.Topic				MQTT サブスクリプション用のデフォルトトピックです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				トピックにはワイルドカード（”+”、”#”）を含めることができます。
				例: `path/to/file`
				`path/to/#`
				`path/+/topic`

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.Url	no		tcp://localhost:1883	デフォルトの MQTT ブローカー接続文字列です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 クエリパラメータを含めないでください。 URL 形式に一致している必要があります。 サポートされるスキーム: tcp（デフォルト）、ws、tls。スキームは省略できます。 ポートは省略できます（デフォルト =1883）。 例: tcp://host:1883 localhost ws://host:8080
Plugins.MQTT.Default.User	no			MQTT への接続に使用するデフォルトのユーザー名です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.MQTT.Sessions.<SessionName>.Password	no			名前付きセッションのパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	no			ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名で、Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	no			エージェント証明書または証明書チェーンを含むファイルのフルパス名で、Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	no			Zabbix エージェント 2 と MQTT ブローカー間の暗号化通信に使用する MQTT 秘密鍵を含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MQTT.Sessions.<SessionName>.Topic	no			MQTT サブスクリプション用の名前付きセッションのトピックです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 トピックにはワイルドカード（”+”、”#”）を含めることができます。 例: path/to/file path/to/# path/+/topic
Plugins.MQTT.Sessions.<SessionName>.Url	no			名前付きセッションの接続文字列です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 クエリパラメータを含めないでください。 URL 形式に一致している必要があります。 サポートされるスキーム: tcp（デフォルト）、ws、tls。スキームは省略できます。 ポートは省略できます（デフォルト =1883）。 例: tcp://host:1883 localhost ws://host:8080
Plugins.MQTT.Sessions.<SessionName>.User	no			名前付きセッションのユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

リクエスト実行タイムアウト（リクエストの完了を待機してから停止するまでの時間）を設定する必要がある場合は、item configuration フォームを使用してください。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

8 MSSQL プラグイン

概要

このセクションでは、MSSQL Zabbix エージェント 2 プラグインの設定ファイル (mssql.conf) でサポートされているパラメータの一覧を示します。

MSSQL プラグインは読み込み可能なプラグインであり、MSSQL プラグインリポジトリ で利用可能で、詳細な説明も記載されています。

以下の点にご注意ください。

- デフォルト値は出荷時設定ファイルの値ではなく、プロセスのデフォルト値です。
- 値は環境変数 をサポートしています。
- Zabbix は BOM なしの UTF-8 エンコードの設定ファイルのみをサポートしています。
- ”#” で始まるコメントは、行の先頭でのみサポートされます。

オプション

Parameter	Description
-V --version	プラグインのバージョンおよびライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（短縮形）。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.CustomQueriesDir			Unix システムでは /usr/local/share/zabbix/custom-queries/mssql、Windows システムでは *:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL。ここで * は ProgramFiles 環境変数のドライブ名です	プラグインが実行できるカスタムクエリを含む、ユーザー定義の.sql ファイルが格納されたディレクトリへのファイルパスを指定します。プラグインの起動時に設定されたディレクトリ内の利用可能な.sql ファイルをすべて読み込みます。つまり、カスタムクエリファイルへの変更は、プラグインを再起動するまで反映されません。プラグインは Zabbix エージェント 2 とともに起動および停止します。
Plugins.MSSQL.CustomQueriesEnabled			false	設定すると、mssql.custom.query アイテムキーの実行を有効にします。無効な場合、カスタムクエリディレクトリパスからクエリは読み込まれません。
Plugins.MSSQL.Default.CACertPath				MSSQL サーバーの証明書を発行した認証局 (CA) の公開鍵証明書へのデフォルトのファイルパスです。証明書は PEM 形式である必要があります。
Plugins.MSSQL.Default.Database				接続先のデフォルトのデータベース名です。

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Default.Encrypt				デフォルトの接続暗号化タイプを指定します。指定可能な値は以下のとおりです: *true* - プラグインとサーバー間で送信されるデータは暗号化されます。 *false* - プラグインとサーバー間で送信されるデータは、ログインパケットを除き暗号化されません。 *strict* - プラグインとサーバー間で送信されるデータは、TDS8 を使用して E2E で暗号化されます。 *disable* - プラグインとサーバー間で送信されるデータは暗号化されません。
Plugins.MSSQL.Default.HostNameInCertificate				デフォルトで使用する MSSQL サーバー証明書のコモンネーム (CN) です。
Plugins.MSSQL.Default.Password				保護された MSSQL サーバーにデフォルトで送信するパスワードです。
Plugins.MSSQL.Default.TLSMinVersion				デフォルトで使用する最小 TLS バージョンです。指定可能な値は: `1.0`, `1.1`, `1.2`, `1.3`。
Plugins.MSSQL.Default.TrustServerCertificate				デフォルトで、検証を行わずにプラグインがサーバー証明書を信頼するかどうかを指定します。指定可能な値: `true`, `false`。
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	接続先のデフォルト URI です。サポートされるスキーマは `sqlserver://` のみです。スキーマは省略できます。埋め込み認証情報は無視されます。
Plugins.MSSQL.Default.User				保護された MSSQL サーバーにデフォルトで送信するユーザー名です。
Plugins.MSSQL.KeepAlive		60-900	300	未使用のプラグイン接続を閉じるまでの最大待機時間 (秒) です。
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				名前付きセッション用の MSSQL サーバー証明書を発行した認証局 (CA) の公開鍵証明書へのファイルパスです。証明書は PEM 形式である必要があります。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Database				名前付きセッションの接続先データベース名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				名前付きセッションの接続暗号化タイプを指定します。指定可能な値は以下のとおりです: *true* - プラグインとサーバー間で送信されるデータは暗号化されます。 *false* - プラグインとサーバー間で送信されるデータは、ログインパケットを除き暗号化されません。 *strict* - プラグインとサーバー間で送信されるデータは、TDS8 を使用して E2E で暗号化されます。 *disable* - プラグインとサーバー間で送信されるデータは暗号化されません。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				名前付きセッション用の MSSQL サーバー証明書のコモンネーム (CN) です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.Password				名前付きセッション用に保護された MSSQL サーバーへ送信するパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				名前付きセッションで使用する最小 TLS バージョンです。指定可能な値は: `1.0`, `1.1`, `1.2`, `1.3`。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				名前付きセッションで、検証を行わずにプラグインがサーバー証明書を信頼するかどうかを指定します。指定可能な値: `true`, `false`。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Sessions.<SessionName>.Uri	no		sqlserver://localhost:1433	名前付きセッションの接続先 URI です。サポートされるスキーマは `sqlserver://` のみです。スキーマは省略できます。埋め込み認証情報は無視されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.Sessions.<SessionName>.User	no			名前付きセッション用に保護された MSSQL サーバーへ送信するユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.MSSQL.System.Path	no			MSSQL プラグイン実行ファイルへのパスです。 MSSQL プラグインのグローバル設定です。すべての接続に適用されます。 使用例: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
Plugins.MSSQL.Timeout	no	1-30	global timeout	最初の接続時およびセッション内の後続操作で、サーバーからの応答を待機する時間 (秒) です。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

9 MySQL プラグイン

概要

このセクションでは、MySQL Zabbix エージェント 2 プラグイン設定ファイル（mysql.conf）でサポートされるパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.CallTimeout	no	1-30	global timeout	リクエストの完了を待機する最大時間（秒）です。
Plugins.Mysql.CustomQueriesPath	no		`/usr/local/share/zabbix/custom.queries/mysql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` for Windows systems, where * is the drive name from the ProgramFiles environment variable	`mysql.custom.query` アイテムキーを含むユーザーパラメータが格納されたディレクトリのフルパス名です。
Plugins.Mysql.CustomQueriesEnabled	no		false	設定すると、`mysql.custom.query` アイテムキーの実行を有効にします。無効な場合、カスタムクエリディレクトリパスからクエリは読み込まれません。

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.Default.Password				MySQL への接続に使用するデフォルトのパスワードです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.Default.TLSCAFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Zabbix agent 2 と監視対象データベース間の暗号化通信において、ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名です。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.Default.TLSCertFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Zabbix agent 2 と監視対象データベース間の暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルのフルパス名です。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.Default.TLSConnect				Zabbix agent 2 と監視対象データベース間の通信に使用する暗号化タイプです。名前付きセッションで値が指定されていない場合に使用されます。 サポートされる値: *required* - TLS 接続を必須にします。 *verify_ca* - 証明書を検証します。 *verify_full* - 証明書と IP アドレスを検証します。
Plugins.Mysql.Default.TLSKeyFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Zabbix agent 2 と監視対象データベース間の暗号化通信に使用する、データベース秘密鍵を含むファイルのフルパス名です。名前付きセッションで値が指定されていない場合に使用されます。

Plugins.Mysql.Default.Uri			tcp://localhost:3306	MySQL への接続に使用するデフォルトの URI です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略できます。
				ポートは省略できます（default=3306）。
				例: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User				MySQL への接続に使用するデフォルトのユーザー名です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Mysql.KeepAlive	no	60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）です。
Plugins.Mysql.Sessions.<SessionName>.Password				名前付きセッションのパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	no (yes, if Plugins.Mysql.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名です。Zabbix agent 2 と監視対象データベース間の暗号化通信に使用されます。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile is specified			Zabbix agent 2 と監視対象データベース間の暗号化通信に使用する、エージェント証明書または証明書チェーンを含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.Sessions.<SessionName>.TLSConnect				Zabbix agent 2 と監視対象データベース間の通信に使用する暗号化タイプです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *required* - TLS 接続を必須にします。 *verify_ca* - 証明書を検証します。 *verify_full* - 証明書と IP アドレスを検証します。
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSCertFile is specified			Zabbix agent 2 と監視対象データベース間の暗号化通信に使用する、データベース秘密鍵を含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Plugins.Mysql.Sessions.<SessionName>.Uri				名前付きセッションの接続文字列です。
				<SessionName> - アイテムキーで使用するセッション名を定義します。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略できます。
				ポートは省略できます（default=3306）。
				例: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User				名前付きセッションのユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Mysql.Timeout		1-30	global timeout	接続を確立する必要がある場合の最大待機時間（秒）です。

関連項目:

- 一般的な Zabbix agent 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

10 NVIDIA GPU プラグイン

概要

このセクションでは、NVIDIA GPU Zabbix エージェント 2 プラグインの設定ファイル（nvidia.conf）でサポートされるパラメータを一覧表示します。

NVIDIA GPU プラグインはロード可能なプラグインであり、NVIDIA GPU plugin repository で利用でき、詳細に説明されています。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.NVIDIA.System.Path				NVIDIA GPU プラグイン実行ファイルへのパス。 使用例: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`

Parameter	Mandatory	Range	Default	Description
Plugins.NVIDIA.Timeout		1-30	グローバルタイムアウト	接続試行時およびセッション内の後続の操作中に、サーバーからの応答を待機する最大時間（秒）を指定します。グローバルなアイテムタイプのタイムアウトまたは個別のアイテムのタイムアウトが、これより大きい場合はこの値を上書きします。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

11 Oracle プラグイン

概要

このセクションでは、Oracle Zabbix エージェント 2 プラグイン設定ファイル（oracle.conf）でサポートされているパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数を使用できます。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CallTimeout		1-30	global timeout	リクエストが完了するまでの最大待機時間（秒）です。
Plugins.Oracle.ConnectTimeout		1-30	global timeout	接続が確立されるまでの最大待機時間（秒）です。
Plugins.Oracle.CustomQueriesPath			`/usr/local/share/zabbix/oracle/custom.queries` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\Oracle` for Windows systems, where * is the drive name from the ProgramFiles environment variable	カスタムクエリファイルが格納されたディレクトリの完全パス名です。 例: `/etc/zabbix/oracle/sql`
Plugins.Oracle.CustomQueriesEnabled			false	設定すると、`oracle.custom.query` アイテムキーの実行を有効にします。無効な場合、カスタムクエリディレクトリパスからクエリは読み込まれません。
Plugins.Oracle.Default.Password				Oracle への接続に使用するデフォルトパスワードです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Oracle.Default.Service				Oracle への接続に使用するデフォルトのサービス名です（SID はサポートされません）。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.Default.Uri			tcp://localhost:1521	Oracle への接続に使用するデフォルト URI です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。 埋め込み認証情報を含めないでください（含まれていても無視されます）。 URI 形式に一致している必要があります。 サポートされるスキームは tcp のみです。スキームは省略できます。 ポートは省略できます（デフォルト =1521）。 **Zabbix 7.4.3** 以降では、接続文字列として TNS キーまたは TNS 値を指定することもできます。TNS 値は空白文字を含まない形式で構成する必要があります。 例: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS キー) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT (TNS 値)
Plugins.Oracle.Default.User				Oracle への接続に使用するデフォルトユーザー名です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Oracle.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）です。
Plugins.Oracle.ResolveTNS		true / false	false	このオプションは、Oracle サーバーに接続するために、メトリクスキー commonParams 内の接続文字列（ConnString）をどのように解釈するかを指定します。 false に設定した場合、デフォルトのスキームとポートが ConnString に追加されます（存在しない場合）。true に設定した場合、デフォルトのスキームとポートは省略され (ConnString で明示的に指定されている場合を除く)、ConnString はそのまま Oracle クライアントに渡されます。Oracle クライアントが tnsnames.ora ファイル内で ConnString を見つけた場合、見つかった接続記述を使用して Oracle サーバーに接続します。 このパラメータは Zabbix 7.4.3 以降でサポートされています。
Plugins.Oracle.Sessions.<SessionName>.Password				名前付きセッションのパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Oracle.Sessions.<SessionName>.Service				接続に使用する名前付きセッションのサービス名です（SID はサポートされません）。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Oracle.Sessions.<SessionName>.Uri				Oracle 用の名前付きセッション接続文字列です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 埋め込み認証情報を含めないでください（含まれていても無視されます）。 URI 形式に一致している必要があります。 サポートされるスキームは tcp のみです。スキームは省略できます。 ポートは省略できます（デフォルト =1521）。 **Zabbix 7.4.3** 以降では、接続文字列として TNS キーまたは TNS 値を指定することもできます。TNS 値は空白文字を含まない形式で構成する必要があります。 例: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS キー) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT (TNS 値)

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.Sessions.<SessionName>.User	no			名前付きセッションのユーザー名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

関連項目:

- 一般的な Zabbix agent 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

12 PostgreSQL プラグイン

概要

このセクションでは、PostgreSQL Zabbix エージェント 2 プラグインの設定ファイル（postgresql.conf）でサポートされるパラメータを一覧表示します。

PostgreSQL プラグインはロード可能なプラグインであり、PostgreSQL plugin repository で利用でき、詳細に説明されています。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix は、BOM なしの UTF-8 エンコーディングの設定ファイルのみをサポートします。
- ”#” で始まるコメントは、行頭でのみサポートされます。

オプション

Parameter	Description
-V --version	プラグインのバージョンおよびライセンス情報を表示します。
-h --help	ヘルプ情報を表示します（短縮形）。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.CacheMode	no		prepare	PostgreSQL 接続のキャッシュモード。 サポートされる値: *prepare* (デフォルト) - PostgreSQL サーバー上に準備済みステートメントを作成します。 *describe* - サーバー上にステートメントを作成せずに、匿名の準備済みステートメントを使用してステートメントを記述します。「describe」は、PgBouncer のような接続プーラーを実行している場合など、環境で準備済みステートメントが許可されていない場合に主に有用であることに注意してください。
Plugins.PostgreSQL.CallTimeout	no	1-30	global timeout	リクエストが完了するまでの最大待機時間（秒）。

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.CustomQueriesPath	いいえ		/usr/local/share/zabbix/include/postgresql.d for Unix systems *:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL for Windows systems, where * is the drive name from the ProgramFiles environment variable	カスタムクエリを含むディレクトリのフルパス名。
Plugins.PostgreSQL.CustomQueriesEnabled	いいえ		false	設定すると、postgresql.custom.query アイテムキーの実行を有効にします。無効な場合、カスタムクエリディレクトリパスからクエリは読み込まれません。
Plugins.PostgreSQL.Default.Database	いいえ			PostgreSQL への接続に使用するデフォルトのデータベース。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.Password	いいえ			PostgreSQL への接続に使用するデフォルトのパスワード。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.TLSCAFile	いいえ (Plugins.PostgreSQL.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合はい)			Zabbix エージェント 2 と監視対象データベース間の暗号化通信において、ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.TLSCertFile	いいえ (Plugins.PostgreSQL.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合はい)			Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する PostgreSQL 証明書または証明書チェーンを含むファイルのフルパス名。名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.Default.TLSConnect	いいえ			Zabbix エージェント 2 と監視対象データベース間の通信に使用する暗号化タイプ。名前付きセッションで値が指定されていない場合に使用されます。 サポートされる値: *required* - ID チェックなしで、転送モードとして TLS を使用して接続します。 *verify_ca* - TLS を使用して接続し、証明書を検証します。 *verify_full* - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID（CN）がその証明書と一致することを検証します。 暗号化タイプが未定義の場合は、暗号化されない接続を意味します。

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.TLSKeyFile	いいえ (Plugins.PostgreSQL.Default.TLSConnect が *verify_ca* または *verify_full* に設定されている場合はい)			Zabbix エージェント 2 と監視対象データベース間の暗号化通信に使用する PostgreSQL 秘密鍵を含むファイルのフルパス名。 名前付きセッションで値が指定されていない場合に使用されます。

Plugins.PostgreSQL.Default.Uri	いいえ			PostgreSQL への接続に使用するデフォルト URI。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。
				例: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	いいえ			PostgreSQL への接続に使用するデフォルトのユーザー名。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.PostgreSQL.KeepAlive	いいえ	60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間 (秒)。
Plugins.PostgreSQL.Sessions.<SessionName>.CachePrepMode	いいえ		prepare	PostgreSQL 接続のキャッシュモード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *prepare* (デフォルト) - PostgreSQL サーバー上に準備済みステートメントを作成します。 *describe* - サーバー上にステートメントを作成せずに、匿名の準備済みステートメントを使用してステートメントを記述します。「describe」は、PgBouncer のような接続プーラーを実行している場合など、環境で準備済みステートメントが許可されていない場合に主に有用であることに注意してください。
Plugins.PostgreSQL.Sessions.<SessionName>.Database	いいえ			セッション接続用のデータベース。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.Sessions.<SessionName>.Password	いいえ	<SessionName> 形式に一致する必要があります。		セッション接続用のパスワード。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	いいえ (Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect が *verify_ca* または *verify_full* に設定されている場合はい)			最上位 CA 証明書を含む、ピア証明書検証用ファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	いいえ (Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile が指定されている場合はいい			PostgreSQL 証明書または証明書チェーンを含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	いいえ			PostgreSQL 接続の暗号化タイプ。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: *required* - ID チェックなしで、転送モードとして TLS を使用して接続します。 *verify_ca* - TLS を使用して接続し、証明書を検証します。 *verify_full* - TLS を使用して接続し、証明書を検証し、DBHost で指定されたデータベース ID（CN）がその証明書と一致することを検証します。 暗号化タイプが未定義の場合は、暗号化されない接続を意味します。
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	TLSConnect が指定されている場合はいい			PostgreSQL 秘密鍵を含むファイルのフルパス名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Plugins.PostgreSQL.Sessions.<SessionName>.Uri	いいえ			名前付きセッションの接続文字列。
				<SessionName> - アイテムキーで使用するセッション名を定義します。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。
				例: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User	いいえ			名前付きセッションのユーザー名。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.PostgreSQL.System.Path	はい			PostgreSQL プラグイン実行可能ファイルへのパス。 使用例: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout	いいえ	1-30	global timeout	リクエスト実行タイムアウト（リクエストが完了するまで待機する時間（秒）で、これを超えるとリクエストは停止されます）。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

13 Redis プラグイン

概要

このセクションでは、Redis Zabbix エージェント 2 プラグインの設定ファイル（redis.conf）でサポートされるパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数を使用できます。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.Password				Redis への接続用のデフォルトパスワードです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Redis.Default.TLSConnect				クライアント（Zabbix エージェント 2）の TLS 検証要件です。 サポートされる値: `required` - 暗号化、未検証（テスト専用） `verify_ca` - 暗号化、サーバー証明書 `verify_full` - 暗号化、SAN でサーバー証明書とサーバー名を検証 空または未設定の値は、tls なしとして扱われます。
Plugins.Redis.Default.TLSCAFile				最上位 CA 証明書を含むファイルのデフォルトのフルパス名です。
Plugins.Redis.Default.TLSCertFile				エージェントの証明書を含むファイルのデフォルトのフルパス名です。
Plugins.Redis.Default.TLSKeyFile				エージェントの秘密鍵を含むファイルのデフォルトのフルパス名です。

Plugins.Redis.Default.Uri			tcp://localhost:6379	Redis への接続用のデフォルト URI です。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略できます。
				ポートは省略できます（デフォルト =6379）。
				例: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User			default	保護された Redis サーバーに送信するデフォルトユーザーです。アイテムキーまたは名前付きセッションで値が指定されていない場合に使用されます。
Plugins.Redis.KeepAlive		60-900	300	未使用のプラグイン接続が閉じられるまでの最大待機時間（秒）です。
Plugins.Redis.Sessions.<SessionName>.Password				名前付きセッションのパスワードです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Sessions.<SessionName>.TLSConnect				クライアント（Zabbix エージェント 2）の TLS 検証要件です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。 サポートされる値: `required` - 暗号化、未検証（テスト専用） `verify_ca` - 暗号化、サーバー証明書 `verify_full` - 暗号化、SAN でサーバー証明書とサーバー名を検証 空または未設定の値は、tls なしとして扱われます。
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	(yes, if TLSConnect is `verify_ca` or `verify_full`)			Redis サーバー証明書の検証用に、最上位 CA 証明書を含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Sessions.<SessionName>.TLSCertFile				クライアント認証用に、エージェントの証明書を含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile				クライアント認証用に、エージェントの秘密鍵を含むファイルのフルパス名です。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。

Plugins.Redis.Sessions.<SessionName>.Uri			localhost:6379	名前付きセッションの接続文字列です。
				<SessionName> - アイテムキーで使用するセッション名を定義します。
				
				埋め込み認証情報を含めないでください（含まれていても無視されます）。
				URI 形式に一致している必要があります。
				サポートされるスキーム: `tcp`, `unix`。スキームは省略できます。
				ポートは省略できます（デフォルト =6379）。
				例: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User			default	保護された Redis サーバーに送信するユーザーです。 **<SessionName>** - アイテムキーで使用するセッション名を定義します。
Plugins.Redis.Timeout		1-30	global timeout	リクエスト実行タイムアウトです（リクエストを停止する前に、その完了を待機する時間（秒））。

関連項目:

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

14 SMART プラグイン

概要

このセクションでは、SMART Zabbix エージェント 2 プラグイン設定ファイル（smart.conf）でサポートされるパラメータを一覧表示します。

以下の点に注意してください。

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値は環境変数をサポートしています。
- `smartctl` 実行ファイルへのパスは、そのディレクトリをシステムの PATH 環境変数に追加するか、`Plugins.Smart.Path` を設定することで指定する必要があります。これは Linux と Windows の両方に適用されます。
- Zabbix がサポートする設定ファイルは、BOM なしの UTF-8 エンコーディングのみです。
- "#" で始まるコメントは、行頭でのみサポートされます。

パラメータ

Parameter	Mandatory	Range	Default	Description
Plugins.Smart.Path			smartctl	smartctl 実行ファイルへのパス。
Plugins.Smart.Timeout		1-30	global timeout	リクエスト実行のタイムアウト（停止する前にリクエストの完了を待機する時間。単位は秒）。

以下も参照してください。

- 一般的な Zabbix エージェント 2 設定パラメータの説明: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- plugins の設定手順

8 Zabbix Java ゲートウェイ

Zabbix Java ゲートウェイ の起動に `startup.sh` および `shutdown.sh` スクリプトを使用する場合、必要な設定パラメータを `settings.sh` ファイルで指定できます。起動スクリプトと停止スクリプトは settings ファイルを読み込み、シェル変数（第 1 列に記載）を Java プロパティ（第 2 列に記載）へ変換します。

java を直接実行して Zabbix Java ゲートウェイを手動で起動する場合は、対応する Java プロパティをコマンドラインで指定します。

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	待ち受ける IP アドレス。
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	待ち受けるポート。
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	PID ファイル名。省略した場合、Zabbix Java Gateway はコンソールアプリケーションとして起動します。
PROPERTIES_FILE	zabbix.propertiesFile	no			プロパティファイル名。コマンドライン上に表示されないようにキーと値の形式で追加のプロパティを設定したり、既存のプロパティを上書きしたりするために使用できます。 例: "javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	起動するワーカースレッド数。
TIMEOUT	zabbix.timeout	no	1-30	3	ネットワーク操作を待機する時間（秒）。

Warning:
ポート 10052 は IANA 登録されていません。

9 Zabbix Web サービス

概要

Zabbix Web サービスは、外部の Web サービスとの通信に使用されるプロセスです。

このセクションでは、Zabbix Web サービスの設定ファイル（zabbix_web_service.conf）でサポートされるパラメータを一覧表示しています。

パラメータは追加情報なしで一覧表示されています。詳細を確認するには、パラメータをクリックしてください。

Parameter	Description
AllowedIP	Zabbix サーバーおよび Zabbix プロキシの、カンマ区切りの IP アドレス一覧です。CIDR 表記または DNS 名も指定できます。
DebugLevel	デバッグレベル。
IgnoreURLCertErrors	Web インターフェースの URL にアクセスする際の、TLS 証明書検証エラーの処理を指定します。
Include	設定ファイルには、個別のファイルまたはディレクトリ内のすべてのファイルを含めることができます。
ListenPort	このサービスは、このポートでサーバーからの接続を待ち受けます。
LogFile	ログファイル名。
LogFileSize	ログファイルの最大サイズ。
LogType	ログ出力の種類。
Timeout	ダッシュボードの PDF report の生成に費やす最大時間（秒）。
TLSAccept	受け入れる受信接続の種類。
TLSCAFile	ピア証明書の検証に使用する最上位 CA 証明書を含むファイルのフルパス名です。Zabbix コンポーネント間の暗号化通信で使用されます。
TLSCertFile	サービス証明書または証明書チェーンを含むファイルのフルパス名です。Zabbix コンポーネント間の暗号化通信で使用されます。
TLSKeyFile	サービス秘密鍵を含むファイルのフルパス名です。Zabbix コンポーネント間の暗号化通信で使用されます。

すべてのパラメータは、明示的に必須と記載されていない限り、必須ではありません。

注意:

- デフォルト値は、配布される設定ファイル内の値ではなく、プロセスのデフォルト値を反映しています。
- 値では環境変数をサポートしています。
- Zabbix がサポートする設定ファイルは、BOMなしの UTF-8 エンコーディングのみです。
- ”#” で始まるコメントは、行頭でのみサポートされます。

パラメータの詳細

AllowedIP

Zabbix サーバーおよび Zabbix プロキシの IP アドレス（CIDR 表記または DNS 名）をカンマ区切りで指定します。接続は、ここに指定されたホストからのみ許可されます。
IPv6 サポートが有効な場合、’127.0.0.1’、’::127.0.0.1’、’::ffff:127.0.0.1’ は同等に扱われ、’::/0’ はすべての IPv4 または IPv6 アドレスを許可します。’0.0.0.0/0’ はすべての IPv4 アドレスを許可します。

例：

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

必須：必須

DebugLevel

デバッグレベルを指定します。
0 - Zabbix プロセスの起動と停止に関する基本情報
1 - 重要な情報
2 - エラー情報
3 - 警告
4 - デバッグ用（多くの情報を生成します）
5 - 拡張デバッグ（さらに多くの情報を生成します）

デフォルト: 3
 範囲: 0-5

IgnoreURLCertErrors

フロントエンド URL へのアクセス時の TLS 証明書検証エラー処理の指定
0 - 証明書エラーを無視しません。
1 - 証明書エラーを無視します。

デフォルト: 0
 範囲: 0-1

Include

設定ファイルには、個別のファイル、またはディレクトリ内のすべてのファイルを含めることができます。
インストール時に、コンパイル時に変更されていない限り、Zabbix は /usr/local/etc に include ディレクトリを作成します。
パスは、*zabbix_web_service.conf* ファイルの場所からの相対パスにすることもできます。
 指定したディレクトリ内で関連するファイルのみを含めるために、パターンマッチング用のアスタリスクワイルドカード文字がサポートされています。
 制限事項については、特記事項を参照してください。

例:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

サービスがサーバーからの接続を待機するポート

デフォルト: 10053
 範囲: 1024-32767

LogFile

ログファイル名

例:

/tmp/zabbix_web_service.log

必須: LogType が *file* に設定されている場合は必須、それ以外の場合は必須ではありません

LogFileSize

ログファイルの最大サイズ（MB）
0 - 自動ログローテーションを無効にします。
 注: ログファイルのサイズ制限に達し、何らかの理由でファイルローテーションが失敗した場合、既存のログファイルは切り捨てられ、新たに開始されます。

デフォルト: 1
 範囲: 0-1024

LogType

ログ出力の種類:
file - LogFile パラメータで指定されたファイルにログを書き込みます。
system - syslog にログを書き込みます。
console - 標準出力にログを書き込みます。

デフォルト: `file`

タイムアウト

ダッシュボードの PDF レポート のフォーマットにかかる最大時間（秒）

デフォルト: 10
 範囲: 1-30

TLSAccept

受け入れる接続の種類:
unencrypted - 暗号化なしの接続を受け入れます (デフォルト)
cert - TLS と証明書を使用した接続を受け入れます

デフォルト: `unencrypted`

TLSCAFile

Zabbix コンポーネント間の暗号化通信に使用される、ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名

TLSCertFile

Zabbix コンポーネントとの暗号化通信に使用されるサービス証明書または証明書チェーンを含むファイルのフルパス名

TLSKeyFile

Zabbix コンポーネント間の暗号化通信に使用されるサービス秘密鍵を含むファイルのフルパス名

10 環境変数

概要

環境変数を使用すると、設定ファイルに値をハードコーディングすることなく、Zabbix コンポーネントを設定できます。これにより、Docker などの動的な環境で設定を簡単に管理でき、Docker では、実行時に変数を渡すことでさまざまな設定に適応させることができます。

最も単純なケースでは、Zabbix サーバーのDebugLevel 設定パラメータ値を環境変数に設定することで、起動時にサーバーの設定に使用できます。

```
#### Zabbix サーバー設定ファイル:
DebugLevel=${NewDebugLevel}

#### Zabbix サーバーの起動:
NewDebugLevel=5 /usr/sbin/zabbix_server
```

環境変数は、以下の Zabbix コンポーネントでサポートされています。

- Server
- Proxy
- エージェント (UNIX またはWindows)
- エージェント 2 (UNIX またはWindows)、プラグインを含む
- Web サービス
- Zabbix Sender (-c, --config オプション 使用時)

重要事項

- 設定パラメータが環境変数に設定されており、コンポーネント実行時にその環境変数が指定されていない場合、パラメータのデフォルト値が使用されます。
- ランタイムコマンドを使用する場合（例：エージェントのログレベルを上げる場合）、以前に使用した環境変数を指定する必要があります。これは、Zabbix コンポーネントが設定ファイルを使用してランタイムコマンドを実行するためです。環境変数が省略されている場合、設定パラメータのデフォルト値が使用されます。例を参照してください。
- `userparameter_reload` ランタイムコマンドは、環境変数の再読み込みをサポートしていません。再読み込み中は変数は無視され、通常の値を持つパラメータのみが再読み込みされます。
- 設定ファイルで使用されていたプロセスの現在の環境変数は、Zabbix コンポーネントの起動後にクリアされます。これにより、子プロセス（例：Zabbix によって実行されるリモートスクリプト）はこれらの変数にアクセスできなくなります。ただし、プロセス初期変数は引き続き取得できることに注意してください。(たとえば、`/proc/<PID>/environ` ファイル経由)

構文

環境変数は、`${英数字/アンダースコア}` という構文を使用する必要があります。

変数名には、英字 (a-z、A-Z)、アンダースコア (_)、数字 (0-9) のみを使用できます。また、数字で始まってはなりません。

必須の構文に一致しない変数、または通常の値と組み合わせられた変数は通常の値として扱われ、エラーが発生する可能性があります。

正しい変数構文:

```
DebugLevel=${NewDebugLevel}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

誤った変数構文:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

Note:
Windows では、環境変数名の大文字と小文字は区別されません。

例

以下の例では、Zabbix コンポーネントでの環境変数の設定および使用方法を示しています。

例 1: Zabbix エージェントの設定とテスト

1. エージェント設定ファイルで環境変数を設定します。

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. 設定ファイルをテストします。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. 環境変数を有効にしてエージェントを起動します。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

ランタイムコマンドを使用する場合（例：エージェントのログレベルを上げる場合）、以前に使用した環境変数を再度指定する必要があります。

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

これは、エージェントがランタイムコマンドを実行するために設定ファイルを使用するためです。環境変数が省略されている場合、設定パラメータのデフォルト値が使用されます。

あるいは、エージェント設定ファイルで環境変数を設定した後、プロセスでそれらを有効に出来るようにすることもできます。(例：export コマンドを使用）これにより、変数の欠落や誤った設定による予期しない動作のリスクを軽減できます。

```
export ZBX_HOSTNAME=" 新しい Zabbix エージェント"
export ServerActive=127.0.0.1
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf --test-config
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
/usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf -R log_level_increase
```

例 2: コンテナ用の Zabbix エージェントの設定

Zabbix コンポーネント（例: Zabbix エージェント）用のカスタムイメージを独自に作成・設定する場合には、環境変数を使用して設定パラメータを定義し、その変数を有効にしてコンテナを起動できます。

1. イメージを準備する際に、エージェント設定ファイルで環境変数を設定します。

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. コンテナイメージをビルドしたら、環境変数を指定してエージェントコンテナ（例：Docker）を起動します。

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. ランタイムコマンドを使用する場合（例：エージェントのログレベルを上げる場合）、コンテナシェルにアクセスし、以下のランタイムコマンドを実行します。

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

userparameter_reload ランタイムコマンドは、環境変数の再読み込みをサポートしていません。再読み込み中は変数は無視され、通常の値を持つパラメータのみが再読み込みされます。

11 インクルード

概要

Include パラメータを使用すると、追加のファイルまたはディレクトリをサーバー、プロキシ、エージェント、および Web サービスの設定に含めることができます。

インクルード時の注意

ファイルを含めるために Include パラメーターを使用する場合、ファイルは読み取り可能である必要があります。

ディレクトリを含めるために Include パラメーターが使用される場合:

- ディレクトリ内のすべてのファイルが読み取り可能である必要があります。
- インクルードの順番を想定する必要はありません (たとえばファイルはアルファベット順で読み込まれません)。したがって複数の" インクルード" ファイルで 1 つのパラメーターを定義しないでください (たとえば特定の設定で一般設定をオーバーライドするため等)。
- ディレクトリ内のすべてのファイルが設定に含まれます。
- 一部のテキストエディタによって自動的に作成されるファイルバックアップコピーに注意してください。たとえば"include/my_specific.conf" ファイルを編集すると、バックアップコピー"include/my_specific_conf.BAK" が生成される場合、両方のファイルがインクルード対象に含まれます。"include/my_specific.conf.BAK" を"Include" ディレクトリから移動してください。Linux では"Include" ディレクトリの内容を"ls -al" コマンドで不要なファイルをチェックできます。

パターンを使用してファイルをインクルードするために Include パラメーターが使用される場合:

- パターンに一致するすべてのファイルは読み取り可能である必要があります。
- インクルードの順番を想定する必要はありません (たとえばファイルはアルファベット順で読み込まれません)。したがって複数の" インクルード" ファイルで 1 つのパラメーターを定義しないでください (たとえば特定の設定で一般設定をオーバーライドするため等)。

3 プロトコル

サイドバーから、参照したい項目を選択してください。

1 サーバープロキシデータ交換プロトコル

概要

サーバー - プロキシ間のデータ交換は JSON 形式に基づいています。

要求メッセージと応答メッセージはヘッダーとデータの長さに則って始まる必要があります。

パッシブプロキシ

設定リクエスト

サーバーは最初に空の proxy config リクエストを送信します。このリクエストは、ProxyConfigFrequency （サーバー設定パラメータ）秒ごとに送信されます。

プロキシは、現在のプロキシバージョン、セッショントークン、および設定リビジョンを返します。サーバーは、更新が必要な設定データを返します。

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	プロキシバージョン (<major>.<minor>.<build>)。
session	*string*	プロキシ設定セッショントークン。
config_revision	*number*	プロキシ設定リビジョン。

server→proxy:

name	value type	description
full_sync	*number*	1 - 完全な設定データが送信される場合。それ以外の場合は存在しません
data	*array*	テーブルデータのオブジェクト。設定が変更されていない場合は存在しま
<table>	*object*	<table> データを含む 1 つ以上のオブジェクト（任意、変更内容に応じ
fields	*array*	フィールド名の配列。
-	*string*	フィールド名。
data	*array*	行の配列。
-	*array*	列の配列。
-	*string,number*	データベーススキーマ内の列タイプに応じた型の列値。
macro.secrets	*object*	シークレットマクロ情報。vault マクロに変更がない場合は存在しません
config_revision	*number*	設定キャッシュリビジョン - 設定データとともに送信されます（任意）。
del_hostids	*array*	削除された hostid の配列（任意）。
-	*number*	ホスト識別子。
del_macro_hostids	*array*	すべてのマクロが削除された hostid の配列（任意）。
-	*number*	ホスト識別子。
proxy→server:		
response	*string*	リクエスト成功情報（'success' または'failed'）。
version	*string*	プロキシバージョン (<major>.<minor>.<build>)。

例:

server→proxy:

```
{
  "request":"proxy config"
}
```

proxy→server:

```
{
  "version": "7.4.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
```

```
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
```

```
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
```

```
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


proxy→server:


```
{
  "response": "success",
  "version": "7.4.0"
}
```


データ要求

`proxy data` リクエストは、ホストインターフェースの可用性、履歴、ディスカバリ、自動登録データをプロキシから取得するために使用されます。このリクエストは `ProxyDataFrequency`（サーバー設定パラメータ）秒ごとに送信されます。

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	データセッショントークン。
interface avail-abil-ity	*array*	*(optional)* インターフェース可用性データオブジェクトの配列。
interfaceid	*number*	インターフェース識別子。
available	*number*	インターフェースの可用性: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - 不明 **1**, *INTERFACE_AVAILABLE_TRUE* - 利用可能 **2**, *INTERFACE_AVAILABLE_FALSE* - 利用不可
error	*string*	インターフェースのエラーメッセージ、または空文字列。
history data	*array*	*(optional)* 履歴データオブジェクトの配列。
itemid	*number*	アイテム識別子。
clock	*number*	アイテム値のタイムスタンプ（秒）。
ns	*number*	アイテム値のタイムスタンプ（ナノ秒）。
value	*string*	*(optional)* アイテム値。
id	*number*	値識別子（昇順カウンター、1 つのデータセッション内で一意）。
timestamp	*number*	*(optional)* ログタイプのアイテムのタイムスタンプ。
source	*string*	*(optional)* イベントログアイテムの source 値。
severity	*number*	*(optional)* イベントログアイテムの severity 値。
eventid	*number*	*(optional)* イベントログアイテムの eventid 値。
state	*string*	*(optional)* アイテムの状態: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* ログタイプのアイテムの最後のログサイズ。
mtime	*number*	*(optional)* ログタイプのアイテムの更新時刻。
discovery data	*array*	*(optional)* ディスカバリデータオブジェクトの配列。
clock	*number*	ディスカバリデータのタイムスタンプ。
druleid	*number*	ディスカバリルール識別子。
dcheckid	*number*	ディスカバリチェック識別子。ディスカバリルールデータの場合は null。

name	value type	description
type	*number*	ディスカバリチェックタイプ: **-1** ディスカバリルールデータ **0**, *SVC_SSH* - SSH サービスチェック **1**, *SVC_LDAP* - LDAP サービスチェック **2**, *SVC_SMTP* - SMTP サービスチェック **3**, *SVC_FTP* - FTP サービスチェック **4**, *SVC_HTTP* - HTTP サービスチェック **5**, *SVC_POP* - POP サービスチェック **6**, *SVC_NNTP* - NNTP サービスチェック **7**, *SVC_IMAP* - IMAP サービスチェック **8**, *SVC_TCP* - TCP ポート可用性チェック **9**, *SVC_AGENT* - Zabbix エージェント **10**, *SVC_SNMPv1* - SNMPv1 エージェント **11**, *SVC_SNMPv2* - SNMPv2 エージェント **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 エージェント **14**, *SVC_HTTPS* - HTTPS サービスチェック **15**, *SVC_TELNET* - Telnet 可用性チェック
ip	*string*	ホスト IP アドレス。
dns	*string*	ホスト DNS 名。
port	*number*	*(optional)* サービスポート番号。
key_	*string*	*(optional)* タイプ **9** *SVC_AGENT* のディスカバリチェック用アイテムキー
value	*string*	*(optional)* サービスから受信した値。ほとんどのサービスでは空の場合があります。
status	*number*	*(optional)* サービスステータス: **0**, *DOBJECT_STATUS_UP* - サービス稼働中 **1**, *DOBJECT_STATUS_DOWN* - サービス停止中
auto reg- is- tra- tion	*array*	*(optional)* 自動登録データオブジェクトの配列。
clock	*number*	自動登録データのタイムスタンプ。
host	*string*	ホスト名。
ip	*string*	*(optional)* ホスト IP アドレス。
dns	*string*	*(optional)* IP アドレスから解決された DNS 名。
port	*string*	*(optional)* ホストポート。
host_metadata	*string*	*(optional)* エージェントによって送信されたホストメタデータ（HostMetadata または HostMetadataItem エージェント設定パラメータに基づく）。
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスクタイプ: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - リモートコマンド結果
status	*number*	リモートコマンド実行ステータス: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - リモートコマンドは正常に完了しました **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - リモートコマンドは失敗しました
error	*string*	*(optional)* エラーメッセージ。
parent_taskid	*number*	親タスク ID。
more	*number*	*(optional)* 1 - 送信する履歴データがさらにあります。
clock	*number*	*(optional)* データ転送タイムスタンプ（秒）。
ns	*number*	*(optional)* データ転送タイムスタンプ（ナノ秒）。
version	*string*	プロキシバージョン（<major>.<minor>.<build>）。
server→proxy:		
response	*string*	リクエスト成功情報（'success' または'failed'）。
tasks	*array*	*(optional)* タスクの配列。

name	value type	description
type	*number*	タスクタイプ: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - リモートコマンド
clock	*number*	タスク作成時刻。
ttl	*number*	タスクの有効期限が切れるまでの秒数。
commandtype	*number*	リモートコマンドタイプ: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - カスタムスクリプトを使用 **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI を使用 **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH を使用 **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet を使用 **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - グローバルスクリプトを使用（現在は機能的にカスタムスクリプトと同等）
command	*string*	実行するリモートコマンド。
execute_on	*number*	カスタムスクリプトの実行対象: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - エージェント上でスクリプトを実行 **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - サーバー上でスクリプトを実行 **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - プロキシ上でスクリプトを実行
port	*number*	*(optional)* Telnet および SSH コマンド用のポート。
authtype	*number*	*(optional)* SSH コマンド用の認証タイプ。
username	*string*	*(optional)* Telnet および SSH コマンド用のユーザー名。
password	*string*	*(optional)* Telnet および SSH コマンド用のパスワード。
publickey	*string*	*(optional)* SSH コマンド用の公開鍵。
privatekey	*string*	*(optional)* SSH コマンド用の秘密鍵。
parent_taskid	*number*	親タスク ID。
hostid	*number*	対象ホスト ID。

例:

server→proxy:

```
{
  "request": "proxy data"
}
```

proxy→server:

```
{
    "session": "12345678901234567890123456789012"
    "interface availability": [
        {
            "interfaceid": 1,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 2,
            "available": 2,
            "error": "Get value from agent failed: cannot connect to [[127.0.0.1]:10049]: [111] Connection
    },
        {
            "interfaceid": 3,
            "available": 1,
            "error": ""
    },
        {
            "interfaceid": 4,
            "available": 1,
            "error": ""
    }
    ],
```

```
    "history data":[
        {
            "itemid":"12345",
            "clock":1478609647,
            "ns":332510044,
            "value":"52956612",
            "id": 1
        },
        {
            "itemid":"12346",
            "clock":1478609647,
            "ns":330690279,
            "state":1,
            "value":"Cannot find information for this network interface in /proc/net/dev.",
            "id": 2
        }
    ],
    "discovery data":[
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":3,
            "type":12,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        },
        {
            "clock":1478608764,
            "drule":2,
            "dcheck":null,
            "type":-1,
            "ip":"10.3.0.10",
            "dns":"vdebian",
            "status":1
        }
    ],
    "auto registration":[
        {
            "clock":1478608371,
            "host":"Logger1",
            "ip":"10.3.0.1",
            "dns":"localhost",
            "port":"10050"
        },
        {
            "clock":1478608381,
            "host":"Logger2",
            "ip":"10.3.0.2",
            "dns":"localhost",
            "port":"10050"
        }
    ],
    "tasks":[
        {
            "type": 0,
            "status": 0,
            "parent_taskid": 10
        },
        {
            "type": 0,
            "status": 1,
```

```
            "error": "No permissions to execute task.",
            "parent_taskid": 20
        }
    ],
    "version":"7.4.0"
}
```

server→proxy:

```
{
  "response": "success",
  "tasks":[
      {
         "type": 1,
         "clock": 1478608371,
         "ttl": 600,
         "commandtype": 2,
         "command": "restart_service1.sh",
         "execute_on": 2,
         "port": 80,
         "authtype": 0,
         "username": "userA",
         "password": "password1",
         "publickey": "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqGKukO1De7zhZj6+H0qtjTkVxwTCpvKe",
         "privatekey": "lsuusFncCzWBQ7RKNUSesmQRMSGkVb1/3j+skZ6UtW+5u09lHNsj6tQ5QCqGKukO1De7zhd",
         "parent_taskid": 10,
         "hostid": 10070
      },
      {
         "type": 1,
         "clock": 1478608381,
         "ttl": 600,
         "commandtype": 1,
         "command": "restart_service2.sh",
         "execute_on": 0,
         "authtype": 0,
         "username": "",
         "password": "",
         "publickey": "",
         "privatekey": "",
         "parent_taskid": 20,
         "hostid": 10084
      }
  ]
}
```

タスク要求

`proxy tasks` のやり取りでは、サーバーとパッシブプロキシの間でリモートコマンドタスクの配信と確認応答を処理します。サーバーがタスク結果を取得するためにプロキシをポーリングすると、`ProxyDataFrequency` で定義された間隔で空の `proxy tasks` リクエストを送信します。プロキシは、現在のバージョンと保留中のタスク結果（タイムスタンプを含む）を返します。その後、サーバーは `response` を返して受信を確認し、必要に応じてプロキシが実行する新しいタスクを含めることがあります。

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	プロキシのバージョン (<major>.<minor>.<build>)。
clock	*number*	*(optional)* データ転送のタイムスタンプ（秒）。
ns	*number*	*(optional)* データ転送のタイムスタンプ（ナノ秒）。
server→proxy:		
response	*string*	リクエスト成功情報（'success' または'failed'）。

例:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

アクティブプロキシ

設定リクエスト

`proxy config` リクエストは、プロキシ設定データを取得するためにアクティブプロキシから送信されます。このリクエストは、`ProxyConfigFrequency`（プロキシ設定パラメータ）秒ごとに送信されます。

name				value type	description
proxy→server:					
request				*string*	'proxy config'
host				*string *	プロキシ名。
version				*string*	プロキシのバージョン (<major>.<minor>.<build>)。
session				*string*	プロキシ設定セッショントークン。
config_revision				*number*	プロキシ設定リビジョン。
hostmap_revision				*number*	現在のホストとプロキシのマッピングリビジョン。
server→proxy:					
full_sync				*number*	1 - 完全な設定データが送信される場合、それ以外は存在しません（オプ
data				*array*	テーブルデータのオブジェクト。設定が変更されていない場合は存在しま
	<table>			*object*	<table> データを持つ 1 つ以上のオブジェクト（オプション、変更内容
		fields		*array*	フィールド名の配列。
			-	*string*	フィールド名。
		data		*array*	行の配列。
			-	*array*	列の配列。
			-	*string,number*	データベーススキーマ内の列タイプに応じた型の列値。
macro.secrets				*object*	シークレットマクロ情報。vault マクロに変更がない場合は存在しません
proxy_group				*string*	プロキシが属するプロキシグループの名前。
config_revision				*number*	設定キャッシュリビジョン - 設定データとともに送信されます（オプシ
del_hostids				*array*	削除された hostid の配列（オプション）。
	-			*number*	ホスト識別子。
del_macro_hostids				*array*	すべてのマクロが削除された hostid の配列（オプション）。
	-			*number*	ホスト識別子。

例:

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"7.4.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:


```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
```

```
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
```

```
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


データリクエスト

proxy data リクエストは、ホストインターフェースの可用性、履歴、ディスカバリ、自動登録データを提供するために、プロキシから送信されます。
このリクエストは、DataSenderFrequency（プロキシ設定パラメータ）秒ごとに送信されます。
アクティブプロキシは、リモートコマンドタスクのために、引き続き 1 秒ごとに Zabbix サーバーをポーリングすることに注意してください（空の proxy data リクエストを使用）。

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	プロキシ名。
session	*string*	データセッショントークン。
interface avail-abil-ity	*array*	*(optional)* インターフェース可用性データオブジェクトの配列。
interfaceid	*number*	インターフェース識別子。
available	*number*	インターフェースの可用性: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - 不明 **1**, *INTERFACE_AVAILABLE_TRUE* - 利用可能 **2**, *INTERFACE_AVAILABLE_FALSE* - 利用不可
error	*string*	インターフェースエラーメッセージ、または空文字列。
history data	*array*	*(optional)* 履歴データオブジェクトの配列。
itemid	*number*	アイテム識別子。
clock	*number*	アイテム値のタイムスタンプ（秒）。

name	value type	description
ns	*number*	アイテム値のタイムスタンプ（ナノ秒）。
value	*string*	*(optional)* アイテム値。
id	*number*	値識別子（昇順カウンター、1 つのデータセッション内で一意）。
timestamp	*number*	*(optional)* ログタイプのアイテムのタイムスタンプ。
source	*string*	*(optional)* イベントログアイテムの source 値。
severity	*number*	*(optional)* イベントログアイテムの severity 値。
eventid	*number*	*(optional)* イベントログアイテムの eventid 値。
state	*string*	*(optional)* アイテム状態: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* ログタイプのアイテムの最後のログサイズ。
mtime	*number*	*(optional)* ログタイプのアイテムの更新時刻。
discovery	*array*	*(optional)* ディスカバリデータオブジェクトの配列。
data		
clock	*number*	ディスカバリデータのタイムスタンプ。
druleid	*number*	ディスカバリルール識別子。
dcheckid	*number*	ディスカバリチェック識別子。ディスカバリルールデータの場合は null。
type	*number*	ディスカバリチェックタイプ: **-1** ディスカバリルールデータ **0**, *SVC_SSH* - SSH サービスチェック **1**, *SVC_LDAP* - LDAP サービスチェック **2**, *SVC_SMTP* - SMTP サービスチェック **3**, *SVC_FTP* - FTP サービスチェック **4**, *SVC_HTTP* - HTTP サービスチェック **5**, *SVC_POP* - POP サービスチェック **6**, *SVC_NNTP* - NNTP サービスチェック **7**, *SVC_IMAP* - IMAP サービスチェック **8**, *SVC_TCP* - TCP ポート可用性チェック **9**, *SVC_AGENT* - Zabbix エージェント **10**, *SVC_SNMPv1* - SNMPv1 エージェント **11**, *SVC_SNMPv2* - SNMPv2 エージェント **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 エージェント **14**, *SVC_HTTPS* - HTTPS サービスチェック **15**, *SVC_TELNET* - Telnet 可用性チェック
ip	*string*	ホスト IP アドレス。
dns	*string*	ホスト DNS 名。
port	*number*	*(optional)* サービスポート番号。
key_	*string*	*(optional)* タイプ **9** *SVC_AGENT* のディスカバリチェック用アイテムキー
value	*string*	*(optional)* サービスから受信した値。ほとんどのサービスでは空の場合があります。
status	*number*	*(optional)* サービスステータス: **0**, *DOBJECT_STATUS_UP* - サービス UP **1**, *DOBJECT_STATUS_DOWN* - サービス DOWN
autoregistration	*array*	*(optional)* 自動登録データオブジェクトの配列。
clock	*number*	自動登録データのタイムスタンプ。
host	*string*	ホスト名。
ip	*string*	*(optional)* ホスト IP アドレス。
dns	*string*	*(optional)* IP アドレスから解決された DNS 名。
port	*string*	*(optional)* ホストポート。
host_metadata	*string*	*(optional)* エージェントによって送信されたホストメタデータ（HostMetadata または HostMetadataItem エージェント設定パラメータに基づく）。
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスクタイプ: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - リモートコマンド結果

name	value type	description
status	*number*	リモートコマンド実行ステータス: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - リモートコマンドは正常に完了 **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - リモートコマンドは失敗
error	*string*	*(optional)* エラーメッセージ。
parent_taskid	*number*	親タスク ID。
more	*number*	*(optional)* 1 - 送信する履歴データがさらにある
clock	*number*	*(optional)* データ転送タイムスタンプ（秒）。
ns	*number*	*(optional)* データ転送タイムスタンプ（ナノ秒）。
version	*string*	プロキシバージョン（<major>.<minor>.<build>）。
server→proxy:		
response	*string*	リクエスト成功情報（'success' または'failed'）。
upload	*string*	履歴データ（履歴、自動登録、ホスト可用性、ネットワークディスカバリ）のアップロード制御。 指定可能な値: **enabled** - 通常動作 **disabled** - サーバーがデータを受け付けていない（内部キャッシュが上限を超えている可能性があります）
tasks	*array*	*(optional)* タスクの配列。
type	*number*	タスクタイプ: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - リモートコマンド
clock	*number*	タスク作成時刻。
ttl	*number*	タスクの有効期限が切れるまでの秒数。
commandtype	*number*	リモートコマンドタイプ: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - カスタムスクリプトを使用 **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI を使用 **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH を使用 **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet を使用 **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - グローバルスクリプトを使用（現在は機能的にカスタムスクリプトと同等）
command	*string*	実行するリモートコマンド。
execute_on	*number*	カスタムスクリプトの実行対象: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - エージェント上でスクリプトを実行 **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - サーバー上でスクリプトを実行 **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - プロキシ上でスクリプトを実行
port	*number*	*(optional)* Telnet および SSH コマンド用のポート。
authtype	*number*	*(optional)* SSH コマンド用の認証タイプ。
username	*string*	*(optional)* Telnet および SSH コマンド用のユーザー名。
password	*string*	*(optional)* Telnet および SSH コマンド用のパスワード。
publickey	*string*	*(optional)* SSH コマンド用の公開鍵。
privatekey	*string*	*(optional)* SSH コマンド用の秘密鍵。
parent_taskid	*number*	親タスク ID。
hostid	*number*	対象ホスト ID。

例:

proxy→server:

```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
```

```
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
"ip": "127.0.0.1",
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "7.4.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
```

```
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
}]
}
```

2 Zabbix エージェント/エージェント **2** のプロトコル

Zabbix エージェントおよび Zabbix エージェント 2 のプロトコルの詳細については、パッシブおよびアクティブエージェントチェックページを参照してください。

4 Zabbix エージェント **2** プラグインプロトコル

Zabbix エージェント 2 プロトコルはコード、サイズ、データモデルに基づいています。

コード

Type	Size	Comments
Byte	4	ペイロードのタイプ。現在は JSON のみがサポートされています。

サイズ

Type	Size	Comments
Byte	4	現在のペイロードのサイズ（バイト単位）。

ペイロードデータ

Type	Size	Comments
Byte	*Size* フィールドで定義	JSON 形式のデータ。

ペイロードデータの定義

共通データ

これらのパラメータは、すべてのリクエスト/レスポンスに含まれます。

Name	Type	Comments
id	uint32	リクエストの場合 - リクエストとレスポンスを関連付けるために使用される、増加していく識別子です。リクエストの方向ごと（つまり、エージェントからプラグイン、またはプラグインからエージェント）に一意です。 レスポンスの場合 - 対応するリクエストの ID です。
type	uint32	リクエストのタイプです。

ログリクエスト

プラグインがログメッセージをエージェントのログファイルに書き込むために送信するリクエストです。

direction	plugin → agent
response	no

ログリクエスト固有のパラメータ:

Name	Type	Comments
severity	uint32	メッセージの重要度（ログレベル）。
message	string	ログに記録するメッセージ。

例:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

登録リクエスト

プラグインを登録するために、提供されるメトリクスを取得する目的で、エージェントの起動フェーズ中にエージェントから送信されるリクエストです。

direction	エージェント → plugin
response	yes

登録リクエスト固有のパラメータ:

Name	Type	Comments
version	string	プロトコルバージョン <major>.<minor>

例:

```
{"id":1,"type":2,"version":"1.0"}
```

Register 応答

register リクエストに対するプラグインの応答です。

direction	plugin → エージェント
response	n/a

register 応答固有のパラメータ:

Name	Type	Comments
name	string	プラグイン名。
metrics	array of strings (optional)	プラグインで使用される、説明付きのメトリクスです。RegisterMetrics() を返します。エラーが返された場合は存在しません。
interfaces	uint32 (optional)	プラグインがサポートするインターフェースのビットマスクです。エラーが返された場合は存在しません。
error	string (optional)	プラグインを起動できない場合に返されるエラーメッセージです。メトリクスが返された場合は存在しません。

例:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

または

```
{"id":2,"type":3,"error":"error message"}
```

Start リクエスト

Runner インターフェースの Start 関数を実行するためのリクエストです。

direction	エージェント → plugin
response	no

このリクエストには固有のパラメータはなく、共通データのパラメータのみが含まれます。

例:

```
{"id":3,"type":4}
```

終了リクエスト

プラグインをシャットダウンするためにエージェントから送信されるリクエストです。

direction	エージェント → plugin
response	no

このリクエストには固有のパラメータはなく、共通データ パラメータのみが含まれます。

例:

```
{"id":3,"type":5}
```

エクスポートリクエスト

Exporter インターフェースの Export 関数を実行するためのリクエスト。

direction	エージェント → plugin
response	no

エクスポートリクエスト固有のパラメータ:

Name	Type	Comments
key	string	プラグインキー。
parameters	array of strings (optional)	Export 関数のパラメータ。

例:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

エクスポート応答

Exporter インターフェースの Export 関数からの応答。

direction	プラグイン → エージェント
response	該当なし

エクスポート応答固有のパラメータ:

Name	Type	Comments
value	string (optional)	Export 関数からの応答値。エラーが返された場合は存在しません。
error	string (optional)	Export 関数が正常に実行されなかった場合のエラーメッセージ。value が返された場合は存在しません。

例:

```
{"id":5,"type":7,"value":"response"}
```

または

```
{"id":5,"type":7,"error":"error message"}
```

リクエストの設定

Configurator インターフェースの *Configure* 機能を実行するためのリクエストです。

direction	エージェント → plugin
response	n/a

Configure リクエスト固有のパラメータ:

Name	Type	Comments
global_options	JSON object	グローバルなエージェント設定オプションを含む JSON オブジェクト。
private_options	JSON object (optional)	指定されている場合、プライベートなプラグイン設定オプションを含む JSON オブジェクト。

例*:*

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

リクエストの検証

Configurator インターフェースの *Validate* 関数を実行するためのリクエスト。

direction	エージェント → plugin
response	yes

Validate リクエスト固有のパラメータ:

Name	Type	Comments
private_options	JSON object (optional)	提供されている場合、プライベートなプラグイン設定オプションを含む JSON オブジェクト。

例*:*

```
{"id":7,"type":9,"private_options":{...}}
```

レスポンスの検証

Configurator インターフェースの *Validate* 関数からのレスポンス。

direction	plugin → エージェント
response	n/a

Validate レスポンス固有のパラメータ:

Name	Type	Comments
error	string (optional)	*Validate* 関数が正常に実行されなかった場合に返されるエラーメッセージです。正常に実行された場合は存在しません。

例*:*

```
{"id":8,"type":10}
```

または

```
{"id":8,"type":10,"error":"error message"}
```

5 Zabbix sender プロトコル

概要

Zabbix サーバーと Zabbix プロキシは、Zabbix sender からデータを受信するのに JSON ベースの通信プロトコルを使用します。トラッパーアイテム、またはトラッピングを有効にしたHTTP エージェントアイテムの助けを借りてデータを受信できます。

リクエストメッセージとレスポンスメッセージは、ヘッダーとデータの長さで始まる必要があります。

Zabbix sender リクエスト

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Zabbix server レスポンス

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

タイムスタンプ付きの Zabbix sender リクエスト

また、Zabbix 送信者はタイムスタンプとナノ秒を含むリクエストを送信できます。

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710794,
            "ns": 592397170
        },
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710795,
            "ns": 192399456
        }
    ],
    "clock": 1516712029,
    "ns": 873386094
}
```

Zabbix server レスポンス

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

6 ヘッダー

概要

ヘッダーは、Zabbix コンポーネント間のすべての要求および応答メッセージに存在します。ヘッダーの情報からメッセージの長さ、圧縮されているかどうか、大きなパケットであるかどうかを判断できます。

Zabbix 通信プロトコルには、接続ごとに 1GB のパケットサイズ制限があります。受信パケットのデータ長、圧縮前のデータ長ともに 1GB の制限が適用されます。

構成を Zabbix プロキシに送信する場合、大きな構成を同期できるように、パケットサイズの制限が 4 GB に増加します。圧縮前のデータ長が 4GB を超えると、Zabbix サーバーは自動的に大きなパケット形式 (0x04 フラグ) を使用して起動し、パケットサイズの制限を 16GB に増やします。

任意のデータの送信に大きなパケット形式を使用できますが、現在、1GB を超えるパケットを処理できるのは Zabbix プロキシ構成シンカーのみであることに注意してください。

構造

ヘッダーは 4 つのフィールドで構成されます。ヘッダー内のすべての数値は、リトルエンディアンとしてフォーマットされています。

フィールド	サイズ	サイズ (ラージパケット)	説明
<PROTOCOL>	4	4	"ZBXD" または 5A 42 58 44
<FLAGS>	1	1	Protocol フラグ: 0x01 - Zabbix 通信プロトコル 0x02 - 圧縮 0x04 - ラージパケット
<DATALEN>	4	8	データ長
<RESERVED>	4	8	圧縮を使用する場合 (0x02 フラグ) - 非圧縮データの長さ 圧縮を使用しない場合 - 00 00 00 00

例

以下は、Zabbix に送信する必要があるパケットを取得して正しく解釈されるようにするために、送信するデータに Zabbix プロトコルヘッダーを追加する方法を示すコードスニペットです。これらのコードスニペットは、データが 1GB を超えないことを前提としているため、ラージパケットフォーマットは使用されません。

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

または

```
def zbx_create_header(plain_data_size, compressed_data_size=None):
    protocol = b"ZBXD"
    flags = 0x01
    if compressed_data_size is None:
        datalen = plain_data_size
        reserved = 0
    else:
        flags |= 0x02
        datalen = compressed_data_size
        reserved = plain_data_size
    return protocol + struct.pack("<BII", flags, datalen, reserved)

packet = zbx_create_header(len(data)) + data
```

Perl

```
my $packet = "ZBXD\1" . pack("(II)<", length($data), 0) . $data;
```

または

```
sub zbx_create_header($;$)
{
    my $plain_data_size = shift;
    my $compressed_data_size = shift;

    my $protocol = "ZBXD";
    my $flags = 0x01;
```

```
    my $datalen;
    my $reserved;

    if (!defined($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }

    return $protocol . chr($flags) . pack("(II)<", $datalen, $reserved);
}

my $packet = zbx_create_header(length($data)) . $data;
```

PHP

```
$packet = "ZBXD\1" . pack("VV", strlen($data), 0) . $data;
```

または

```
function zbx_create_header($plain_data_size, $compressed_data_size = null)
{
    $protocol = "ZBXD";
    $flags = 0x01;
    if (is_null($compressed_data_size))
    {
        $datalen = $plain_data_size;
        $reserved = 0;
    }
    else
    {
        $flags |= 0x02;
        $datalen = $compressed_data_size;
        $reserved = $plain_data_size;
    }
    return $protocol . chr($flags) . pack("VV", $datalen, $reserved);
}

$packet = zbx_create_header(strlen($data)) . $data;
```

Bash

```
datalen=$(printf "%08x" ${#data})
datalen="\\x${datalen:6:2}\\x${datalen:4:2}\\x${datalen:2:2}\\x${datalen:0:2}"
printf "ZBXD\1${datalen}\0\0\0\0%s" "$data"
```

7 改行区切り JSON エクスポートプロトコル

このセクションでは、以下で使用される改行区切り JSON 形式のエクスポートプロトコルの詳細を説明します。

- ファイルへのデータエクスポート
- 外部システムへのストリーミング

エクスポートできるものは次のとおりです。

- トリガーイベント
- アイテムの値
- トレンド（ファイルへのエクスポートのみ）

すべてのファイルの拡張子は.ndjson です。エクスポートファイルの各行は JSON オブジェクトです。

トリガーイベント

障害イベントについては、以下の情報がエクスポートされます。

Field			Type	Description
clock			number	障害が検出された時点までの Epoch からの経過秒数（整数部）。
ns			number	正確な障害検出時刻を取得するために clock に加算されるナノ秒数。
value			number	1（常に）。
eventid			number	障害イベント ID。
name			string	障害イベント名。
severity			number	障害イベントの深刻度（0 - 未分類、1 - 情報、2 - 警告、3 - 平均的、4 - 高、5 - 致命的）。
hosts			array	トリガー式に含まれるホストの一覧。配列には少なくとも 1 つの要素が含まれます。
	-		object	
		host	string	ホスト名。
		name	string	表示用ホスト名。
groups			array	トリガー式に含まれるすべてのホストのホストグループ一覧。配列には少なくとも 1 つの要素が含まれます。
	-		string	ホストグループ名。
tags			array	障害タグの一覧（空の場合があります）。
	-		object	
		tag	string	タグ名。
		value	string	タグの値（空の場合があります）。

復旧イベントについては、以下の情報がエクスポートされます。

Field	Type	Description
clock	number	障害が復旧した時点までの Epoch からの経過秒数（整数部）。
ns	number	正確な障害復旧時刻を取得するために clock に加算されるナノ秒数。
value	number	0（常に）。
eventid	number	復旧イベント ID。
p_eventid	number	障害イベント ID。

例

障害:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

リカバリ:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

障害 (多重障害イベントの生成):

```
{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Either Zabbix agent is unreachable on Ho
```

リカバリ:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

アイテムの値

収集されたアイテムの値について、以下の情報がエクスポートされます。

Field		Type	Description
host		object	アイテムのホストのホスト名。
	host	string	ホスト名。
	name	string	表示用ホスト名。
groups		array	アイテムのホストが属するホストグループの一覧。配列には少なくとも 1 つの要素が含まれている必要があります。
	-	string	ホストグループ名。
item_tags		array	アイテムタグの一覧（空の場合があります）。

Field		Type	Description
-		object	
	tag	string	タグ名。
	value	string	タグの値（空の場合があります）。
itemid		number	アイテム ID。
name		string	表示用アイテム名。
clock		number	値が収集された時点までの、Epoch からの経過秒数（整数部）。
ns		number	正確な値の収集時刻を取得するために `clock` に加算されるナノ秒数。
timestamp (*Log* の み)		number	利用できない場合は 0。
source (*Log* の み)		string	利用できない場合は空文字列。
severity (*Log* の み)		number	利用できない場合は 0。
eventid (*Log* の み)		number	利用できない場合は 0。
value		number (数値アイテムの場合) または string (テキストアイテムの場合)	収集されたアイテムの値。
type		number	収集された値のタイプ： 0 - 数値（浮動小数点）, 1 - 文字列, 2 - ログ, 3 - 数値（符号なし）, 4 - テキスト, 5 - バイナリ

例

数値（符号なし）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

数値（浮動小数点）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

文字列、テキスト値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

ログ値:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

トレンド

計算されたトレンド値について、以下の情報がエクスポートされます。

Field			Type	Description
host			object	アイテムのホストのホスト名。
	host		string	ホスト名。
	name		string	表示用ホスト名。
groups			array	アイテムのホストに属するホストグループの一覧。配列には少なくとも 1 つの要素が含まれている必要
	-		string	ホストグループ名。
item_tags			array	アイテムタグの一覧（空の場合があります）。
	-		object	
		tag	string	タグ名。
		value	string	タグの値（空の場合があります）。
itemid			number	アイテム ID。

Field	Type	Description
name	string	表示用アイテム名。
clock	number	値が収集された時点までの、Epoch からの経過秒数（整数部）。
count	number	指定された 1 時間に収集された値の数。
min	number	指定された 1 時間におけるアイテムの最小値。
avg	number	指定された 1 時間におけるアイテムの平均値。
max	number	指定された 1 時間におけるアイテムの最大値。
type	number	値のタイプ: 0 - 数値（浮動小数）, 3 - 数値（符号なし）

例

数値（符号なし）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

数値（浮動小数点）値:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

4 アイテム

サイドバーから、参照したい項目を選択してください。

1 vm.memory.size パラメーター

概要

このセクションでは、vm.memory.size[<mode>] エージェントアイテムのいくつかのパラメータの詳細を説明します。

パラメータ

このアイテムでは、以下のパラメータを使用できます。

- **active** - 現在使用中、またはごく最近使用されたメモリであり、RAM 上に存在します
- **anon** - ファイルに関連付けられていないメモリ（そこから再読み込みできない）
- **available** - 利用可能なメモリ。プラットフォームによって計算方法が異なります（以下の表を参照）
- **buffers** - ファイルシステムのメタデータなどのためのキャッシュ
- **cached** - さまざまな用途のキャッシュ
- **exec** - 実行可能コード。通常は（プログラム）ファイルに由来します
- **file** - 最近アクセスされたファイルの内容のキャッシュ
- **free** - メモリを要求する任意のエンティティがすぐに利用できるメモリ
- **inactive** - 未使用としてマークされているメモリ
- **pavailable** - 'total' に対する'available' メモリの割合（`available/total*100` として計算）
- **pinned** - 'wired' と同じ
- **pused** - 'total' に対する'used' メモリの割合（`used/total*100` として計算）
- **shared** - 複数のプロセスから同時にアクセスされる可能性があるメモリ
- **slab** - カーネルが自身の使用のためにデータ構造をキャッシュするのに使用するメモリの総量
- **total** - 利用可能な物理メモリの総量
- **used** - 使用中のメモリ。プラットフォームによって計算方法が異なります（以下の表を参照）
- **wired** - 常に RAM 上に保持されるようマークされたメモリです。ディスクへ移動されることはありません。

Warning:
これらのパラメータの一部はプラットフォーム固有であり、使用しているプラットフォームでは利用できない場合があります。詳細はZabbix agent items を参照してください。

available および **used** のプラットフォーム固有の計算:

Platform	**"available"**	**"used"**
AIX	free + cached	実際に使用中のメモリ
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free

Platform	”available”	”used”
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (RHEL 7 の 3.10 にもバックポート済み)	/proc/meminfo。詳細は Linux カーネルのdocumentationの”MemAvailable” を参照してください。 なお、すべてのページキャッシュが解放可能なわけではなく、また計算に low watermark が使用されるため、free + buffers + cached はもはや’available’ と等しくありません。	total - free
NetBSD	inactive + execpages + file + free	total - free
OpenBSD	inactive + free + cached	active + wired
OSX	inactive + free	active + wired
Solaris	free	total - free
Win32	free	total - free

Attention:
vm.memory.size[used] と *vm.memory.size[available]* の合計は、必ずしも total と等しくなるとは限りません。たとえば FreeBSD では:
* active、inactive、wired、cached メモリは、何らかの有用な情報を保持しているため、used と見なされます。
* 同時に、inactive、cached、free メモリは、より多くのメモリを要求するプロセスに即座に割り当てることができるため、available と見なされます。

したがって、inactive メモリは同時に used でもあり available でもあります。このため、*vm.memory.size[used]* アイテムは情報提供のみを目的として設計されており、一方 *vm.memory.size[available]* はトリガーで使用することを目的として設計されています。

以下も参照

1. 異なる OS でのメモリ計算に関する追加の詳細

2 パッシブおよびアクティブエージェントチェック

概要

このセクションでは、Zabbix エージェント およびZabbix エージェント 2 によって実行されるパッシブチェックとアクティブチェックの詳細を説明します。

Zabbix は、エージェントとの通信に JSON ベースの通信プロトコルを使用します。

Zabbix 7.0 以降、Zabbix エージェントと Zabbix エージェント 2 のプロトコルは統一されています。Zabbix エージェントと Zabbix エージェント 2 のリクエスト/レスポンスの違いは、”variant” タグの値で表されます。

パッシブチェック

パッシブチェックは単純なデータ要求です。Zabbix サーバーまたはプロキシが何らかのデータ（たとえば CPU 負荷）を要求し、Zabbix エージェントがその結果をサーバーに返します。

パッシブチェックは非同期で実行されます。つまり、他のチェックを開始する前に、1 つの要求に対する応答を受信している必要はありません。DNS 名前解決も同様に非同期です。

エージェント poller は、DNS ルックアップで返されたすべてのアドレスへの接続を試みます。これにより、1 つの IP アドレスに到達できない場合でも、poller は次に利用可能なアドレスを試行し、接続成功の可能性が高まります。この機能強化は、Zabbix サーバーとプロキシの両方に適用されます。

非同期チェックの最大同時実行数は 1000 です（MaxConcurrentChecksPerPollerで定義）。

非同期エージェント poller の数は、StartAgentPollersパラメータで定義されます。

サーバー要求

ヘッダーおよびデータ長の定義については、プロトコルの詳細を参照してください。

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
```

```
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	"passive checks"
data		*array of object*	yes	パッシブチェックアイテム。
	key	*string*	yes	マクロ展開後のアイテムキー。
	timeout	*number*	yes	通信タイムアウト。

エージェント応答

```
{
  "version": "7.4.0",
  "variant": 2,
  "data": [
    {
      "value": "7.4.0"
    }
  ]
}
```

Field		Type	Mandatory	Value
version		*string*	yes	エージェントのバージョン番号。
variant		*number*	yes	エージェントのバリアント（*1* - Zabbix エージェント、*2* - Zabbix エージェント 2）。
data		*array of object*	yes	チェック結果を含みます。
	value	*string*	no	チェックが成功した場合のアイテム値。
	error	*string*	no	チェックが成功しなかった場合のエラーメッセージ。

たとえば、サポートされているアイテムの場合:

1. サーバーが TCP 接続を開く
2. サーバーが **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”agent.ping”,”timeout”:3}]}** を送信する
3. エージェントが要求を読み取り、次のように応答する **<HEADER><DATALEN>{”version”:”7.4.0”,”variant”:2,”data”:[{”value”:1}]}**
4. サーバーがデータを処理して値を取得する。この場合は「1」
5. TCP 接続が閉じられる

サポートされていないアイテムの場合:

1. サーバーが TCP 接続を開く
2. サーバーが **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”vfs.fs.size[/nono]”,”timeout”:3}]}** を送信する
3. エージェントが要求を読み取り、次のように応答する **<HEADER><DATALEN>{”version”:”7.4.0”,”variant”:2,”data”:[{”error”:”Unsupported item key.”}]}**
4. サーバーがデータを処理し、指定されたエラーメッセージとともにアイテムの状態を「未サポート」に変更する
5. TCP 接続が閉じられる

旧プロトコルへのフェイルオーバー

Zabbix サーバーまたはプロキシが、プレーンテキストプロトコルを使用する 7.2 より前のバージョンのエージェントと確実に連携できるように、旧プロトコルへのフェイルオーバーが実装されています。

再起動後、またはインターフェース設定の変更時に、JSON プロトコル（7.0 以降）を使用したパッシブチェックが実行されます。有効な JSON が応答として受信されない場合（エージェントが”ZBX_NOTSUPPORTED” を送信した場合）、Zabbix はインターフェースを旧プロトコルとしてキャッシュし、アイテムキーのみを送信してチェックを再試行します。

Zabbix サーバー/プロキシは 1 時間ごとにすべてのインターフェースで新プロトコルの使用を再度試み、必要に応じて旧プロトコルにフォールバックします。

アクティブチェック

アクティブチェックでは、より複雑な処理が必要です。
エージェントはまず、サーバー/プロキシから、独立して処理するためのアイテムおよび/またはリモートコマンドの一覧を取得する必要があります。

アクティブチェックの取得元となるサーバー/プロキシは、エージェントの設定ファイルの「ServerActive」パラメータに一覧表示されます。
これらのチェックを問い合わせる頻度は、同じ設定ファイル内の「RefreshActiveChecks」パラメータで設定されます。
ただし、アクティブチェックの更新に失敗した場合は、ハードコードされた 60 秒後に再試行されます。

Zabbix 6.4 以降、エージェント（アクティブモード）は、サーバー/プロキシから 2 分ごと（デフォルト）に設定の完全なコピーを受信しなくなりました。代わりに、ネットワークトラフィックとリソース使用量を削減するために、5 秒ごと（デフォルト）に設定の増分同期が実行され、エージェントがまだ設定を受け取っていない場合、またはホスト設定、グローバルマクロ、またはグローバル正規表現に変更があった場合にのみ、サーバー/プロキシから設定の完全なコピーが提供されます。

エージェントは定期的に新しい値をサーバーに送信します。エージェントが実行すべきリモートコマンドを受信した場合は、実行結果も送信されます。アクティブエージェントでのリモートコマンド実行は、Zabbix エージェント 7.0 以降でサポートされています。

エージェントがファイアウォールの内側にある場合は、アクティブチェックのみを使用することを検討してください。この場合、最初の受信接続を許可するためにファイアウォールを変更する必要はありません。

アイテムのリストの取得

エージェントのリクエスト

アクティブチェックのリクエストは、エージェントによって処理されるアクティブチェックを取得するために使用されます。
このリクエストは、エージェントの起動時に送信され、その後はRefreshActiveChecks 間隔で送信されます。

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "7.4.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	ホスト名。
host_metadata	*string*	no	設定パラメータ HostMetadata または HostMetadataItem メトリクス値。
interface	*string*	no	設定パラメータ HostInterface または HostInterfaceItem メトリクス値。
ip	*string*	no	設定されている場合の、設定パラメータ ListenIP の最初の IP。
port	*number*	no	設定されており、かつデフォルトのエージェント待ち受けポートではない場合の設定パラメータ List
version	*string*	yes	エージェントのバージョン番号。
variant	*number*	yes	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）。
config_revision	*number*	no	増分設定同期 の設定識別子。
session	*string*	no	増分設定同期 のセッション識別子。

サーバーのレスポンス

アクティブチェックのレスポンスは、アクティブチェックのリクエストを処理した後にサーバーからエージェントへ送信されます。

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
```

```
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```


Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	失敗時のエラー情報。
data	*array of objects*	no	アクティブチェックのアイテム。ホスト設定が変更されていない場合は省略されます。
key	*string*	no	マクロ展開後のアイテムキー。
itemid	*number*	no	アイテム識別子。
delay	*string*	no	アイテムの更新間隔。 柔軟な間隔/スケジューリング間隔は、Zabbix 7.0 以降、Zabbix agent と Zabbix agent 2 の両方でサポートされています。
lastlogsize	*number*	no	アイテムの lastlogsize。
mtime	*number*	no	アイテムの mtime。
timeout	*string*	no	アイテムのタイムアウト。
refresh_unsupported	*number*	no	未サポートのアイテムの更新間隔。
regexp	*array of objects*	no	グローバル正規表現。
name	*string*	no	グローバル正規表現名。
expression	*string*	no	グローバル正規表現。
expression_type	*number*	no	グローバル正規表現のタイプ。
exp_delimiter	*string*	no	グローバル正規表現の区切り文字。
case_sensitive	*number*	no	グローバル正規表現の大文字・小文字の区別設定。
commands	*array of objects*	no	実行するリモートコマンド。アクションのoperation または手動のscript 実行によってリモートコマンド実行がトリガーされた場合に含まれます。アクティブなエージェントでのリモートコマンド実行は Zabbix agent 7.0 以降でサポートされていることに注意してください。古いアクティブエージェントは、アクティブチェックのサーバーレスポンスに含まれるリモートコマンドを無視します。
command	*string*	no	リモートコマンド。
id	*number*	no	リモートコマンド識別子。
wait	*number*	no	リモートコマンドの実行モード（アクションのoperations からのコマンドは”0”（nowait）、手動のscript 実行からのコマンドは”1”（wait））。
config_revision	*number*	no	増分設定同期 の設定識別子。ホスト設定が変更されていない場合は省略されます。ホスト設定が変更された場合にインクリメントされます。

サーバーは success で応答する必要があります。

例:

1. エージェントが TCP 接続を開く
2. エージェントがチェックのリストを要求する
3. サーバーがアイテムのリストと実行するリモートコマンドを返す
4. エージェントがレスポンスを解析する

5. TCP 接続が閉じられる
6. エージェントが定期的なデータ収集を開始し、リモートコマンドを実行する（Zabbix agent 7.0 以降でサポート）

Attention:
アクティブチェックを使用する場合、（機密性の高い）設定データが、Zabbix server trapper ポートにアクセスできる第三者に利用可能になる可能性があることに注意してください。これは、誰でもアクティブエージェントを装ってアイテム設定データを要求できるためです。encryption オプションを使用しない限り、認証は行われません。

収集したデータの送信

エージェントからの送信

エージェントのデータリクエストには、収集されたアイテムの値と、実行されたリモートコマンドの値（存在する場合）が含まれます。

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
  "commands": [
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "8495cd52070e6ca52b371f29c8574165",
  "host": "Zabbix server",
  "version": "7.4.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	アイテムの値。
id	*number*	yes	値の識別子（ネットワーク障害時に重複した値を確認するために使用される増分カウンター）。
itemid	*number*	yes	アイテムの識別子。
value	*string*	no	アイテムの値。
lastlogsize	*number*	no	アイテムの lastlogsize。
mtime	*number*	no	アイテムの mtime。
state	*number*	no	アイテムの state。
source	*string*	no	値のイベントログソース。
eventid	*number*	no	値のイベントログ eventid。
severity	*number*	no	値のイベントログ severity。
timestamp	*number*	no	値のイベントログ timestamp。
clock	*number*	yes	値のタイムスタンプ（Epoch からの秒数）。
ns	*number*	yes	値のタイムスタンプのナノ秒。

Field	Type	Mandatory	Value
commands	*array of objects*	no	リモートコマンド実行結果。アクティブエージェントでのリモートコマンド実行は、Zabbix agent 7.0 以降でサポートされています。これより古いアクティブエージェントは、アクティブチェックのサーバー応答に含まれるリモートコマンドを無視します。
id	*number*	no	リモートコマンドの識別子。
value	*string*	no	実行が成功した場合のリモートコマンド実行結果。
error	*string*	no	実行に失敗した場合のリモートコマンド実行エラーメッセージ。
session	*string*	yes	エージェントの起動時ごとに生成される一意のセッション識別子。
host	*string*	yes	ホスト名。
version	*string*	yes	エージェントのバージョン番号。
variant	*number*	yes	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）。

各値には仮想 ID が割り当てられます。値 ID は単純な昇順カウンターであり、1 つのデータセッション内で一意です（session トークンで識別されます）。この ID は、接続性の悪い環境で送信される可能性がある重複した値を破棄するために使用されます。

サーバー応答

エージェントデータ応答は、エージェントデータリクエストを処理した後にサーバーからエージェントへ送信されます。

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	アイテム処理結果。

Attention:
サーバーで一部の値の送信に失敗した場合（たとえば、ホストまたはアイテムが無効化または削除されている場合）、エージェントはそれらの値の再送を行いません。

例:

1. エージェントが TCP 接続を開く
2. エージェントが値の一覧を送信する
3. サーバーがデータを処理し、ステータスを返送する
4. TCP 接続が閉じられる

Attention:
エラーメッセージはサーバー側で 2048 文字に切り詰められます。

ハートビートメッセージ

エージェントが送信

ハートビートメッセージは、アクティブエージェントから Zabbix サーバー/プロキシに対して、HeartbeatFrequency 秒ごとに送信されます（Zabbix agent/ agent 2 の設定ファイルで設定）。

これは、アクティブチェックの可用性を監視するために使用されます。

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "7.4.0",
  "variant": 2
}
```

フィールド	型	必須	値
request	*string*	はい	`active check heartbeat`

フィールド	型	必須	値
host	*string*	はい	ホスト名。
heartbeat_freq	*number*	はい	エージェントのハートビート頻度（HeartbeatFrequency 設定パラメータ）。
version	*string*	はい	エージェントのバージョン番号。
variant	*number*	はい	エージェントのバリアント（*1* - Zabbix agent、*2* - Zabbix agent 2）。

リダイレクト応答

ホストが再割り当てされた場合、サーバーはエージェントに対して、そのハートビート（および後続のアクティブチェック）を別のプロキシまたはサーバーインスタンスへリダイレクトするよう指示することがあります。

```
{
  "response": "failed",
  "redirect": {
    "revision": 2,
    "address": "192.0.2.0:10055"
  }
}
```

フィールド		型	必須	値
response		*string*	はい	`success` | `failed`
redirect		*object*	はい	リダイレクト指示。
	revision	*number*	はい	設定リビジョン識別子。
	address	*string*	はい	対象のサーバー/プロキシアドレス。

古い XML プロトコル

Note:
Zabbix は、Base64 にエンコードされた XML のデータを 16MB まで取得しますが、デコードされる値は 1 つが 64KB 以下でなければなりません。そうでないと、デコード中にそれが 64KB に切り捨てられます。

3 Windows エージェントアイテムの最小アクセス許可レベル

概要

エージェントを使用してシステムを監視する場合は、エージェントがインストールされているホストからメトリックを取得することをお勧めします。最小特権の原則を使用するには、エージェントから取得するメトリックを判別する必要があります。

このドキュメントの表では、Zabbix エージェントの正しい動作を保証するための最小権限を選択できます。

エージェントの実行ユーザーを「LocalSystem」ではなく別のユーザーにした場合、エージェントを Windows サービスとして実行するために対象ユーザーは「ローカルポリシー」→「ユーザー権限の割り当て」から「サービスとしてログオン」の権限と、Zabbix エージェントのログファイルを作成、書き込み、削除する権限を持っている必要があります。対象ユーザーを ActiveDirectory の *PerformanceMonitorUsers* グループに追加する必要があります。

Note:
「技術的に許容できる最小限の」グループに基づくエージェントの権利を使用する場合は、監視対象のオブジェクトに対する権利を事前に提供する必要があります。

Windows でサポートされている一般的なエージェントアイテム

アイテムキー	ユーザーグループ	
	推奨	技術的に許容できる最小値 (機能は制限されています)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests

アイテムキー	ユーザーグループ	
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Windows 固有のアイテムキー

アイテムキー	ユーザーグループ	
	推奨	技術的に許容できる最小値 (機能は制限されています)
eventlog	Event Log Readers	Guests
net.if.list	Guests	Guests
perf_counter	Performance Monitor Users	Performance Monitor Users
proc_info	Administrators	Guests
service.discovery	Guests	Guests
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Administrators	Guests
vm.vmemory.size	Guests	Guests

4 戻り値のエンコード

Zabbix サーバーは、UTF8 エンコーディングで返されるすべてのテキスト値を想定しています。これは Zabbix エージェント、SSH、Telnet などのあらゆるタイプのチェックに関連しています。

監視対象のさまざまなシステム/デバイスおよびチェックで、値に非 ASCII 文字が返される場合があります。このような場合、ほとんどすべての **zabbix** キーには、追加のアイテムキーパラメーター - **<encoding>** が含まれています。このキーパラメーターはオプションですが、戻り値が UTF8 エンコーディングではなく、非 ASCII 文字が含まれている場合は指定する必要があります。指定していないと結果が予期せぬ値になる可能性があります。

このような場合のさまざまなデータベースバックエンドでの動作の説明は次のとおりです。

MySQL

少なくとも MySQL バージョン 5.1.61 では、値に非 UTF8 エンコーディングの非 ASCII 文字が含まれている場合、この文字と次の文字は、データベースが値を格納するときに破棄されます。警告メッセージは *zabbix_server.log* に書き込まれません。

PostgreSQL

少なくとも PostgreSQL バージョン 9.1.3 では、値に非 UTF8 エンコーディングの非 ASCII 文字が含まれている場合、SQL クエリが失敗し (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) データは保存されません。適切な警告メッセージが *zabbix_server.log* に書き込まれます。

5 ラージファイルサポート

ラージファイルサポートとは、多くの場合 LFS と略され、32bit オペレーティングシステムで 2GB を超えるファイルを処理する機能に適用される用語です。ラージファイルサポートは、少なくともログファイルの監視とすべてのvfs.file.* アイテムに影響します。ラージファイルサポートは、Zabbix コンパイル時のシステムの機能に依存しますが、procfs および swapctl との互換性がないため、32bit Solaris では完全に無効になっています。

6 センサー

概要

各センサーチップは、sysfs の `/sys/devices` ツリー内にそれぞれ専用のディレクトリを持ちます。すべてのセンサーチップを見つけるには、`/sys/class/hwmon/hwmon*` からデバイスのシンボリックリンクをたどる方が簡単です。ここで * は実際の数値 (0,1,2,...) です。

センサーの読み取り値は、仮想デバイスの場合は `/sys/class/hwmon/hwmon*/` ディレクトリに、非仮想デバイスの場合は `/sys/class/hwmon/hwmon*/device` ディレクトリにあります。`hwmon*` または `hwmon*/device` ディレクトリ内にある `name` という名前のファイルにはチップ名が含まれており、これはそのセンサーチップで使用されているカーネルドライバー名に対応します。

1 つのファイルには 1 つのセンサー読み取り値しか含まれません。上記のいずれのディレクトリ内でも、センサー読み取り値を含むファイルの一般的な命名規則は `<type><number>_<item>` です。ここで

- **type** - センサーチップでは”in” (電圧)、”temp” (温度)、”fan” (ファン) などです。
- **item** - ”input” (測定値)、”max” (上限しきい値)、”min” (下限しきい値) などです。
- **number** - 複数存在し得る要素には常に使用されます (通常は 1 から始まりますが、電圧は 0 から始まります)。ファイルが特定の要素を参照しない場合は、番号のない単純な名前になります。

ホストで利用可能なセンサーに関する情報は、**sensors-detect** および **sensors** ツール (lm-sensors package) を使用して取得できます。**Sensors-detect** は、利用可能なセンサーに必要なモジュールを判別するのに役立ちます。モジュールがロードされると、**sensors** プログラムを使用してすべてのセンサーチップの読み取り値を表示できます。このプログラムで使用されるセンサー読み取り値のラベルは、一般的な命名規則 (`<type><number>_<item>`) と異なる場合があります。

- `<type><number>_label` というファイルがある場合、`<type><number><item>` 名の代わりに、そのファイル内のラベルが使用されます。
- `<type><number>_label` ファイルがない場合、プログラムは `/etc/sensors.conf` (または `/etc/sensors3.conf`、あるいは別の場所) 内で名前の置換を検索します。

このラベル付けにより、ユーザーはどの種類のハードウェアが使用されているかを判断できます。`<type><number>_label` ファイルも設定ファイル内のラベルも存在しない場合、ハードウェアの種類は name 属性 (`hwmon*/device/name`) によって判別できます。`zabbix_agent` が受け付ける実際のセンサー名は、**sensors** プログラムを `-u` パラメータ付きで実行することで取得できます (`sensors -u`)。

sensors プログラムでは、利用可能なセンサーはバスの種類ごとに分けて表示されます (ISA adapter、PCI adapter、SPI adapter、Virtual device、ACPI interface、HID adapter)。

センサー識別子の取得

sensors コマンドで表示されるセンサーラベルは、チップベンダーによってラベル付けが異なるため、常にそのまま直接使用できるとは限りません。例えば、sensors の出力には次のような行が含まれる場合があります。

```
sensors
in0:          +2.24 V  (min =  +0.00 V, max =  +3.32 V)
Vcore:        +1.15 V  (min =  +0.00 V, max =  +2.99 V)
+3.3V:        +3.30 V  (min =  +2.97 V, max =  +3.63 V)
+12V:        +13.00 V  (min =  +0.00 V, max = +15.94 V)
M/B Temp:     +30.0°C  (low  = -127.0°C, high = +127.0°C)
```

これらのうち、表示名が内部識別子に直接対応しているのは 1 つだけです。

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

他の表示ラベル（Vcore や +12V など）を使用しようとしても動作しません。

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Zabbix が受け付ける内部センサー識別子を確認するには、sensors -u を実行します。出力には次のような内容が表示される場合があります。

```
sensors -u
...
Vcore:
  in1_input: 1.15
  in1_min: 0.00
  in1_max: 2.99
  in1_alarm: 0.00
...
+12V:
  in4_input: 13.00
  in4_min: 0.00
  in4_max: 15.94
  in4_alarm: 0.00
...
```

したがって、Vcore ラベルは識別子 in1 に対応し、+12V は in4 に対応します。カーネルの hwmon sysfs 仕様によると、これらはチップのピン上の電圧であり、一般的にはスケーリングが必要になる場合があります。

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix では、電圧（in）だけでなく、電流（curr）、温度（temp）、ファン速度（fan）の読み取り値も取得できます。

互換性

レガシーディストリビューション

センサーの読み取り値は、/proc/sys/dev/sensors ディレクトリから取得されます。

- **device** - デバイス名（<mode> が使用されている場合は、正規表現です）
- **sensor** - センサー名（<mode> が使用されている場合は、正規表現です）
- **mode** - 使用可能な値: avg、max、min（このパラメータが省略された場合、device と sensor はそのままの文字列として扱われます）

キーの例: sensor[w83781d-i2c-0-2d,temp1]。

現在のディストリビューション

センサーの読み取り値は /sys/class/hwmon ディレクトリから取得されます。

- **device** - デバイス名（正規表現ではありません）。デバイス名には、実際のデバイス名（例: 0000:00:18.3）または sensors プログラムを使用して取得した名前（例: k8temp-pci-00c3）を指定できます。どちらの名前を使用するかはユーザーが選択します。
- **sensor** - センサー名（正規表現ではありません）。
- **mode** - 指定可能な値: avg、max、min（このパラメータを省略した場合、device と sensor はそのままの値として扱われます）。

キーの例:

sensor[k8temp-pci-00c3,temp,max] または sensor[0000:00:18.3,temp1]。

sensor[smsc47b397-isa-0880,in,avg] または sensor[smsc47b397.2176,in1]。

7 proc.mem アイテムの **memtype** パラメーターに関する注意

概要

memtype パラメーターは、Linux,AIX,FreeBSD, および Solaris でサポートされています。

'memtype' の 3 つの一般的な値である pmem、rss、vsize は、これらすべてのプラットフォームでサポートされています。さらに一部のプラットフォームでは固有の'memtype' 値がサポートされています。

AIX

以下の表が AIX の'memtype' パラメーターでサポートされている値です。

サポート値	説明	procentry64 構造体のソース	互換
vsize [1]	仮想メモリサイズ	pi_size	
pmem	実メモリのパーセンテージ	pi_prm	ps -o pmem
rss	物理メモリ消費量	pi_trss + pi_drss	ps -o rssize
size	プロセスのサイズ (コード + データ)	pi_dvm	"ps gvw" SIZE column
dsize	データサイズ	pi_dsize	
tsize	テキスト (コード) サイズ	pi_tsize	"ps gvw" TSIZ column
sdsize	シェアードライブラリのデータサイズ	pi_sdsize	
drss	データの物理メモリ消費量	pi_drss	
trss	テキストの物理メモリ消費量	pi_trss	

AIX に関する注意事項:

1. AIX で proc.mem[] 項目キーのパラメータを選択する際は、プロセス選択基準を絞り込むようにしてください。そうしないと、proc.mem[] の結果に不要なプロセスがカウントされてしまうおそれがあります。

例:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss] [u|2879488]
```

この例では、コマンドラインで（正規表現で一致させた）パラメータのみを指定すると、Zabbix エージェントが自分のプロセスを検知してしまう結果を表示しています。これはおそらく望ましい結果ではありません。

2. プロセスを参照するために"ps -ef" を使用しないでください。これはカーネルプロセス以外のプロセスのみを表示します。Zabbix エージェントが認識するすべてのプロセスを表示するには、"ps -Af" を使用してください。

3. Zabbix エージェント proc.mem[] がプロセスを選択する方法を'topasrec' の例で確認してみましょう。

```
$ ps -Af | grep topasrec
root 10747984 1 0 Mar 16 - 0:00 /usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf daily/ -ypersistent=1 -
```

proc.mem[] には引数があります:

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

1 つ目の条件はプロセス名 (引数 <name>) です。この例では、Zabbix エージェントはこれを'topasrec' として認識します。一致させるには、'topasrec' を指定するか、空のままにする必要があります。2 つ目の条件はユーザー名 (引数 <user>) です。一致させるには、'root' を指定するか、空のままにする必要があります。プロセス選択に使用される 3 つ目の条件は引数 <cmdline> です。Zabbix エージェントは、この値を'/usr/bin/topasrec -L -s 300 -R 1 -r 6 -o /var/perf/daily/ -ypersistent=1 -O type=bin -ystart_time=04:08:54,Mar16,2023' として認識します。一致させるには、この文字列に一致する正規表現を指定するか、空のままにする必要があります。

引数 <mode> と <memtype> は、上記の 3 つの基準を使用した後に適用されます。

FreeBSD

以下の表が FreeBSD の'memtype' パラメーターでサポートされている値です。

サポート値	説明	kinfo_proc 構造体のソース	互換
vsize	仮想メモリサイズ	kp_eproc.e_vm.vm_map.size or ki_size	ps -o vsz
pmem	実メモリのパーセンテージ	calculated from rss	ps -o pmem

サポート値	説明	kinfo_proc 構造体のソース	互換
rss	物理メモリ消費量	kp_eproc.e_vm.vm_rssize or ki_rssize	ps -o rss
size [1]	プロセスのサイズ (コード + データ + スタック)	tsize + dsize + ssize	
tsize	テキスト (コード) サイズ	kp_eproc.e_vm.vm_tsize or ki_tsize	ps -o tsiz
dsize	データサイズ	kp_eproc.e_vm.vm_dsize or ki_dsize	ps -o dsiz
ssize	スタックサイズ	kp_eproc.e_vm.vm_ssize or ki_ssize	ps -o ssiz

Linux

以下の表が Linux の'memtype' パラメーターでサポートされている値です。

サポート値	説明	/proc/<pid>/status ファイルのソース
vsize [1]	仮想メモリサイズ	VmSize
pmem	実メモリのパーセンテージ	(VmRSS/total_memory) * 100
rss	物理メモリ消費量	VmRSS
data	データセグメントサイズ	VmData
exe	コードセグメントサイズ	VmExe
hwm	物理メモリ消費のピーク	VmHWM
lck	ロックメモリのサイズ	VmLck
lib	シェアードライブラリのサイズ	VmLib
peak	仮想メモリサイズのピーク	VmPeak
pin	固定されたページのサイズ	VmPin
pte	ページテーブルエントリのサイズ	VmPTE
size	プロセスコードとデータとスタックセグメントの合計値	VmExe + VmData + VmStk
stk	スタックセグメントのサイズ	VmStk
swap	スワップスペース使用量	VmSwap

Linux に関する注意：

1. 古い Linux カーネルではサポートされていない `memtype` 値があります。たとえば Linux 2.4 カーネルは、`hwm`、`pin`、`peak`、`pte`、および `swap` の値をサポートしていません。
2. `proc.mem[...,...,...,...,data]` を使用した Zabbix エージェントのアクティブチェックプロセス自己監視は、エージェントの/proc/<pid>/status ファイルの `VmData` 行で報告される値よりも 4kB 大きい値が示されていることがわかりました。自己測定時にエージェントのデータセグメントは 4 kB 増加し、その後前のサイズに戻ります。

Solaris

以下の表が Solaris の'memtype' パラメーターでサポートされている値です。

サポート値	説明	psinfo 構造のソース	互換
vsize [1]	プロセスイメージのサイズ	pr_size	ps -o vsz
pmem	実メモリのパーセンテージ	pr_pctmem	ps -o pmem
rss	物理メモリ消費量 過小評価されている可能性があります。"man ps" の rss の説明を参照してください	pr_rssize	ps -o rss

脚注

[1] デフォルト値

8 proc.mem および proc.num アイテムでのプロセスの選択に関する注意

コマンドラインを変更するプロセス

一部のプログラムは、現在の動作状況を表示する方法としてコマンドラインを変更します。ユーザーは ps や top コマンドを実行することで、その動作状況を確認できます。そのようなプログラムの例としては、*PostgreSQL*、*Sendmail*、*Zabbix* があります。

Linux の例を見てみましょう。ここでは、複数の Zabbix エージェントプロセスを監視したいとします。

ps コマンドでは、対象のプロセスは次のように表示されます。

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6318     1  0 12:01 ?        00:00:00 sbin/zabbix_agentd -c /home/zabbix/zabbix_agentd.conf
zabbix    6319  6318  0 12:01 ?        00:00:01 sbin/zabbix_agentd: collector [idle 1 sec]
zabbix    6320  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #1 [waiting for connection]
zabbix    6321  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #2 [waiting for connection]
zabbix    6322  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: listener #3 [waiting for connection]
zabbix    6323  6318  0 12:01 ?        00:00:00 sbin/zabbix_agentd: active checks #1 [idle 1 sec]
...
```

名前とユーザーでプロセスを選択すると、期待どおりの結果になります。

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

次に、zabbix_agentd 実行ファイルの名前を zabbix_agentd_30 に変更して再起動してみます。

このとき ps の表示は次のようになります。

```
$ ps -fu zabbix
UID        PID  PPID  C STIME TTY          TIME CMD
...
zabbix    6715     1  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30 -c /home/zabbix/zabbix_agentd.conf
zabbix    6716  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: collector [idle 1 sec]
zabbix    6717  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #1 [waiting for connection
zabbix    6718  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #2 [waiting for connection
zabbix    6719  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: listener #3 [waiting for connection
zabbix    6720  6715  0 12:53 ?        00:00:00 sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]
...
```

しかし、この状態で名前とユーザーによってプロセスを選択すると、誤った結果になります。

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

なぜ、実行ファイル名を少し長い名前に変更しただけで、これほど異なる結果になるのでしょうか。

Zabbix エージェントはまずプロセス名を確認します。/proc/<pid>/status ファイルを開き、Name 行を確認します。この例では、Name 行は次のようになっています。

```
$ grep Name /proc/{6715,6716,6717,6718,6719,6720}/status
/proc/6715/status:Name:   zabbix_agentd_3
/proc/6716/status:Name:   zabbix_agentd_3
/proc/6717/status:Name:   zabbix_agentd_3
/proc/6718/status:Name:   zabbix_agentd_3
/proc/6719/status:Name:   zabbix_agentd_3
/proc/6720/status:Name:   zabbix_agentd_3
```

status ファイル内のプロセス名は 15 文字に切り詰められます。

同様の結果は ps コマンドでも確認できます。

```
$ ps -u zabbix
  PID TTY          TIME CMD
...
 6715 ?        00:00:00 zabbix_agentd_3
 6716 ?        00:00:01 zabbix_agentd_3
 6717 ?        00:00:00 zabbix_agentd_3
 6718 ?        00:00:00 zabbix_agentd_3
 6719 ?        00:00:00 zabbix_agentd_3
 6720 ?        00:00:00 zabbix_agentd_3
 ...
```

当然ながら、これは proc.num[] の name パラメータ値 zabbix_agentd_30 とは一致しません。status ファイルのプロセス名との照合に失敗すると、Zabbix エージェントは /proc/<pid>/cmdline ファイルを参照します。

エージェントから見た"cmdline" ファイルの内容は、次のコマンドを実行すると確認できます。

```
$ for i in 6715 6716 6717 6718 6719 6720; do cat /proc/$i/cmdline | awk '{gsub(/\x0/,"<NUL>"); print};'; d
sbin/zabbix_agentd_30<NUL>-c<NUL>/home/zabbix/zabbix_agentd.conf<NUL>
```

```
sbin/zabbix_agentd_30: collector [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><
sbin/zabbix_agentd_30: listener #1 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #2 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: listener #3 [waiting for connection]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><N
sbin/zabbix_agentd_30: active checks #1 [idle 1 sec]<NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL><NUL
```

この例では、/proc/<pid>/cmdline ファイルには、*C* 言語で文字列終端に使用される不可視の非表示文字である null バイトが含まれています。この例では null バイトを"<NUL>" として表示しています。

Zabbix エージェントはメインプロセスの"cmdline" を確認し、zabbix_agentd_30 を取得します。これは name パラメータ値 zabbix_agentd_30 と一致します。そのため、メインプロセスはアイテム proc.num[zabbix_agentd_30,zabbix] によってカウントされます。

次のプロセスを確認する際、エージェントは cmdline ファイルから zabbix_agentd_30: collector [idle 1 sec] を取得します。これは name パラメータ zabbix_agentd_30 に一致しません。そのため、コマンドラインを変更しないメインプロセスだけがカウントされます。他のエージェントプロセスはコマンドラインを変更するため、無視されます。

この例は、このようなケースでは proc.mem[] および proc.num[] でプロセスを選択するために name パラメータを使用できないことを示しています。

> **Note:**
> proc.get[] アイテムでは、Zabbix エージェントがプロセス名のために"cmdline" を確認する際、最後のスラッシュ以降から最初のスペースまたはコロン記号までの名前部分のみを使用します。cmdline ファイルから取得したプロセス名は、その先頭部分が status ファイル内の短縮されたプロセス名と完全に一致する場合にのみ使用されます。このアルゴリズムは、フィルター内のプロセス名と JSON 出力内のプロセス名の両方に対して同じです。

適切な正規表現を指定した cmdline パラメータを使用すると、正しい結果が得られます。

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

コマンドラインを変更するプログラムを監視する際は、proc.get[]、proc.mem[]、proc.num[] アイテムの使用に注意してください。

name および cmdline パラメータを proc.get[]、proc.mem[]、proc.num[] アイテムに設定する前に、proc.num[] アイテムと ps コマンドを使ってパラメータをテストすることをお勧めします。

Linux カーネルスレッド

proc.get[]、proc.mem[]、proc.num[] アイテムでは、cmdline パラメータでスレッドを選択できません

例として、カーネルスレッドの 1 つを見てみましょう。

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

これは、プロセスの name パラメータでは選択できます。

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

しかし、プロセスの cmdline パラメータによる選択は機能しません。

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

その理由は、Zabbix エージェントが cmdline パラメータで指定された正規表現を取得し、それをプロセスの /proc/<pid>/cmdline の内容に適用するためです。カーネルスレッドでは、/proc/<pid>/cmdline ファイルは空です。そのため、cmdline パラメータが一致することはありません。

proc.mem[] および proc.num[] アイテムのスレッドカウント

Linux カーネルスレッドは proc.num[] アイテムによってカウントされますが、proc.mem[] アイテムのメモリは報告されません。

例：

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
1

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
```

```
ZBX_NOTSUPPORTED: Cannot get amount of "VmSize" memory.
```

しかし、カーネルスレッドと同じ名前のユーザープロセスがある場合は次のようになります。

```
$ ps -ef | grep kthreadd
root         2     0  0 09:51 ?        00:00:00 [kthreadd]
zabbix    9611  6133  0 17:58 pts/1    00:00:00 ./kthreadd

$ zabbix_get -s localhost -k 'proc.num[kthreadd]'
2

$ zabbix_get -s localhost -k 'proc.mem[kthreadd]'
4157440
```

proc.num [] はカーネルスレッドとユーザープロセスの両方をカウントしました。proc.mem [] はユーザープロセスのメモリのみを報告し、カーネルスレッドメモリを 0 であるかのようにカウントします。これは ZBX_NOTSUPPORTED が報告された上記の場合とは異なります。

プログラム名がスレッドの 1 つと一致する場合は、proc.mem[] および proc.num[] アイテムを使用するときに注意してください。

パラメータを proc.mem[] および proc.num[] アイテムに入れる前に、proc.num[] アイテムと ps コマンドを使用してパラメータをテストすることをお勧めします。

9 net.tcp.service および **net.udp.service** チェックの実装の詳細

このページでは net.tcp.service および net.udp.service チェック実装における、service パラメーターで指定されたさまざまなサービスについて詳しく説明しています。

net.tcp.service アイテムのパラメータ

ftp

TCP 接続を作成し、応答の最初の 4 文字が"220 " であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 21 が使用されます。

http

特に何もせず TCP 接続を作成します。ポートが指定されていない場合はデフォルトのポート 80 が使用されます。

https

libcurl を使用し（libcurl のみで動作します）、証明書の信頼性を検証せず、SSL 証明書のホスト名を検証せず、応答ヘッダー（HEAD 要求）のみを読み出します。ポートが指定されていない場合はデフォルトポートの 443 が使用されます。

imap

TCP 接続を作成し、応答の最初の 4 文字が"* OK" であることを検証して"a1 LOGOUT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 143 が使用されます。

ldap

LDAP サーバーへの接続を開き、フィルターを (objectClass=*) に設定して LDAP 検索操作を実行して、最初のエントリの最初の属性が正常に取得できることを検証します。ポートが指定されていない場合はデフォルトのポート 389 が使用されます。

nntp

TCP 接続を作成し、応答の最初の 3 文字が"200" または"201" であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 119 が使用されます。

pop

TCP 接続を作成し、応答の最初の 3 文字が"+OK" であることを検証して"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 110 が使用されます。

smtp

TCP 接続を作成し、応答の最初の 3 文字が"220" であり、その後にスペース、行末、またはダッシュが続くことを検証します。ダッシュを含む行は複数行の応答で、ダッシュのない行が受信されるまで応答が再読み取りされます。次に"QUIT\r\n" を送信します。ポートが指定されていない場合はデフォルトのポート 25 が使用されます。

ssh

TCP 接続を作成します。接続が確立されている場合、両側で識別文字列（SSH-major.minor-XXXX）が交換されます。ここでのメジャーとマイナーはプロトコルバージョンであり、XXXX は文字列です。Zabbix は仕様に一致する文字列が見つかったかどうかを確認し、不一致の場合は文字列”SSH-major.minor-zabbix_agent\r\n” または”0\n” を送り返します。ポートが指定されていない場合は、デフォルトのポート 22 が使用されます。

tcp

特に何もせず TCP 接続を作成します。他のチェックとは異なり、ポート指定必須です。

telnet

TCP 接続を作成し、ログインプロンプト (の最後に':') が返ってくることを検証します。ポートが指定されていない場合はデフォルトのポート 23 が使用されます。

net.udp.service アイテムのパラメータ

ntp

UDP を介して SNTP パケットを送信し、RFC 4330、セクション 5に従って応答を検証します。ポートが指定されていない場合は、デフォルトのポート 123 が使用されます。

10 proc.get パラメーター

概要

アイテム**proc.get**[<name>,<user>,<cmdline>,<mode>] は、Linux、Windows、FreeBSD、OpenBSD、NetBSD でサポートされています。

アイテムによって返されるプロセスパラメータの一覧は、オペレーティングシステムおよび'mode' 引数の値によって異なります。

Linux

Linux では、各モードについて以下のプロセスパラメータが返されます。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親 PID	ppid: 親 PID	processes: プロセス数
name: プロセス名	name: プロセス名	vsize: 仮想メモリサイズ
cmdline: 引数を含むコマンド	user: プロセスが実行されるユーザー (実ユーザー)	pmem: 実メモリの割合
user: プロセスが実行されるユーザー (実ユーザー)	group: プロセスが実行されるグループ (実グループ)	rss: 常駐セットサイズ
group: プロセスが実行されるグループ (実グループ)	uid: ユーザー ID	data: データセグメントのサイズ
uid: ユーザー ID	gid: プロセスが実行されるグループの ID	exe: コードセグメントのサイズ
gid: プロセスが実行されるグループの ID	tid: スレッド ID	lib: 共有ライブラリのサイズ
vsize: 仮想メモリサイズ	tname: スレッド名	lck: ロックされたメモリのサイズ
pmem: 実メモリの割合	cputime_user: ユーザーモードで消費された CPU 時間の合計（値はクロックティック単位で表されます。秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で除算してください）	pin: 固定ページのサイズ
rss: 常駐セットサイズ	cputime_system: システムモードで消費された CPU 時間の合計（値はクロックティック単位で表されます。秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で除算してください）	pte: ページテーブルエントリのサイズ
data: データセグメントのサイズ	state: スレッドの状態	size: プロセスのコード + データ + スタックセグメントのサイズ
exe: コードセグメントのサイズ	ctx_switches: コンテキストスイッチ数	stk: スタックセグメントのサイズ
hwm: 常駐セットサイズのピーク値	page_faults: ページフォールト数	swap: 使用中のスワップ領域のサイズ

mode=process	mode=thread	mode=summary
lck: ロックされたメモリのサイズ		cputime_user: ユーザーモードで消費された CPU 時間の合計（値はクロックティック単位で表されます。秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で除算してください）
lib: 共有ライブラリのサイズ		cputime_system: システムモードで消費された CPU 時間の合計（値はクロックティック単位で表されます。秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で除算してください）
peak: 仮想メモリサイズのピーク値		ctx_switches: コンテキストスイッチ数
pin: 固定ページのサイズ		threads: スレッド数
pte: ページテーブルエントリのサイズ		page_faults: ページフォールト数
size: プロセスのコード + データ + スタックセグメントのサイズ		pss: 比例セットサイズメモリ
stk: スタックセグメントのサイズ		
swap: 使用中のスワップ領域のサイズ		
cputime_user: ユーザーモードで消費された CPU 時間の合計（値はクロックティック単位で表されます。秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で除算してください）		
cputime_system: システムモードで消費された CPU 時間の合計（値はクロックティック単位で表されます。秒に変換するには `getconf CLK_TCK` または `sysconf(_SC_CLK_TCK)` で除算してください）		
state: プロセスの状態（procfs から透過的に取得、長い形式）		
ctx_switches: コンテキストスイッチ数		
threads: スレッド数		
page_faults: ページフォールト数		
pss: 比例セットサイズメモリ		

BSD ベースの OS

以下のプロセスパラメータは、各モードについて FreeBSD、OpenBSD、NetBSD で返されます。

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: プロセス名
ppid: 親 PID	ppid: 親 PID	processes: プロセス数
jid: jail の ID（FreeBSD のみ）	jid: jail の ID（FreeBSD のみ）	vsize: 仮想メモリサイズ
jname: jail 名（FreeBSD のみ）	jname: jail 名（FreeBSD のみ）	pmem: 実メモリの使用率（FreeBSD のみ）
name: プロセス名	name: プロセス名	rss: 常駐セットサイズ
cmdline: 引数を含むコマンド	user: プロセスを実行しているユーザー(実ユーザー)	size: プロセスサイズ（コード + データ + スタック）
user: プロセスを実行しているユーザー(実ユーザー)	group: プロセスを実行しているグループ（実グループ）	tsize: テキスト（コード）サイズ
group: プロセスを実行しているグループ（実グループ）	uid: ユーザー ID	dsize: データサイズ
uid: ユーザー ID	gid: プロセスを実行しているグループの ID	ssize: スタックサイズ
gid: プロセスを実行しているグループの ID	tid: スレッド ID	cputime_user: ユーザーモードで消費された合計 CPU 時間（秒単位で報告）

mode=process	mode=thread	mode=summary
vsize: 仮想メモリサイズ	tname: スレッド名	cputime_system: システムモードで消費された合計 CPU 時間（秒単位で報告）
pmem: 実メモリの使用率（FreeBSD のみ） rss: 常駐セットサイズ	cputime_user: ユーザーモードで消費された合計 CPU 時間（秒単位で報告） cputime_system: システムモードで消費された合計 CPU 時間（秒単位で報告）	ctx_switches: コンテキストスイッチ数 threads: スレッド数（NetBSD ではサポートされません）
size: プロセスサイズ（コード + データ + スタック） tsize: テキスト（コード）サイズ dsize: データサイズ ssize: スタックサイズ	state: スレッド状態 ctx_switches: コンテキストスイッチ数 io_read_op: システムが入力を実行しなければならなかった回数 io_write_op: システムが出力を実行しなければならなかった回数	stk: スタックセグメントのサイズ page_faults: ページフォールト数 fds: ファイルディスクリプタ数（OpenBSD のみ） swap: 使用中のスワップ領域サイズ
cputime_user: ユーザーモードで消費された合計 CPU 時間（秒単位で報告） cputime_system: システムモードで消費された合計 CPU 時間（秒単位で報告）		io_read_op: システムが入力を実行しなければならなかった回数 io_write_op: システムが出力を実行しなければならなかった回数
state: プロセス状態（ディスクスリープ/実行中/スリープ中/トレース停止/ゾンビ/その他） ctx_switches: コンテキストスイッチ数 threads: スレッド数（NetBSD ではサポートされません） page_faults: ページフォールト数 fds: ファイルディスクリプタ数（OpenBSD のみ） swap: 使用中のスワップ領域サイズ io_read_op: システムが入力を実行しなければならなかった回数 io_write_op: システムが出力を実行しなければならなかった回数		

Windows

Windows では、各モードについて以下のプロセスパラメータが返されます。

mode=process	mode=thread	mode=summary
pid: PID ppid: 親 PID name: プロセス名 user: プロセスの実行ユーザー	pid: PID ppid: 親 PID name: プロセス名 user: プロセスの実行ユーザー	name: プロセス名 processes: プロセス数 vmsize: 仮想メモリサイズ wkset: プロセスのワーキングセットサイズ
sid: ユーザー SID	sid: ユーザー SID	cputime_user: ユーザーモードで消費された合計 CPU 時間（ミリ秒単位で報告）
vmsize: 仮想メモリサイズ	tid: スレッド ID	cputime_system: システムモードで消費された合計 CPU 時間（ミリ秒単位で報告）
wkset: プロセスのワーキングセットサイズ		threads: スレッド数
cputime_user: ユーザーモードで消費された合計 CPU 時間（ミリ秒単位で報告）		page_faults: ページフォールト数
cputime_system: システムモードで消費された合計 CPU 時間（ミリ秒単位で報告）		handles: ハンドル数
threads: スレッド数		io_read_b: 読み取られた IO バイト数

mode=process	mode=thread	mode=summary
page_faults: ページフォールト数 handles: ハンドル数 io_read_b: 読み取られた IO バイト数 io_write_b: 書き込まれた IO バイト数 io_read_op: IO 読み取り操作数 io_write_op: IO 書き込み操作数 io_other_b: 読み取りおよび書き込み以外で転送された IO バイト数 io_other_op: 読み取りおよび書き込み以外の IO 操作数		io_write_b: 書き込まれた IO バイト数 io_read_op: IO 読み取り操作数 io_write_op: IO 書き込み操作数 io_other_b: 読み取りおよび書き込み以外で転送された IO バイト数 io_other_op: 読み取りおよび書き込み以外の IO 操作数

11 到達不能/使用不可のホストインターフェース設定

概要

いくつかの設定パラメータは、エージェントチェック（Zabbix、SNMP、IPMI、JMX）が失敗し、ホストインターフェイスに到達できなくなった場合の Zabbix サーバーの動作を定義します。

到達不能なインターフェース

ホストインターフェースは、Zabbix、SNMP、IPMI、または JMX エージェントによるチェックが失敗した後（ネットワークエラー、タイムアウト）に、到達不能として扱われます。Zabbix 6.2.0 以降では、アクティブな Zabbix エージェントチェックもインターフェースの可用性に影響します。アクティブチェックが利用不可になると、エージェントインターフェース全体の可用性ステータスに反映されます。

インターフェースが到達不能になった時点から、**UnreachableDelay** は、1 つのアイテム（LLD ルールを含む）を使用してどのくらいの頻度で再チェックするかを定義します。
これらの再チェックは、到達不能ポーラー（または IPMI チェックの場合は IPMI ポーラー）によって実行されます。
デフォルトでは、連続する到達性チェックの間隔は 15 秒です。

> **Attention:**
> 非同期ポーラーによって実行されるチェックは、到達不能ポーラーには移されません。

Zabbix サーバーのログでは、到達不能は次のようなメッセージで示されます。

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

ログメッセージには、失敗した正確なアイテムとそのタイプ（Zabbix エージェント）が示されます。

> **Note:**
> *Timeout* パラメータも、到達不能時にインターフェースがどれだけ早く再チェックされるかに影響します。Timeout が 20 秒、UnreachableDelay が 30 秒に設定されている場合、次のチェックは最初の試行から 50 秒後に行われます。

UnreachablePeriod パラメータは、到達不能期間の合計時間を定義します。デフォルトでは UnreachablePeriod は 45 秒です。
この値は、インターフェースが利用不可とマークされる前に複数回再チェックされるようにするため、UnreachableDelay の数倍以上にする必要があります。

内部アイテムzabbix[host,active_agent,available] を使用すると、到達不能シナリオにおけるアクティブチェックの可用性を監視できます。

インターフェイスを到達可能に戻す

到達不能期間が終了すると、インターフェイスが再度ポーリングされ、インターフェイスを到達不能状態にしたアイテムの優先度が低下します。到達不能なインターフェイスが再び到達可能になると、監視は自動的に通常に戻ります。

```
resuming Zabbix agent checks on host "New host": connection restored
```

Note:
インターフェイスが使用可能になっても、ホストは次の 2 つの理由ですべてのアイテムをすぐにポーリングしません。
- ホストが過負荷になる可能性があります。
- インターフェイスの復元時間は、計画されたアイテムのポーリングスケジュール時間と常に一致するとは限りません。

そのため、インターフェイスが使用可能になった後、アイテムはすぐにはポーリングされず、次のポーリングラウンドに再スケジュールされます。

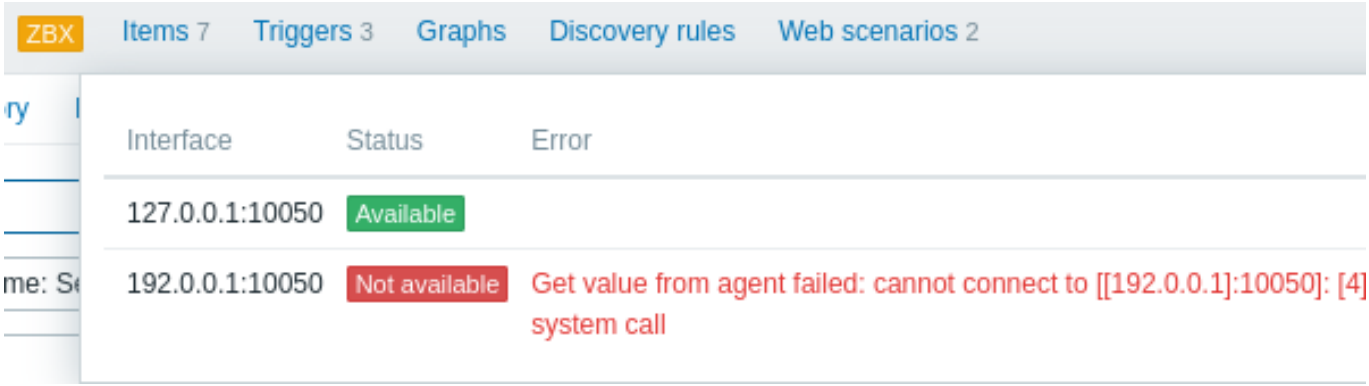
利用不可のインターフェース

UnreachablePeriod が終了してもインターフェースが再び現れない場合、そのインターフェースは利用不可として扱われます。

サーバーのログでは、次のようなメッセージで示されます。

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

また、Web インターフェース では、ホストの可用性アイコンが緑/グレーから黄/赤に変わります（利用不可のインターフェースの詳細は、ホストの可用性アイコンにマウスカーソルを合わせたときに表示されるヒントボックスで確認できます）。



UnavailableDelay パラメータは、インターフェースが利用不可の間に、そのインターフェースをどのくらいの頻度でチェックするかを定義します。

デフォルトでは 60 秒です（したがって、この場合、上記のログメッセージにある”temporarily disabling” は、1 分間チェックを無効にすることを意味します）。

インターフェースへの接続が復旧すると、監視も自動的に通常の状態に戻ります。

```
enabling Zabbix agent checks on host "New host": interface became available
```

12 Zabbix 統計のリモート監視

概要

Zabbix サーバーおよびプロキシの一部の内部メトリクスは、別の Zabbix インスタンスまたはサードパーティーツールからリモートでアクセスできるようにすることが可能です。これは、サポーター/サービスプロバイダーが顧客の Zabbix サーバー/プロキシをリモートで監視したり、または Zabbix が主要な監視ツールではない組織において、包括的な監視構成の中で Zabbix の内部メトリクスをサードパーティーシステムで監視したりする場合に役立ちます。

Zabbix の内部統計は、新しい’StatsAllowedIP’ server/proxy パラメータに一覧表示された、設定可能なアドレスのセットに対して公開されます。リクエストは、これらのアドレスからのみ受け付けられます。

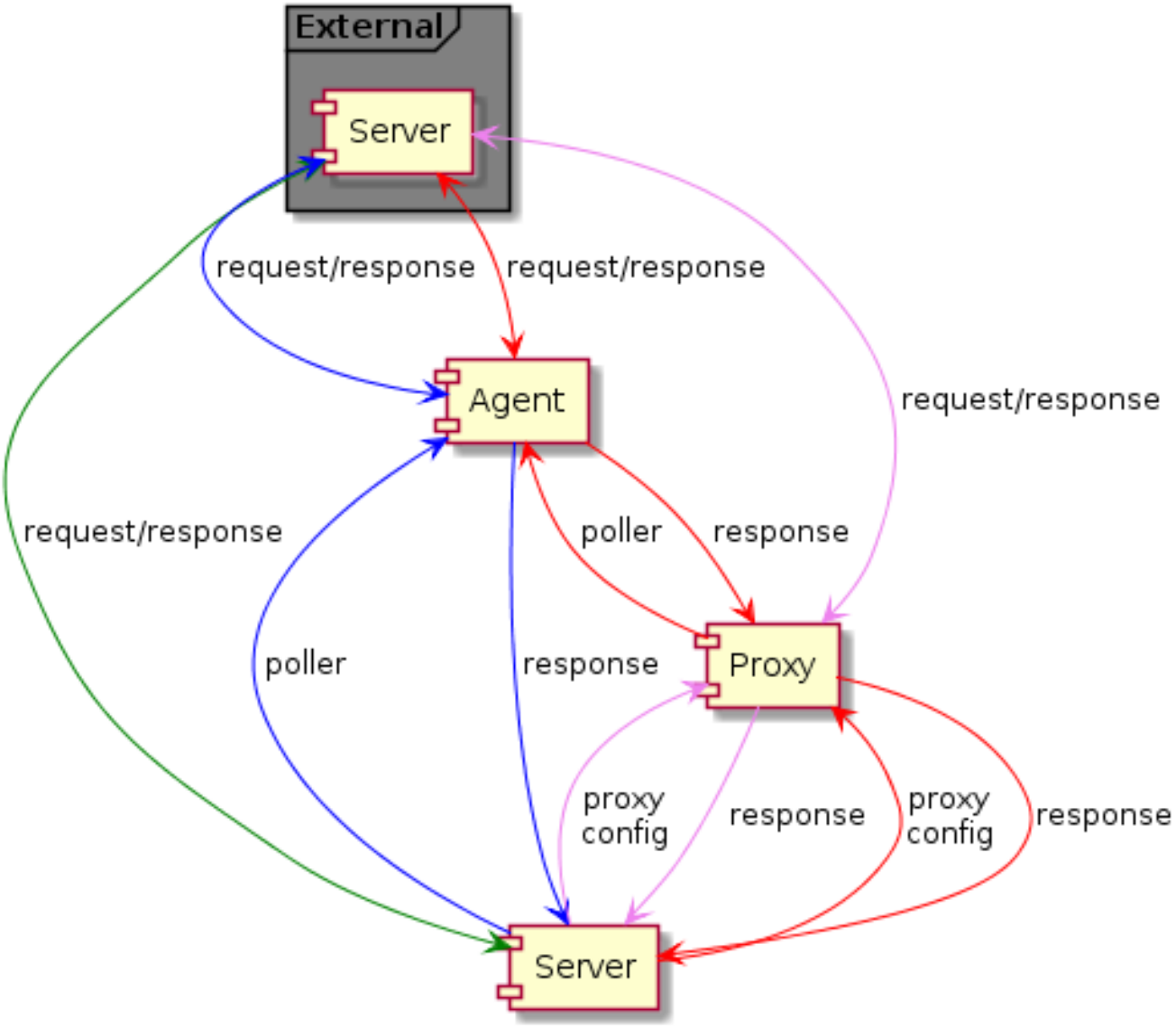
アイテム

別の Zabbix インスタンスで内部統計のクエリを設定するには、次の 2 つの項目を使用します。

- `zabbix[stats,<ip>,<port>]` internal item - Zabbix サーバー/プロキシの直接リモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用します
- `zabbix.stats[<ip>,<port>]` agent item - Zabbix サーバー/プロキシのエージェントベースのリモートクエリ用。<ip> と <port> は、ターゲットインスタンスを識別するために使用されます。

参考情報: 内部項目, Zabbix エージェントアイテム

次の図はコンテキストに応じた、いずれかのアイテムの使用法を示しています。



- - Server → external Zabbix instance (zabbix[stats,<ip>,<port>])
- - Server → proxy → external Zabbix instance (zabbix[stats,<ip>,<port>])
- - Server → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])
- - Server → proxy → agent → external Zabbix instance (zabbix.stats[<ip>,<port>])

ターゲットインスタンスが外部インスタンスによるクエリを許可するには、ターゲットインスタンスの'StatsAllowedIP' パラメータに外部インスタンスのアドレスをリストします。

公開されるメトリクス

stats アイテムは統計情報を一括で収集して JSON を返します。これは、従属アイテムがデータを取得するための基盤となります。以下の内部メトリクスは、2 つのアイテムのいずれかによって返されます。

- zabbix[boottime]
- zabbix[hosts]
- zabbix[items]
- zabbix[items_unsupported]
- zabbix[preprocessing]（サーバーのみ）
- zabbix[preprocessing_queue]（サーバーのみ）
- zabbix[process,<type>,<mode>,<state>]（プロセスタイプベースの統計情報のみ）
- zabbix[rcache,<cache>,<mode>]
- zabbix[requiredperformance]
- zabbix[triggers]（サーバーのみ）
- zabbix[uptime]
- zabbix[vcache,buffer,<mode>]（サーバーのみ）

- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>]（'trends' キャッシュタイプはサーバーのみ）

テンプレート

外部インスタンスから Zabbix サーバーまたはプロキシの内部メトリクスをリモート監視するためのテンプレートを利用できます。

- リモート Zabbix サーバーのヘルス
- リモート Zabbix プロキシのヘルス

複数の外部インスタンスをリモート監視するためにテンプレートを使用する場合は、外部インスタンスの監視ごとに個別のホストが必要になることに注意してください。

トラッパープロセス

外部の Zabbix インスタンスからの内部メトリクス要求の受信は、トラッパープロセスによって処理されます。このプロセスは要求を検証し、メトリクスを収集し、JSON データバッファを作成して、準備した JSON を送り返します。以下は、例えばサーバーから返される JSON です。

```
{
  "response": "success",
  "data": {
    "boottime": N,
    "uptime": N,
    "hosts": N,
    "items": N,
    "items_unsupported": N,
    "preprocessing": {
        "queued": {
          "count": N,
          "size": N
        },
        "direct": {
          "count": N,
          "size": N
        },
        "queue": N
    },
    "preprocessing_queue": N,
    "process": {
      "alert manager": {
        "busy": {
          "avg": N,
          "max": N,
          "min": N
        },
        "idle": {
          "avg": N,
          "max": N,
          "min": N
        },
        "count": N
      },
...
    },
    "queue": N,
    "rcache": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "requiredperformance": N,
```

```
    "triggers": N,
    "uptime": N,
    "vcache": {
      "buffer": {
        "total": N,
        "free": N,
        "pfree": N,
        "used": N,
        "pused": N
      },
      "cache": {
        "requests": N,
        "hits": N,
        "misses": N,
        "mode": N
      }
    },
    "vmware": {
      "total": N,
      "free": N,
      "pfree": N,
      "used": N,
      "pused": N
    },
    "version": "N",
    "wcache": {
      "values": {
        "all": N,
        "float": N,
        "uint": N,
        "str": N,
        "log": N,
        "text": N,
        "not supported": N
      },
      "history": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "index": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      },
      "trend": {
        "pfree": N,
        "free": N,
        "total": N,
        "used": N,
        "pused": N
      }
    }
  }
}
```

内部キューアイテム

別の Zabbix インスタンスで内部キュー統計をリモートクエリできるようにする別の 2 つの項目もあります。

- zabbix[stats,<ip>,<port>,queue,<from>,<to>] internal item - リモート Zabbix サーバー/プロキシへの直接内部キュークエリ用
- zabbix.stats[<ip>,<port>,queue,<from>,<to>] agent item - リモート Zabbix サーバー/プロキシへのエージェントベースの内部キュークエリ用

参考情報: 内部項目, Zabbix エージェントアイテム

13 Zabbix を使用した **Kerberos** の設定

概要

Kerberos 認証は、Zabbix の Web 監視および HTTP アイテムで使用できます。

このページでは、Debian/Ubuntu 上で Zabbix サーバーが `www.example.com` の Web 監視を実行できるように、Zabbix プロセス用の Kerberos プリンシパルを使用して Kerberos を設定する例について説明します。

設定

1. KDC とクライアントユーティリティをインストールします:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

パッケージのセットアップ中に、プロンプトに回答します。例:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (or your FQDN)
Administrative server for your Kerberos realm: localhost (or your FQDN)
```

2. わかりやすいホスト名をマッピングします（任意、ローカルテスト用）。

DNS がない場合は、/etc/hosts を編集して DC と Web サーバーのエントリを追加します:

```
sudo vi /etc/hosts
```

追加する行の例:

```
192.168.1.100  dc01.example.com dc01
```

3. Kerberos クライアントと KDC realm を設定します:

```
sudo vi /etc/krb5.conf
```

設定例:

```
[libdefaults]
    default_realm = EXAMPLE.COM
    dns_lookup_realm = false
    dns_lookup_kdc = false
    rdns = false
    ticket_lifetime = 24h
    renew_lifetime = 7d
    forwardable = true

[realms]
    EXAMPLE.COM = {
        kdc = dc01.example.com
        admin_server = dc01.example.com
    }

[domain_realm]
    .example.com = EXAMPLE.COM
    example.com = EXAMPLE.COM
```

`.localdomain` やその他の非公開名を使用する予定がある場合は、hostname→realm のマッピングが機能するように、明示的な domain→realm マッピングを追加してください。
ここに不一致があると、`Server not found in Kerberos database` エラーが発生します。

4. Kerberos データベースを初期化します（1 回のみ、KDC ホスト上）。プロンプトが表示されたら、安全なマスターパスワードを設定します:

```
sudo krb5_newrealm
```

5. クライアントが使用する正確なホスト名で HTTP/host.fqdn@REALM principal を作成します。小文字を推奨します（例: HTTP/web.example.com@EXAMPLE.COM）。大文字小文字や名前の不一致があると、Server not found in Kerberos database が発生します。

```
sudo kadmin.local
```

kadmin.local 内で:

```
addprinc kerb-admin@EXAMPLE.COM      # 管理 principal
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

keytab を Web ホストに移動します（同じマシンであればそのままでも可）。その後、Apache から使用できる権限を設定します:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verify
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Apache GSSAPI モジュールをインストールして有効化します:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> すべての mod_auth_gssapi バージョンが、すべての Gssapi* ディレクティブをサポートしているわけではありません。Apache が Invalid command 'GssapiCredStore' で失敗する場合は、未対応のディレクティブを削除するか、モジュールをアップグレードしてください。

7. VirtualHost を設定します（DocumentRoot / Zabbix UI へのパスは環境に合わせて調整してください）:

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

zabbix.conf の内容:

```
<VirtualHost *:80>
    ServerName dc01.example.com
    DocumentRoot /usr/share/zabbix/ui
    <Directory /usr/share/zabbix/ui>
        Options FollowSymLinks
        AllowOverride None
        Require all granted
        AuthType GSSAPI
        AuthName "Kerberos Login"
        GssapiCredStore keytab:/etc/apache2/http.keytab
        GssapiLocalName On
        Require valid-user
    </Directory>
    RequestHeader set X-Remote-User %{REMOTE_USER}s env=REMOTE_USER
    RequestHeader unset Authorization
</VirtualHost>
```

Apache を再起動します:

```
sudo systemctl restart apache2
```

8. KDC サービスを有効化/起動し、待ち受けポートを確認します（KDC ホスト上）:

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # or: sudo netstat -tnlp | grep :80
```

9. テスト用に TGT を取得します（チケットを使用するユーザーとして実行します）。

チケット一覧に krbtgt/EXAMPLE.COM@EXAMPLE.COM が表示されるはずです。

チケットを必要とする同じ OS ユーザーとして kinit を実行してください（例: Web 監視の場合は zabbix、対話型ブラウザー SSO テストの場合は www-data/Apache）。
別の OS ユーザーに発行されたチケットは、KRB5CCNAME と権限を調整しない限り参照できません。

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. curl で SPNEGO 交換をテストします（有効な TGT を持つクライアントから実行）。200 OK（またはアプリへのリダイレクト）が返れば、SPNEGO は成功しています:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. 必要に応じて、Zabbix UI で HTTP 認証ログインを受け付ける場合は、Zabbix Web インターフェースで HTTP 認証を有効にします（ui/conf/zabbix.conf.php）:

```
$ALLOW_HTTP_AUTH = true;
```

Web UI で、*Users* > *Authentication* に移動し、*HTTP settings* タブを開きます。
Enable HTTP authentication チェックボックスをオンにし、ポップアップで *Ok* をクリックします。
Default login form ドロップダウンで"HTTP login form" を選択します。
Case-sensitive login がディレクトリポリシーに適しているかを判断してください。
最後に *Update* ボタンをクリックします。

12. ブラウザー設定(例として Firefox を使用): ブラウザーが Kerberos トークンを自動送信するように、network.negotiate-auth.trusted-に Negotiate を実行するホスト（dc01.example.com）を設定します。

about:config 内で:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

これで http://dc01.example.com にアクセスすると、フォームを表示せずにそのまま Zabbix にログインできるはずです。

13. キーとチケットを最新の状態に保ちます。Kerberos チケットのデフォルト有効期間は約 10 時間です。
期限切れを防ぐために、cron/systemd タイマーを追加します:

```
####for the web service
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####for the monitoring user
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. 保守確認項目:

- klist -k /etc/apache2/http.keytab — keytab にサービス principal が存在することを確認します。
- sudo tail -f /var/log/apache2/error.log — GSSAPI エラーを監視します（gss_acquire_cred[_from]() failed to get server creds は、keytab/権限の問題、または principal の欠落を意味します）。
- curl --negotiate が 401/403 を返す場合は、多くの場合 principal の誤り、チケットなし、host ヘッダー不一致、またはファイルシステム権限の問題が原因です。ログと /etc/krb5.conf の domain マッピングを確認してください。

セキュリティおよびファイル権限に関する注意

keytab ファイルは、それを必要とするアカウントのみが読み取り可能でなければなりません。
権限の例: zabbix ユーザーの keytab では所有者を zabbix:zabbix とした 0400、Apache の keytab では 0440 と root:www-data。

ホスト上に長期間有効な平文パスワードを保存することは避けてください。
可能な場合は、keytab またはドメイン参加済みマシンプリンシパルを使用してください。

KRB5CCNAME を設定するテストやスクリプトを実行したり、keytab をコピーしたりする場合は、操作後に所有者と権限を必ず再確認してください。Web サーバーが認証情報を拒否する原因として、ファイル権限の問題はよくあります。

14 modbus.get パラメーター

概要

以下の表は、アイテムmodbus.get のパラメーターの詳細を示しています。

パラメータ

Parameter	Description	Defaults	Example
endpoint	エンドポイントのプロトコルとアドレス。 `protocol://connection_string` として定義します。 使用可能なプロトコル値: *rtu*、*ascii*（Agent 2 のみ）、*tcp* 接続文字列の形式: *tcp* の場合 - `address:port` シリアル回線の場合: *rtu*、*ascii* - `port_name:speed:params` ここで 'speed' - 1200、9600 など 'params' - データビット（5、6、7 または 8）、パリティ（なし/偶数/奇数を表す n、e、o）、ストップビット（1 または 2）	protocol: なし *rtu/ascii* プロトコル: port_name: なし speed: 115200 params: 8n1 *tcp* プロトコル: address: なし port: 502	tcp://192.168.6.1:511 tcp://192.168.6.2 tcp://[::1]:511 tcp://::1 tcp://localhost:511 tcp://localhost rtu://COM1:9600:8n ascii://COM2:1200:7o2 rtu://ttyS0:9600 ascii://ttyS1
slave id	対象デバイスの Modbus アドレス（1 ～247）。 MODBUS Messaging Implementation Guide（23 ページ）を参照してください。 tcp デバイス（GW ではない）はこのフィールドを無視します	serial: 1 tcp: 255 (0xFF)	2
function	空、またはサポートされている機能の値: 1 - Read Coil、 2 - Read Discrete Input、 3 - Read Holding Registers、 4 - Read Input Registers	空	3
address	最初のレジスタ、コイル、または入力のアドレス。 'function' が空の場合、'address' は次の範囲である必要があります: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 'function' が空でない場合、'address' フィールドは 0 ～65535 となり、変更せずにそのまま使用されます（PDU）	空の function: 00001 空でない function: 0	9999
count	'type' の連続した値のうち、デバイスから読み取る数。 ここで: Coil または Discrete input の場合、'type' = 1 ビット それ以外の場合: `(count\*sizeof(type))/2` = 読み取り対象レジスタの実際の数 'offset' が 0 でない場合、その値がこの「実際の数」に加算されます 「実際の数」の許容範囲は 1:65535 です	1	2

Parameter	Description	Defaults	Example
type	データ型: Read Coil および Read Discrete Input の場合 - *bit* Read Holding Registers および Read Input Registers の場合: *int8* - 8 ビット *uint8* - 8 ビット（符号なし） *int16* - 16 ビット *uint16* - 16 ビット（符号なし） *int32* - 32 ビット *uint32* - 32 ビット（符号なし） *float* - 32 ビット *uint64* - 64 ビット（符号なし） *double* - 64 ビット	bit uint16	uint64
endianness	エンディアンの種類: *be* - ビッグエンディアン *le* - リトルエンディアン *mbe* - ミッドビッグエンディアン *mle* - ミッドリトルエンディアン 制限事項: 1 ビットの場合 - be 8 ビットの場合 - be、le 16 ビットの場合 - be、le	be	le
offset	'address' から開始するレジスタ数。この結果は破棄されます。 各レジスタのサイズは 16 ビットです（ランダム読み取りアクセスをサポートしない機器に対応するために必要です）。	0	4

15 VMware のカスタムパフォーマンスカウンター名作成

概要

VMware のパフォーマンスカウンターパスは、`group/counter[rollup]` 形式です。各要素の意味は次のとおりです。

- `group` - パフォーマンスカウンターのグループ。例: *cpu*
- `counter` - パフォーマンスカウンター名。例: *usagemhz*
- `rollup` - パフォーマンスカウンターのロールアップタイプ。例: *average*

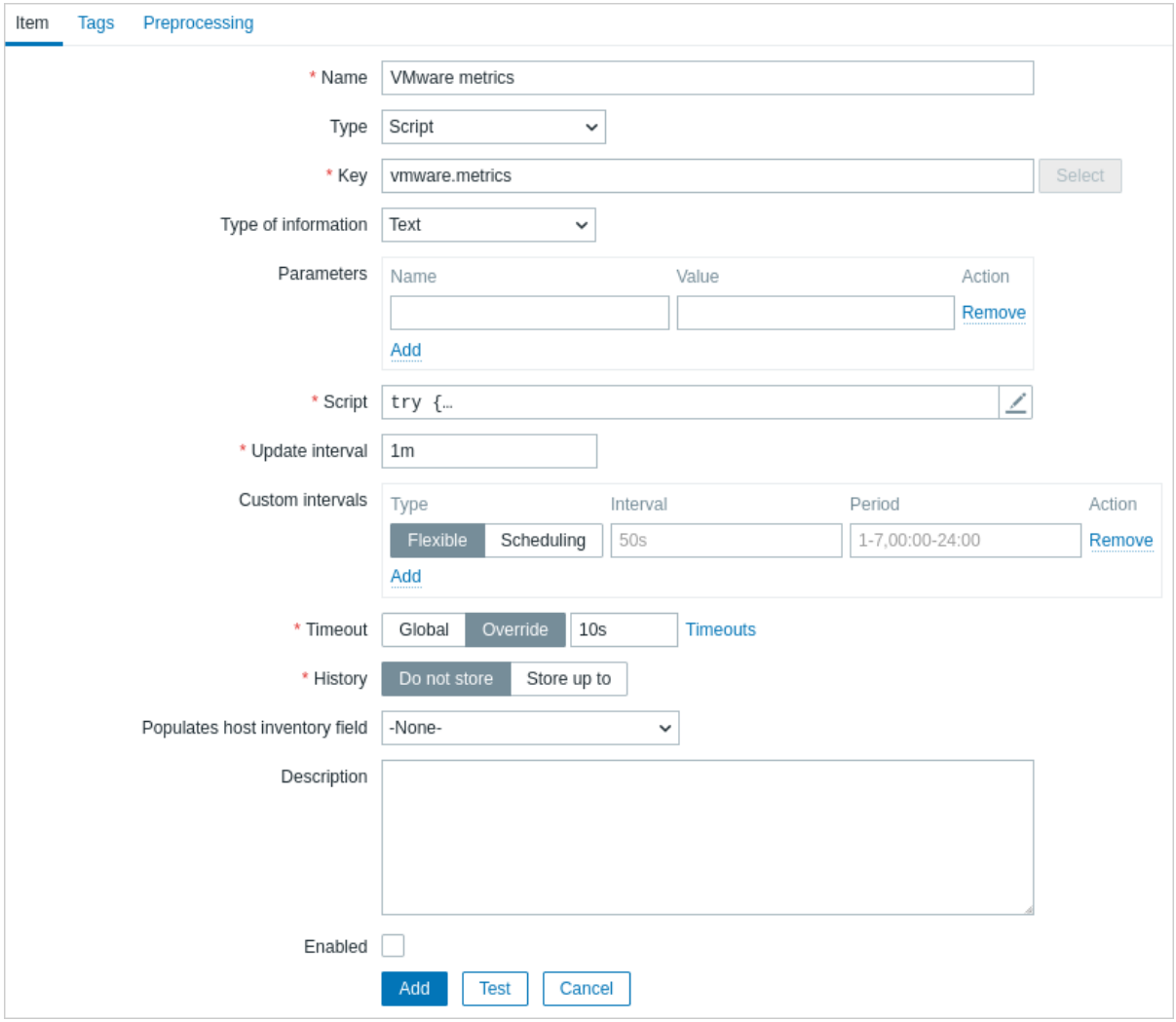
したがって、上記の例では次のカウンターパスになります。`cpu/usagemhz[average]`

パフォーマンスカウンターのグループの説明、カウンター名、およびロールアップタイプは、VMware documentationで確認できます。

Zabbix のスクリプトアイテムを使用して内部名を取得し、カスタムのパフォーマンスカウンター名を作成することも可能です。

設定

1. メインの VMware ホスト（**eventlog[]** アイテムが存在するホスト）に、以下のパラメータで無効化された Script アイテムを作成します。



- 名前: VMware metrics
- タイプ: Script
- キー: vmware.metrics
- 保存する値の型: Text
- スクリプト: 以下で提供されているscriptをコピーして貼り付けます
- タイムアウト: 10
- ヒストリ: 保存しない
- 有効: チェックを外す

Script

```
try {
    Zabbix.log(4, 'vmware metrics script');

    var result, resp,
    req = new HttpRequest();
    req.addHeader('Content-Type: application/xml');
    req.addHeader('SOAPAction: "urn:vim25/6.0"');

    login = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Login>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
            <urn:userName>{$VMWARE.USERNAME}</urn:userName>\
            <urn:password>{$VMWARE.PASSWORD}</urn:password>\
        </urn:Login>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", login);
    if (req.getStatus() != 200) {
```

```
        throw 'Response code: '+req.getStatus();
    }

    query = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:vi
<soapenv:Header/>\
    <soapenv:Body>\
        <urn:RetrieveProperties>\
            <urn:_this type="PropertyCollector">propertyCollector</urn:_this>\
            <urn:specSet>\
                <urn:propSet>\
                   <urn:type>PerformanceManager</urn:type>\
                   <urn:pathSet>perfCounter</urn:pathSet>\
                </urn:propSet>\
                <urn:objectSet>\
                   <urn:obj type="PerformanceManager">PerfMgr</urn:obj>\
                </urn:objectSet>\
            </urn:specSet>\
        </urn:RetrieveProperties>\
    </soapenv:Body>\
</soapenv:Envelope>'
    resp = req.post("{$VMWARE.URL}", query);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }
    Zabbix.log(4, 'vmware metrics=' + resp);
    result = resp;

    logout = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:v
    <soapenv:Header/>\
    <soapenv:Body>\
        <urn:Logout>\
            <urn:_this type="SessionManager">SessionManager</urn:_this>\
        </urn:Logout>\
    </soapenv:Body>\
</soapenv:Envelope>'

    resp = req.post("{$VMWARE.URL}",logout);
    if (req.getStatus() != 200) {
        throw 'Response code: '+req.getStatus();
    }

} catch (error) {
    Zabbix.log(4, 'vmware call failed : '+error);
    result = {};
}

return result;
```

アイテムの設定が完了したら、*Test* ボタンを押し、次に *Get value* を押します。



受信した XML を任意の XML フォーマッタにコピーし、目的のメトリクスを見つけます。

1 つのメトリクスに対する XML の例:

```
<PerfCounterInfo xsi:type="PerfCounterInfo">
    <key>6</key>
```

```
    <nameInfo>
        <label>Usage in MHz</label>
        <summary>CPU usage in megahertz during the interval</summary>
        <key>usagemhz</key>
    </nameInfo>
    <groupInfo>
        <label>CPU</label>
        <summary>CPU</summary>
        <key>cpu</key>
    </groupInfo>
    <unitInfo>
        <label>MHz</label>
        <summary>Megahertz</summary>
        <key>megaHertz</key>
    </unitInfo>
    <rollupType>average</rollupType>
    <statsType>rate</statsType>
    <level>1</level>
    <perDeviceLevel>3</perDeviceLevel>
</PerfCounterInfo>
```

XPath を使用して、受信した XML からカウンターパスを抽出します。上記の例では、XPath は次のようになります。

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

この場合、結果のパフォーマンスカウンターパスは `cpu/usagemhz[average]` です。

16 system.sw.packages.get の戻り値

概要

このセクションでは、`system.sw.packages.get` Zabbix エージェントアイテムの戻り値の詳細について説明します。

詳細

このアイテムの出力はオブジェクトの配列であり、各オブジェクトには以下のキーが含まれます。

- **name** - パッケージ名
- **manager** - このデータを報告したパッケージマネージャー（`rpm`、`dpkg`、`pacman`、`pkgtool`、または `portage`）
- **version** - パッケージのバージョン
- **size** - 非圧縮パッケージサイズ（バイト単位）（利用できない場合は 0 に設定）
- **arch** - パッケージアーキテクチャ
- **buildtime** - 2 つのエントリを持つオブジェクト:
  - **timestamp** - パッケージがビルドされた時刻の UNIX タイムスタンプ（利用できない場合は 0 に設定）
  - **value** - パッケージがビルドされた日時の人間が読みやすい形式の値（利用できない場合は空文字列に設定）
- **installtime** - 2 つのエントリを持つオブジェクト:
  - **timestamp** - パッケージがインストールされた時刻の UNIX タイムスタンプ（利用できない場合は 0 に設定）
  - **value** - パッケージがインストールされた日時の人間が読みやすい形式の値（利用できない場合は空文字列に設定）

例:

```
[
    {
        "name": "util-linux-core",
        "manager": "rpm",
        "version": "2.37.4-3.el9",
        "size": 1296335,
        "arch": "x86_64",
        "buildtime": {
            "timestamp" : 1653552239,
            "value" : "Sep 20 01:39:40 2021 UTC"
```

```
        },
        "installtime": {
            "timestamp" : 1660780885,
            "value" : "Aug 18 00:01:25 2022 UTC"
        }
    },
    {
        "name": "xfonts-base",
        "manager": "dpkg",
        "version": "1:1.0.5",
        "size": 7337984,
        "arch": "all",
        "buildtime": {
            "timestamp": 0,
            "value": ""
        },
        "installtime": {
            "timestamp": 0,
            "value": ""
        }
    }
]
```

17 net.dns.get の戻り値

概要

このセクションでは、Zabbix エージェント 2 アイテムの`net.dns.get` の戻り値の詳細について説明します。

詳細

このアイテムの出力は、アイテムキーで指定されたパラメータに基づく DNS レコード情報を含むオブジェクトです。

例えば、`net.dns.get[,example.com]` アイテムは、拒否されたクエリについて次の JSON を返す場合があります。

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

DNS サーバーの IP アドレスを指定すると、`net.dns.get[192.0.2.0,example.com]` アイテムは次の JSON を返す場合があります。

```
{
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "expire": 1209600,
                "mbox": "noc.dns.example.org.",
                "minttl": 3600,
                "ns": "ns.example.org.",
                "refresh": 7200,
```

```
                "retry": 3600,
                "serial": 2022091378
            },
            "rdlength": 44,
            "ttl": 1205,
            "type": "SOA"
        }
    ],
    "flags": [
        "RA"
    ],
    "query_time": "0.029556",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


接続の問題がある場合、net.dns.get[192.0.2.0,example.com] アイテムはエラーを返すことがあります。


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


次の種類のエラーコードが考えられます。

シナリオ	"zbx_error_code"	"zbx_error_msg"
エラーはなく、DNS 応答が受信されて解析された。	0	
DNS が停止している。	-1	"Communication error"
JSON 解析中にエラーが発生する	-2	"Received unexpected response"

追加のパラメータを指定すると、net.dns.get[192.0.2.0,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,nsid,edns0,aaf
アイテムは次の JSON を返す場合があります。


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
            ]
        },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
```

```
                "a": "192.0.2.0"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


参照

DNS レコードの詳細については、以下を参照してください。

- ドメイン名 - 実装と仕様
- ドメインネームシステム (DNS) パラメータ

18 Windows 上の **system.cpu.util** アイテムに関する注意事項

system.cpu.util アイテムは、CPU 使用率のパーセンテージを提供します。

Windows 用 Zabbix エージェントで collector プロセスが起動されると、N 個の論理プロセッサー（スレッド）に対する N 個のアイテム用バッファが初期化されます。値は Windows パフォーマンスカウンターを使用して定期的にバッファ内で更新されます。これらの値は、`system.cpu.util[n]` が要求されたときに報告されます。ここで、0 <= n < N です。

バッファ内のもう 1 つのアイテムは、`system.cpu.util[total]` に使用されます。

Zabbix は、NUMA システムと、プロセッサーグループをサポートしない古い Windows バージョンの両方をサポートするために、異なるパフォーマンスカウンターを自動的に選択します。

論理プロセッサー（スレッド）数が 64 以下で、プロセッサーグループ数が 1 である場合:

```
N is logical processor (thread) count.

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

それ以外の場合:

```
G is processor group count and N is logical processor (thread) count in group.

\Processor Information(_Total)\% Processor Time
\Processor Information(0,0)\% Processor Time
\Processor Information(0,1)\% Processor Time
\Processor Information(0,2)\% Processor Time
...
\Processor Information(0,N-1)\% Processor Time
...
\Processor Information(G-1,0)\% Processor Time
\Processor Information(G-1,1)\% Processor Time
\Processor Information(G-1,2)\% Processor Time
...
\Processor Information(G-1,N-1)\% Processor Time
```

5 サポートされている関数

詳細を確認するには、それぞれの関数グループをクリックしてください。

関数グループ		関数
集計関数		avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
	Foreach 関数	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
ビット演算関数		bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
日付と時刻の関数		date, dayofmonth, dayofweek, now, time
履歴関数		change, changecount, count, countunique, find, first, firstclock, fuzzytime, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate

関数グループ	関数
トレンド関数	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
数学関数	abs, acos, asin, atan, atan2, avg, cbrt, ceil, cos, cosh, cot, degrees, e, exp, expm1, floor, log, log10, max, min, mod, pi, power, radians, rand, round, signum, sin, sinh, sqrt, sum, tan, truncate
演算子関数	between, in
予測関数	forecast, timeleft
文字列関数	ascii, bitlength, bytelength, char, concat, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, trim, xmlxpath

特に明記されていない限り、これらの関数は以下でサポートされています。

- トリガー条件式
- 計算アイテム
- 式マクロ

Foreach 関数は、集計計算でのみサポートされています。

1 集計関数

特に明記されていない限り、ここに記載されているすべての関数は以下でサポートされています。

- トリガー条件式
- 計算アイテム
- 式マクロ

集計関数は、以下のいずれかを対象に動作します。

- アイテムの履歴。例: `min(/host/key,1h)`
- 唯一のパラメータとしてのforeach 関数。例: `min(last_foreach(/*/key))`（計算アイテムでのみ使用可能。トリガーでは使用できません）

関数は追加情報なしで一覧表示されています。詳細については、関数をクリックしてください。

Function	Description
avg	定義された評価期間内におけるアイテムの平均値。
bucket_percentile	ヒストグラムのバケットからパーセンタイルを計算します。
count	foreach 関数が返す配列内の値の数。
histogram_quantile	ヒストグラムのバケットから φ-quantile を計算します。
item_count	フィルター条件に一致する、設定内の既存アイテム数。
kurtosis	定義された評価期間内に収集された値における、確率分布の「裾の重さ」。
mad	定義された評価期間内に収集された値の中央値絶対偏差。

Function	Description
max	定義された評価期間内におけるアイテムの最大値。
min	定義された評価期間内におけるアイテムの最小値。
skewness	定義された評価期間内に収集された値における、確率分布の非対称性。
stddevpop	定義された評価期間内に収集された値の母標準偏差。
stddevsamp	定義された評価期間内に収集された値の標本標準偏差。
sum	定義された評価期間内に収集された値の合計。
sumofsquares	定義された評価期間内に収集された値の二乗和。
varpop	定義された評価期間内に収集された値の母分散。
varsamp	定義された評価期間内に収集された値の標本分散。

共通パラメータ

- /host/key は、ホストのアイテム履歴を参照する関数に共通する必須の第 1 パラメータです
- (sec|#num)<:time shift> は、ホストのアイテム履歴を参照する関数に共通する第 2 パラメータで、以下を意味します:
  - **sec** - 秒単位の最大評価期間 （時間のサフィックスを使用できます）、または
  - **#num** - 直近に収集された値における最大評価範囲 （ハッシュ記号が前に付く場合）
  - **time shift** （任意）- 評価時点を過去に移動できます。time shift の指定方法の詳細は、こちらを参照してください。

関数の詳細

関数のパラメータに関する一般的な注意事項:

- 関数のパラメータはカンマで区切ります。
- オプションの関数パラメータ（またはパラメータ部分）は、< > で示されます。
- 関数固有のパラメータは、各関数で説明されています。
- /host/key および (sec|#num)<:timeshift> パラメータは引用符で囲んではなりません。

avg(/host/key,(sec|#num)<:time shift>)

定義された評価期間内のアイテムの平均値
 サポートされる値の型: *Float*、*Integer*
 サポートされるforeach 関数: *avg_foreach*、*count_foreach*、*exists_foreach*、*last_foreach*、*max_foreach*、*min_foreach*、*sum_foreach*

パラメーター: 共通パラメーター を参照してください。

時間シフトは、現在の平均値と少し前の平均値を比較する必要がある場合に便利です。

例:

```
avg(/host/key,1h) #過去1時間の平均値
avg(/host/key,1h:now-1d) #現在から25時間前から24時間前までの1時間の平均値
avg(/host/key,#5) #過去5つの値の平均値
avg(/host/key,#5:now-1d) #過去24時間に受信した値を除いた、過去5つの値の平均値
```

bucket_percentile(item filter,time period,percentage)

ヒストグラムのバケットからパーセンタイルを計算します。

パラメータ:

- **item filter** - item filterを参照してください。
- **time period** - time periodを参照してください。
- **percentage** - パーセンテージ (0-100)。

コメント:

- 計算アイテムでのみサポートされています。
- この関数は histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1)) のエイリアスです。

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

foreach 関数が返す配列内の値の数です。
 サポートされるforeach 関数: *avg_foreach*、*count_foreach*、*exists_foreach*、*last_foreach*、*max_foreach*、*min_foreach*、*sum_foreach*。

パラメータ:

- **func_foreach** - 返された値の数をカウントする対象の foreach 関数。foreach 関数を参照してください。count_foreach および bucket_rate_foreach は追加パラメータをサポートすることに注意してください。
- **item filter** - アイテムフィルターを参照してください。
- **time period** - 期間を参照してください。

- **operator**（二重引用符で囲む必要があります）。サポートされる operators:
eq - 等しい
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like - パターンを含む場合に一致（大文字・小文字を区別）
bitand - ビット単位 AND
regexp - pattern で指定された正規表現との大文字・小文字を区別する一致
iregexp - pattern で指定された正規表現との大文字・小文字を区別しない一致
- **pattern** - 必要なパターン（文字列引数は二重引用符で囲む必要があります）。3 番目のパラメータで *operator* が指定されている場合にサポートされます。

コメント:

- 履歴関連の foreach 関数（max_foreach、avg_foreach など）で **count()** を使用すると、パフォーマンスに影響する可能性があります。一方、設定データのみを扱う **exists_foreach()** を使用した場合、そのような影響はありません。
- オプションのパラメータ *operator* または *pattern* は、カンマの後に空のまま残すことはできず、完全に省略する必要があります。
- 3 番目のパラメータに *bitand* を指定した場合、4 番目の pattern パラメータは、'/' で区切られた 2 つの数値 **number_to_compare_with/mask** として指定できます。count() は値と *mask* の「ビット単位 AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビット単位 AND」の結果が *number_to_compare_with* と等しい場合、その値がカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定できます（'/' は不要）。
- 3 番目のパラメータに *regexp* または *iregexp* を指定した場合、4 番目の pattern パラメータには通常の正規表現またはグローバル（'@' で始まる）正規表現を指定できます。グローバル正規表現の場合、大文字・小文字の区別はグローバル正規表現の設定を継承します。regexp 一致の目的では、float 値は常に小数点以下 4 桁で表現されます。また、大きな数値では 10 進数表現（データベースに保存）と 2 進数表現（Zabbix サーバーで使用）の違いが小数点以下 4 桁目に影響する可能性があることにも注意してください。

例:

```
count(max_foreach(/*/net.if.in[*],1h)) #現在までの直近1時間にデータを受信したnet.if.inアイテムの数
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #ディスク使用率が95%を超えるファイルシステムの数
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

ヒストグラムのバケットから φ-quantile を計算します。
 サポートされているforeach 関数: *bucket_rate_foreach*。

パラメータ:

- **quantile** - 0 ≤ φ ≤ 1;
- **bucketN, valueN** - 手動で入力するパラメータのペア（2 組以上）、またはbucket_rate_foreach の応答。

コメント:

- 計算アイテムでのみサポートされています。
- 機能的には、PromQL の'histogram_quantile' に対応します。
- 最後の'Infinity' バケット（"+*inf*"）の値が 0 に等しい場合は -1 を返します。

例:

```
histogram_quantile(0.75,1.0,last(/host/rate_bucket[1.0]),"+Inf",last(/host/rate_bucket[Inf]))
histogram_quantile(0.5,bucket_rate_foreach(//item_key,30s))
```

item_count(item filter)

フィルター条件に一致する、設定内の既存のアイテム数です。
 サポートされる値の型: *Integer*。

パラメータ:

- **item filter** - アイテム選択の条件です。ホストグループ、ホスト、アイテムキー、タグによる参照が可能です。ワイルドカードをサポートしています。詳細はitem filterを参照してください。

コメント:

- 計算アイテムでのみサポートされます。
- *count(exists_foreach(item_filter))* 関数のエイリアスとして動作します。

例:

```
item_count(/*/agent.ping?[group="Host group 1"]) #「Host group 1」内で *agent.ping* アイテムを持つホスト数
```

kurtosis(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の確率分布の" 裾野度"。Kurtosis も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメータ: 共通パラメータ を参照してください。

例:

kurtosis(/host/key,1h) #現在までの過去1時間のkurtosis

mad(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の絶対偏差の中央値 絶対偏差の中央値も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメーター: 共通パラメーターを参照してください。

例:

mad(/host/key,1h) #過去1時間から現在までの絶対偏差の中央値

max(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内における項目の最高値
 サポートされる値の型: *Float*、*Integer*
 サポートされるforeach 関数: *avg_foreach*、*count_foreach*、*exists_foreach*、*last_foreach*、*max_foreach*、*min_foreach*、*sum_foreach*

パラメーター: 共通パラメーター を参照してください。

例:

max(/host/key,1h) - min(/host/key,1h) #過去1時間以内の最大値と最小値の差（値の差分）を計算します。

min(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内における項目の最小値
 サポートされる値の型: *Float*、*Integer*
 サポートされるforeach 関数: *avg_foreach*、*count_foreach*、*exists_foreach*、*last_foreach*、*max_foreach*、*min_foreach*、*sum_foreach*

パラメーター: 共通パラメーター を参照してください。

例:

max(/host/key,1h) - min(/host/key,1h) #過去1時間以内の最大値と最小値の差（値の差分）を計算します。

skewness(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の確率分布の非対称性。Skewness も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメーター: 共通パラメーター を参照してください。

例:

skewness(/host/key,1h) #過去1時間から現在までの歪度

stddevpop(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の母標準偏差。標準偏差も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメーター: 共通パラメーターを参照してください。

例:

stddevpop(/host/key,1h) #過去1時間から現在までの母標準偏差

stddevsamp(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の標本標準偏差。標準偏差も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメーター: 共通パラメーターを参照してください。

この関数が動作するには、少なくとも 2 つのデータ値が必要です。

例:

stddevsamp(/host/key,1h) #過去1時間から現在までの標本標準偏差

sum(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の合計
 サポートされる値の型: *Float*、*Integer*
 サポートされるforeach 関数: *avg_foreach*、*count_foreach*、*exists_foreach*、*last_foreach*、*max_foreach*、*min_foreach*、*sum_foreach*

パラメーター: 共通パラメーター を参照してください。

例:

sum(/host/key,1h) #過去1時間から現在までの値の合計

sumofsquares(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の平方和
 サポートされる値の型: *Float*、*Integer*
 サポートされるforeach 関数: *last_foreach*

パラメーター: 共通パラメーターを参照してください。

例:

```
sumofsquares(/host/key,1h) #過去1時間から現在までの平方和
```

varpop(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の母分散。分散も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメーター: 共通パラメーターを参照してください。

例:

```
varpop(/host/key,1h) #過去1時間から現在までの母分散
```

varsamp(/host/key,(sec|#num)<:timeshift>)

定義された評価期間内に収集された値の標本分散。分散も参照してください。
 サポートされている値の型: *Float*、*Integer*
 サポートされているforeach 関数: *last_foreach*

パラメーター: 共通パラメーターを参照してください。

この関数が動作するには、少なくとも 2 つのデータ値が必要です。

例:

```
varsamp(/host/key,1h) #過去1時間から現在までの標本分散
```

サポートされているすべての関数を参照してください。

1 Foreach 関数

概要

foreach 関数は、使用されるアイテムフィルターによって選択された各アイテムに対して 1 つの集計値を返すために、集計計算で使用されます。値の配列が返されます。

> **Attention:**
> foreach 関数は、集計計算式の一部として、計算アイテム内でのみサポートされています。このコンテキスト外では、通常のアイテムキーやトリガー式で呼び出すことはできません。

たとえば、*avg_foreach* 関数は値の配列を返します。各値は、指定された時間間隔における、選択されたアイテムの履歴値の平均です。

item filter は、foreach 関数で使用される構文の一部です。アイテムフィルターではワイルドカードの使用がサポートされているため、必要なアイテムを非常に柔軟に選択できます。

サポートされている関数

Function	Description
avg_foreach	各アイテムの平均値を返します。
bucket_rate_foreach	histogram_quantile() 関数での使用に適した (バケットの上限値、レート値) のペアを返します。ここで「バケットの上限値」は、<parameter number> parameter で定義されたアイテムキーのパラメータの値です。
count_foreach	各アイテムの値の数を返します。
exists_foreach	有効な各アイテムに対して'1' を返します。
last_foreach	各アイテムの最新の値を返します。
max_foreach	各アイテムの最大値を返します。
min_foreach	各アイテムの最小値を返します。
sum_foreach	各アイテムの値の合計を返します。

関数の構文

foreach 関数は、2 つの共通パラメータ `item filter`（詳細は以下を参照）と `time period` をサポートします。

foreach_function(item filter,time period)

例:

avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)

は、MySQL サーバーグループ内の各'mysql.qps' アイテムの 5 分間の平均値を返します。

一部の関数は追加のパラメータもサポートすることに注意してください。

アイテムフィルター構文

アイテムフィルター:

/host/key[parameters]?[conditions]

は 4 つの部分で構成されます。

- host - ホスト名
- key - アイテムキー（パラメータを除く）
- parameters - アイテムキーのパラメータ
- conditions - ホストグループおよび/またはアイテムタグに基づく条件（式として指定）

スペースを使用できるのは、conditions 式の内部のみです。

ワイルドカードの使用

- ワイルドカードは、ホスト名、アイテムキー、または個々のアイテムキーパラメータの置き換えに使用できます。
- host または item key のいずれか一方は、ワイルドカードなしで指定する必要があります。したがって、/host/* と /*/key は有効なフィルターですが、/*/* は無効です。
- ワイルドカードは、ホスト名、アイテムキー、アイテムキーパラメータの一部に対して使用することはできません。
- ワイルドカードは、複数のアイテムキーパラメータには一致しません。したがって、各パラメータごとに個別にワイルドカードを指定する必要があります（例: key[abc,*,*]）。

条件式

条件式では、以下をサポートしています。

- オペランド:
  - group - ホストグループ
  - tag - アイテムタグ
  - "<text>" - 文字列定数。" および \ をエスケープするためのエスケープ文字 \ を使用します
- 大文字と小文字を区別する文字列比較演算子: =, <>
- 論理演算子: and, or, not
- 括弧によるグループ化: ( )

文字列定数の引用符は必須です。サポートされるのは、大文字と小文字を区別する完全一致の文字列比較のみです。

Warning:
フィルターでタグを指定する場合（例: tag="tagname:value"）、コロン":" は区切り文字として使用されます。
その後ろのすべてがタグ値と見なされます。そのため、現時点では":" を含むタグ名を指定することはサポートされていません。

例

以下の例のように、アイテムキー、ホストグループ、タグを参照する複雑なフィルターを使用できます。

Syntax example	Description
/host/key[abc,*]	このホスト上の類似アイテムに一致します。
/*/key	任意のホストの同じアイテムに一致します。
/*/key?[group="ABC" and tag="tagname:value"]	ABC グループに属し、'tagname:value' タグを持つ任意のホストの同じアイテムに一致します。
/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]	ABC または DEF グループに属し、それぞれのタグを持つ任意のホストの類似アイテムに一致します。

参照されるすべてのアイテムは、存在し、データを収集している必要があります。
計算に含まれるのは、有効なホスト上の有効なアイテムのみです。
未サポート状態のアイテムは含まれません。

Attention:
参照されるアイテムのアイテムキーが変更された場合は、フィルターを手動で更新する必要があります。

親ホストグループを指定すると、その親グループと、配下にネストされたすべてのホストグループおよびそれらのアイテムが含まれます。

期間

2 番目のパラメータでは、集計の期間を指定できます。期間は時間としてのみ指定でき、値の数（**#** を接頭辞として付ける形式）はサポートされていません。

利便性のため、このパラメータではサポートされている単位記号を使用できます。例えば、'300s'（300 秒）の代わりに'5m'（5 分）、または'86400'（86400 秒）の代わりに'1d'（1 日）を使用できます。

last_foreach 関数では、期間は省略可能なパラメータです（Zabbix 7.0 以降でサポート）。

```
last_foreach(/*/key?[group="host group"])
```

exists_foreach 関数では期間はサポートされていません。

追加パラメータ

bucket_rate_foreach

bucket_rate_foreach 関数では、3 つ目のオプションパラメータがサポートされています。

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

ここで、<parameter number> はアイテムキー内の"bucket" 値の位置です。例えば、myItem[aaa,0.2] の"bucket" 値が'0.2' の場合、その位置は 2 です。

<parameter number> のデフォルト値は'1' です。

count_foreach

count_foreach 関数では、3 つ目および 4 つ目のオプションパラメータがサポートされています。

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

ここで:

- **operator** はアイテム値に対する条件演算子です（必ずダブルクォートで囲む必要があります）。サポートされる operators:
eq - 等しい
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like - パターンを含む場合に一致（大文字・小文字を区別）
bitand - ビット単位 AND
regexp - pattern で指定された正規表現との大文字・小文字を区別する一致
iregexp - pattern で指定された正規表現との大文字・小文字を区別しない一致
- **pattern** は必要なパターンです（文字列引数はダブルクォートで囲む必要があります）。3 つ目のパラメータで *operator* が指定されている場合にサポートされます。

コメント:

- オプションパラメータ *operator* または *pattern* は、カンマの後に空のまま残すことはできず、省略する場合は完全に省略する必要があります。
- 3 つ目のパラメータとして *bitand* を使用する場合、4 つ目の pattern パラメータは'/' で区切られた 2 つの数値、**number_to_compare_with/mask** として指定できます。count_foreach() は値と *mask* の「ビット単位 AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビット単位 AND」の結果が *number_to_compare_with* と等しい場合、その値がカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定すればよく（'/' は不要です）。
- 3 つ目のパラメータとして *regexp* または *iregexp* を使用する場合、4 つ目の pattern パラメータには通常の正規表現またはグローバル（'@' で始まる）正規表現を指定できます。グローバル正規表現の場合、大文字・小文字の区別はグローバル正規表現の設定を継承します。regexp マッチングのために、浮動小数点値は常に'.' の後に小数点以下 4 桁で表現されます。また、大きな数値では 10 進表現（データベースに保存される値）と 2 進表現（Zabbix server で使用される値）の違いが、小数点以下 4 桁目に影響する可能性がある点にも注意してください。

foreach 関数の使用方法に関する詳細および例については、集計計算 を参照してください。

空き状況に応じた動作

次の表は、ホスト/アイテムおよびヒストリデータの可用性が制限されている場合に各関数がどのように動作するかを示しています。

関数	無効なホスト	データのある利用不可能なホスト	データのない利用不可能なホスト	無効な項目	取得不可のアイ
avg_foreach	無視	平均値	無視	無視	無視
bucket_rate_foreach	無視	バケット率	無視	無視	無視
count_foreach	無視	カウント数	0	無視	無視

関数	無効なホスト	データのある利用不可能なホスト	データのない利用不可能なホスト	無効な項目	取得不可のアイ
exists_foreach	無視	1	1	無視	1
last_foreach	無視	最新値	無視	無視	無視
max_foreach	無視	最大値	無視	無視	無視
min_foreach	無視	最小値	無視	無視	無視
sum_foreach	無視	合計値	無視	無視	無視

アイテムが無視された場合、集計には何も追加されません。

2 Bitwise 関数

ここにリストされているすべての関数は、次でサポートされています。

- トリガー式
- 計算アイテム

関数は追加情報なしでリストされています。関数をクリックして詳細を確認してください。

関数	Description
bitand	アイテム値とマスクの” ビット論理積” の値。
bitlshift	アイテム値のビット単位左シフト。
bitnot	アイテム値の” ビット否定” の値。
bitor	アイテム値とマスクの” ビット論理和” の値。
bitrshift	アイテム値のビット単位右シフト。
bitxor	アイテム値とマスクの” ビット排他的論理和” の値。

関数の詳細

関数のパラメータに関する一般的な注意事項:

- 関数のパラメータはカンマで区切ります。
- 式もパラメータとして使用できます。
- オプションの関数パラメータ（またはパラメータ部分）は、< > で示されます。

bitand(value,mask)

項目の値とマスクとの” ビット単位の AND” の結果
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値
- **mask** (必須) - 64 ビット符号なし整数 (0 - 18446744073709551615)

比較はビット単位で行われますが、すべての値を指定する必要があり、結果は 10 進数で返されます。例えば、3 番目のビットのチェックは、100 ではなく 4 と比較することで行われます。

例:

`bitand(last(/host/key),12)=8` または `bitand(last(/host/key),12)=4` #3番目または4番目のビットが設定されてい
`bitand(last(/host/key),20)=16` #3番目のビットが設定されておらず、5番目のビットが設定されています。

bitlshift(値, シフトするビット数)

項目の値をビット単位で左にシフトします。
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値
- **bits to shift** (必須) - シフトするビット数

比較はビット単位で行われますが、すべての値を指定する必要があり、結果は 10 進数で返されます。例えば、3 番目のビットのチェックは 100 ではなく 4 と比較することで行われます。

bitnot(value)

項目値の” ビット単位の NOT” の結果
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値

比較はビット単位で行われますが、すべての値を指定する必要があり、結果は 10 進数で返されます。例えば、3 番目のビットのチェックは 100 ではなく 4 と比較することで行われます。

bitor(value,mask)

項目の値とマスクの” ビット単位の OR” 演算の結果
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビット単位で行われますが、すべての値を指定する必要があり、結果は 10 進数で返されます。例えば、3 番目のビットのチェックは 100 ではなく 4 と比較することで行われます。

bitrshift(値, シフトするビット数)

項目の値をビット単位で右にシフト
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値
- **bits to shift** (必須) - シフトするビット数

比較はビット単位で行われますが、すべての値を指定する必要があり、結果は 10 進数で返されます。例えば、3 番目のビットのチェックは 100 ではなく 4 と比較することで行われます。

bitxor(value,mask)

項目の値とマスクの” ビット単位の排他的論理和” の値
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値
- **mask** (必須) - 64 ビットの符号なし整数 (0 - 18446744073709551615)

比較はビット単位で行われますが、すべての値を指定する必要があり、結果は 10 進数で返されます。例えば、3 番目のビットのチェックは 100 ではなく 4 と比較することで行われます。

サポートされているすべての関数を参照してください。

3 日付と時刻の関数

ここに記載されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテム
- 式マクロ

Attention:
日付および時刻の関数は、それ単体では式内で使用できません。式には、ホストのアイテムを参照する別のグループの関数を少なくとも 1 つ含める必要があります（nodata() 関数を除く）。日付および時刻の関数が式内でどのように動作するかの詳細については、計算時刻を参照してください。

関数は追加情報なしで一覧表示されています。詳細を確認するには、関数をクリックしてください。

Function	Description
date	YYYYMMDD 形式の現在の日付。
dayofmonth	1～31 の範囲の月の日。
dayofweek	1～7 の範囲の曜日。
now	エポック（1970 年 1 月 1 日 00:00:00 UTC）からの経過秒数。
time	HHMMSS 形式の現在の時刻。

関数の詳細

date

YYYYMMDD 形式の現在の日付

例:

```
date()<20220101
```

dayofmonth

1 から 31 までの範囲の月内の日

例:

```
dayofmonth()=1
```

dayofweek

曜日を 1 から 7 の範囲で指定します（月曜：1、日曜：7）

例（平日のみ）:

```
dayofweek()<6
```

例（週末のみ）:

```
dayofweek()>5
```

now

エポック（1970 年 1 月 1 日 00:00:00 UTC）からの秒数

例:

```
now()<1640998800
```

time

HHMMSS 形式で表した現在の時刻

例 (夜間のみ、00:00～06:00):

```
time()<060000
```

サポートされているすべての関数を参照してください。

4 履歴関数

ここに記載されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしで一覧表示されています。
関数をクリックすると、詳細を確認できます。

Function	Description
change	直前の値と最新の値の差分量。
changecount	定義された評価期間内における隣接する値の間の変化回数。
count	定義された評価期間内の値の数。
countunique	定義された評価期間内の一意な値の数。
find	定義された評価期間内で値の一致を検索。
first	定義された評価期間内の最初の（最も古い）値。
firstclock	定義された評価期間内の最初の（最も古い）値のタイムスタンプ。
fuzzytime	パッシブエージェントの時刻が Zabbix サーバー/プロキシの時刻とどの程度異なるかを確認。
last	最新の値。
lastclock	定義された評価期間内の新しい順で N 番目の値のタイムスタンプ。
logeventid	最後のログエントリのイベント ID が正規表現に一致するかを確認。
logseverity	最後のログエントリのログ重要度。
logsource	最後のログエントリのログソースが正規表現に一致するかを確認。
logtimestamp	新しい順で N 番目のログアイテム値のログメッセージタイムスタンプ。
monodec	値が単調減少しているかを確認。
monoinc	値が単調増加しているかを確認。
nodata	データを受信していないことを確認。
percentile	期間の P パーセンタイル値。P（パーセンテージ）は 3 番目のパラメータで指定されます。
rate	定義された期間内で単調増加するカウンターの増加量について、1 秒あたりの平均レート。

共通パラメータ

- /host/key は、ホストのアイテム履歴を参照する関数に共通する必須の第 1 パラメータです
- (sec|#num)<:time shift> は、ホストのアイテム履歴を参照する関数に共通する第 2 パラメータで、以下を意味します。
  - **sec** - 秒単位の最大評価期間（時間のサフィックスを使用できます）、または
  - **#num** - 直近に収集された値における最大評価範囲（ハッシュ記号が前に付く場合）
  - **time shift**（任意）- 評価時点を過去に移動できます。time shift の指定方法の詳細は、こちらを参照してください。

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます
- 関数固有のパラメータは各関数ごとに説明されています
- /host/key および (sec|#num)<:time shift> パラメータは、引用符で囲んではいけません

change(/host/key)

直前の値と最新の値の差分量です。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。
 文字列の場合の戻り値: 0 - 値が等しい; 1 - 値が異なる。

パラメータ: 共通パラメータを参照してください。

コメント:

- 数値の差分は、以下の受信した値の例のように計算されます（「直前」の値と「最新」の値 = 差分）:
'1' と '5' = +4
'3' と '1' = −2
'0' と '-2.5' = −2.5。
- 比較については、absも参照してください。

例:

```
change(/host/key)>10
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

定義された評価期間内における、隣接する値の間の変化回数。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照してください
- **mode**（二重引用符で囲む必要があります）- 指定可能な値: *all* - すべての変化をカウント（デフォルト）; *dec* - 減少をカウント; *inc* - 増加をカウント

数値以外の値の型では、*mode* パラメータは無視されます。

例:

```
changecount(/host/key,1w) #the number of value changes for the last week until now
changecount(/host/key,#10,"inc") #the number of value increases (relative to the adjacent value) among the
changecount(/host/key,24h,"dec") #the number of value decreases (relative to the adjacent value) for the l
```

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内の値の数です。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照してください
- **operator**（二重引用符で囲む必要があります）サポートされる operators:
eq - 等しい（integer、float のデフォルト）
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like（string、text、log のデフォルト）- パターンを含む場合に一致（大文字・小文字を区別）
bitand - ビット単位 AND
regexp - pattern で指定された正規表現との大文字・小文字を区別する一致
iregexp - pattern で指定された正規表現との大文字・小文字を区別しない一致
- **pattern** - 必要なパターン（文字列引数は二重引用符で囲む必要があります）

コメント:

- Float アイテムは 2.22e-16 の精度で一致判定されます。
- *like* は integer 値の operator としてはサポートされていません。
- *like* と *bitand* は float 値の operator としてはサポートされていません。
- string、text、log 値では、*eq*、*ne*、*like*、*regexp*、*iregexp* の operator のみがサポートされます。

- operator として *bitand* を使用する場合、4 番目の pattern パラメータは'/' で区切られた 2 つの数値として指定できます: **number_to_compare_with/mask**。count() は値と *mask* の「ビット単位 AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビット単位 AND」の結果が *number_to_compare_with* と等しい場合、その値はカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定できます（'/' なし）。
- operator として *regexp* または *iregexp* を使用する場合、4 番目の pattern パラメータには通常の正規表現またはグローバル（'@' で始まる）正規表現を指定できます。グローバル正規表現の場合、大文字・小文字の区別はグローバル正規表現の設定を継承します。regexp 照合のために、float 値は常に'.' の後に小数点以下 4 桁で表現されます。また、大きな数値では 10 進表現（データベースに保存）と 2 進表現（Zabbix サーバーで使用）の違いにより、小数点以下 4 桁目に影響する場合があることにも注意してください。

例:

```
count(/host/key,10m) # 現在までの直近 10 分間の値の数
count(/host/key,10m,"like","error") # 現在までの直近 10 分間で 'error' を含む値の数
count(/host/key,10m,,12) # 現在までの直近 10 分間で '12' に等しい値の数
count(/host/key,10m,"gt",12) # 現在までの直近 10 分間で '12' を超える値の数
count(/host/key,#10,"gt",12) # 現在までの直近 10 個の値のうち '12' を超える値の数
count(/host/key,10m:now-1d,"gt",12) # 現在から 24 時間前の時点を基準に、その 24 時間前から 10 分後までの間で '12' を超
count(/host/key,10m,"bitand","6/7") # 現在までの直近 10 分間で、下位 3 ビットが '110'（2 進数）である値の数
count(/host/key,10m:now-1d) # 現在から 24 時間前の時点を基準に、その 24 時間前から 10 分後までの間の値の数
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内の一意な値の数です。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照してください
- **operator**（二重引用符で囲む必要があります）。サポートされる operator:
eq - 等しい（integer、float のデフォルト）
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like（string、text、log のデフォルト）- パターンを含む場合に一致（大文字・小文字を区別）
bitand - ビット単位 AND
regexp - pattern で指定された正規表現に大文字・小文字を区別して一致
iregexp - pattern で指定された正規表現に大文字・小文字を区別せず一致
- **pattern** - 必要なパターン（文字列引数は二重引用符で囲む必要があります）

コメント:

- Float アイテムは 2.22e-16 の精度で一致します。
- integer 値では、*like* は operator としてサポートされません。
- float 値では、*like* および *bitand* は operator としてサポートされません。
- string、text、log 値では、*eq*、*ne*、*like*、*regexp*、*iregexp* の各 operator のみがサポートされます。
- operator に *bitand* を使用する場合、4 番目の pattern パラメータは 比較値/マスクのように、'/' で区切られた 2 つの数値として指定できます。countunique() は値と *mask* の「ビット単位 AND」を計算し、その結果を *number_to_compare_with* と比較します。「ビット単位 AND」の結果が *number_to_compare_with* と等しい場合、その値はカウントされます。
number_to_compare_with と *mask* が等しい場合は、*mask* のみを指定できます（'/' は不要）。
- operator に *regexp* または *iregexp* を使用する場合、4 番目の pattern パラメータには通常の正規表現、またはグローバル（'@' で始まる）正規表現を指定できます。グローバル正規表現の場合、大文字・小文字の区別はグローバル正規表現の設定から継承されます。正規表現マッチングのために、float 値は常に'.' の後に小数点以下 4 桁で表現されます。また、大きな数値では 10 進表現（データベースに保存される値）と 2 進表現（Zabbix サーバーで使用される値）の違いが、小数点以下 4 桁目に影響する可能性があることにも注意してください。

例:

```
countunique(/host/key,10m) # 現時点までの直近 10 分間の一意な値の数
countunique(/host/key,10m,"like","error") # 現時点までの直近 10 分間で 'error' を含む一意な値の数
countunique(/host/key,10m,,12) # 現時点までの直近 10 分間で '12' に等しい一意な値の数
countunique(/host/key,10m,"gt",12) # 現時点までの直近 10 分間で '12' より大きい一意な値の数
countunique(/host/key,#10,"gt",12) # 現時点までの直近 10 個の値の範囲内で '12' より大きい一意な値の数
countunique(/host/key,10m:now-1d,"gt",12) # 現時点から 24 時間前の時点を基準に、その 10 分前から 24 時間前までの間で
countunique(/host/key,10m,"bitand","6/7") # 現時点までの直近 10 分間で、下位 3 ビットに '110'（2 進数）を持つ一意な値
countunique(/host/key,10m:now-1d) # 現時点から 24 時間前の時点を基準に、その 10 分前から 24 時間前までの間の一意な値の数
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

定義された評価期間内で一致する値を検索します。
 サポートされる値の型: *Float*、*Integer*、*String*、*Text*、*Log*。
 戻り値: 1 - 見つかった場合、0 - それ以外。

パラメータ:

- 共通パラメータを参照
- **sec** または **#num** (任意) - 指定しない場合は最新の値がデフォルトで使用されます
- **operator** (二重引用符で囲む必要があります) サポートされる operator:
eq - 等しい (integer、float のデフォルト)
ne - 等しくない
gt - より大きい
ge - 以上
lt - より小さい
le - 以下
like (string、text、log のデフォルト) - pattern で指定された文字列を含む場合に一致します (大文字と小文字を区別)
bitand - ビット単位 AND
regexp - pattern で指定された正規表現との大文字と小文字を区別する一致
iregexp - pattern で指定された正規表現との大文字と小文字を区別しない一致
- **pattern** - 必要なパターン (文字列引数は二重引用符で囲む必要があります); operator が *regexp*、*iregexp* の場合は Perl Compatible Regular Expression (PCRE) 正規表現

コメント:

- 複数の値が処理される場合、一致する値が少なくとも 1 つあれば'1' が返されます。
- integer 値では *like* は operator としてサポートされません。
- float 値では *like* と *bitand* は operator としてサポートされません。
- string、text、log 値では *eq*、*ne*、*like*、*regexp*、*iregexp* の各 operator のみがサポートされます。
- operator として *regexp* または *iregexp* を使用する場合、4 番目の pattern パラメータには通常の正規表現またはグローバル (先頭が'@') 正規表現を指定できます。グローバル正規表現の場合、大文字と小文字の区別はグローバル正規表現の設定から継承されます。

例:

```
find(/host/key,10m,"like","error") # 直近 10 分間から現在までの間に 'error' を含む値を検索
```

first(/host/key,sec<:time shift>)

定義された評価期間内の最初の（最も古い）値です。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照してください

last()も参照してください。

例:

```
first(/host/key,1h) #retrieve the oldest value within the last hour until now
```

firstclock(/host/key,sec<:timeshift>)

定義された評価期間内で最も古い値のタイムスタンプ
 サポートされる値の型: *Float*、*Integer*、*String*、*Text*、*Log*

パラメーター:

- 共通パラメーター を参照してください。

指定された期間内にデータが収集されていない場合、この関数はエラーで失敗します。

lastclock() も参照してください。

例:

```
firstclock(/host/key,1h) #過去1時間以内で最も古い値のタイムスタンプを取得します。
firstclock(/host/key,1h:now-24h) #1日前の過去1時間以内で最も古い値のタイムスタンプを取得します。
```

fuzzytime(/host/key,sec)

パッシブエージェントの時刻が、Zabbix サーバー/プロキシの時刻とどの程度異なるかを確認します。
 サポートされる値の型: *Float*, *Integer*。
 戻り値: 1 - パッシブアイテムの値（タイムスタンプとして扱われる）と Zabbix サーバー/プロキシのタイムスタンプ（値収集時の時刻）の差が *sec* 秒以下の場合。0 - それ以外。

パラメータ:

- 共通パラメータを参照してください。

コメント:

- 通常、ローカル時刻が Zabbix サーバーのローカル時刻と同期していることを確認するために、'system.localtime' アイテムとともに使用されます。注:'system.localtime' は、Zabbix エージェントではパッシブチェック として設定する必要があります。Zabbix エージェント 2 では、アクティブチェックとして設定することもできます。
- vfs.file.time[/path/file,modify] キーとともに使用して、ファイルが長時間更新されていないことを確認することもできます。
- この関数は、複雑なトリガー条件式 (複数のアイテムが関係するもの) での使用は推奨されません。予期しない結果を引き起こす可能性があるためです (時刻差は最新のメトリクスで測定されます)。例: fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0

例:

```
fuzzytime(/host/key,60s)=0 # 時刻差が 60 秒を超えた場合に問題を検出<br><br>
```

last(/host/key,<#num<:time shift»)

最新の値。
 サポートされる値の型: *Float*, *Integer*, *String*, *Text*, *Log*。

パラメータ:

- 共通パラメータを参照
- **#num** (任意) - N 番目に新しい値

コメント:

- ハッシュタグ付きの期間 (#N) は、ここでは他の多くの関数とは異なる動作をすることに注意してください。例えば、`last(/host/key)` は常に `last(/host/key,#1)` と等しくなります。`last(/host/key,#3)` は 3 番目に新しい値です (直近 3 つの値ではありません)。
- Zabbix は、履歴内で 1 秒以内に 2 つを超える値が存在する場合、値の正確な順序を保証しません。
- first() も参照してください。

例:

```
last(/host/key) # 最後の値を取得
last(/host/key,#2) #1 つ前の値を取得
last(/host/key,#1) <> last(/host/key,#2) # 最後の値と 1 つ前の値が異なる
```

lastclock(/host/key,<#num<:timeshift»)

定義された評価期間における N 番目に最新の値のタイムスタンプ
 サポートされる値の型: *Float*、*Integer*、*String*、*Text*、*Log*

パラメーター:

- 共通パラメーター を参照してください。
- **#num** (オプション) - N 番目に最新の値

指定された期間にデータが収集されていない場合、または N 番目の値が収集されていない場合、関数はエラーで失敗します。

firstclock() も参照してください。

例:

```
lastclock(/host/key) # 最新の値のタイムスタンプを取得
lastclock(/host/key,#2) # 2番目に新しい値のタイムスタンプを取得
lastclock(/host/key,#3:now-1d) # 1日前の3番目に新しい値のタイムスタンプを取得
```

logeventid(/host/key,<#num<:timeshift»,<pattern>)

最後のログエントリのイベント ID が正規表現と一致するかどうかを確認します。
 サポートされる値の型: *Log*
 戻り値: 0 - 一致しない; 1 - 一致する

パラメーター:

- 共通パラメーター を参照してください。
- **#num** (オプション) - N 番目に最近の値
- **pattern** (オプション) - 必要なパターンを記述する正規表現Perl 互換正規表現 (PCRE) 形式（文字列引数は二重引用符で囲む必要があります）

logseverity(/host/key,<#num<:timeshift»)

最後のログエントリのログ重要度
 サポートされる値の型: *Log*
 戻り値: 0 - デフォルトの重要度、N - 重要度（整数、Windows イベントログで有効：1 - 情報、2 - 警告、4 - エラー、7 - 監査失敗、8 - 監査成功、9 - 重大、10 - 詳細）

パラメーター:

- 共通パラメーター を参照してください。
- **#num** (オプション) - N 番目に新しい値。

Zabbix は、Windows イベントログの **Information** フィールドからログ重要度を取得します。

logsource(/host/key,<#num<:time shift»,<pattern>)

直近のログエントリのログソースが正規表現に一致するかどうかを確認します。
 サポートされる値の型: *Log*。
 戻り値: 0 - 一致しない; 1 - 一致する。

パラメータ:

- 共通パラメータを参照してください
- **#num** (任意) - 新しい順で N 番目の値
- **pattern** (任意) - 必要なパターンを記述する正規表現、Perl Compatible Regular Expression (PCRE) 形式（文字列引数は二重引用符で囲む必要があります）

通常は Windows イベントログで使用されます。

例:

```
logsource(/host/key,,"VMware Server")
```

logtimestamp(/host/key,<#num<:timeshift»)

N 番目に最新のログ項目値のログメッセージのタイムスタンプ
 サポートされる値の型: *Log*

パラメータ:

- 共通パラメータ を参照してください。
- **#num** (オプション) - N 番目に最新の値

コメント:

- タイムシフトの計算は、ログメッセージのタイムスタンプではなく、項目値のクロックに基づきます。
- 以下の場合、関数はエラーで失敗します。
- ログタイプではない項目を受信した場合。
- 指定された期間内にデータが収集されていない場合。
- N 番目の値が収集されていない場合。
- ログメッセージにタイムスタンプ値が含まれていない場合。

例:

```
logtimestamp(/host/key) # 最新のログメッセージのタイムスタンプを取得
logtimestamp(/host/key,#2) # 2番目に新しいログメッセージのタイムスタンプを取得
logtimestamp(/host/key,#3:now-1d) # 1日前の3番目に新しいログメッセージのタイムスタンプを取得
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

値が単調に減少しているかどうかを確認します。
 サポートされる値の型: *Integer*。
 戻り値: 1 - 指定期間内のすべての要素が継続的に減少している場合; 0 - それ以外。

パラメータ:

- 共通パラメータを参照してください
- **mode** (二重引用符で囲む必要があります) - *weak* (各値が直前の値以下である; デフォルト) または *strict* (各値が減少している)

例:

```
monodec(/Host1/system.swap.size[all,free],60s) + monodec(/Host2/system.swap.size[all,free],60s) + monodec(
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

値が単調増加しているかどうかを確認します。
 サポートされる値の型: 整数。
 戻り値: 1 - 期間内のすべての要素が継続的に増加している場合。0 - それ以外。

パラメータ:

- 共通パラメータを参照してください
- **mode**（二重引用符で囲む必要があります）- *weak*（各値が直前の値以上である; デフォルト）または *strict*（各値が増加している）

例:

```
monoinc(/Host1/system.localtime,#3,"strict")=0 # システムのローカル時刻が一貫して増加しているかどうかを確認
```

nodata(/host/key,sec,<mode>)

データを受信していないことを確認します。
 サポートされる値の型: *Integer*, *Float*, *Character*, *Text*, *Log*。
 戻り値: 定義された期間中にデータを受信していない場合は 1、それ以外は 0。

パラメータ:

- 共通パラメータを参照してください
- **sec** - history syncer プロセスはこの関数を 30 秒ごとにしか計算しないため、期間は 30 秒未満にできません。`nodata(/host/key,0)` は許可されていません
- **mode** - *strict*（二重引用符付き）に設定すると、この関数はプロキシの可用性の影響を受けなくなります（詳細はコメントを参照）

コメント:

- プロキシによって監視される'nodata' トリガーは、デフォルトでプロキシの可用性の影響を受けます。プロキシが利用不可になった場合、接続が復旧した直後に'nodata' トリガーが発火することはなく、遅延した期間のデータはスキップされます。パッシブプロキシでは、接続が 2 秒以上 15 秒超遅れて復旧した場合に抑制が有効になります。アクティブプロキシでは、接続が 15 秒超遅れて復旧した場合に抑制が有効になります。プロキシの可用性の影響を無効にするには、第 3 パラメータを使用します。例: `nodata(/host/key,5m,"strict")`。この場合、データがない評価期間（5 分）が経過するとすぐにこの関数は発火します。
- この関数は、第 1 パラメータの期間内に次の場合エラーを表示します:
- データがなく、Zabbix サーバーが再起動された場合
- データがなく、メンテナンスが完了した場合
- データがなく、アイテムが追加または再度有効化された場合。
- エラーは、トリガーの設定の *Info* 列に表示されます。
- Zabbix サーバー、プロキシ、エージェントの間に時刻のずれがある場合、この関数は正しく動作しない可能性があります。関連項目: 時刻同期の要件。

percentile(/host/key,(sec|#num)<:timeshift>,percentage)

期間の P 番目のパーセンタイル。P（パーセンテージ）は 3 番目のパラメータで指定します。
 サポートされる値の型 : *Float*、*Integer*

パラメータ：

- 共通パラメータを参照してください。
- **percentage** - 0 から 100（両端値を含む）までの小数点以下 4 桁までの浮動小数点数

rate(/host/key,sec<:time shift>)

定義された期間内における、単調増加するカウンターの増加量の 1 秒あたりの平均レートです。
 サポートされる値の型: *Float*, *Integer*。

パラメータ:

- 共通パラメータを参照してください

機能的には、PromQL の'rate' に対応します。

例:

```
rate(/host/key,30s) #30 秒間の単調増加量が 20 の場合、この関数は 0.67 を返します。
```

サポートされているすべての関数を参照してください。

5 トレンド関数

トレンド関数は、履歴関数とは対照的に、計算にトレンドデータを使用します。

トレンドには、1 時間ごとの集計値が保存されます。トレンド関数はこれらの 1 時間平均を使用するため、長期的な分析に役立ちます。

トレンド関数の結果はキャッシュされるため、同じパラメータで同じ関数を複数回呼び出しても、データベースから取得される情報は 1 回だけです。トレンド関数キャッシュは、TrendFunctionCacheSize サーバーパラメータによって制御されます。

トレンド関数のみを参照するトリガーは、式内の最小の期間ごとに 1 回評価されます。たとえば、次のようなトリガーは

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

1 日に 1 回評価されます。トリガーにトレンド関数と履歴関数（または日付と時刻および/またはnodata()）の両方が含まれている場合は、通常の原則に従って計算されます。

ここに記載されているすべての関数は、以下でサポートされています。

- トリガー式
- 計算アイテム
- 式マクロ

関数は追加情報なしで一覧表示されています。関数をクリックすると、詳細を確認できます。

Function	Description
baselinedev	直近のデータ期間と、先行するシーズン内の同じデータ期間との間の偏差数（stddevpop アルゴリズムによる）を返します。
baselinewma	加重移動平均アルゴリズムを使用して、複数の等しい期間（「シーズン」）における同じ時間枠のデータを平均化し、ベースラインを計算します。

Function	Description
trendavg	定義された期間内のトレンド値の平均。
trendcount	定義された期間内のトレンド値の計算に使用された、正常に取得された履歴値の数。
trendmax	定義された期間内のトレンド値の最大値。
trendmin	定義された期間内のトレンド値の最小値。
trendstl	検出期間中の異常の割合を返します。0 から 1 の間の 10 進値で、((異常値の数)/(値の総数)) です。
trendsum	定義された期間内のトレンド値の合計。

共通パラメータ

- /host/key は共通の必須の第 1 パラメータです
- time period:time shift は共通の第 2 パラメータで、以下を意味します:
  - **time period** - 期間（最小値は「1h」）。<N><time unit> の形式で定義します。ここで、N は時間単位の数、time unit は h（時間）、d（日）、w（週）、M（月）、または y（年）です。
  - **time shift** - 期間のオフセット（関数の例を参照）

関数の詳細

関数パラメータに関する一般的な注意事項:

- 関数パラメータはカンマで区切ります
- オプションの関数パラメータ（またはパラメータの一部）は < > で示されます
- 関数固有のパラメータは各関数ごとに説明されています
- /host/key および time period:time shift パラメータは、決して引用符で囲んではいけません

baselinedev(/host/key,data period:time shift,season unit,num seasons)

直近のデータ期間と、先行する各シーズンにおける同じデータ期間との間の偏差数を、stddevpop アルゴリズムによって返します。

パラメータ:

- 共通パラメータを参照してください
- **data period** - シーズン内のデータ収集期間で、<N><time unit> として定義されます。ここで:
N - 時間単位の数
time unit - h（時間）、d（日）、w（週）、M（月）、y（年）のいずれかで、season 以下でなければなりません
- **season unit** - 1 つのシーズンを定義するカレンダー単位（h、d、w、M、y）。data period より小さくすることはできません
- **num seasons** - 評価するシーズン数

例:

```
baselinedev(/host/key,1d:now/d,"M",6) # 前日と、過去 6 か月における同じ日との間の標準偏差（母集団）の数を計算します。前月
baselinedev(/host/key,1h:now/h,"d",10)  # 前の 1 時間と、直前の 10 日間における同じ時刻との間の母集団標準偏差の数を計算
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

加重移動平均アルゴリズムを使用して、複数の等しい時間期間（「シーズン」）における同じ時間枠のデータを平均化することで、ベースラインを計算します。

パラメータ:

- 共通パラメータを参照してください
- **data period** - シーズン内のデータ収集期間で、<N><time unit> として定義されます。ここで:
N - 時間単位の数
time unit - h（時間）、d（日）、w（週）、M（月）または y（年）で、season 以下でなければなりません
Time shift - 時間期間のオフセットで、シーズン単位でのデータ収集時間枠の終了位置を定義します（例を参照）
- **season unit** - 1 つのシーズンを定義するカレンダー単位（h、d、w、M、y）。data period より小さくすることはできません
- **num seasons** - 評価するシーズン数

例:

```
baselinewma(/host/key,1h:now/h,"d",3) # 昨日で終了する直近 3 つの完全な日について、同じ時刻帯のベースラインを計算します。
baselinewma(/host/key,2h:now/h,"d",3) # 昨日で終了する直近 3 つの完全な日について、同じ 2 時間枠のベースラインを計算しま
baselinewma(/host/key,1d:now/d,"M",4) # 直前の完全な月より前の 4 か月について、'yesterday' と同じ月内日付のベースライ
```

trendavg(/host/key,time period:time shift)

定義された期間内のトレンド値の平均です。

パラメータ:

- 共通パラメータを参照してください

例:

```
trendavg(/host/key,1h:now/h) # 直前の 1 時間の平均（例: 12:00-13:00）
trendavg(/host/key,1h:now/h-1h) #2 時間前の 1 時間の平均（11:00-12:00）
trendavg(/host/key,1h:now/h-2h) #3 時間前の 1 時間の平均（10:00-11:00）
trendavg(/host/key,1M:now/M-1y) #1 年前の前月の平均
```

trendcount(/host/key,time period:time shift)

定義された期間内でトレンド値の計算に使用された、正常に取得された履歴値の数。

パラメータ:

- 共通パラメータを参照してください

例:

```
trendcount(/host/key,1h:now/h) # 直前の 1 時間（例: 12:00-13:00）の値の件数
trendcount(/host/key,1h:now/h-1h) #2 時間前（11:00-12:00）の値の件数
trendcount(/host/key,1h:now/h-2h) #3 時間前（10:00-11:00）の値の件数
trendcount(/host/key,1M:now/M-1y) #1 年前の前月の値の件数
```

trendmax(/host/key,time period:time shift)

定義された期間内のトレンド値における最大値。

パラメータ:

- 共通パラメータを参照してください

例:

```
trendmax(/host/key,1h:now/h) # 直前の 1 時間の最大値（例: 12:00-13:00）
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → 直前の 1 時間（12:00-13:00）における最大値と最小値
trendmax(/host/key,1h:now/h-1h) #2 時間前の最大値（11:00-12:00）
trendmax(/host/key,1h:now/h-2h) #3 時間前の最大値（10:00-11:00）
trendmax(/host/key,1M:now/M-1y) #1 年前の前月の最大値
```

trendmin(/host/key,time period:time shift)

定義された期間内のトレンド値における最小値です。

パラメータ:

- 共通パラメータを参照してください

例:

```
trendmin(/host/key,1h:now/h) # 直前の 1 時間の最小値（例: 12:00-13:00）
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → 直前の 1 時間（12:00-13:00）の最大値と最小値の差（
trendmin(/host/key,1h:now/h-1h) #2 時間前（11:00-12:00）の最小値
trendmin(/host/key,1h:now/h-2h) #3 時間前（10:00-11:00）の最小値
trendmin(/host/key,1M:now/M-1y) #1 年前の前月の最小値
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

検出期間中の異常の割合を返します。値は 0 から 1 の間の 10 進数で、((異常値の数)/(値の総数)) です。

パラメータ:

- 共通パラメータを参照してください
- **eval period** - 分解対象となる期間 (最小'1h')。<N><time unit> の形式で定義します。ここで
N - 時間単位の数
time unit - h（時間）、d（日）、w（週）、M（月）、または y（年）
- **detection period** - eval period の終了前の期間で、この期間について異常を計算します（最小'1h'、eval period より長くすることはできません）。<N><time unit> の形式で定義します。ここで
N - 時間単位の数
time unit - h（時間）、d（日）、w（週）
- **season** - 繰り返しパターン（「季節性」）が想定される最短の期間（最小'2h'、eval period より長くすることはできません。また、eval period 内のエントリ数は、結果として得られる頻度（season/h）の 2 倍より大きくなければなりません）。<N><time unit> の形式で定義します。ここで
N - 時間単位の数
time unit - h（時間）、d（日）、w（週）
- **deviations** - 異常として数える偏差の数（devalg によって計算、小数可）（1 以上である必要があります。デフォルトは 3）
- **devalg**（二重引用符で囲む必要があります）- 偏差アルゴリズム。*stddevpop*、*stddevsamp*、または *mad*（デフォルト）を指定できます

- **s window** - 季節成分抽出のための loess ウィンドウのスパン（ラグ数単位）（デフォルトは 10 * eval period 内のエントリ数 + 1）

例:

```
trendstl(/host/key,100h:now/h,10h,2h) # 直近 100 時間のトレンドデータを分析し、その期間の最後の 10 時間における異常率を
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #10 時間前までの 100 時間分のトレンドデータを分析し、周期性は
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #1 日前までの 100 日分のトレンドデータを分析し、その期間の最後の 10d
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #1 年前の前月を分析し、周期性は 2h を想定して、その期間の最後の
```

trendsum(/host/key,time period:time shift)

定義された期間内のトレンド値の合計です。

パラメータ:

- 共通パラメータを参照してください

例:

```
trendsum(/host/key,1h:now/h) # 直前の 1 時間の合計（例: 12:00-13:00）
trendsum(/host/key,1h:now/h-1h) #2 時間前の 1 時間の合計（11:00-12:00）
trendsum(/host/key,1h:now/h-2h) #3 時間前の 1 時間の合計（10:00-11:00）
trendsum(/host/key,1M:now/M-1y) #1 年前の前月の合計
```

サポートされているすべての関数を参照してください。

6 数学関数

ここに挙げたすべての関数は、以下の関数でサポートされています。

- トリガー式
- 計算項目

特に明記されていない限り、数学関数は浮動小数点型および整数型でサポートされます。

関数の一覧には追加情報は記載されていません。関数をクリックすると詳細が表示されます。

関数	説明
abs	値の絶対値
acos	値の逆余弦を角度で表し、ラジアンで表します。
asin	値の逆正弦を角度で表し、ラジアンで表します。
atan	ラジアンで表された角度としての値の逆正接
atan2	ラジアンで表された角度として指定された縦座標（値）と横座標の逆正接
avg	参照されている項目の値の平均値
cbrt	値の立方根
ceil	最も近い整数に切り上げます。
cos	ラジアンで表された角度としての値のコサイン
cosh	ラジアンで表された角度としての値のコサイン
cot	ラジアンで表された角度としての値のコタンジェント
degrees	値をラジアンから度に変換します。
e	オイラー数 (2.718281828459045)
exp	値のべき乗におけるオイラー数
expm1	値のべき乗から 1 を引いた値におけるオイラー数
floor	値を最も近い整数に切り捨てます。
log	自然対数
log10	小数対数
max	参照されている項目の値の最大値
min	参照されている項目の値の最小値
mod	割り算の余り
pi	円周率定数 (3.14159265358979)
power	値のべき乗
radians	値を度からラジアンに変換します。
rand	ランダムな整数値を返します。
round	値を小数点以下を切り捨てます。
signum	値が負の場合は'-1'、値がゼロの場合は'0'、値が正の場合は'1' を返します。
sin	ラジアンで表された角度の正弦

関数	説明
sinh	値の双曲線正弦（角度はラジアンで表されます）
sqrt	値の平方根
sum	参照されている項目の値の合計
tan	値の正接
truncate	値を小数点以下を切り捨てます。

関数の詳細

関数のパラメータに関する一般的な注意事項:

- 関数のパラメータはカンマで区切ります。
- 式もパラメータとして使用できます。
- オプションの関数パラメータ（またはパラメータ部分）は、< > で示されます。

abs(value)

値の絶対値（0 から）

パラメータ:

- **value** - チェックする値

例えば、'3' または'-3' のどちらの場合も、絶対値は'3' になります。

例:

```
abs(last(/host/key))>10
```

asin(value)

値の逆正弦（角度、ラジアン表記）

パラメータ:

- **value** - チェックする値

値は -1 から 1 の範囲でなければなりません。例えば、値'0.5' の逆正弦は'-0.523598776' になります。

例:

```
asin(last(/host/key))
```

asin(value)

値の逆正弦（角度、ラジアン表記）

パラメータ:

- **value** - チェックする値

値は -1 から 1 の範囲でなければなりません。例えば、値'0.5' の逆正弦は'-0.523598776' になります。

例:

```
asin(last(/host/key))
```

atan(value)

値の逆正接を角度としてラジアンで表したものです。

パラメータ:

- **value** - チェックする値

例えば、値「1」の逆正接は「0.785398163」になります。

例:

```
atan(last(/host/key))
```

atan2(value,abscissa)

縦座標（value）と横座標の逆正接（アークタンジェント）を、角度（ラジアン）で指定します。

パラメーター：

- **value** - チェックする値

- **abscissa** - 横座標の値

例えば、値'1' の縦座標と横座標の逆正接は'2.21429744' になります。

例：

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

参照されたアイテム値の平均値です。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

値の立方根

パラメータ:

- **value** - チェックする値

例えば、'64' の立方根は'4'、'63' の立方根は'3.97905721' になります。

例:

```
cbrt(last(/host/key))
```

ceil(value)

値を最も近い整数に切り上げます。

パラメータ:

- **value** - チェックする値

例えば、'2.4' は'3' に切り上げられます。floor() も参照してください。

例:

```
ceil(last(/host/key))
```

cos(value)

値のコサイン（余弦）。値はラジアンで表された角度です。

パラメーター:

- **value** - チェックする値

例えば、値'1' のコサインは'0.54030230586' になります。

例:

```
cos(last(/host/key))
```

cosh(value)

値の双曲線余弦を返します。値は科学的記数法ではなく実数として返されます。

パラメータ:

- **value** - チェックする値

例えば、値'1' の双曲線余弦は'1.54308063482' になります。

例:

```
cosh(last(/host/key))
```

cot(value)

値のコタンジェント（余弦）。値はラジアンで表された角度です。

パラメーター:

- **value** - チェックする値

例えば、値'1' のコタンジェントは'0.54030230586' になります。

例:

```
cot(last(/host/key))
```

degrees(value) {# degrees}

値をラジアンから度に変換します。

パラメーター:

- **value** - チェックする値

例えば、値'1' を度に変換すると'57.2957795' になります。

例:

```
degrees(last(/host/key))
```

e

オイラー数 (2.718281828459045)

例:

```
e()
```

exp(value)

値のべき乗におけるオイラー数

パラメータ:

- **value** - チェックする値

例えば、値'2' のべき乗におけるオイラー数は'7.38905609893065' になります。

例:

```
exp(last(/host/key))
```

expm1(value)

値から 1 を引いた値の累乗におけるオイラー数

パラメータ:

- **value** - チェックする値

例えば、値'2' から 1 を引いた値の累乗におけるオイラー数は'6.38905609893065' となります。

例:

```
expm1(last(/host/key))
```

floor(value)

値を切り捨てして最も近い整数にします。

パラメータ:

- **value** - チェックする値

例えば、'2.6' は'2' に切り捨てられます。ceil() も参照してください。

例:

```
floor(last(/host/key))
```

log(value)

自然対数

パラメータ:

- **value** - チェックする値

例えば、値'2' の自然対数は'0.69314718055994529' になります。

例:

```
log(last(/host/key))
```

log10(value)

常用対数

パラメータ:

- **value** - チェックする値

例えば、値'5' の常用対数は'0.69897000433' になります。

例:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

参照されたアイテム値の中で最も高い値です。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<value1>,<value2>,...)

参照されたアイテムの値のうち、最も低い値です。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

割り算での剰余

パラメータ:

- **value** - チェックする値
- **denominator** - 割り算の分母

例えば、値'5' を分母'2' で割った場合、剰余は'1' になります。

例:

```
mod(last(/host/key),2)
```

pi

円周率定数 (3.141592653589793)

例:

```
pi()
```

power(value,power value)

値のべき乗

パラメータ:

- **value** - チェックする値
- **power value** - 使用する N 乗

例えば、値'2' の 3 乗は'8' になります。

例:

```
power(last(/host/key),3)
```

radians(value)

値を度からラジアンに変換します。

パラメーター:

- **value** - チェックする値

例えば、値'1' をラジアンに変換すると'0.0174532925' になります。

例:

```
radians(last(/host/key))
```

rand

ランダムな整数値を返します。これは時間をシードとして生成した疑似乱数です（数学的には十分ですが、暗号には適していません）

例:

```
rand()
```

round(value,decimal places)

値を指定した小数点以下の桁数で丸めます。

パラメータ:

- **value** - チェックする値
- **decimal places** - 丸める小数点以下の桁数を指定します（0 も指定可能です）

例えば、値'2.5482' を小数点以下 2 桁に丸めると'2.55' になります。

例:

```
round(last(/host/key),2)
```

signum(value)

値が負の場合は'-1'、値がゼロの場合は'0'、値が正の場合は'1' を返します。

パラメータ:

- **value** - チェックする値

例:

```
signum(last(/host/key))
```

sin(value)

値の正弦（サイン）。値はラジアンで表された角度です。

パラメーター:

- **value** - チェックする値

例えば、値'1' の正弦は'0.8414709848' になります。

例:

```
sin(last(/host/key))
```

sinh(value)

値の双曲線正弦です。

パラメータ:

- **value** - チェックする値

例えば、値'1' の双曲線正弦は'1.17520119364' になります。

例:

```
sinh(last(/host/key))
```

sqrt(value)

値の平方根
 この関数は負の値の場合に失敗します。

パラメーター:

- **value** - チェックする値

例えば、値'3.5' の平方根は'1.87082869339' になります。

例:

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

参照されたアイテムの値の合計です。

パラメータ:

- **valueX** - アイテムの履歴を扱う別の関数によって返される値。

例:

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

値の正接

パラメータ:

- **value** - チェックする値

例えば、値'1' の正接は'1.55740772465' になります。

例:

```
tan(last(/host/key))
```

truncate(値, 小数点以下桁数)

値を指定した小数点以下桁数で切り捨てます。

パラメーター:

- 値 - チェックする値
- 小数点以下桁数 - 切り捨てる小数点以下の桁数を指定します（0 も指定可能です）

例えば、値'2.5482' を小数点以下 2 桁で切り捨てると'2.54' になります。

例:

```
truncate(last(/host/key),2)
```

サポートされているすべての関数を参照してください。

7 演算子関数

ここに挙げたすべての関数は、以下の関数でサポートされています。

- トリガー式
- 計算項目

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
between	値が指定された範囲に属しているかどうかを確認します。
in	値がリストされた値の少なくとも 1 つと等しいかどうかを確認します。

関数の詳細

関数のパラメータに関する一般的な注意事項:

- 関数のパラメータはカンマで区切ります。
- 式もパラメータとして使用できます。

between(value,min,max)

値が指定された範囲に属しているかどうかを確認します。
 サポートされる値の型: *Integer*、*Float*
 戻り値: 1 - 範囲内、0 - 範囲内以外

パラメーター:

- **value** - 確認する値
- **min** - 最小値
- **max** - 最大値

例:

`between(last(/host/key),1,10)=1 #値が1から10の範囲にある場合にトリガー`

in(value,value1,value2,...valueN)

値がリストされた値の少なくとも 1 つと等しいかどうかを確認します。
 サポートされる値の型: *Integer*、*Float*、*Character*、*Text*、*Log*
 戻り値: 等しい場合は 1、そうでない場合は 0

パラメーター:

- **value** - 確認する値
- **valueX** - リストされた値（文字列値は二重引用符で囲む必要があります）

リストされた値すべてが数値に変換できる場合は、数値として値と比較し、そうでない場合は文字列として比較します。

例:

```
in(last(/host/key),5,10)=1 #最後の値が5または10に等しい場合にトリガー
in("text",last(/host/key),last(/host/key,#2))=1 #"text"が最後の2つの値のいずれかに等しい場合にトリガー
```

サポートされているすべての関数を参照してください。

8 予測関数

ここに記載されたすべての関数は、以下の関数でサポートされています。

- トリガー式
- 計算項目

関数は追加情報なしでリストされています。関数をクリックすると詳細が表示されます。

関数	説明
予測	項目の将来値、最大値、最小値、差分、または平均値
残り時間	項目が指定されたしきい値に達するまでに必要な時間（秒）

共通パラメータ

- `/host/key` は、ホストのアイテム履歴を参照する関数に共通する必須の第 1 パラメータです
- `(sec|#num)<:time shift>` は、ホストのアイテム履歴を参照する関数に共通する第 2 パラメータで、以下を意味します:
  - **sec** - 秒単位の最大評価期間 （時間のサフィックスを使用できます）、または
  - **#num** - 直近に収集された値における最大評価範囲 （先頭にハッシュ記号を付けた場合）
  - **time shift** （任意）- 評価時点を過去に移動できます。time shift の指定方法の詳細は、こちらを参照してください。

関数の詳細

関数のパラメータに関する一般的な注意事項:

- 関数のパラメータはカンマで区切ります。
- オプションの関数パラメータ（またはパラメータ部分）は、< > で示されます。
- 関数固有のパラメータは、各関数で説明されています。
- `/host/key` および `(sec|#num)<:timeshift>` パラメータは引用符で囲んではなりません。

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

アイテムの将来値、最大値、最小値、差分、または平均値
 サポートされる値の型: *Float*、*Integer*

パラメーター:

- 共通パラメーター を参照してください。
- **time** - 予測期間（秒単位）(時間サフィックスを使用できます)。負の値もサポートされています。
- **fit** (オプション。二重引用符で囲む必要があります) - 履歴データをフィッティングするために使用する関数。サポートされている近似値:
linear - 線形関数 (デフォルト)
polynomialN - N 次多項式 (1 <= N <= 6)
exponential - 指数関数
logarithmic - 対数関数
power - べき乗関数
polynomial1 は *linear* と同等であることに注意してください。
- **mode** (オプション。二重引用符で囲む必要があります) - 要求される出力。サポートされているモード:
value - 値 (デフォルト)
max - 最大値
min - 最小値
delta - *max-min*
avg - 平均
value は、`now + time` の時点における項目値を推定することに注意してください。*max*、*min*、*delta*、および *avg* は、`now` から `now + time` までの区間における項目値の推定値を調べます。

コメント:

- 返される値が 1.7976931348623158E+308 より大きい場合、または -1.7976931348623158E+308 より小さい場合、返される値はそれぞれ 1.7976931348623158E+308 または -1.7976931348623158E+308 に切り詰められます。

- 式内で誤って使用されている場合（項目タイプが間違っている、パラメータが無効）のみサポート対象外となり、それ以外の場合はエラー時に -1 を返します。
- 予測トリガー関数 に関する追加情報も参照してください。

例:

```
forecast(/host/key,#10,1h) # 過去10個の値に基づいて、1時間後のアイテムの値を予測します。
forecast(/host/key,1h,30m) # 過去1時間のデータに基づいて、30分後のアイテムの値を予測します。
forecast(/host/key,1h:now-1d,12h) # 1日前の1時間に基づいて、12時間後のアイテムの値を予測しま。す。
forecast(/host/key,1h,10m,"exponential") # 過去1時間のデータと指数関数に基づいて、10分後のアイテムの値を予
forecast(/host/key,1h,2h,"polynomial3","max") # 過去1時間のデータと3次多項式に基づいて、今後2時間でアイテ
forecast(/host/key,#2,-20m) # 過去2つの値に基づいて、20分前のアイテムの値を推定します（これはさらに長い場
```

timeleft(/host/key,(sec|#num)<:timeshift>,threshold,<fit>)

アイテムが指定されたしきい値に達するまでに必要な時間（秒）
 サポートされる値の型：*Float*、*Integer*

パラメーター：

- 共通パラメーター を参照してください。
- **threshold** - 到達すべき値（単位サフィックス を使用できます。）
- **fit** (オプション。二重引用符で囲む必要があります) - forecast() を参照してください。

コメント:

- 返される値が 1.7976931348623158E+308 より大きい場合、返される値は 1.7976931348623158E+308 に切り詰められます。
- しきい値に達しない場合は 1.7976931348623158E+308 を返します。
- 式で誤って使用されている場合（項目タイプが間違っている、パラメータが無効）のみサポート対象外となり、それ以外の場合はエラー時に -1 を返します。
- 予測トリガー関数 に関する追加情報も参照してください。

例:

```
timeleft(/host/key,#10,0) # 過去10回分のデータに基づき、アイテムの値が0に達するまでの時間
timeleft(/host/key,1h,100) # 過去1時間分のデータに基づき、アイテムの値が100に達するまでの時間
timeleft(/host/key,1h:now-1d,100) # 1日前の1時間分に基づき、アイテムの値が100に達するまでの時間
timeleft(/host/key,1h,200,"polynomial2") # 過去1時間分のデータに基づき、アイテムが2次多項式のように振舞う
```

サポートされているすべての関数を参照してください。

9 文字列関数

ここに挙げたすべての関数は、以下の関数でサポートされています。

- トリガー式
- 計算項目

関数の一覧には追加情報は記載されていません。関数をクリックすると詳細が表示されます。

関数	説明
ascii	値の左端の文字の ASCII コード
bitlength	値のビット数
bytelength	値のバイト数
char	値を ASCII コードとして解釈し、その文字を返します。
concat	参照されている項目値または定数値を連結した文字列。
insert	文字列内の指定された位置から、指定された文字またはスペースを文字列に挿入します。
jsonpath	JSONPath の結果を返します。
left	値の左端の文字を返します。
length	値の長さ（文字数）
ltrim	文字列の先頭から指定された文字を削除します。
mid	'start' で指定された文字位置から N 文字の部分文字列を返します。
repeat	文字列を繰り返します。
replace	値内のパターンを検索し、置換文字列で置き換えます。
right	値の右端の文字を返します。
rtrim	文字列の末尾から指定された文字を削除します。
trim	文字列の先頭と末尾から指定された文字を削除します。
xmlxpath	XML XPath の結果を返します。

関数の詳細

関数のパラメータに関する一般的な注意事項:

- 関数のパラメータはカンマで区切ります。
- 式もパラメータとして使用できます。
- 文字列パラメータは二重引用符で囲む必要があります。そうでないと、誤って解釈される可能性があります。
- オプションの関数パラメータ（またはパラメータ部分）は、< > で示されます。

ascii(value)

値の左端の文字の ASCII コード
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値

例えば、'Abc' のような値は'65'（'A' の ASCII コード）を返します。

例:

```
ascii(last(/host/key))
```

bitlength(value)

値の長さ（ビット単位）
 サポートされる値の型: *String*、*Text*、*Log*、*Integer*

パラメーター:

- **value** - チェックする値

例:

```
bitlength(last(/host/key))
```

bytelength(value)

値の長さ（バイト単位）
 サポートされる値の型: *String*、*Text*、*Log*、*Integer*

パラメーター:

- **value** - チェックする値

例:

```
bytelength(last(/host/key))
```

char(value)

値を ASCII コードとして解釈し、文字を返します。
 サポートされる値の型: *Integer*

パラメーター:

- **value** - チェックする値

値は 0-255 の範囲でなければなりません。例えば、'65'（ASCII コードとして解釈）のような値は'A' を返します。

例:

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

参照されている項目の値または定数値を連結した文字列
 サポートされる値の型: *String*、*Text*、*Log*、*Float*、*Integer*

パラメーター:

- **valueX** - 履歴関数のいずれかによって返される値、または定数値（文字列、整数、または浮動小数点数）。少なくとも 2 つのパラメーターを含める必要があります。

例えば、'Zab' という値を'bix'（定数文字列）と連結すると、'Zabbix' が返されます。

例:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15]))
```

insert(value,start,length,replacement)

文字列内の指定された位置から、指定された文字またはスペースを挿入します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **start** - 開始位置
- **length** - 置換する位置
- **replacement** - 置換文字列

例えば、'Zabbbix' という値は、'bb'（開始位置 3、置換位置 2）を'b' に置き換えると、'Zabbix' に置き換えられます。

例:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

JSONPath の結果を返します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **path** - パス（引用符で囲む必要があります）
- **default** - JSONPath クエリがデータを返さない場合のオプションのフォールバック値。その他のエラーが発生した場合は、失敗が返されます（例: "unsupported construct"）

例:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

値の左端の文字を返します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **count** - 返す文字数

例えば、'Zabbix' から'Zab' を返すには、左端から 3 文字を返します。right() も参照してください。

例:

```
left(last(/host/key),3) # 左端から3文字を返します
```

length(value)

値の長さ（文字数）
 サポートされる値の型: *String*、*Text*、*Log*。

パラメーター:

- **value** - チェックする値

例:

```
length(last(/host/key)) #最新の値の長さ
length(last(/host/key,#3)) #3番目に新しい値の長さ
length(last(/host/key,#1:now-1d)) #1日前の最新の値の長さ
```

ltrim(value,<chars>)

文字列の先頭から指定された文字を削除します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **chars** (オプション) - 削除する文字を指定します。

空白文字は、オプションの文字が指定されていない場合、デフォルトで左側から切り捨てられます。rtrim()、trim() も参照してください。

例:

```
ltrim(last(/host/key)) #文字列の先頭から空白文字を削除します。
ltrim(last(/host/key),"Z") #文字列の先頭からすべての'Z'を削除します。
ltrim(last(/host/key),"Z") #文字列の先頭からすべてのスペースと'Z'を削除します。
```

mid(value,start,length)

'start' で指定された文字位置から始まる N 文字の部分文字列を返します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **start** - 部分文字列の開始位置
- **length** - 部分文字列内で返す位置

例えば、開始位置が 2 で、返す位置が 4 の場合、'Zabbix' という値から'abbi' を返すことができます。

例:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

文字列を繰り返します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **count** - 繰り返し回数

例:

```
repeat(last(/host/key),2) # 値を2回繰り返します。
```

replace(value,pattern,replacement)

値内のパターンを検索し、置換文字列で置き換えます。パターンに一致するすべての箇所が置換されます。
 サポートされる値の型: *String*、*Text*、*Log

パラメーター:

- **value** - チェックする値
- **pattern** - 検索するパターン
- **replacement** - パターンを置換する文字列

例:

```
replace(last(/host/key),"ibb","abb") # すべての'ibb'を'abb'に置き換えます。
```

right(value,count)

値の右端の文字を返します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **count** - 返す文字数

例えば、'Zabbix' から'bix' を返すには、右端から 3 文字を返します。left() も参照してください。

例:

```
right(last(/host/key),3) # 右端から3文字を返します。
```

rtrim(value,<chars>)

文字列の末尾から指定された文字を削除します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **chars** (オプション) - 削除する文字を指定します。

空白はデフォルトで右側から切り取られます（オプションの文字が指定されていない場合）。ltrim()、trim() も参照してください。

例:

```
rtrim(last(/host/key)) #文字列の末尾から空白を削除します。
rtrim(last(/host/key),"x") #文字列の末尾からすべての 'x' を削除します。
rtrim(last(/host/key),"x ") #文字列の末尾からすべての 'x' とスペースを削除します。
```

trim(value,<chars>)

文字列の先頭と末尾から指定された文字を削除します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **chars** (オプション) - 削除する文字を指定

デフォルトでは、両端の空白が削除されます (オプションの文字が指定されていない場合)。ltrim()、rtrim() も参照してください。

例:

```
trim(last(/host/key)) #文字列の先頭と末尾から空白を削除します。
trim(last(/host/key),"_") #文字列の先頭と末尾から '_' を削除します。
```

xmlxpath(value,path,<default>)

XML XPath の結果を返します。
 サポートされる値の型: *String*、*Text*、*Log*

パラメーター:

- **value** - チェックする値
- **path** - パス（引用符で囲む必要があります）
- **default** - XML XPath クエリが空のノードセットを返す場合のオプションのフォールバック値。空の結果がノードセットでない場合（つまり、空文字列の場合）、この値は返されません。その他のエラーの場合は、失敗が返されます（例: ”invalid expression”）

例:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

サポートされているすべての関数を参照してください。

6 マクロ

サポートされているマクロとロケーションでサポートされているユーザーマクロをすぐに使用することができます。

1 ロケーションでサポートされているマクロ

概要

場所ごとにサポートされるマクロ の表には、特定の場所でサポートされるすべてのマクロ（組み込み、式、およびユーザー定義）のリファレンスが含まれています。

関連するグループのすべてのマクロがサポートされている場合は、グループ名のみが表示されます。各マクロ名またはグループ名をクリックすると、マクロの詳細をすべて確認できます。

マクロの詳細 の一覧には、アプリケーション領域ごとにグループ化された、すべての組み込みマクロの詳細なリストが含まれています。マクロの値をカスタマイズするには（たとえば、特定の部分文字列を短縮または抽出する場合）、マクロ関数 を使用できます。

場所ごとにサポートされるマクロ

場所	サポートされるマクロ	コメント
アクション 期間条件、デフォルトの実行ステップの期間、ステップの期間	ユーザー定義マクロ（グローバルのみ）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません
通知とコマンド		

場所	サポートされるマクロ	コメント
トリガーベース	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ、原因/症状イベントマクロ イベント更新マクロ: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} 式マクロ: {?EXPRESSION} {FUNCTION.VALUE} ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} 対象ホストマクロ（コマンドのみ） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}, {ITEM.STATE.ERROR} を除く） プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} ユーザー定義マクロ	あわせて参照: メッセージでのマクロの使用
障害更新	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ、原因/症状イベントマクロ、復旧イベントマクロ イベント更新マクロ: すべて（{EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY} を除く） 式マクロ: {?EXPRESSION} 関数マクロ ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} 対象ホストマクロ（コマンドのみ） ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}, {ITEM.STATE.ERROR} を除く） プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.ID}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} {USER.FULLNAME} ユーザー定義マクロ	
障害復旧	復旧イベントマクロ {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	

場所		サポートされるマクロ	コメント
	ディスカバリ	アクションマクロ 日付/時刻マクロ ディスカバリマクロ イベントマクロ: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} 対象ホストマクロ（コマンドのみ） プロキシマクロ	
	自動登録	アクションマクロ 日付/時刻マクロ イベントマクロ: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} ホストマクロ: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} 対象ホストマクロ（コマンドのみ） プロキシマクロ	
	サービスベース	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ: {EVENT.ACK.STATUS}, {EVENT.OPDATA} を除くすべて サービスマクロ ユーザー定義マクロ	
	サービス復旧	イベントマクロ（{EVENT.ACK.STATUS}, {EVENT.OPDATA} を除く）、復旧イベントマクロ	
	サービス更新	アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ（{EVENT.ACK.STATUS}, {EVENT.OPDATA} を除く） イベント更新マクロ: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} サービスマクロ ユーザー定義マクロ	
内部通知		アクションマクロ 日付/時刻マクロ {ESC.HISTORY} イベントマクロ（{EVENT.NSEVERITY}, {EVENT.SEVERITY} を除く）、復旧イベントマクロ ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} プロキシマクロ	
	アイテムベース	アイテムマクロ: {ITEM.STATE}, {ITEM.STATE.ERROR}	
	LLD ルールベース	ローレベルディスカバリ通知マクロ	

場所		サポートされるマクロ	コメント
アラートスクリプトパラメータ	トリガーベース	トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATE}, {TRIGGER.STATE.ERROR}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME} ユーザー定義マクロ アラートマクロ ユーザー定義マクロ（グローバルのみ）	
コネクタ	*URL*、ユーザー名、パスワード、*Bearer* トークン、*HTTP* プロキシ、*SSL* 証明書、*SSL* キーファイル、*SSL* キーパスワードフィールド	ユーザー定義マクロ（グローバルのみ）	
	タイムアウトフィールド	ユーザー定義マクロ（グローバルのみ）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
グラフ名		式マクロ: {?EXPRESSION}	この場所では、このマクロ内で秒をパラメータとする単一の **avg**、**last**、**max**、**min** 関数のみがサポートされます（時間のサフィックスを使用できます）。 {HOST.HOST<1-9>} マクロをマクロ内のホストとして使用できます（インデックス付きマクロも参照）。 使用例: {?avg(/{HOST.HOST}/item.key,1h)} サポートされない使用例: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}

場所		サポートされるマクロ	コメント
ホスト、ホストプロトタイプ	インターフェース *IP/DNS*	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} ユーザー定義マクロ [1]	[1] IP アドレスでは、単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません
	インターフェースポート	ユーザー定義マクロ	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません
	SNMP コミュニティ *SNMPv3* コンテキスト名、セキュリティ名、認証パスフレーズ、プライバシーパスフレーズ *IPMI* ユーザー名、パスワード	ユーザー定義マクロ	
アイテム、アイテムプロトタイプ、**LLD** ルール	名前	ユーザー定義マクロ	LLD ルール名ではサポートされません。
	説明	ユーザー定義マクロ	

場所		サポートされるマクロ	コメント
アイテムタイプ	キーのパラメータ	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	サポートされる {HOST.*} マクロは、そのアイテムに選択されたインターフェースに解決されます。インターフェースのないアイテムで使用した場合は、優先順位に従ってホストの Zabbix agent、SNMP、JMX、または IPMI インターフェースに解決され、ホストにインターフェースがない場合は'UNKNOWN'に解決されます。
	更新間隔、カスタム間隔、タイムアウト（サポートされるアイテムタイプで利用可能）、保存期間（履歴/トレンド用）フィールド	ユーザー定義マクロ	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	失われたリソースを削除、失われたリソースを無効化フィールド	ユーザー定義マクロ	これらのフィールドは LLD ルールでのみサポートされます。 単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	前処理ステップのパラメータ（カスタムスクリプトを含む）、カスタムエラー処理パラメータ（値を設定および エラーを設定フィールド）	ユーザー定義マクロ	
	フィルター正規表現	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	これらのフィールドは LLD ルールでのみサポートされます。
	オーバーライドフィルター正規表現	ユーザー定義マクロ	これらのフィールドは LLD ルールでのみサポートされます。
	オーバーライド操作: 更新間隔、履歴保存期間、トレンド保存期間（アイテムプロトタイプ用）	ユーザー定義マクロ	これらのフィールドは LLD ルールでのみサポートされます。 単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。

場所		サポートされるマクロ	コメント
	ブラウザアイテム パラメータ名と値	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} ユーザー定義マクロ	
	スクリプトアイテム パラメータ名と値		
	計算/集計アイテムの数式 (式定数、関数パラメータ、アイテムキーのパラメータ、(集計アイテムのみ）フィルター条件（ホストグループ名とタグ名))	ユーザー定義マクロ	
	データベースモニタアイテム ユーザー名、パスワード、*SQL* クエリフィールド	ユーザー定義マクロ	
	HTTP エージェントアイテム *URL*、クエリフィールド、リクエストボディ、ヘッダー（名前と値)、*SSL* 証明書ファイル、*SSL* キーファイルフィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} ユーザー定義マクロ	シークレットマクロ を含む URL は動作しません。これは、その中のマクロが”******” として解決されるためです
	HTTP エージェントアイテム *HTTP* プロキシフィールド	アイテムマクロ: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} ユーザー定義マクロ	
	HTTP エージェントアイテム 必要なステータスコード、*HTTP* 認証のユーザー名/パスワード、*SSL* キーパスワードフィールド	ユーザー定義マクロ	
	HTTP エージェントアイテム 許可されたホストフィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	JMX エージェントアイテム エンドポイントフィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} ユーザー定義マクロ	
	トラッパーアイテム 許可されたホストフィールド		
	SNMP エージェントアイテム *SNMP OID* フィールド	ユーザー定義マクロ	
	SSH および **Telnet** アイテム スクリプト	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	

場所		サポートされるマクロ	コメント
	SSH エージェントアイテム ユーザー名、公開鍵ファイル、秘密鍵ファイル、パスワードフィールド	ユーザー定義マクロ	
	Telnet エージェントアイテム ユーザー名、パスワードフィールド	ユーザー定義マクロ	
マップ			
	URL、*URL* 名	ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} ホストインベントリマクロ マップマクロ {TRIGGER.ID}	
	要素ラベル	式マクロ: {?EXPRESSION} [1] ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} ホストインベントリマクロ マップマクロ トリガーマクロ: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] この場所では、このマクロ内で秒をパラメータとする単一の **avg**、**last**、**max**、**min** 関数のみがサポートされます（時間のサフィックスを使用できます）。 {HOST.HOST<1-9>} マクロをマクロ内のホストとして使用できます。 使用例: {?avg(/{HOST.HOST}/item.key,1h)} サポートされない使用例: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
	リンクラベル	式マクロ: {?EXPRESSION} [1]	
	図形ラベル	式マクロ: {?EXPRESSION} [1]	
	マップ図形内のテキストフィールド	{MAP.NAME}	
ネットワークディスカバリ			
	更新間隔フィールド	ユーザー定義マクロ（グローバルのみ）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	SNMP コミュニティ、*SNMP OID* フィールド	ユーザー定義マクロ（グローバルのみ）	

場所		サポートされるマクロ	コメント
プロキシ	*SNMPv3* コンテキスト名、セキュリティ名、認証パスフレーズ、プライバシーパスフレーズフィールド		
プロキシグループ	アクティブエージェント用アドレス > ポート (プロキシがグループに属している場合) パッシブプロキシモードの場合: インターフェースの アドレスおよび ポートフィールド オーバーライドモードの アイテムタイプのタイムアウトフィールド	ユーザー定義マクロ（グローバルのみ）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	フェイルオーバー期間、最小プロキシ数フィールド	ユーザー定義マクロ（グローバルのみ）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
スクリプト	コマンド、確認テキスト (タイプ: スクリプト、手動 ホストアクション)	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ [2] {MANUALINPUT} ユーザー名マクロ ユーザー定義マクロ	[1] 確認テキストのみ [2] Zabbix server および Zabbix proxy でサポート

場所		サポートされるマクロ	コメント
	コマンド (タイプ: スクリプト、手動 イベントアクション)	日付/時刻マクロ イベントマクロ、原因/症状イベントマクロ、復旧イベントマクロ (復旧が行われた場合) イベント更新マクロ: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ（{ITEM.STATE}, {ITEM.STATE.ERROR} を除く） {MANUALINPUT} プロキシマクロ トリガーマクロ: {TRIGGER.DESCRIPTION}, {TRIGGER.EVENTS.*}, {TRIGGER.EXPRESSION}, {TRIGGER.EXPRESSION.EXPLAIN}, {TRIGGER.EXPRESSION.RECOVERY}, {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}, {TRIGGER.HOSTGROUP.NAME}, {TRIGGER.ID}, {TRIGGER.NAME}, {TRIGGER.NAME.ORIG}, {TRIGGER.NSEVERITY}, {TRIGGER.SEVERITY}, {TRIGGER.STATUS}, {TRIGGER.TEMPLATE.NAME}, {TRIGGER.URL}, {TRIGGER.URL.NAME}, {TRIGGER.VALUE} ユーザー名マクロ ユーザー定義マクロ	
	確認テキスト (タイプ: スクリプト、手動 イベントアクション)	イベントマクロ: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ {MANUALINPUT} ユーザー名マクロ ユーザー定義マクロ	
	URL、確認テキスト (タイプ: URL、手動 ホスト/イベントアクション)		
タグ名と値		ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} トリガーマクロ: {TRIGGER.ID}（トリガータグ値のみ） ユーザー定義マクロ	タグ名および値のマクロは、イベント生成処理中にのみ解決されます。
トリガー			
	名前	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} 位置マクロ/参照: $1...$9 ユーザー定義マクロ	

場所		サポートされるマクロ	コメント
	イベント名	日付/時刻マクロ: {TIME}, {TIMESTAMP} 式マクロ: {?EXPRESSION} 関数マクロ ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} トリガーマクロ: {TRIGGER.EXPRESSION.EXPLAIN}	
	運用データ	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} ユーザー定義マクロ	
	式	{TRIGGER.VALUE} ユーザー定義マクロ [1]	[1] 定数および関数パラメータ内のみ。シークレットマクロはサポートされません
	一致用タグ、メニューエントリ名、メニューエントリ *URL* フィールド	ユーザー定義マクロ	シークレットマクロ を含む URL は動作しません。これは、その中のマクロが"******" として解決されるためです。
	説明	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} ユーザー定義マクロ	
	URL	{EVENT.ID} ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} アイテムマクロ: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID} 位置マクロ/参照: $1...$9	
ユーザーパラメータコマンド			
Web シナリオ			
	名前、エージェント、*HTTP* プロキシフィールド	ユーザー定義マクロ	
	更新間隔フィールド	ユーザー定義マクロ	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。

場所		サポートされるマクロ	コメント
	変数、ヘッダー、*SSL* 証明書、*SSL* キーファイルフィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	ユーザー、パスワード、*SSL* キーパスワードフィールド	ユーザー定義マクロ	
	Web シナリオステップ名前、変数（値のみ）フィールド	ユーザー定義マクロ	
	Web シナリオステップタイムアウト、必要なステータスコードフィールド	ユーザー定義マクロ	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	Web シナリオステップ *URL*、ヘッダー（名前と値）、必要な文字列フィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	Web シナリオステップ *Post* フィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
webhook			
メディアタイプ			
	パラメータ値	アラートマクロ トリガーベースの障害通知でサポートされるすべてのマクロ（上記参照）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	JavaScript スクリプト、*JavaScript* スクリプトパラメータ名	ユーザー定義マクロ（グローバルのみ）	
	メニューエントリ名、メニューエントリ *URL*	{EVENT.TAGS.<tag name>}	
ウィジェット			
	ゲージウィジェット説明フィールド	ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ アイテムマクロ（{ITEM.STATE.ERROR} を除く） ユーザー定義マクロ（グローバルのみ）	
	ハニカムウィジェット主/副ラベルフィールド		
	アイテム値ウィジェット説明フィールド		

場所		サポートされるマクロ	コメント
その他の場所	トップホストウィジェット テキストデータ列	ホストマクロ: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ホストインベントリマクロ ユーザー定義マクロ（グローバルのみ）	
	URL（動的）ウィジェット *URL* フィールド	ホストマクロ: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} ユーザー定義マクロ	
	稼働時間フィールド	ユーザー定義マクロ（グローバルのみ）	単一のマクロでフィールド全体を埋める必要があります。複数のマクロやテキストとの混在はサポートされません。
	アイテムタイプのタイムアウトフィールド (グローバル設定) ユーザーメディアアクティブな時間フィールド		
	メールメディアタイプユーザー名、パスワードフィールド	ユーザー定義マクロ（グローバルのみ）	シークレットマクロ の使用を推奨します。
	定期レポート件名、メッセージフィールド	日付/時刻マクロ: {TIME}, {TIMESTAMP}	

マクロの詳細

この一覧には、アプリケーション領域ごとに分類された各組み込みマクロの詳細が含まれています。

- アクションマクロ
- アラートマクロ
- 日付/時刻マクロ
- ディスカバリマクロ
- イベントマクロ
  - イベント更新マクロ
  - 原因/症状イベントマクロ
  - リカバリイベントマクロ
- エスカレーションマクロ
- 関数マクロ
- ホストマクロ
  - 対象ホストマクロ
- ホストグループマクロ
- ホストインベントリマクロ
- アイテムマクロ
- ローレベルディスカバリ通知マクロ
- マップマクロ
- プロキシマクロ
- スクリプトマクロ
- サービスマクロ
- トリガーマクロ
- ユーザー名マクロ

アクションマクロ

{ACTION.ID}

トリガーされたアクションの数値 ID

{ACTION.NAME}

トリガーされたアクションの名前

アラートマクロ

{ALERT.MESSAGE}

アクション設定のデフォルトメッセージの値

{ALERT.SENDTO}

ユーザーメディア設定の送信先の値

{ALERT.SUBJECT}

アクション設定のデフォルトの件名の値

日付と時刻のマクロ

{DATE}

現在の日付を `yyyy.mm.dd.` 形式で表します。

{TIME}

現在の時刻を `hh:mm:ss` 形式で表します。

{TIMESTAMP}

UNIX タイムスタンプ形式での現在の時刻

ディスカバリマクロ

{DISCOVERY.DEVICE.IPADDRESS}

検出されたデバイスの IP アドレス
 追加されるホストに関係なく、常に利用可能です。

{DISCOVERY.DEVICE.DNS}

検出されたデバイスの DNS 名
 追加されるホストに関係なく、常に利用可能です。

{DISCOVERY.DEVICE.STATUS}

検出されたデバイスのステータス（UP/DOWN）

{DISCOVERY.DEVICE.UPTIME}

特定のデバイスの検出ステータスが最後に変更されてからの経過時間
 例: 1 時間 29 分 01 秒
 ステータスが DOWN のデバイスの場合、これはダウンタイムの期間です。

{DISCOVERY.RULE.NAME}

デバイス/サービスの有無を検出した検出ルールの名前

{DISCOVERY.SERVICE.NAME}

検出されたサービスの名前
 例: HTTP

{DISCOVERY.SERVICE.PORT}

検出されたサービスのポート
 例: 80

{DISCOVERY.SERVICE.STATUS}

検出されたサービスのステータス（UP/DOWN）

{DISCOVERY.SERVICE.UPTIME}

特定のサービスの検出ステータスが最後に変更されてからの経過時間
 例: 1 時間 29 分 01 秒
 ステータスが DOWN のサービスの場合、これはダウンタイムの期間です。

エスカレーションマクロ

{ESC.HISTORY}

以前送信された通知のログ、エスカレーション手順、ステータス（送信済み、進行中、失敗）

イベントマクロ

{EVENT.ACK.STATUS}

イベントの確認ステータス（はい/いいえ）

{EVENT.AGE}

アクションをトリガーしたイベントの経過時間（秒単位の精度）
 エスカレーションされたメッセージで役立ちます。例：9 分 13 秒

{EVENT.DATE}

アクションをトリガーしたイベントの日付です。形式は `yyyy.mm.dd` です。
 例: 2025.04.14。

{EVENT.DURATION}

イベントの継続時間（問題発生イベントと回復イベント間の時間差）。精度は 1 秒単位です。
 問題発生時の回復メッセージで役立ちます。

{EVENT.ID}

アクションをトリガーしたイベントの数値 ID

{EVENT.NAME}

アクションをトリガーしたイベントの名前

{EVENT.NSEVERITY}

イベントの重大度を表す数値
 有効な値: *0* - 未分類、*1* - 情報、*2* - 警告、*3* - 平均、*4* - 高、*5* - 災害

{EVENT.OBJECT}

イベントオブジェクトの数値
 可能な値: *0* - トリガー、*1* - 検出されたホスト、*2* - 検出されたサービス、*3* - 自動登録、*4* - アイテム、*5* - 低レベル検出ルール

{EVENT.OPDATA}

問題の根本的なトリガーの運用データ

{EVENT.SEVERITY}

イベントの重大度名

{EVENT.SOURCE}

イベントソースの数値
 有効な値: *0* - トリガー、*1* - 検出、*2* - 自動登録、*3* - 内部、*4* - サービス

{EVENT.STATUS}

アクションをトリガーしたイベントの言語値

{EVENT.TAGS}

イベントタグのコンマ区切りリスト
 タグが存在しない場合は空の文字列に展開されます。

{EVENT.TAGSJSON}

イベントタグのオブジェクトを含む JSON 配列です。
 タグが存在しない場合は、空の配列に展開されます。

{EVENT.TAGS< タグ名 >}

タグ名が参照するイベントタグの値
 タグ名に英数字以外の文字（英語以外のマルチバイト UTF 文字を含む）が含まれる場合は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。

{EVENT.TIME}

アクションをトリガーしたイベントの時刻です。形式は `hh:mm:ss` です。
 例: 12:57:53。

{EVENT.TIMESTAMP}

アクションをトリガーしたイベントの UNIX タイムスタンプ

{EVENT.VALUE}

アクションをトリガーしたイベントの数値
 有効な値: *1* - 問題あり *0* - 回復中

原因/症状イベントマクロ

{EVENT.CAUSE.ACK.STATUS}

原因イベントの確認ステータス（Yes/No）
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.AGE}

原因イベントの発生期間（秒単位の精度）
 エスカレーションされたメッセージで役立ちます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.DATE}

原因イベントの日付（yyyy.mm.dd 形式）。
 症状イベントのコンテキストで使用されます。

{EVENT.DURATION}

イベントの継続時間（問題発生イベントと回復イベント間の時間差）。精度は 1 秒単位です。
 問題発生時の回復メッセージで役立ちます。

{EVENT.CAUSE.ID}

原因イベントの数値 ID
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.NAME}

原因イベントの名前
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.NSEVERITY}

原因イベントの重大度を表す数値
 有効な値: *0* - 未分類、*1* - 情報、*2* - 警告、*3* - 平均、*4* - 高、*5* - 災害。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.OBJECT}

原因イベントオブジェクトの数値
 可能な値: *0* - トリガー、*1* - 検出されたホスト、*2* - 検出されたサービス、*3* - 自動登録、*4* - 項目、*5* - 低レベル検出ルール
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.OPDATA}

原因となる問題の根本的なトリガーの運用データ
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.SEVERITY}

原因イベントの重大度の名前
 可能な値: 未分類、情報、警告、平均、高、災害
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.SOURCE}

原因イベントソースの数値
 有効な値: *0* - トリガー、*1* - 検出、*2* - 自動登録、*3* - 内部
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.STATUS}

原因イベントの言語的値
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGS}

原因イベントタグのコンマ区切りリスト
 タグが存在しない場合は空の文字列に展開されます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGSJSON}

原因イベントのタグオブジェクトを含む JSON 配列。
 タグが存在しない場合は空の配列に展開されます。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TAGS.< タグ名 >}

タグ名が参照する原因イベントタグの値
 英数字以外の文字（英語以外のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名内の引用符とバックスラッシュは、バックスラッシュでエスケープする必要があります。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TIME}

原因イベントの時刻です。形式は hh:mm:ss です。
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.TIMESTAMP}

原因イベントの UNIX タイムスタンプ
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.UPDATE.HISTORY}

原因となる問題の更新（確認応答など）のログ
 症状イベントのコンテキストで使用されます。

{EVENT.CAUSE.VALUE}

原因イベントの数値
 有効な値: *1* - 問題、*0* - 回復中
 症状イベントのコンテキストで使用されます。

{EVENT.SYMPTOMS}

症状イベントのリスト。ホスト名、イベント名、重大度、経過時間、サービスタグと値などの詳細が含まれます。
 このマクロは原因イベントのコンテキストで使用され、症状イベントに関する情報を返します。

リカバリイベントマクロ

{EVENT.RECOVERY.DATE}

復旧イベントの日付です。形式は `yyyy.mm.dd` です。

{EVENT.RECOVERY.ID}

リカバリイベントの数値 ID

{EVENT.RECOVERY.NAME}

リカバリイベントの名前

{EVENT.RECOVERY.STATUS}

リカバリイベントの言語値

{EVENT.RECOVERY.TAGS}

リカバリイベントタグのカンマ区切りリスト。タグが存在しない場合は空の文字列に展開されます。

{EVENT.RECOVERY.TAGSJSON}

リカバリーイベントのタグオブジェクトを含む JSON 配列です。タグが存在しない場合は、空の配列に展開されます。

{EVENT.RECOVERY.TIME}

復旧イベントの時刻です。`hh:mm:ss` 形式です。

{EVENT.RECOVERY.TIMESTAMP}

リカバリイベントの UNIX タイムスタンプ

{EVENT.RECOVERY.VALUE}

リカバリイベントの数値

イベント更新マクロ

{EVENT.UPDATE.ACTION}

障害の更新中に実行されたアクションの、人が読める形式の名前です。
 次の値に展開されます: *acknowledged*、*unacknowledged*、*commented*、*changed severity from (original severity) to (updated severity)*、*closed*（1 回の更新で実行されたアクション数に応じて異なります）。

{EVENT.UPDATE.ACTIONJSON}

障害更新中に実行されたアクションの詳細を含む JSON 配列。
 使用可能な JSON プロパティ値:
- `true`（`acknowledge`、`unacknowledge`、`close`、`unsuppress`、`cause`、`symptom` プロパティの場合）;
- < メッセージ文字列 >（`message` プロパティの場合）;
- タイムスタンプ（`suppress_until` および `timestamp` プロパティの場合）、または 0（無期限に抑制されている場合の `suppress_until` プロパティ）;
- 0、1、2、3、4、5（`old` および `new` の深刻度プロパティの場合）。

 例: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851`

{EVENT.UPDATE.DATE}

イベントの更新（障害確認など）の日付です。形式は `yyyy.mm.dd` です。
`{ACK.DATE}` は非推奨であり、サポートされている場所では解決されない場合があります。

{EVENT.UPDATE.HISTORY}

障害の更新（障害確認など）のログです。
 `{EVENT.ACK.HISTORY}` は非推奨であり、サポートされている場所では解決されない可能性があります。

{EVENT.UPDATE.MESSAGE}

障害更新メッセージ。
 `{ACK.MESSAGE}` は非推奨であり、サポートされている場所でも解決されない場合があります。

{EVENT.UPDATE.NSEVERITY}

問題更新操作中に設定された新しいイベント重大度の数値

{EVENT.UPDATE.SEVERITY}

問題更新操作中に設定された新しいイベント重大度の名前

{EVENT.UPDATE.STATUS}

問題の更新ステータスの数値
 有効な値: *0* - 問題/回復イベントにより Webhook が呼び出された。*1* - 更新操作

{EVENT.UPDATE.TIME}

イベントの更新（障害確認など）の時刻です。形式は `hh:mm:ss` です。
{ACK.TIME} は非推奨であり、サポートされている場所では解決されない場合があります。

{EVENT.UPDATE.TIMESTAMP}

イベント更新（確認応答など）の UNIX タイムスタンプ

関数マクロ

{FUNCTION.VALUE}

イベント発生時点における、トリガー式内の N 番目のアイテムベース関数の値です。
 第 1 パラメータが `/host/key` の関数のみがカウントされます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などの関数を指定するために、数値インデックスとともに使用できます。例: {FUNCTION.VALUE<1-9>}。詳細はindexed macrosを参照してください。

{FUNCTION.RECOVERY.VALUE}

イベント発生時点における、リカバリ式内の N 番目のアイテムベース関数の値。*
 第 1 パラメータが`/host/key` である関数のみがカウントされます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などの関数を指定するために、数値インデックスとともに使用できます。例: {FUNCTION.RECOVERY.VALUE<1-9>}。indexed macrosを参照してください。

ホストマクロ

{HOST.CONN}

ホストの設定に応じた、ホストの IP アドレスまたは DNS 名です。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースが 1 つもない場合、このマクロは'UNKNOWN' に解決されます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.CONN<1-9>} のように数値インデックス付きで使用できます。indexed macrosを参照してください。

{HOST.DESCRIPTION}

ホストの説明。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.DESCRIPTION<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.DNS}

ホストの DNS 名です。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースが 1 つもない場合、このマクロは「UNKNOWN」に解決されます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.DNS<1-9>} のように数値インデックス付きで使用できます。indexed macros を参照してください。

{HOST.HOST}

技術的なホスト名です。
 このマクロが履歴関数の最初のパラメータ内のプレースホルダーとして使用される場合、たとえば `last(/{HOST.HOST}/{ITEM.KEY})` では、マクロ関数はこのマクロではサポートされません。
 {HOSTNAME<1-9>} は非推奨であり、サポートされている場所で解決されない場合があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.HOST<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.ID}

ホスト ID。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.ID<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.IP}

ホストの IP アドレス。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインター

フェースが 1 つもない場合、このマクロは'UNKNOWN' に解決されます。
 {IPADDRESS<1-9>} は非推奨であり、サポートされている場所では解決されない可能性があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.IP<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.METADATA}

ホストのメタデータ

 アクティブエージェントの自動登録にのみ使用されます。

{HOST.NAME}

表示されるホスト名。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.NAME<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{HOST.PORT}

ホストの（エージェント）ポートです。
 グローバルスクリプト、インターフェースの IP/DNS フィールド、および Web シナリオでは、このマクロはメインのエージェントインターフェースに解決されます。エージェントインターフェースが定義されていない場合は、メインの SNMP インターフェースが使用されます。SNMP インターフェースも定義されていない場合は、メインの JMX インターフェースが使用されます。JMX インターフェースも定義されていない場合は、メインの IPMI インターフェースが使用されます。ホストにインターフェースが 1 つもない場合、このマクロは'UNKNOWN' に解決されます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{HOST.PORT<1-9>} のように数値インデックス付きで使用できます。indexed macros を参照してください。

ターゲットホストマクロ

{HOST.TARGET.CONN}

ターゲットホストの IP アドレスまたは DNS 名

{HOST.TARGET.DNS}

ターゲットホストの DNS 名

{HOST.TARGET.HOST}

ターゲットホストの技術名

{HOST.TARGET.IP}

ターゲットホストの IP アドレス

{HOST.TARGET.NAME}

ターゲットホストの表示名

ホストグループマクロ

{HOSTGROUP.ID}

ホストグループ ID

ホストインベントリマクロ

{INVENTORY.ALIAS}

ホストインベントリの *Alias* フィールド。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.ALIAS<1-9>} のように記述します。indexed macrosを参照してください。

{INVENTORY.ASSET.TAG}

ホストインベントリの資産タグフィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.ASSET.TAG<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.CHASSIS}

ホストインベントリ内のシャーシフィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.CHASSIS<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.CONTACT}

ホストインベントリの *Contact* フィールドです。
 {PROFILE.CONTACT<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.CONTACT<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.CONTRACT.NUMBER}

ホストインベントリの契約番号フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.CONTRACT.NUMBER<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.DEPLOYMENT.STATUS}

ホストインベントリの *Deployment status* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.DEPLOYMENT.STATUS<1-9>}。indexed macrosを参照してください。

{INVENTORY.HARDWARE}

ホストインベントリ内のハードウェアフィールド。
 {PROFILE.HARDWARE<1-9>} は非推奨であり、サポートされている場所では解決されない可能性があります。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.HARDWARE<1-9>}。詳細はindexed macros を参照してください。

{INVENTORY.HARDWARE.FULL}

ホストインベントリのハードウェア（詳細）フィールド。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.HARDWARE.FULL<1-9>} のように指定します。インデックス付きマクロを参照してください。

{INVENTORY.HOST.NETMASK}

ホストインベントリのホストのサブネットマスクフィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.HOST.NETMASK<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.HOST.NETWORKS}

ホストインベントリのホストのネットワークフィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.HOST.NETWORKS<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.HOST.ROUTER}

ホストインベントリ内のホストルーターフィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.HOST.ROUTER<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.HW.ARCH}

ホストインベントリのハードウェアアーキテクチャフィールド。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.HW.ARCH<1-9>} を使用します。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.DECOMM}

ホストインベントリのハードウェアの廃棄日フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.HW.DATE.DECOMM<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.EXPIRY}

ホストインベントリ内のハードウェア保守期限日フィールド。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.HW.DATE.EXPIRY<1-9>}。indexed macrosを参照してください。

{INVENTORY.HW.DATE.INSTALL}

ホストインベントリのハードウェアの設置日フィールド。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.HW.DATE.INSTALL<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.HW.DATE.PURCHASE}

ホストインベントリ内のハードウェア購入日フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.HW.DATE.PURCHASE<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.INSTALLER.NAME}

ホストインベントリ内のインストーラー名フィールド。

 このマクロは数値インデックスとともに使用できます。例えば、{INVENTORY.INSTALLER.NAME<1-9>} のように指定すると、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指すことができます。インデックス付きマクロを参照してください。

{INVENTORY.LOCATION}

ホストインベントリ内の *Location* フィールドです。
 {PROFILE.LOCATION<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.LOCATION<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.LOCATION.LAT}

ホストインベントリ内の位置の緯度フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.LOCATION.LAT<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.LOCATION.LON}

ホストインベントリの位置の経度フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.LOCATION.LON<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.MACADDRESS.A}

ホストインベントリ内の *MAC address A* フィールドです。
 {PROFILE.MACADDRESS<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.MACADDRESS.A<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.MACADDRESS.B}

ホストインベントリ内の *MAC address B* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.MACADDRESS.B<1-9>}。詳細はインデックス付きマクロを参照してください。

{INVENTORY.MODEL}

ホストインベントリ内のモデルフィールド。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー式で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.MODEL<1-9>} のように指定します。インデックス付きマクロを参照してください。

{INVENTORY.NAME}

ホストインベントリ内の *Name* フィールド。
 {PROFILE.NAME<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.NAME<1-9>}。詳細はindexed macros を参照してください。

{INVENTORY.NOTES}

ホストインベントリ内の *Notes* フィールド。
 {PROFILE.NOTES<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.NOTES<1-9>}。詳細はindexed macros を参照してください。

{INVENTORY.OOB.IP}

ホストインベントリ内の *OOB IP* アドレスフィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.OOB.IP<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.OOB.NETMASK}

ホストインベントリ内の *OOB* サブネットマスクフィールド。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.OOB.NETMASK<1-9>}。indexed macrosを参照してください。

{INVENTORY.OOB.ROUTER}

ホストインベントリ内の *OOB* ルーターフィールドです。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.OOB.ROUTER<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.OS}

ホストインベントリ内の *OS* フィールドです。
 {PROFILE.OS<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.OS<1-9>}。詳細はindexed macros を参照してください。

{INVENTORY.OS.FULL}

ホストインベントリ内の *OS*（詳細）フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.OS.FULL<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.OS.SHORT}

ホストインベントリ内の *OS*（短縮名）フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.OS.SHORT<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.CELL}

ホストインベントリ内の *Primary POC cell* フィールド。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.POC.PRIMARY.CELL<1-9>}。詳細はインデックス付きマクロを参照してください。

{INVENTORY.POC.PRIMARY.EMAIL}

ホストインベントリの *Primary POC email* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.POC.PRIMARY.EMAIL<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.PRIMARY.NAME}

ホストインベントリ内の *Primary POC name* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.POC.PRIMARY.NAME<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.PRIMARY.NOTES}

ホストインベントリ内の *Primary POC notes* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます（例: {INVENTORY.POC.PRIMARY.NOTES<1-9>}）。indexed macrosを参照してください。

{INVENTORY.POC.PRIMARY.PHONE.A}

ホストインベントリ内の *Primary POC phone A* フィールド。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.PRIMARY.PHONE.B}

ホストインベントリ内の *Primary POC phone B* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.POC.PRIMARY.PHONE.B<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.POC.PRIMARY.SCREEN}

ホストインベントリ内の *Primary POC screen name* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.POC.PRIMARY.SCREEN<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.CELL}

ホストインベントリ内の *Secondary POC cell* フィールドです。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.POC.SECONDARY.CELL<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.EMAIL}

ホストインベントリ内の *Secondary POC email* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.POC.SECONDARY.EMAIL<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.NAME}

ホストインベントリ内の *Secondary POC name* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.POC.SECONDARY.NAME<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.NOTES}

ホストインベントリ内の *Secondary POC notes* フィールド。

 このマクロは数値インデックスとともに使用できます。例: {INVENTORY.POC.SECONDARY.NOTES<1-9>}。これにより、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定できます。indexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.PHONE.A}

ホストインベントリ内の *Secondary POC phone A* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}。indexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.PHONE.B}

ホストインベントリ内の *Secondary POC phone B* フィールド。

 このマクロは、トリガー式内で最初、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.POC.SECONDARY.PHONE.B<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.POC.SECONDARY.SCREEN}

ホストインベントリ内の *Secondary POC screen name* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます（例: {INVENTORY.POC.SECONDARY.SCREEN<1-9>}）。詳細はindexed macrosを参照してください。

{INVENTORY.SERIALNO.A}

ホストインベントリのシリアル番号 *A* フィールド。
 {PROFILE.SERIALNO<1-9>} は非推奨であり、サポートされている場所では解決されない可能性があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.SERIALNO.A<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.SERIALNO.B}

ホストインベントリのシリアル番号 *B* フィールドです。

 このマクロは数値インデックスとともに使用できます。例えば、{INVENTORY.SERIALNO.B<1-9>} のように指定すると、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macrosを参照してください。

{INVENTORY.SITE.ADDRESS.A}

ホストインベントリ内の *Site address A* フィールドです。

 このマクロは数値インデックスとともに使用できます。例えば、{INVENTORY.SITE.ADDRESS.A<1-9>} のように指定すると、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macrosを参照してください。

{INVENTORY.SITE.ADDRESS.B}

ホストインベントリ内の *Site address B* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SITE.ADDRESS.B<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.SITE.ADDRESS.C}

ホストインベントリの *Site address C* フィールド。

 このマクロは数値インデックスとともに使用できます。例えば、{INVENTORY.SITE.ADDRESS.C<1-9>} のように指定すると、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指すことができます。indexed macrosを参照してください。

{INVENTORY.SITE.CITY}

ホストインベントリの *Site city* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SITE.CITY<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.SITE.COUNTRY}

ホストインベントリの *Site country* フィールドです。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SITE.COUNTRY<1-9>}。indexed macrosを参照してください。

{INVENTORY.SITE.NOTES}

ホストインベントリ内の *Site notes* フィールドです。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すには、{INVENTORY.SITE.NOTES<1-9>} のように指定します。indexed macrosを参照してください。

{INVENTORY.SITE.RACK}

ホストインベントリの *Site rack location* フィールドです。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SITE.RACK<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.SITE.STATE}

ホストインベントリの都道府県/州フィールドです。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SITE.STATE<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.SITE.ZIP}

ホストインベントリの *Site ZIP/postal* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.SITE.ZIP<1-9>}。indexed macrosを参照してください。

{INVENTORY.SOFTWARE}

ホストインベントリのソフトウェアフィールド。
 {PROFILE.SOFTWARE<1-9>} は非推奨であり、サポートされている場所では解決されない可能性があります。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SOFTWARE<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.SOFTWARE.APP.A}

ホストインベントリ内のソフトウェアアプリケーション *A* フィールドです。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.SOFTWARE.APP.A<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.SOFTWARE.APP.B}

ホストインベントリのソフトウェアアプリケーション *B* フィールド。

 このマクロは、数値インデックスとともに使用できます。例えば、トリガー条件式で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.SOFTWARE.APP.B<1-9>} のように記述します。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.C}

ホストインベントリのソフトウェアアプリケーション *C* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.SOFTWARE.APP.C<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.D}

ホストインベントリ内のソフトウェアアプリケーション *D* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SOFTWARE.APP.D<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.SOFTWARE.APP.E}

ホストインベントリ内のソフトウェアアプリケーション *E* フィールド。

 このマクロは、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.SOFTWARE.APP.E<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.SOFTWARE.FULL}

ホストインベントリ内のソフトウェア（詳細）フィールド。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー条件式内で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.SOFTWARE.FULL<1-9>} のように記述します。インデックス付きマクロを参照してください。

{INVENTORY.TAG}

ホストインベントリのタグフィールド。
 {PROFILE.TAG<1-9>} は非推奨であり、サポートされている場所では解決されない場合があります。

 このマクロは数値インデックスとともに使用できます。例えば、{INVENTORY.TAG<1-9>} はトリガー式内の 1 番目、2 番目、3 番目などのホストを指します。インデックス付きマクロを参照してください。

{INVENTORY.TYPE}

ホストインベントリ内のタイプフィールド。
 {PROFILE.DEVICETYPE<1-9>} は非推奨であり、サポートされている場所では解決されない可能性があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、数値インデックスとともに使用できます。例: {INVENTORY.TYPE<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.TYPE.FULL}

ホストインベントリの *Type (Full details)* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで使用できます。例: {INVENTORY.TYPE.FULL<1-9>}。indexed macros を参照してください。

{INVENTORY.URL.A}

ホストインベントリの *URL A* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指すために、数値インデックス付きで使用できます。例: {INVENTORY.URL.A<1-9>}。詳細はindexed macrosを参照してください。

{INVENTORY.URL.B}

ホストインベントリ内の *URL B* フィールド。

 このマクロは、数値インデックスとともに使用できます。例えば、トリガー式で 1 番目、2 番目、3 番目などのホストを指すために {INVENTORY.URL.B<1-9>} のように指定します。インデックス付きマクロを参照してください。

{INVENTORY.URL.C}

ホストインベントリの *URL C* フィールド。

 このマクロは、トリガー式内で 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックスとともに使用できます。例: {INVENTORY.URL.C<1-9>}。インデックス付きマクロを参照してください。

{INVENTORY.VENDOR}

ホストインベントリの *Vendor* フィールドです。

 このマクロは数値インデックスとともに使用できます。例えば、トリガー式で 1 番目、2 番目、3 番目などのホストを指定するには、{INVENTORY.VENDOR<1-9>} のように記述します。indexed macros を参照してください。

アイテムマクロ

{ITEM.DESCRIPTION}

通知の原因となったトリガー式内の N 番目のアイテムの説明。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定するために、{ITEM.DESCRIPTION<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{ITEM.DESCRIPTION.ORIG}

通知の原因となったトリガー式内の N 番目のアイテムの元の説明（マクロが未解決の状態）。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定するために、数値インデックス付きで {ITEM.DESCRIPTION.ORIG<1-9>} のように使用できます。インデックス付きマクロを参照してください。

{ITEM.ID}

通知の原因となったトリガー式内の N 番目のアイテムの数値 ID です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定するために、{ITEM.ID<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{ITEM.KEY}

通知の原因となったトリガー式内の N 番目のアイテムのキー。
 このマクロが履歴関数の最初のパラメータ内のプレースホルダーとして使用される場合、たとえば `last(/{HOST.HOST}/{ITEM.KEY})` のような場合には、このマクロではマクロ関数はサポートされません。
 `{TRIGGER.KEY}` は非推奨であり、サポートされている場所では解決されない可能性があります。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指すために、{ITEM.KEY<1-9>} のように数値インデックス付きで使用できます。インデックス付きマクロを参照してください。

{ITEM.KEY.ORIG}

通知の原因となったトリガー式内の N 番目のアイテムの元のキー（マクロ未解決）。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのホストを指定するために、{ITEM.KEY.ORIG<1-9>} のように数値インデックス付きで使用できます。indexed macros を参照してください。

{ITEM.LASTVALUE}

通知の原因となったトリガー式内の N 番目のアイテムの最新値です。
 最新の履歴値が *Max history display period* で指定された時間より前に収集された場合（Administration → General メニューセクションで設定）、Web インターフェースでは *UNKNOWN* に展開されます。
 障害名で使用した場合、このマクロは障害イベントの表示時に最新のアイテム値には展開されず、代わりに障害発生時点のアイテム値が保持されます。
 通知で使用した場合、状況によっては、このマクロがトリガー発生時点の最新アイテム値に展開されないことがあります。例えば、あるアイテムが短時間のうちに”A” と”B” の 2 つの値を受信し、トリガーが”A” で発生した場合でも、処理のわずかな遅延により、通知には最新値として”B” が表示されることがあります。これは、トリガー発生時点から通知作成時点までの間に、アイテムの最新値が変化したためです。これを避けるには、トリガー発生時点の値に展開される {ITEM.VALUE} マクロを使用できます。これにより、通知で正しい値が使用されます。
 これは `last(/{HOST.HOST}/{ITEM.KEY})` の別名です。
 text/log アイテムの展開後の値は、以下の場所では Web インターフェースによって 20 文字に切り詰められます。
- 運用データ
- トリガーの説明
- トリガー URL
- トリガー URL ラベル
- アイテム値ウィジェットの説明
 完全な値に展開するには、サーバーでは値が切り詰められないため、マクロ関数を使用できます。例: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックス付きで使用できます。例: {ITEM.LASTVALUE<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.AGE}

最新のアイテム値の収集からマクロ評価までの間に経過した時間。
 人間が読みやすい形式（例: 1m 45s）で表示されます。

 このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.LASTVALUE.AGE<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.DATE}

最新のアイテム値が収集された日付です。
 YYYYMMDD 形式で表示されます。
 トリガー名（監視 -> 障害のリスト内）、イベント名、タグ名およびタグ値ではサーバーのタイムゾーンが使用されます。それ以外のすべての場合では、ユーザーのタイムゾーンが使用されます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {ITEM.LASTVALUE.DATE<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LASTVALUE.TIME}

最新のアイテム値が収集された時刻です。
 HHMMSS 形式で表示されます。
 トリガー名（Monitoring -> Problems リスト内）、イベント名、タグ名およびタグ値では、サーバーのタイムゾーンが使用されます。その他のすべての場合では、ユーザーのタイムゾーンが使用されます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.LASTVALUE.TIME<1-9>}。indexed macrosを参照してください。

{ITEM.LASTVALUE.TIMESTAMP}

最新のアイテム値が収集された時刻の UNIX タイムスタンプです。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス付きで使用できます。例: {ITEM.LASTVALUE.TIMESTAMP<1-9>}。indexed macrosを参照してください。

{ITEM.LOG.AGE}

ログイベントの経過時間です。
 秒単位の精度です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.LOG.AGE<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LOG.DATE}

ログエントリがログに書き込まれた日付です。

 このマクロは数値インデックスとともに使用できます。例えば、{ITEM.LOG.DATE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.LOG.EVENTID}

イベントログ内のイベントの ID。
 Windows イベントログの監視でのみ使用できます。

 このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {ITEM.LOG.EVENTID<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LOG.NSEVERITY}

イベントログ内のイベントの数値の重大度。
 Windows イベントログの監視でのみ使用できます。

 このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {ITEM.LOG.NSEVERITY<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LOG.SEVERITY}

イベントログ内のイベントの重大度を表す文字列表現です。
 Windows イベントログの監視でのみ使用できます。

 このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.LOG.SEVERITY<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LOG.SOURCE}

イベントログ内のイベントのソースです。
 Windows イベントログの監視でのみ使用できます。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {ITEM.LOG.SOURCE<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LOG.TIME}

ログエントリがログに書き込まれた時刻です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {ITEM.LOG.TIME<1-9>}。インデックス付きマクロを参照してください。

{ITEM.LOG.TIMESTAMP}

ログエントリがログに書き込まれた時刻の UNIX タイムスタンプです。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {ITEM.LOG.TIMESTAMP<1-9>}。インデックス付きマクロを参照してください。

{ITEM.NAME}

すべてのマクロが展開されたアイテムの名前です。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.NAME<1-9>}。インデックス付きマクロを参照してください。

{ITEM.NAME.ORIG}

アイテムの元の名前（マクロが未展開の状態）。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス付きで使用できます。例: {ITEM.NAME.ORIG<1-9>}。インデックス付きマクロを参照してください。

{ITEM.STATE}

通知の原因となったトリガー式内の N 番目のアイテムの最新の状態。
 使用可能な値: 未サポート, 正常。

 このマクロは、数値インデックスとともに使用できます。例えば、{ITEM.STATE<1-9>} は、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.STATE.ERROR}

アイテムが未サポートになった理由の詳細を含むエラーメッセージです。
 アイテムが未サポート状態になった後、すぐに再びサポート状態に戻った場合、エラーフィールドは空になることがあります。

{ITEM.VALUE}

次のいずれかに展開されます。

- トリガーのステータス変更のコンテキストで使用された場合、トリガー式内の N 番目のアイテムの履歴上の値（イベント発生時点の値）。例えば、イベントの表示時や通知の送信時などです。
- トリガーのステータス変更のコンテキストなしで使用された場合、トリガー式内の N 番目のアイテムの最新値。例えば、ポップアップ選択ウィンドウでトリガーの一覧を表示する場合などです。この場合、{ITEM.LASTVALUE} と同じように動作します。

最初のケースでは、履歴値がすでに削除されているか、一度も保存されていない場合、*UNKNOWN* に展開されます。
2 番目のケースでは、かつ Web インターフェースでのみ、最新の履歴値が *Max history display period* で指定された時間より前に収集されたものである場合、*UNKNOWN* に展開されます（Administration→General メニューセクションで設定）。
text/log アイテムの展開値は、Web インターフェースの以下の場所で 20 文字に切り詰められます。
- 運用データ;
- トリガーの説明;
- トリガー URL;
- トリガー URL ラベル;
- アイテム値ウィジェットの説明。
 完全な値に展開するには、サーバーでは値が切り詰められないため、マクロ関数を使用できます。例: {{ITEM.VALUE}.regsub("(.*)", \1)}

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックス付きで使用できます。例: {ITEM.VALUE<1-9>}。インデックス付きマクロを参照してください。

{ITEM.VALUE.AGE}

アイテム値の収集からマクロ評価までの間に経過した時間。
 人が読みやすい形式で表示されます（例: 1m 45s）。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.VALUE.AGE<1-9>}。indexed macrosを参照してください。

{ITEM.VALUE.DATE}

アイテムの値が収集された日付です。
 YYYYMMDD 形式で表示されます。
 トリガー名（監視 -> 障害リスト内）、イベント名、タグ名およびタグ値ではサーバーのタイムゾーンが使用されます。それ以外のすべての場合では、ユーザーのタイムゾーンが使用されます。

 このマクロは、数値インデックスとともに使用できます。例えば {ITEM.VALUE.DATE<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを参照できます。indexed macros を参照してください。

{ITEM.VALUE.TIME}

アイテムの値が収集された時刻です。
 HHMMSS 形式で表示されます。
 トリガー名（監視 -> 障害一覧内）、イベント名、タグ名およびタグ値では、サーバーのタイムゾーンが使用されます。それ以外の場合は、ユーザーのタイムゾーンが使用されます。

 このマクロは、数値インデックスとともに使用できます。例えば {ITEM.VALUE.TIME<1-9>} のように指定すると、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指します。indexed macros を参照してください。

{ITEM.VALUE.TIMESTAMP}

アイテムの値が収集された時刻の UNIX タイムスタンプです。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.VALUE.TIMESTAMP<1-9>}。詳細はindexed macrosを参照してください。

{ITEM.VALUETYPE}

通知の原因となったトリガー式内の N 番目のアイテムの値のタイプです。
 使用可能な値: *0* - 数値（浮動小数）, *1* - 文字列, *2* - ログ, *3* - 数値（符号なし）, *4* - テキスト。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指定するために、数値インデックスとともに使用できます。例: {ITEM.VALUETYPE<1-9>}。indexed macrosを参照してください。

ローレベルディスカバリ通知マクロ

{LLDRULE.DESCRIPTION}

通知を発生させたローレベルディスカバリルールの説明。

{LLDRULE.DESCRIPTION.ORIG}

通知を発生させたローレベルディスカバリルールの元の説明（マクロが未解決の状態）。

{LLDRULE.ID}

通知を発生させたローレベルディスカバリルールの数値 ID。

{LLDRULE.KEY}

通知の原因となったローレベルディスカバリルールのキー。

{LLDRULE.KEY.ORIG}

通知を発生させたローレベルディスカバリルールの元のキー（マクロが未解決のもの）。

{LLDRULE.NAME}

通知を発生させたローレベルディスカバリルールの名前。

{LLDRULE.NAME.ORIG}

通知を発生させたローレベルディスカバリルールの元の名前（マクロが未解決の状態）。

{LLDRULE.STATE}

ローレベルディスカバリルールの最新の状態です。
 使用可能な値: 未サポート、正常。

 このマクロは、トリガー式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックスとともに使用できます。例: {LLDRULE.STATE<1-9>}。indexed macrosを参照してください。

{LLDRULE.STATE.ERROR}

ローレベルディスカバリルールが未サポートになった理由の詳細を示すエラーメッセージです。
 ローレベルディスカバリルールが未サポート状態になった後、すぐに再びサポート状態に戻った場合、エラーフィールドは空になることがあります。

マップマクロ

{MAP.ID}

ネットワークマップ ID。

{MAP.NAME}

ネットワークマップ名。

プロキシマクロ

{PROXY.DESCRIPTION}

プロキシの説明です。
 以下のいずれかに解決されます。- トリガー条件式内の N 番目のアイテムのプロキシ（トリガーベースの通知の場合）。ここではindexedマクロを使用できます。- ディスカバリを実行したプロキシ（ディスカバリ通知の場合）。ここではインデックスなしで {PROXY.DESCRIPTION} を使用します。- アクティブエージェントが登録した先のプロキシ（自動登録通知の場合）。ここではインデックスなしで {PROXY.DESCRIPTION} を使用します。

このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス付きで使用できます（例: {PROXY.DESCRIPTION<1-9>}）。indexed macrosを参照してください。

{PROXY.NAME}

プロキシ名です。
 これは次のいずれかに展開されます。- トリガー条件式内の N 番目のアイテムのプロキシ（トリガーベースの通知の場合）。ここではindexedマクロを使用できます。- ディスカバリを実行したプロキシ（ディスカバリ通知の場合）。ここでは、インデックスなしで {PROXY.NAME} を使用します。- アクティブエージェントが登録した先のプロキシ（自動登録通知の場合）。ここでは、インデックスなしで {PROXY.NAME} を使用します。

このマクロは、トリガー条件式内の 1 番目、2 番目、3 番目などのアイテムを指すために、数値インデックス付きで使用できます。例: {PROXY.NAME<1-9>}。indexed macrosを参照してください。

スクリプトマクロ

{MANUALINPUT}

スクリプト実行時にユーザーが指定した手動入力値。

サービスマクロ

{SERVICE.DESCRIPTION}

マクロが展開されたサービスの説明。

{SERVICE.ID}

アクションをトリガーしたサービスの数値 ID。

{SERVICE.NAME}

マクロが解決されたサービス名。

{SERVICE.ROOTCAUSE}

サービスの障害の原因となったトリガーの障害イベントの一覧です。重要度およびホスト名でソートされます。
 以下の詳細が含まれます: ホスト名、イベント名、重要度、経過時間、サービスのタグと値。

{SERVICE.TAGS}

サービスイベントタグのカンマ区切りリストです。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。タグが存在しない場合は空文字列に展開されます。

{SERVICE.TAGSJSON}

サービスイベントタグオブジェクトを含む JSON 配列です。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。タグが存在しない場合は、空の配列に展開されます。

{SERVICE.TAGS.<tag name>}

タグ名によって参照されるサービスイベントタグの値です。
 サービスイベントタグは、サービス設定のタグセクションで定義できます。
 英数字以外の文字（英語以外のマルチバイト UTF 文字を含む）を含むタグ名は、二重引用符で囲む必要があります。引用符で囲まれたタグ名の中の引用符およびバックスラッシュは、バックスラッシュでエスケープする必要があります。

トリガーマクロ

{TRIGGER.DESCRIPTION}

トリガーの説明です。
 通知テキストで `{TRIGGER.DESCRIPTION}` を使用すると、トリガーの説明でサポートされているすべてのマクロが展開されます。
 `{TRIGGER.COMMENT}` は非推奨であり、サポートされている場所でも解決されない場合があります。

{TRIGGER.EVENTS.ACK}

マップ内のマップ要素についての確認済みイベント数、または通知において現在のイベントを生成したトリガーの確認済みイベント数。

{TRIGGER.EVENTS.PROBLEM.ACK}

状態に関係なく、すべてのトリガーにおける確認済みの障害イベント数。

{TRIGGER.EVENTS.PROBLEM.UNACK}

状態に関係なく、すべてのトリガーにおける未確認の障害イベント数。

{TRIGGER.EVENTS.UNACK}

マップではマップ要素の未確認イベント数、通知では現在のイベントを生成したトリガーの未確認イベント数。

{TRIGGER.EXPRESSION}

トリガーの条件式です。

{TRIGGER.EXPRESSION.EXPLAIN}

部分的に評価されたトリガー式です。
 アイテムベースの関数はイベント生成時点の結果に評価・置換され、その他のすべての関数は式に記述されたとおりに表示されます。トリガー式のデバッグに使用できます。

{TRIGGER.EXPRESSION.RECOVERY}

トリガー設定 の *OK event generation* が「Recovery expression」に設定されている場合のトリガーのリカバリー条件式です。それ以外の場合は空文字列が返されます。

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

部分的に評価されたトリガーのリカバリー条件式。
 アイテムベースの関数はイベント生成時点の結果に評価・置換され、その他のすべての関数は条件式に記述されたとおりに表示されます。トリガーのリカバリー条件式のデバッグに使用できます。

{TRIGGER.HOSTGROUP.NAME}

トリガーが定義されているホストグループの、SQL クエリによってソートされたカンマとスペース区切りのリスト。

{TRIGGER.ID}

このアクションを実行したトリガーの数値 ID です。

{TRIGGER.NAME}

マクロが展開されたトリガー名です。
 アクションでは、`{EVENT.NAME}` を使用して、マクロが展開されたトリガー済みのイベント/障害名を表示できることに注意してください。

{TRIGGER.NAME.ORIG}

元のトリガー名（マクロは未展開）。

{TRIGGER.NSEVERITY}

トリガーの重大度を数値で表したものです。
 指定可能な値: *0* - 未分類、*1* - 情報、*2* - 警告、*3* - 平均、*4* - 高、*5* - 障害。

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

障害状態のトリガーに対する、確認済みの障害イベント数。

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

障害状態にあるトリガーの未確認の障害イベント数。

{TRIGGER.SEVERITY}

トリガーの深刻度名です。
 Administration > *General* > *Trigger displaying options* で定義できます。

{TRIGGER.STATE}

トリガーの最新の状態です。
 指定可能な値: *Unknown*、*Normal*。

{TRIGGER.STATE.ERROR}

トリガーが未サポートになった理由の詳細を含むエラーメッセージです。
 トリガーが未サポート状態になった後、すぐに再びサポート状態に戻った場合、エラーフィールドが空になることがあります。

{TRIGGER.STATUS}

操作ステップの実行時点におけるトリガーの値です。
 使用可能な値: *Problem*、*OK*。
 {STATUS} は非推奨であり、サポートされている場所でも解決されない場合があります。

{TRIGGER.TEMPLATE.NAME}

トリガーが定義されているテンプレートの、ソート済み（SQL クエリによる）でカンマとスペース区切りのリストです。トリガーがホストで定義されている場合は、*UNKNOWN* になります。

{TRIGGER.URL}

トリガー URL。

{TRIGGER.URL.NAME}

トリガー URL のラベル。

{TRIGGER.VALUE}

現在の数値トリガー値です。
 使用可能な値: *0* - トリガーが OK 状態、*1* - トリガーが障害状態。

{TRIGGERS.ACK}

トリガーの状態に関係なく、マップ要素の確認済みトリガーの数。
 トリガーは、そのすべての障害イベントが確認済みである場合に、確認済みと見なされます。

{TRIGGERS.PROBLEM.ACK}

マップ要素の確認済みの障害トリガー数。
 トリガーは、そのすべての障害イベントが確認済みである場合に、確認済みと見なされます。

{TRIGGERS.PROBLEM.UNACK}

マップ要素の未確認の障害トリガー数。
 トリガーは、その障害イベントのうち少なくとも 1 つが未確認である場合、未確認と見なされます。

{TRIGGERS.UNACK}

マップ要素の未確認トリガーの数です。トリガーの状態は考慮されません。
 トリガーは、その障害イベントのうち少なくとも 1 つが未確認である場合、未確認と見なされます。

ユーザー名マクロ

{USER.FULLNAME}

イベントの障害対応を追加した、またはスクリプトを開始したユーザーの名前、姓、ユーザー名。

{USER.NAME}

スクリプトを開始したユーザーの名前。

{USER.SURNAME}

スクリプトを開始したユーザーの姓。

{USER.USERNAME}

スクリプトを開始したユーザーのユーザー名。
 {USER.ALIAS} は現在非推奨であり、サポートされている場所では解決されない可能性があります。

インデックス付きマクロ

インデックス付きマクロ構文 **{MACRO<1-9>}** は、トリガーの *Expression* フィールド内の N 番目のアイテム、関数、またはホストのみを参照できます。

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** は、トリガー式内の 1 番目、2 番目、3 番目のホストの IP アドレスに展開されます（存在する場合）。
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** は、イベント発生時点におけるトリガー式内の 1 番目、2 番目、3 番目のアイテムの値に展開されます（存在する場合）。
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** は、イベント発生時点における 1 番目、2 番目、3 番目のアイテムベースの関数の値に展開されます（存在する場合）。

トリガーの文脈では、インデックス付きマクロは常にトリガー設定の *Expression* フィールドを参照し、*Recovery expression* は参照しません。例えば、リカバリイベントでは、**{ITEM.VALUE2}** はリカバリ時点における障害式の 2 番目のアイテムの値に展開されます。

{HOST.HOST<1-9>} マクロは、**graph names** 内の {?func(/host/key,param)} 式マクロでもサポートされています。例えば、グラフ名内の {?func(/{HOST.HOST2}/key,param)} は、グラフ内の 2 番目のアイテムのホストに展開されます。

Warning:
インデックス付きマクロは、ここで説明したケース以外の文脈では展開されません。その他の文脈では、インデックスなしのマクロ（**{HOST.HOST}, {HOST.IP}** など）を使用してください。

2 ロケーションでサポートされているユーザーマクロ

概要

このセクションには、ユーザー定義マクロがサポートされている場所の一覧が含まれています。

Note:
アクション、ネットワークディスカバリ、プロキシ、およびこのページのその他の場所セクションに記載されているすべての場所では、グローバルレベルのユーザーマクロのみがサポートされています。上記の場所では、ホストレベルおよびテンプレートレベルのマクロは展開されません。

Note:
マクロ値をカスタマイズするには（たとえば、特定の部分文字列を短縮または抽出する場合）、マクロ関数を使用できます。

アクション

アクションでは、ユーザーマクロを次のフィールドで使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
トリガーベースの通知とコマンド		yes
トリガーベースの内部通知		yes
障害の更新通知		yes
サービスベースの通知とコマンド		yes
サービス更新通知		yes
期間		no
オペレーション		
	デフォルトの操作ステップ期間	no
	ステップ期間	no

ホスト/ホストプロトタイプ

ホスト および ホストプロトタイプ の設定では、ユーザーマクロを次のフィールドで使用できます。

場所	複数のマクロ/テキストとの混在[1]
インターフェースの IP/DNS	DNS のみ

場所		複数のマクロ/テキストとの混在[1]
インターフェースのポート		いいえ
SNMP v1、v2		
	SNMP コミュニティ	はい
SNMP v3		
	コンテキスト名	はい
	セキュリティ名	はい
	認証パスフレーズ	はい
	プライバシーパスフレーズ	はい
IPMI		
	ユーザー名	はい
	パスワード	はい
タグ[2]		
	タグ名	はい
	タグ値	はい

アイテム / アイテムのプロトタイプ

アイテム またはアイテムのプロトタイプ の設定では、ユーザーマクロを以下のフィールドで使用できます。

場所
アイテム名
アイテムキーのパラメータ
更新間隔
カスタム間隔
タイムアウト（サポートされているアイテムタイプで利用可能）
保存期間（履歴およびトレンド）
説明
計算/集計アイテム
数式（式定数および関数パラメータ、アイテムキーのパラメータ、(集計アイ
データベースモニタ
ユーザー名
パスワード
SQL クエリ
HTTP エージェント
URL[3]
クエリフィールド
リクエストボディ
ヘッダー（名前と値）
必要なステータスコード
HTTP プロキシ
HTTP 認証ユーザー名
HTTP 認証パスワード
SSI 証明書ファイル
SSI キーファイル
SSI キーパスワード
許可されたホスト
JMX エージェント
JMX エンドポイント
スクリプトアイテム
パラメータ名と値
ブラウザアイテム
パラメータ名と値
SNMP エージェント
SNMP OID
SSH エージェント
ユーザー名
公開鍵ファイル
秘密鍵ファイル
パスワード
スクリプト

場所	
TELNET エージェント	
	ユーザー名
	パスワード
	スクリプト
Zabbix トラッパー	
	許可されたホスト
タグ[2]	
	タグ名
	タグ値
前処理ステップ	
	パラメータ（カスタムスクリプトを含む）
	カスタムエラー処理パラメータ（値を設定および エラーを設定フィールド）

ローレベルディスカバリ

ローレベルディスカバリルールでは、ユーザーマクロを以下のフィールドで使用できます。

場所		複数のマクロ/テキスト
キーのパラメータ		はい
更新間隔		いいえ
カスタム間隔		いいえ
タイムアウト（サポートされているアイテムタイプで使用可能）		いいえ
失われたリソースの削除		いいえ
失われたリソースの無効化		いいえ
説明		はい
SNMP エージェント		
	SNMP OID	はい
SSH エージェント		
	ユーザー名	はい
	公開鍵ファイル	はい
	秘密鍵ファイル	はい
	パスワード	はい
	スクリプト	はい
TELNET エージェント		
	ユーザー名	はい
	パスワード	はい
	スクリプト	はい
Zabbix トラッパー		
	許可されたホスト	はい
データベースモニタ		
	ユーザー名	はい
	パスワード	はい
	SQL クエリ	はい
JMX エージェント		
	JMX エンドポイント	はい
HTTP エージェント		
	URL[3]	はい
	クエリフィールド	はい
	リクエストボディ	はい
	ヘッダー（名前と値）	はい
	必要なステータスコード	はい
	HTTP 認証のユーザー名	はい
	HTTP 認証のパスワード	はい
フィルター		
	正規表現	はい
オーバーライド		
	フィルター: 正規表現	はい
	操作: 更新間隔（アイテムプロトタイプ用）	いいえ
	操作: 履歴保存期間（アイテムプロトタイプ用）	いいえ
	操作: トレンド保存期間（アイテムプロトタイプ用）	いいえ

ネットワークディスカバリ

ネットワークディスカバリルール では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
監視間隔		no
SNMP v1, v2		
	SNMP コミュニティ	yes
	SNMP OID	yes
SNMP v3		
	コンテキスト名	yes
	セキュリティ名	yes
	認証パスフレーズ	yes
	プライバシーパスフレーズ	yes
	SNMP OID	yes

プロキシ

プロキシ の設定では、ユーザーマクロを以下のフィールドで使用できます。

場所	複数のマクロ/テキストとの混在[1]
Address for active agents > *Port*（プロキシがグループに属している場合）	いいえ
インターフェースのアドレスとポート（パッシブプロキシの場合）	いいえ
アイテムタイプのタイムアウト	いいえ

プロキシグループ

プロキシグループ 構成では、以下のフィールドでユーザーマクロを使用できます。

場所	複数のマクロ/テキストとの混在[1]
フェイルオーバー期間	いいえ
プロキシの最小数	いいえ

テンプレート

テンプレートの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混在[1]
タグ[2]		
	タグ名	yes
	タグ値	yes

トリガー

トリガーの設定では、ユーザーマクロを以下のフィールドで使用できます。

場所		複数のマクロ/テキストとの混在[1]
名前		はい
運用データ		はい
条件式（定数および関数パラメータ内のみ。シークレットマクロはサポートされません）		はい
一致用タグ		はい
メニューエントリ名		はい
メニューエントリ URL[3]		はい
説明		はい
タグ[2]		
	タグ名	はい
	タグの値	はい

Web シナリオ

Web シナリオの設定では、次のフィールドでユーザーマクロを使用できます。

ロケーション		複数のマクロ/テキストとの混合[1]
名前		yes
監視間隔		no
エージェント		yes
HTTP プロキシ		yes
変数 (値のみ)		yes
ヘッダー (名前と値)		yes
ステップ		
	名前	yes
	URL[3]	yes
	変数 (値のみ)	yes
	ヘッダー (名前と変数)	yes
	タイムアウト	no
	要求文字列	yes
	要求ステータスコード	no
認証		
	ユーザー	yes
	パスワード	yes
	SSL 証明書ファイル	yes
	SSL 秘密鍵ファイル	yes
	SSL 秘密鍵パスワード	yes
タグ[2]		
	タグ名	yes
	タグ値	yes

その他の場所

ここに記載されている場所に加えて、ユーザーマクロは以下のフィールドでも使用できます。

場所	複数のマクロ/テキストとの混在[1]
グローバルスクリプト(URL、スクリプト、SSH、Telnet、IPMI)、確認テキストを含むwebhook	はい
JavaScript スクリプト	いいえ
JavaScript スクリプトパラメータ名	いいえ
JavaScript スクリプトパラメータ値	はい
ダッシュボード	
Top hosts ダッシュボードウィジェットのデータ型 *Text* の列	はい
Item value および *Gauge* ダッシュボードウィジェットの *Description* パラメータ	はい
Honeycomb ダッシュボードウィジェットの Primary/Secondary label の *Text* パラメータ	はい
URL ダッシュボードウィジェットの *URL*[3] パラメータ	はい
ユーザー→ユーザー→メディア	
アクティブ時間	いいえ

場所		複数のマクロ/テキストとの混在[1]
管理→一般→*GUI*	稼働時間	いいえ
管理→一般→タイムアウト	アイテムタイプのタイムアウト	いいえ
管理→一般→コネクタ	URL	はい
	ユーザー名	はい
	パスワード	はい
	Bearer トークン	はい
	タイムアウト	いいえ
	HTTP プロキシ	はい
	SSL 証明書ファイル	はい
	SSL キーファイル	はい
	SSL キーのパスワード	はい

場所		複数のマクロ/テキストとの混在[1]
アラート→メディアタイプ→メッセージテンプレート	件名	はい
	メッセージ	はい
アラート→メディアタイプ→スクリプト	スクリプトパラメータ	はい

場所		複数のマクロ/テキストとの混在[1]
アラート→メディアタイプ→メディアタイプ	*Email* メディアタイプの *Username* および *Password* フィールド（*Authentication* が「Username and password」に設定されている場合。secret macros の使用を推奨）	はい

Zabbix でサポートされているすべてのマクロの完全な一覧については、サポートされているマクロを参照してください。

脚注

[1] フィールド内の複数のマクロまたはテキストと混合されたマクロがそのロケーションでサポートされていない場合は、1 つのマクロでフィールド全体を埋める必要があります。

[2] タグ名と値で使用されるマクロは、イベント生成プロセス中にのみ解決されます。

[3] シークレットマクロを含む URL は機能しません。それらのマクロは”******” のように解決されます。

7 単位記号

概要

86400、104857600、1000000 のような大きな値を扱うのは難しく、エラーの原因になることがあります。設定を簡素化し、可読性を向上させるために、Zabbix は値の乗数として機能する単位記号（接尾辞）をサポートしています。

接尾辞なしのトリガー式:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

接尾辞ありのトリガー式:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

接尾辞は、アイテム、ウィジェットなど、他のエンティティの設定を簡素化するためにも使用でき、アイテムの値を人間にとって読みやすい形式で表示するのにも役立ちます。

Note:
エンティティの設定フィールドが接尾辞をサポートしているかどうかを確認するには、必ず設定対象のエンティティに対応するページを参照してください。

時間のサフィックス

Zabbix Web インターフェースは、エンティティ設定で以下の時間サフィックスをサポートしています。

- **s** - 秒 *(使用した場合、生の値と同じように機能します)*
- **m** - 分
- **h** - 時間
- **d** - 日
- **w** - 週
- **M** - か月 *(trend functions のみ)*
- **y** - 年 *(trend functions のみ)*

時間サフィックスは整数値でのみサポートされています。
例えば、1h はサポートされていますが、1,5h や 1.5h はサポートされていません。代わりに 90m を使用してください。

メモリサイズの接尾辞

Zabbix は、以下のメモリサイズの接尾辞をサポートしています。

- **K** - キロバイト
- **M** - メガバイト
- **G** - ギガバイト
- **T** - テラバイト

アイテム値のサフィックス

サフィックスは、数値のアイテム値を人間にとって読みやすい形式で表示するためにも使用できます。

これを有効にするには、アイテムの設定時に、*Units* フィールドで以下のいずれかのサフィックスを使用します。

- **B** - バイト
- **Bps** - 1 秒あたりのバイト数
- **s** - 秒。最大 3 つまでの最も大きい非ゼロの時間単位を使用して表示
- **uptime** - hh:mm:ss または N days, hh:mm:ss 形式の経過時間
- **unixtime** - Unix タイムスタンプ。yyyy.mm.dd hh:mm:ss 形式で表示

これらのサフィックスの解釈および表示方法には、以下の追加ルールが適用されます。

- B および Bps については、Zabbix は JEDEC 標準に従い、2 進数ベースの変換を使用します（1K = 1024B）。
- その他の単位（Hz、W など）については、Zabbix は 10 進数ベースの変換を使用します（1K = 1000）。
- s（秒）について:
  - 形式には yyy mmm ddd hhh mmm sss ms が含まれ、最大 3 つまでの最も大きい非ゼロの時間単位のみが表示されます（例: 1M 10d 4h）。
  - ある単位がゼロで、かつ 2 つの非ゼロ単位の間にある場合は省略されます（例: 10d 0h 56m ではなく 10d 56m）。

Units を使用すると、以下の乗数サフィックスがアイテム値に自動的に適用されます。

- **K**, **M**, **G**, **T** - キロ、メガ、ギガ、テラ
- **P**, **E**, **Z**, **Y** - ペタ、エクサ、ゼタ、ヨタ *(これらは Web インターフェースでのみ適用されます)*

単位変換を防ぐには、! プレフィックスを使用します（例: !B または!s）。

以下の例は、受信したアイテム値が指定された単位に基づいてどのように変換されるかを示しています。

```
1 B → 1 B
1024 B → 1 KB
1536 B → 1.5 KB
881764 B → 881.76 KB
881764 !B → 881764 B

0.0000155 s → 0.016ms
3470400 s → 1M 10d 4h
2606400 s → 1M 4h
2592000 s → 1M
2592001 s → 1M
2592001 !s → 2592001 s

17764 uptime → 04:56:04
86400 uptime → 1 day, 00:00:00
881764 uptime → 10 days, 04:56:04
32417764 uptime → 375 days, 04:56:04
32417764 !uptime → 32417764 uptime
```

```
881764 unixtime → 1970-01-11 04:56:04 AM

17764 Hz → 17.76 KHz
86400 Hz → 86.4 KHz
881764 Hz → 881.76 KHz
32417764 Hz → 32.42 MHz

0 ! → 0
0 !! → 0 !
```

Note:
Zabbix 4.0 より前は、ms、rpm、RPM、% からなるハードコードされた単位の停止リストがありました。この停止リストは非推奨となったため、このような単位の変換を防ぐ正しい方法は !ms、!rpm、!RPM、!% です。

8 日時フォーマット

概要

日時を設定するには、次の形式を使用する必要があります。

`d-d,hh:mm-hh:mm`

記号は次のことを表します。

記号	説明
d	曜日: 1 - 月、2 - 火、...、7 - 日
hh	時: 00-24
mm	分: 00-59

セミコロン (;) をセパレーターとして複数の期間を指定できます。

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

空白ではデフォルト値である 1-7,00:00-24:00 になります。

Attention:
指定時間の上限は含まれていません。したがって、09:00-18:00 を指定すると、最後の 1 秒は 17:59:59 になります。

例

営業時間月～金 9:00～18:00

`1-5,09:00-18:00`

平日営業時間と週末営業時間月～金 9:00～18:00 土～日 10:00～16:00

`1-5,09:00-18:00; 6-7,10:00-16:00`

9 コマンドの実行

Zabbix は、外部チェック、ユーザーパラメーター、system.run アイテム、カスタムアラートスクリプト、リモートコマンド、およびグローバルスクリプトに共通の機能を使用します。

実行手順

Note:
デフォルトでは、Zabbix のすべてのスクリプトは *sh* シェルを使用して実行され、デフォルトのシェルを変更することはできません。別のシェルを使用するには、回避策としてスクリプトファイルを作成し、コマンド実行時にそのスクリプトを呼び出します。

コマンド/スクリプトは、Unix と Windows の両プラットフォームで同様に実行されます。

1. Zabbix（親プロセス）が通信用のパイプを作成します
2. Zabbix は、そのパイプをこれから作成される子プロセスの出力先として設定します
3. Zabbix が子プロセスを作成します（コマンド/スクリプトを実行します）
4. 子プロセス用に新しいプロセスグループ（Unix）またはジョブ（Windows）が作成されます
5. Zabbix は、タイムアウトが発生するか、または反対側に書き込むプロセスがいなくなるまでパイプから読み取ります（すべてのハンドル/ファイルディスクリプタが閉じられるまで）。子プロセスは追加のプロセスを作成し、それらのプロセスが終了したりハンドル/ファイルディスクリプタを閉じたりする前に終了する場合があることに注意してください。
6. タイムアウトに達していない場合、Zabbix は最初の子プロセスが終了するか、タイムアウトが発生するまで待機します
7. 最初の子プロセスが終了しており、かつタイムアウトに達していない場合、Zabbix は最初の子プロセスの終了コードを確認し、0 と比較します（0 以外の値は実行失敗と見なされます。これは、カスタムアラートスクリプト、リモートコマンド、および Zabbix server と Zabbix proxy 上で実行されるユーザースクリプトにのみ適用されます）
8. この時点で、すべての処理が完了したと見なされ、プロセスツリー全体（つまり、プロセスグループまたはジョブ）が終了されます

Attention:
Zabbix は、最初の子プロセスが終了しており、かつ他のどのプロセスも出力ハンドル/ファイルディスクリプタを開いたままにしていない場合に、コマンド/スクリプトの処理が完了したと見なします。処理が完了すると、作成されたすべてのプロセスが終了されます。

コマンド内のすべての二重引用符とバックスラッシュは、バックスラッシュでエスケープされ、コマンド全体は二重引用符で囲まれます。

終了コードのチェック

終了コードは、以下の条件でチェックされます。

- Zabbix サーバーおよび Zabbix プロキシ上で実行されるカスタムアラートスクリプト、リモートコマンド、ユーザースクリプトに対してのみ適用されます。
- 0 以外の終了コードは、すべて実行失敗と見なされます。
- 実行に失敗した場合の標準エラー出力および標準出力の内容は収集され、Web インターフェースで利用できます（実行結果が表示される場所）。
- Zabbix エージェント/プロキシ上で実行されるリモートコマンドについては、agent/proxy の LogRemoteCommands パラメータを有効にすることで、追加のログエントリが作成される場合があります。

失敗したコマンド/スクリプトに対して、Web インターフェースに表示される可能性のあるメッセージおよびログエントリ:

- 実行に失敗した場合の標準エラー出力および標準出力の内容（存在する場合）。
- 「Process exited with code: N.」（出力が空で、終了コードが 0 以外の場合）。
- 「Process killed by signal: N.」（シグナルによってプロセスが終了した場合、Linux のみ）。
- 「Process terminated unexpectedly.」（不明な理由でプロセスが終了した場合）。

関連項目

- 外部チェック
- ユーザーパラメータ
- system.run アイテム
- カスタムアラートスクリプト
- リモートコマンド
- グローバルスクリプト

10 バージョン間の互換性

サポートされるエージェント

Zabbix 7.4 と互換性を持たせるには、Zabbix エージェントはバージョン 1.4 より古くてはならず、7.4 より新しくてもなりません。

一部のパラメータが変更されているため、古いエージェントの設定を見直す必要がある場合があります。たとえば、3.0 より前のバージョンでは、logging に関連するパラメータが該当します。

最新の機能、メトリクス、改善されたパフォーマンス、およびメモリ使用量の削減を最大限に活用するには、最新のサポート対象エージェントを使用してください。

Windows XP に関する注意

- 32 ビット版 Windows XP では、6.0.x より新しい Zabbix エージェントを使用しないでください。

- Windows XP/Server 2003 では、Zabbix 4.0.x より新しいエージェントテンプレートを使用しないでください。新しいテンプレートでは英語のパフォーマンスカウンターが使用されており、これがサポートされているのは Windows Vista/Server 2008 以降のみです。

サポートされるエージェント 2

バージョン 4.4 以降の旧しい Zabbix エージェント 2 は Zabbix 7.4 と互換性があります。Zabbix エージェント 2 のバージョンは 7.4 を超えてはいけません。

Zabbix エージェント 2 のバージョン 4.4 および 5.0 を使用する場合、未サポートのアイテムを再確認する際のデフォルト間隔として 10 分が使用されることに注意してください。

最新の機能、メトリクス、パフォーマンス向上、およびメモリ使用量の削減を最大限に活用するには、サポートされている最新のエージェント 2 を使用してください。

サポートされる Zabbix プロキシ

Zabbix 7.4 との完全な互換性を確保するには、プロキシはサーバーのメジャーバージョンと一致している必要があります。
Zabbix 7.4.x プロキシのみが、Zabbix 7.4.x サーバーと完全に互換性があります。

古いプロキシは部分的にサポートされています。引き続きデータの収集やスクリプトの実行は可能ですが、新しいアイテムなどの設定更新を受信することはできません。

Zabbix サーバーとの関係において、プロキシは次のいずれかになります。

- *Current*（プロキシとサーバーのメジャーバージョンが同じ）;
- *Outdated*（プロキシのバージョンがサーバーのバージョンより古いが、部分的にサポートされる）;
- *Unsupported*（プロキシのバージョンが、サーバーの 1 つ前の LTS リリースのバージョンより古い、またはプロキシのバージョンがサーバーのメジャーバージョンより新しい）。

例:

Server version	*Current* proxy version	*Outdated* proxy version	*Unsupported* proxy version
6.4	6.4	6.0、6.2	6.0 より古い; 6.4 より新しい
7.0	7.0	6.0、6.2、6.4	6.0 より古い; 7.0 より新しい
7.2	7.2	7.0	7.0 より古い; 7.2 より新しい
7.4	7.4	7.0	7.0 より古い; 7.4 より新しい

プロキシでサポートされる機能:

プロキシのバージョン	データ更新	設定更新	タスク
Current	はい	はい	はい
Outdated	はい	いいえ	リモートコマンド（例: シェルスクリプト）; アイテムの値の即時チェック（つまり、今すぐ実行）; 注: 前処理の実際の値を使用したテストはサポートされていません。
Unsupported	いいえ	いいえ	いいえ

互換性のないデーモン起動された Zabbix プロセスのバージョンがログに警告メッセージとして出力されます。

サポートされる XML ファイル

バージョン 1.8 以降の XML ファイルは、Zabbix 7.4 へのインポートでサポートされています。

Attention:
XML エクスポート形式では、トリガーの依存関係は名前のみで保存されます。同じ名前のトリガーが複数ある場合（たとえば、深刻度や条件式が異なる場合）で、それらの間に依存関係が定義されていると、インポートすることはできません。このような依存関係は、XML ファイルから手動で削除し、インポート後に再度追加する必要があります。

12 Windows 向け Zabbix sender ダイナミックリンクライブラリ

概要

Windows 環境では、アプリケーションは外部プロセス (zabbix_sender.exe) を起動する代わりに、Zabbix sender ダイナミックリンクライブラリ (zabbix_sender.dll) を使用して、Zabbix サーバー/プロキシにデータを送信できます。

zabbix_sender.dll を使用するユーザーアプリケーションをコンパイルするには、zabbix_sender.h と zabbix_sender.lib が必要です。

入手方法

zabbix_sender.dll を入手する方法は 2 つあります。

1. zabbix_sender.h、zabbix_sender.lib、zabbix_sender.dll ファイルを ZIP アーカイブとしてダウンロードします。

ダウンロードオプションを選択する際は、*Encryption* で"No encryption"、*Packaging* で"Archive" を選択してください。その後、Zabbix agent 2 ではなく Zabbix agent をダウンロードします。

zabbix_sender.h、zabbix_sender.lib、zabbix_sender.dll ファイルは、ダウンロードした ZIP アーカイブ内の `bin\dev` ディレクトリにあります。必要な場所にファイルを展開してください。

2. ソースから zabbix_sender.dll をビルドします（手順を参照）。

開発用ファイルを含むダイナミックリンクライブラリは、`bin\winXX\dev` ディレクトリに配置されます。使用するには、zabbix_sender.h ヘッダーファイルをインクルードし、zabbix_sender.lib ライブラリとリンクしてください。

関連項目

- ライブラリの使用方法を示すために、Zabbix sender ダイナミックリンクライブラリで実装されたシンプルな Zabbix sender ユーティリティの例。
- Zabbix sender ダイナミックリンクライブラリのインターフェース関数について記載した zabbix_sender.h ファイル。このファイルには、各インターフェース関数の目的、引数、戻り値を説明するドキュメントが含まれています。

13 サービス監視のアップグレード

概要　Zabbix 6.0 ではサービスモニタリング機能が大幅に作り直されました。(変更についてはZabbix 6.0.0 の新機能を参照)

このページでは、Zabbix 6.0 以降へのアップグレード中に、以前の Zabbix バージョンで定義されたサービスと SLA がどのように変更されるかについて説明します。

サービス　古い Zabbix バージョンでは、サービスにはソフトとハードの 2 種類の依存関係がありました。アップグレード後、すべての依存関係は等しくなります。

サービス「子サービス」が以前にハード依存関係を介して「親サービス 1」にリンクされ、さらにソフト依存関係を介して「親サービス 2」にリンクされていた場合、アップグレード後「子サービス」は単に 2 つの親サービスにリンクされます（「親サービス 1」および「親サービス 2」）。

問題とサービス間のトリガーベースのマッピングは、タグベースのマッピングに置き換えられました。Zabbix 6.0 以降では、サービス設定フォームに新しいパラメータ *Problem tags* が追加され、問題のマッチングのために 1 つまたは複数のタグ名と値のペアを指定できるようになりました。サービスにリンクされているトリガーは、新しいタグ `ServiceLink:<トリガー ID>:<トリガー名>` を取得します。(タグ値は 32 文字に切り捨てられます）リンクされたサービスは、同じ値の `ServiceLink`Problem tags を取得します。

ステータス計算ルール

「ステータス計算アルゴリズム」は、以下のルールを使用してアップグレードされます。

- 計算しない → ステータスを OK に設定
- 障害（少なくとも 1 つの子サービスに障害がある場合）→ 子サービスの中で最も重大なもの
- 障害（すべての子サービスに障害がある場合）→ すべての子サービスに障害がある場合の最も重大なもの

SLA　以前は、SLA 目標を各サービスごとに個別に定義する必要がありました。Zabbix 6.0 以降、SLA は独立したエンティティとなり、サービススケジュール、期待されるサービスレベル目標（SLO)、および計算から除外するダウンタイム期間に関する情報を含むようになりました。設定後、SLA はサービスタグを通じて複数のサービスに割り当てることができます。

アップグレード中:

- 各サービスに対して定義された同一の SLA はグループ化され、各グループごとに 1 つの SLA が作成されます。
- 影響を受ける各サービスには特別なタグ `SLA:<ID>` が付与され、対応する SLA の *Service tags* パラメータにも同じタグが指定されます。
- SLA レポートの新しいメトリクスであるサービス作成時刻は、既存のサービスについては 01/01/2000 00:00 に設定されます。

ログインと systemd

ログインできないシステムユーザーとして *zabbix* ユーザーを作成することをお勧めします。一部のユーザーはこの推奨事項を無視し、同じアカウントを使用して Zabbix を実行しているホストにログイン（SSH を使用するなど）しますが、ログアウト時に Zabbix デーモンがクラッシュする可能性があります。この場合、Zabbix サーバーログに次のようなものが表示されます。

```
zabbix_server [27730]: [file:'selfmon.c',line:375] lock failed: [22] Invalid argument
zabbix_server [27716]: [file:'dbconfig.c',line:5266] lock failed: [22] Invalid argument
zabbix_server [27706]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Zabbix エージェントログ：

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

これは、`/etc/systemd/logind.conf` で構成されたデフォルトの systemd 設定 `RemoveIPC=yes` が原因で発生します。システムからログアウトすると、Zabbix によって以前に作成されたセマフォが削除され、クラッシュが発生します。

systemd ドキュメントからの引用：

```
RemoveIPC=

ユーザーが完全にログアウトしたときに、ユーザーに属するSystemVおよびPOSIXIPCオブジェクトを削除するかどうか
```

この問題には 2 つの解決策があります。

1. （推奨）Zabbix プロセス以外の目的で * zabbix * アカウントの使用を停止し、専用アカウントを作成します。
2. （非推奨）`/etc/systemd/logind.conf` で `RemoveIPC= no` を設定し、システムを再起動します。RemoveIPC はシステム全体のパラメータであり、変更するとシステム全体に影響することに注意してください。

プロキシの背後で Zabbix フロントエンドを使用する

Zabbix フロントエンドがプロキシサーバーの背後で実行されている場合、リバースプロキシパスと一致させるために、プロキシ設定ファイルの Cookie パスを書き換える必要があります。以下の例を参照してください。Cookie パスが書き換えられていない場合、ユーザーが Zabbix フロントエンドにログインしようとしたときに認証の問題が発生する可能性があります。

nginx の設定ファイル例

```
# ..
location / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Apache の設定ファイル例

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
# ..
```

15 エージェント **vs** エージェント **2** の比較

このセクションでは、Zabbix エージェントと Zabbix エージェント 2 の違いについて説明します。

Parameter	Zabbix agent	Zabbix agent 2
プログラミング言語	C	Go（一部は C）
デーモン化	はい	systemd のみ（Windows では可）
サポートされる拡張機能	C によるカスタムロード可能モジュール。	Go によるカスタムプラグイン。
要件		

Parameter	Zabbix agent	Zabbix agent 2
サポートされるプラットフォーム	Linux、IBM AIX、FreeBSD、NetBSD、OpenBSD、HP-UX、Mac OS X、Solaris: 9、10、11、Windows: XP 以降のすべてのデスクトップ版およびサーバー版	Linux、Windows: サポートされる Go バージョンをインストールできる、すべてのデスクトップ版およびサーバー版。
サポートされる暗号ライブラリ	GnuTLS 3.1.18 以降 OpenSSL 1.0.1、1.0.2、1.1.0、1.1.1、3.0.x LibreSSL - バージョン 2.7.4、2.8.2 でテスト済み (一定の制限があります。詳細は暗号化 ページを参照してください)。	Linux: OpenSSL 1.0.1 以降。 MS Windows: OpenSSL 1.1.1 以降。 OpenSSL ライブラリでは PSK サポートが有効になっている必要があります。LibreSSL はサポートされていません。
プロセス監視プロセス	server/proxy レコードごとに個別のアクティブチェックプロセス。	自動的に作成されるスレッドを持つ単一プロセス。スレッドの最大数は、GOMAXPROCS 環境変数によって決まります。
メトリクス	**UNIX**: サポートされるアイテムの一覧を参照してください。 **Windows**: Windows 固有の追加アイテムの一覧を参照してください。	**UNIX**: Zabbix エージェントでサポートされるすべてのメトリクス。 さらに、エージェント 2 は Docker、Memcached、MySQL、PostgreSQL、Redis、systemd、その他の監視対象向けに Zabbix ネイティブの監視ソリューションを提供します。完全な一覧は、エージェント 2 固有のアイテムを参照してください。 **Windows**: Zabbix エージェントでサポートされるすべてのメトリクスに加え、HTTPS、LDAP の net.tcp.service* チェックもサポートします。 さらに、エージェント 2 は PostgreSQL、Redis 向けに Zabbix ネイティブの監視ソリューションを提供します。
並行性	単一のサーバーに対するアクティブチェックは順次実行されます。	異なるプラグインからのチェック、または 1 つのプラグイン内の複数のチェックを並行して実行できます。
サードパーティトラップ	いいえ	はい
追加機能		
永続ストレージ	いいえ	はい
log*[] メトリクス用の永続ファイル	はい（Unix のみ）	いいえ
ログデータのアップロード	バッファを解放するため、ログ収集中に実行できます。	バッファがいっぱいになるとログ収集は停止するため、BufferSize パラメータは少なくとも MaxLinesPerSecond x 2 である必要があります。
実行時のユーザー変更	はい（Unix 系システムのみ）	いいえ（systemd により制御されます）
ユーザー設定可能な暗号スイート	はい	いいえ

関連情報:

- *Zabbix* プロセスの説明: Zabbix エージェント、Zabbix エージェント 2
- 設定パラメータ: Zabbix エージェント UNIX / Windows、Zabbix エージェント 2 UNIX / Windows

16 エスケープの例

概要

このページでは、さまざまなコンテキストで正規表現を使用する際に正しいエスケープを使用する例を示します。

Note:
トリガー式コンストラクターを使用すると、正規表現に正しいエスケープが自動的に追加されます。

例

コンテキスト付きユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキスト付きユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}`

注意:

- バックスラッシュはエスケープされません;
- 引用符はエスケープされます。

アイテムキーパラメータ内のマクロ関数

正規表現: `.+:(\d+)$<br>` アイテムキー: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

注意:

- regsub マクロ関数内の正規表現は二重引用符で囲まれています (閉じ括弧が含まれているため)。
- 正規表現を囲む引用符はエスケープされます（3 番目の項目パラメータ全体が二重引用符で囲まれているため)。
- 3 番目の項目キーパラメータはカンマを含むため二重引用符で囲まれています。

LLD マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

注意:

- バックスラッシュはエスケープされません。
- 引用符はエスケープされます。

ユーザーマクロコンテキスト内の **LLD** マクロ関数

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` コンテキスト付きユーザーマクロ: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

注意:

- LLD マクロのバックスラッシュエスケープは変更されません。
- LLD マクロをユーザーマクロコンテキストに挿入する際は、文字列に組み込む必要があります。
  1. マクロ式の前後に引用符が追加されます。
  2. 引用符がエスケープされます。合計で 3 つの新しいバックスラッシュが導入されます。

関数の文字列パラメータ **(**任意**)**

concat を例として使用します。

文字列の内容: `\.+\"[a-z]+<br>` 式: `concat("abc", "\\.\\\"[a-z]+")`

注意:

- 文字列パラメータでは、バックスラッシュと引用符の両方をエスケープする必要があります。

関数の文字列パラメータ内に **LLD** マクロ関数があります

正規表現: `\.+\"([a-z]+)<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` 式: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

注意:

- 文字列パラメータでは、バックスラッシュと引用符の両方をエスケープする必要があります。
- マクロは文字列が引用符で囲まれていない場合にのみ解決されます。

関数の文字列パラメータ内にコンテキストを持つユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキストを持つユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` 式: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

注意:

- 前の例と同様に、追加のエスケープ処理が必要です。
- バックスラッシュと引用符は、最上位レベルのエスケープ処理（文字列パラメータであるため）でのみエスケープされます。

関数内のユーザーマクロコンテキスト内の **LLD** マクロ関数

正規表現: `\.+\"([a-z])+<br>` LLD マクロ: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` コンテキスト付きユーザーマクロ: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` 式: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

エスケープの 3 つのレイヤーに注意してください。

1. LLD マクロ関数の場合（バックスラッシュをエスケープしません）
2. コンテキスト付きユーザーマクロの場合（バックスラッシュをエスケープしません）
3. 関数の文字列パラメータの場合（バックスラッシュをエスケープします）

文字列のすぐ内側にコンテキストがあるユーザーマクロ

正規表現: `\.+\"[a-z]+<br>` コンテキストがあるユーザーマクロ: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` 何らかの表現の文字列内。例: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

注意:

- 文字列ではバックスラッシュによるエスケープも必要です。
- 文字列では引用符によるエスケープも必要です。
- ここでも 2 段階のエスケープが必要です。
    1. ユーザーマクロのコンテキストをエスケープするため、バックスラッシュエスケープは不要です。
    2. 文字列であるため、バックスラッシュエスケープは必要です。

23. クイックリファレンスガイド

概要

このドキュメントセクションには、一般的に必要な監視目標に合わせて Zabbix をセットアップするための簡単なレシピが含まれています。

これは、新しい Zabbix ユーザーを念頭に置いて設計されており、タスクを解決するために必要な情報を含む他のドキュメントセクションをナビゲーターとして使用できます。

次のクイックリファレンスガイドが利用可能です。

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

1 Zabbix エージェントで **Linux** を監視する

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

このガイドの対象者

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

前提条件

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

Zabbix エージェントのインストール　Zabbix エージェントは、データ収集を担当するプロセスです。

Zabbix サーバーのバージョンを確認します。

```
zabbix_server -V
```

監視したい Linux マシンに、同じバージョンの Zabbix エージェントをインストールします（推奨）。
監視要件に応じて、Zabbix サーバーがインストールされている同じマシンにインストールすることも、まったく別のマシンにインストールすることもできます。

最適なインストール方法を選択してください。

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

監視用に **Zabbix** を設定する　Zabbix エージェントは、アクティブモードまたはパッシブモードでメトリクスを収集できます（同時使用も可能です）。

> **Note:**
> パッシブチェックは、単純なデータ要求です。Zabbix サーバーまたはプロキシが何らかのデータ（たとえば CPU 負荷）を要求し、Zabbix エージェントがその結果をサーバーに返します。アクティブチェックでは、より複雑な処理が必要です。エージェントはまず、独立して処理するアイテムの一覧をサーバーから取得し、その後データをまとめて送り返す必要があります。
> 詳細はPassive and active agent checksを参照してください。

Zabbix が提供する監視テンプレートには、通常 2 つの選択肢があります。1 つは Zabbix エージェント用のテンプレート、もう 1 つは Zabbix エージェント（アクティブ）用のテンプレートです。前者を選択した場合、エージェントはパッシブモードでメトリクスを収集します。
これらのテンプレートは、使用する通信プロトコルは異なりますが、同一の監視結果を提供します。

以降の Zabbix 設定は、active またはpassive のどちらの Zabbix エージェントチェック用テンプレートを選択するかによって異なります。

パッシブチェック　Zabbix エージェント

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

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（および必要に応じてポート）を server パラメータに追加します。例:

```
Server=192.0.2.0:10051
```

Zabbix エージェントはこのアドレスを使用して、指定された Zabbix サーバーまたはプロキシからの受信接続のみを受け付け、データ取得を許可します。

3. Zabbix エージェントを再起動します。

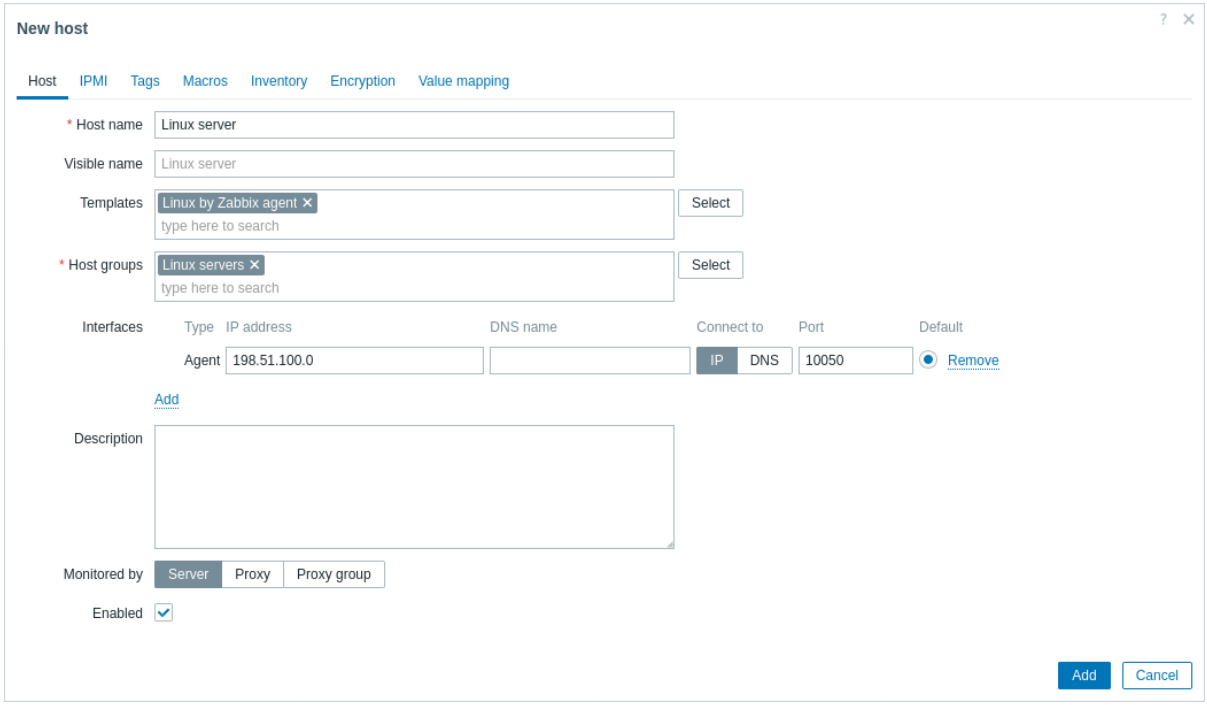
```
systemctl restart zabbix-agent
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

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



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

アクティブチェック　Zabbix エージェント

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

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（および必要に応じてポート）を ServerActive パラメータに追加します。例:

```
ServerActive=192.0.2.0:10051
```

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

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

```
Hostname=Linux server
```

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

4. Zabbix エージェントを再起動します。

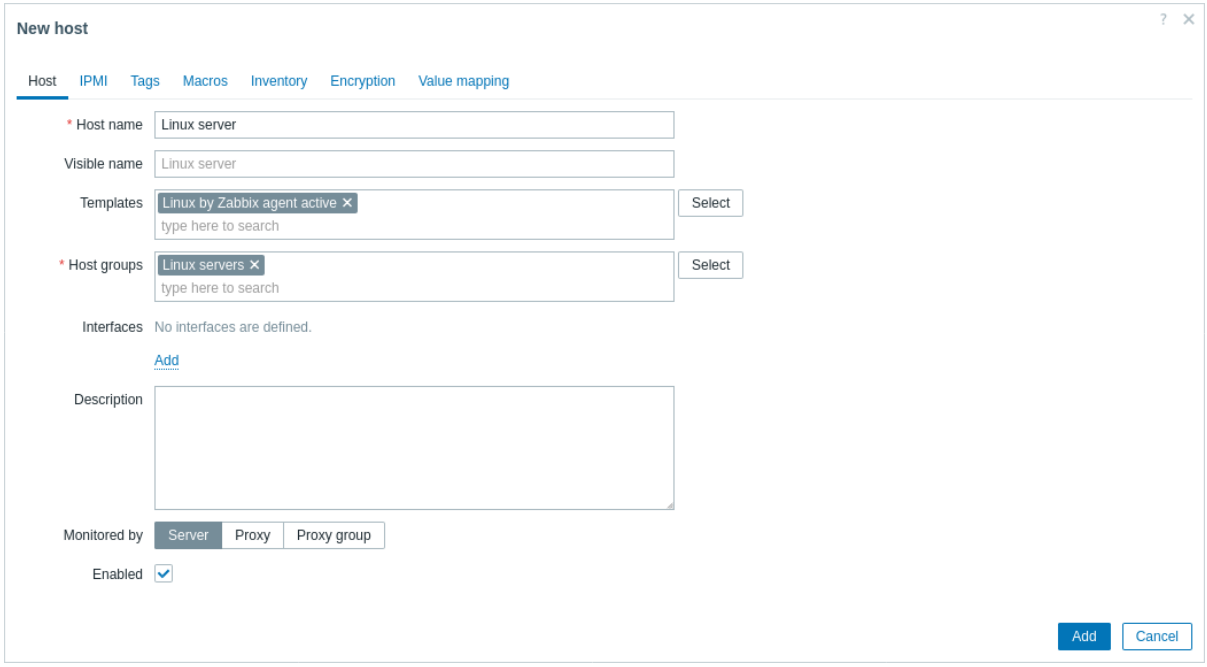
```
systemctl restart zabbix-agent
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

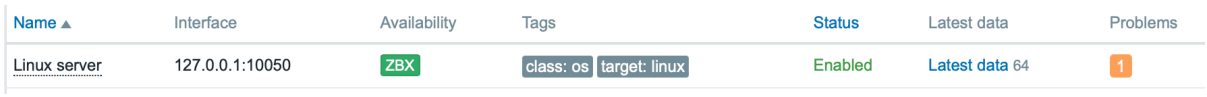
- *Host name* フィールドに、先ほどエージェント設定ファイルで定義した *Hostname* パラメータの値と一致するホスト名（例: "Linux server"）を入力します。
- *Templates* フィールドで、ホストにリンクするテンプレート"Linux by Zabbix agent active" を入力または選択します。
- *Host groups* フィールドで、ホストグループ（例: "Linux servers"）を入力または選択します。
- *Interfaces* は未設定のままにします。アクティブチェックでは、エージェントはサーバーからの接続を待ち受けるのではなく、エージェント側からサーバーへの接続を開始するため、インターフェースは不要です。



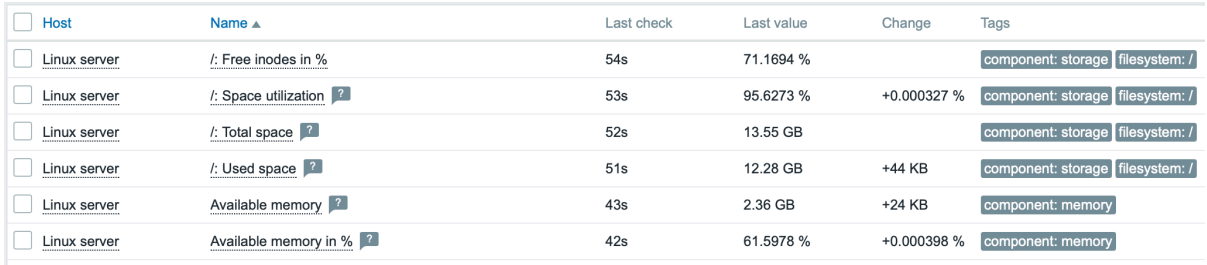
3. *Add* をクリックしてホストを追加します。このホストは、監視対象の Linux マシンを表します。

収集したメトリクスの表示　おめでとうございます。ここまでで、Zabbix はすでに Linux マシンの監視を開始しています。

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

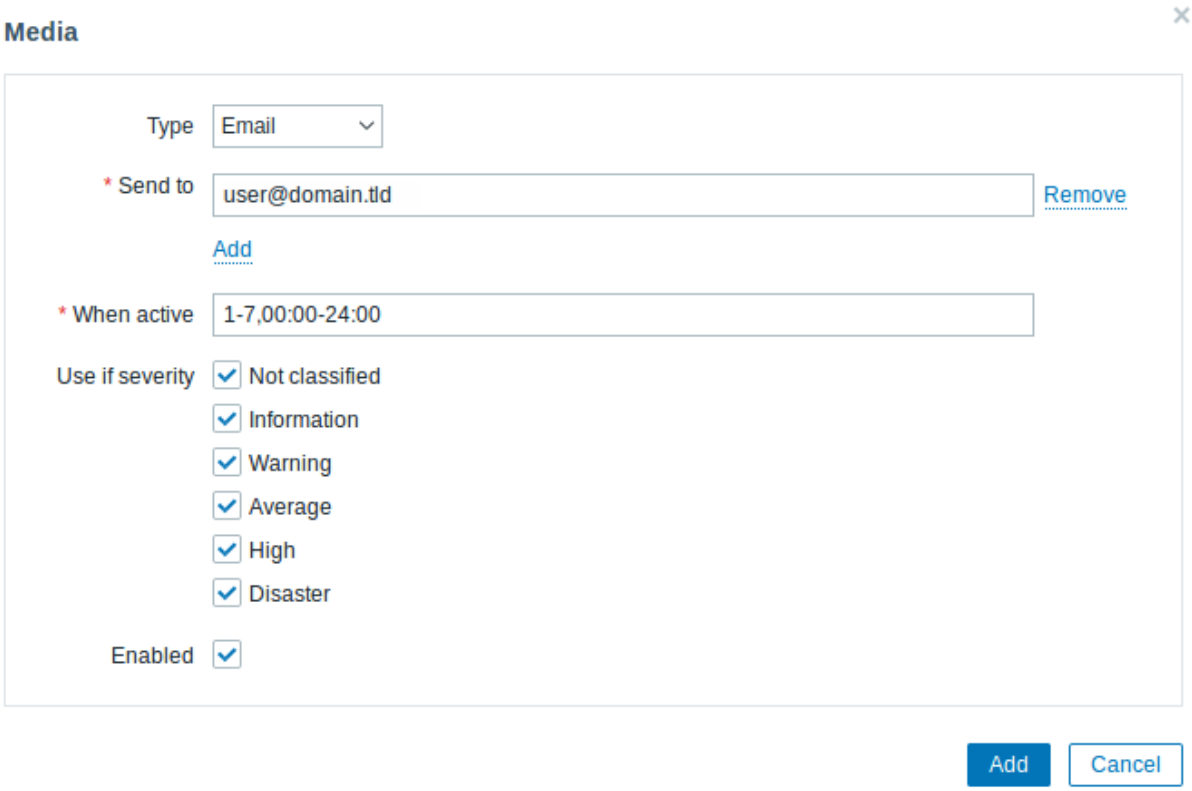


この操作により、Linux サーバーホストから収集された最新のメトリクスの一覧が表示されます。



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

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



2. 問題通知の受信 のガイドに従ってください。

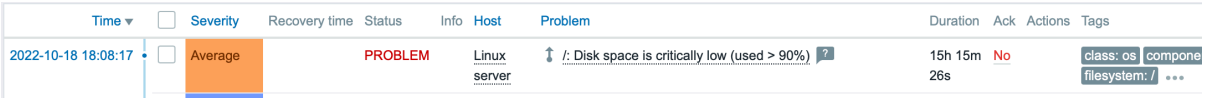
次回、Zabbix が問題を検出したときに、メールでアラートを受信するはずです。

設定をテストする　Linux では、次を実行することで高い CPU 負荷をシミュレートし、その結果として障害アラートを受信できます。

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

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

Zabbix が問題を検出すると、Monitoring->Problems セクションに表示されます。



アラートが設定されている場合は、問題通知も受信します。

関連項目:

- アイテムの作成 -
  追加のメトリクスの監視を開始する方法（テンプレートを使用しないカスタム監視）。
- Zabbix エージェントアイテム, Windows 向け Zabbix エージェントアイテム - Windows で Zabbix エージェントを使用して監視できるメトリクスの完全な一覧。
- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法（例: まずシステム管理者にメッセージを送信し、その後、障害が 45 分以内に解決されない場合は、データセンターマネージャーにメッセージを送信します）。
- パッケージからのインストール - 各種 Linux ディストリビューション向けの公式 RPM および DEB パッケージを使用して Zabbix コンポーネントをインストールする方法。これにより、最新機能およびバグ修正を利用できます。

2 Zabbix エージェントで Windows を監視する

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

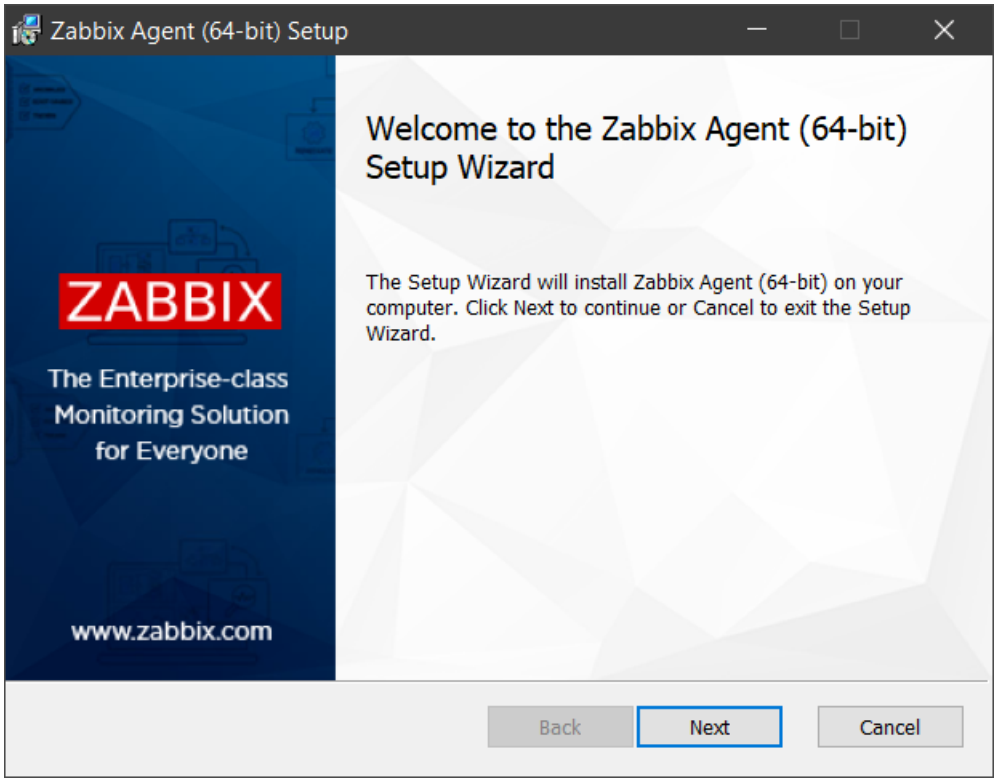
このガイドの対象者

このガイドは、Zabbix を初めて利用するユーザー向けに作成されており、Windows マシンの基本的な監視を有効にするために必要な最小限の手順を記載しています。
詳細なカスタマイズオプションを探している場合、またはより高度な設定が必要な場合は、Zabbix マニュアルの設定 セクションを参照してください。

前提条件

このインストールガイドを進める前に、お使いの OS 向けの手順に従って、Zabbix サーバーおよび Zabbix Web インターフェースをダウンロードしてインストール しておく必要があります。

Zabbix エージェントのインストール　Zabbix エージェントは、データの収集を担当するプロセスです。監視したい Windows マシンにインストールする必要があります。Windows向けの Zabbix エージェントのインストール手順に従ってください。



監視用に **Zabbix** を設定する　Zabbix エージェントは、アクティブモードまたはパッシブモードでメトリクスを収集できます（同時使用も可能です）。

Note:
パッシブチェックは、単純なデータ要求です。Zabbix サーバーまたはプロキシが何らかのデータ（たとえば CPU 負荷）を要求し、Zabbix エージェントがその結果をサーバーに返します。アクティブチェックでは、より複雑な処理が必要です。エージェントはまず、独立して処理するためのアイテムの一覧をサーバーから取得し、その後データをまとめて送り返す必要があります。詳細は、Passive and active agent checksを参照してください。

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

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

パッシブチェック　Zabbix frontend

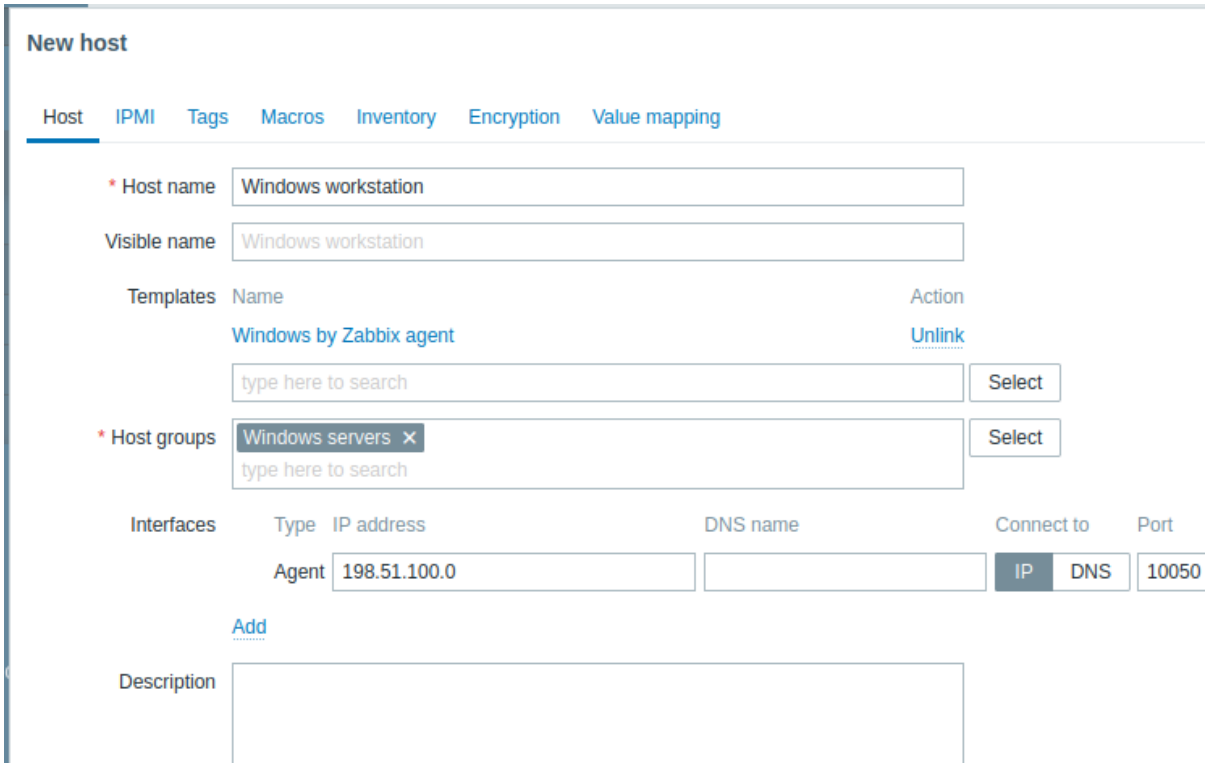
1. Log into Zabbix frontend.

2. Create a host in Zabbix web interface.

This host will represent your Windows machine.

3. In the *Interfaces* parameter, add *Agent* interface and specify the IP address or DNS name of the Windows machine where the agent is installed.

4. In the *Templates* parameter, type or select *Windows by Zabbix agent*.



Zabbix agent

For passive checks Zabbix agent needs to know the IP address or DNS name of Zabbix server. If you have provided correct information during the agent installation, the configuration file is already updated. Otherwise, you need to manually specify it. Go to the `C:\Program files\Zabbix Agent` folder, open the file *zabbix_agentd.conf* and add the IP/DNS of your Zabbix server to the *Server* parameter.

Example:

```
Server=192.0.2.22
```

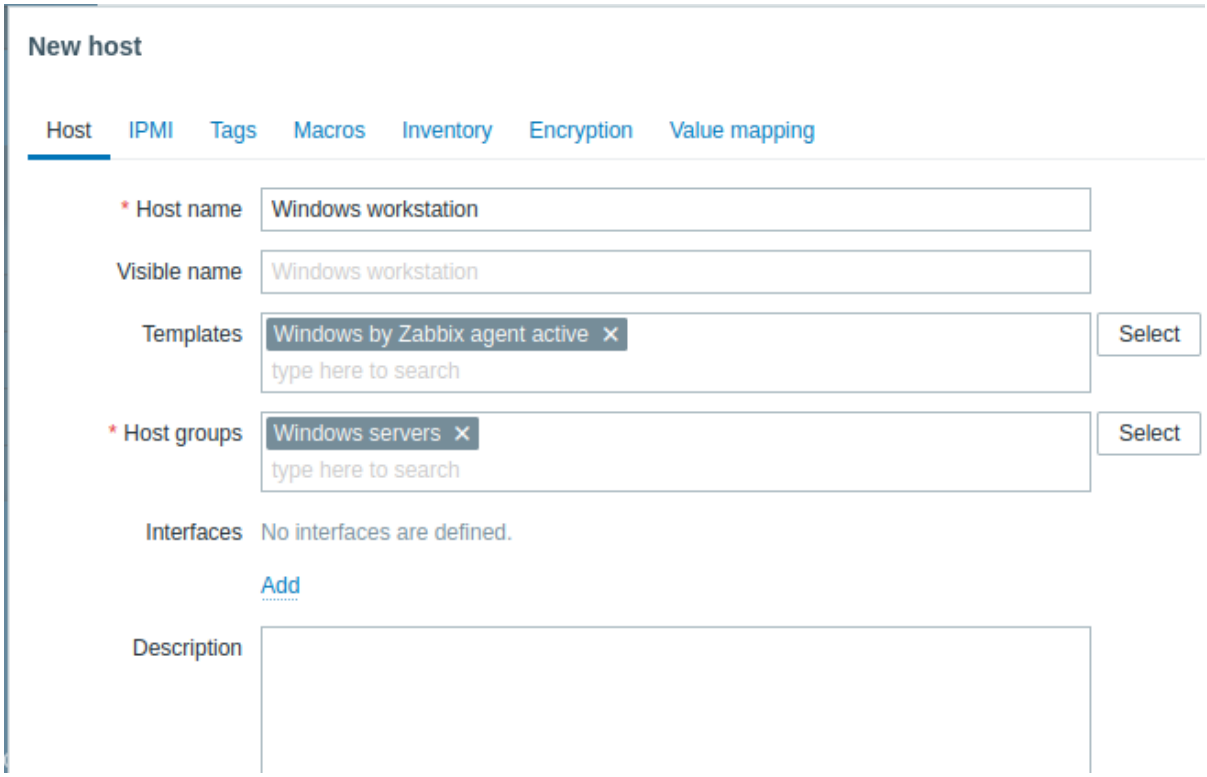
アクティブチェック　Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

このホストは Windows マシンを表します。

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



Zabbix agent

In the `C:\Program files\Zabbix Agent` folder open the file *zabbix_agentd.conf* and add:

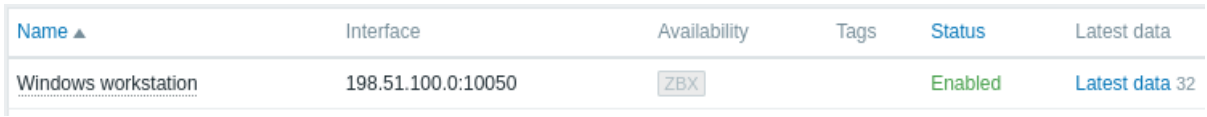
- The name of the host you created in Zabbix web interface to the *Hostname* parameter.
- The IP address or DNS name of your Zabbix server machine to the *ServerActive* parameter (might be prefilled if you have provided it during Zabbix agent setup).

Example:

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

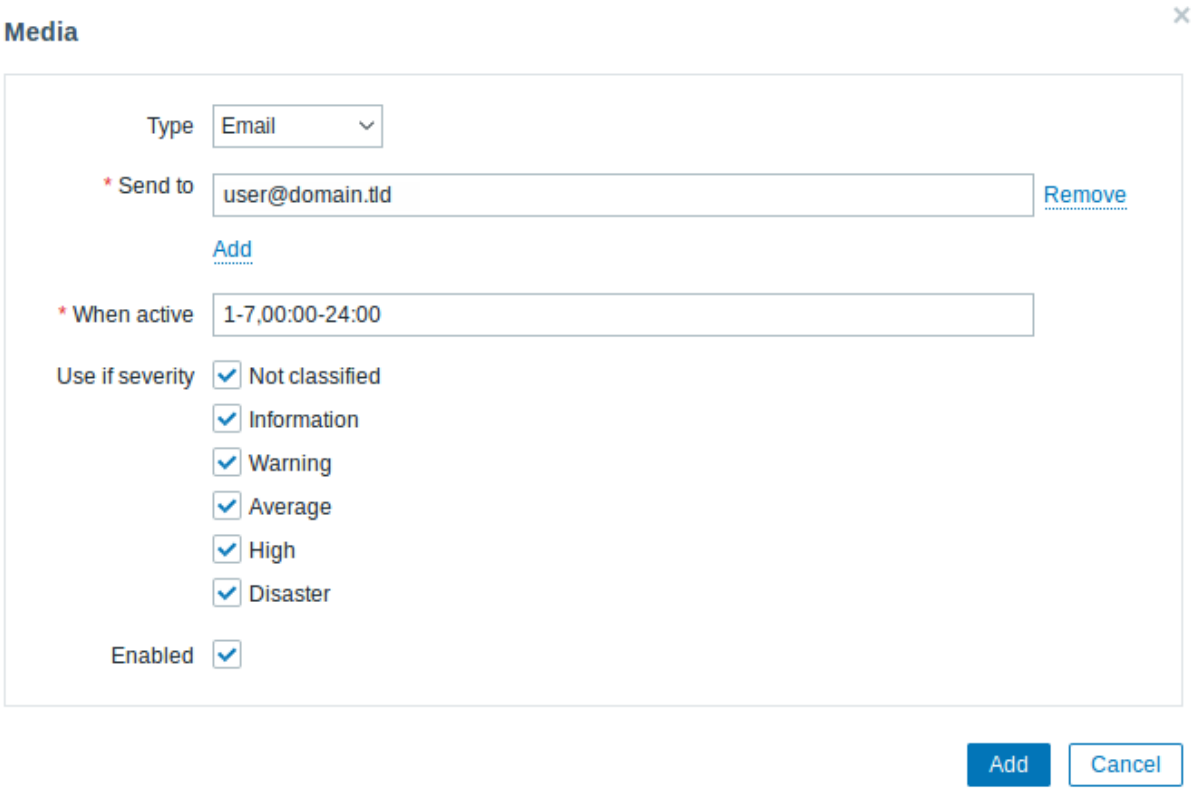
収集したメトリクスの表示　おめでとうございます。この時点で、Zabbix はすでに Windows マシンの監視を行っています。

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



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

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



2. 障害通知の受信 のガイドに従ってください。

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

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

関連情報:

- アイテムの作成 - 追加のメトリクスの監視を開始する方法（テンプレートを使用しないカスタム監視）。
- Zabbix エージェントのアイテム、Windows 向け Zabbix エージェントのアイテム - Windows で Zabbix エージェントを使用して監視できるメトリクスの完全な一覧。
- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法（例: 最初にシステム管理者にメッセージを送信し、その後、45 分以内に障害が解決されない場合はデータセンターマネージャーにメッセージを送信する）。

3 HTTP 経由で **Apache** を監視する

はじめに　このページでは、追加のソフトウェアをインストールせずに Apache Web サーバーの監視を開始する簡単な方法を説明します。

このガイドの対象者

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

前提条件

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

Apache の準備　1. 使用している Apache のバージョンを確認します。

RHEL ベースのシステムでは、次を実行します。

```
httpd -v
```

Debian/Ubuntu では、次を実行します。

```
apache2 -v
```

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

RHEL ベースのシステムでは、次を実行します。

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

Debian/Ubuntu では、次を実行します。

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

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

RHEL ベースのシステムでは、次を実行します。

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

Debian/Ubuntu では、次を実行します。

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

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

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

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

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

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

ファイルに次の行を追加します（**198.51.100.255** を使用している Zabbix サーバーの IP アドレスに置き換えてください）。

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

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

4. Apache を再起動します。

RHEL ベースのシステムでは、次を実行します。

```
sudo systemctl restart httpd
```

Debian/Ubuntu では、次を実行します。

```
sudo systemctl restart apache2
```

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

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

レスポンスには、Apache Web サーバーの統計情報が含まれているはずです。

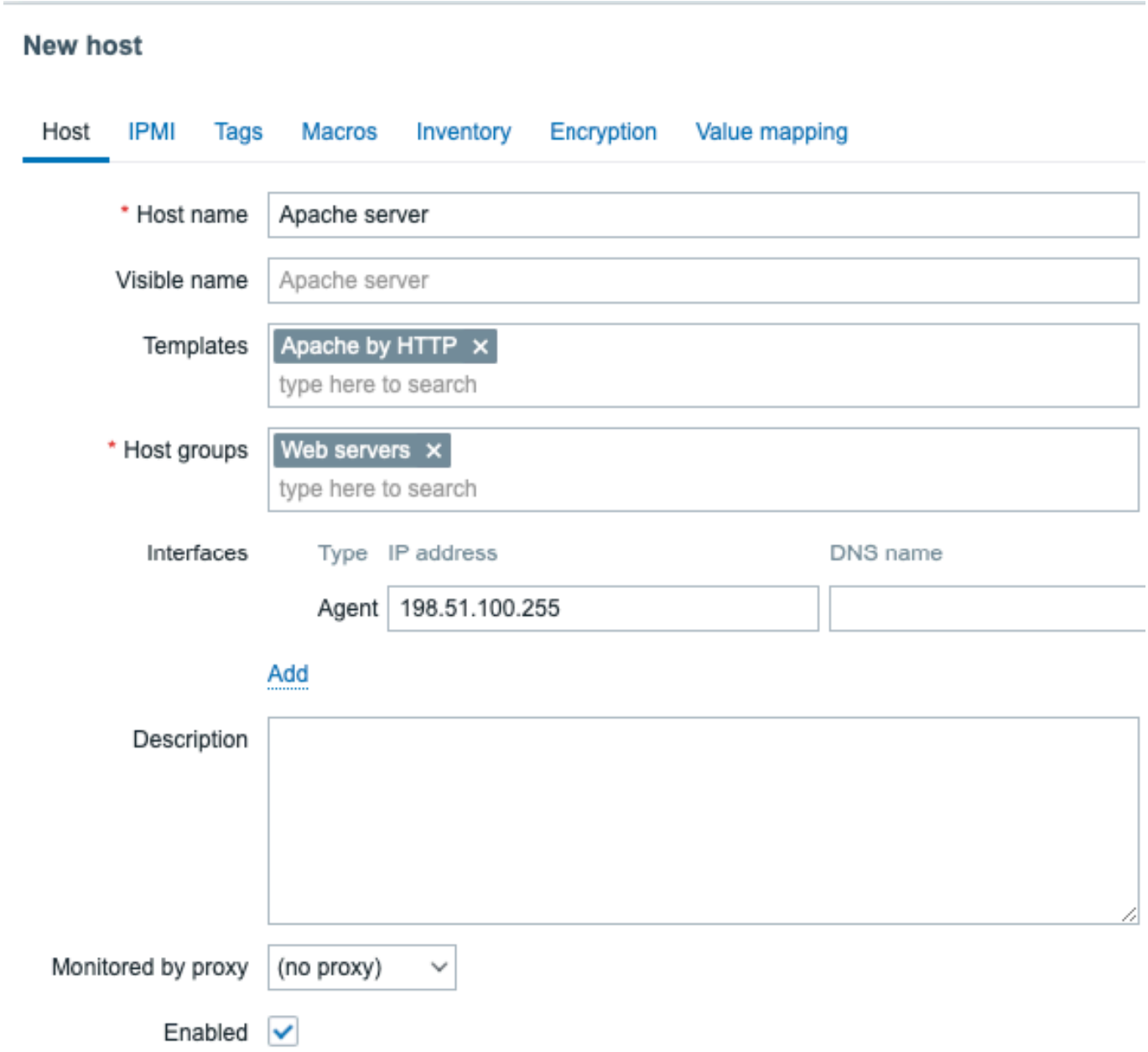
監視のための **Zabbix** の設定　1. Zabbix の Web インターフェースにログインします。

2. Zabbix の Web インターフェースでホストを作成します。

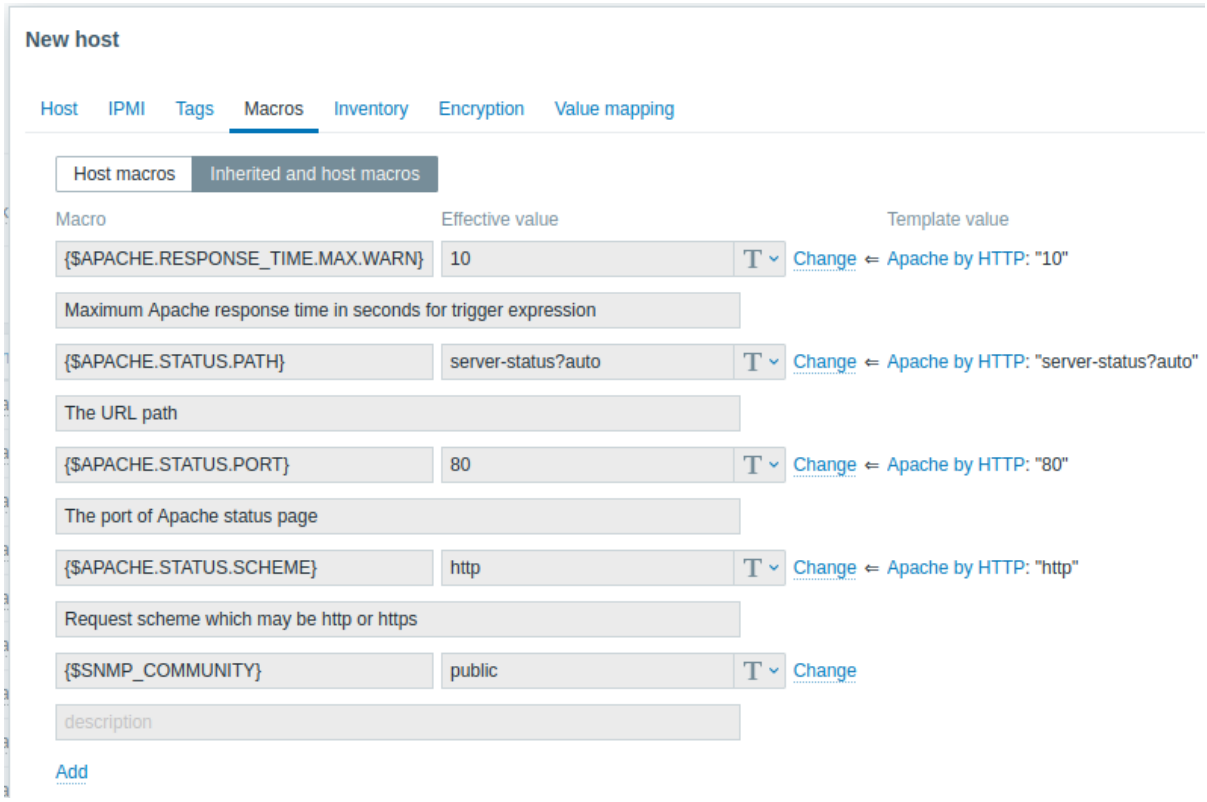
このホストは Apache サーバーを表します。

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

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



5. **Macros** タブに切り替え、*Inherited and host macros* モードを選択します。マクロ {$APACHE.STATUS.PORT} および {$APACHE.STATUS.SCHEME} の値がインストール設定に適していることを確認します。デフォルトでは、ポートは 80、スキームは http です。異なるポートやスキームを使用している場合は、マクロの値を変更してください。

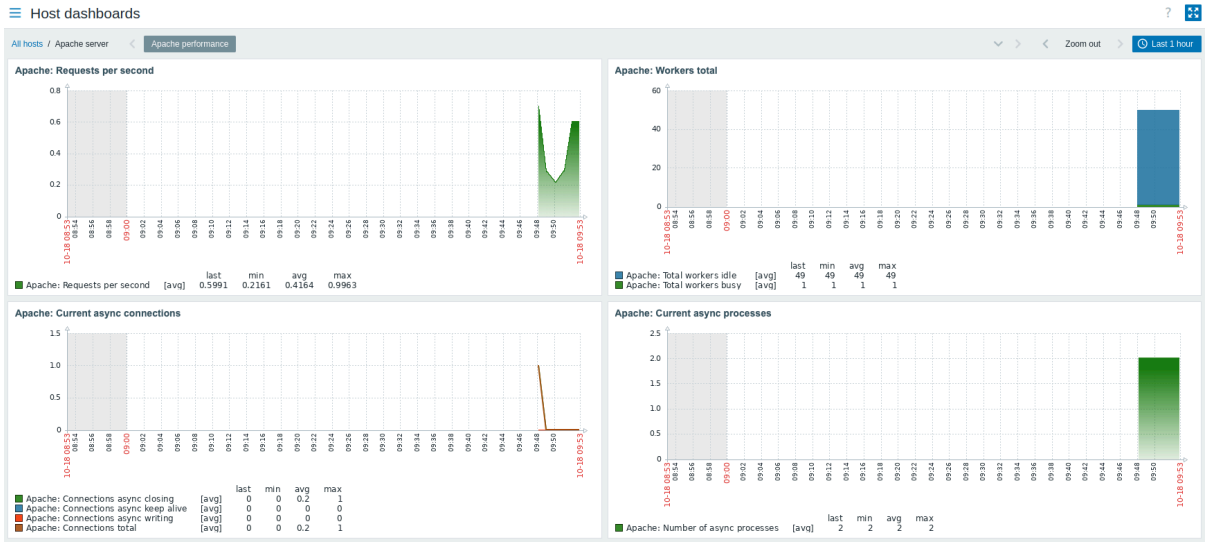


収集したメトリクスを表示する　おめでとうございます。この時点で、Zabbix はすでに Apache Web サーバーの監視を開始しています。

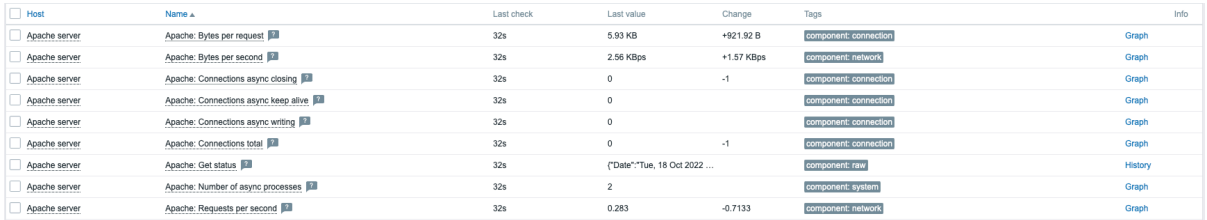
収集したメトリクスを表示するには、監視-> ホストのメニューセクション を開き、ホストの横にある ダッシュボードをクリックします。



この操作により、Apache の /server-status ページから収集された最も重要なメトリクスが表示されるホストダッシュボードに移動します。

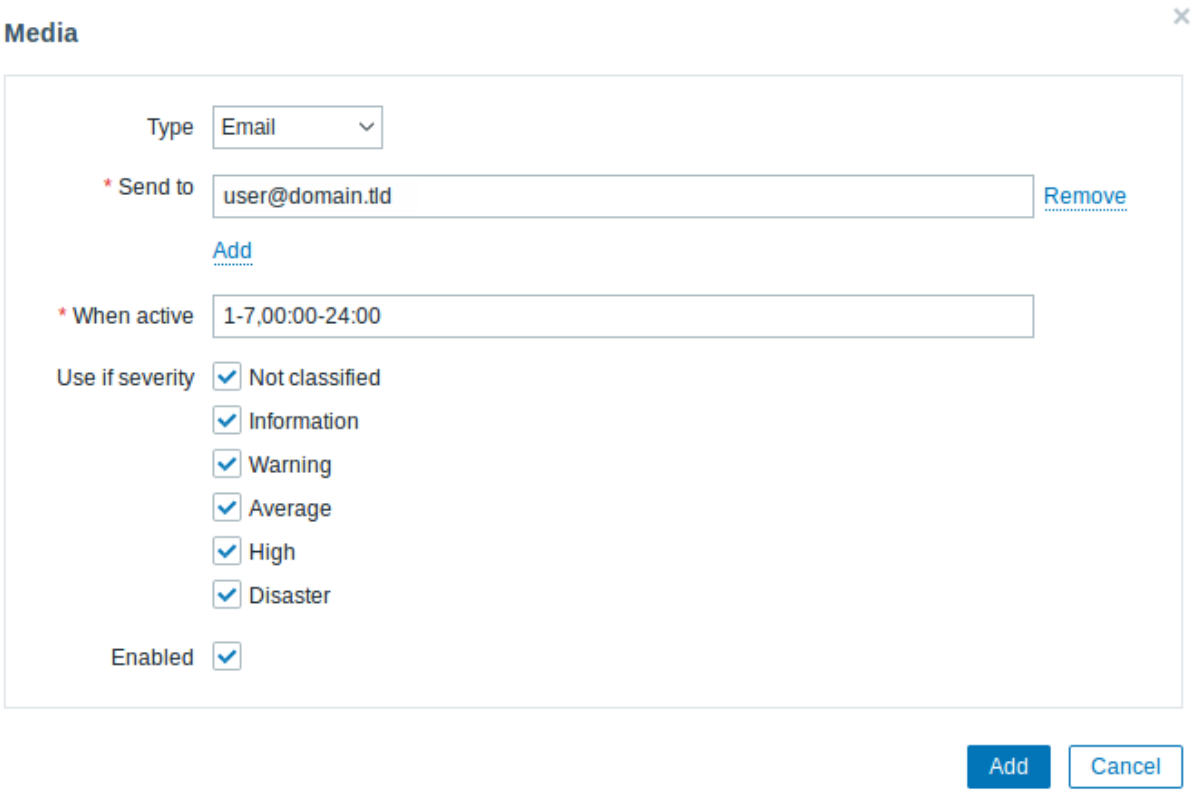


または、監視-> ホストから 最新データをクリックすると、収集された最新のメトリクスを一覧で表示できます。



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

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



2. 問題通知の受信 のガイドに従ってください。

次回、Zabbix が問題を検出すると、メールでアラートを受信するはずです。

設定をテストする　実際の問題をシミュレートしてテスト用の障害アラートを受信するには、次の手順を実行します。

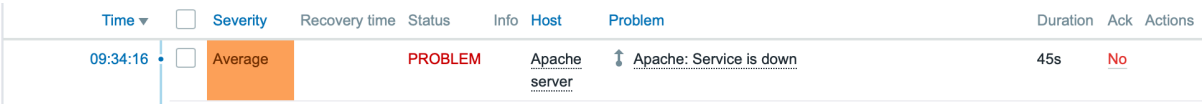
1. Zabbix で *Apache server* ホストの設定を開きます。

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

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

4. 「Update」を押してホスト設定を保存します。

5. 数分以内に、Zabbix はインスタンスに接続できなくなるため、*Apache service is down* という問題を検出します。
これは「Monitoring->Problems」セクションに表示されます。



アラートが設定されている場合は、問題通知も受信します。

6. 問題を解決して Apache の監視を継続するには、マクロの値を元に戻します。

関連項目:

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

4 Zabbix エージェント **2** で **MySQL** を監視する

はじめに

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

MySQL サーバーを監視するには、いくつかの方法があります: Zabbix エージェント、Zabbix エージェント 2、または Open Database Connectivity（ODBC）標準です。このガイドでは、主に Zabbix エージェント 2 を使用して MySQL サーバーを監視する方法に焦点を当てています。これは、さまざまな構成においてシームレスに設定できるため、推奨される方法です。ただし、このページではその他の方法についての手順も案内していますので、要件に最も適したものを自由に選択してください。

このガイドの対象者

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

前提条件

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

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

MySQL ユーザーの作成

MySQL サーバーを監視するには、Zabbix がそのサーバーおよびそのプロセスにアクセスできる必要があります。
MySQL のインストール環境には、必要なアクセスレベルを持つユーザー（Zabbix のインストール時に作成された「zabbix」ユーザー）がすでに存在しますが、
このユーザーには、単純な監視に必要な範囲を超える権限（データベースの DROP、テーブルからのエントリ削除など）が付与されています。
そのため、MySQL サーバーを監視専用とする MySQL ユーザーを作成する必要があります。

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

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

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

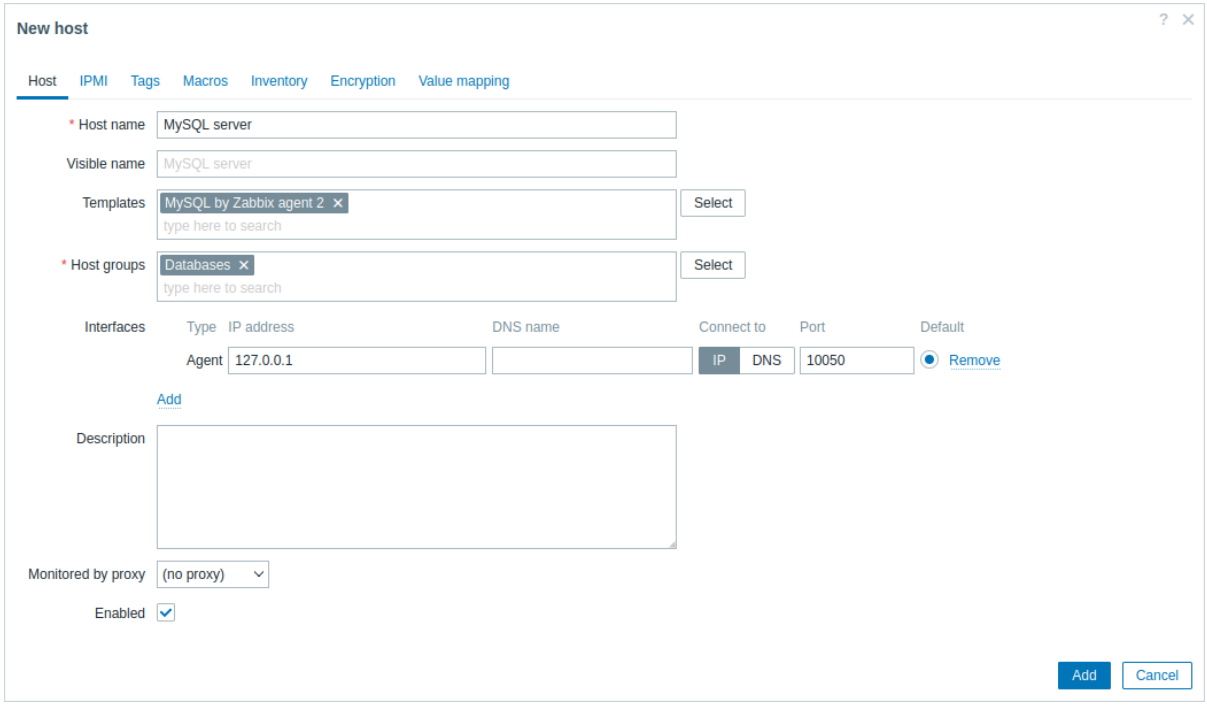
ユーザーの作成が完了したら、次の手順に進むことができます。

Zabbix Web インターフェースの設定

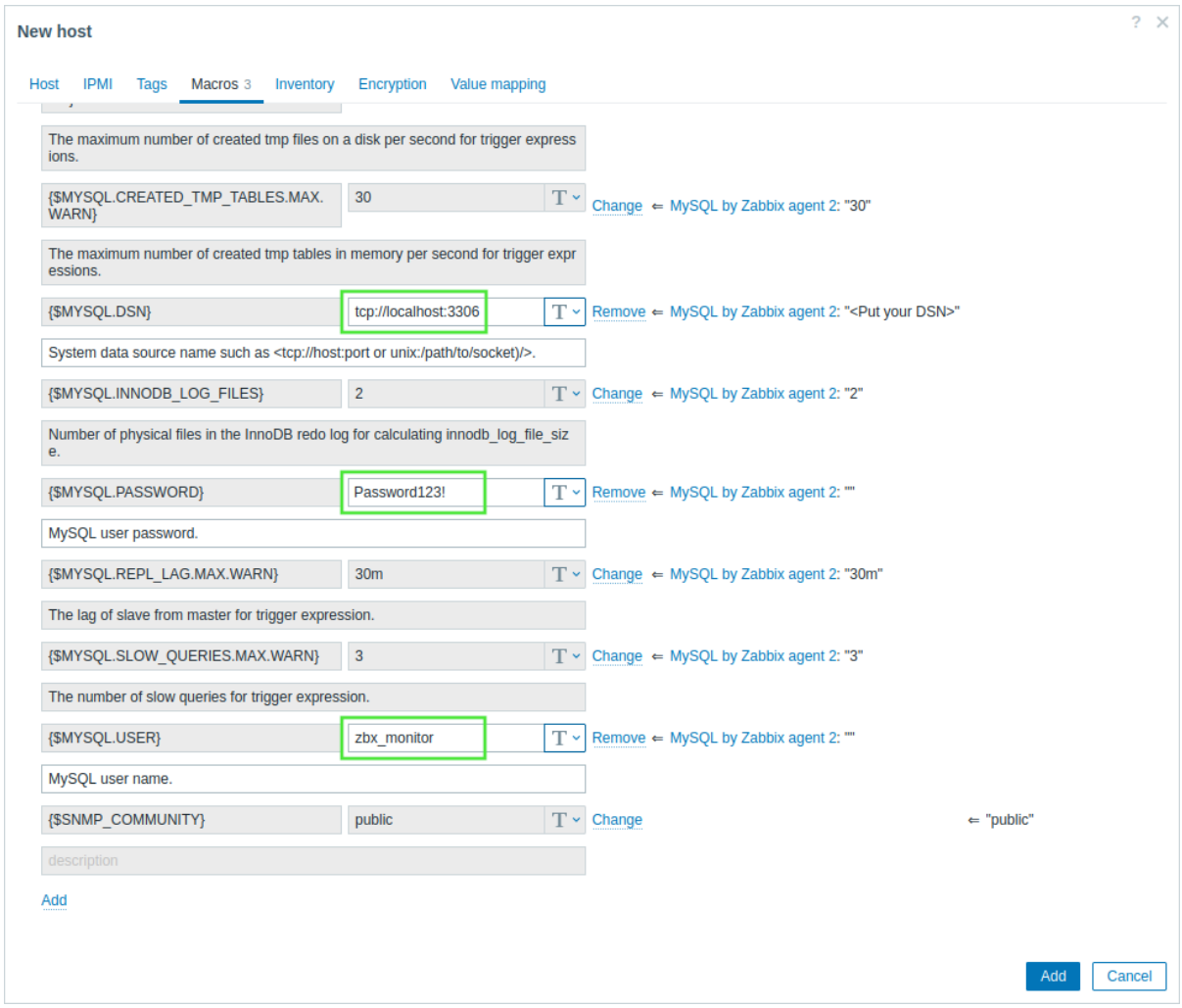
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

- *Host name* フィールドに、ホスト名（例: ”MySQL server”）を入力します。
- *Templates* フィールドで、ホストにリンクするテンプレート”MySQL by Zabbix agent 2” を入力または選択します。
- *Host groups* フィールドで、ホストグループ（例: ”Databases”）を入力または選択します。
- *Interfaces* フィールドで、タイプ”Agent” のインターフェースを追加し、MySQL サーバーの IP アドレスを指定します。
  このガイドでは、Zabbix サーバーおよび Zabbix agent 2 と同じマシンにインストールされた MySQL サーバーを監視するために、”127.0.0.1”（localhost）を使用します。



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

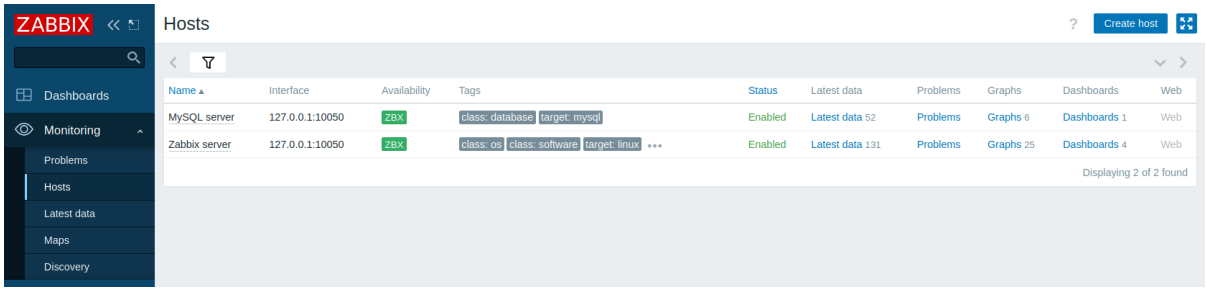


3. *Add* をクリックしてホストを追加します。このホストは MySQL サーバーを表します。

収集したメトリクスの表示

おめでとうございます。この時点で、Zabbix はすでに MySQL サーバーの監視を開始しています。

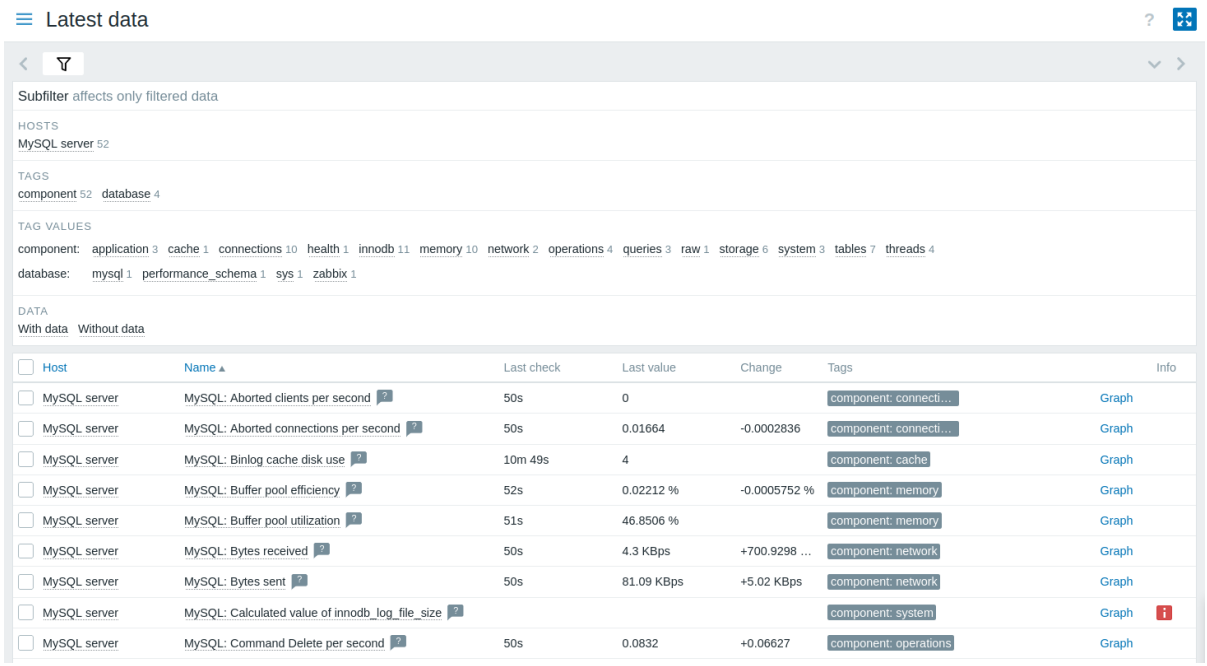
収集したメトリクスを表示するには、監視 → ホスト メニューセクションに移動し、ホストの横にある ダッシュボードをクリックします。



この操作により、MySQL サーバーから収集された最も重要なメトリクスが表示されるホストダッシュボード（テンプレートレベルで設定）に移動します。



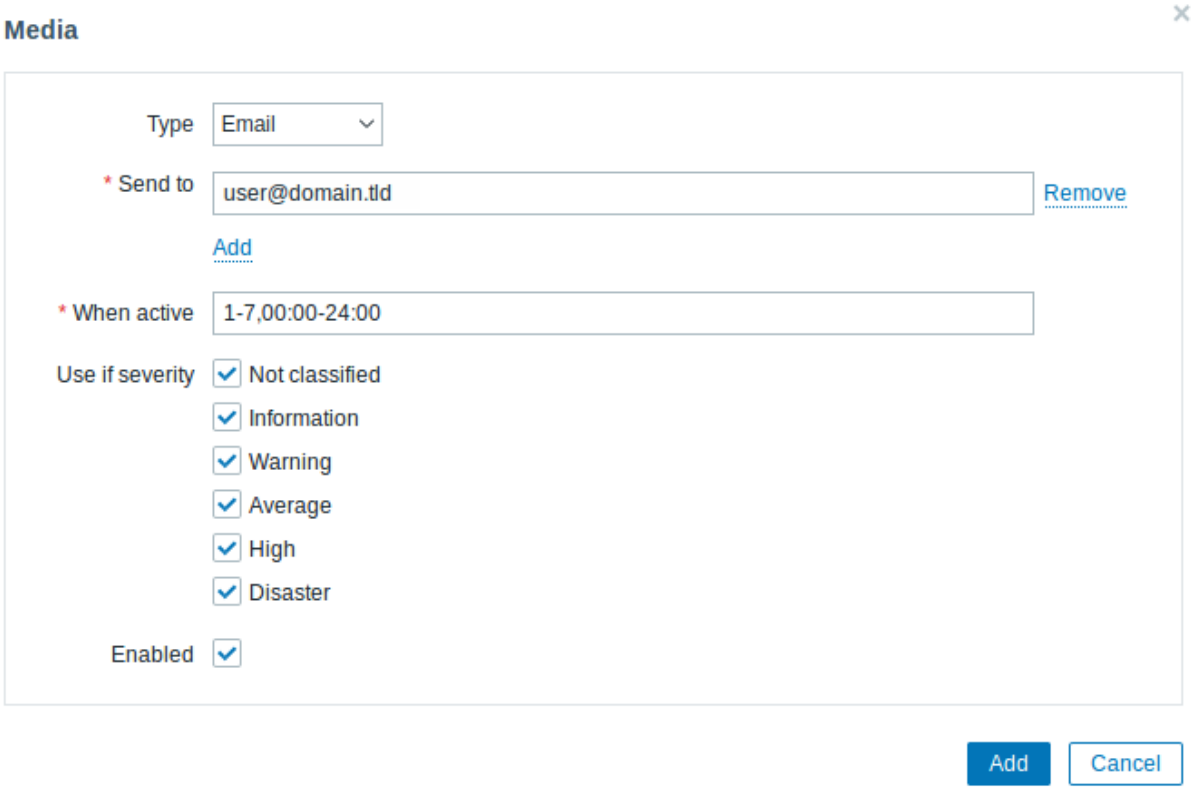
または、監視 → ホスト メニューセクションから、最新データ をクリックして、最新の収集メトリクスを一覧で表示することもできます。
なお、アイテム *MySQL: Calculated value of innodb_log_file_size* にはデータがないことが想定されます。これは、この値が直近 1 時間のデータから計算されるためです。



問題アラートの設定

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

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



2. 問題通知を受信するのガイドに従います。

次回 Zabbix が問題を検出したときに、メールでアラートを受信できるようになります。

設定をテストする

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

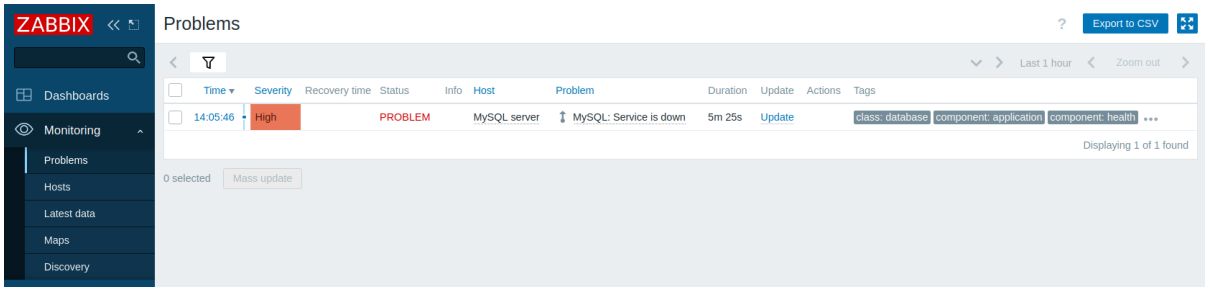
1. Zabbix で MySQL サーバーのホスト設定を開きます。

2. マクロタブに切り替え、継承されたホストマクロとホストマクロを選択します。

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

4. 更新をクリックしてホスト設定を更新します。

5. しばらくすると、Zabbix は MySQL サーバーに接続できなくなるため、「MySQL: Service is down」という問題を検出します。この問題は監視 → 障害に表示されます。



アラートが設定されている場合は、問題通知も受信します。

6. 問題を解決して MySQL サーバーの監視を継続するには、マクロ値を元の値に戻します。

MySQL を監視するその他の方法

Zabbix agent 2 で MySQL サーバーを監視する代わりに、Zabbix エージェントまたは Open Database Connectivity（ODBC）標準を使用することもできます。Zabbix agent 2 の使用が推奨されますが、環境によっては Zabbix agent 2 をサポートしていない場合や、カスタムの方法が必要になる場合があります。

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

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

Zabbix エージェントで MySQL を監視する

Zabbix エージェントで MySQL サーバーを監視するには、お使いの OS 向けの手順に従って、Zabbix サーバー、Zabbix Web インターフェース、および Zabbix エージェントをダウンロードしてインストールする必要があります。

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

MySQL ユーザーを作成した後は、MySQL サーバーへの接続を確立して監視できるように、Zabbix エージェントを設定する必要があります。
これには、カスタムエージェントチェックを実行するための複数のuser parametersの設定と、前の手順で作成した「zbx_monitor」ユーザーとして MySQL サーバーに接続するために必要な認証情報を Zabbix エージェントに提供することが含まれます。

Zabbix エージェントの設定

1. Zabbix エージェントの追加設定ディレクトリに移動します。

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

> **Attention:**
> Zabbix エージェントの追加設定ディレクトリは、Zabbix エージェント設定ファイル（*zabbix_agentd.conf*）と同じディレクトリに配置する必要があります。OS および Zabbix のインストール方法によっては、このディレクトリの場所がこのガイドで示されているものと異なる場合があります。デフォルトの場所については、Zabbix エージェント設定ファイル内の`Include`パラメータを確認してください。

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

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

```
vi template_db_mysql.conf
```

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

4. 設定を更新するために、Zabbix エージェントを再起動します。

```
systemctl restart zabbix-agent
```

Zabbix エージェントのユーザーパラメータを設定したら、次に Zabbix エージェントが MySQL サーバーにアクセスするための認証情報を設定できます。

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

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

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

```
vi .my.cnf
```

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

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

Zabbix Web インターフェースを設定し、設定をテストする

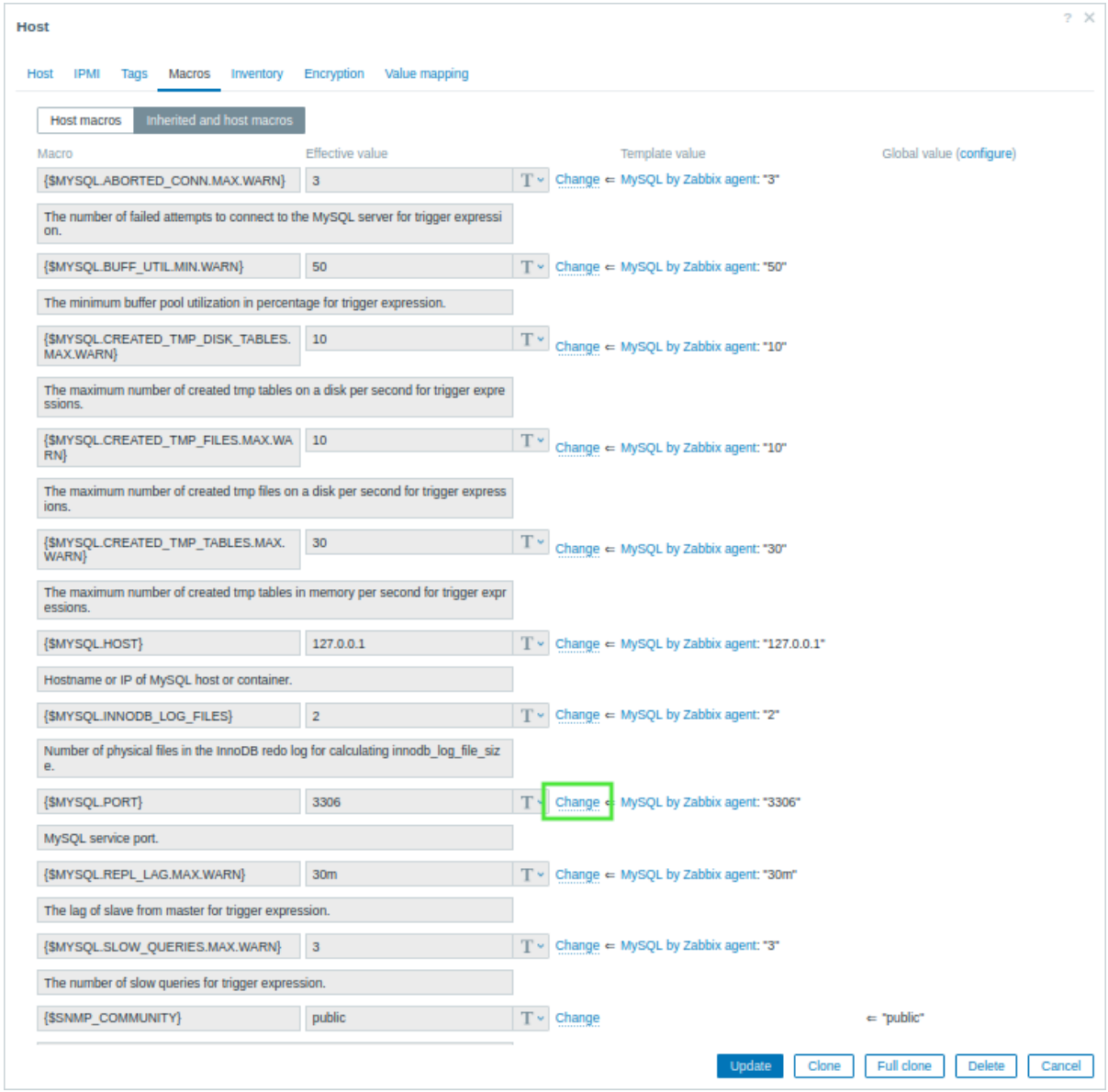
Zabbix Web インターフェースを設定するには、*Zabbix Web インターフェースを設定する* セクションの手順に従い、以下の調整を行ってください。

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

Zabbix Web インターフェースの設定が完了したら、収集されたメトリクスを表示し、障害アラートを設定できます。

設定をテストするには、設定をテストする セクションの手順に従い、以下の調整を行ってください。

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



ODBC で MySQL を監視する

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

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

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

ODBC を設定する

1. unixODBC をインストールします。unixODBC のインストールには、Linux オペレーティングシステムのデフォルトのパッケージリポジトリを使用する方法が推奨されます。

```
apt install unixodbc
```

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

```
apt install odbc-mariadb
```

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

```
odbcinst -j
```

このコマンドの実行結果は、以下のようになります。

```
unixODBC 2.3.9
```

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

4. MySQL データベースを監視するために ODBC ドライバーを設定するには、*odbcinst.ini* ファイルに記載されているドライバー名が必要です。以下の *odbcinst.ini* ファイルの例では、ドライバー名は”MariaDB Unicode” です。

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

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

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

Zabbix Web インターフェースを設定し、設定をテストする

Zabbix Web インターフェースを設定するには、*Zabbix Web インターフェースの設定* セクションの手順に従い、以下の調整を行ってください。

- *Templates* フィールドで、ホストにリンクされるテンプレート「MySQL by ODBC」を入力または選択します。
- *Interfaces* の設定は不要です。
- MySQL サーバーホスト設定の *Inherited and host macros* セクションにある {$MYSQL.DSN} マクロの値は、*odbc.ini* ファイルの DSN 名に設定する必要があります。

Zabbix Web インターフェースの設定が完了したら、収集されたメトリクスを表示したり、障害アラートを設定したり、設定をテストしたりできます。

関連項目

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法（例: 最初にシステム管理者にメッセージを送信し、その後、障害が 45 分以内に解決されない場合は、データセンターマネージャーにメッセージを送信します）。
- ODBC 監視 - 他の Linux ディストリビューションで ODBC を設定する方法、および ODBC を使用して追加のデータベース関連メトリクスの監視を開始する方法。
- テンプレート *MySQL by Zabbix agent* - *MySQL by Zabbix agent* テンプレートに関する追加情報。
- テンプレート *MySQL by Zabbix agent 2* - *MySQL by Zabbix agent 2* テンプレートに関する追加情報。
- テンプレート *MySQL by ODBC* - *MySQL by ODBC* テンプレートに関する追加情報。

5 Zabbix で VMware を監視する

はじめに

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

このガイドの対象者

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

前提条件

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

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

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

Zabbix サーバーの設定

VMware を監視するには、*vmware collector* Zabbix プロセスを有効にする必要があります。
VMware 監視がどのように実行されるかの詳細については、仮想マシン監視を参照してください。

1. Zabbix サーバーの設定ファイルを開きます。

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

2. Zabbix サーバーの設定ファイルで`StartVMwareCollectors` パラメータを見つけて、2 以上 に設定します（デフォルト値は 0 です）。

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

StartVMwareCollectors=2
```

3. Zabbix サーバーを再起動します。

```
systemctl restart zabbix-server
```

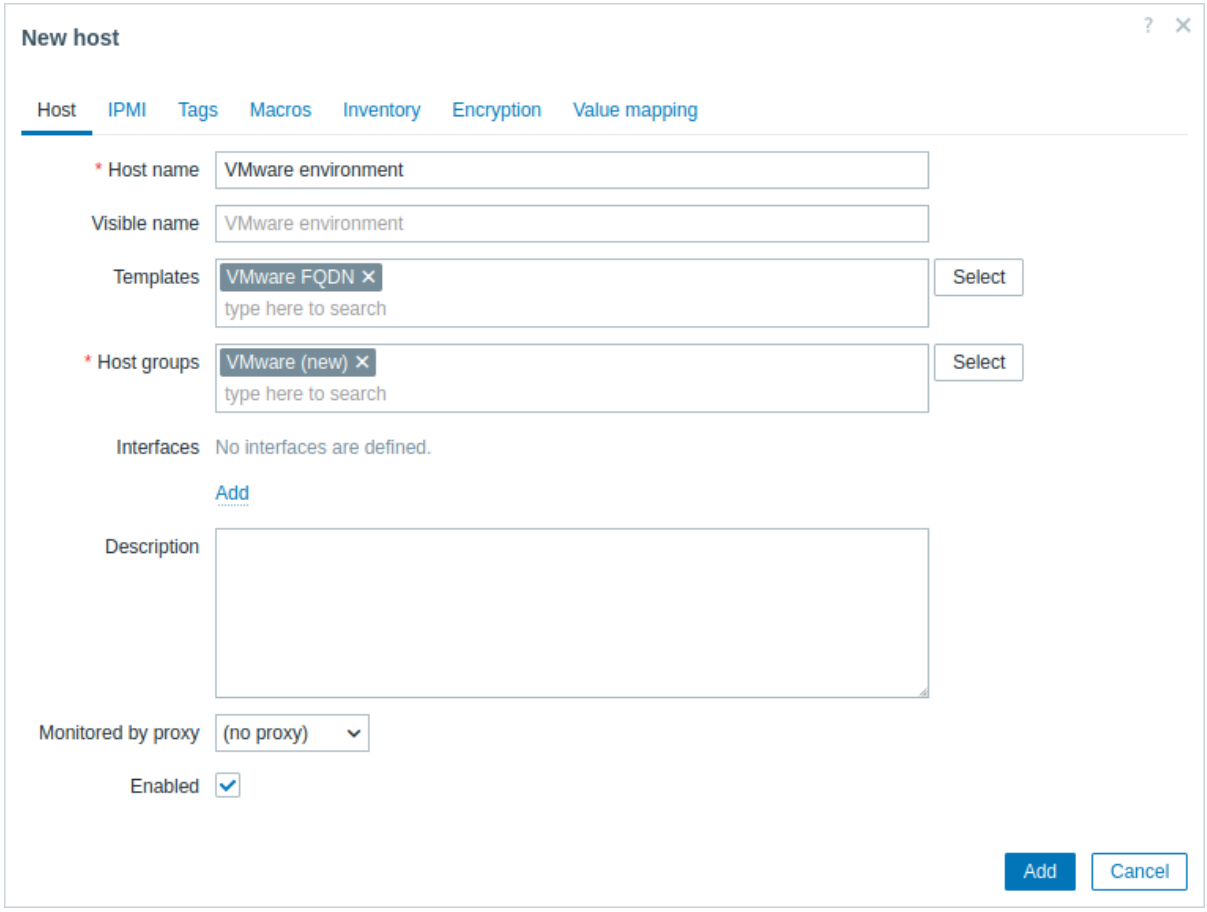
vmware collector プロセスが起動したら、次の手順に進みます。

Zabbix Web インターフェースの設定

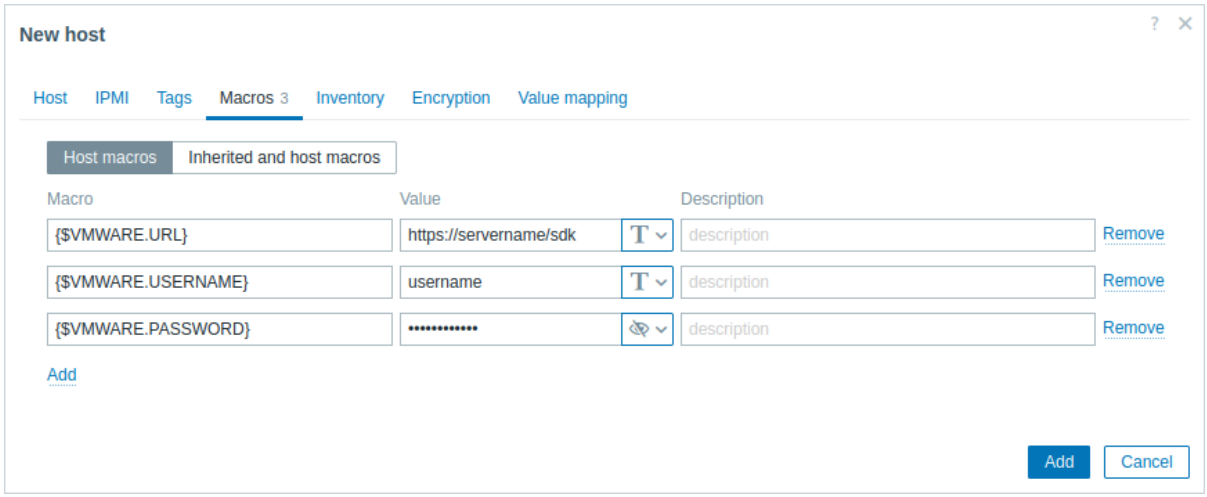
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

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



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



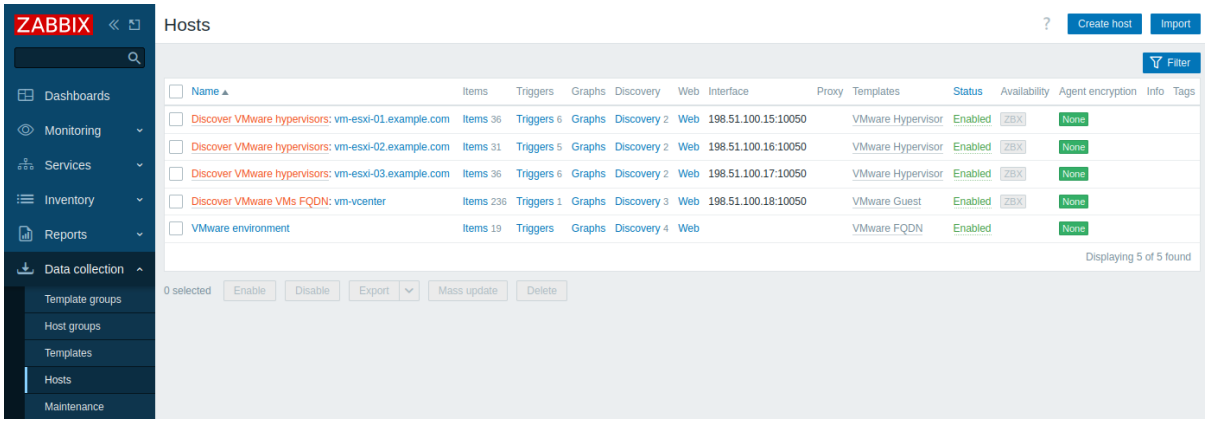
3. *Add* ボタンをクリックしてホストを作成します。このホストは VMware 環境を表します。

収集したメトリクスの表示

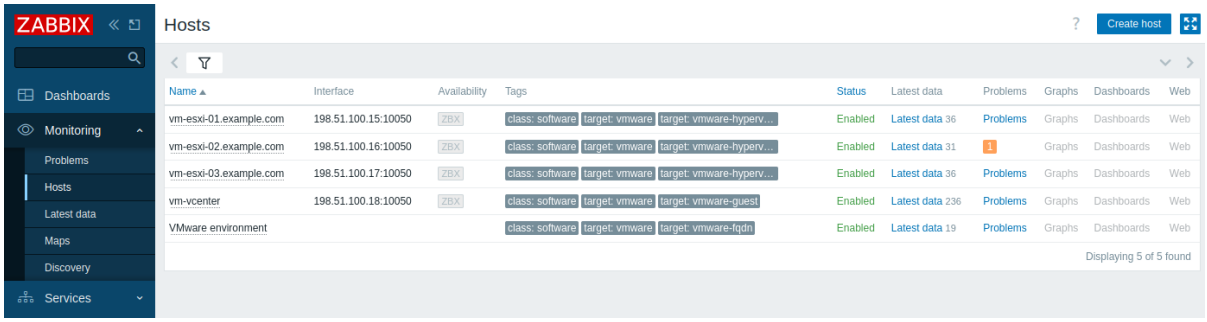
おめでとうございます。ここまでで、Zabbix はすでに VMware 環境の監視を開始しています。

VMware 環境の設定によっては、Zabbix がディスカバリを実行し、検出されたエンティティ用のホストを作成する場合があります。必要に応じて、ディスカバリとホストの作成は手動で実行することもできます。

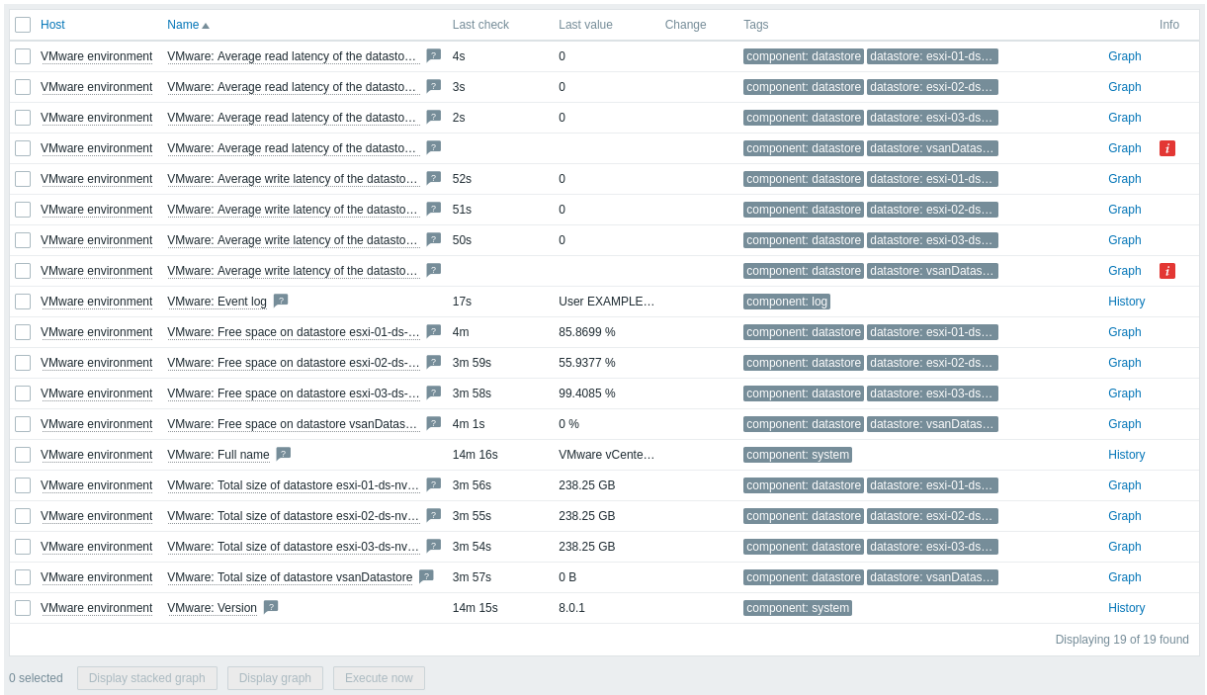
作成されたホストを表示するには、データ収集 → ホストメニューセクションに移動します。



収集したメトリクスを表示するには、監視 → ホストメニューセクションに移動し、作成された「VMware environment」ホスト、または検出されたエンティティ用に作成されたホストのいずれかの横にある *Latest data* をクリックします。



この操作により、選択したホストから収集された最新のメトリクスの一覧が表示されます。

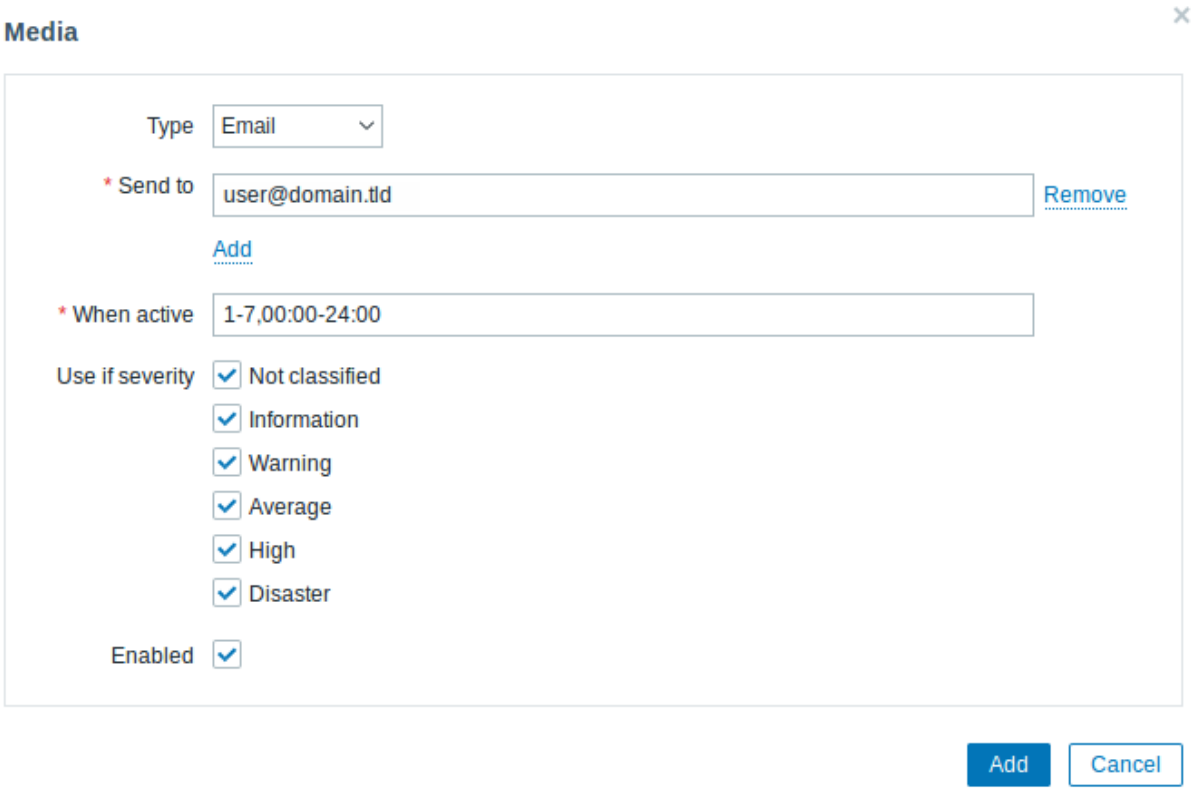


一部のアイテムにはデータがなく、*Not supported* 状態になっていることがあります。これは、監視対象の VMware 環境で有効になっていないため、Zabbix が特定のデータストア上で有効なパフォーマンスカウンターを見つけられないことが原因です。

問題アラートの設定

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

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



2. 問題通知を受信する のガイドに従います。

次回 Zabbix が問題を検出したときに、メールでアラートを受信できるようになります。

関連項目

- アイテムの作成 - 追加のメトリクスの監視を開始する方法。
- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法（例: まずシステム管理者にメッセージを送信し、その後、障害が 45 分以内に解決されない場合は、データセンターマネージャーにメッセージを送信します）。
- 仮想マシンの監視 - VMware 監視に関する追加情報（データ収集プロセス、サーバー設定オプション、トラブルシューティングガイドなど）。
- VMware 監視アイテムキー - Zabbix を使用して監視できる VMware メトリクスの完全な一覧。
- テンプレート *VMware* - *VMware* テンプレートに関する追加情報。
- テンプレート *VMware FQDN* - *VMware FQDN* テンプレートに関する追加情報。

6 Zabbix でネットワークトラフィックを監視する

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

このガイドの対象者

このガイドは、Zabbix を初めて利用するユーザー向けに作成されており、ネットワークトラフィックの基本的な監視を有効にするために必要な最小限の手順を説明しています。より詳細なカスタマイズオプションを探している場合や、より高度な設定が必要な場合は、Zabbix マニュアルの設定セクションを参照してください。

前提条件

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

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

監視用に **Zabbix** を設定する　Zabbix エージェントは、アクティブモードまたはパッシブモードで（同時に）メトリクスを収集できます。詳細はパッシブおよびアクティブなエージェントチェックを参照してください。このガイドでは、パッシブチェックによる監

視について説明します。

Zabbix エージェントの設定

1. エージェントがインストールされているマシンで、エージェントの設定ファイルを開きます（デフォルトでは、パスは */usr/local/etc/zabbix_agentd.conf* です）:

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

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

```
Server=192.0.2.22
```

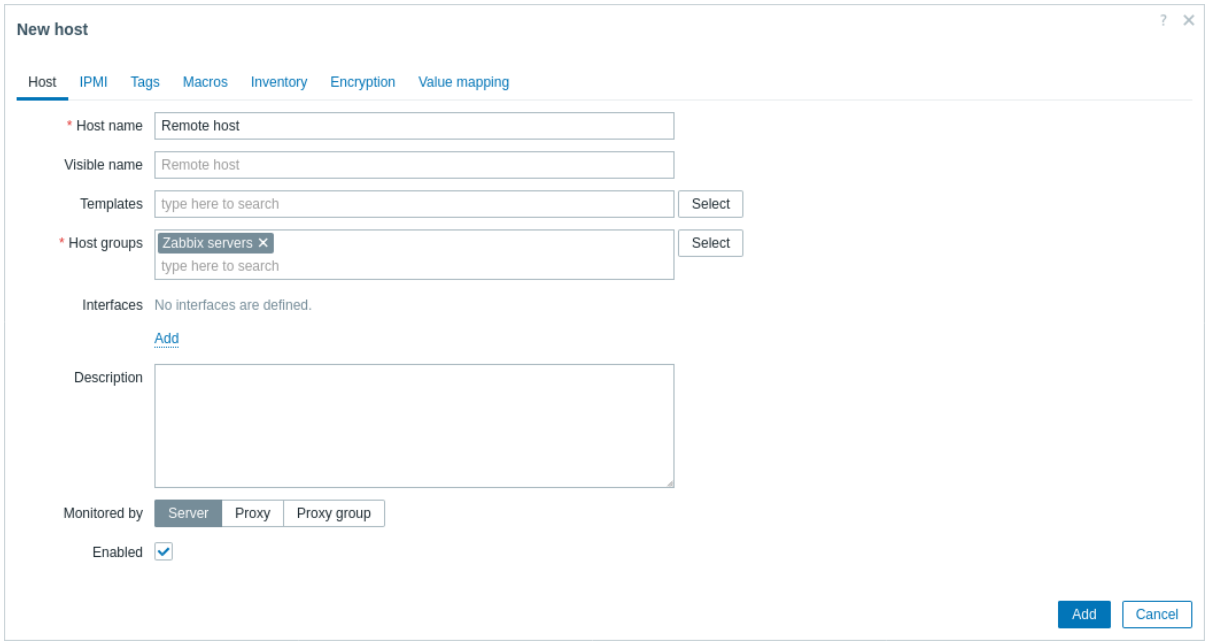
3. Zabbix エージェントを再起動します:

```
systemctl restart zabbix-agent
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

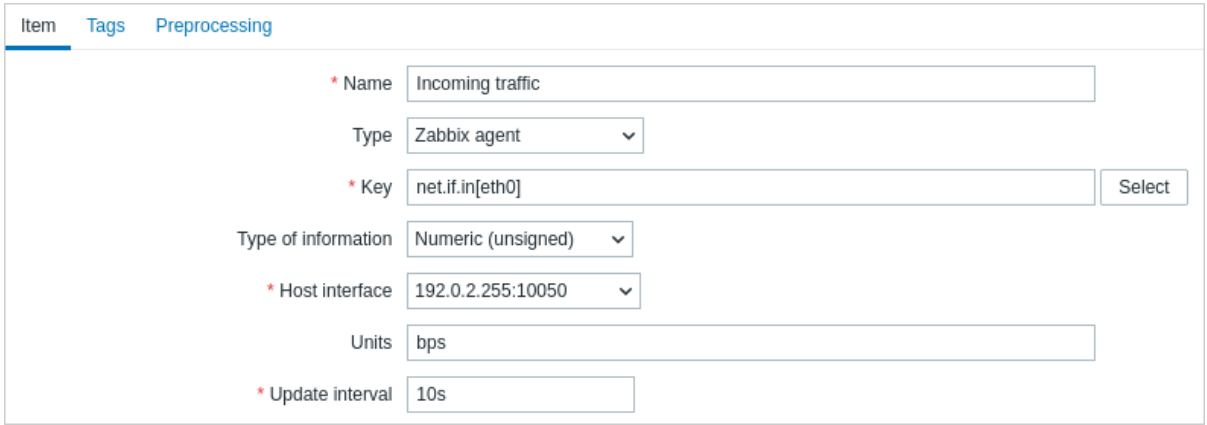
2. Zabbix Web インターフェースでホストを作成し、エージェントがインストールされているマシンの IP アドレスまたは DNS 名を指定します。



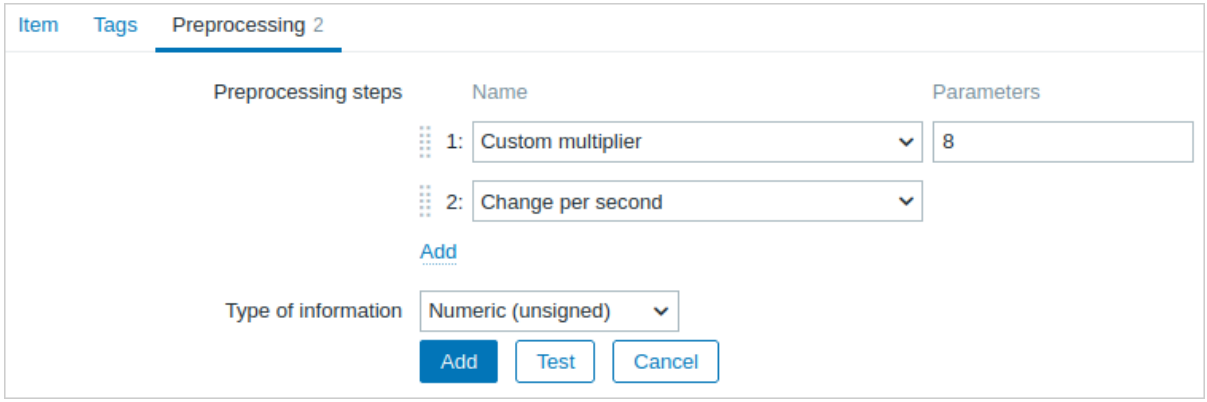
アイテムの作成　トラフィック監視用のアイテムを追加するには、アイテムの作成 の手順に従ってください。追加するアイテムは次のとおりです。

- 受信トラフィック
- 送信トラフィック
- 総トラフィック

受信トラフィック監視用アイテムのシンプルな設定例は、次のようになります。

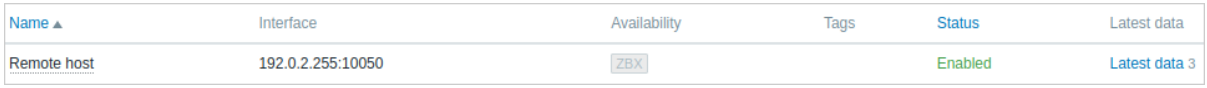


収集したデータを実用的に利用できるようにするため、アイテムの作成時にいくつかの前処理手順を設定できます。この例では、バイトをビットに変換するために 8 を掛け、1 秒あたりの変化量として表示します。

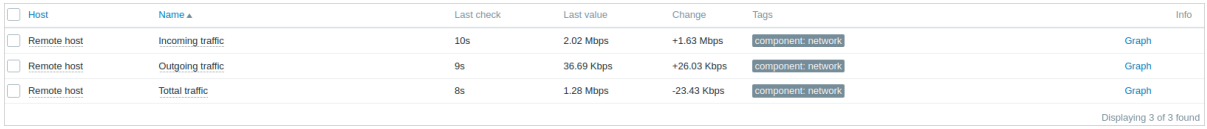


収集したデータの表示　おめでとうございます。この時点で、Zabbix はすでにネットワークトラフィックを監視しています。

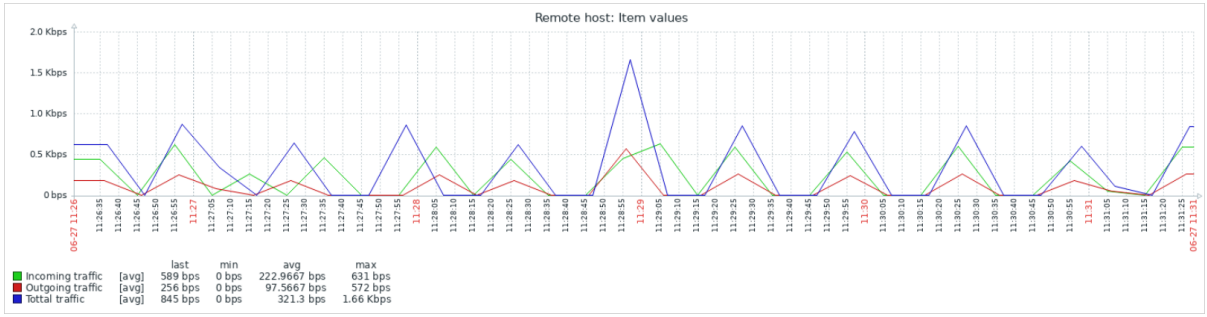
収集したメトリクスを表示するには、*Monitoring → Hosts* メニューセクションを開き、ホストの行にある *Latest data* をクリックします。



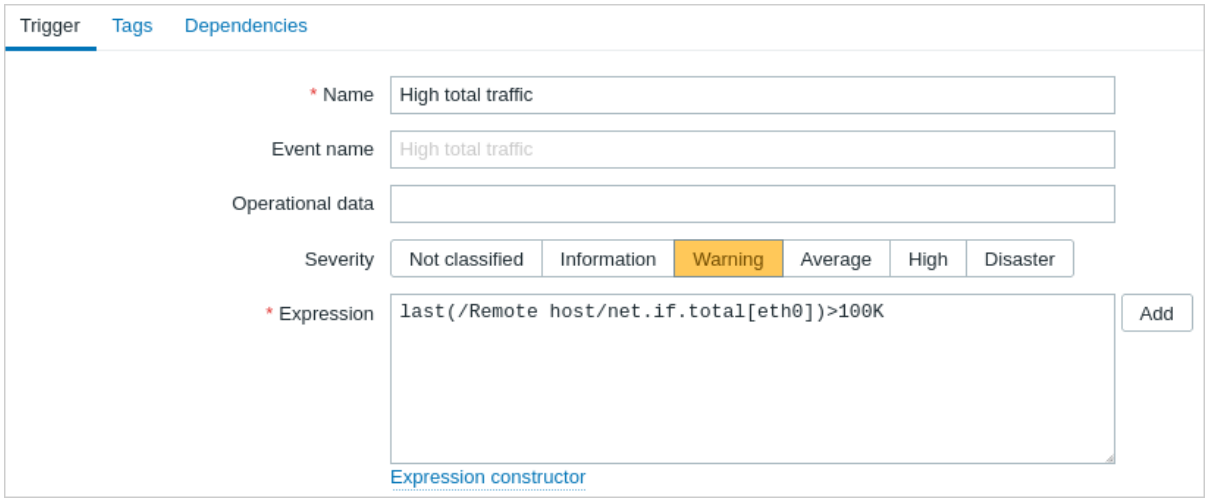
トラフィックデータが表示されます。



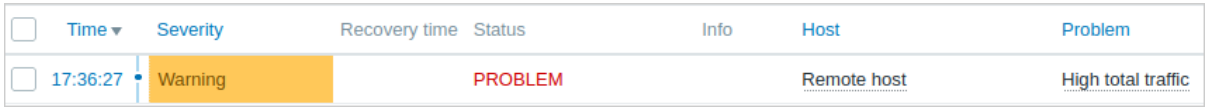
グラフの表示　収集されたデータは、グラフとして表示できます。
これらを表示するには、最新データセクションで対象のアイテムの行にあるグラフをクリックするか、
必要なアイテムを選択して下部のグラフを表示をクリックします。



トリガーの設定　異常なネットワークトラフィックを検出するために、トリガーを設定できます。トリガーの設定の手順を参照し、例えば合計トラフィックが高すぎることを示すトリガーを追加します。



次に、トリガー条件式で設定したしきい値を超えるようにトラフィックを発生させ、監視 → 障害に移動して、問題がそこに一覧表示されていることを確認します。



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

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

関連項目**:**

- 障害のエスカレーション - 複数ステップのアラートシナリオを作成する方法
(例: まずシステム管理者にメッセージを送信し、その後、45 分以内に障害が解決されない場合は、データセンターマネージャーにメッセージを送信します)。
- 障害の確認 - 障害が既知であることを示し、その解決に関するコメントを追加し、障害を抑制またはクローズする方法。
- Zabbix エージェントによる Linux 監視 - 事前設定済みのテンプレートをリンクして、最も重要なアイテムの基本的な監視を開始する方法。

7 アクティブチェックを使用してネットワークトラフィックを監視する

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

このガイドの対象者

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

前提条件

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

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

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

Zabbix エージェントの設定

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

Zabbix エージェントを使用している場合:

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

Zabbix エージェント 2 を使用している場合:

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

2. Zabbix サーバーの IP アドレスまたは DNS 名（および必要に応じてポート）を ServerActive パラメータに追加します。例:

```
ServerActive=192.0.2.0:10051
```

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

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

```
Hostname=Remote host
```

アクティブチェックでは、Zabbix エージェントがこの値を使用してサーバーから正しいホスト設定を取得するため、Hostname の値は一致している必要があります。具体的には、エージェントはサーバーへの接続を開始し、Hostname の値を使用して自身を識別し

ます。その後、サーバーはそのホストの監視設定を提供します。これらの値が異なる場合、エージェントは適切な設定を受信できず、メトリクスの欠落や監視上の問題が発生します。

4. Zabbix エージェントを再起動します。

Zabbix エージェントを使用している場合:

```
systemctl restart zabbix-agent
```

Zabbix エージェント 2 を使用している場合:

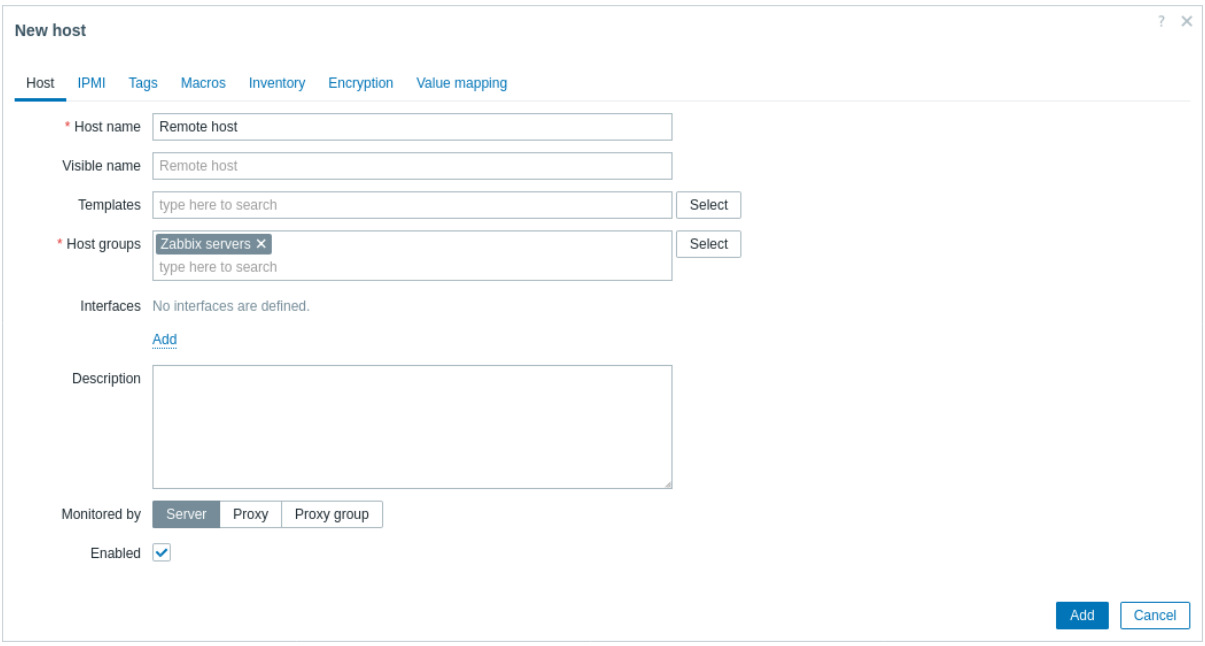
```
systemctl restart zabbix-agent2
```

Zabbix Web インターフェース

1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

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

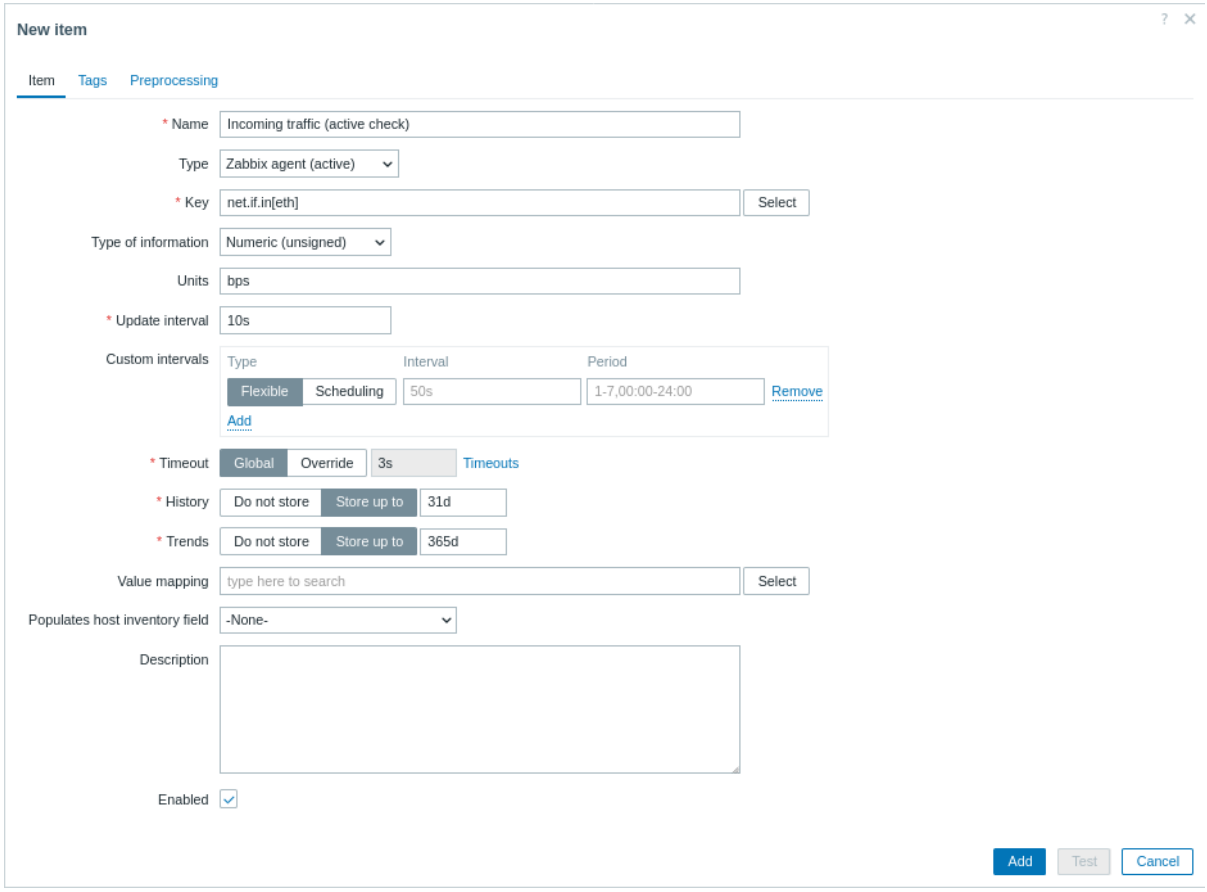


3. *Add* をクリックしてホストを追加します。このホストは、監視対象の Linux マシンを表します。

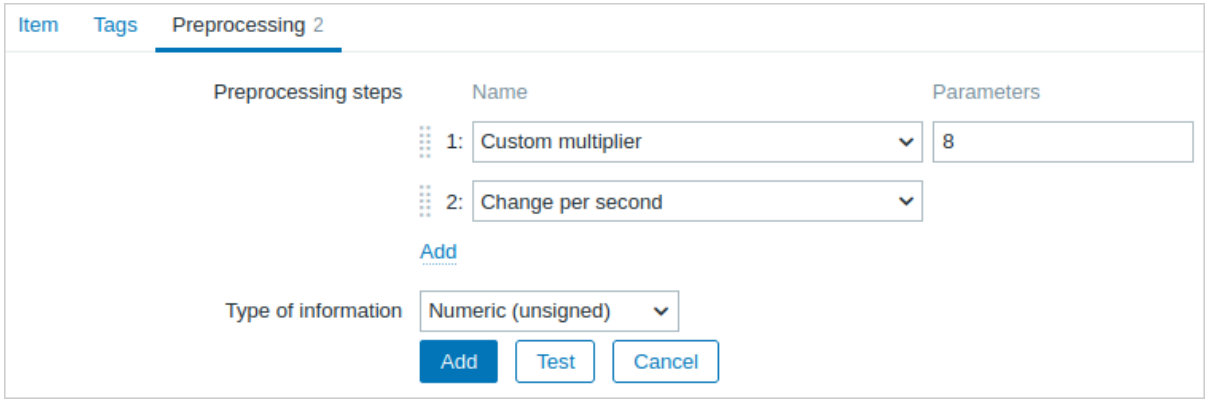
アイテムの作成　トラフィック監視用のアイテム、すなわち以下を追加するには、アイテムの作成 の手順に従ってください。

- 受信トラフィック
- 送信トラフィック
- 総トラフィック

アクティブチェックを使用した受信トラフィック監視アイテムのシンプルな設定例は、次のようになります。

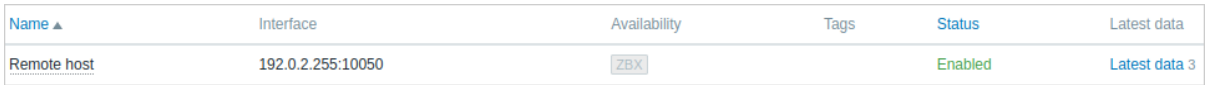


収集したデータを実用に適したものにするため、アイテムの作成時にいくつかの前処理 手順を設定できます。この場合は、8 倍すること（バイトをビットに変換するため）と、1 秒あたりの変化量として表示することが該当します。

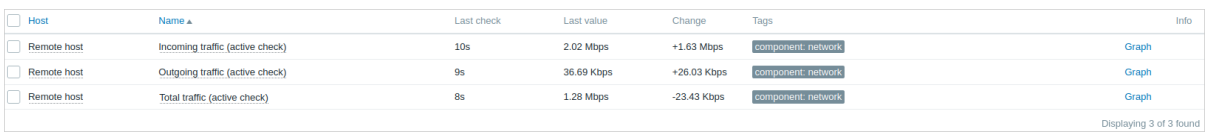


収集したデータの表示　おめでとうございます。この時点で、Zabbix はすでにネットワークトラフィックを監視しています。

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



トラフィックデータが表示されます。



関連項目:

- グラフの表示 - 収集したデータをグラフとして表示する方法。
- トリガーの設定 - 異常なネットワークトラフィックを検出するためのトリガーを設定する方法。
- 障害アラートの設定 - 障害の発生時に通知を設定する方法。

- 障害のエスカレーション - 複数段階のアラートシナリオを作成する方法
(例: 最初にシステム管理者にメッセージを送信し、その後、45 分以内に障害が解決されない場合は、データセンター管理者にメッセージを送信します)。
- 障害の確認 - 障害が既知であることを示し、その解決に関するコメントを追加したり、障害を抑制またはクローズしたりする方法。
- Zabbix エージェントによる Linux 監視 - 事前設定済みのテンプレートをリンクして、最も重要なアイテムの基本的な監視を開始する方法。
- パッケージからのインストール - 各種 Linux ディストリビューション向けの公式 RPM および DEB パッケージを使用して Zabbix コンポーネントをインストールし、最新機能とバグ修正を利用できるようにする方法。

8 ブラウザアイテムでウェブサイトの監視

はじめに

このページでは、ブラウザアイテムを使用してウェブサイトの基本的な監視を開始するために必要な手順について説明します。

このガイドの対象者

このガイドは、Zabbix を初めて利用するユーザー向けに作成されており、Browser アイテムを使用して Web サイトの基本的な監視を有効にするために必要な最小限の手順を説明しています。
詳細なカスタマイズオプションを探している場合や、より高度な設定が必要な場合は、Zabbix マニュアルのBrowser アイテム ページまたは設定 セクションを参照してください。

前提条件

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

環境によっては、このガイドの一部の手順が若干異なる場合があります。
このガイドは、Ubuntu が稼働している環境を前提としています。

WebDriver の設定

ブラウザアイテムでは、ブラウザを制御して操作し、ボタンのクリックやテキスト入力などのテストコマンドを実行する Web テストエンドポイントとして、自動化フレームワーク（Selenium Server またはプレーンな WebDriver。例: ChromeDriver）が必要です。
例として、このガイドでは Docker コンテナ内で Chrome とともに Selenium Server を使用します。

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

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

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

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

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

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

```
ip addr

### 1: lo: <LOOPBACK,UP,LOWER_UP>
###     ...
### 3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> ...
```

```
###     inet 192.0.2.1/16 brd 192.0.255.255 scope global docker0
###     ...
```

- Ncatを使用してコンテナへの接続をテストします。

```
nc -zv 192.0.2.1 4444

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

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

```
curl -L 192.0.2.1:4444

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

Note:
トラブルシューティングについては、Docker documentationを参照してください。

Zabbix サーバーの設定

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

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

1. Zabbix サーバーの設定ファイルを開きます。

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

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

```
##### オプション: WebDriverURL
###WebDriver インターフェースの HTTP[S] URL。たとえば、Selenium WebDriver スタンドアロンサーバーとともに使用する http:
###
### 必須: いいえ
### デフォルト:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Zabbix サーバーを再起動します。

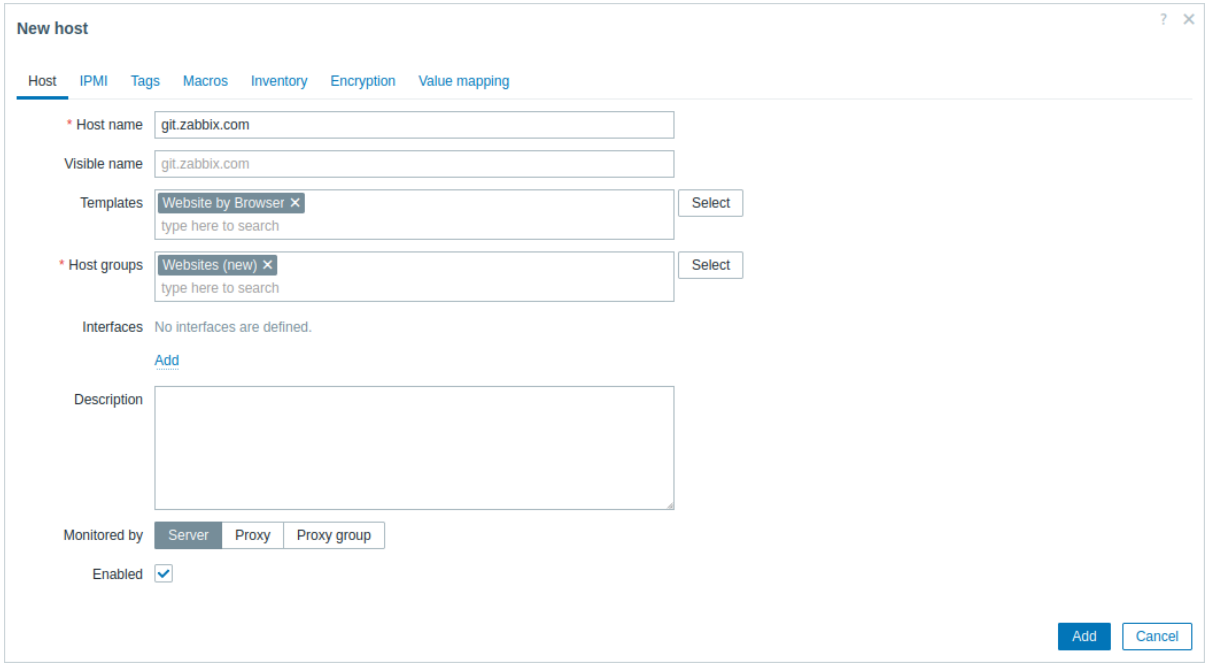
```
systemctl restart zabbix-server
```

Zabbix Web インターフェースの設定

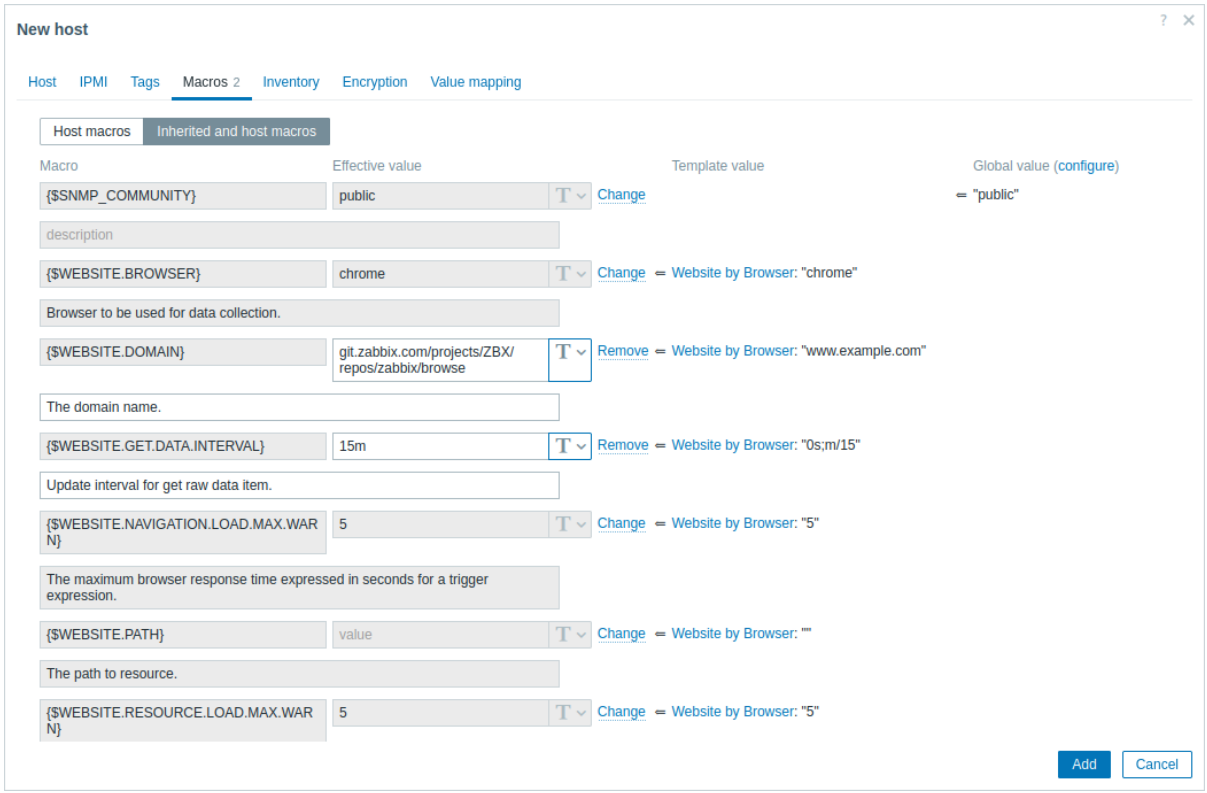
1. Zabbix Web インターフェースにログインします。

2. Zabbix Web インターフェースでホストを作成します。

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



- *Macros* タブで、*Inherited and host macros* に切り替え、次のマクロを探して、マクロ値の横にある *Change* をクリックして更新します。
  - {$WEBSITE.DOMAIN} - ドメイン名（例: git.zabbix.com/projects/ZBX/repos/zabbix/browse）
  - {$WEBSITE.GET.DATA.INTERVAL} - アイテムデータの更新間隔（例: 15m）

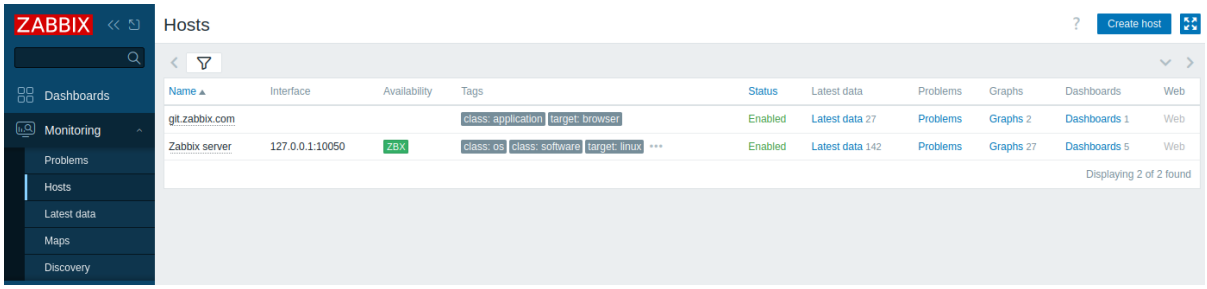


3. *Add* ボタンをクリックしてホストを作成します。このホストは、監視したい Web サイトを表します。

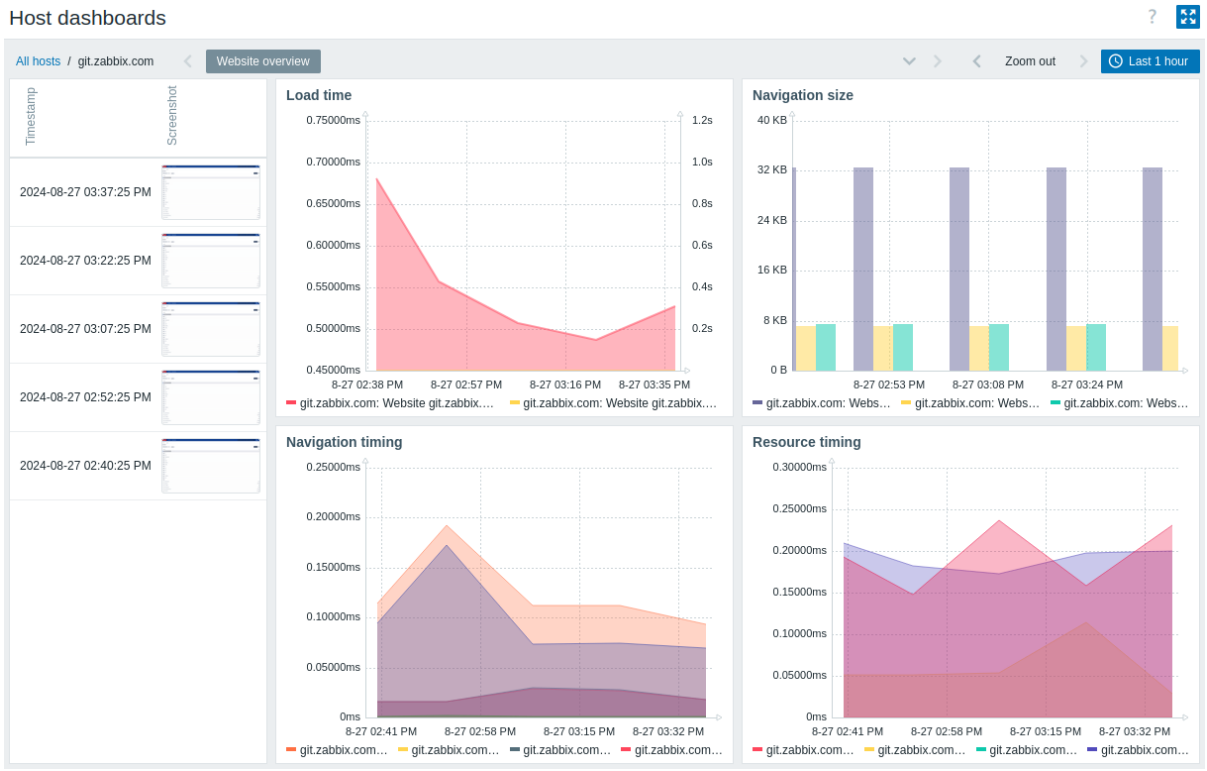
収集されたメトリクスを表示する

おめでとうございます。ここまでで、Zabbix は指定した Web サイトの監視をすでに開始しています。

収集されたメトリクスを表示するには、監視 → ホスト メニューセクションに移動し、そのホストの横にある ダッシュボードをクリックします。



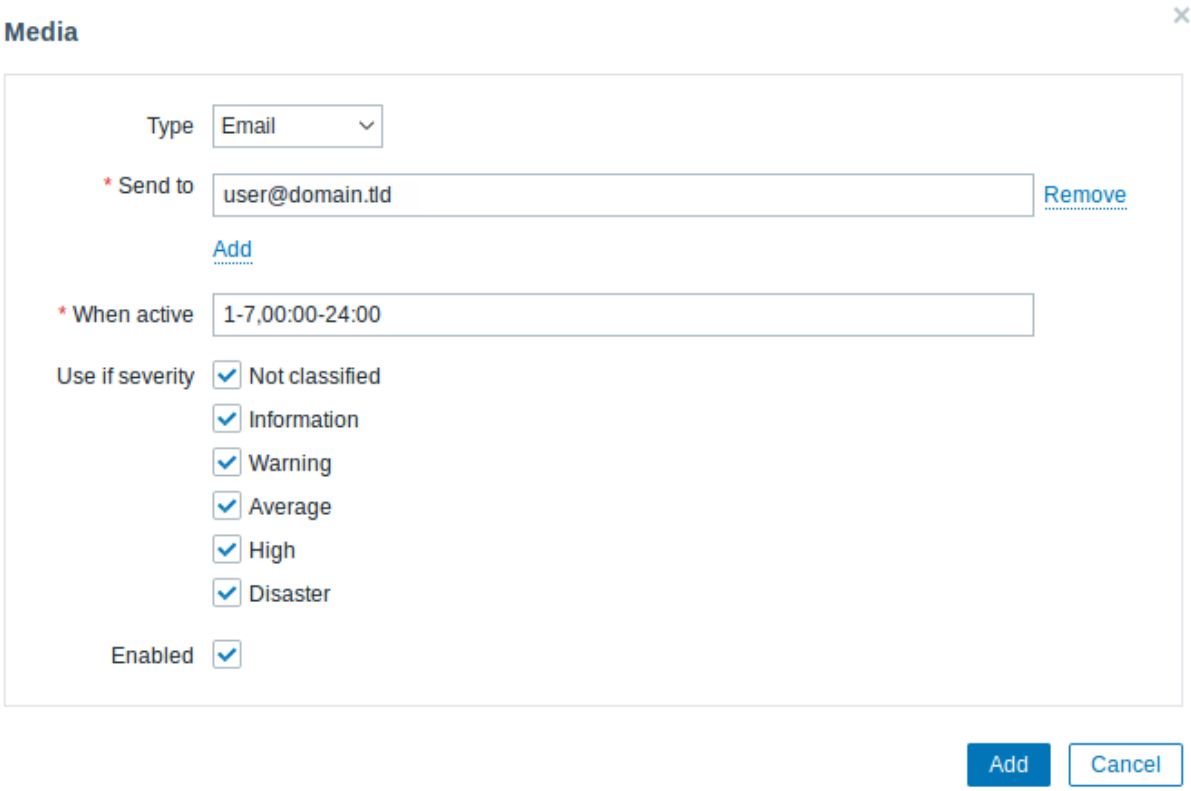
この操作により、ホストダッシュボード（テンプレートレベルで設定）に移動し、Web サイトから収集された最も重要なメトリクスを確認できます。



問題アラートの設定

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

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



2. 問題通知を受信する のガイドに従います。

次回、Zabbix が問題を検出したときに、メールでアラートを受信できるようになります。

設定をテストする

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

1. Zabbix で Web サイトのホスト設定を開きます。

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

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

4. *Update* をクリックしてホスト設定を更新します。

5. しばらくすると、指定した Web サイトに接続できないため、Zabbix は「要求された Web サイトの JSON の取得に失敗しました」という問題を検出します。この問題は、*Monitoring → Problems* に表示されます。



アラートが設定されている場合は、問題通知も受信します。

6. 問題を解決して Web サイトの監視を継続するには、マクロ値を元の値に戻します。

関連項目

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

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

はじめに

このガイドでは、Zabbix エージェント 2 の `web.certificate.get` キーを使用して SSL/TLS 証明書を設定および監視する方法について、包括的な概要を説明します。これは、単一または複数の Web サイトの証明書監視を簡素化することを目的としており、管理者が期限切れや無効な証明書などの潜在的な問題を迅速に特定できるようにします。

このガイドの対象者

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

前提条件

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

Zabbix エージェント 2 の設定

1. Zabbix エージェントの設定ファイル（デフォルトパス: `/etc/zabbix/zabbix_agent2.conf`）を開きます:

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

2. エージェントとサーバーが同じマシン上で動作しているため、Server パラメータを 127.0.0.1 に設定します:

```
Server=127.0.0.1
```

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

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

4. セットアップと Zabbix エージェント 2 の設定後、次のコマンドで可用性をテストします:

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

Zabbix エージェント 2 には WebCertificate プラグインがデフォルトで含まれているため、個別のインストールや設定は不要です。

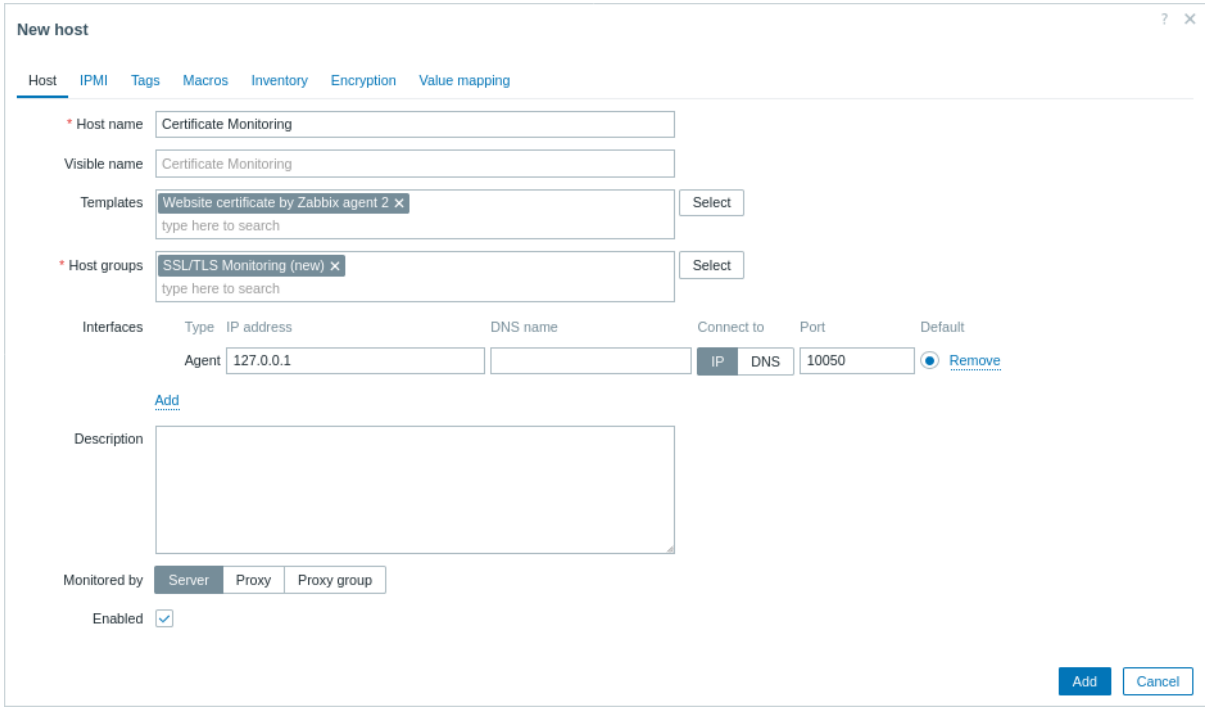
Zabbix Web インターフェースの設定

1. Zabbix Web インターフェースにログインします。

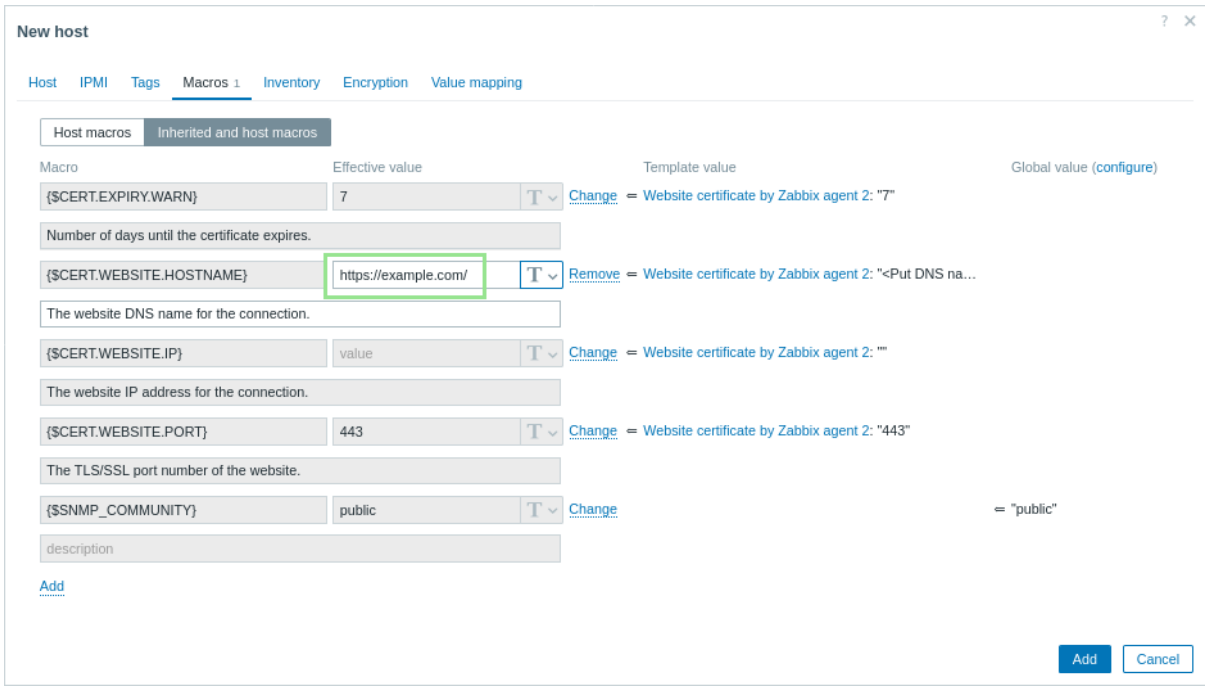
2. 監視 > ホストに移動します。

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

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



- マクロタブで、継承したホストマクロとホストマクロに切り替え、次のマクロを探して、マクロ値の横にある 変更をクリックして更新します。
  - `{$CERT.WEBSITE.HOSTNAME}` - 値として目的の Web サイトの DNS 名を入力します。



4. 追加をクリックしてホストを追加します。

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

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

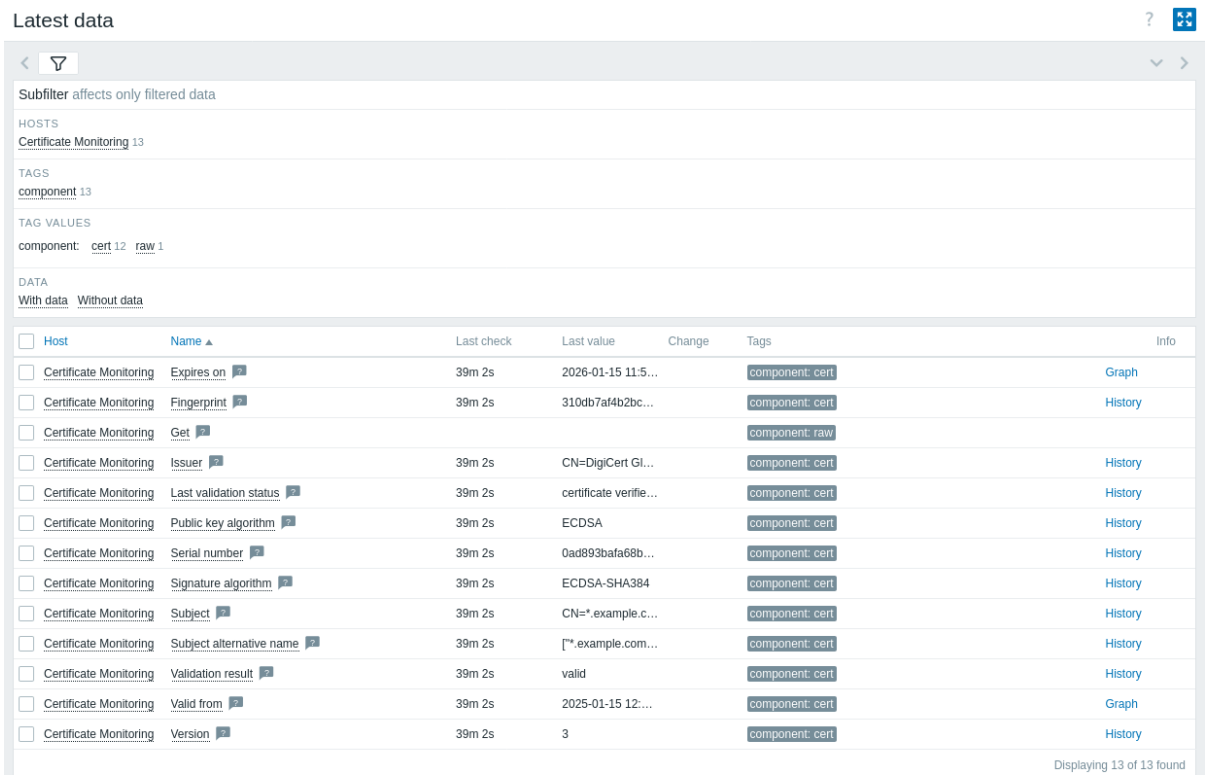
`{$CERT.WEBSITE.HOSTNAME}` マクロで指定した各 Web サイトに対して、Zabbix は対応する一連のアイテムとトリガーを作成します。これにより、各 Web サイトの SSL 証明書を個別に監視し、アラートを生成できます。

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

収集したメトリクスを表示する

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

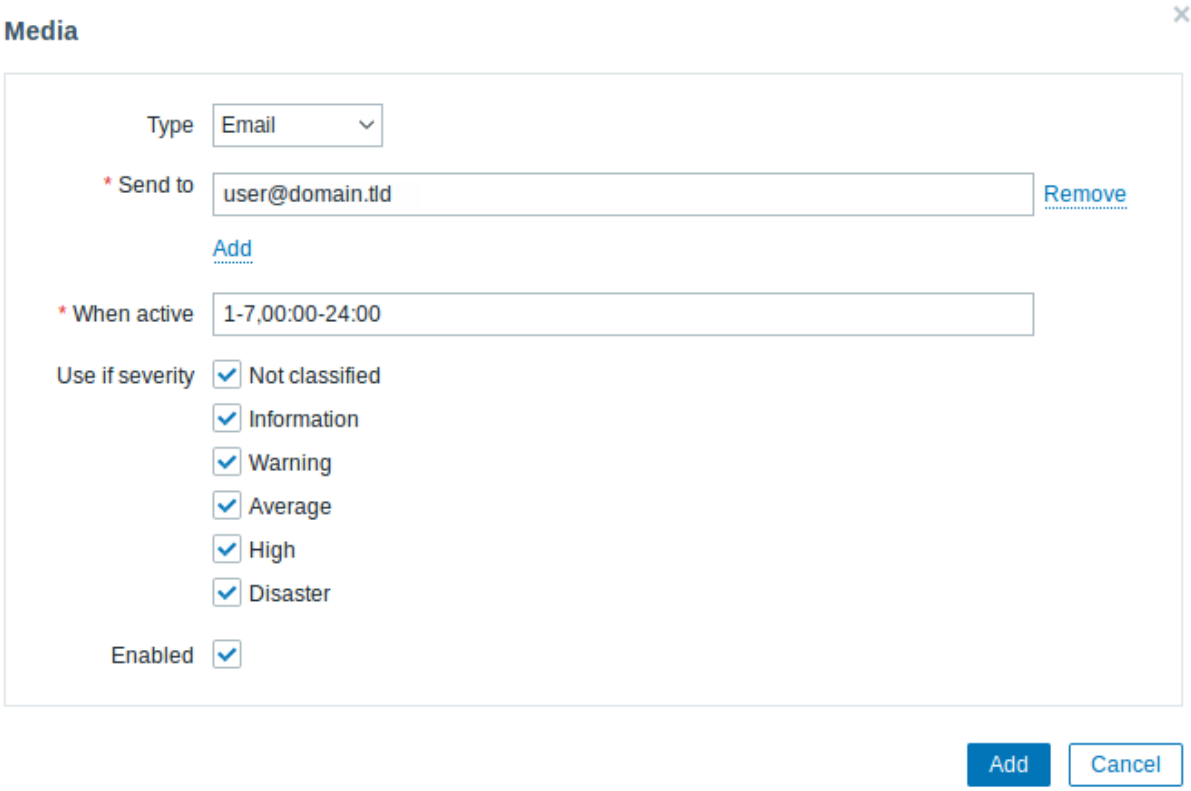
収集したメトリクスを表示するには、監視 > ホスト メニューセクションに移動し、ホストの横にある最新データ をクリックして、収集された最新のメトリクスを有効期限、発行者、サブジェクトなどの一覧で表示します。



問題アラートの設定

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

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



2. 問題通知を受信する のガイドに従ってください。

次回、Zabbix が問題を検出したときに、メールでアラートを受信できるようになります。

設定をテストする

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

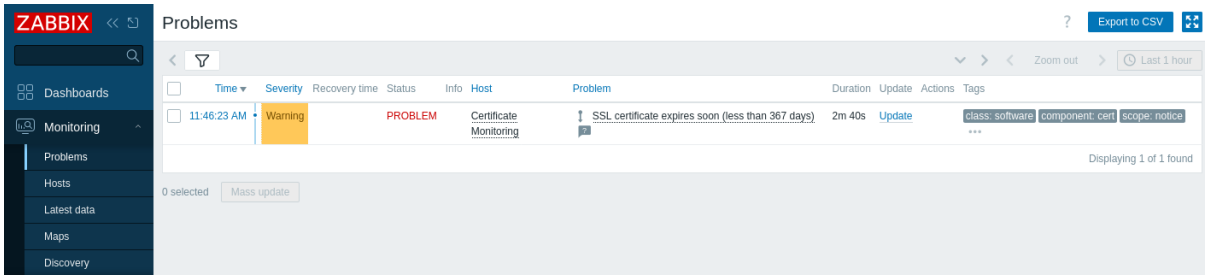
1. Zabbix で「Certificate Monitoring」ホストの設定を開きます。

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

3. 以前に設定した {$CERT.EXPIRY.WARN} マクロ値の横にある *Change* をクリックし、証明書の有効期限が切れる前に警告を受け取れるよう、日数を非常に大きな値（365 日より大きい値で十分です）に設定します。

4. *Update* をクリックしてホスト設定を更新します。

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



アラートが設定されている場合は、問題の通知も受信します。

6. 問題を解決し、証明書の値の監視を継続するために、マクロ値を元の値に戻します。

関連項目

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

10 Zabbix でネットワークスイッチやルーターを監視する

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

このガイドの対象者

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

前提条件

このガイドを進める前に、以下を満たしていることを確認してください。

- Zabbix サーバーおよび Zabbix Web インターフェースがインストールされていること: ご使用のオペレーティングシステム向けの手順に従ってインストールしてください（パッケージからのインストールおよびWeb インターフェースのインストールを参照）。
- ローカルネットワークメトリクスを監視する場合は、Zabbix エージェントがインストールされていること。
- SNMP 対応デバイス: SNMP が有効化されたネットワークスイッチまたはルーター（たとえば Cisco ルーター）。
- MIB ファイルがインストールされていること: MIB ファイルをインストールすると、Zabbix は数値 OID を人が読める名前や説明に変換できるようになります。適切な MIB サポートがない場合、数値のみが表示されることがあり、アイテムの設定や問題のトラブルシュートが難しくなります。

Ubuntu に MIB ファイルをインストールするには:

1. MIB ダウンローダーパッケージをインストールします:

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

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

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

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

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

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

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

- Cisco MIBs
- Juniper MIBs

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

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

環境によっては、このガイド内の一部の手順が若干異なる場合があります。このガイドは、Ubuntu が稼働し、Cisco Catalyst 3750V2-24FS ネットワークデバイスを監視している環境を前提としています。

ネットワークデバイスは、すでに物理的に設置され接続されているものとします。

ネットワークデバイスを設定する（**Cisco** ルーターの例）　SNMP による監視を行うには、SNMP クエリを許可するようにネットワークデバイスを設定する必要があります。
以下の例は SNMPv2 を対象としており、既存の設定は考慮していません。
注意: これらのコマンドを適用すると、現在の SNMP 設定が上書きされる可能性があります。

Cisco ルーターでは、通常、設定には以下に記載した手順が含まれます。

SNMPv2 の例　1. SNMP を有効にし、コミュニティ文字列を設定します。

Cisco ルーターのコンソールにログインし、設定モードに入ります。

```
configure terminal
```

次に、読み取り専用のコミュニティ文字列を指定してSNMP を有効化します。例:

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

`<your_community_string>` は、安全なコミュニティ文字列に置き換えてください。注: RO（Read-Only）オプションを指定すると、SNMP でデバイスからデータを取得できますが、設定変更はできません。

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

2. 設定を保存します。

再起動後も SNMP 設定が保持されるように、変更を保存します。

```
write memory
```

SNMPv3 の例　SNMPv3 は、認証と暗号化によって強化されたセキュリティを提供します。
その設定は SNMPv2 よりも安全ですが、お使いのデバイス固有のドキュメントに照らして確認する必要があります。

1. SNMP グループを作成します。

プライバシー（暗号化）を有効にした SNMPv3 グループを設定します。

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

2. SNMP ユーザーを作成します。

認証とプライバシーを備えた SNMPv3 ユーザーを追加します。
プレースホルダーは任意の値に置き換えてください。

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

3. 設定を保存します。

```
write memory
```

詳細やモデル固有の手順については、外部の Cisco SNMP configuration tutorials を参照できます。
ただし、このガイドでは SNMP 監視を有効にするための基本的な手順を説明しています。

Zabbix Web インターフェースの設定

Zabbix Web インターフェースでホストを作成する　1. Zabbix Web インターフェースにログインします。

2. 新しいホストを追加します。

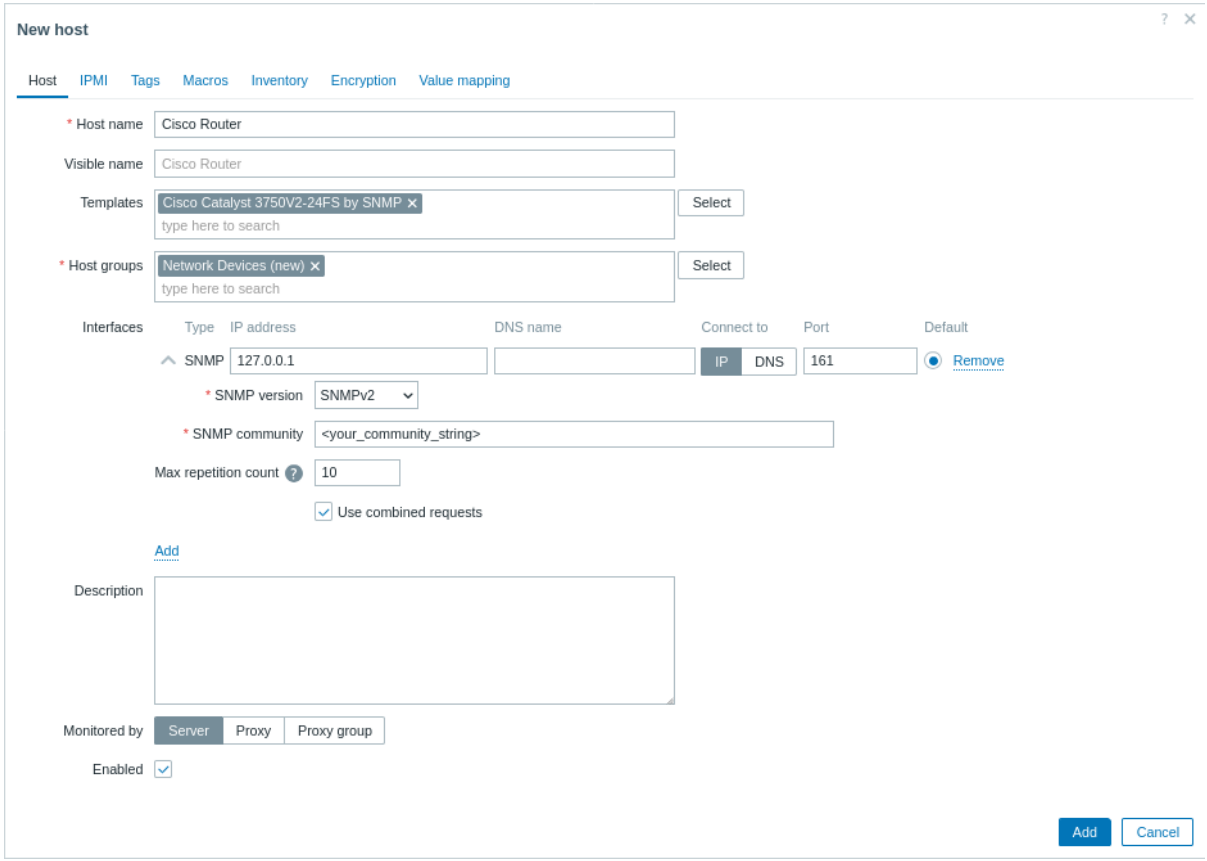
データ収集 > ホストに移動し、ホストの作成をクリックします。

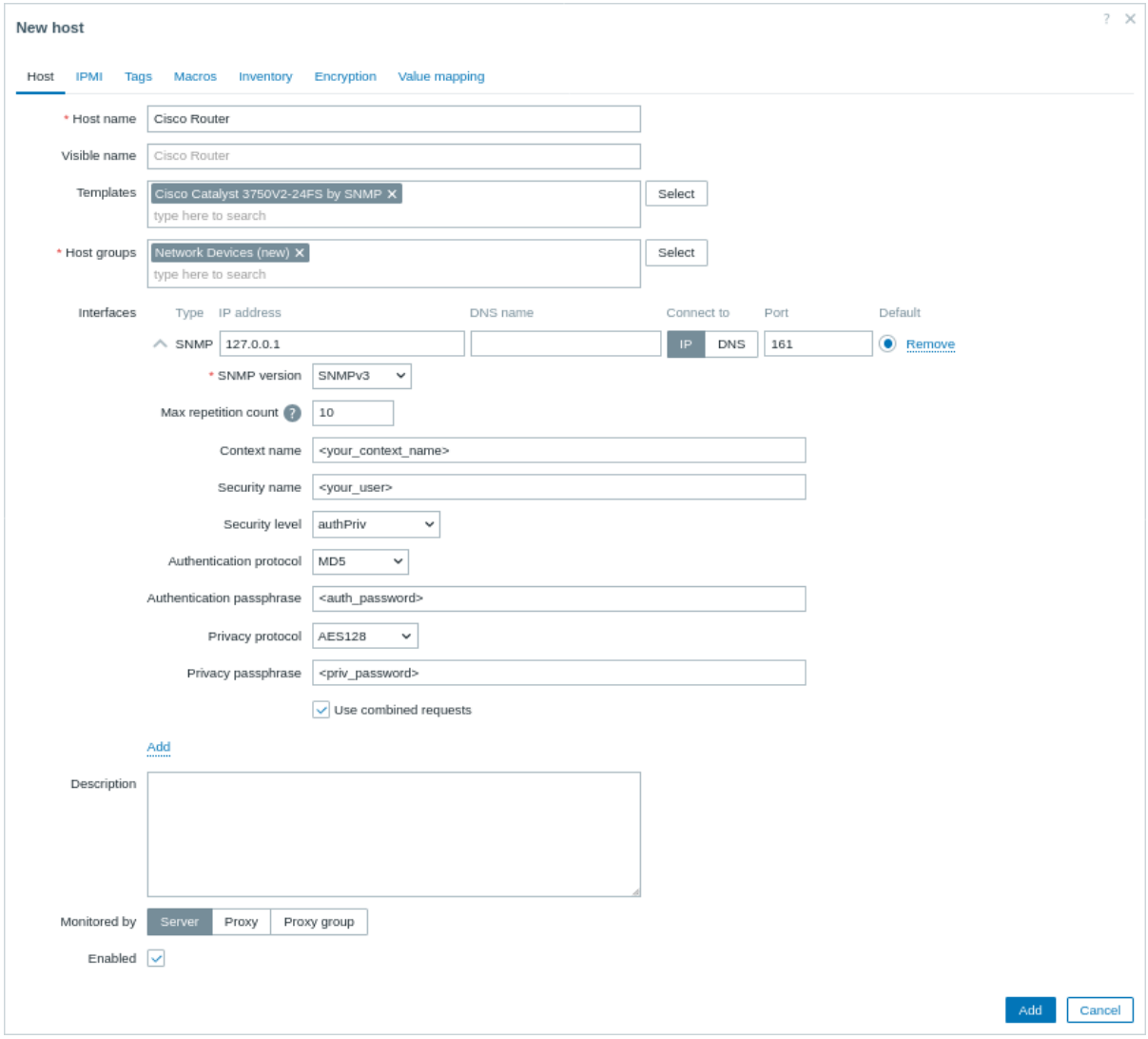
- ホスト名: デバイスの名前を入力します（例: 「Cisco Router」）。
- ホストグループ: 既存のグループを選択するか、「Network Devices」などの新しいグループを作成します。
- インターフェース:
  - インターフェースの下にある 追加をクリックします。
  - インターフェースタイプとして *SNMP* を選択します。
  - Cisco ルーターの IP アドレスまたは DNS 名を入力します。
  - デフォルトの SNMP ポート（通常は 161）を設定します。
  - ドロップダウンメニューを使用して、適切な SNMP バージョン（例: SNMPv2）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

3. テンプレートをリンクします

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

4. 追加をクリックしてホストを保存します。

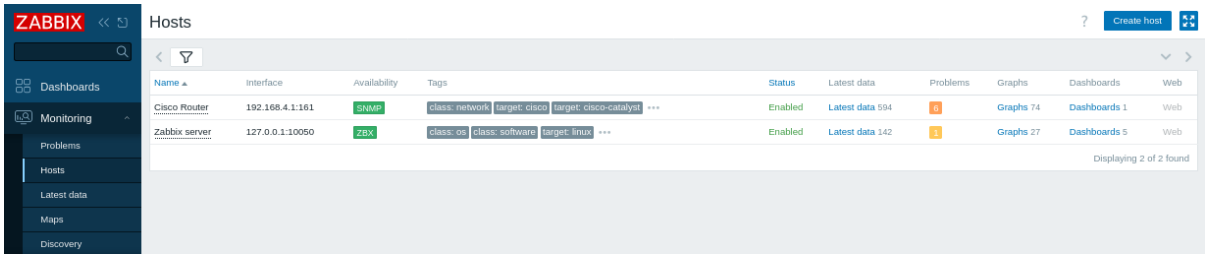




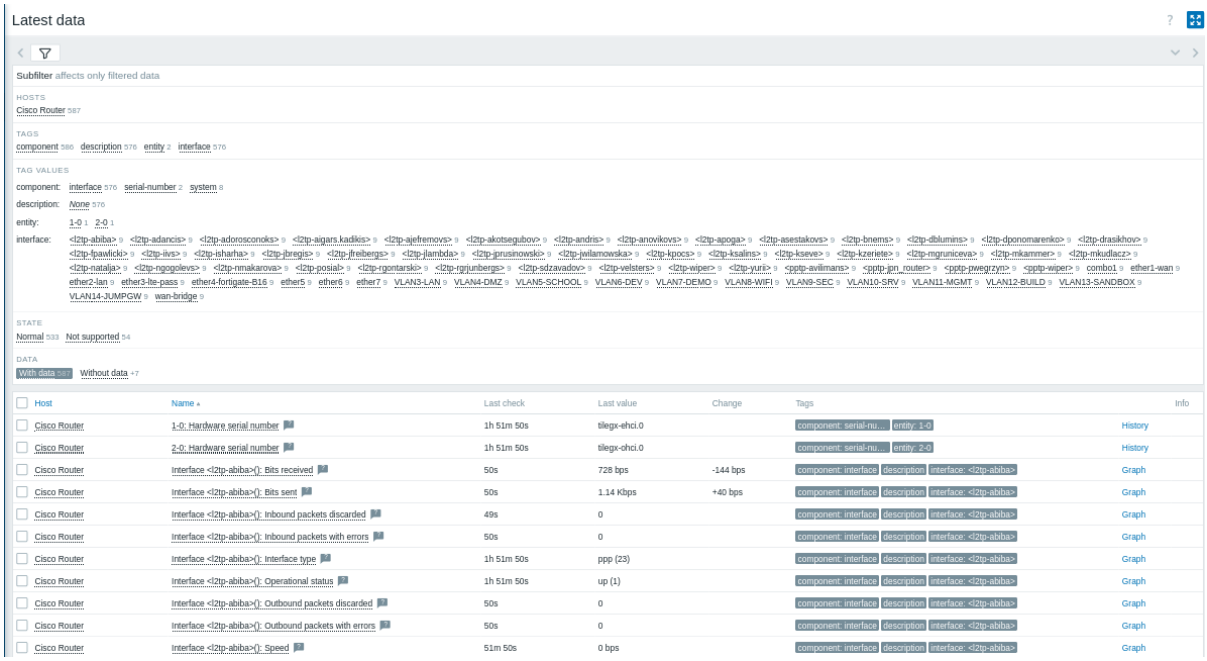
収集したメトリクスを表示する　おめでとうございます。これで、Zabbix はネットワークデバイスを監視するように設定されました。

最新データ:

- Zabbix Web インターフェースで、監視 > 最新データに移動します。



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



- グラフとスクリーン:

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

次のステップとして、以下を実行できます。

- カスタム SNMP アイテムを追加して、追加のメトリクスを監視する。
- 障害アラートを設定して、潜在的な問題に関する通知を受け取る。

SNMP アイテムの作成　ホストの設定が完了したら、特定のメトリクスを監視するためのアイテムを作成できます。
注: テンプレートを使用している場合、この手順は任意です。テンプレートには既にデフォルトのアイテムセットが含まれているためです。

1. SNMP OID を特定します:

`snmpwalk` コマンドを使用して、デバイスで利用可能な OID を一覧表示します。
例:

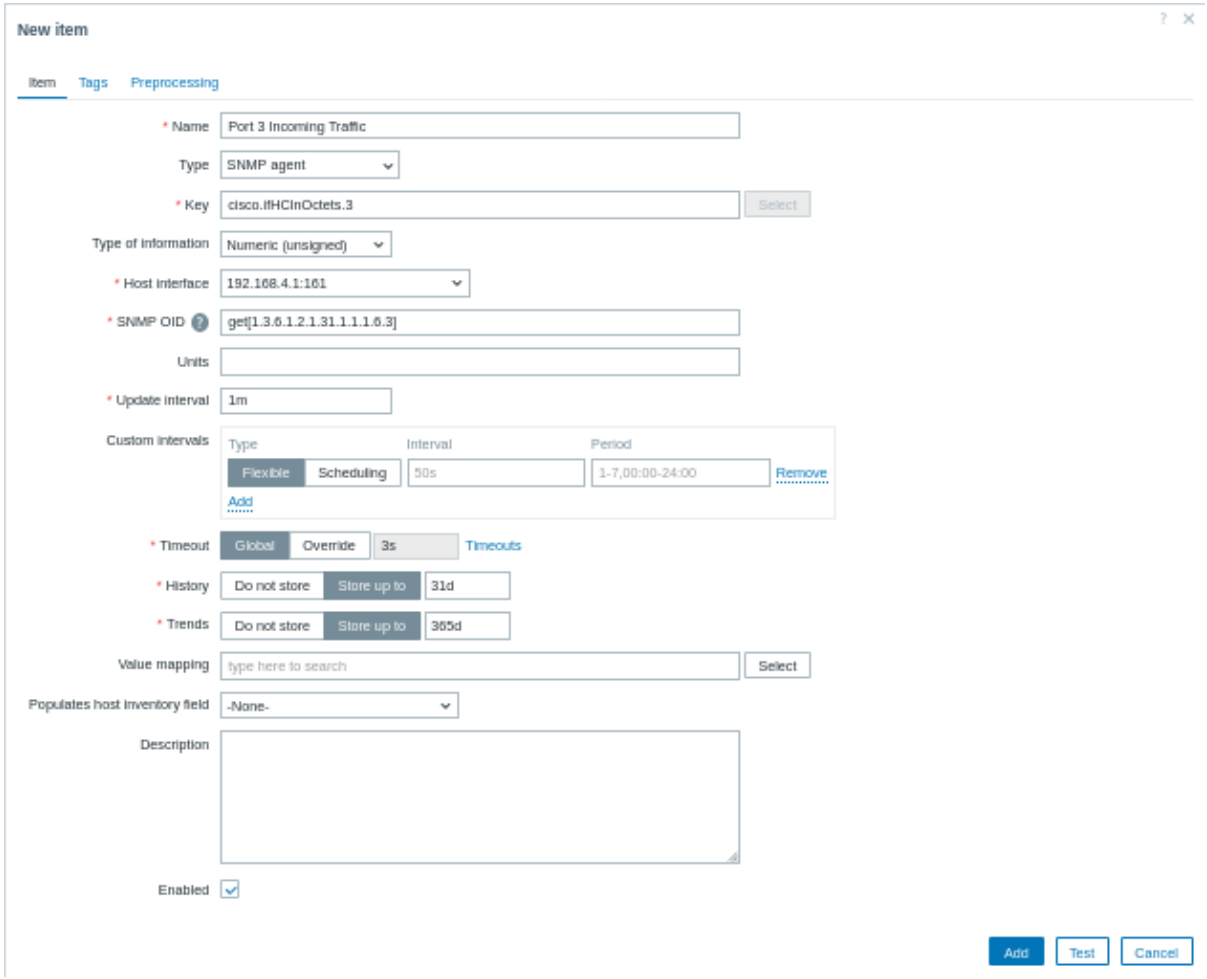
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

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

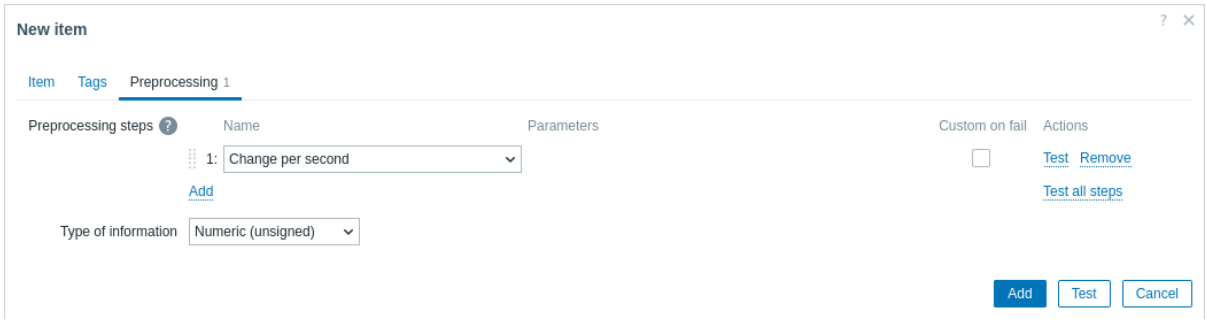
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. SNMP アイテムを作成します:

- *Data collection* > *Hosts* に移動し、SNMP ホストの *Items* タブをクリックして、*Create item* をクリックします。
- *Name*: わかりやすい名前を入力します（例: 「ポート 3 の受信トラフィック」）。
- *Type*: *SNMP agent* を選択します。
- *Key*: 意味のあるキーを指定します（例: `cisco.ifHCInOctets.3`）。
- *Host interface*: SNMP インターフェースが選択されていることを確認します。
- *SNMP OID*: サポートされている形式のいずれかで OID を入力します。例:
  - 単一の値の場合は `get[1.3.6.1.2.1.31.1.1.1.6.3]`;
  - 値のサブツリーを非同期に取得するには `walk[1.3.6.1.2.1.31.1.1.1.6.3]`。



- *Preprocessing*（必要な場合）: アイテムが累積カウンター（インターフェーストラフィックなど）を返す場合は、*Preprocessing* タブに移動し、「秒ごとの差分」のような前処理ステップを追加して、レートを計算します。



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

数値 **OID** と **MIB** 名の相互変換　SNMP を扱う際には、数値 OID と対応する MIB 名の間で変換する必要が生じることがあります。この変換により、メトリクスの特定やトラブルシューティングをより容易に行えます。

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

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

このコマンドの出力例:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- 数値 OID を MIB 名に変換するには、`snmptranslate` コマンドに`-IR`（または`-m ALL`）オプションを付けて、逆方向の変換を行います。たとえば、数値 OID `.1.3.6.1.2.1.31.1.1.1.6.3` を MIB 名に戻すには、次を実行します。

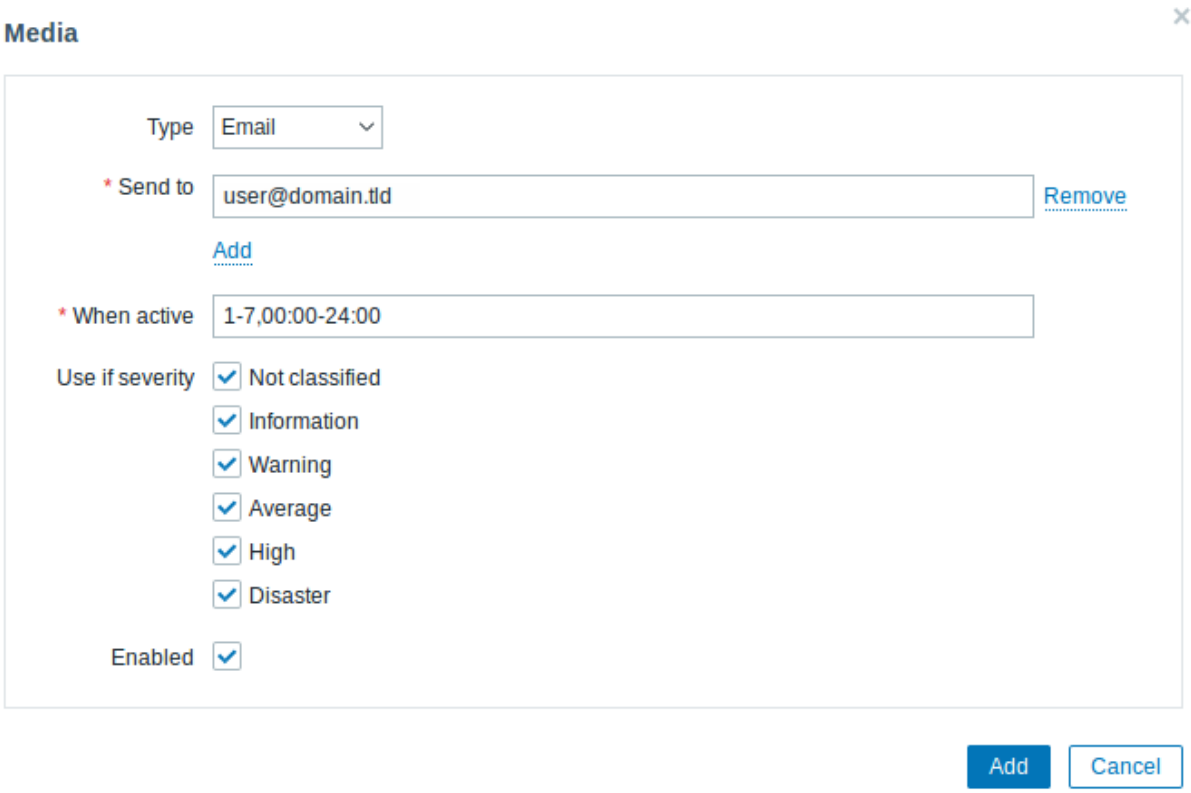
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

このコマンドの出力例:

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

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

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

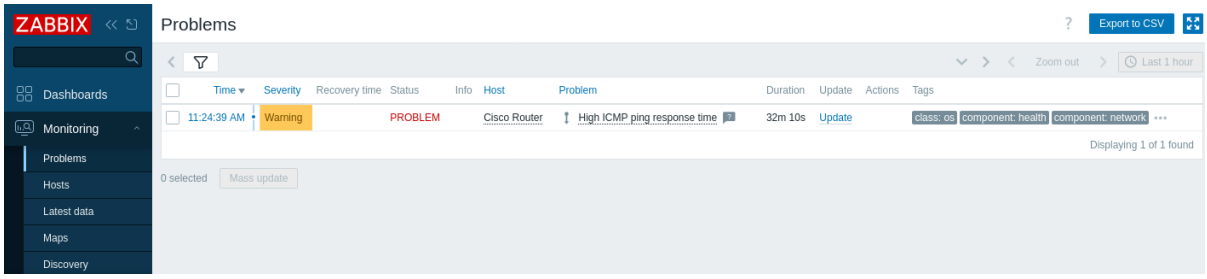


2. 問題通知を受信する のガイドに従ってください。

次回、Zabbix が問題を検知すると、メールでアラートを受信するはずです。

設定をテストする　Zabbix がネットワークパフォーマンスの問題を正しく検出することを確認するために、ICMP ping の応答時間しきい値を上げて実際の問題をシミュレートします。

1. Zabbix で「Cisco Router」ホストの設定を開きます。
2. マクロタブに移動し、継承されたホストマクロとホストマクロを選択します。
3. {$ICMP_RESPONSE_TIME_WARN} マクロ（または同様の応答時間しきい値マクロ）を探します。
4. ping 応答がこの値を超えたときにアラートが発生するよう、非常に低い値（例: 0.001）を設定します。
5. 更新をクリックして変更を適用します。
6. Zabbix がシミュレートした問題を検出するまで、しばらく待ちます。
7. 監視 > 障害に移動し、アラート（例: 「ICMP ping 応答時間が高い」）が表示されることを確認します。



アラートが設定されている場合は、障害アラートも受信するはずです。

8. マクロの値を元の設定に戻し、更新をクリックして変更を保存します。
9. 問題が解決され、障害セクションから消えることを確認します。

SNMP 監視のトラブルシューティング　Zabbix の Web インターフェースで SNMP アイコンが赤色で表示される、またはデータが収集されない場合は、以下の手順を試してください。

1. SNMP 接続を確認します。

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

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

このコマンドは、デバイスが SNMP クエリに応答することを確認します。

SNMPv3 の場合は、適切な SNMPv3 認証情報を含めます。

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

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

2. 前提条件で説明されているように、MIB ファイルがインストールされ、有効になっていることを確認します。
これを確認するには、ネットワークデバイスに対してクエリを実行した際に、次のコマンドでエラーが発生してはいけません。

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

これにより、エラーなしで変換済みの OID が返されるはずです。

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

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

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

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

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

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

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

nc -zv <device_IP> 161

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

さらに、Ubuntu で UFW を使用している場合は、ファイアウォールの状態を確認してください。

```
sudo ufw status
```

または、iptables の場合は次を実行します。

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

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

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

`tail -f` を使用すると、ログの更新をリアルタイムで監視できます。

関連項目**:**

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

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

はじめに

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

このガイドの対象者

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

前提条件

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

Windows イベントログ監視用に Zabbix エージェントを設定する

1. Windows ホスト上で `zabbix_agentd.conf`（デフォルトパス: `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`）を開き、ServerActive パラメータが Zabbix サーバーの IP アドレスに設定されていること、およびHostname パラメータがZabbix Web インターフェース で定義するホスト名と一致していることを確認します。これにより、エージェントは指定された Zabbix サーバーから自身のホストに対するアクティブチェックを要求できるようになります。例:

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

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

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

3. Windows ホストが稼働していることを確認します。

- Windows ホスト上で Zabbix エージェントサービスが実行中であることを確認します。
- Windows ホストからポート 10051 で Zabbix サーバーに接続できることを確認します。Windows ホストから接続性をテストするには、PowerShell を開いて次のコマンドを実行します。

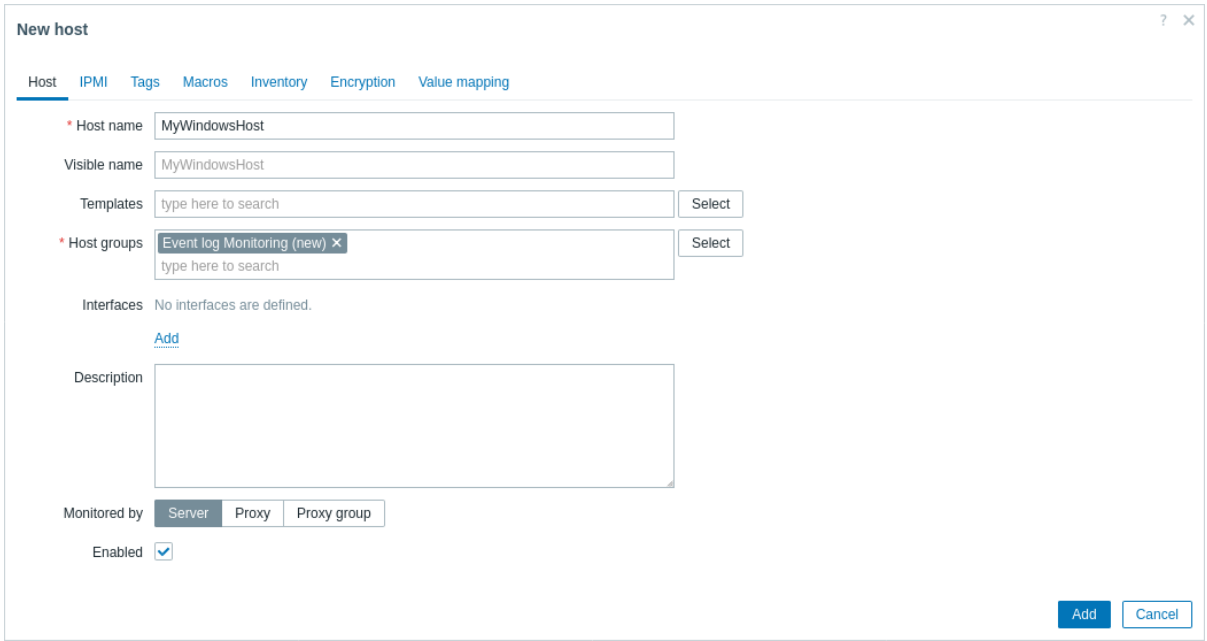
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Zabbix Web インターフェースの設定

1. データ収集 > ホストに移動し、ホストを作成します。

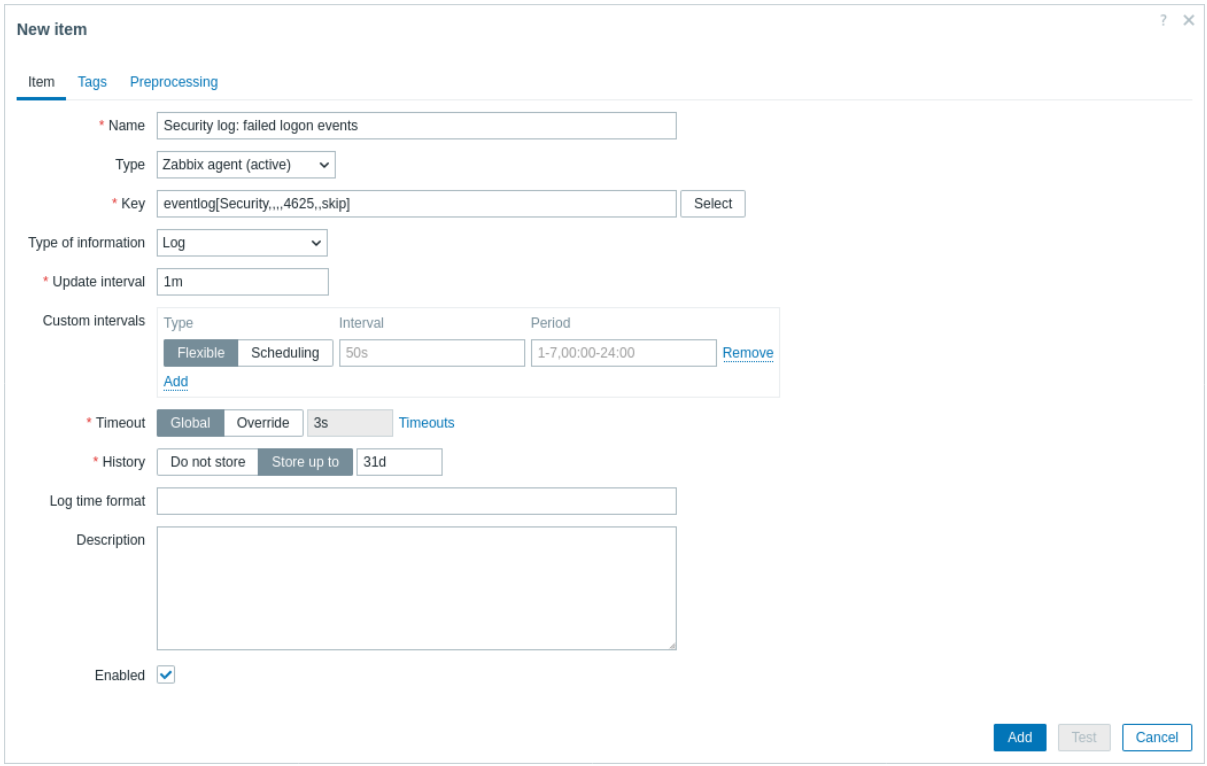
- ホスト名フィールドに、ホスト名（例: ”MyWindowsHost”）を入力します。
- ホストグループフィールドで、ホストグループを入力または選択します（例: ”Event log Monitoring”）。
- 追加を押して、設定したホストを保存します。

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



2. 次のパラメータで新しいアイテムを作成します。

- 名前フィールドに、わかりやすいアイテム名を入力します（例: ”Security log: failed logon events”）。
- タイプドロップダウンで、”Zabbix agent (active)” を選択します（イベントログ監視に必須）。
- キーフィールドで、eventlog アイテムキーを使用します。たとえば、Security ログ内のログオン失敗試行（イベント ID: 4625）を監視し、アイテムの前回チェック時点より古いエントリを無視する（`skip` パラメータを使用する）には、次のアイテムキーを入力します: `eventlog[Security,,,,4625,,skip]`
- 情報の型ドロップダウンで、” ログ” を選択します。



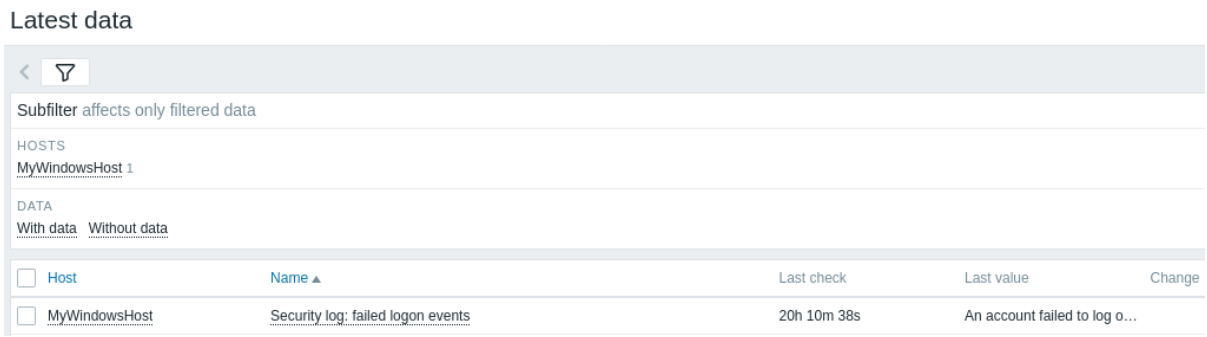
3. 追加をクリックしてアイテムを保存します。

収集したメトリクスのテストと表示

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

次に、Zabbix Web インターフェースで収集したログを表示します。

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



2. *Name* フィールドで、「MyWindowsHost」ホストを指定して絞り込みます。

3. *History* をクリックして、記録されたログの値を表示します。

MyWindowsHost: Security log: failed logon events

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

4. ログの値が表示されない場合は、このガイドのトラブルシューティング セクションに進んでください。

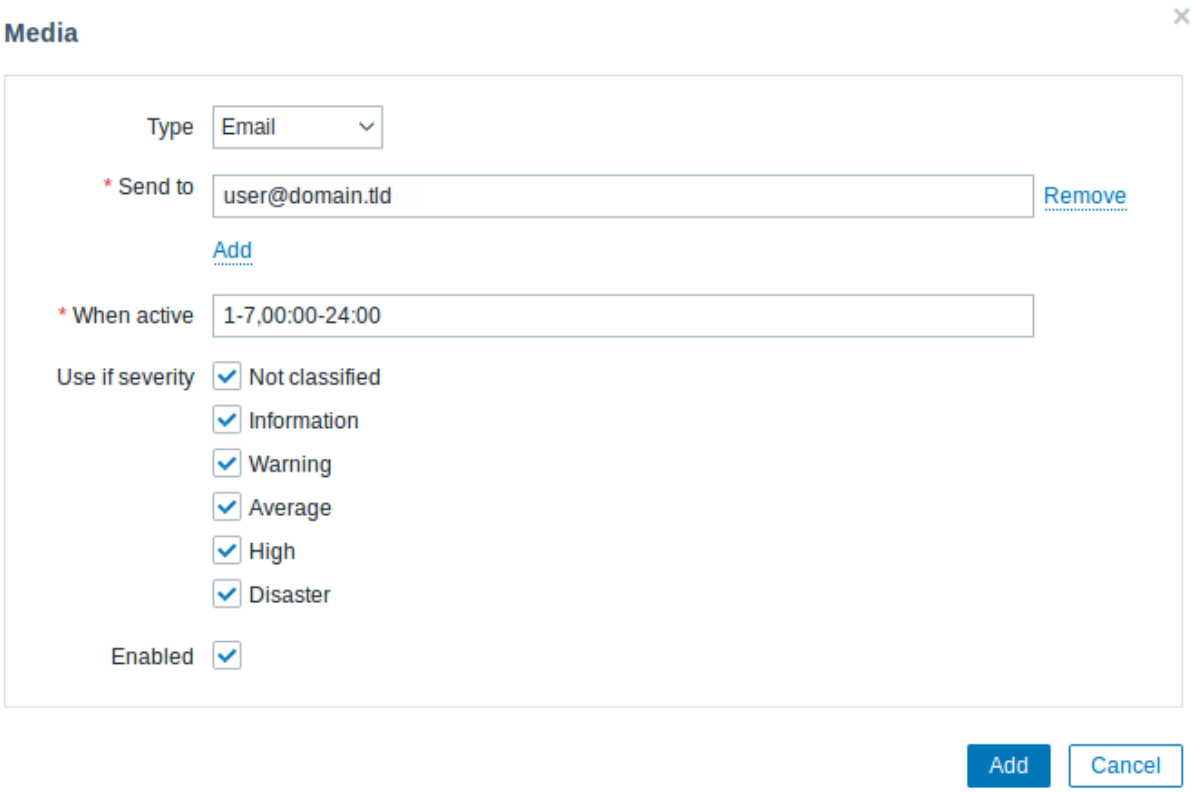
問題アラートの設定

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

1. Data collection > Hosts に移動し、イベントログのアイテムが監視したいパターンを記録したときに発生するトリガーを定義します。例えば、Security ログ内のログオン失敗の試行を検出するには、find() 関数を使用します。

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

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



3. 問題通知を受信する ガイドに従います。

次回 Zabbix が問題を検出したときに、メールでアラートを受信できるようになります。

トラブルシューティング

Windows イベントログの収集または表示で問題が発生した場合は、以下のヒントを使用して一般的な問題を特定し、解決してください。

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

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

TCP ポート 10051 に対する ACCEPT ルールがあることを確認してください。

2. `eventlog[...]` キーで、ログ名（大文字と小文字を区別）、イベント ID、モード（例: skip）、およびその他のパラメータが、Windows 固有のアイテムキー に記載されているとおりに正確に指定されていることを確認してください。

関連項目:

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

Zabbix Cloud

Zabbix Cloud

Zabbix Cloud で利用可能な設定オプションと、組織およびノードのアクセス制御について説明します。

Zabbix Cloud のセットアップと管理

Zabbix Cloud の利用を開始する　初期設定や監視対象デバイスの接続を含め、Zabbix Cloud インスタンスをすばやく稼働させるためのガイド付き手順です。

ノード設定　アクセス、データ処理、リソース設定に重点を置いて、Zabbix Cloud ノードの設定方法を学びます。

ユーザーの追加　Zabbix Cloud 組織およびそのノード内で、ユーザーを追加し、ロールを割り当てる方法を確認してください。

Zabbix Cloud について学ぶ

Zabbix Cloud とオンプレミスの比較　2 つの導入形態における管理、スケーラビリティ、メンテナンスの違いを比較し、どちらがあなたのワークフローを最もシームレスにサポートできるかを確認してください。

Zabbix Cloud を確認する　主な機能、利点、サブスクリプションに含まれる内容など、Zabbix Cloud の概要を確認できます。セットアップや使用方法に関する一般的な質問については、FAQ を参照してください。

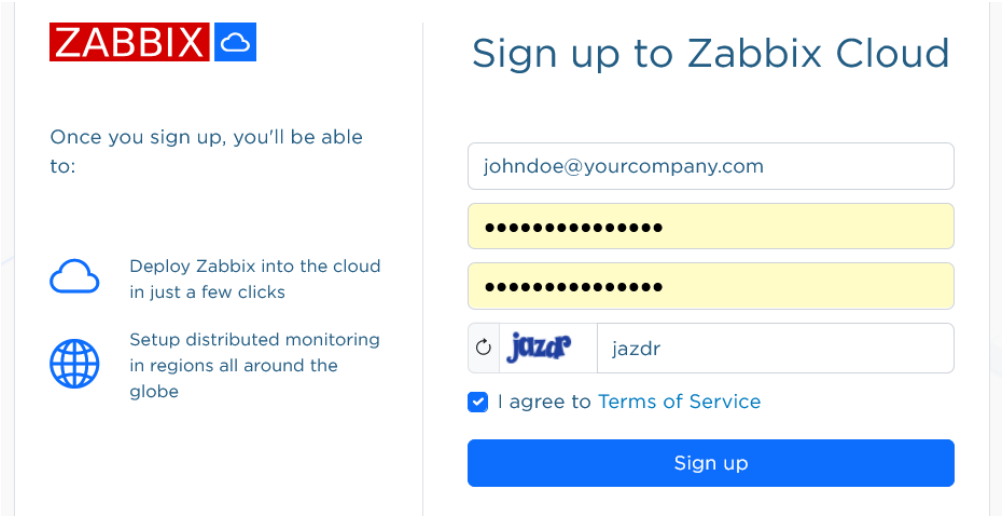
クラウドに **Zabbix** をデプロイする

Zabbix Cloud を開始するには、次の手順に従います。

- cloud.zabbix.com でログインまたはサインアップします
- クラウドノードを作成します
- Zabbix Cloud の利用を開始します

ログインまたはサインアップ　既存の Google、GitHub、または Microsoft アカウントを使用して、cloud.zabbix.com にログインできます。

または、sign up ページに移動して登録することもできます。



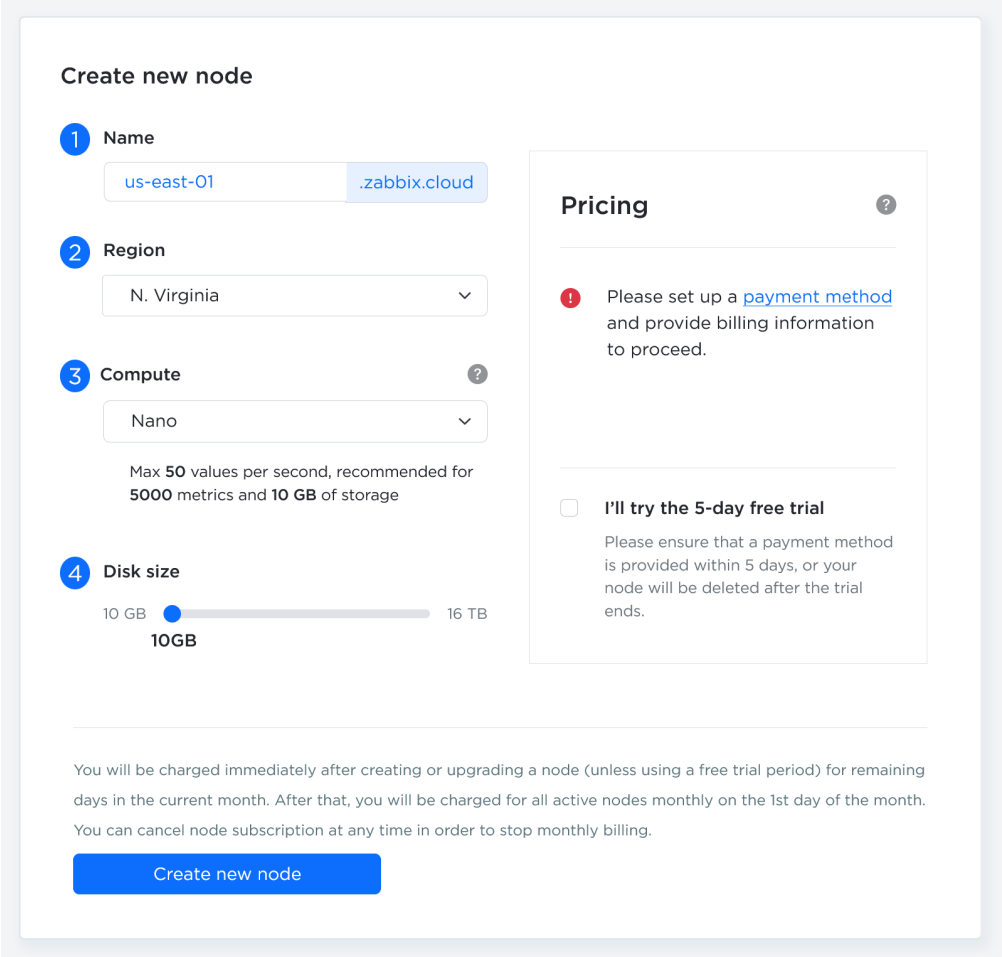
サインアップ後、新しいアカウントの確認メールが送信されます。

cloud.zabbix.com にアクセスしてログインします。ログイン時にはワンタイムパスワードの入力を求められます。このパスワードはメールで送信されます。

クラウドノードの作成　クラウドノードは、クラウド上に構築される Zabbix インスタンスです。これには、Zabbix サーバー、データベース、および Web インターフェースが含まれます。

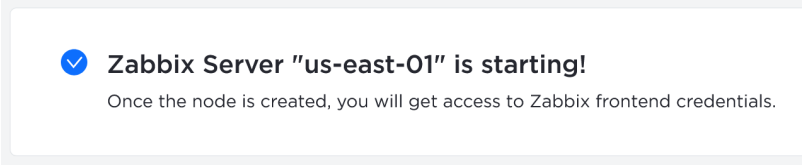
ログインすると、アカウントにはデフォルトで *Private* organization がすでに存在していることがわかります。デフォルトの organization は後で名前を変更できます。

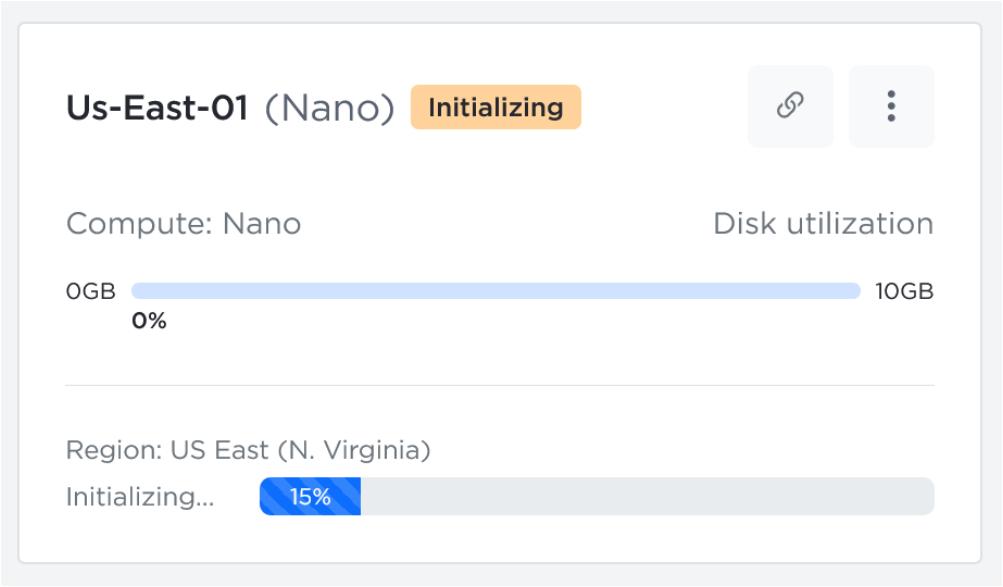
Create New Node をクリックします。ノード設定フォームが開きます。

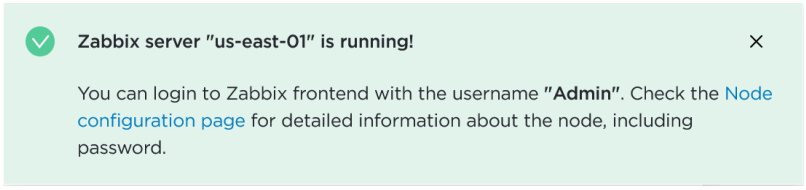


Parameter	Description
Name	一意のノード名を選択します。 ノード名は、`<your-name>.zabbix.cloud` の形式で、クラウドサーバー/Web インターフェース用の一意の URL を構成します。
Region	ネットワーク遅延を最小限に抑えるため、最も近いデータセンターの場所を選択します。
Compute	ノードの計算能力を選択します。 必要なリソースレベルを判断するには、pricing details を参照してください。より高い性能が必要になった場合は、後からいつでもアップグレードできます。試用版ではないノードを作成する前に、請求方法の設定が必要です。請求情報は *Billing* メニューから追加します。 *Nano* ティアのノードを試すために、5 日間の無料トライアルを利用できます。トライアル期間中は課金されませんが、終了後に有料ノードへ変換されていない場合、ノードは削除されます。
Disk size	スライダーの初期値は、ティアに基づく推奨最小値に設定されています。 大量の履歴データやトレンドデータを保存する場合は、値を増やしてください。ディスク容量が大きいほど月額コストも高くなる点に注意してください。また、ディスク容量を自分で縮小することはできず、減らすにはサポートへの連絡が必要です。

フォームへの入力が完了したら、*Create New Node* をクリックします。ノードの初期化中は対応するメッセージが表示され、ノードはすでに Organizations の下に表示されます。





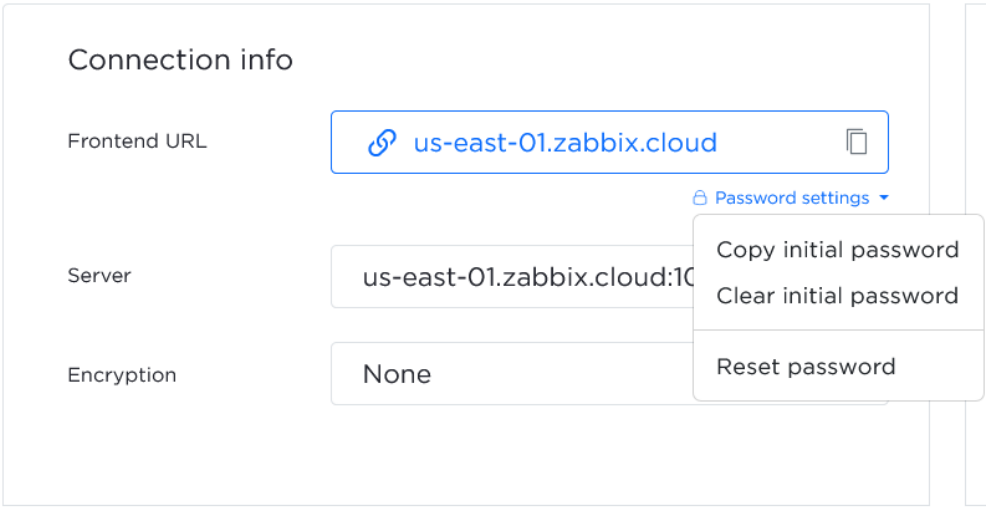


Zabbix の利用を開始する

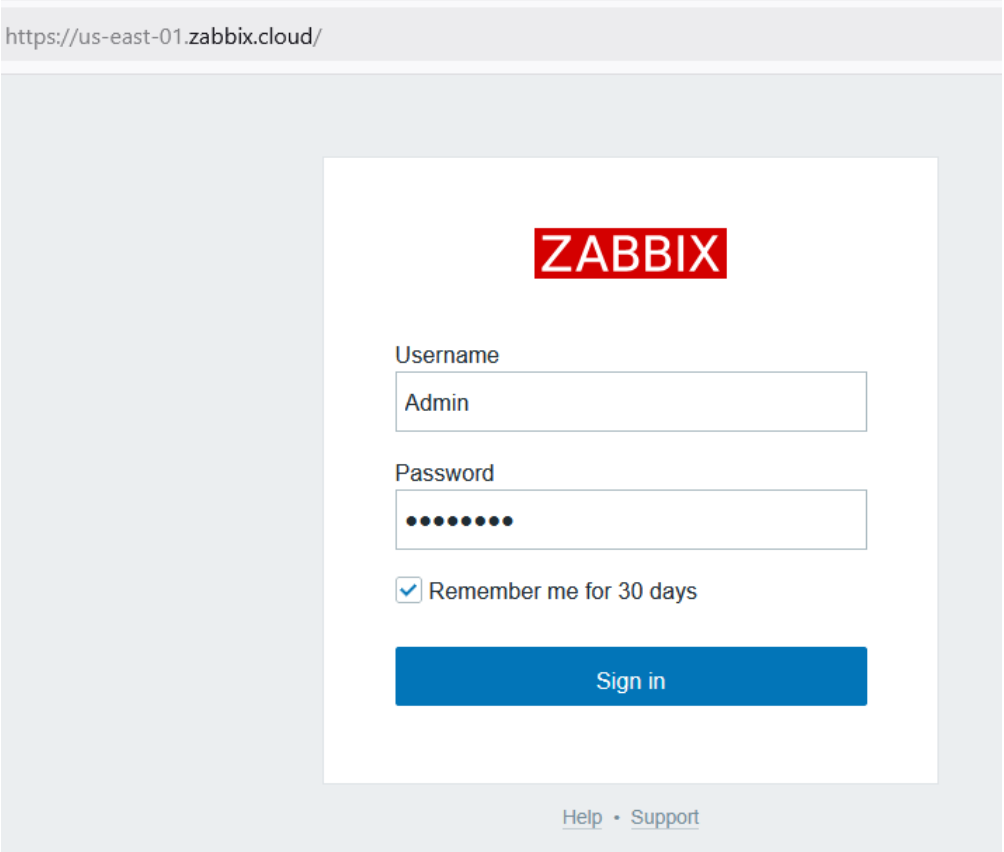
ノードのステータスが *Initializing* から *Running* に変わると、Zabbix の Web インターフェースにログインできます。

初期の Web インターフェースパスワードにアクセスするには、作成したノードをクリックします。

ノードの設定（概要、アクセスフィルター、暗号化、バックアップなど）が表示されます。**Connection info** で、*Password settings* をクリックし、ドロップダウンから *Copy initial password* を選択します。



Web インターフェースの URL をクリックして、Zabbix の Web インターフェース（`us-east-01.zabbix.cloud`）を開きます。



次の認証情報を使用してログインします。

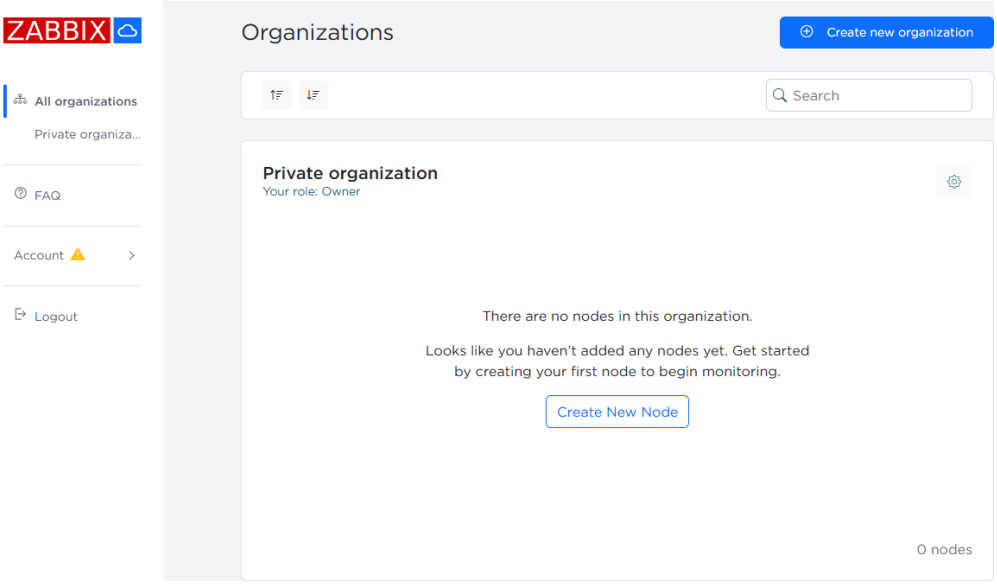
- *Username* - `Admin`
- *Password* - 初期パスワードを入力します（例: HzTG9t7Y）

監視を開始するには、アクセスフィルターを定義する必要があります。これは、Zabbix クラウドサーバーへの接続を許可する IP アドレスです。少なくとも 1 つ定義する必要があります。

組織管理　組織は、1 つ以上のノードを格納するコンテナです。組織は、次のようなリソースの構造化や責任の委任にも役立ちます。

- ノード管理
- 請求

Zabbix Cloud にログインすると、すべての組織ページが表示されます。デフォルトでは、アカウント用のプライベート組織が作成されます。



この組織の所有者として、ユーザー、管理者、ノード、および組織名を管理できます。管理オプションにアクセスするには、 設定ボタンをクリックします。

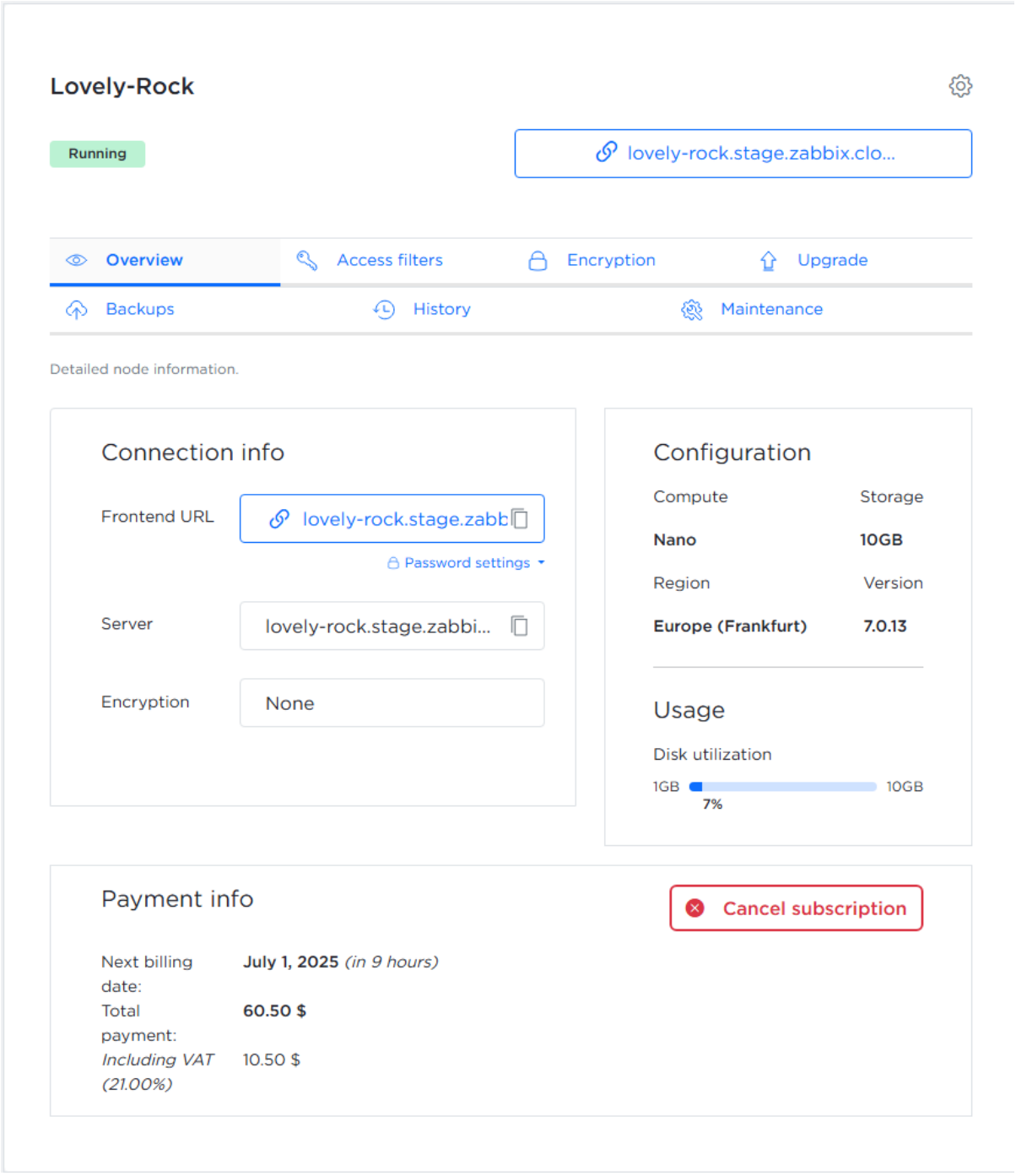
ノード設定

概要

ノード設定ページにアクセスするには、すべての組織でノードをクリックするか、組織メニュー からアクセスするか、ノードの起動後に表示される *Node configuration page* リンクを使用します。

ノード設定ビューでは、各タブでノードの特定の側面を管理できます。

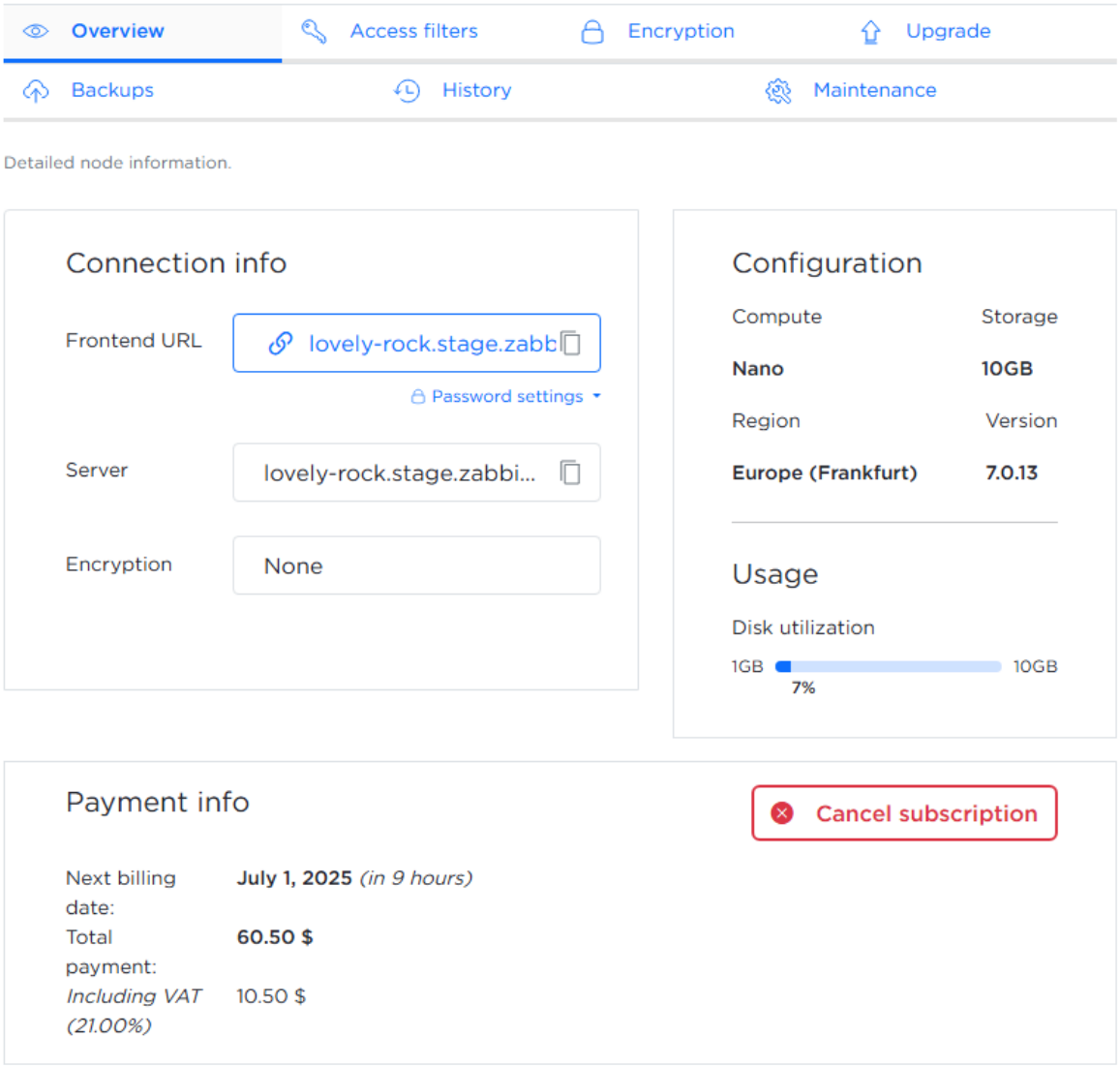
- Overview - ノードの詳細を表示し、その Web インターフェースを開き、ノードユーザーを管理します。
- Access filters - IP アドレスまたはサブネットによるアクセス制御を行います。
- Encryption - 証明書、鍵、および任意の CRL をアップロードして TLS を設定します。
- Backups - ノードスナップショットの作成、復元、削除を行い、次回予定されているバックアップを表示します。
- History - 監査ログ、履歴、トレンドの保持期間を設定します。あらかじめ用意された単位が使用でき、更新は 1 日 3 回までに制限されています。
- Maintenance - タイムゾーン、曜日、時間ごとに毎週のメンテナンス時間帯を定義します。
- Upgrade - コンピュート/ストレージリソースをスケールするか、トライアル/支払いステータスを管理します。



Note:
ページを表示しているユーザーの種類およびノードのステータス（トライアル/非トライアル）によっては、一部のタブを利用できない場合があります。

概要タブ

概要タブには、ノードに関する詳細情報が表示されます。



このタブは 3 つのセクションに分かれています。

接続情報:

- クリック可能な Web インターフェースの URL（クリップボードへのコピー機能付き）
- パスワード設定
- 現在の暗号化方式

設定と使用状況:

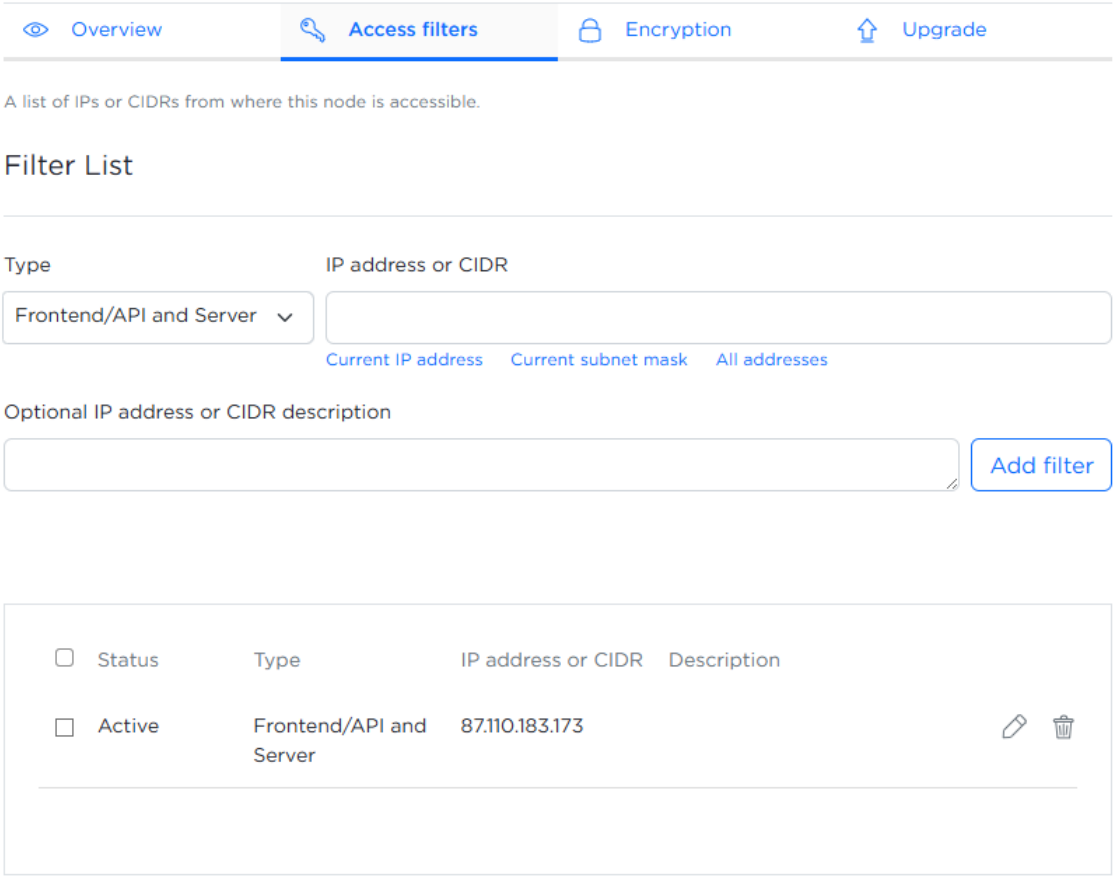
- コンピュートティア、最大ストレージ、リージョン
- Zabbix コンポーネントのバージョン
- ディスク使用率

ノード情報/支払い情報:

- 有効期限日（トライアルノードおよび前払いノードの場合）
- ノードの初期化完了後、ノードタイプに応じてコンテキスト固有のボタンが表示されます。- 有料版にアップグレード - 無料トライアル中のノードで利用可能 - サブスクリプションをキャンセル - 有効なサブスクリプションを持つ有料ノードで利用可能 - 有料版を再購読 - サブスクリプションがキャンセルされている場合に表示されます - 長期間有効でキャンセルできない前払いノードには、アクションボタンは表示されません

アクセスフィルタータブ

アクセスフィルタータブでは、IP アドレスまたはサブネットによって Zabbix ノードへのアクセスを制限できます。



アクセスフィルターを設定するには、次のフィールドを入力します。

- タイプ - ドロップダウンを使用して、制限する Zabbix コンポーネントを選択します: Web インターフェース/API のみ、サーバーのみ、またはその両方。
- *IP* アドレスまたは *CIDR* - IP アドレスまたは CIDR を手動で入力するか、該当するオプションを選択してフィールドを自動入力します。
- 任意の *IP* アドレスまたは *CIDR* の説明 - 各 IP またはサブネットに任意の説明を追加します。

フィールドの入力後、*Add filter* をクリックします。
追加したフィルターは下の表に表示され、ステータス、タイプ、IP アドレスまたはサブネット、説明が示されます。

Warning:
ノードの初期化中に、使用した IP アドレスに対するアクセスフィルターが自動的に作成されます。後で別の IP またはネットワークから接続する場合は、そのアドレスをここに追加する必要があります。追加しないとブロックされます。

次の操作が可能です。

-  - フィルターを編集します。
-  - フィルターを削除します。チェックボックスを使用して複数のフィルターを削除することもできます。

暗号化タブ

暗号化タブでは、必要な証明書ファイルをアップロードして、Zabbix コンポーネント間の TLS暗号化を設定できます。

証明書を使用して安全な通信を有効にするには、*CERTIFICATE* を選択し、コンピューターから次のファイルをアップロードします。

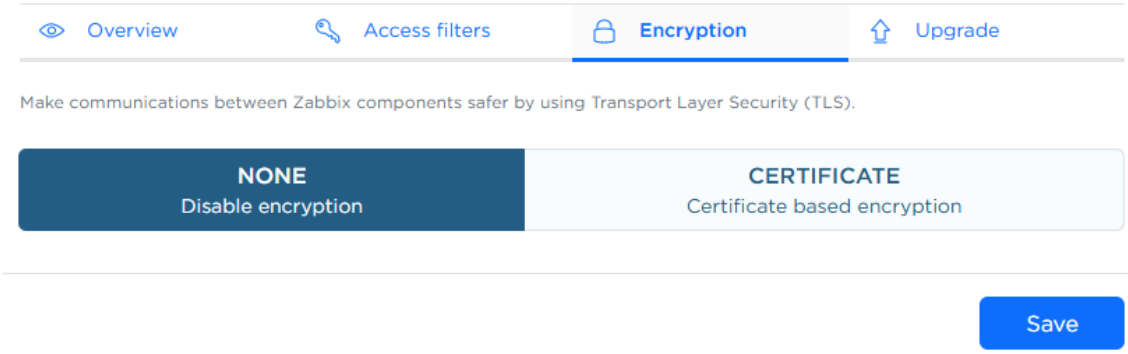
- ルート *CA* 証明書 - ピア証明書の真正性を検証するために使用されます。
  Choose file をクリックし、信頼されたルート CA 証明書を含むファイル（通常は PEM 形式でエンコードされた `.crt` または `.pem` ファイル）を選択します。

- 証明書チェーン - 自身の証明書に対する信頼の連鎖を表します。
  Choose file をクリックし、証明書と中間 CA 証明書を含む PEM 形式でエンコードされたファイル（例: `.crt` または `.pem`）を選択します。

- 秘密鍵 - 証明書に対応する秘密鍵です。
  Choose file をクリックし、PEM 形式でエンコードされた.key ファイルを選択します。

- 失効した証明書（任意）- 証明書失効リスト（CRL）を管理している場合は、*Choose file* をクリックして CRL ファイルを選択し、信頼しない証明書を指定します。

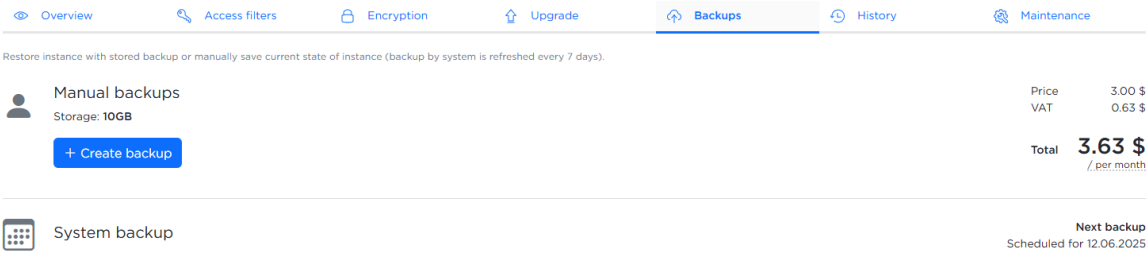
ファイルを選択したら、*Save* をクリックします。ファイルがアップロード、検証、適用され、Zabbix コンポーネント間の TLS 暗号化が設定されます。



バックアップタブ

バックアップタブでは、Zabbix ノードのバックアップの表示、作成、復元、削除を行えます。

保存済みの任意のバックアップからインスタンスを復元したり、いつでも新しい手動バックアップを作成したりできます。
このタブには、次回予定されているバックアップの日付も表示されます。



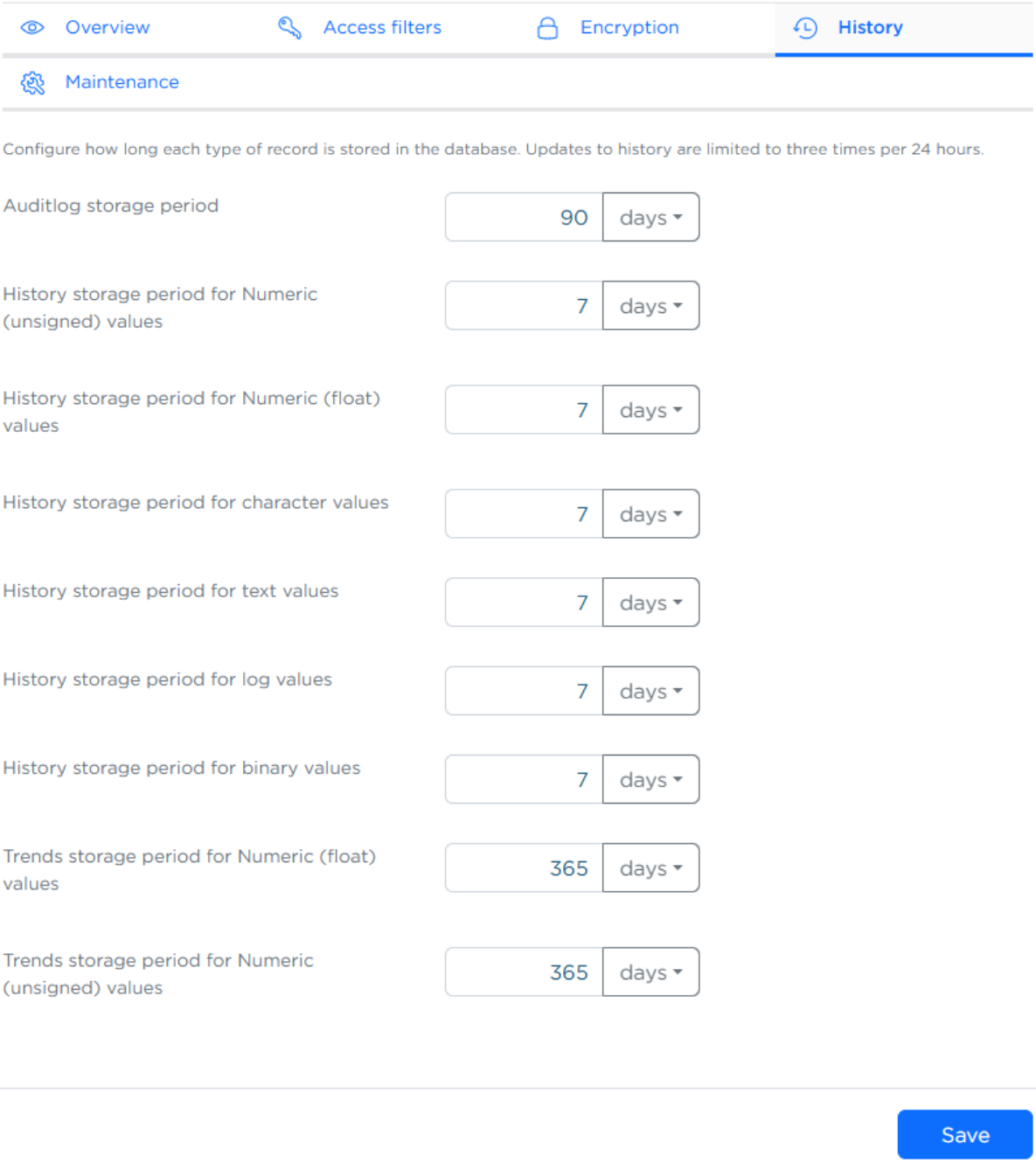
バックアップを作成するには:

1. *Create backup* ボタンをクリックします。
2. ポップアップウィンドウで任意の説明を入力します。
3. *Create* をクリックして確定します。

Warning:
追加のバックアップには、ストレージ使用量に基づいて月額料金が発生します。

履歴タブ

履歴タブでは、さまざまな種類の履歴データをどのくらいの期間保存するかを設定できます。



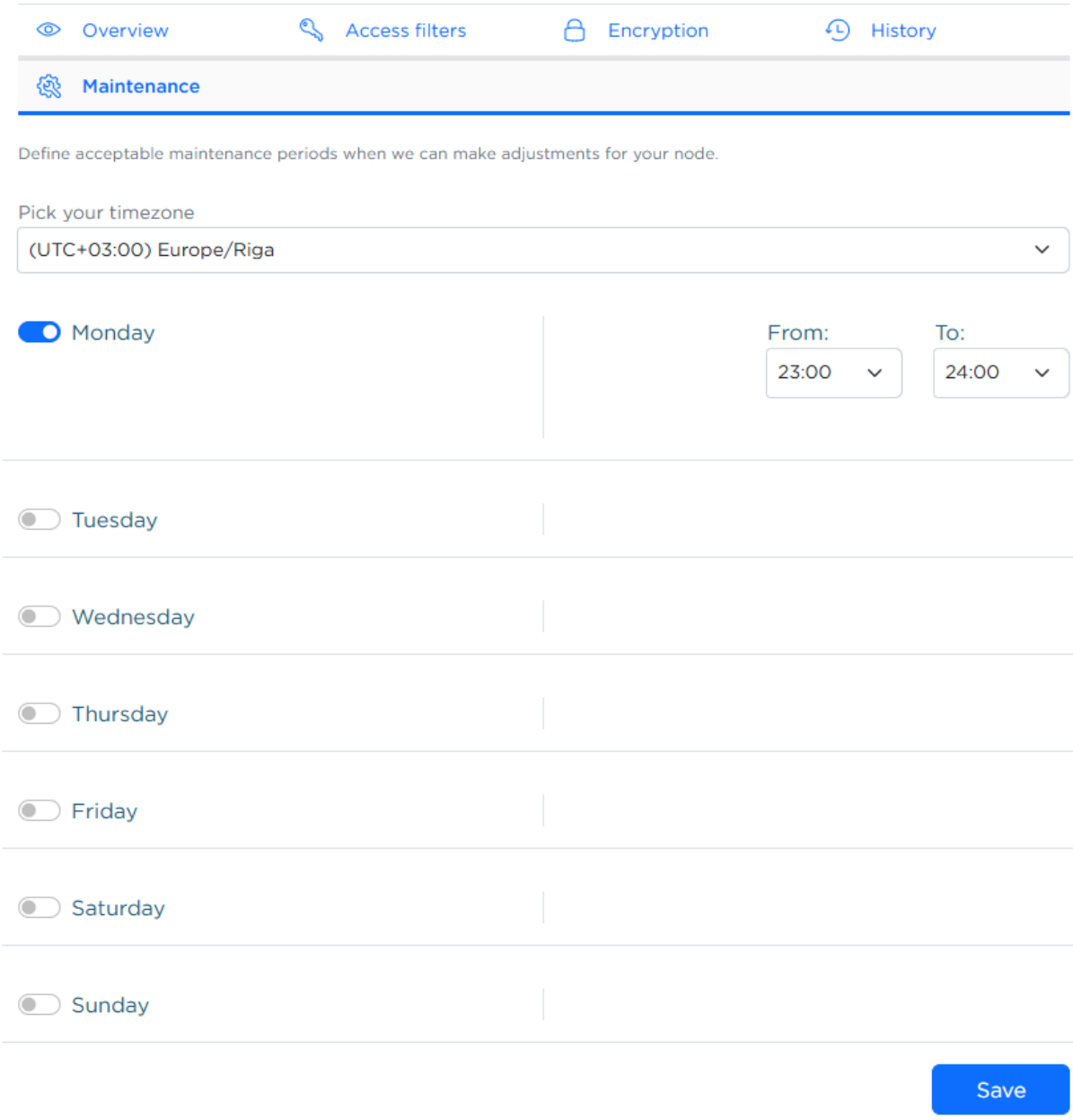
各データタイプ（例：数値、テキスト、ログ）ごとに保存期間を設定し、単位（日、月、年）を選択できます。
変更は 24 時間ごとに最大 3 回まで行えます。

値を更新したら、変更を適用するために保存をクリックします。

Warning:
保持期間を長くすると、ディスク使用量が増加し、パフォーマンスに影響する可能性があります。
実際のデータ量と分析要件に基づいて保持期間を設定することを推奨します。
 Zabbix Cloud では、従来の Zabbix housekeeping の代わりに、データ保存の管理にデータベースパーティショニングを使用しています。これにより、より高速なパフォーマンスと、より予測しやすい保持管理が可能になります。

メンテナンスタブ

メンテナンスタブでは、ノードで更新や調整を実行できる許容メンテナンス時間帯を定義できます。

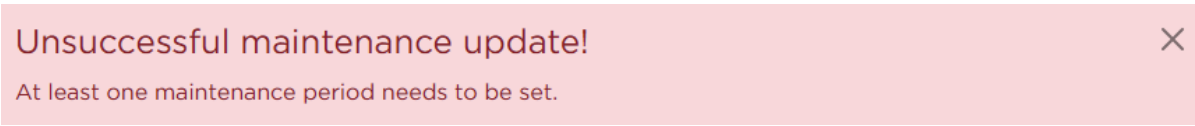


以下を行う必要があります。

- ドロップダウンからタイムゾーンを選択します。
- メンテナンス用に、少なくとも 1 つの曜日と 1 つの時間帯を選択します。

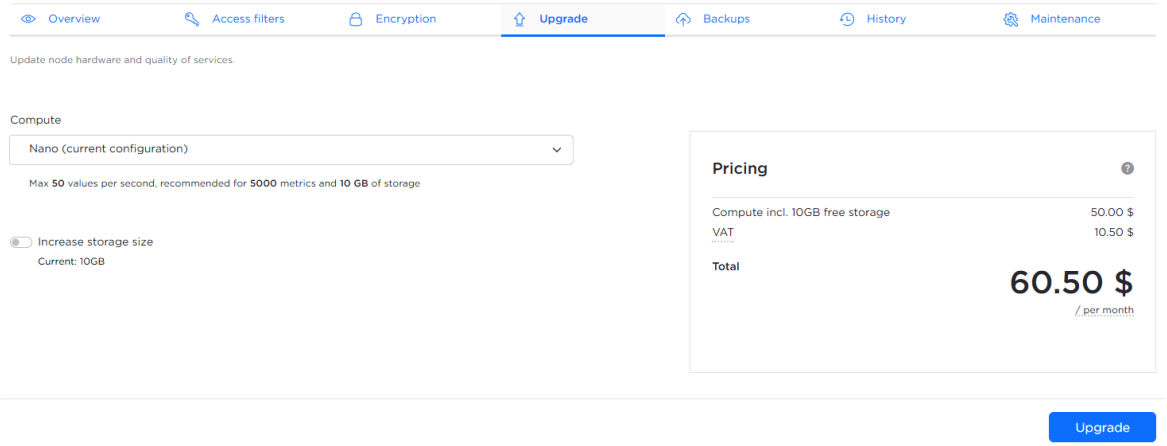
変更を適用するには、保存をクリックします。

曜日または時間帯が選択されていない場合は、エラーが表示されます。



アップグレードタブ

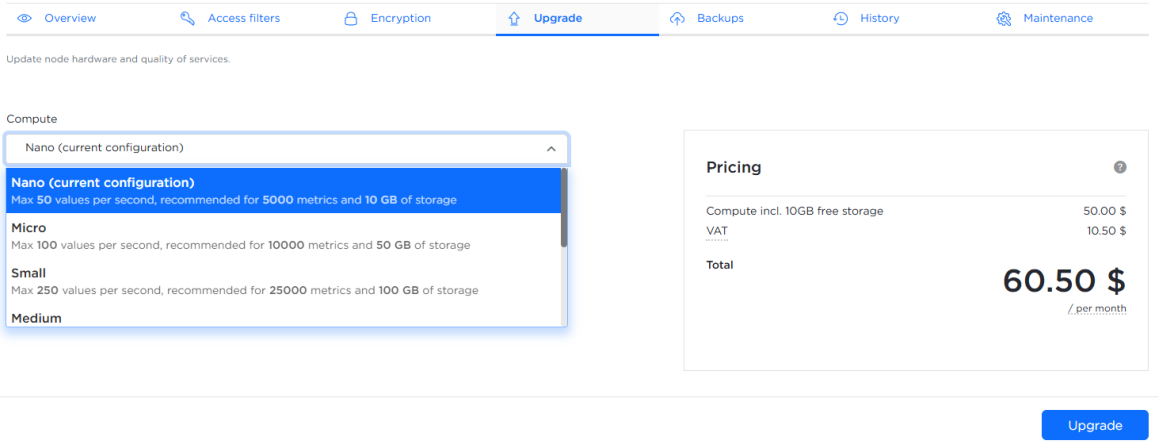
Upgrade タブでは、ノードのリソースをスケールアップしたり、サブスクリプションを管理したりできます。



コンピュートティアまたはディスクサイズを増やすことができます。なお、ディスクサイズは 24 時間に 1 回しか増やせず、UI から縮小することはできません（縮小を希望する場合は Zabbix サポートにお問い合わせください）。
無料トライアルのノードは有料サブスクリプションに変更できます。

コンピュートティアを増やすには:

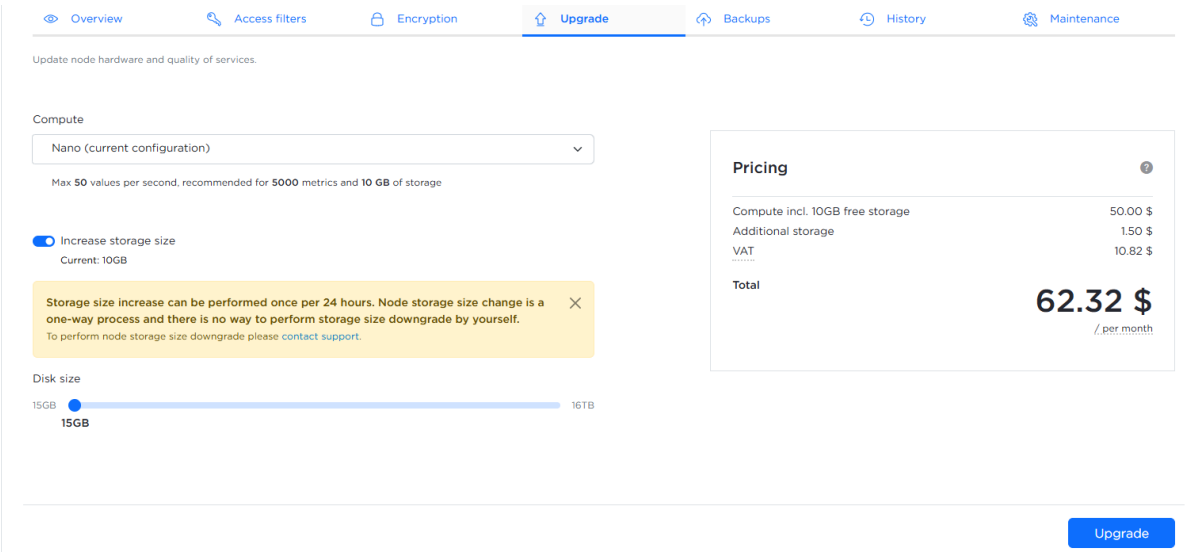
1. *Compute* ドロップダウンをクリックします。



2. 希望する構成ティアを選択します。
3. *Upgrade* ボタンをクリックします。

ディスクサイズを増やすには:

1. *Increase storage size* ラジオボタンをクリックします。
2. *Disk size* スライダーを希望する値に調整します。



3. *Upgrade* ボタンをクリックします。

Note:
変更を確定する前に、更新後の月額合計料金が *Pricing* テーブルに表示されます。

ユーザーの追加

はじめに

以下の例では、組織とノードにユーザーを追加する方法について説明します。

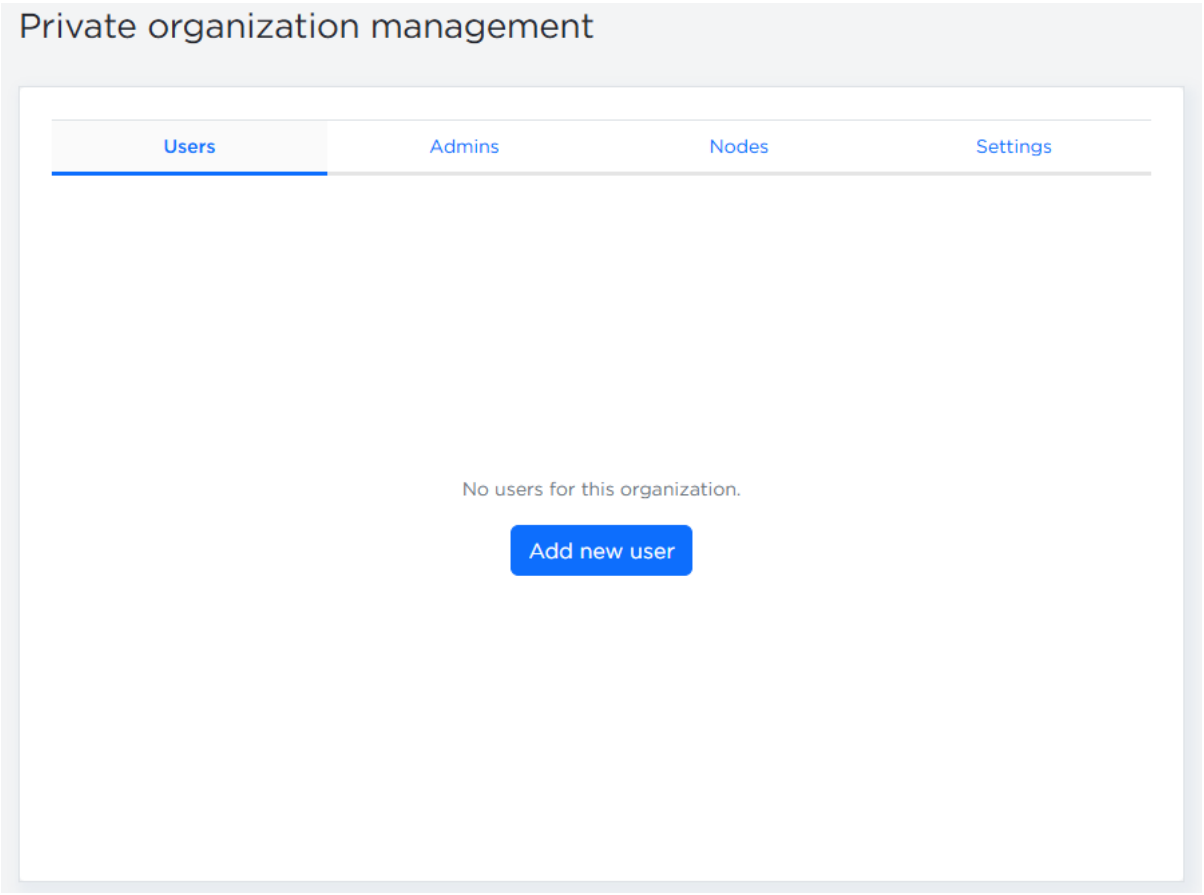
組織へのユーザーの追加

Zabbix のusers と同様に、Zabbix Cloud でもユーザーを作成し、組織およびそのノード内で実行できる操作を制御するためのロールを割り当てることができます。

組織にユーザーを追加するには、次の手順を実行します。

1. *All organizations* ページを開きます。

2. 対象の組織の  設定ボタンをクリックして、その管理メニューに入ります。



3. *Users* または *Admins* タブで、*Add new user* をクリックします。

4. ユーザーのメールアドレスを入力し、ロールとして *Administrator* または *User* を選択します。

User ロールを選択するとユーザーは組織に追加されますが、どのノードにもアクセス権は付与されません。

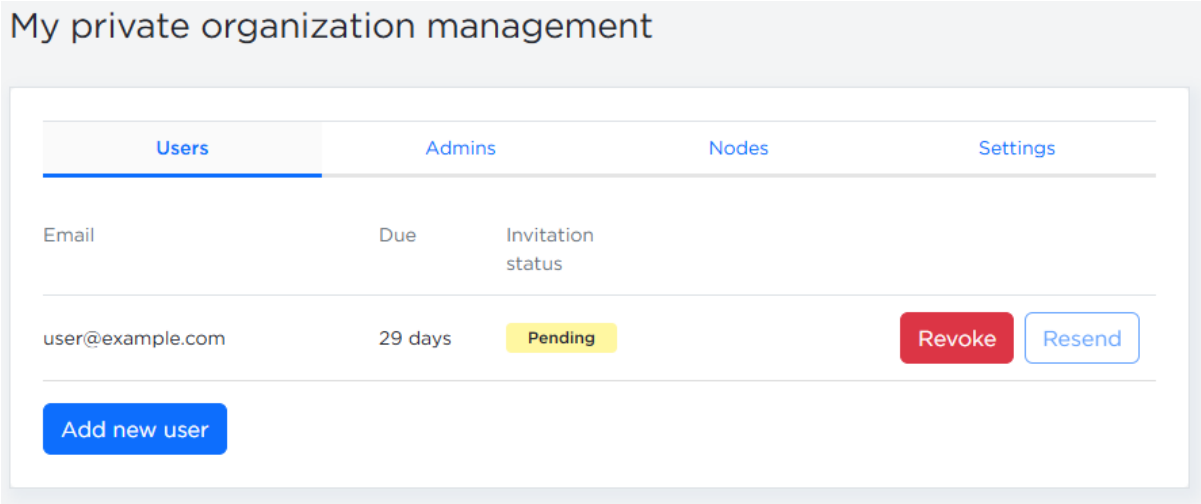
各ロールには、異なるレベルのアクセス権が付与されます。

- **User** - 割り当てられたノードにアクセスでき、課金が発生しない設定タスクを実行できます。たとえば、次のような操作です。
  - access filters の管理
  - encryption の設定
  - history/trend の保持期間の設定
  - maintenance のスケジュール設定

- **Administrator** - User のすべての権限に加えて、課金が発生する可能性のある操作も実行できます。たとえば、次のような操作です。
  - backups の作成と復元
  - ノードリソース（コンピュートティアおよびディスクサイズ）のアップグレードまたはダウングレード
  - サブスクリプションのキャンセル
- **Owner** - 組織に対する完全な制御権を持ちます。Administrator のすべての権限に加えて、Owner は次のこともできます。
  - ノードの作成および削除
  - User および Administrator の追加または削除
  - 組織名の変更または組織の削除
  - 請求情報の管理

> **Attention:**
> すべてのサービス料金は Owner の支払い方法に請求されます。銀行確認が必要な場合、Administrator が開始した取引であっても、Owner がその取引を承認する必要があります。

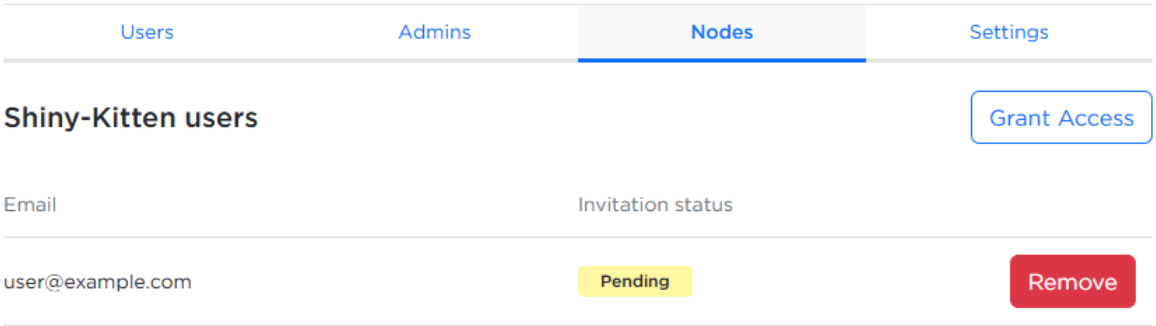
招待されたユーザーは *Pending* ステータスで追加されます。
招待は取り消すことも再送することもできます（再送は 10 分に 1 回のみ可能で、この制限に達するとボタンは無効になります）。



ノードへのユーザーの追加

ユーザーをノードに追加するには、以下の手順に従います。

1. ノードタブに移動します。
2. アクセスを許可をクリックします。
3. ドロップダウンから組織に追加するユーザーを選択します。
4. アクセスを許可をクリックします。



または、ノード構成ウィンドウからアクセスを許可することもできます。

1.  設定ボタンをクリックします。
2. ユーザーを追加を選択します。
3. ユーザーを選択し、アクセスを許可をクリックします。

招待されたユーザーは、招待を送信したのと同じメールアドレスを使用して Zabbix Cloud アカウントを作成し、招待を承諾する必要があります。ユーザーがそのメールアドレスで既に Zabbix Cloud アカウントを持っている場合は、サインインして招待を確認する必要があります。招待は特定のメールアドレスに関連付けられており、別のアカウントから承諾することはできません。

Zabbix Cloud の主な違い

比較表　オンプレミスの Zabbix インストールとクラウドノードの機能の大部分は重複していますが、違いもあります。以下に概要の比較表を示します。続くセクションでは、選択したアイテムについて詳しく説明します。

機能 / 項目	Zabbix Cloud	オンプレミスの Zabbix
リリースサイクル	LTS リリースに合わせて提供	Zabbix のバージョン（安定版、ベータ版、カスタムビルド）を選択可能
使いやすさ	すぐに使えるプラットフォームで、Zabbix により完全管理	インストール、設定、保守が必要
脆弱性パッチ適用	OS/Zabbix のパッチ適用は自動	ユーザーによる手動パッチ適用
HTTPS 証明書	最初から有効な証明書を利用可能（Let's Encrypt 不要）	デフォルトでは自己署名証明書（Let's Encrypt または他の CA を設定可能）
DB パフォーマンス (INSERT/UPDATE/SELECT)	最大速度のための自動パーティショニング、長時間実行される DELETE なし	手動パーティショニングとクリーンアップ（長時間の DELETE トランザクションが発生する可能性あり）
パフォーマンスチューニング	Zabbix Cloud チームにより事前調整済み（変更履歴なし）	チューニングパラメータの可視性と制御を完全に確保
ファイアウォール制御	Zabbix Cloud チームが管理、GUI/trapper ポートは公開されない	ファイアウォールルールと公開ポートを完全に制御可能
DB 分離と保護	高度に分離され、DB/SSH への直接アクセスなし	ネットワークおよびホストのセキュリティに依存
高可用性 *(HA)*	高可用性のクラウドプラットフォーム上で稼働し、 データ収集の耐障害性向上のために Zabbix プロキシを使用可能	ユーザー管理の HA
SNMP トラップ	専用プロキシ経由のみ（HA/負荷分散されたトラップレシーバーなし）	直接またはプロキシ経由、HA/負荷分散が可能
SNMP ポーリング	プロキシなしでは困難（各デバイスに NAT/カスタムポートが必要）	ネイティブのSNMP ポーリング、プロキシは任意
保持設定	履歴/トレンド/監査は UI 経由のみ（API または設定ファイルは不可）	Zabbix のserver configuration fileまたはAPIで設定可能
カスタムスクリプト	`AlertScriptsPath`、`ExternalScripts`、Web インターフェースおよびコミュニティモジュールは非対応	完全対応（scriptsパス、modules、連携機能）
ODBC 監視	PostgreSQL のみ（ドライバー `{postgresql}`）；MariaDB プラグインはプレースホルダーとして存在	PostgreSQL、MySQL、Oracle などの ODBC に完全対応、設定可能
ODBC 呼び出しの制限	不可（`StartODBCPollers=1`のみ；重い同期クエリが他の処理をブロック）	ポーラー数とスケジューリングを完全に調整可能
SAML 証明書のアップロード	非対応	UI または API で対応
定期レポート	独自の Email メディアを作成する必要あり（スクリプトメディアは非対応）	スクリプトメディアと Email メディアの両方を最初からサポート
アクティブチェック用ホストインターフェース	プラットフォーム発行の IP アドレス（自動作成）	インターフェースはユーザーが管理し、IP はユーザーが制御

機能の違い　SNMP トラップ

SNMP トラップ は、専用の Zabbix プロキシ経由でのみサポートされています。
SNMP トラップの監視が必要な場合、SNMP トラップは単一の IP アドレスに送信する必要があるため、プロキシの自動負荷分散または高可用性は使用できません。

SNMP ポーリング

プロキシを使用しない場合、SNMP ポーリングでは各デバイスを NAT およびカスタムポート経由で公開する必要があります。

保持設定

クラウドでは、履歴、トレンド、監査ログ の保持期間は Web インターフェースからのみ設定できます。
これらの設定は、`zabbix_server.conf` や API では設定できません。

アイテムごとの履歴保持期間の手動上書きはサポートされていません（パーティショニングはグローバルに制御されます）。クラウドノードの URL では、オンプレミス環境のインストールと同じクエリパラメータは使用できません。

カスタムスクリプト

以下のカスタムスクリプトタイプは、Zabbix Cloud ではサポートされていません。

- アラートスクリプト (`AlertScriptsPath`)
- 外部スクリプト (`ExternalScripts`)
- Zabbix サーバー上の Web インターフェーススクリプト

コミュニティによって開発された Web インターフェースモジュールはインストールできません。

ODBC 監視

Zabbix Cloud は、PostgreSQL に対してのみODBC 監視 をサポートしています。公式の Zabbix ODBC テンプレート を使用し、テンプレート内で以下を使用して接続文字列を定義してください。

```
Driver={postgresql}
```

MariaDB プラグインもインストールされていますが、現在は機能しません。使用する場合は、以下を定義してください。

```
Driver={mysql}
```

Attention:
Zabbix Cloud において、Oracle MySQL 8.0 で動作する既知の設定はありません。単純なクエリ（SELECT 1 など）は成功する場合がありますが、より複雑なクエリは SQL_ERROR を返します。

 ODBC 呼び出しを制限することはできません（StartODBCPollers=1 のみ）。負荷の高い同期レポート処理はパフォーマンスに影響する可能性があり、一度に実行できる SELECT は 1 つだけです。

インフラストラクチャへのアクセス

Zabbix Cloud では、基盤となるノードへの SSH アクセスは提供されず、データベースへの直接接続（たとえばポート 3306 への接続）も許可されていません。
設定、監視、トラブルシューティングに関するすべての作業は、クラウド UI または API を介して実施する必要があり、これにより OS およびデータベース層は分離され、安全に保たれます。

アクティブチェックインターフェース

クラウド内のアクティブチェックでは、オンプレミスネットワークとは無関係な IP を持つホストインターフェースが自動的に作成されます。
デフォルトでは、この IP は外部 IP である場合があります。
一貫性を維持するため、作成後にホストインターフェースを手動で削除または調整できます。

SAML 証明書のアップロード

暗号化されていない、または署名されていない SAML メッセージを受け入れるプロバイダーがほとんどないため、SAML 認証はサポートされていません。

スケジュールされたレポート

デフォルトの *Cloud Email* メディアタイプはスクリプト転送を使用しているため、スケジュールされたレポート には使用できません。
レポートをメールで送信するには:

1. 新しい Email メディアタイプを作成します。
2. ユーザー設定 > メディアで、そのメールメディアをユーザーに割り当てます。
3. スケジュールされたレポートを設定する際に、そのメディアタイプを選択します。

開発者センター

開発者センター

モジュール、ウィジェット、プラグインに関するガイダンスを活用して、Zabbix 向けのカスタム拡張機能の開発を始めましょう。

Zabbix 上で構築する

Web インターフェースモジュール　特定のユースケースに合わせて Zabbix の Web インターフェースを拡張または変更するカスタムモジュールを構築するためのガイドとリファレンスです。

ウィジェット　ウィジェットの構造とロジックの内訳、およびニーズに合わせてカスタムダッシュボード要素を作成するための手順を説明します。

プラグイン　機能を拡張したり外部システムと統合したりするための、Zabbix プラグインの開発および管理方法の概要です。

モジュール

PHP フロントエンドモジュールとは

- モジュールは、マニフェストファイルで定義された一意の ID、名前、説明、作成者、およびその他のフィールドを持つエンティティであり、PHP、Javascript、およびその他のファイルは、Zabbix フロントエンドインストールの *modules* ディレクトリ内の 1 つのディレクトリ (例: *zabbix/ui/modules*) に配置されます。
- モジュールは、正しい操作を保証するために、単純なルールに準拠する必要があります。
- モジュールは、管理者によってフロントエンドにインストール (解凍) され、有効にされる必要があります。

モジュールの用途

- カスタムフロントエンドセクションを介して新しい機能を追加する。
- カスタムダッシュボードウィジェットタイプを作成する (ウィジェットモジュールを参照)。
- 既存の機能をオーバーライドまたは拡張する。

モジュールが使用できない用途

- 新しい API メソッドの登録、または既存のメソッドの変更。

モジュールの動作

- アクションコードを実行する前に、各 HTTP リクエストで有効なモジュールが起動されます。
- モジュールは新しいアクションを登録するか、既存のアクションを再定義します。
- モジュールは新しいフロントエンドセクションを追加し、既存のセクションを削除または再定義します。
- モジュールは必要に応じて、onBeforeAction や onTerminate などのフロントエンドイベントにフックします。
- 要求されたアクションは、最終的にアクションコード (デフォルトのアクションまたはモジュール定義のアクションコード) を実行することによって実行されます。

次に進むべき道　実際にやってみるか、最初にガイドラインを読むかに関わらず、これらのページには独自のモジュールを作成するために必要な情報と手順が記載されています。

- 最初のモジュールを作成するためのステップバイステップのチュートリアル
- モジュールファイル構造
- ウィジェットモジュールの詳細
- 再利用するモジュールの例

モジュールファイルの構造

モジュールに関連するすべてのコードは、Zabbix フロントエンドインストールの **modules** ディレクトリ内の 1 つのディレクトリに保存されます (例: *zabbix/ui/modules*)。

```
example_module_directory/          (必須)
   manifest.json                   (必須)   メタデータとアクションの定義。
   Module.php                               モジュールの初期化とイベント処理。
   actions/                                 アクションコントローラーファイル。
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
```

```
        data_export/
            ExportAsXml.php
            ExportAsExcel.php
    views/                                  ビューファイル。
        example.something.view.php
        example.something.delete.php
    assets/                                 ビューで使用する追加ファイル。manifest.json で指定する必要があります。
        js/                                 ビューで使用する JavaScript ファイル。
            example.something.view.js.php
        css/                                ビューで使用する CSS ファイル。
            example.something.css
         image.png                          ビューで使用する画像。
         example.something.file             ビューで使用する任意のファイル。
```

モジュールファイルツリー

モジュールの作成　サンプルモジュールの作成の流れは次の通りです (利用可能な場合は、ファイル名またはフォルダー名をクリックすると、手順の詳細が表示されます)。

1. **zabbix/ui/modules/**内にモジュール用の新しいディレクトリを作成します。
2. モジュールメタデータを含むmanifest.jsonファイルを追加します。
3. *views*フォルダーを作成し、モジュールビューを定義します。
4. *actions*フォルダーを作成し、モジュールアクションを定義します。
5. Module.php (またはダッシュボードウィジェットの場合は Widget.php) ファイルを作成し、初期化およびイベント処理ルールを定義します。
6. JavaScript ファイル (*assets/js* に配置)、CSS スタイル (*assets/css* に配置)、またはその他の追加ファイル用の*assets*フォルダーを作成します。
7. manifest.json で必要なビュー、アクション、アセットファイルを指定していることを確認します。
8. Zabbix フロントエンドでモジュールを登録して、使用を開始します。

Note:
manifest.json ファイルを作成するとすぐに、モジュールを登録して有効にすることができます。モジュールが有効になると、Zabbix フロントエンドを更新することで、モジュールファイルに加えられたすべての変更をすぐにプレビューできます。

manifest.json

すべてのモジュールには、manifest.json ファイルが必要です。ファイルはモジュールのプライマリディレクトリに存在する必要があります (例: *zabbix/ui/modules/module_name/manifest.json*)。

最低限として、manifest.json では次のフィールドを指定する必要があります。

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

manifest.json でサポートされているパラメーター (パラメーター名をクリックすると詳細な説明が表示されます):

パラメーター	説明	必須
manifest_version	モジュールのマニフェストバージョン。	Yes
id	一意のモジュール ID。	
name	管理セクションに表示されるモジュール名。	
namespace	モジュールクラスの PHP 名前空間。	
version	モジュールのバージョン。	
type	モジュールのタイプ。ウィジェットの場合は *widget* と設定する必要があります。	Yes (ウィジェットのみ)
widget	ウィジェットの設定。ウィジェットのみに使用されます。	
actions	モジュールに登録するアクション。	
assets	含める CSS スタイルと JavaScript ファイル。	No
author	モジュールの作成者。	

パラメーター	説明	必須
config	カスタムモジュールオプションのデフォルト値。	
description	モジュールの説明。	
url	モジュールの説明へのリンク。	

manifest_version

モジュールのマニフェストバージョン。現在サポートされているバージョンは **2.0** です。

タイプ: Double

例:

```
"manifest_version": 2.0
```

id

モジュール ID。一意である必要があります。将来的に名前の競合が発生しないように、モジュールにはプレフィックス (作成者名、会社名など) を使用することをお勧めします。たとえば、レッスンのサンプルのモジュールで、モジュール名が”My module” の場合、ID は”example_my_module” になります。

タイプ: String

例:

```
"id": "example_my_module"
```

name

管理セクションに表示されるモジュール名。

タイプ: String

例:

```
"name": "My module"
```

namespace

モジュールクラスの PHP 名前空間。

タイプ: String

例:

```
"namespace": "ClockWidget"
```

version

モジュールのバージョン。バージョンは管理セクションに表示されます。

タイプ: String

例:

```
"version": "1.0"
```

type

モジュールのタイプ。ウィジェットの場合必須で、”widget” と等しくなければなりません。

タイプ: String

デフォルト: ”module”

例:

```
"type": "widget"
```

actions

モジュールに登録するアクション。各アクションの *class* オブジェクトキーの定義は必須ですが、他のアクションキーはオプションです。

タイプ: Object

*type*が *module* の場合にサポートされるオブジェクトキー:

- **write.your.action.name** (object) - アクション名は小文字 [a-z] で記述し、単語をドットで区切ります。サポートするキー:
  - **class** (string; 必須) - アクションクラス名。
  - **layout** (string) - アクションレイアウト。サポートする値: *layout.json*、*layout.htmlpage* (デフォルト)、*null*。
  - **view** (string) - アクションビュー。

例:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

*type*が *widget* の場合にサポートされるオブジェクトキー:

- **widget.{id}.view** (object) - ウィジェットビューのファイルとクラス名。**{id}** をウィジェットのid値に置き換えます (例: *widget.example_clock.view*)。サポートするキー:
  - **class** (string; 必須) - デフォルトの CControllerDashboardWidgetView クラスを拡張するウィジェットビューモードのアクションクラス名。クラスソースファイルは *actions* ディレクトリに配置する必要があります。
  - **view** (string) - ウィジェットビュー。*views* ディレクトリに配置する必要があります。ビューファイルがデフォルトで期待される *widget.view.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。
- **widget.{id}.edit** (object) - ウィジェット設定ビューのファイル名。**{id}** をウィジェットのid値に置き換えます (例: *widget.example_clock.edit*)。サポートするキー:
  - **class** (string; 必須) - ウィジェット設定ビューモードのアクションクラス名。クラスソースファイルは *actions* ディレクトリに配置する必要があります。
  - **view** (string) - ウィジェット設定ビュー。*views* ディレクトリに配置する必要があります。ビューファイルがデフォルトで期待される *widget.edit.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。

例:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

assets

含める CSS スタイルと JavaScript ファイル。

タイプ: Object

サポートするオブジェクトキー:

- **css** (array) - 含める CSS ファイル。ファイルは *assets/css* に配置する必要があります。
- **js** (array) - 含める JavaScript ファイル。ファイルは *assets/js* に配置する必要があります。

例:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

author

モジュールの作成者。作成者は管理セクションに表示されます。

タイプ: String

例:

```
"author": "John Smith"
```

config

モジュールオプションのデフォルト値。オブジェクトにはカスタムキーが含まれる場合があります。指定した場合、これらの値はモジュールの登録中にデータベースに書き込まれます。後で追加された新しい変数は、最初の呼び出し時に書き込まれます。その後、変数値はデータベース内でのみ直接変更できます。

タイプ: Object

例:

```
"config": {
    "username": "Admin",
    "password": "",
    "auth_url": "https://example.com/auth"
}
```

description

モジュールの説明。

タイプ: String

例:

```
"description": "This is a clock widget."
```

widget

ウィジェットの設定。*type*が *widget* に設定されている場合に使用されます。

タイプ: Object

サポートするオブジェクトキー*:*

- **name** (string) - ウィジェットリストおよびデフォルトのヘッダーとして使用されます。空の場合、モジュールの”name” パラメーターが使用されます。
- **size** (object) - デフォルトのウィジェットの寸法。サポートするキー:
  - *width* (integer) - デフォルトのウィジェットの幅。
  - *height* (integer) - デフォルトのウィジェットの高さ。
- **form_class** (string) - ウィジェットフィールドフォームを持つクラス。*includes* ディレクトリに配置する必要があります。クラスがデフォルトで期待する *WidgetForm.php* の場合、このパラメーターは省略できます。別の名前を使用する場合は、ここで指定します。
- **js_class** (string) - デフォルトの CWidget クラスを拡張するウィジェットビューモードの JavaScript クラスの名前。クラスにはダッシュボードがロードされます。クラスのソースファイルは、assets/js ディレクトリに配置する必要があります。参照: assets。
- **use_time_selector** (boolean) - ウィジェットにダッシュボード時間セレクターが必要かどうかを決定します。サポートする値: *true*、*false* (デフォルト)。
- **refresh_rate** (integer) - ウィジェットのリフレッシュレート (秒単位) (デフォルト: 60)。

例:

```
"widget": {
    "name": "",
    "size": {
        "width": 12,
        "height": 5
    },
    "form_class": "WidgetForm",
    "js_class": "CWidget",
    "use_time_selector": false,
    "refresh_rate": 60
}
```

url

モジュールの説明へのリンク。ウィジェットの場合、ウィジェットの追加またはウィジェットの編集ウィンドウでヘルプアイコンをクリックすると、このリンクが開きます。**url** が指定されていない場合、ヘルプアイコン ? をクリックすると、一般的なダッシュボードウィジェットページが開きます。

タイプ: String

例:

```
"url": "http://example.com"
```

アクション

アクションは、モジュールの' ビジネスロジック' を担当します。アクションは通常、コントローラーとアクションビューで構成されます。

モジュールは次の操作を実行できます。

- Zabbix フロントエンドで既に定義されているアクションを呼び出す。
- デフォルトのアクションをカスタムアクションで上書きする。
- まったく新しいアクションを定義する。

デフォルトのアクション動作をカスタム動作で上書きするには、モジュール設定で同じ名前のアクションを定義します。アクションが呼び出されると、デフォルトの Zabbix アクションの代わりにモジュールアクションが実行されます。

アクションファイルは *actions* フォルダーに保存する必要があります。アクションはmanifest.jsonで指定する必要があります。

コントローラー

アクションコントローラーのワークフロー:

1. HTTP リクエストで渡されたすべてのパラメータが有効であることを確認します:
   - コントローラーの *checkInput()* メソッドを呼び出します;
   - CNewValidator.php で定義されたバリデーションルールを使用します;
   - *validateInput()* メソッドを呼び出します。
2. ユーザー権限を確認します。
3. 渡されたパラメータに従ってデータを準備します: *checkInput()* が true を返した場合、Zabbix はコントローラーの *doAction()* メソッドを呼び出します。
4. ビュー用に **$data** 配列を準備します。レスポンスを **$data** 配列に格納するには、*CControllerResponseData* と *setResponse()* メソッドを使用します。

例:

```
/**
 * 入力パラメータを検証します。
 *
 * @return bool
 */
protected function checkInput(): bool {
    $ret = $this->validateInput([
        'status' => 'in '.implode(',', [HOST_STATUS_MONITORED, HOST_STATUS_NOT_MONITORED])
    ]);

    if (!$ret) {
        $this->setResponse(new CControllerResponseFatal());
    }

    return $ret;
}

/**
 * ユーザー権限を確認します。
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
```

```
}

/**
 * アクションを実行し、レスポンスオブジェクトを生成します。
 */
protected function do Action(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
利用可能なコントローラークラスの完全な一覧は、Zabbix のソースコードで確認できます。

ビュー

ビューファイルは、コントローラーからデータを受け取り、その HTML の外観を準備します。

Note:
モジュールがウィジェットでない限り、フロントエンドモジュールのビューの定義はオプションです。
ダッシュボードウィジェットには少なくとも 2 つのビューが必要です。1 つは編集モード用、もう 1 つは表示モード用です (*views* ディレクトリに保存する必要があります)。

ビューでは、定義済みの Zabbix HTML クラス (*/zabbix/ui/include/classes/html* から) を使用したり、新しい HTML クラスや CSS クラスを追加したりできます。新しいクラスは、モジュールの *assets* フォルダに保存する必要があります。

例:

```
...
(new CColHeader(_('Name')))
```

これにより、新しい列名 *Name* が追加され、他の Zabbix ページと同様にテーブルの上部の行のスタイルが設定されます。

アクションビュー

アクションビューを定義するためのリファレンスファイルです。

```
<?php declare(strict_types = 1);

/**
* @var CView $this
*/

$this->includeJsFile('example.something.view.js.php');

(new CWidget())
    ->setTitle(_('Something view'))
    ->addItem(new CDiv($data['name']))
    ->addItem(new CPartial('module.example.something.reusable', [
        'contacts' => $data['contacts']
    ])
    ->show();
```

アセット

assets フォルダーには、他のディレクトリに属さないファイルやサブフォルダーを含めることができます。次の目的で使用できます:

- JavaScript スタイル (*assets/js*内にある必要があります)
- CSS スタイル (*assets/css*内にある必要があります)
- 画像
- フォント
- その他、含める必要のあるもの

assets/js

assets/js ディレクトリは予約されており、JavaScript ファイルのみを含める必要があります。ウィジェットで使用するには、これらのファイルを*manifest.json*で指定します。

例:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css は予約されており、CSS スタイルファイルのみを含める必要があります。ウィジェットで使用するには、これらのファイルを*manifest.json*で指定します。

例:

```
"assets": {
    "css": ["mywidget.css"]
}
```

CSS スタイル

CSS ファイルには、特定のフロントエンドテーマの異なるスタイルを定義するためのカスタム属性 `theme` が含まれる場合があります。

利用可能なテーマとその属性値:

- ブルー - [theme='blue-theme']
- ダーク - [theme='dark-theme']
- ハイコントラストライト - [theme='hc-light']
- ハイコントラストダーク - [theme='hc-dark']

例:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

新しいモジュールの登録

このセクションでは、Zabbix フロントエンドに新しいモジュールを追加する方法について説明します。

前提条件　続行する前に、次のことを確認してください:

- モジュールは、Zabbix フロントエンドインストールの *modules* ディレクトリ内にあります (例: *zabbix/ui/modules*)。
- モジュールには、少なくともmanifest.jsonファイルの基本バージョンがあります。
- Zabbix の管理メニューセクションにアクセスできます (ユーザーの役割の super admin が必要です)。

> **Note:**
> フロントエンドは、互換性のないモジュールをインストールせず、認識もしません。

モジュールの追加　管理 → 一般 → モジュールページを開き、ディレクトリのスキャンを押します。



リストでモジュールを見つけてアクティブ化します。

モジュールをアクティブ化するには、無効ハイパーリンクを押します。モジュールの状態が有効に変わります。

モジュール名を押すと、作成者、バージョン、短い説明 (マニフェストで定義されている場合) など、モジュールに関する追加情報が表示されます。

ウィジェットのプレビュー　ウィジェットモジュールは、追加されるとすぐにダッシュボードウィジェットリストに表示されます。

ダッシュボードを開いて編集モードに切り替え、通常どおりダッシュボードにウィジェットを追加できます。

ウィジェットに変更を加えた場合は、ダッシュボードを更新して、最新の更新でウィジェットがどのように見えるかを確認します。

ウィジェット

ウィジェットは、ダッシュボードに使用される Zabbix フロントエンドモジュールです。特に明記されていない限り、すべてのモジュールガイドラインはウィジェットにも適用されます。

ただし、ウィジェットはモジュールとは大きく異なります。ウィジェットを作成するには、次の操作を行います。

- manifest.json ファイルでタイプ”widget” を指定します (”type”: ”widget”)。
- 少なくとも 2 つのビューを含めます。1 つはウィジェットプレゼンテーションモード用、もう 1 つはウィジェット構成モード用です (example.widget.view.php と example.widget.edit.php)。
- ウィジェットプレゼンテーション (WidgetView.php) 用のコントローラー。
- デフォルトのウィジェットクラスを使用して拡張します。

設定

このページでは、カスタム設定フィールドを使用してウィジェット設定ビューを作成するために使用できるクラスについて説明します。ウィジェット設定ビューは、ユーザーがプレゼンテーションのウィジェットパラメーターを設定できるウィジェットの一部です。

Widget

プライマリウィジェットクラス、すべてのダッシュボードウィジェットの基本クラス *CWidget* を拡張します。デフォルトのウィジェットの動作をオーバーライドするために必要です。

Widget クラスは、ウィジェットのルートディレクトリ (*zabbix/ui/modules/my_custom_widget* など) に配置する必要があります。

Widget.php example

```
<?php

namespace Modules\MyCustomWidget;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const MY_CONSTANT = 0;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

WidgetForm

WidgetForm クラスは、デフォルトクラス *CWidgetForm* を拡張し、データベース内でウィジェット構成の保存構造を定義し、入力検証を処理するために必要な*CWidgetField*フィールドのセットを含みます。

WidgetForm クラスは *includes* ディレクトリに配置する必要があります。クラスに別の名前が付いている場合は、その名前を *manifest.json* ファイルの*widget/form_class*パラメーターで指定する必要があります。

includes/WidgetForm.php example

```php
<?php

namespace Modules\MyCustomWidget\Includes;

use Modules\MyCustomWidget\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox,
    CWidgetFieldColor
};


class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            );
    }
}
```

CWidgetFormView

CWidgetFormView クラスは、*WidgetForm* クラスで定義されたフィールドのプレゼンテーションロジックを指定し、設定ビューで表示されるときの外観と動作を決定するために必要です。

CWidgetFormView クラスは次のメソッドをサポートします。

- *addField()* - *CWidgetFieldView* クラスのインスタンスをパラメーターとして受け取ります。各*CWidgetField*クラスには、ウィジェット設定ビューで使用するためのそれぞれの *CWidgetFieldView* クラスがあります。
- *addFieldset()* - フィールドを結合して折りたたみ可能なコンテナにする *CWidgetFieldsGroupView* クラスのインスタンスを受け取ります。
- *addFieldsGroup()* - フィールドを視覚的に (境界線を使用して) グループに結合する *CWidgetFormFieldsetCollapsibleView* のインスタンスを受け取ります。
- *includeJsFile()* - JavaScript ファイルをウィジェット設定ビューに追加できます。
- *addJavaScript()* - ウィジェット設定ビューがロードされるとすぐに実行されるインライン JavaScript を追加できます。

CWidgetFormView クラスは *views* ディレクトリに配置する必要があります。

views/widget.edit.php example

```php
<?php

/**
 * My custom widget form view.
 *
```

```
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

JavaScript クラスを使用して、ウィジェット設定ビューに動的な動作と対話性を追加できます。たとえば、*CWidgetFormView*クラスで定義されたカラーピッカーを初期化できます。

JavaScript クラスはフォームとともにロードする必要があるため、メソッド *includeJsFile()* および *addJavaScript()* を使用して*CWidgetFormView*クラスで参照する必要があります。

以下の例では、シングルトンクラスインスタンスがすぐに作成され、*window.my_custom_widget_form* という名前で保存されます。したがって、フォームを 2 回目に開くとインスタンスが再作成されます。

JavaScript クラスは *views* ディレクトリに配置する必要があります。

views/widget.edit.js.php example

```
<?php

use Modules\MyCustomWidget\Widget;

?>

window.my_custom_widget_form = new class {

    init({color_palette}) {
        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }
    }
};
```

CWidgetField

CWidgetField クラスは、すべてのフォームフィールドクラス (*CWidgetFieldCheckBox*、*CWidgetFieldTextArea*、*CWidgetFieldRadioButtonList* など) が継承される基本クラスです。*CWidgetField* を拡張するクラスは、ウィジェット設定値の受信、保存、検証を担

当します。

次の *CWidgetField* クラスが利用可能です。

CWidgetField クラス	データベースフィールドタイプ	説明
CWidgetFieldCheckBox	int32	Single checkbox.
CWidgetFieldCheckBoxList	array of int32	Multiple checkboxes under a single configuration field.
CWidgetFieldColor	string	Color selection field.
CWidgetFieldDatePicker	string	Date selection field.
CWidgetFieldHostPatternSelect	string	Multiselect field that allows to select one or multiple hosts. Supports defining host name patterns (all matching hosts will be selected).
CWidgetFieldIntegerBox	int32	Field to enter an integer. Can be used to configure minimum and maximum values.
CWidgetFieldLatLng	string	Text box that allows to enter comma-separated latitude, longitude, and map zoom level.
CWidgetFieldMultiSelectAction	ID	Multiselect field for selecting actions (from the list of actions defined in the *Alerts → Actions*).
CWidgetFieldMultiSelectGraph	ID	Multiselect field for selecting custom graphs.
CWidgetFieldMultiSelectGraphPrototype	ID	Multiselect field for selecting custom graph prototypes.
CWidgetFieldMultiSelectGroup	ID	Multiselect field for selecting host groups.
CWidgetFieldMultiSelectHost	ID	Multiselect field for selecting hosts.
CWidgetFieldMultiSelectItem	ID	Multiselect field for selecting items.
CWidgetFieldMultiSelectItemPattern	ID	Multiselect field for selecting item patterns.
CWidgetFieldMultiSelectItemPrototype	ID	Multiselect field for selecting item prototypes.
CWidgetFieldMultiSelectMap	ID	Multiselect field for selecting maps.
CWidgetFieldMultiSelectMediaType	ID	Multiselect field for selecting media types.
CWidgetFieldMultiSelectOverrideHost	ID	Multiselect field for selecting a data source (dashboard or other widget) containing a host for which the widget can display data.
CWidgetFieldMultiSelectService	ID	Multiselect field for selecting services.
CWidgetFieldMultiSelectSla	ID	Multiselect field for selecting SLAs.
CWidgetFieldMultiSelectUser	ID	Multiselect field for selecting users.
CWidgetFieldNumericBox	string	Field to enter a float number.
CWidgetFieldRadioButtonList	int32	Radio box group that consists of one or more radio boxes.
CWidgetFieldRangeControl	int32	Slider to select an integer type value.
CWidgetFieldReference	string	Creates a unique identifier for this widget on dashboard. It is used to reference this widget from other widgets.
CWidgetFieldSelect	int32	Dropdown select box.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* preset with trigger severities.
CWidgetFieldTags	array of (string, int32, string)	Allows to configure one or more tag filter rows.
CWidgetFieldTextArea	string	Text area for entering multi-line text.
CWidgetFieldTextBox	string	Text box for entering single-line text.
CWidgetFieldTimePeriod	array of string	Time period selecting field.
CWidgetFieldTimeZone	string	Dropdown with timezones.
CWidgetFieldThresholds	array of (string, string)	Allows configuring color and number pairs.
CWidgetFieldUrl	string	Text box that allows to enter URLs.

次の *CWidgetField* クラスが特定のウィジェット用に作成されています。これらのクラスには非常に特殊な使用例がありますが、必要に応じて再利用することもできます。

CWidgetField クラス	データベースフィールドタイプ	説明
CWidgetFieldColumnsList	array of (multiple mixed)	For *Top hosts* widget. Create a table with custom columns of allowed types.
CWidgetFieldNavTree	string	For *Map navigation tree* widget. Replaces widget view in edit mode with the map selection tree.

プレゼンテーション

このページでは、ウィジェットプレゼンテーションビューの作成に使用できるコンポーネントについて説明します。ウィジェットプレゼンテーションビューは、ウィジェットの設定に従ってデータを受信し、コンテナ内のダッシュボードに表示するウィジェットの一部です。

プレゼンテーションビューは、次の 3 つの部分で設定されます。

- ウィジェットアクション
- ウィジェットビュー
- JavaScript

Widget action

ウィジェットアクションクラス (*WidgetView*) には、プレゼンテーションビューモードでウィジェットを操作するためのメソッドが含まれています。ウィジェットアクションの大部分は、デフォルトのコントローラークラス *CControllerDashboardWidgetView* を使用または拡張します。

ウィジェットのアクションクラスは *actions* ディレクトリに配置され、*manifest.json* ファイル内の*actions*パラメーター (*actions/widget.{id}.view/class*) で指定される必要があります。

actions/WidgetView.php example (implemented in the Zabbix-native *System information* widget)

```
class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getDefaultName()),
            'system_info' => CSystemInfoHelper::getData(),
            'info_type' => $this->fields_values['info_type'],
            'user_type' => CWebUser::getType(),
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

Widget view

ウィジェットビュークラス (*CWidgetView*) は、ウィジェットプレゼンテーションビューの構築を担当します。

ウィジェットビュークラスは *views* ディレクトリに配置する必要があります。ウィジェットビュークラスを含むファイルの名前がデフォルト (*widget.view.php*) とは異なる場合は、*manifest.json* ファイルの*actions*パラメーター (*actions/widget.{id}.view/view*) で指定する必要があります。

views/widget.view.php example

```
<?php

/**
 * My custom widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

JavaScript クラスは、ウィジェットデータの更新、ウィジェットのサイズの変更、ウィジェット要素の表示など、ウィジェットの動作を決定する責任があります。

すべての JavaScript 操作は、すべてのダッシュボードウィジェットのベース JavaScript クラスである *CWidget* を使用または拡張をします。*CWidget* クラスには、ウィジェット動作のデフォルトの実装を備えた一連のメソッドが含まれています。ウィジェットの複雑さに応じて、これらのメソッドはそのまま使用または拡張できます。

CWidget クラスには、次のメソッドが含まれています。

- ウィジェットのライフサイクルを定義するメソッド: *_init()*, *_registerEvents()*, *_doActivate()*, *_doDeactivate()*, *_doDestroy()*, *setEditMode()*.
- ウィジェットのデータ更新と表示を処理するメソッド: *_promiseUpdate()*, *_getUpdateRequestData()*, *_processUpdateResponse(response)*, *_processUpdateErrorResponse(error)*, *_setContents()*.
- ウィジェットの外観を変更するメソッド: *resize()*, *_hasPadding()*.

JavaScript クラスは *assets/js* ディレクトリに配置し、*manifest.json* ファイルの*assets* (*assets/js*) パラメーターで指定する必要があります。

ライフサイクルメソッド

ウィジェットのライフサイクルメソッドは、ダッシュボードによって呼び出され、ダッシュボード内に存在する間、ウィジェットのライフサイクルのさまざまな段階で呼び出されます。

onInitialize() メソッドは、HTML またはデータ操作を実行せずに、ウィジェットの初期状態や値を定義します。このメソッドは通常、ウィジェットをダッシュボードページに追加するか、ダッシュボードページをロードするか、ウィジェットが作成されたときに呼び出されます (ウィジェットオブジェクトがインスタンス化されています)。

例:

```
onInitialize() {
    this._time_offset = 0;
    this._interval_id = null;
    this._clock_type = CWidgetClock.TYPE_ANALOG;
    this._time_zone = null;
    this._show_seconds = true;
    this._time_format = 0;
    this._tzone_format = 0;
    this._show = [];
    this._has_contents = false;
    this._is_enabled = true;
}
```

onStart() メソッドは、データ操作を実行せずに、ウィジェットの HTML 構造を定義します。このメソッドは、ダッシュボードページの最初のアクティブ化、つまりダッシュボードとそのウィジェットがユーザーに完全に表示される前に呼び出されます。

例:

```
onStart() {
    this._events.resize = () => {
        const padding = 25;
        const header_height = this._view_mode === ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER
            ? 0
            : this._header.offsetHeight;

        this._target.style.setProperty(
            '--content-height',
            `${this._cell_height * this._pos.height - padding * 2 - header_height}px`
        );
    }
}
```

onActivate() メソッドは、カスタムイベントリスナー (ユーザーアクションに応答するためのもの) を有効にし、ウィジェットの更新サイクル (コンテンツを最新の状態に保つためのもの) を開始することにより、ウィジェットをアクティブにします。このメソッドは、ダッシュボードページがアクティブになったとき、つまりユーザーインターフェースに完全に表示されるときに呼び出されます。

onActivate() メソッドが呼び出される前は、ウィジェットは非アクティブ状態 (WIDGET_STATE_INACTIVE) であることに注意してください。呼び出しが成功した後、ウィジェットはアクティブ状態 (WIDGET_STATE_ACTIVE) に移行します。アクティブ状態では、ウィジェットはレスポンシブで、イベントに耳を傾け、コンテンツを定期的に更新し、他のウィジェットと対話できます。

例:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

onDeactivate() メソッドは、カスタムイベントリスナーを非アクティブ化し、ウィジェットの更新サイクルを停止することにより、ウィジェットのアクティビティとインタラクティブ性を停止します。このメソッドは、ダッシュボードページが無効になっている場合、つまり切り替えまたは削除されたとき、またはダッシュボードページからウィジェットが削除されたときに呼び出されます。

onDeactivate() メソッドが呼び出される前は、ウィジェットはアクティブ状態 (WIDGET_STATE_ACTIVE) であることに注意してください。呼び出しが成功した後、ウィジェットは非アクティブ状態 (WIDGET_STATE_INACTIVE に移行します。

例:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

onDestroy() メソッドは、ウィジェットがダッシュボードから削除される前にクリーンアップタスクを実行します。クリーンアップタスクとは、ウィジェットの初期化中に確立されたデータベース接続を閉じたり、リソースリークを回避するために一時的なデータをクリーンアップしてシステムメモリを解放したり、不要なイベントの取り扱いやメモリリークを防ぐためにサイズ変更イベントまたはボタンクリックに関連するイベントリスナーの登録を解除したりなどします。このメソッドは、ウィジェットまたはそれを含むダッシュボードページが削除されたときに呼び出されます。

Note that before the *onDestroy()* method is invoked, a widget in an active state (WIDGET_STATE_ACTIVE) is always deactivated with the invocation of the *onDeactivate()* method. *onDestroy()* メソッドが呼び出される前にアクティブな状態 (WIDGET_STATE_ACTIVE) のウィジェットは、*onDeactivate()* メソッドの呼び出しで常に非アクティブ化されることに注意してください。

例:

```
onDestroy() {
    if (this._filter_widget) {
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_MARK, this._events.mark);
        this._filter_widget.off(CWidgetMap.WIDGET_NAVTREE_EVENT_SELECT, this._events.select);
    }
}
```

onEdit() メソッドは、ダッシュボードが編集モードに変換されるときのウィジェットの外観と動作を定義します。このメソッドは、通常ユーザーがウィジェットの編集ボタンまたはダッシュボードのダッシュボードの変更ボタンを押し、ダッシュボードが編集モードに移行するときに呼び出されます。

例:

```
onEdit() {
    this._deactivateGraph();
}
```

更新プロセスメソッド

ウィジェット更新プロセスメソッドは、Zabbix サーバーまたはその他のデータソースから更新されたデータを取得し、ウィジェットに表示する役割を担います。

promiseUpdate() メソッドは、通常、Web リクエストまたは API 呼び出しを使用してデータを取得することにより、データ更新プロセスを開始します。このメソッドは、ダッシュボードページが表示されたときに呼び出され、その後、ダッシュボードページが別のダッシュボードページに切り替わるまで定期的に呼び出されます。

以下は、ほとんどの Zabbix ネイティブウィジェットが使用する *promiseUpdate()* メソッドのデフォルトの実装の例です。デフォルトの実装では、*promiseUpdate()* メソッドは、サーバーからデータを取得するための一般的なパターンに従います。適切な URL とリクエストパラメーターを使用して新しい Curl オブジェクトを作成し、*getUpdateRequestData()* メソッドで構築されたデータオブジェクトを使用して *fetch()* メソッドで POST リクエストを送信し、*processUpdateResponse(response)* または *processUpdateErrorResponse(error)* に応じてレスポンス (またはエラーレスポンス) を処理します。この実装は、通常 JSON 形式でデータを取得し、一貫した方法で処理するため、ほとんどのウィジェットに適しています。

```
promiseUpdate() {
    const curl = new Curl('zabbix.php');

    curl.setArgument('action', `widget.${this._type}.view`);

    return fetch(curl.getUrl(), {
        method: 'POST',
        headers: {'Content-Type': 'application/json'},
        body: JSON.stringify(this.getUpdateRequestData()),
        signal: this._update_abort_controller.signal
```

```
    })
        .then((response) => response.json())
        .then((response) => {
            if ('error' in response) {
                this.processUpdateErrorResponse(response.error);

                return;
            }

            this.processUpdateResponse(response);
        });
    }
```

_getUpdateRequestData() メソッドは、ウィジェットの状態と設定からさまざまなプロパティとその対応する値 (ウィジェット識別子、フィルター設定、時間範囲など) を収集し、ウィジェットを更新するためのサーバーリクエストデータを準備し、更新リクエストでサーバーに送信される必要な情報を表すデータオブジェクトを構築します。このメソッドは、ウィジェット更新プロセス中にデフォルトの [*_promiseUpdate()*] メソッドの一部としてのみ呼び出されます。

デフォルトの実装:

```
_getUpdateRequestData() {
    return {
        templateid: this._dashboard.templateid ?? undefined,
        dashboardid: this._dashboard.dashboardid ?? undefined,
        widgetid: this._widgetid ?? undefined,
        name: this._name !== '' ? this._name : undefined,
        fields: Object.keys(this._fields).length > 0 ? this._fields : undefined,
        view_mode: this._view_mode,
        edit_mode: this._is_edit_mode ? 1 : 0,
        dynamic_hostid: this._dashboard.templateid !== null || this.supportsDynamicHosts()
            ? (this._dynamic_hostid ?? undefined)
            : undefined,
        ...this._content_size
    };
}
```

processUpdateResponse(response) メソッドは、更新リクエスト後にサーバーから受信したレスポンスを処理し、更新プロセスがエラーなしで成功している場合、ウィジェットデータをクリアし、*setContents()* メソッドで新しいコンテンツを表示します。このメソッドは、デフォルトの *promiseUpdate()* メソッドの一部として、つまりウィジェットの更新プロセス中のみに呼び出されます。

デフォルトの実装:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

processUpdateErrorResponse(error) メソッドは、レスポンスがエラーである場合、更新リクエスト後にサーバーから受信したレスポンスを処理し、エラーメッセージ/-s を表示します。このメソッドは、デフォルトの *promiseUpdate()* メソッドの一部として、つまりウィジェット更新プロセス中のみ呼び出されます。

デフォルトの実装:

```
_processUpdateErrorResponse(error) {
    this._setErrorContents({error});
}

_setErrorContents({error}) {
    const message_box = makeMessageBox('bad', error.messages, error.title)[0];

    this._content_body.innerHTML = '';
```

```
        this._content_body.appendChild(message_box);

        this._removeInfoButtons();
    }
```

setContents(response) メソッドは、ウィジェットの更新プロセスがエラーなく成功した場合にウィジェットの内容を表示します。これには、DOM 要素の操作、UI コンポーネントの更新、スタイルの適用、フォーマットなどが含まれます。このメソッドは、デフォルトの *processUpdateResponse(response)* メソッドの一部として、つまり更新リクエスト後にサーバーから受信したレスポンスを処理するプロセス中のみ呼び出されます。

デフォルトの実装:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

プレゼンテーション変更メソッド

ウィジェットのプレゼンテーション変更メソッドは、ウィジェットの外観を変更します。

onResize() メソッドは、新しいウィジェットサイズに対応するためにウィジェットの視覚要素の調整を担当します。これには、要素の再配置、要素の寸法調整、テキストの切り捨て、サイズ変更中の応答性を向上させるための遅延読み込みの実装などが含まれます。このメソッドは、ウィジェットがサイズ変更されたときに呼び出されます。たとえば、ユーザーがウィジェットを手動でサイズ変更するときやブラウザウィンドウのサイズが変更されたときです。

例:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

hasPadding() メソッドは、ヘッダーを表示に構成されている場合、ウィジェットの下部に 8px の垂直パディングを適用します。このメソッドは、ダッシュボードページがアクティブになったとき、つまりユーザーインターフェースの表示されるページになったときに呼び出されます。

デフォルトの実装:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

一部のウィジェットの場合、使用可能なすべてのウィジェットスペースを使用して、たとえばカスタムの背景色を構成する必要があります。以下は、zabbix ネイティブのアイテムの値ウィジェットで使用される *hasPadding()* メソッドの実装の例です。

```
hasPadding() {
    return false;
}
```

チュートリアル

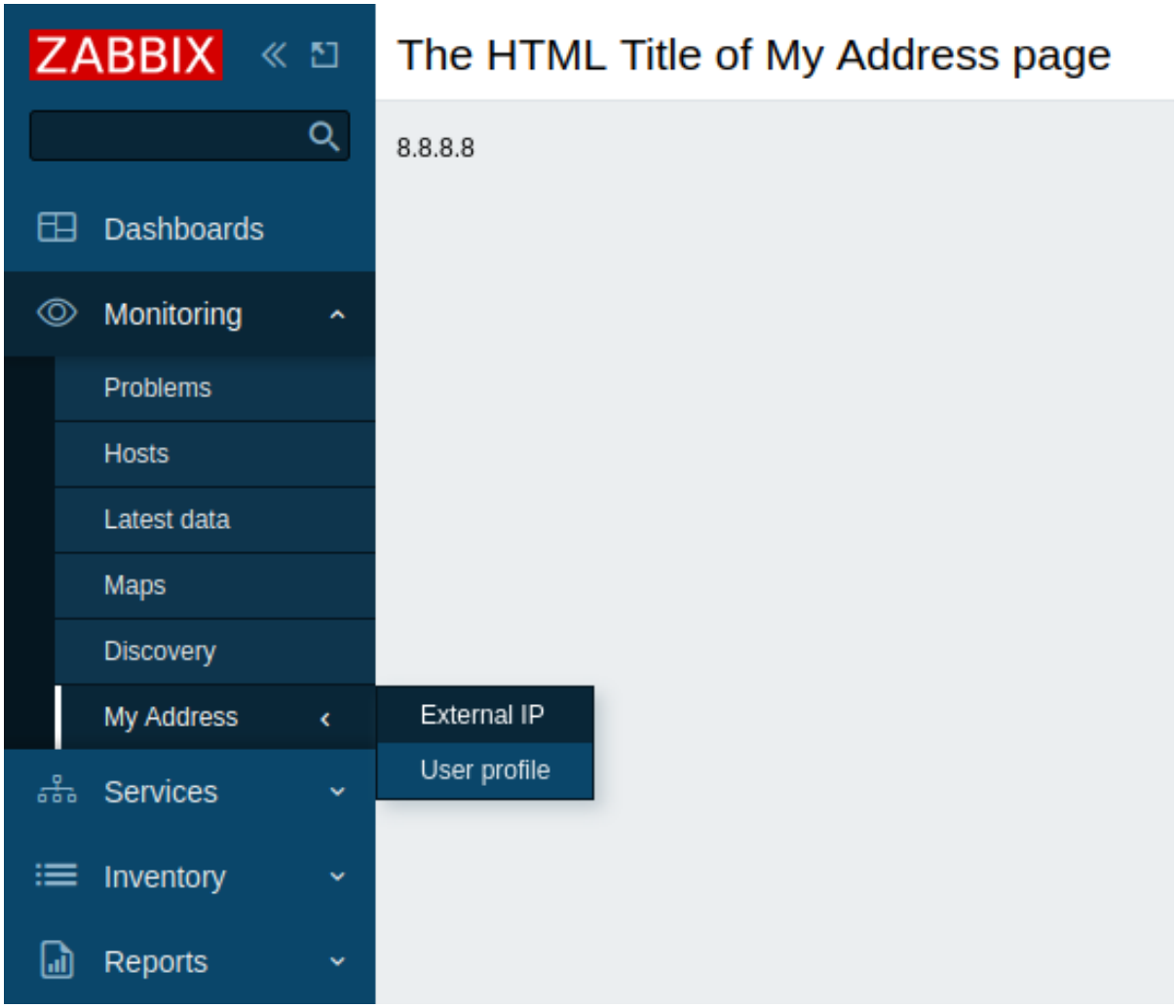
このセクションには、Zabbix でカスタムモジュールとウィジェットを構築する方法を説明する、実践的なステップバイステップのチュートリアルが含まれています。

モジュールの作成 **(**チュートリアル**)**

これは、シンプルな Zabbix フロントエンドモジュールを作成する方法を示すステップバイステップのチュートリアルです。このモジュールのすべてのファイルを ZIP アーカイブとしてダウンロードできます: MyAddress.zip。

作成するもの

このチュートリアルでは、まず新しい *My Address* メニューセクションを追加するフロントエンドモジュールを構築し、次にそれをより高度なフロントエンドモジュールに変換します。このモジュールは、*https://api.seeip.org* に HTTP リクエストを送信し、レスポンス (コンピューターの IP アドレス) を新しく作成された *My Address* メニューセクションの新しいページに表示します。完成したモジュールは次のようになります:



パート I - 新しいメニューセクション

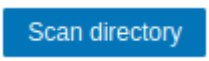
Zabbix フロントエンドに空のモジュールを追加

1. Zabbix フロントエンドインストールの *modules* ディレクトリに *MyAddress* ディレクトリを作成します (例: *zabbix/ui/modules*)。
2. 基本モジュールのメタデータを含む *manifest.json* ファイルを作成します (サポートされているパラメーターの説明を参照)。

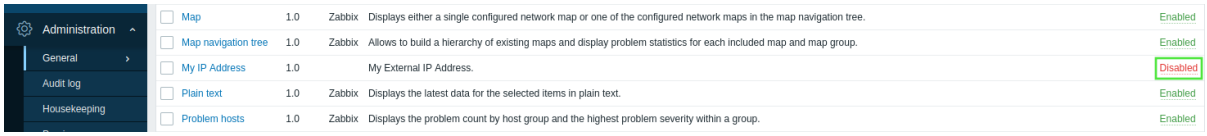
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Zabbix フロントエンドで、管理 → 一般 → モジュールセクションに移動し、ディレクトリのスキャンボタンをクリックします。



4. リスト内で新しいモジュール *My IP Address* を見つけ、" 無効" ハイパーリンクをクリックしてモジュールのステータスを" 無効" から" 有効" に変更します (モジュールがリストされていない場合、トラブルシューティングセクションを確認してください)。



これで、モジュールがフロントエンドに登録されました。ただし、モジュールの機能を定義する必要があるため、まだ表示されていません。モジュールディレクトリにコンテンツを追加し、ページを更新すると、Zabbix フロントエンドの変更がすぐに表示されます。

メニューセクションの作成

1. *MyAddress* ディレクトリに *Module.php* ファイルを作成します。

このファイルは、デフォルトの *CModule* クラスを拡張する新しい *Module* クラスを実装します。*Module* クラスは、新しい *My Address* メニューセクションをメインメニューに挿入します。

setAction() メソッドは、メニューセクションをクリックしたときに実行されるアクションを指定します。まず、事前定義されたアクション *userprofile.edit* を使用すると、ユーザープロファイルページが開きます。このチュートリアルのパート IIIでは、カスタムアクションの作成方法を学習します。

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

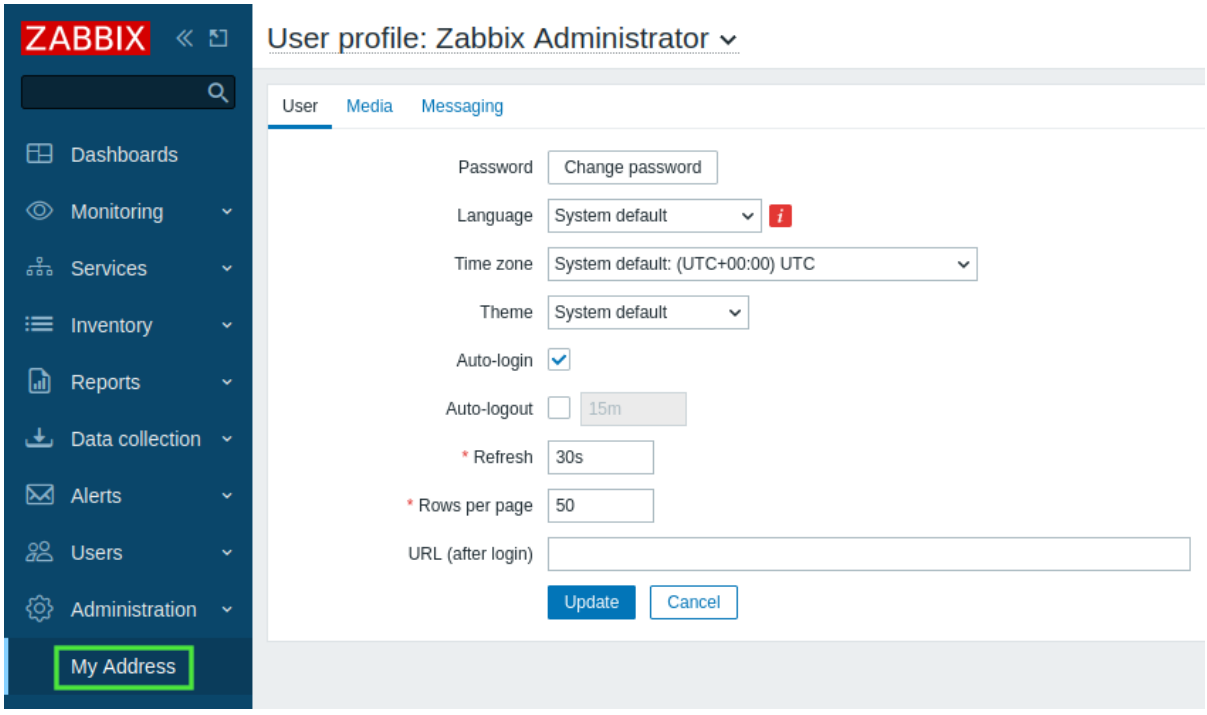
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
'userprofile.edit'を他のアクション、たとえば'charts.view'(カスタムグラフを開く)、'problems.view'(データ監視 → 障害を開く)、または 'report.status'(システム情報レポートを開く) に置き換えることができます。

3. Zabbix フロントエンドを更新します。Zabbix メインメニューの下部に、新しい *My Address* セクションが追加されました。*My Address* をクリックして、*User profile* ページを開きます。



パート II - メニューセクションの位置変更

このパートでは、*My Address* メニューセクションを監視データセクションに移動しし、そこにネストされたメニューを追加します。その結果、ユーザーは監視データ → マイアドレスメニューセクションから 2 つのサブメニューページにアクセスできるようになり

ます。

1. *Module.php* ファイルを開いて編集します。

ui/modules/MyAddress/Module.php

```
<?php

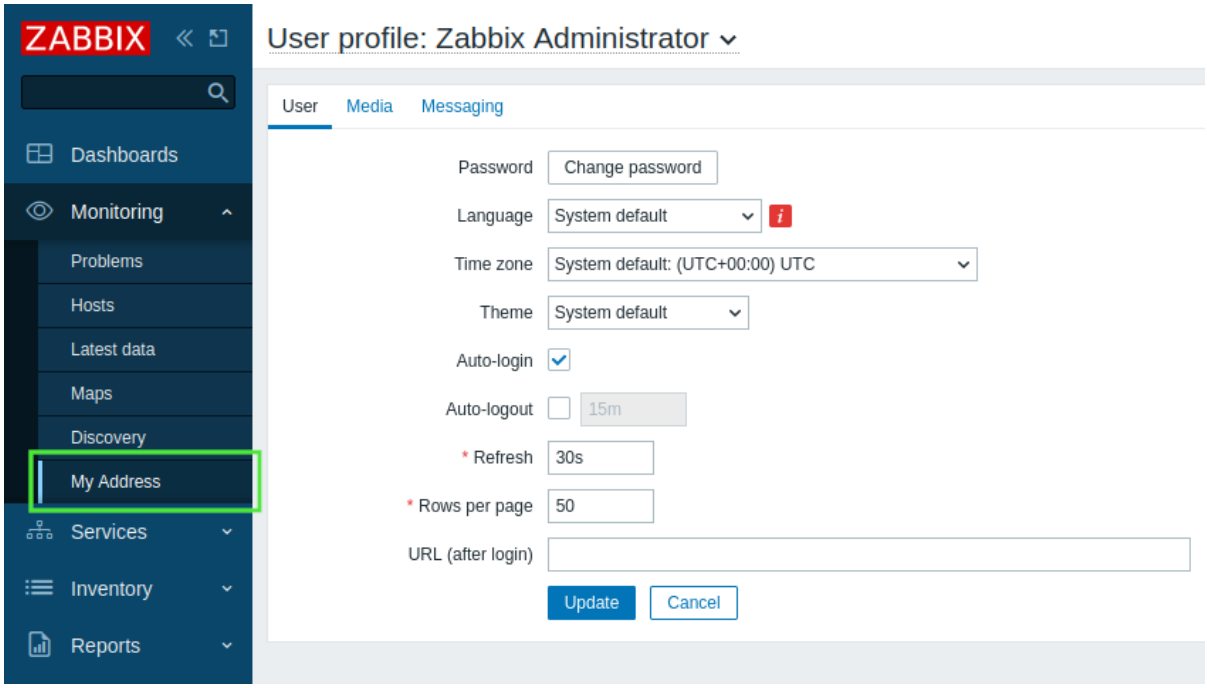
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Zabbix フロントエンドを更新します。監視データメニューセクションを展開し、*My address* セクションがディスカバリセクションの下にあることを確認します。



3. ネストされたページを *My Address* メニューセクションに追加するには、*Module.php* ファイルを再度開いて編集します。

このステップでは、2 つのサブセクションが作成されます。

- 次の手順で定義する新しい'my.address' アクションを実行する *External IP*。
- *User profile* ページを開くための事前定義された'userprofile.edit' アクションを実行する *User profile*。

ネストされたメニューの場合は、前の手順で使用したクラスに加えて *CMenu* クラスを使用する必要があることに注意してください。

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

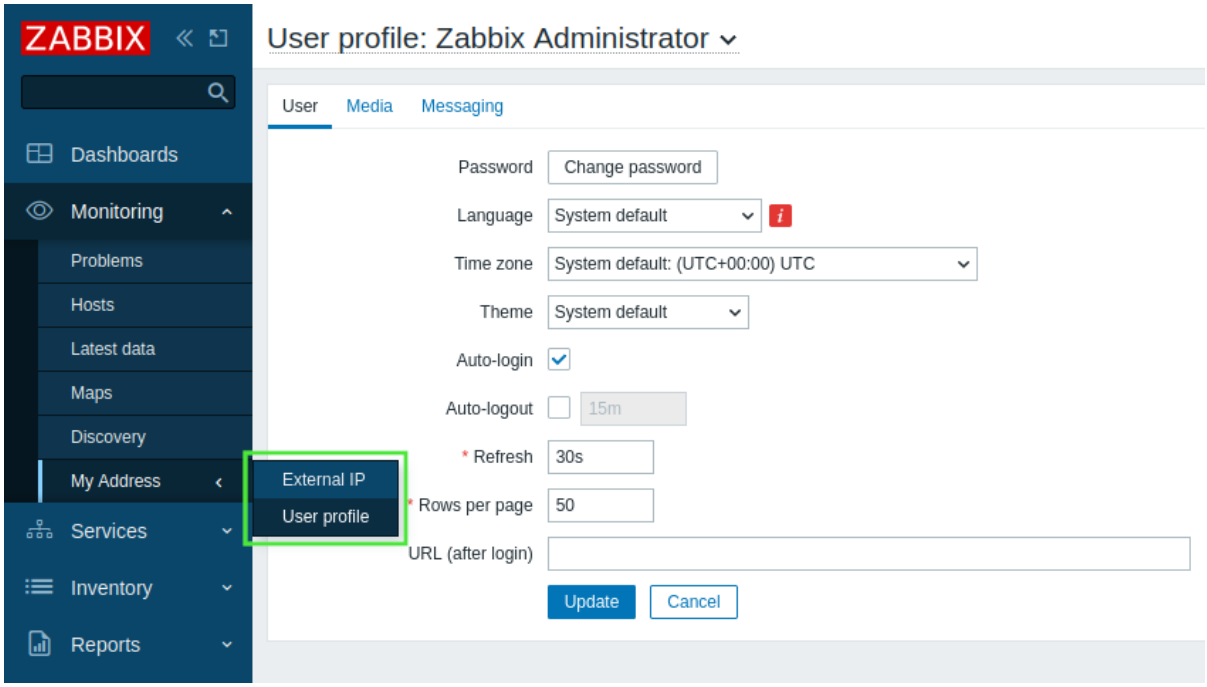
use Zabbix\Core\CModule,
    APP,
```

```
    CMenu,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Zabbix フロントエンドを更新します。*My address* メニューセクションに、*External IP* と *User profile* の 2 ページからなる第 3 レベルのメニューが含まれていることを確認します。



パート III - モジュールアクション

アクションは、*actions/MyAddress.php* と *views/my.address.php* の 2 つのファイルに実装されます。***actions/MyAddress.php*** ファイルはビジネスロジックの実装を担当し、***views/my.address.php*** ファイルはビューを担当します。

1. *MyAddress* ディレクトリに *actions* ディレクトリを作成します。
2. *actions* ディレクトリに *MyAddress.php* ファイルを作成します。

アクションロジックは *MyAddress* クラスで定義されます。このアクションクラスは、*init()*、*checkInput()*、*checkPermissions()*、*doAction()* の 4 つの関数を実装します。Zabbix フロントエンドは、アクションが要求されると *doAction()* 関数を呼び出します。この関数は、モジュールのビジネスロジックを担当します。

Attention:
データは連想配列として編成する必要があります。配列は多次元にすることができ、ビューで予期される任意のデータを含めることができます。

ui/modules/MyAddress/actions/MyAddress.php

```php
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
        $response = new CControllerResponseData($data);
        $this->setResponse($response);
    }
}
```

3. *MyAddress* ディレクトリに新しいディレクトリ *views* を作成します。
4. *views* ディレクトリに *my.address.php* ファイルを作成し、モジュールビューを定義します。

変数 `$data` は、特に定義しなくてもビューで使用できることに注意してください。フレームワークは自動的に連想配列をビューに渡します。

ui/modules/MyAddress/views/my.address.php

```php
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

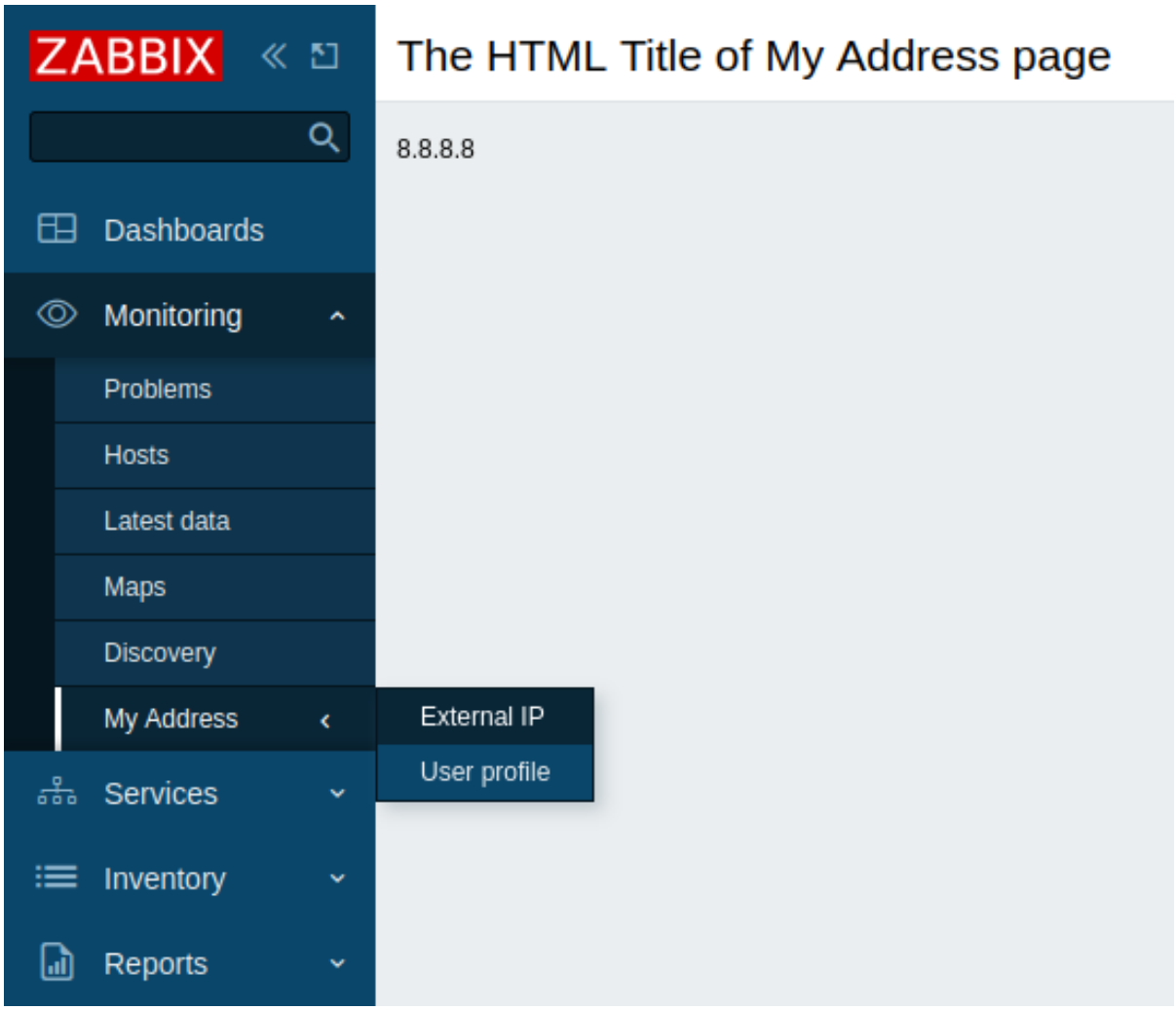
5. モジュールアクションは *manifest.json* ファイルに登録する必要があります。*manifest.json* を開き、以下を含む新しいオブジェクト `actions` を追加します。

- 小文字 (a~z) で書かれ、単語がドットで区切られたアクション名を持つアクションキー (例: `my.address`)
- `my.address` オブジェクトの `class` キーの値としてのアクションクラス名 (`MyAddress`)
- `my.address` オブジェクトの `view` キーの値としてのアクションビュー名 (`my.address`)

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

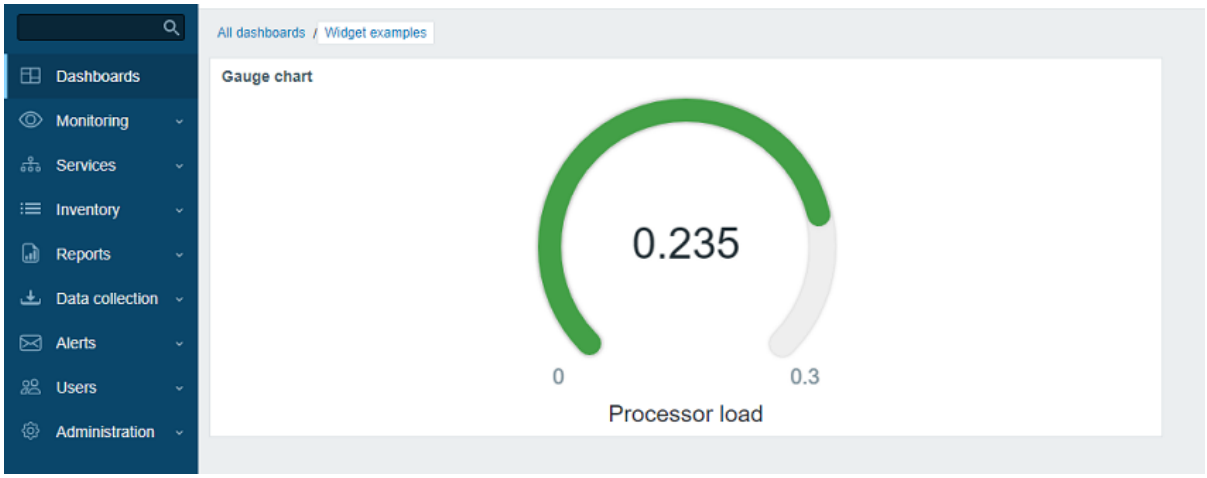
6. Zabbix フロントエンドを更新します。*My address* → *External IP* をクリックして、コンピュータの IP アドレスを確認します。



ウィジェットの作成 **(**チュートリアル**)**

これは、シンプルなダッシュボードウィジェットを作成する方法を示すステップバイステップのチュートリアルです。このウィジェットのすべてのファイルを ZIP アーカイブとしてダウンロードできます: lesson_gauge_chart.zip。

作成するもの　このチュートリアルでは、まず基本的な”Hello, world!” ウィジェットを作成し、次にそれをより高度なウィジェットに変換して、アイテムの値をゲージチャートとして表示します。完成したウィジェットは次のようになります:



パート **I - ”Hello, world!”**　このセクションでは、最小限必要なウィジェット要素を作成し、新しいウィジェットを Zabbix フロントエンドに追加する方法を学びます。

Zabbix フロントエンドに空のウィジェットを追加する

1. Zabbix フロントエンドインストールの *modules* ディレクトリに *lesson_gauge_chart* ディレクトリを作成します (例: *zabbix/ui/modules*)

> **Note:**
> すべてのカスタムウィジェットは外部モジュールとして扱われるため、Zabbix フロントエンドインストールの *modules* ディレクトリに追加する必要があります (例: *zabbix/ui/modules*) *zabbix/ui/widgets* ディレクトリは、Zabbix 組み込みウィジェット用に予約されており、Zabbix UI とともに更新されます。

2. 基本的なウィジェットのメタデータを含む *manifest.json* ファイルを作成します (サポートされているパラメーターの説明を参照)

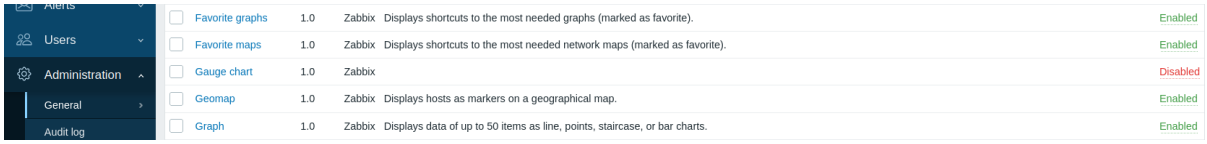
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

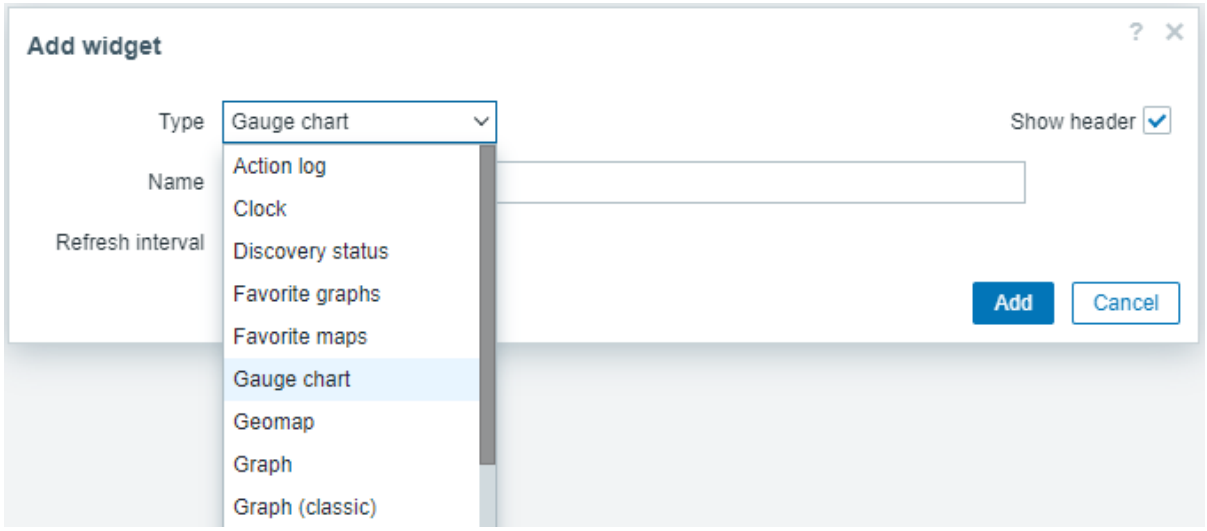
3. Zabbix フロントエンドで、管理 → 一般 → モジュールセクションに移動し、ディレクトリのスキャンボタンをクリックします。



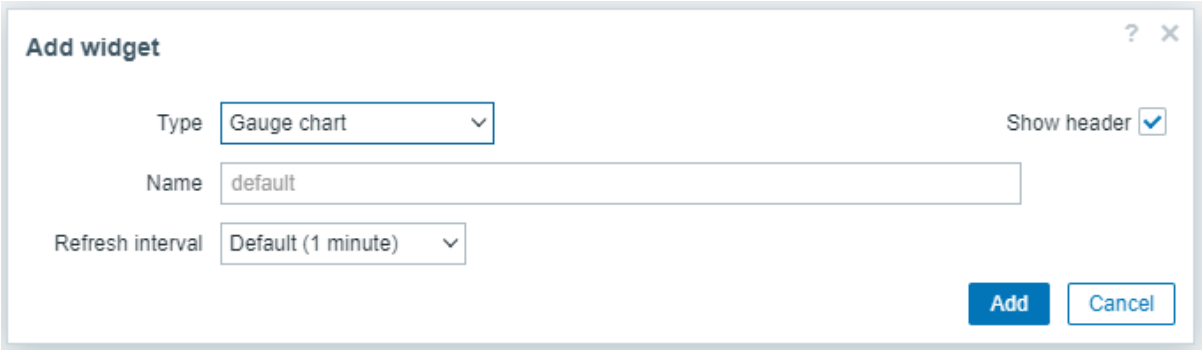
4. リスト内で新しいモジュール *Gauge chart* を見つけ、” 無効” ハイパーリンクをクリックしてモジュールのステータスを” 無効” から” 有効” に変更します (モジュールがリストされていない場合、トラブルシューティングセクションを確認してください)



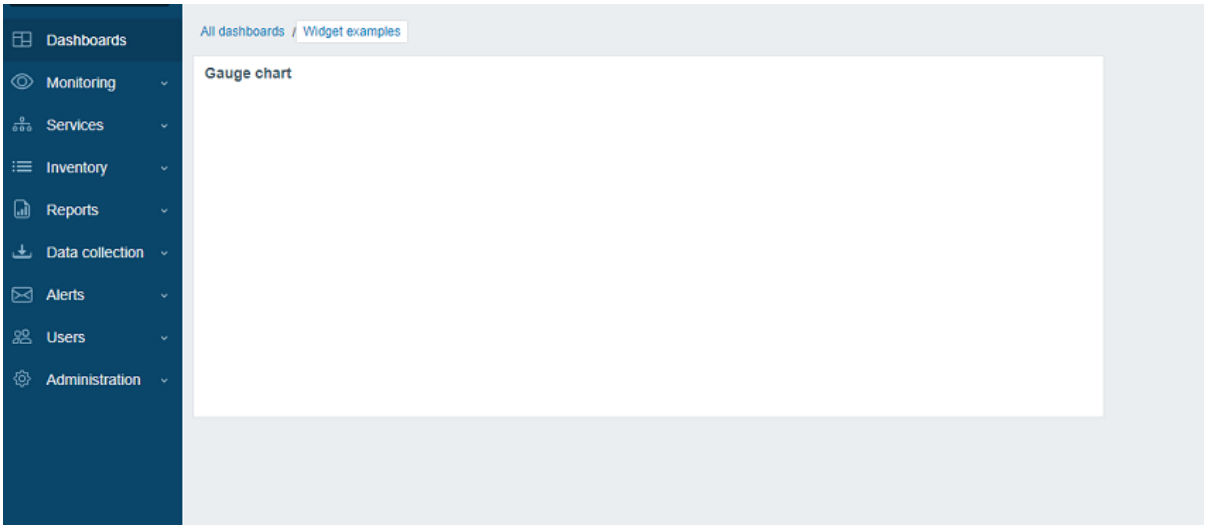
5. ダッシュボードを開き、編集モードに切り替えて、新しいウィジェットを追加します。タイプフィールドで、”Gauge chart” を選択します。



6. この時点で、*Gauge chart* ウィジェット構成には、共通のウィジェットフィールド *Name* と *Refresh interval* のみが含まれています。追加をクリックしてウィジェットをダッシュボードに追加します。



7. 空のウィジェットがダッシュボードに表示されます。右上隅にある変更を保存をクリックして、ダッシュボードを保存します。



ウィジェットビューの追加

Note:
ウィジェットの **view** ファイルは、*views* ディレクトリ（このチュートリアルの場合は、*ui/modules/lesson_gauge_chart/views/*）に配置する必要があります。ファイルのデフォルト名が *widget.view.php* の場合、*manifest.json* ファイルに登録する必要はありません。ファイルの名前が異なる場合は、manifest.jsonファイルの *actions/widget.lesson_gauge_chart.view* セクションで指定します。

1. *lesson_gauge_chart* ディレクトリに *views* ディレクトリを作成します。
2. *views* ディレクトリに *widget.view.php* ファイルを作成します。

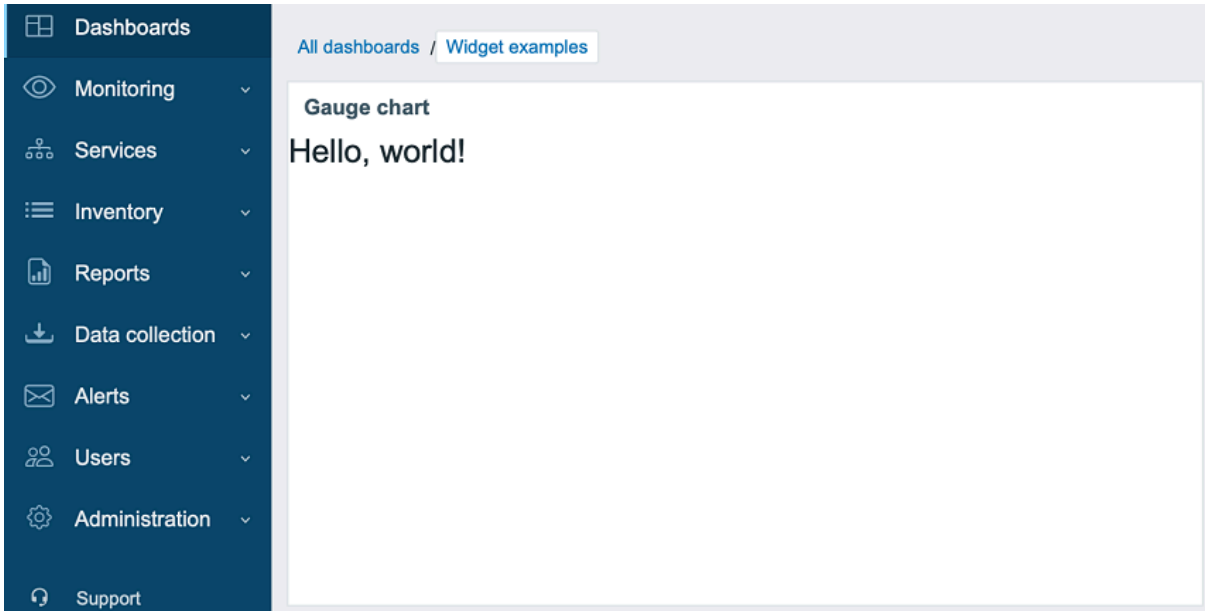
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. ダッシュボードを更新します。*Gauge chart* ウィジェットに”Hello, world!” が表示されるようになりました。



パート **II -** ゲージチャート　設定を構成ビューに追加し、ウィジェットビューで使用する

このセクションでは、ウィジェット構成フィールドを追加し、入力された値をウィジェットビューにテキストとして表示する方法を学習します。

ウィジェット構成は、フォーム (*Zabbix\Widgets\CWidgetForm*) とウィジェットフォームビュー (*widget.edit.php*) で構成されます。フィールド (*Zabbix\Widgets\CWidgetField*) を追加するには、*Zabbix\Widgets\CWidgetForm* を拡張する *WidgetForm* クラスを作成する必要があります。

フォームには、ユーザーが入力した値を検証するために使用されるさまざまなタイプのフィールドのセット (*Zabbix\Widgets\CWidgetField*) が含まれています。各入力要素タイプのフォームフィールド (*Zabbix\Widgets\CWidgetField*) は、値を単一の形式に変換してデータベースに保存します。

> **Note:**
> ウィジェットの **form** ファイルは、*includes* ディレクトリ (このチュートリアルの場合は、*ui/modules/lesson_gauge_chart/includes/*) に配置する必要があります。ファイルのデフォルト名が *WidgetForm.php* の場合、*manifest.json* ファイルに登録する必要はありません。ファイルの名前が異なる場合は、manifest.jsonファイルの *widget/form_class* セクションで指定します。

1. *lesson_gauge_chart* ディレクトリに新しいディレクトリ *includes* を作成します。
2. *WidgetForm.php* ファイルを *includes* ディレクトリに作成します。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. ウィジェット設定フォームに *Description* フィールドを追加します。これは通常のテキストフィールドで、ユーザーは任意の文字セットを入力できます。*CWidgetFieldTextBox* クラスを使用できます。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

use Zabbix\Widgets\Fields\CWidgetFieldTextBox;
```

```
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

4. *views* ディレクトリに、ウィジェット構成ビューファイル *widget.edit.php* を作成し、新しい *Description* フィールドのビューを追加します。*CWidgetFieldTextBox* フィールドクラスの場合、ビューは *CWidgetFieldTextBoxView* です。

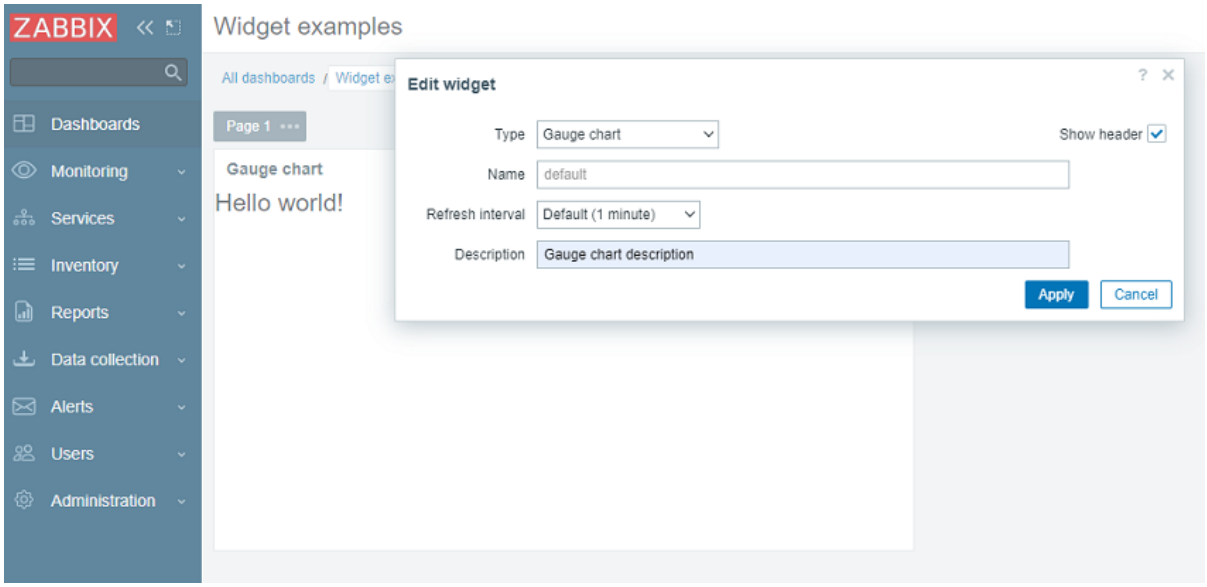
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. ダッシュボードに移動し、ウィジェットの歯車アイコンをクリックしてウィジェット構成フォームを開きます。
6. ウィジェット設定フォームには、新しい *Description* テキストフィールドが含まれるようになりました。*Gauge chart description* のような、任意の値を入力します。



7. ウィジェット設定フォームで適用をクリックします。次に、右上隅にある変更を保存をクリックしてダッシュボードを保存します。新しい説明はどこにも表示されず、ウィジェットには”Hello, world!” が表示されたままであることに注意してください。

新しい説明をウィジェットに表示するには、*Description* フィールドの値をデータベースから取得し、ウィジェットビューに渡す必要があります。そのためには、アクションクラスを作成する必要があります。

8. *lesson_gauge_chart* ディレクトリに新しいディレクトリ *actions* を作成します。
9. *actions* ディレクトリに *WidgetView.php* ファイルを作成します。*WidgetView* アクションクラスは *CControllerDashboardWidgetView* クラスを拡張します。

ウィジェット構成フィールドの値は、アクションクラスの **$fields_values** プロパティに保存されます。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

10. *manifest.json* を開き、*actions/widget.lesson_gauge_chart.view* セクションに *WidgetView* をアクションクラスとして登録します。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. これで、*$data['description']* に含まれる説明フィールドの値をウィジェットビューで使用できるようになりました。*views/widget.view.php* を開き、静的テキスト"Hello, world!" を *$data['description']* に置き換えます。

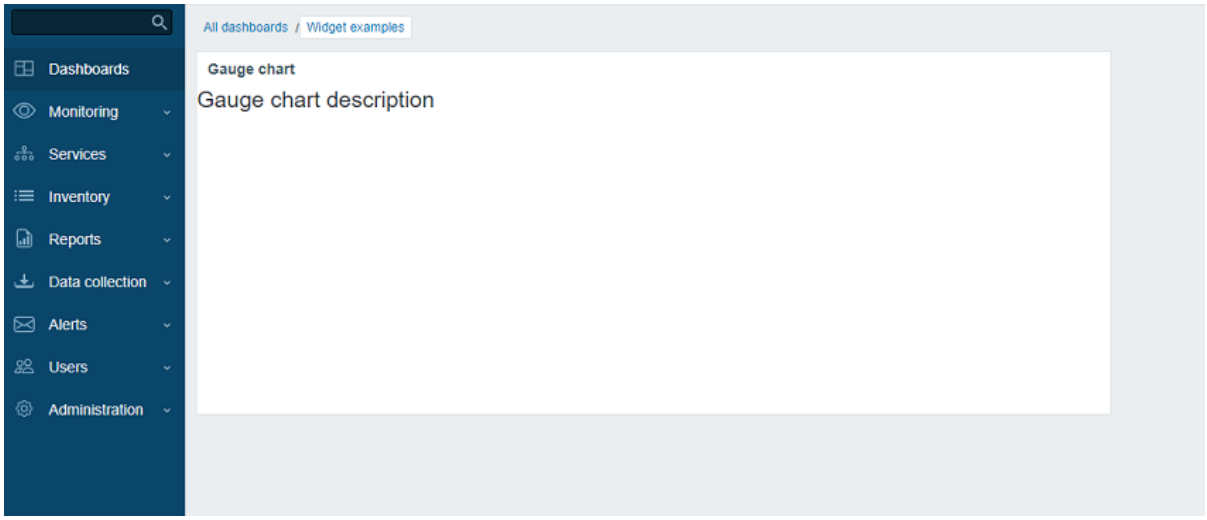
ui/modules/lesson_gauge_chart/views/widget.view.php

```php
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. ダッシュボードページを更新します。"Hello, world!" の代わりにウィジェットの説明テキストが表示されるはずです。



API 経由でアイテム値の取得

ウィジェットには、ユーザーが選択したアイテムの最新の値が表示される必要があります。そのためには、ウィジェット構成でアイテムを選択する機能を追加する必要があります。

このセクションでは、アイテム選択フィールドをウィジェットフォームに追加する方法と、このフィールドのビジュアル部分を構成ビューに追加する方法を学習します。その後、ウィジェットコントローラーは API リクエストを介してアイテムデータとその値を取得できるようになります。値を受信すると、ウィジェットビューに表示できます。

1. *includes/WidgetForm.php* を開き、*CWidgetFieldMultiSelectItem* フィールドを追加します。これにより、設定フォームでアイテムを選択できるようになります。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

2. *views/widget.edit.php* を開き、フィールドビジュアルコンポーネントを構成ビューに追加します。

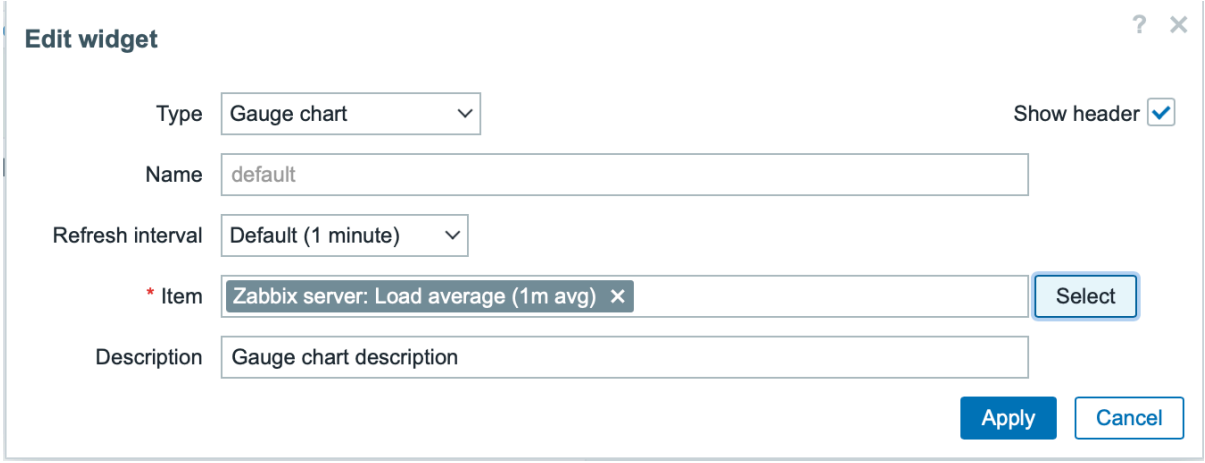
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. ダッシュボードに戻り、ウィジェットの歯車アイコンをクリックしてウィジェット構成フォームを開きます。
4. ウィジェット設定フォームには、新しい入力フィールド *Item* が含まれるようになりました。ホスト”Zabbix server” とアイテム”Load average (1m avg)” を選択します。



5. ウィジェット設定フォームで適用をクリックします。次に、右上隅にある変更を保存をクリックしてダッシュボードを保存します。
6. *actions/WidgetView.php* を開いて変更します。

今後、アイテム ID は *$this->fields_values['itemid']* のウィジェットコントローラーで利用できるようになります。*doAction()* コントローラーメソッドは、API メソッド*item.get*を使用してアイテムデータ (名前、値のタイプ、単位) を、API メソッド*history.get*を使用してアイテムの最新の値を収集します。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```php
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
```

```
            ]
        ]);

        $value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'value' => $value,
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

7. *views/widget.view.php* を開き、アイテム値をウィジェットビューに追加します。

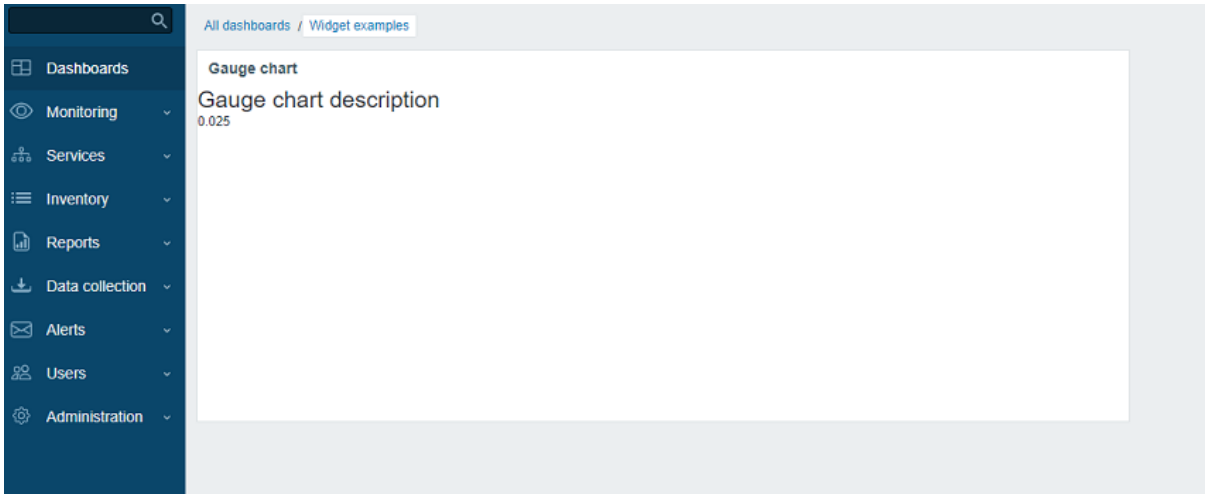
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('No data'))
    ])
    ->show();
```

8. ダッシュボードページを更新します。ウィジェットには最新のアイテム値が表示されます。



Advanced configuration を構成ビューに追加

このセクションでは、色、最小値と最大値、単位、前に作成した *Description* フィールドなどのオプションのパラメーターを含む、展開/折りたたみ可能な *Advanced configuration* セクションを追加する方法を学習します。

1. メインウィジェットディレクトリ *lesson_gauge_chart* に *Widget.php* ファイルを作成し、新しいクラス *Widget* を作成します。

Widget クラスは、*CWidget* 基本クラスを拡張して、デフォルトのウィジェット設定 (この場合は翻訳) を追加/上書きします。以下に示す JavaScript は、データが欠落している場合に文字列"No data" を表示します。Zabbix UI 翻訳ファイルには"No data" という文字列が存在します。

ウィジェット定数がある場合は、*Widget* クラスにも指定することをお勧めします。

ui/modules/lesson_gauge_chart/Widget.php

```
<?php

namespace Modules\LessonGaugeChart;

use Zabbix\Core\CWidget;

class Widget extends CWidget {

    public const UNIT_AUTO = 0;
    public const UNIT_STATIC = 1;

    public function getTranslationStrings(): array {
        return [
            'class.widget.js' => [
                'No data' => _('No data')
            ]
        ];
    }
}
```

2. *includes/WidgetForm.php* を開き、新しいフィールド *Color*(カラーピッカー)、*Min*(数値フィールド)、*Max*(数値フィールド)、および *Units*(選択) を追加し、カラーピッカーのパレットのデフォルトの色を定義します。次の手順で使用できるようにします。

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};
```

```
use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. *views/widget.edit.php* を開き、フィールドのビジュアルコンポーネントを構成ビューに追加します。

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
```

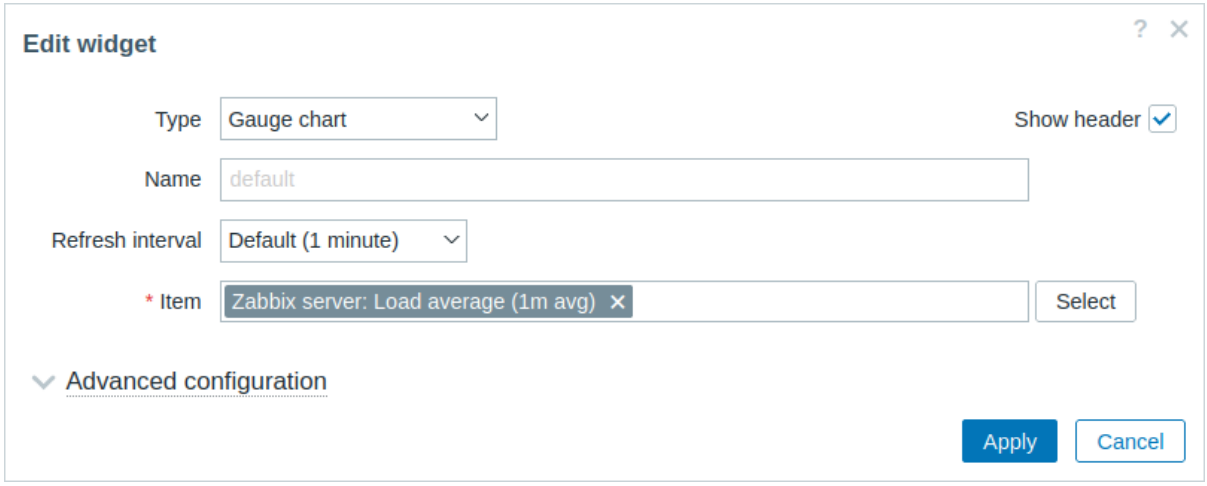
```
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

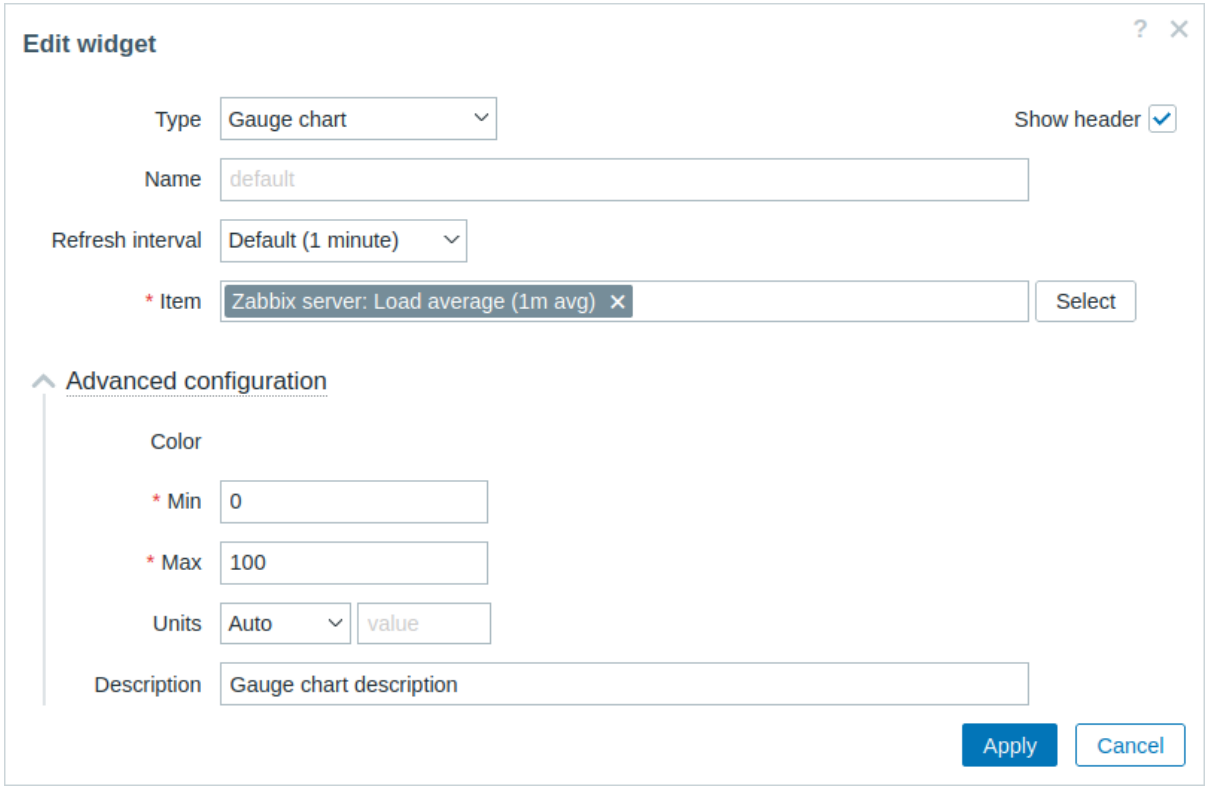
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

Note:
CWidgetFormView クラスの *addField()* メソッドは、CSS クラス文字列を 2 番目のパラメーターとして受け取ります。

4. ダッシュボードに戻り、編集モードに切り替えて、ウィジェットの歯車アイコンをクリックしてウィジェット構成フォームを開きます。ウィジェット設定フォームには、新しい展開/折りたたみ可能なセクション *Advanced configuration* が含まれるようになりました。



5. *Advanced configuration* セクションを展開すると、追加のウィジェット構成フィールドが表示されます。*Color* フィールドにはまだカラーピッカーがないことに注意してください。これは、カラーピッカーを JavaScript で初期化する必要があるためです。これは、次のセクションウィジェットに *JavaScript* を追加で追加されます。



ウィジェットに JavaScript を追加する

このセクションでは、JavaScript を使用して作成したゲージチャートを追加し、最新の値が正常か、高すぎるか、低すぎるかを表示する方法を学びます。

1. *views* ディレクトリに *widget.edit.js.php* ファイルを作成します。

JavaScript は、設定ビューでカラーピッカーを初期化する役割を担います。

ui/modules/lesson_gauge_chart/views/widget.edit.js.php

```
<?php

use Modules\LessonGaugeChart\Widget;

?>

window.widget_lesson_gauge_chart_form = new class {

    init({color_palette}) {
        this._unit_select = document.getElementById('value_units');
        this._unit_value = document.getElementById('value_static_units');

        this._unit_select.addEventListener('change', () => this.updateForm());

        colorPalette.setThemeColors(color_palette);

        for (const colorpicker of jQuery('.<?= ZBX_STYLE_COLOR_PICKER ?> input')) {
            jQuery(colorpicker).colorpicker();
        }

        const overlay = overlays_stack.getById('widget_properties');

        for (const event of ['overlay.reload', 'overlay.close']) {
            overlay.$dialogue[0].addEventListener(event, () => { jQuery.colorpicker('hide'); });
        }

        this.updateForm();
    }
```

```
    updateForm() {
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

2. *views/widget.edit.php* を開き、JavaScript を含む *widget.edit.js.php* ファイルを設定ビューに追加します。
   これを行うには、*includeJsFile()* メソッドを使用します。
   インライン JavaScript を追加するには、*addJavaScript()* メソッドを使用します。

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

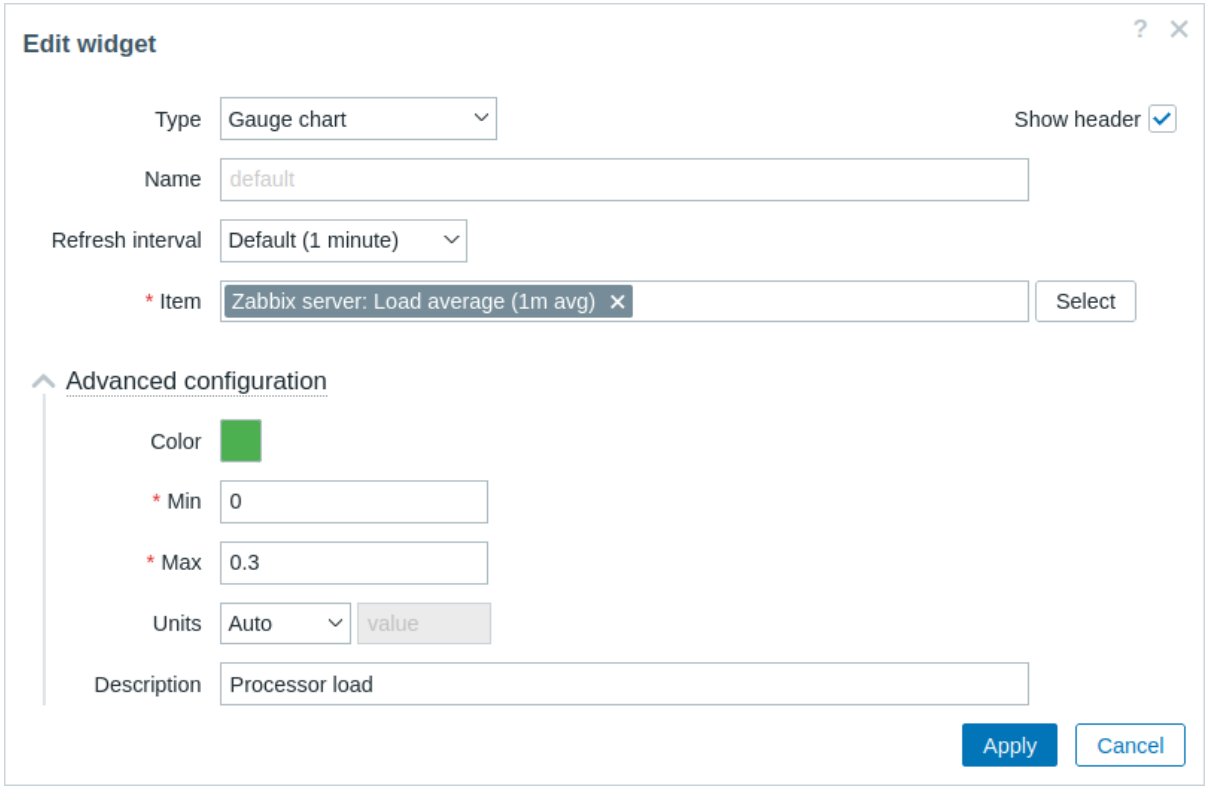
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. ダッシュボードに戻り、ウィジェット内の歯車アイコンをクリックして、ウィジェット設定フォームを開きます。
   次に、初期化されたカラーピッカーを表示するために、*Advanced configuration* セクションを展開します。
   各フィールドに値を入力し、ゲージチャートの色を選択します。



4. ウィジェット設定フォームで *Apply* をクリックします。
   次に、右上隅の *Save changes* をクリックしてダッシュボードを保存します。

5. *actions/WidgetView.php* を開き、コントローラーを更新します。

$this->fields_values プロパティには、すべての *Advanced configuration* フィールドの値が含まれるようになりました。
設定内容と選択したアイテムの値をウィジェットビューに渡せるように、コントローラーを完成させます。

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```
<?php

namespace Modules\LessonGaugeChart\Actions;

use API,
    CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $db_items = API::Item()->get([
            'output' => ['itemid', 'value_type', 'name', 'units'],
            'itemids' => $this->fields_values['itemid'],
            'webitems' => true,
            'filter' => [
                'value_type' => [ITEM_VALUE_TYPE_UINT64, ITEM_VALUE_TYPE_FLOAT]
            ]
        ]);

        $history_value = null;

        if ($db_items) {
            $item = $db_items[0];

            $history = API::History()->get([
                'output' => API_OUTPUT_EXTEND,
                'itemids' => $item['itemid'],
                'history' => $item['value_type'],
                'sortfield' => 'clock',
```

```
                'sortorder' => ZBX_SORT_DOWN,
                'limit' => 1
            ]);

            if ($history) {
                $history_value = convertUnitsRaw([
                    'value' => $history[0]['value'],
                    'units' => $item['units']
                ]);
            }
        }

        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'history' => $history_value,
            'fields_values' => $this->fields_values,
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

6. *views/widget.view.php* を開いて変更します。

次のステップで描画するゲージチャート用のコンテナと、説明用のコンテナを作成する必要があります。

値を JSON オブジェクトとして JavaScript に渡すには、*setVar()* メソッドを使用します。

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. *lesson_gauge_chart* ディレクトリ内に新しい *assets* ディレクトリを作成します。
   このディレクトリは、JavaScript、CSS、およびフォントや画像などのその他のアセットを保存するために使用されます。
8. ウィジェットビュー用の JavaScript のために、*assets* ディレクトリ内に *js* ディレクトリを作成します。
9. *assets/js* ディレクトリに *class.widget.js* ファイルを作成します。

この JavaScript ウィジェットクラスは、すべてのダッシュボードウィジェットのベース JavaScript クラスである *CWidget* を拡張します。

ダッシュボードは、ウィジェットが正しく実装されていることを前提としており、対応する JavaScript メソッドを呼び出すことで、関連する情報をウィジェットに伝えます。
また、何らかの操作が発生した際に、ウィジェットがイベントを生成することも期待しています。
そのため、*CWidget* クラスには、ウィジェット動作のデフォルト実装を持つ一連のメソッドが含まれており、クラスを拡張することでカスタマイズできます。

この場合はいくつかのカスタマイズが必要なため、以下のウィジェット動作に対してカスタムロジックを実装します。

- ウィジェットの初期化。これはウィジェットの初期状態を定義する役割を担います（*onInitialize()* メソッドを参照）。
- ウィジェットの更新処理が成功し、エラーがない場合のウィジェット内容の表示（つまりゲージチャートの描画）（*processUpdateResponse(response)* メソッド、および関連する *_resizeChart()* と *_updatedChart()* メソッドを参照）
- ウィジェットのサイズ変更（*onResize()* メソッド、および関連する *_resizeChart()* メソッドを参照）

ゲージチャートウィジェットのその他の側面については、ウィジェット動作のデフォルト実装を使用します。
CWidget クラスの JavaScript メソッドの詳細については、JavaScript を参照してください。

この JavaScript はウィジェットビューに必要なため、ダッシュボードページとともに読み込まれる必要があります。
JavaScript の読み込みを有効にするには、ステップ 10 に示すように、***manifest.json*** ファイルの *assets/js* および *js_class* パラメータを更新する必要があります。

ui/modules/lesson_gauge_chart/assets/js/class.widget.js

```
class WidgetLessonGaugeChart extends CWidget {

    static UNIT_AUTO = 0;
    static UNIT_STATIC = 1;

    onInitialize() {
        super.onInitialize();

        this._refresh_frame = null;
        this._chart_container = null;
        this._canvas = null;
        this._chart_color = null;
        this._min = null;
        this._max = null;
        this._value = null;
        this._last_value = null;
        this._units = '';
    }

    processUpdateResponse(response) {
        if (response.history === null) {
            this._value = null;
            this._units = '';
        }
        else {
            this._value = Number(response.history.value);
            this._units = response.fields_values.value_units == WidgetLessonGaugeChart.UNIT_AUTO
                ? response.history.units
                : response.fields_values.value_static_units;
        }

        this._chart_color = response.fields_values.chart_color;
        this._min = Number(response.fields_values.value_min);
        this._max = Number(response.fields_values.value_max);

        super.processUpdateResponse(response);
    }

    setContents(response) {
        if (this._canvas === null) {
            super.setContents(response);

            this._chart_container = this._body.querySelector('.chart');
            this._chart_container.style.height =
                `${this._getContentsSize().height - this._body.querySelector('.description').clientHeight}
            this._canvas = document.createElement('canvas');

            this._chart_container.appendChild(this._canvas);

            this._resizeChart();
        }
```

```
        this._updatedChart();
    }

    onResize() {
        super.onResize();

        if (this._state === WIDGET_STATE_ACTIVE) {
            this._resizeChart();
        }
    }

    _resizeChart() {
        const ctx = this._canvas.getContext('2d');
        const dpr = window.devicePixelRatio;

        this._canvas.style.display = 'none';
        const size = Math.min(this._chart_container.offsetWidth, this._chart_container.offsetHeight);
        this._canvas.style.display = '';

        this._canvas.width = size * dpr;
        this._canvas.height = size * dpr;

        ctx.scale(dpr, dpr);

        this._canvas.style.width = `${size}px`;
        this._canvas.style.height = `${size}px`;

        this._refresh_frame = null;

        this._updatedChart();
    }

    _updatedChart() {
        if (this._last_value === null) {
            this._last_value = this._min;
        }

        const start_time = Date.now();
        const end_time = start_time + 400;

        const animate = () => {
            const time = Date.now();

            if (time <= end_time) {
                const progress = (time - start_time) / (end_time - start_time);
                const smooth_progress = 0.5 + Math.sin(Math.PI * (progress - 0.5)) / 2;
                let value = this._value !== null ? this._value : this._min;
                value = (this._last_value + (value - this._last_value) * smooth_progress - this._min) / (t

                const ctx = this._canvas.getContext('2d');
                const size = this._canvas.width;
                const char_weight = size / 12;
                const char_shadow = 3;
                const char_x = size / 2;
                const char_y = size / 2;
                const char_radius = (size - char_weight) / 2 - char_shadow;

                const font_ratio = 32 / 100;

                ctx.clearRect(0, 0, size, size);

                ctx.beginPath();
```

```
                ctx.shadowBlur = char_shadow;
                ctx.shadowColor = '#bbb';
                ctx.strokeStyle = '#eee';
                ctx.lineWidth = char_weight;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.749, Math.PI * 2.251, false);
                ctx.stroke();

                ctx.beginPath();
                ctx.strokeStyle = `#${this._chart_color}`;
                ctx.lineWidth = char_weight - 2;
                ctx.lineCap = 'round';
                ctx.arc(char_x, char_y, char_radius, Math.PI * 0.75,
                    Math.PI * (0.75 + (1.5 * Math.min(1, Math.max(0, value)))), false
                    );
                ctx.stroke();

                ctx.shadowBlur = 2;
                ctx.fillStyle = '#1f2c33';
                ctx.font = `${(char_radius * font_ratio)|0}px Arial`;
                ctx.textAlign = 'center';
                ctx.textBaseline = 'middle';
                ctx.fillText(`${this._value !== null ? this._value : t('No data')}${this._units}`,
                    char_x, char_y, size - char_shadow * 4 - char_weight * 2
                );

                ctx.fillStyle = '#768d99';
                ctx.font = `${(char_radius * font_ratio * .5)|0}px Arial`;
                ctx.textBaseline = 'top';

                ctx.textAlign = 'left';
                ctx.fillText(`${this._min}${this._min != '' ? this._units : ''}`,
                    char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                ctx.textAlign = 'right';
                ctx.fillText(`${this._max}${this._max != '' ? this._units : ''}`,
                    size - char_weight * .75, size - char_weight * 1.25, size / 2 - char_weight
                );

                requestAnimationFrame(animate);
            }
            else {
                this._last_value = this._value;
            }
        };

        requestAnimationFrame(animate);
    }
}
```

10. *manifest.json* を開き、以下を追加します。

- *assets/js* セクションの配列にファイル名（*class.widget.js*）
- *widget* セクションの *js_class* パラメータにクラス名（*WidgetLessonGaugeChart*）

これで、*WidgetLessonGaugeChart* クラスはダッシュボードとともに自動的に読み込まれるようになります。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
```

```
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "js": ["class.widget.js"]
    }
}
```

CSS スタイルをウィジェットに追加

このセクションでは、カスタム CSS スタイルを追加してウィジェットをより魅力的に見せる方法を学びます。

1. ウィジェットスタイルの場合は、*assets* ディレクトリに新しいディレクトリ *css* を作成します。
2. *assets/css* ディレクトリに *widget.css* ファイルを作成します。ウィジェット要素のスタイルを設定するには、セレクター *div.dashboard-widget-{widget id}* を使用します。ウィジェット全体の CSS を設定するには、セレクター *form.dashboard-widget-{widget id}* を使用します。

ui/modules/lesson_gauge_chart/assets/css/widget.css

```
div.dashboard-widget-lesson_gauge_chart {
    display: grid;
    grid-template-rows: 1fr;
    padding: 0;
}

div.dashboard-widget-lesson_gauge_chart .chart {
    display: grid;
    align-items: center;
    justify-items: center;
}

div.dashboard-widget-lesson_gauge_chart .chart canvas {
    background: white;
}

div.dashboard-widget-lesson_gauge_chart .description {
    padding-bottom: 8px;
    font-size: 1.750em;
    line-height: 1.2;
    text-align: center;
}

.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

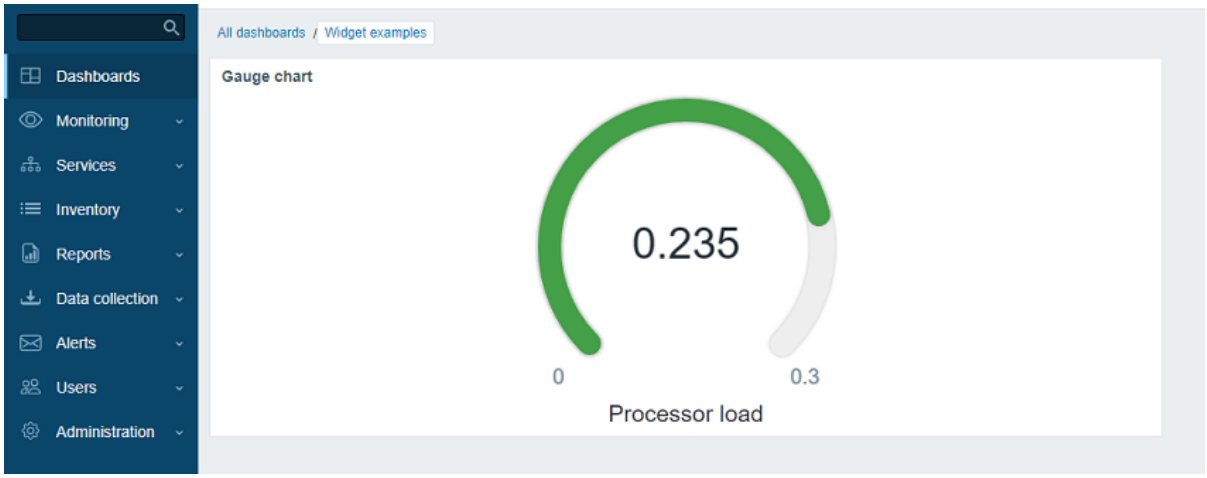
3. *manifest.json* を開き、CSS ファイル名 (*widget.css*) を *assets/css* セクションの配列に追加します。これにより、*widget.css* で定義された CSS スタイルをダッシュボードページに読み込むことができるようになります。

ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
```

```
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. ダッシュボードページを更新して、ウィジェットの完成版を確認します。



例

このセクションでは、カスタムモジュールのベースとして使用できる、サンプルモジュールとウィジェットのファイルを提供します。

モジュールの使用方法：

1. ZIP アーカイブをダウンロードします。
2. Zabbix Web インターフェースをインストールの *modules* ディレクトリ内の別のディレクトリにコンテンツを解凍します (例: */zabbix/ui/modules*)。
3. Zabbix Web インターフェースにモジュールを登録します。

モジュールの例

- ホストグループ作成時に、設定されたユーザーグループに読み取り権限を付与します - hg_auto_perm.zip

ウィジェットの例

- 最小限のウィジェット - widget_min.zip
- CSS のみを使用した”Hello, world” ウィジェット - hello_world_css.zip
- JavaScript のみを使用した”Hello, world” ウィジェット - hello_world_js.zip
- PHP を使用した”Hello, world” ウィジェット - hello_world_php.zip

Note:
Zabbix ネイティブウィジェットも例として利用できます。

Zabbix 用 Python ライブラリ

zabbix_utils は、Zabbix の公式 Python ライブラリです。

zabbix_utils では、次のことができます。

- Zabbix API を使用して Zabbix オブジェクトを管理する（ホストの作成、アイテムの更新、イベントの取得など）。
- Zabbix エージェントからデータを収集する（Zabbix get と同様）。
- Zabbix サーバーまたはプロキシにデータを送信する（Zabbix sender と同様）。

このライブラリは、同期および非同期のスクリプト実行の両方をサポートしており、シンプルなワークフローから、大量データ収集、一括エクスポート、アラート処理、効率的なシステムスケーリングといった大規模な並列処理まで対応できます。

このライブラリは、サポート対象の Zabbix サーバーのバージョンおよび最新の Python リリースと互換性があります。

利用を開始するには、まずライブラリをインストールし、その後クイックスタート ガイドに従ってください。

インストール

zabbix_utils は、以下で説明するいずれの方法でもインストールできます。

要件:

- Zabbix 6.0 以降（6.0、7.0、7.2、7.4 でテスト済み）
- Python 3.8 以降（3.8～3.13 でテスト済み）
- aiohttp 非同期 HTTP フレームワーク（非同期モードでのみ必要）

Python Package Index (PyPI)

これは、ほとんどの環境で最も一般的な方法です。

```
pip install zabbix_utils
pip install zabbix_utils[async] # 非同期サポートの場合のみ
```

Zabbix リポジトリ

システムのパッケージマネージャーを通じて依存関係を管理したい場合は、この方法を使用してください。

お使いの Linux ディストリビューション向けのZabbix パッケージをダウンロードし、次のコマンドを実行します。

RHEL およびその派生ディストリビューションの場合:

```
dnf install python3-zabbix-utils
dnf install epel-release # 非同期サポートの場合のみ
dnf install python3-aiohttp # 非同期サポートの場合のみ
```

Debian/Ubuntu およびその派生ディストリビューションの場合:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # 非同期サポートの場合のみ
```

ソース（GitHub）

最新の開発版を使用したい場合は、この方法を使用してください。

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # 非同期サポートの場合のみ
python3 setup.py install
```

非同期モードに必要な依存関係をインストールするには、上記で説明したいずれかの方法を使用することもできます。

クイックスタートガイド

zabbix_utils をインストールした後、スクリプトで使用できます。

このクイックスタートガイドでは、次の方法を紹介します。

- Zabbix エージェントからホスト名を取得する。
- Zabbix API を使用してホストとトラッパーアイテムを作成する。
- アイテムに値を送信する。

このガイドでは、スクリプトを段階的に示し、各部分を導入しながら説明します。完全なスクリプトはページの最後にあります。

また、このガイドでは、Zabbix サーバー、エージェント、および API がローカルで実行されていることを前提としています。

Zabbix エージェントからホスト名を取得する

まず、Zabbix エージェントが実行されているシステムのホスト名を取得します。このホスト名は、Zabbix API を使用してホストを作成する際に必要です。

1. zabbix_utils から `Getter` クラスをインポートします。このクラスはZabbix get と同様に動作し、Zabbix エージェントにデータを要求できます。

2. `127.0.0.1` のポート `10050` でローカルの Zabbix エージェントに接続する `Getter` インスタンスを作成します。

3. `Getter` インスタンスで `get()` メソッドを呼び出し、Zabbix エージェントからホスト名を要求します。

- `get()` メソッドは、アイテムキーをパラメータとして受け取り、そのアイテムに対するリクエストを Zabbix エージェントに送信します。
- `get()` メソッドはオブジェクトを返し、ホスト名はそのオブジェクトの `value` 属性に格納されます。
- `system.hostname` は、ホスト名を返す組み込みの Zabbix エージェントキーです。

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Zabbix API に接続してログインする

次に、Zabbix API に接続します。
これにより、スクリプトでホストやアイテムなどの Zabbix オブジェクトを管理できるようになります。

1. zabbix_utils から `ZabbixAPI` クラスをインポートします。

2. `ZabbixAPI` インスタンスを作成し、Zabbix Web インターフェースの URL を指定します。

3. `ZabbixAPI` インスタンスで `login()` メソッドを呼び出し、ユーザー名とパスワードを指定します。
アカウントには、Zabbix API にアクセスするための権限が必要です。

4. API 操作の完了後にセッションを閉じるため、`logout()` メソッドを呼び出します。

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### API operations go here

api.logout()
```

Zabbix でホストを作成

Zabbix エージェントからホスト名を取得し、Zabbix API に接続できたので、Zabbix で新しいホストを作成できます。

1. `ZabbixAPI` インスタンスで`host.create()` API メソッドを呼び出し、ホストの詳細を指定します。

- `host` - Zabbix エージェントから取得したホスト名を含む `hostname` 変数を設定します。
- `interfaces` - ホスト上で実行されている Zabbix エージェントの接続情報を含みます。
- `groups` - ホストが所属する少なくとも 1 つのホストグループを含みます。

2. Zabbix API は新しく作成されたホストの ID を返すため、この ID を後で使用できるよう `host_id` 変数に保存します。

3. ホストが作成されたことを確認するメッセージを表示します。

```
api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
```

```
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")
```

Zabbix で新しいアイテムを作成する

ホストを作成した後、そのホストにアイテムを追加できます。

1. 作成するアイテムに対して一意のキーを定義します。

2. ZabbixAPI インスタンスでitem.create() メソッドを呼び出し、アイテムの詳細を指定します。

- hostid - 作成したばかりのホストの ID を含む host_id 変数に設定します。
- name - アイテムの名前です。
- key_ - 先ほど定義した item_key 変数に設定します。
- type - 2 (トラッパーアイテム) に設定します。これは、後続の手順でスクリプトから送信される値を受信するために必要です。
- value_type - 3（符号なし数値）に設定します。これは、このアイテムに保存されるデータの種類です。

3. Zabbix API は新しく作成されたアイテムの ID を返すため、この ID を後で使用できるように item_id 変数に保存します。

4. アイテムが作成されたことを確認するメッセージを出力します。

```
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3,
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")
```

ホストに値を送信

ホストとアイテムを作成したので、そこにデータを送信できます。

1. time クラスをインポートし、データを送信する前に数秒待機します。
これにより、Zabbix が新しいホストとアイテムを完全に処理したことを確認できます。

2. zabbix_utils から Sender クラスをインポートします。
このクラスはZabbix sender と同様に動作し、スクリプトから Zabbix にデータを送信できます。

3. 127.0.0.1 のポート 10051 でローカルの Zabbix サーバーに接続する Sender インスタンスを作成します。

4. Sender インスタンスで send_value() メソッドを呼び出して、ホストのアイテムに値を送信し、次の詳細を指定します。

- hostname - Zabbix エージェントから取得したホスト名を含む hostname 変数に設定します。
- item_key - スクリプト内で先ほど定義した item_key 変数に設定します。
- 1 - 送信する値です。

4. 値が送信されたことを確認するメッセージを出力します。

```
import time
time.sleep(10)

from zabbix_utils import Sender

sender = Sender(server='127.0.0.1', port=10051)

response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {response}")
```

このコードが正常に実行されると、Zabbix が値を受信したことを示す成功メッセージが表示されます。

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

これで、Zabbix の Web インターフェース (*Monitoring* > 最新データ) でも値を確認できます。

完全なスクリプト

以下は、すべての手順を組み合わせた完全なスクリプトです。ホスト名の取得、Zabbix API を使用したホストとアイテムの作成、そして Zabbix へのデータ送信を行います。

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Zabbix エージェントからホスト名を取得
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Zabbix API に接続してログイン
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Zabbix にホストを作成
host = api.host.create(
    host=hostname,
    interfaces=[{
        'type': 1,
        'main': 1,
        'useip': 1,
        'ip': '127.0.0.1',
        'dns': '',
        'port': '10050',
    }],
    groups=[{'groupid': '2'}]
)

host_id = host['hostids'][0]

print(f"Host '{hostname}' created with ID {host_id}")

### Zabbix にアイテムを作成
item_key = 'app.myservice.heartbeat'

item = api.item.create(
    hostid=host_id,
    name='App heartbeat',
    key_=item_key,
    type=2,
    value_type=3
)

item_id = item['itemids'][0]

print(f"Item '{item_key}' created with ID {item_id}")

### API からログアウト
api.logout()

### Zabbix が新しいホストを処理するまで待機
time.sleep(10)

### ホストに値を送信
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {response}")
```

zabbix_utils を使用すると、Zabbix API を利用して、ホストの作成、アイテムの更新、イベントの取得など、Zabbix オブジェクトを管理できます。

API リクエストは、同期モードまたは非同期モードで実行できます。

- 同期モードでは、Python スクリプトはリクエストを送信し、応答を受信してから処理を続行します。これは、単純で順次的かつ予測可能な操作に適しています。
- 非同期モードでは、スクリプトは各応答を待たずにリクエストを送信するため、他の処理を並行して進めることができます。これは、応答に時間がかかるリクエストや大量のデータをまとめて処理する場合に、より効率的です。

このページの例では同期モードに重点を置いていますが、非同期モード でも同様のパターンに従います。追加の例は、GitHub リポジトリの zabbix_utils で利用できます。

インポート

Zabbix API を操作するために zabbix_utils を使用するには、スクリプトで ZabbixAPI クラスをインポートします。

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> すでにコミュニティの Python ライブラリを使用している場合は、通常、そのインポートを zabbix_utils の ZabbixAPI に置き換えることができます。

ログイン

API リクエストを行う前に、ZabbixAPI インスタンスを作成し、Zabbix API にログインする必要があります。

認証情報の管理方法に最も適した方法を選択してください。
ユーザー名とパスワード、またはAPI トークンを使用できます。

初期化時

ZabbixAPI インスタンスを作成する際に、Zabbix Web インターフェースの URL と認証情報をまとめて指定できます。

```
### ユーザー名とパスワード:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### API トークン:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

接続パラメータをまとめて扱いたい場合は、Python の辞書として渡し、Python のキーワード引数構文を使用して展開できます。

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

login() の使用

まず、Zabbix Web インターフェースの URL だけを指定して ZabbixAPI インスタンスを作成し、その後で認証情報を指定して login() メソッドを使用できます。

```
### ユーザー名とパスワード:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API トークン:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

環境変数の使用

まず、認証情報を環境変数として保存できます。たとえば、ターミナルで次のように設定します。

```
### ユーザー名とパスワード：
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### トークン：
export ZABBIX_TOKEN="your_api_token"
```

その後、同じターミナルセッションからスクリプトを開始する場合、スクリプト内の ZabbixAPI インスタンスに必要なのは Zabbix Web インターフェースの URL だけです。

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

URL も環境変数に保存できます。

```
### ユーザー名とパスワード：
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
export ZABBIX_URL="https://127.0.0.1/zabbix"

### トークン：
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

URL と認証情報がすべて環境変数として設定されている場合、スクリプト内の ZabbixAPI インスタンスには引数はまったく不要です。

```
api = ZabbixAPI()
```

ログアウト

ユーザー名とパスワードを使用してログインした場合は、API 操作の完了後に `logout()` メソッドを呼び出します。

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API actions go here

api.logout()
```

> **Note:**
> API トークンを使用する場合、`logout()` メソッドを呼び出す必要はありません。

API リクエスト

ログイン後、Zabbix API のメソッドリファレンスで説明されているメソッドを呼び出すことで、任意の API リクエストを実行できます。

API メソッドは、次の形式で呼び出します。

```
api_instance.zabbix_object.method(parameters)
```

> **Note:**
> 一部の Zabbix API メソッド名またはオブジェクト名では、Python の予約語である単語（例: `import`）が使用されています。Python のエラーを回避するには、Python で使用する際にメソッド名またはオブジェクト名の末尾にアンダースコアを追加してください（例: `api.configuration.import_`）。

たとえば、`host.get` を使用してホストの一覧を表示するには、次のようにします。

```
### 1. zabbix_utils から ZabbixAPI をインポート：
from zabbix_utils import ZabbixAPI

### 2. ZabbixAPI インスタンスを作成してログイン：
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")
```

```
### 3. フィルターに一致するホストを取得:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. 返されたホストを反復処理し、各ホストの詳細を表示:
for host in hosts:
    print(host)

### 5. セッションを閉じるため API からログアウト:
api.logout()
```

非同期モード

非同期モードでは、スクリプトは各 API リクエストの完了を待たずに API リクエストを送信できます。
これにより、多数の API リクエストを実行する必要がある場合や、一部のリクエストに長い時間がかかる場合に、スクリプトの効率を高めることができます。

非同期モードを使用する場合、同期モードと比べて重要な違いがあります。

- Python の `asyncio` モジュールをインポートします（最初に必要な依存関係をinstallする必要があります）。
- `ZabbixAPI` の代わりに `AsyncZabbixAPI` をインポートします。
- コードを `async` 関数内に記述します。
- `login()` および `logout()` を含む API メソッドの呼び出し時には `await` を使用します。

> **Attention:**
> AsyncZabbixAPI インスタンスを作成する際、初期化中にログインすることはできません。

たとえば、非同期モードでhost.getを使用してホストの一覧を表示するには、次のようにします。

```
### 1. 非同期モード用に asyncio と、zabbix_utils から AsyncZabbixAPI をインポートします:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. すべての API 操作を実行するメインの async 関数を定義します:
async def main():

    # 3. AsyncZabbixAPI インスタンスを作成してログインします（await が必要です）:
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. フィルターに一致するホストを取得します（await が必要です）:
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. 返されたホストを反復処理し、各ホストの詳細を表示します:
    for host in hosts:
        print(host)

    # 6. セッションを閉じるために API からログアウトします（await が必要です）:
    await api.logout()

### 7. asyncio のイベントループを使用して async main() 関数を実行します:
```

```
asyncio.run(main())
```

例

以下の例では、zabbix_utils ライブラリを使用した一般的な Zabbix API タスクを示します。

追加の例は、zabbix_utils GitHub リポジトリの examples ディレクトリで利用できます。

Zabbix エージェントからデータを収集する

概要

zabbix_utils を使用すると、Zabbix エージェントからデータを収集できます（Zabbix get と同様）。

データは同期モードまたは非同期モードで収集できます。

- 同期モードでは、Python スクリプトはデータを要求し、受信するまで待機してから処理を続行します。これは、単純で順次的かつ予測可能な操作に適しています。
- 非同期モードでは、スクリプトは各応答を待たずにデータを要求するため、他の処理を並行して進めることができます。これは、応答の遅い要求や大量のデータを扱う場合に、より効率的です。

このページの例は同期モードに重点を置いていますが、非同期モード も同様のパターンに従います。追加の例は、GitHub リポジトリの zabbix_utils で利用できます。

インポート

アイテムの値を収集するために zabbix_utils を使用するには、Python スクリプトで Getter クラスをインポートします。

```
from zabbix_utils import Getter
```

リクエストデータ

アイテムの値をリクエストするには:

1. Zabbix エージェントの IP アドレスとポートを指定して、Getter インスタンスを作成します。
2. Getter インスタンスで get() メソッドを呼び出し、取得したいアイテムのキーを指定します。

例えば、system.uname アイテムのデータをリクエストするには:

```
agent = Getter(host='192.0.2.0', port=10050)
response = agent.get('system.uname')
```

デフォルト以外の IP を使用する

スクリプトを実行しているサーバーが複数の IP アドレスを持つ場合、Zabbix エージェントへの接続時に Getter が使用する source_ip を指定できます。

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

timeout の使用

Getter に応答 timeout を設定して、スクリプトが Zabbix エージェントからの応答を待機する時間を制御できます。指定した時間を過ぎると待機を中止します。

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    timeout=30
)
```

暗号化の使用

Getter には組み込みの暗号化サポートは含まれていませんが、サードパーティライブラリを使用してラッパーを作成することで対応できます。

```
def psk_wrapper(sock, tls):
    # ...
    # ソケット用の TLS PSK ラッパーの実装
    # ...

agent = Getter(
    host='192.0.2.0',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

レスポンス

Zabbix エージェントからのレスポンスはライブラリによって処理され、AgentResponse オブジェクトとして返されます。

```
print(response)
### {
###     "error": null,
###     "raw": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64",
###     "value": "Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64"
### }

print(response.value)
### Linux zabbix_server 5.15.0-3.60.5.1.el9uek.x86_64

print(response.error)
### None
```

非同期モード

非同期モードを使用すると、スクリプトは各値の到着を待たずに値を収集できます。
これにより、多数の値を収集する必要がある場合や、一部の値の収集に長い時間がかかる場合に、スクリプトをより効率的にできます。

非同期モードを使用する場合、同期モードと比べて重要な違いがあります。

- Python の asyncio モジュールをインポートします（最初に必要な依存関係をinstallする必要があります）。
- Getter の代わりに AsyncGetter をインポートします。
- コードを async 関数内に記述します。
- get() メソッドを呼び出す際に await を使用します。

例えば、非同期モードを使用して単一の値を収集するには、次のようにします。

```
### 1. 非同期モード用に asyncio と、zabbix_utils から AsyncGetter をインポートします:
import asyncio
from zabbix_utils import AsyncGetter

### 2. すべてのデータ要求を実行するメインの async 関数を定義します:
async def main():
    agent = AsyncGetter(host='192.0.2.0', port=10050)

    # 3. Zabbix エージェントから system.uname の値を取得します（await が必要です）:
    response = await agent.get('system.uname')

    # 4. Zabbix エージェントが返した値を出力します:
    print(response.value)

### 5. asyncio のイベントループを使用して async main() 関数を実行します:
asyncio.run(main())
```

Zabbix サーバーまたはプロキシにデータを送信する

zabbix_utils を使用すると、Zabbix サーバーまたはプロキシ上のトラッパーアイテムにアイテムの値を送信できます（Zabbix sender と同様）。

単一の値、複数の値、さらには複数の Zabbix クラスターを対象に送信することもできます。

データは同期モードまたは非同期モードで送信できます。

- 同期モードでは、Python スクリプトは値を送信し、応答を待ってから処理を続行します。これは、単純で順次的かつ予測可能な操作に適しています。
- 非同期モードでは、スクリプトは各応答を待たずに値を送信するため、他の処理を並行して進めることができます。これは、応答の遅いリクエストや大量のデータのバッチ処理に対してより効率的です。

このページの例は同期モードに重点を置いていますが、非同期モード も同様のパターンに従います。追加の例は、GitHub リポジトリの zabbix_utils で利用できます。

インポート

アイテムの値を送信するために zabbix_utils を使用するには、スクリプトで Sender クラスをインポートします。

```
from zabbix_utils import Sender
```

複数の値を送信する場合は、ItemValue クラスもインポートできます。

```
from zabbix_utils import Sender, ItemValue
```

単一の値を送信

アイテムの値を送信するには、次のようにします。

1. Sender インスタンスを作成し、Zabbix サーバーまたはプロキシの IP アドレスとポートを指定します。
2. 次の形式で、Sender インスタンスの send_value() メソッドを呼び出します。

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

たとえば、Linux server ホストの service.status トラッパーアイテムに 1 を送信するには、次のようにします。

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

デフォルト以外の IP を使用する

スクリプトを実行しているサーバーが複数の IP アドレスを持つ場合、Zabbix server またはプロキシに値を送信する際に、Sender が使用する source_ip を指定できます。

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

タイムアウトの使用

Sender に応答 timeout を設定して、スクリプトが Zabbix server または proxy からの応答をあきらめるまでに、どれくらい待機するかを制御できます。

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

エージェント設定ファイルの使用

zabbix_utils に、ローカルの Zabbix エージェントまたは agent 2 の設定ファイルからServer またはServerActive パラメータを読み取らせることができます。この場合、Sender インスタンスの作成時に接続パラメータを指定する必要はありません。

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
ServerActive に複数のサーバーインスタンスを持つ 1 つ以上の Zabbix クラスターが含まれている場合、Sender は各クラスター内で最初に利用可能なサーバーにデータを送信します。ServerActive が設定されていない場合は、Server のアドレスがデフォルトポート (10051) とともに使用されます。

暗号化の使用

Sender には組み込みの暗号化サポートは含まれていませんが、サードパーティライブラリを使用してラッパーを作成することで提供できます。

```
def psk_wrapper(sock, tls):
    # ...
    # ソケット用の TLS PSK ラッパーの実装
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

単一値に対するレスポンス

Zabbix サーバーまたはプロキシから返されたレスポンスは、ライブラリによって処理され、TrapperResponse オブジェクトとして返されます。

```
print(response)
### {"processed": 1, "failed": 0, "total": 1, "time": "0.000123", "chunk": 1}

print(response.processed)
### 1

print(response.failed)
### 0

print(response.total)
### 1
```

非同期モード

非同期モードでは、Python スクリプトは Zabbix サーバーまたはプロキシからの応答を待たずに値を送信できます。
これにより、多数の値を送信する必要がある場合や、一部の値の送信に長い時間がかかる場合に、スクリプトの効率を高めることができます。

非同期モードを使用する場合、同期モードと比べていくつか重要な違いがあります。

- Python の asyncio モジュールをインポートします（最初に必要な依存関係をインストールする必要があります）。
- Sender の代わりに AsyncSender をインポートします。
- コードを async 関数の中に記述します。
- send_value() メソッドを呼び出す際に await を使用します。

例えば、非同期モードを使用して単一の値を送信するには、次のようにします。

```
### 1. 非同期モード用に asyncio と、zabbix_utils から AsyncSender をインポートします:
import asyncio
from zabbix_utils import AsyncSender

### 2. すべてのデータ送信操作（await が必要）が実行されるメインの async 関数を定義します:
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Zabbix サーバーまたはプロキシから返された応答を表示します:
    print(response)

### 4. asyncio のイベントループを使用して async main() 関数を実行します:
asyncio.run(main())
```

複数の値を送信

複数の値を送信するには:

1. ItemValue オブジェクトの配列を準備します。各オブジェクトは、send_value() メソッドと同じ形式を使用します。
2. Sender インスタンスを作成し、Zabbix サーバーまたはプロキシの IP アドレスとポートを指定します。

3. Sender インスタンスで send_value() の代わりに send() メソッドを呼び出し、送信する値を含むオブジェクトの配列を指定します。

たとえば、5 つの値を異なるホストに送信するには、次のようにします:

```
items = [
    ItemValue('server-de', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-fr', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-uk', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-nl', 'service.status', 'up', 1770887205, 100),
    ItemValue('server-pl', 'service.status', 'up', 1770887205, 100),
]

sender = Sender(server='127.0.0.1', port=10051)
response = sender.send(items)
```

カスタムチャンクサイズの使用

1 回のリクエストでトラッパーアイテムが受け付けられる数より多くの値を送信する必要がある場合は、それらをチャンクに分割できます。

デフォルトでは、チャンクサイズは 250 個の値です。Sender インスタンスの作成時に chunk_size パラメータを設定することで、これを変更できます。

例えば、5 個の値を 3 つのチャンク（2-2-1）で送信するには、chunk_size パラメータを 2 に設定します。

```
items = [
    ItemValue('server-de', 'service.status', 'up'),
    ItemValue('server-fr', 'service.status', 'up'),
    ItemValue('server-uk', 'service.status', 'up'),
    ItemValue('server-nl', 'service.status', 'up'),
    ItemValue('server-pl', 'service.status', 'up'),
]

sender = Sender(server='127.0.0.1', port=10051, chunk_size=2)
response = sender.send(items)
```

複数の Zabbix クラスターに値を送信する

複数の Zabbix クラスターに値を送信するには、次の手順を実行します。

1. Zabbix クラスターの配列を準備します。クラスターに複数のノードがある場合、値は各クラスターの最初の利用可能なノードに送信されます。
2. Zabbix クラスターの配列を指定して、Sender を作成します。
3. send_value() メソッドと同じ形式で、Sender インスタンスの send_value() メソッドを呼び出します。

たとえば、各クラスターの最初の利用可能なノードに値を送信するには、次のようにします。

```
zabbix_clusters = [
    ['zabbix.cluster1.node1', 'zabbix.cluster1.node2:10051'],
    ['zabbix.cluster2.node1:10051', 'zabbix.cluster2.node2', 'zabbix.cluster2.node3']
]

sender = Sender(clusters=zabbix_clusters)
response = sender.send_value('Linux server', 'service.status', 1)
```

複数の値に対するレスポンス

デフォルトでは、Sender はすべてのホストまたはクラスターにまたがる値の送信結果を集約して返します。

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

より詳細な情報が必要な場合は、response.details 属性を使用して、各クラスターおよび各チャンクの結果を確認できます。

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"{response.time} で値の送信に成功しました")
```

```
else:
    print(response.details)
    # {
    #     127.0.0.1:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000051",
    #             "chunk": 1
    #         }
    #     ],
    #     zabbix.example.local:10051: [
    #         {
    #             "processed": 1,
    #             "failed": 0,
    #             "total": 1,
    #             "time": "0.000057",
    #             "chunk": 1
    #         }
    #     ]
    # }

    for node, chunks in response.details.items():
        for resp in chunks:
            print(f"{node.address}:{node.port} で {resp.total} 件中 {resp.processed} 件を処理しました")
            # 127.0.0.1:10051 で 1 件中 1 件を処理しました
            # zabbix.example.local:10051 で 1 件中 1 件を処理しました
```

デバッグログ

Zabbix API の使用時、Zabbix エージェントからのデータ収集時、または Zabbix サーバーやプロキシへのデータ送信時に問題をトラブルシュートするには、デバッグログを有効にできます。

zabbix_utils ライブラリは、Python 標準の `logging` モジュールを使用します。

ライブラリからの詳細なデバッグメッセージを表示するには、次のようにします。

1. スクリプトで Python の `logging` モジュールをインポートします。
2. メッセージが読みやすい形式で出力されるようにロギングシステムを設定し、ロギングレベルを DEBUG に設定します。

```
import logging
from zabbix_utils import Getter

logging.basicConfig(
    format=u'[%(asctime)s] %(levelname)s %(message)s',
    level=logging.DEBUG
)

agent = Getter(host='127.0.0.1', port=10050)
resp = agent.get('system.uname')

print(resp.value)
```

デバッグログを有効にすると、リクエストとレスポンスに関する詳細情報が出力されます。例えば、次のようになります。

```
[2023-10-01 12:00:01,587] DEBUG Content of the packet: b'ZBXD\x01\x0c\x00\x00\x00\x00\x00\x00\x00system.un
[2023-10-01 12:00:01,722] DEBUG Zabbix response header: b'ZBXD\x01C\x00\x00\x00C\x00\x00\x00'
[2023-10-01 12:00:01,723] DEBUG Zabbix response body: Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
[2023-10-01 12:00:01,724] DEBUG Response from [127.0.0.1:10050]: Linux test_server 5.15.0-3.60.5.1.el9uek.
Linux test_server 5.15.0-3.60.5.1.el9uek.x86_64
```

プラグイン

概要　カスタムのロード可能なプラグインは、Zabbix エージェント 2 の機能を拡張します。これらは個別にコンパイルされますが、Zabbix エージェント 2 と共有されるパッケージを使用します。

各プラグインは、構造を定義し、1 つまたは複数のプラグインインターフェース (*Exporter*、*Configurator*、*Runner*) を実装する *Go* パッケージです。

ジャンプ先:

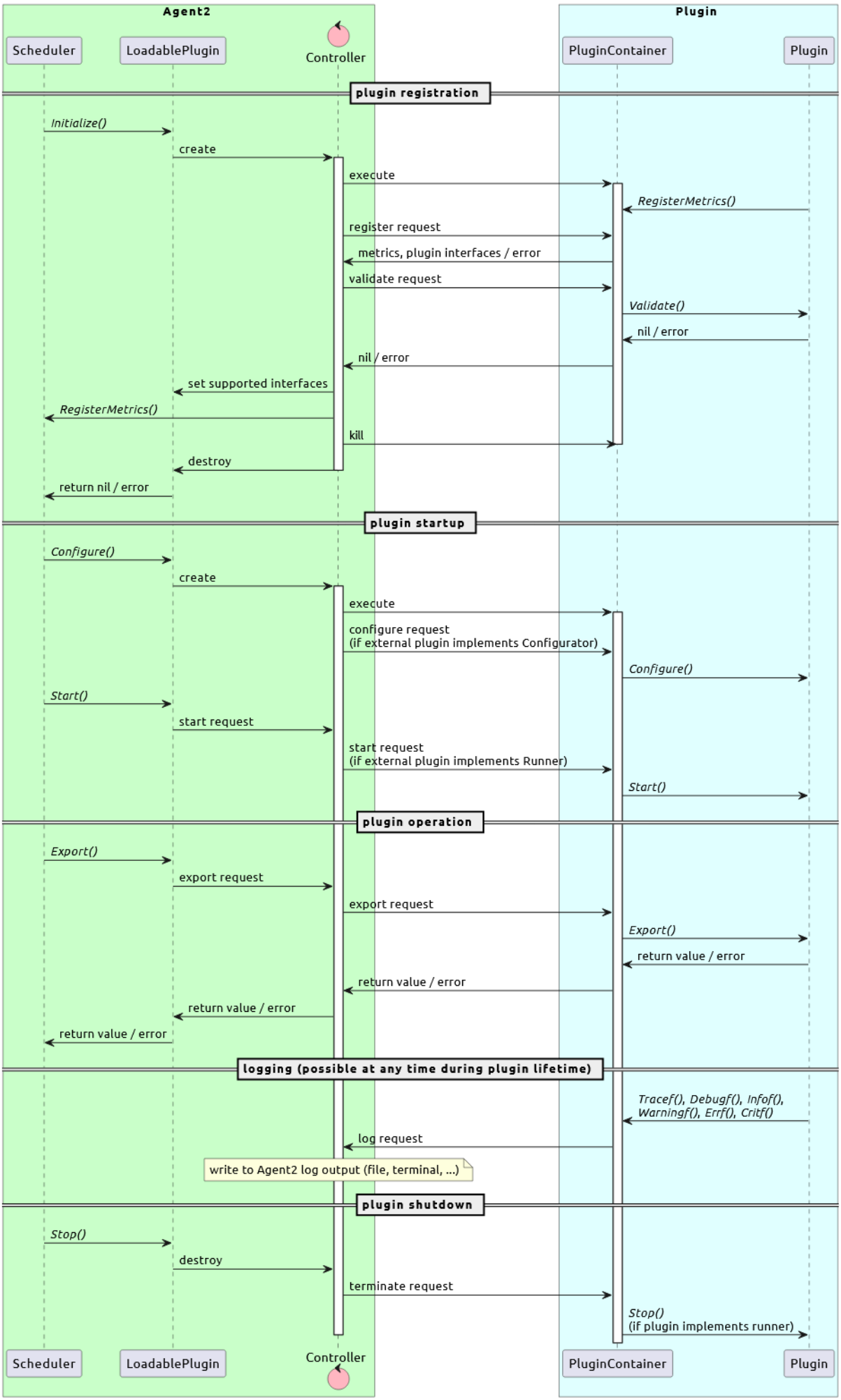
- 最初のプラグインを作成する
- プラグインインターフェース

参照:

- Zabbix エージェント 2 のプラグインの例

接続図　Zabbix エージェント 2 は、Linux では UNIX ソケット、Windows では名前付きパイプを使用して、プラグインに双方向に接続します。

以下の接続図は、Zabbix エージェント 2 とロード可能なプラグイン間の通信プロセスと、メトリクス収集プロセスを示しています。


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

例

リファレンスとして、いくつかの空のサンプルに加えて、Zabbix が提供する既存のロード可能なプラグインを使用できます。

- Zabbix エージェント 2 用サンプルプラグイン
- MSSQL プラグイン
- MongoDB プラグイン
- PostgreSQL プラグイン

Zabbix が提供するプラグインの完全な一覧を参照してください。

プラグインの作成 **(**チュートリアル**)**

これは、Zabbix エージェント 2 用のシンプルでロード可能なプラグインを作成するためのステップバイステップのチュートリアルです。

独自のプラグインを作成するためのテンプレートまたはガイドとして、サンプルリポジトリを使用することもできます。

作成するもの　このチュートリアルでは、新しいロード可能なプラグイン **MyIP** を作成する方法を説明します。このプラグインは、Zabbix エージェント 2 が動作しているホストの外部 IP アドレスを返す単一のアイテムキー **myip** を実装します。

ステップ **1:** セットアップ

1. プラグインは標準的な Go モジュールです。まず、プラグインの依存関係を追跡するために、プラグインディレクトリ内の go.mod ファイルを初期化します。

```
cd path/to/plugins/myip # プラグインディレクトリに移動します。
go mod init myip
```

2. 必須の依存関係である Zabbix Go SDK (golang.zabbix.com/sdk) をインストールします。

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

$LATEST_COMMIT_HASH を、適切なリリースブランチの golang.zabbix.com/sdkリポジトリにある最新の HEAD コミットハッシュに置き換えます。例:

```
go get golang.zabbix.com/sdk@af85407
```

golang.zabbix.com/sdk のバージョン管理は現在サポートされていませんが、将来変更される可能性があります。

必要に応じて、go get を使用して追加の依存関係をインストールできます。

3. プラグインのソースコード用の空の main.go ファイルを作成します。

```
touch main.go
```

これで初期設定が完了し、プラグインの開発準備が整いました。

ステップ **2:** プラグインの構造　前のステップでインストールした golang.zabbix.com/sdk モジュールは、プラグイン開発に必要なフレームワークを提供し、すべてのプラグインの一貫した構造を保証します。

1. 基本的な実行フローを設定します。

まず、プラグインのメインの実行フローを定義します。次のコードを main.go に追加します。

```
package main

func main() {
err := run()
if err != nil {
panic(err)
}
}

func run() error {
```

```
return nil
}
```

これで、プラグインの基本的な実行フローが確立されます。run 関数には、後でプラグインのコアロジックが含まれます。

2. プラグインのインターフェースを確認します。

Zabbix エージェント 2 プラグインは、golang.zabbix.com/sdk/plugin パッケージのインターフェースを実装する構造体で表現されます。

- *Accessor* - プラグイン名の設定やアイテムキーのタイムアウト処理など、すべてのプラグインが実装する必要がある必須メソッドを定義します。
- 以下の機能プラグインインターフェース (1 つ以上) :
  - *Exporter* - ポーリングを実行し、値 (複数可)、何も返さない、またはエラーを返します。多くの場合、*Collector* インターフェースと併用されます。
  - *Collector* - 定期的なデータ収集を管理します。
  - *Runner* - プラグインの起動およびシャットダウン手順を定義します。
  - *Watcher* - エージェントの内部スケジューラをバイパスし、独立したメトリックポーリングを実装できます。トラップベースまたはイベントドリブンの監視に便利です。
  - *Configurator* - プラグインが設定を読み込んで適用する方法を定義します。

これらのインターフェースは独自に実装することも、Zabbix Go SDK が提供するデフォルトのインターフェースを必要に応じて変更して使用することもできます。このチュートリアルではデフォルトの実装を使用します。

3. プラグイン構造体を作成します。

plugin パッケージをインポートし、plugin.Base 構造体を埋め込む myIP 構造体を作成します。

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
plugin.Base
}
```

myIP 構造体は現在、Accessor インターフェースを満たしています。関数型プラグインインターフェースの 1 つである Exporter を実装するメソッドは、このチュートリアルの後半で追加します。

ステップ **3:** アイテムキーの定義 プラグインは、データを収集し、Zabbix サーバーまたはプロキシに提供するためにアイテムキーが必要です。

1. エラー処理用の *errs* パッケージをインポートします。

```
import "golang.zabbix.com/sdk/errs"
```

2. run() 関数内の plugin.RegisterMetrics() 関数を使用して、アイテムキーを登録します。

```
func run() error {
p := &myIP{}

// `myip`アイテムキーを登録します。
err := plugin.RegisterMetrics(
p,
"MyIP",                             // プラグイン名
"myip",                             // アイテムキー名
"Returns the host's IP address.", // アイテムキーの説明
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

return nil
}
```

複数のアイテムキーを登録するには、各メトリックに対してパラメーター *metric name* と *description* を繰り返します。例:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

ステップ **4:** ハンドラーの設定 ハンドラーは、エージェントとプラグイン間の通信を容易にします。

1. *container* パッケージをインポートします。

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. run() 関数に、ハンドラーの作成と設定を行うコードを追加します。

```
func run() error {
p := &myIP{}

// `myip`アイテムキーを登録します。
err := plugin.RegisterMetrics(
p,
"MyIP",                              // プラグイン名
"myip",                              // アイテムキー名
"Returns the host's IP address.", // アイテムキーの説明
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

// 新しいハンドラーを作成します。
h, err := container.NewHandler("MyIP") // プラグイン名
if err != nil {
return errs.Wrap(err, "failed to create new handler")
}

// プラグインからエージェントへログを転送するためのログ記録を設定します。
// p.Logger.Infof、p.Logger.Debugf などから利用可能です。
p.Logger = h

// プラグインの実行を開始します。
// エージェントから終了要求を受信するまでブロックします。
err = h.Execute()
if err != nil {
return errs.Wrap(err, "failed to execute plugin handler")
}

return nil
}
```

ステップ **5:** データ収集の実装　データ収集は、Export メソッドを記述する Exporter インターフェースを介して行われます。

```
func Export(
  key string,              // 収集するアイテムキー
  params []string,         // アイテムキー（`myip[arg1, arg2]`）に渡される引数
  context ContextProvider // アイテムキーデータ収集に関するメタデータ
) (any, error)
```

1. HTTP リクエストとレスポンスの読み取りに必要なパッケージをインポートします。

```
import (
"io"
"net/http"
)
```

2. myIP 構造体の Export メソッドを実装します。

```
func (p *myIP) Export(
key string, params []string, context plugin.ContextProvider,
) (any, error) {
// プラグインは、`key`パラメーターに基づいて異なるデータ収集ロジックを使用できます。
    // この実装は、指定された`key`がサポートされているかどうかのみを検証します。
if key != "myip" {
return nil, errs.Errorf("unknown item key %q", key)
}
```

```
// ログはエージェント 2 のログに転送されます。
p.Infof(
"received request to handle %q key with %d parameters",
key,
len(params),
)

// データを収集して返します。

resp, err := http.Get("https://api.ipify.org")
if err != nil {
return nil, errs.Wrap(err, "failed to get IP address")
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "failed to read response body")
}

return string(body), nil
}
```

ステップ **6:** プラグインのビルドと設定

1. プラグインをビルドするには、次のコマンドを実行します。

```
go mod tidy
go build
```

これにより、カレントディレクトリに実行ファイル `myip` が作成されます。

2. プラグインを使用するように Zabbix エージェント 2 を設定します。

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

`$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` を、ステップ 5 で作成した `myip` へのパスに置き換えます。

設定パラメーター名 (このチュートリアルでは _MyIP_) のプラグイン名は、_plugin.RegisterMetrics()_ 関数で定義されたプラグイン名と一致する必要があります。

3. プラグインとその `myip` アイテムをテストするには、次のコマンドを実行します。

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

出力にはホストの外部 IP アドレスが含まれ、以下のようになります。

```
myip                                          [s|192.0.2.0]
```

これで、Zabbix エージェント 2 用のシンプルな読み込み可能なプラグインが作成されました。おめでとうございます！

```
package main

import (
"io"
"net/http"

"golang.zabbix.com/sdk/errs"
"golang.zabbix.com/sdk/plugin"
"golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
```

```
plugin.Base
}

func main() {
err := run()
if err != nil {
panic(err)
}
}

func run() error {
p := &myIP{}

// `myip`アイテムキーを登録します。
err := plugin.RegisterMetrics(
p,
"MyIP",                             // プラグイン名
"myip",                             // アイテムキー名
"Returns the host's IP address.", // アイテムキーの説明
)
if err != nil {
return errs.Wrap(err, "failed to register metrics")
}

// 新しいハンドラーを作成します。
h, err := container.NewHandler("MyIP") // Plugin name
if err != nil {
return errs.Wrap(err, "failed to create new handler")
}

// プラグインからエージェントにログを転送するためのログ設定を行います。
// p.Logger.Infof、p.Logger.Debugf などから利用可能です。
p.Logger = h

// プラグインの実行を開始します。
// エージェントからの終了要求を受信するまでブロックします。
err = h.Execute()
if err != nil {
return errs.Wrap(err, "failed to execute plugin handler")
}

return nil
}

func (p *myIP) Export(
key string, params []string, context plugin.ContextProvider,
) (any, error) {
// プラグインは、`key`パラメーターに基づいて異なるデータ収集ロジックを使用できます。
    // この実装では、指定された`key`がサポートされているかどうかのみ確認します。
if key != "myip" {
return nil, errs.Errorf("unknown item key %q", key)
}

// ログはエージェント 2 のログに転送されます。
p.Infof(
"received request to handle %q key with %d parameters",
key,
len(params),
)

// データを収集して返します。
```

```
resp, err := http.Get("https://api.ipify.org")
if err != nil {
return nil, errs.Wrap(err, "failed to get IP address")
}

defer resp.Body.Close()

body, err := io.ReadAll(resp.Body)
if err != nil {
return nil, errs.Wrap(err, "failed to read response body")
}

return string(body), nil
}
```

完全なソースコード

プラグインインターフェース

このセクションでは、利用可能なプラグインインターフェースについて説明します。

plugin.Exporter *Exporter* は、ポーリングを実行して値 (複数可)、何も返さない、またはエラーを返す最もシンプルなインターフェースです。これは、事前に解析されたアイテムキー、パラメーター、およびコンテキストを受け入れます。他のすべてのプラグインインターフェースへのアクセスは排他的であり、プラグインがすでにタスクを実行している場合はメソッドを呼び出すことはできません。また、プラグインごとに最大 100 回の同時 *Export()* 呼び出しの制限がありますが、これは各プラグインの要件に応じて減らすことができます。

plugin.Configurator *Configurator* インターフェースは、Zabbix エージェント 2 の設定ファイルからプラグイン設定パラメーターを提供します。

plugin.Runner *Runner* インターフェースは、プラグインの開始時 (アクティブ化) に初期化を実行し、プラグインの停止時 (非アクティブ化) に非初期化を実行する手段を提供します。たとえば、プラグインは Runner インターフェースを実装することで、バックグラウンドの *goroutine* を開始/停止できます。

拡張機能開発に関する変更

このページには、カスタム Zabbix 拡張機能の開発に関するすべての変更点（ある場合には）が記載されています。

Zabbix man ページ

利用可能なコマンド、オプション、使用例を一覧で示した、Zabbix のコマンドラインツールおよびサービスのリファレンスドキュメント。

zabbix_agent2

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_agent2 - Zabbix エージェント 2

概要

zabbix_agent2 [**-c** *config-file*]
zabbix_agent2 [**-c** *config-file*] [**-v**] **-p**
zabbix_agent2 [**-c** *config-file*] [**-v**] **-t** *item-key*
zabbix_agent2 [**-c** *config-file*] **-R** *runtime-option*
zabbix_agent2 [**-c** *config-file*] **-T**
zabbix_agent2 -h
zabbix_agent2 -V

説明

zabbix_agent2 は、様々なサービスのパラメーターを監視するためのアプリケーションです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに、別の *config-file* を使用します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

ランタイム制御オプション**:** **userparameter_reload**
設定ファイルからユーザーパラメーターを再ロードします。

loglevel increase
ログレベルを上げます。

loglevel decrease
ログレベルを下げます。

help
利用可能なランタイム制御オプションを一覧表示します。

metrics
利用可能なメトリクスを一覧表示します。

version
バージョンを表示します。

-p, **--print**
既知のアイテムを表示して終了します。各アイテムに対して、一般的なデフォルトが使用されるか、テスト用の特定のデフォルトが提供されます。これらのデフォルトは、アイテムキーのパラメーターとして角括弧内にリストされています。戻り値は角括弧で囲まれ、パイプ文字で区切られた戻り値の型が前に付けられます。ユーザーパラメーターの型は常に **t** となります。これは、エージェントが可能なすべての戻り値を決定できるわけではないためです。権限または環境が異なる可能性があるため、実行中のエージェントデーモンをクエリする場合、動作していると表示されるアイテムは、Zabbix サーバーまたは zabbix_get から動作することが保証されません。戻り値の型は以下の通りです。

d
小数部分を含む数値。

m
サポートされていません。これは、ログ監視アイテムや複数の収集値を必要とするアイテムなど、アクティブモードでのみ機能するアイテムをクエリしたことが原因である可能性があります。権限の問題や、ユーザーパラメーターが正しくない場合も、サポートされていない状態になる可能性があります。

s
テキスト。長さに制限はありません。

t
テキスト。**s** と同じです。

u
符号なし整数。

-t, **--test** *item-key*
単一のアイテムをテストして終了します。出力の説明については、**--print** を参照してください。

-v, **--verbose** アイテムをテストするとき、または既知のアイテムを印刷するときに、詳細出力を有効にします。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_agent2.conf
Zabbix エージェント 2 の設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

目次

名前
概要
説明
オプション
ファイル
参照
著者

ドキュメント作成日: 2021 年 11 月 22 日 14:07:57 GMT

zabbix_agentd

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_agentd - Zabbix エージェントデーモン

概要

zabbix_agentd [**-c** *config-file*]
zabbix_agentd [**-c** *config-file*] **-p**
zabbix_agentd [**-c** *config-file*] **-t** *item-key*
zabbix_agentd [**-c** *config-file*] **-R** *runtime-option*
zabbix_agentd [**-c** *config-file*] **-T**
zabbix_agentd -h
zabbix_agentd -V

説明

zabbix_agentd は、様々なサーバーのパラメーターを監視するためのデーモンです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-f, **--foreground**
Zabbix エージェントをフォアグラウンドで実行します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

ランタイム制御オプション　**userparameter_reload**
設定ファイルからユーザーパラメーターを再ロードします。

log_level_increase[=*target*]
ログレベルを上げます。ターゲットが指定されていない場合、全てのプロセスに影響します。

log_level_decrease[=*target*]
ログレベルを下げます。ターゲットが指定されていない場合、全てのプロセスに影響します。

ログレベル制御対象

process-type
指定されたタイプのすべてのプロセス (アクティブチェック、コレクター、リスナー)。

process-type,N
プロセスの種類と数 (例: listener,3)。

pid
プロセス ID。最大 65535。値がさらに大きい場合はターゲットを”process-type,N” と指定します。

-p, **--print**
既知のアイテムを表示して終了します。各アイテムに対して、一般的なデフォルトが使用されるか、テスト用の特定のデフォルトが提供されます。これらのデフォルトは、アイテムキーのパラメーターとして角括弧内にリストされています。戻り値は角括弧で囲まれ、パイプ文字で区切られた戻り値の型が前に付けられます。ユーザーパラメーターの型は常に **t** となります。これは、エージェントが可能なすべての戻り値を決定できるわけではないためです。権限または環境が異なる可能性があるため、実行中のエージェントデーモンをクエリする場合、動作していると表示されるアイテムは、Zabbix サーバーまたは zabbix_get から動作することが保証されません。戻り値の型は以下の通りです。

d
小数部分を含む数値。

m
サポートされていません。これは、ログ監視アイテムや複数の収集値を必要とするアイテムなど、アクティブモードでのみ機能するアイテムをクエリしたことが原因である可能性があります。権限の問題や、ユーザーパラメーターが正しくない場合も、サポートされていない状態になる可能性があります。

s
テキスト。長さに制限はありません。

t
テキスト。**s** と同じです。

u
符号なし整数。

-t, **--test** *item-key*
単一のアイテムをテストして終了します。出力の説明については、**--print** を参照してください。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
このヘルプを出力して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_agentd.conf Zabbix エージェントの設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

目次

名前
概要
説明
オプション
ファイル
参照
著者

ドキュメント作成日: 2021 年 11 月 22 日 20:50:13 GMT

zabbix_get

セクション: ユーザーコマンド (1)
更新日: 2021-06-01
目次 メインコンテンツに戻る

名前

zabbix_get - Zabbix get ユーティリティ

概要

zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-agent-cert-issuer** *cert-issuer*] [**--tls-agent-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -h
zabbix_get -V

説明

zabbix_get は、Zabbix エージェントからデータを取得するためのコマンドラインユーティリティです。

オプション

-s, **--host** *host-name-or-IP*
ホスト名またはホストの IP アドレスを指定します。

-p, **--port** *port-number*
ホスト上で動作するエージェントのポート番号を指定します。デフォルトは 10050 です。

-I, **--source-address** *IP-address*
ソース IP アドレスを指定します。

-t, **--timeout** *seconds*
タイムアウトを指定します。有効範囲: 1-30 秒 (デフォルト: 30)

-k, **--key** *item-key*
値を取得するアイテムのキーを指定します。

-P, **--protocol** *value* エージェントとの通信に使用されるプロトコル。値:

auto JSON プロトコルを使用して接続し、プレーンテキストプロトコルでフォールバックして再試行します (デフォルト)。

json JSON プロトコルを使用して接続します。

plaintext アイテムキーのみが送信されるプレーンテキストプロトコルを使用して接続します (6.4.x 以降のリリース)。

--tls-connect *value*
エージェントへの接続方法。値:

unencrypted
暗号化せずに接続します (デフォルト)。

psk
TLS と事前共有キーを使用して接続します。

cert
TLS と証明書を使用して接続します。

--tls-ca-file *CA-file*
ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス。

--tls-crl-file *CRL-file*
失効した証明書を含むファイルのフルパス。

--tls-agent-cert-issuer *cert-issuer*
許可されたエージェント証明書の発行者。

--tls-agent-cert-subject *cert-subject*
許可されたエージェント証明書のサブジェクト。

--tls-cert-file *cert-file*
証明書または証明書チェーンを含むファイルのフルパス。

--tls-key-file *key-file*
秘密鍵を含むファイルのフルパス。

--tls-psk-identity *PSK-identity*
PSK ID 文字列。

--tls-psk-file *PSK-file*
事前共有キーを含むファイルのフルパス。

--tls-cipher13 *cipher-string*
TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準をオーバーライドします。OpenSSL のバージョンが 1.1.1 未満の場合、このオプションは使用できません。

--tls-cipher *cipher-string*
GnuTLS 優先度文字列 (TLS 1.2 以降の場合) または OpenSSL 暗号文字列 (TLS 1.2 のみ)。デフォルトの暗号スイート選択基準をオーバーライドします。

-h, **--help**
ヘルプを出力して終了します。

-V, **--version**
バージョン情報を出力して終了します。

例

zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]”
zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]” --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-agent-cert-issuer ”CN=Signing CA,OU=IT operations,O=Example Corp,DC=example,DC=com” --tls-agent-cert-subject ”CN=server1,OU=IT operations,O=Example Corp,DC=example,DC=com” --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k ”system.cpu.load[all,avg1]” --tls-connect psk --tls-psk-identity ”PSK ID Zabbix agentd” --tls-psk-file /home/zabbix/zabbix_agentd.psk

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

目次

名前
概要
説明
オプション
例
参照
著者

ドキュメント作成日: 2021 年 6 月 11 日 08:42:29 GMT

zabbix_js

セクション: ユーザーコマンド (1)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_js - Zabbix JS ユーティリティ

概要

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

説明

zabbix_js は、組み込みスクリプトのテストに使用できるコマンドラインユーティリティです。

オプション

-s, **--script** *script-file*
実行するスクリプトのファイル名を指定します。ファイル名に'-' を指定するとスクリプトは標準入力から読み込まれます。

-p, **--param** *input-param*
入力パラメーターを指定します。

-i, **--input** *input-file*
入力パラメーターのファイル名を指定します。ファイル名に'-' を指定すると入力は標準入力から読み込まれます。

-l, **--loglevel** *log-level*
ログレベルを指定します。

-w, **--webdriver** *url* Web ドライバーの URL を指定します。

-t, **--timeout** *timeout*
タイムアウトを秒単位で指定します。有効範囲: 1-60 秒 (デフォルト: 10)

-h, **--help**
ヘルプを表示して終了します。

-V, **--version**
バージョン情報を表示して終了します。

例

zabbix_js -s script-file.js -p example

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

目次

名前
概要
説明
オプション
例
参照

ドキュメント作成日: 2020 年 3 月 18 日 21:23:35 GMT

zabbix_proxy

セクション: メンテナンスコマンド (8)
更新日: 2020-09-04
目次 メインコンテンツに戻る

名前

zabbix_proxy - Zabbix プロキシデーモン

概要

zabbix_proxy [**-c** *config-file*]
zabbix_proxy [**-c** *config-file*] **-R** *runtime-option*
zabbix_proxy [**-c** *config-file*] **-T**
zabbix_proxy -h
zabbix_proxy -V

説明

zabbix_proxy は、デバイスから監視データを収集し、Zabbix サーバーに送信するデーモンです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-f, **--foreground**
Zabbix プロキシをフォアグラウンドで実行します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

ランタイム制御オプション

config_cache_reload
設定キャッシュをリロードします。キャッシュが読込中の場合は無視されます。アクティブな Zabbix プロキシは Zabbix サーバーに接続し、設定データを要求します。(**-c** オプションが指定されていない場合) デフォルトの設定ファイルが PID ファイルの検索に使用され、シグナルがプロセスに送信され、PID ファイルにリストされます。

snmp_cache_reload
SNMP キャッシュをリロードします。

housekeeper_execute
ハウスキーパーを実行します。実行中の場合は無視されます。

diaginfo[=*section*]
指定されたセクションの内部診断情報をログに記録します。対象セクションは *historycache*、*preprocessing*、*locks* です。デフォルトではすべてのセクションの診断情報がログに記録されます。

log_level_increase[=*target*]
ログレベルを上げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。

log_level_decrease[=*target*]
ログレベルを下げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。

ログレベル制御対象

process-type
指定されたタイプのすべてのプロセス (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector)。

process-type,N
プロセスの種類と数 (例: poller,3)。

pid
プロセス ID。最大 65535。値が大きい場合はターゲットを”process-type,N” と指定します。

制御対象のプロファイリング

process-type
指定されたタイプのすべてのプロセス (configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, preprocessing manager, icmp pinger, ipmi manager, ipmi poller, java poller, poller, agent poller, http agent poller, snmp poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector, history poller, availability manager, odbc poller)。

process-type,N
プロセスの種類と数 (例: history syncer,1)。

pid
プロセス ID。最大 65535。値が大きい場合はターゲットを”process-type,N” と指定します。

scope プロファイリング対象 (rwlock、mutex、processing) はプロセスタイプで使用可能 (例: history syncer,1,processing)。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
ヘルプを出力して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_proxy.conf Zabbix プロキシの設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

目次

名前
概要
説明
オプション
ファイル
参照
著者

ドキュメント作成日: 2020 年 9 月 4 日 16:12:22 GMT

zabbix_sender

セクション: ユーザーコマンド (1)
更新日: 2021-06-01
目次 メインコンテンツに戻る

名前

zabbix_sender - Zabbix sender ユーティリティ

概要

zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert** **--tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk** **--tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender -h
zabbix_sender -V

説明

zabbix_sender は、監視データを Zabbix サーバーまたはプロキシに送信するためのコマンドラインユーティリティです。Zabbix サーバーでは、**Zabbix trapper** タイプのアイテムに対応するキーで作成する必要があります。受信データは、このアイテムの許可されたホストフィールドで指定されたホストからのみ受け入れられることに注意してください。

オプション

-c, **--config** *config-file*
config-file を使用します。**Zabbix sender** は agentd 設定ファイルからサーバーの詳細を読み取ります。デフォルトでは、**Zabbix sender** は設定ファイルを読み取りません。サポートされるのは **Hostname**、**ServerActive**、**SourceIP**、**TLSConnect**、**TLSCAFile**、**TLSCRLFile**、**TLSServerCertIssuer**、**TLSServerCertSubject**、**TLSCertFile**、**TLSKeyFile**、**TLSPSKIdentity**、**TLSPSKFile** パラメータのみです。
HostnameItem パラメータで定義されたホスト名は取得されません。この場合、ホスト名はコマンドラインで指定する必要があります（**-s** オプションを参照）。
エージェントの **ServerActive** 設定パラメータで定義されたすべてのアドレスが、データ送信に使用されます。あるアドレスへのバッチデータの送信に失敗した場合、以降のバッチはそのアドレスには送信されません。

-z, **--zabbix-server** *server*
Zabbix サーバーのホスト名または IP アドレスを指定します。ホストがプロキシによって監視されている場合は、代わりにプロキ

シのホスト名または IP アドレスを使用する必要があります。**--config** と一緒に使用した場合、agentd 設定ファイルで指定された **ServerActive** パラメータのエントリを上書きします。

-p, **--port** *port*
サーバー上で動作している Zabbix server trapper のポート番号を指定します。デフォルトは 10051 です。**--config** と一緒に使用した場合、agentd 設定ファイルで指定された **ServerActive** パラメータのポートエントリを上書きします。

-I, **--source-address** *IP-address*
送信元 IP アドレスを指定します。**--config** と一緒に使用した場合、agentd 設定ファイルで指定された **SourceIP** パラメータを上書きします。

-t, **--timeout** *seconds*
タイムアウトを指定します。有効範囲: 1～300 秒（デフォルト: 60）

-s, **--host** *host*
値を送信するアイテムが属するホスト名を指定します（Zabbix Web インターフェースに登録されている名前）。ホストの IP アドレスや DNS 名は使用できません。**--config** と一緒に使用した場合、agentd 設定ファイルで指定された **Hostname** パラメータを上書きします。

-k, **--key** *key*
値の送信先となるアイテムキーを指定します。

-o, **--value** *value*
アイテムの値を指定します。

-i, **--input-file** *input-file*
入力ファイルから値を読み込みます。標準入力から値を読み込むには、**<input-file>** として **-** を指定します。ファイルの各行には、空白区切りで **<hostname> <key> <value>** を記述します。各値はそれぞれ別の行に指定する必要があります。各行には、空白区切りの 3 つのエントリ **<hostname> <key> <value>** を含める必要があります。ここで、”hostname” は Zabbix Web インターフェースに登録されている監視対象ホスト名、”key” は対象アイテムのキー、”value” は送信する値です。agent 設定ファイルまたは **--host** 引数のホスト名を使用するには、**<hostname>** として **-** を指定します。

入力ファイルの 1 行の例:

”Linux DB3” db.connections 43

値の型は、Zabbix Web インターフェースのアイテム設定で正しく設定されている必要があります。
Zabbix sender は 1 回の接続で最大 250 個の値を送信します。
入力ファイルから値を送信する際のサイズ制限は、Zabbix 通信プロトコルで定義されているサイズに依存します。
入力ファイルの内容は UTF-8 エンコーディングである必要があります。
入力ファイル内のすべての値は、上から順に順次送信されます。
エントリは、次のルールに従って書式設定する必要があります:

- 引用符付きエントリと引用符なしエントリの両方がサポートされます。
- 二重引用符が引用文字です。
- 空白を含むエントリは引用符で囲む必要があります。
- 引用符付きエントリ内の二重引用符およびバックスラッシュ文字は、バックスラッシュでエスケープする必要があります。
- 引用符なしエントリではエスケープはサポートされません。
- 改行エスケープシーケンス (\n) は引用符付き文字列でサポートされます。
- 改行エスケープシーケンスは、エントリの末尾からは削除されます。

-T, **--with-timestamps**
このオプションは **--input-file** オプションと一緒にのみ使用できます。

入力ファイルの各行には、空白区切りの 4 つのエントリ
<hostname> <key> <timestamp> <value> を含める必要があります。
timestamp は Unix タイムスタンプ形式で指定する必要があります。対象アイテムを参照するトリガーがある場合、すべての timestamp は昇順でなければなりません。そうでないと、イベント計算は正しく行われません。

入力ファイルの 1 行の例:

”Linux DB3” db.connections 1429533600 43

詳細については、**--input-file** オプションを参照してください。

タイムスタンプ付きの値が「データなし」メンテナンスタイプのホストに送信された場合、その値は破棄されます。ただし、期限切れのメンテナンス期間に対するタイムスタンプ付きの値を送信することは可能であり、その場合は受け付けられます。

-N, **--with-ns**
このオプションは **--with-timestamps** オプションと一緒にのみ使用できます。

入力ファイルの各行には、空白区切りの 5 つのエントリ
<hostname> <key> <timestamp> <ns> <value> を含める必要があります。

入力ファイルの 1 行の例:

”Linux DB3” db.connections 1429533600 7402561 43

詳細については、**--input-file** オプションを参照してください。

-r, **--real-time**
受信した値を、受信後すぐに 1 つずつ送信します。これは標準入力から読み込む場合に使用できます。

-g, **--group**
値をホストごとにグループ化し、各ホストへ個別のバッチとして送信します。

--tls-connect *value*
サーバーまたはプロキシへの接続方法。値:

unencrypted 暗号化なしで接続します (デフォルト)。

psk TLS と事前共有キーを使用して接続します。

cert TLS と証明書を使用して接続します。

--tls-ca-file *CA-file* ピア証明書検証用の最上位 CA 証明書を含むファイルのフルパス名。

--tls-crl-file *CRL-file* 失効した証明書を含むファイルのフルパス名。

--tls-server-cert-issuer *cert-issuer* 許可されたサーバー証明書の発行者。

--tls-server-cert-subject *cert-subject* 許可されたサーバー証明書のサブジェクト。

--tls-cert-file *cert-file* 証明書または証明書チェーンを含むファイルのフルパス名。

--tls-key-file *key-file* 秘密鍵を含むファイルのフルパス名。

--tls-psk-identity *PSK-identity* PSK ID 文字列。

--tls-psk-file *PSK-file* 事前共有キーを含むファイルのフルパス名。

--tls-cipher13 *cipher-string* TLS 1.3 の OpenSSL 1.1.1 以降の暗号文字列。デフォルトの暗号スイート選択基準をオーバーライドします。OpenSSL のバージョンが 1.1.1 未満の場合、このオプションは使用できません。

--tls-cipher *cipher-string* GnuTLS 優先度文字列 (TLS 1.2 以降の場合) または OpenSSL 暗号文字列 (TLS 1.2 のみ)。デフォルトの暗号スイート選択基準をオーバーライドします。

-v, **--verbose** 冗長化モード。詳細は**-vv** を参照して下さい。

-h, **--help** ヘルプを出力して終了します。

-V, **--version** バージョン情報を出力して終了します。

終了ステータス

値が送信され、それらすべてがサーバーによって正常に処理された時の終了ステータスは 0 です。データが送信されたが、少なくとも 1 つの値の処理が失敗した時の終了ステータスは 2 です。データ送信が失敗した時の終了ステータスは 1 です。

例

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

監視対象ホストの **mysql.queries** アイテムの値として **342.45** を送信します。エージェントの設定ファイルで定義された監視対象ホストと Zabbix サーバーを使用します。

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s ”Monitored Host” -k mysql.queries -o 342.45

エージェントの設定ファイルで定義された Zabbix サーバーを使用して **Monitored Host** ホストの **mysql.queries** アイテムの値として **342.45** を送信します。

zabbix_sender -z 192.168.1.113 -i data_values.txt

ファイル **data_values.txt** から IP **192.168.1.113** の Zabbix サーバーに値を送信します。ホスト名とキーはファイルで定義されます。

echo ”- hw.serial.number 1287872261 SQ4321ASDF” | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

エージェントの設定ファイルで指定されたコマンドラインから Zabbix サーバーにタイムスタンプ付きの値を送信します。入力データのダッシュは、ホスト名も同じ設定ファイルから使用する必要があることを示します。

echo ’”Zabbix server” trapper.item ””’ | zabbix_sender -z 192.168.1.113 -p 10000 -i -

コマンドラインから IP アドレス **192.168.1.113** のポート **10000** でリッスンしている Zabbix サーバーに空のアイテムの値を送信します。空の値は、空の二重引用符で示します。

zabbix_sender -z 192.168.1.113 -s ”Monitored Host” -k mysql.queries -o 342.45 --tls-connect cert --tls-ca-file /home/zabbix/zabbix_ca_file --tls-cert-file /home/zabbix/zabbix_agentd.crt --tls-key-file /home/zabbix/zabbix_agentd.key

*Monitored Host** ホスト * の **mysql.queries** アイテムの値として **342.45** を、証明書付きの TLS を使用して IP **192.168.1.113** のサーバーに送信します。

zabbix_sender -z 192.168.1.113 -s ”Monitored Host” -k mysql.queries -o 342.45 --tls-connect psk --tls-psk-identity ”PSK ID Zabbix agentd” --tls-psk-file /home/zabbix/zabbix_agentd.psk

Monitored Host ホストの **mysql.queries** アイテムの値として **342.45** を IP **192.168.1.113** のサーバーに、TLS と事前共有キー (PSK) を使用して送信します。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

目次

名前
概要
説明
オプション
終了ステータス
例
参照
著者

ドキュメント作成日: 2021 年 6 月 11 日 08:42:39 GMT

zabbix_server

セクション: メンテナンスコマンド (8)
更新日: 2020-09-04
目次 メインコンテンツに戻る

名前

zabbix_server - Zabbix サーバーデーモン

概要

zabbix_server [**-c** *config-file*]
zabbix_server [**-c** *config-file*] **-R** *runtime-option*
zabbix_server [**-c** *config-file*] **-T**
zabbix_server -h
zabbix_server -V

説明

zabbix_server は、Zabbix ソフトウェアのコアデーモンです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-f, **--foreground**
Zabbix サーバーをフォアグラウンドで実行します。

-R, **--runtime-control** *runtime-option*
runtime-option に従って管理機能を実行します。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
このヘルプを表示して終了します。

-V, **--version**
バージョン情報を表示して終了します。

コマンドラインパラメーターを使用した Zabbix サーバーの実行例:

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

ランタイム制御

ランタイム制御オプション:

config_cache_reload
設定のキャッシュをリロードします。キャッシュが読み込み中の場合は無視されます。(**-c** オプションが指定されていない場合) デフォルトの設定ファイルを使用して PID ファイルの検索し、PID ファイルにリストされたプロセスにシグナルを送信します。

snmp_cache_reload
SNMP キャッシュを再ロードし、すべてのホストの SNMP プロパティ (エンジン時間、エンジンの起動、エンジン ID、資格情報) をクリアします。

housekeeper_execute
ハウスキーパーを実行します。ハウスキーパーが実行中の場合は無視します。

trigger_housekeeper_execute
トリガーハウスキーパーを実行します。トリガーハウスキーパーが実行中の場合は無視されます。

diaginfo[=*section*]
指定されたセクションの内部診断情報をログに記録します。セクションには、*historycache*、*preprocessing*、*alerting*、*lld*、*valuecache*、*locks* を指定できます。デフォルトでは、すべてのセクションの診断情報がログに記録されます。

ha_status
高可用性 (HA) クラスターのステータスをログに記録します。

ha_remove_node[=*target*]
名前または ID で指定された高可用性 (HA) ノードを削除します。アクティブ/スタンバイ状態のノードは削除できません。

ha_set_failover_delay[=*delay*]
高可用性 (HA) フェールオーバーの遅延時間を設定します。時刻サフィックスがサポートされています。例: 10s、1m。

proxy_config_cache_reload[=*target*] プロキシ設定のキャッシュをリロードします。

secrets_reload
Vault からシークレットをリロードします。

service_cache_reload サービスマネージャーのキャッシュをリロードします。

prof_enable[=*target*]
プロファイリングを有効にします。ターゲットが指定されていない場合、すべてのプロセスに影響します。プロファイリングを有効にすると、関数名からすべての rwlocks/mutexes の詳細を見ることができます。

prof_disable[=*target*]
プロファイリングを無効にします。ターゲットが指定されていない場合、すべてのプロセスに影響します。

log_level_increase[=*target*]
ログレベルを上げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。

log_level_decrease[=*target*]
ログレベルを下げます。ターゲットが指定されていない場合、すべてのプロセスに影響します。

ログレベル制御の対象

process-type
指定したタイプのすべてのプロセス（alerter、alert manager、availability manager、browser poller、configuration syncer、configuration syncer worker、connector manager、connector worker、discovery manager、escalator、ha manager、history poller、history syncer、housekeeper、http poller、icmp pinger、internal poller、ipmi manager、ipmi poller、java poller、odbc poller、poller、agent poller、http agent poller、snmp poller、preprocessing manager、proxy group manager、proxy poller、self-monitoring、service manager、snmp trapper、task manager、timer、trapper、unreachable poller、vmware collector）

process-type,N
プロセスタイプと番号（例: poller,3）

pid
プロセス識別子。65535 まで指定できます。これより大きい値の場合は、対象を”process-type,N” として指定してください

プロファイリング制御の対象

process-type
指定したタイプのすべてのプロセス（alerter、alert manager、availability manager、browser poller、configuration syncer、configuration syncer worker、connector manager、connector worker、discovery manager、escalator、ha manager、history poller、history syncer、housekeeper、http poller、icmp pinger、internal poller、ipmi manager、ipmi poller、java poller、odbc poller、poller、agent poller、http agent poller、snmp poller、preprocessing manager、proxy group manager、proxy poller、self-monitoring、service manager、snmp trapper、task manager、timer、trapper、unreachable poller、vmware collector）

process-type,N
プロセスタイプと番号（例: history syncer,1）

pid
プロセス識別子。65535 まで指定できます。これより大きい値の場合は、対象を”process-type,N” として指定してください

scope
プロファイリングスコープ（rwlock、mutex、processing）。プロセスタイプと組み合わせて使用できます（例: history syncer,1,processing）

ファイル

/usr/local/etc/zabbix_server.conf Zabbix サーバーの設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

目次

名前

概要

説明

オプション

ファイル

参照

著者

ドキュメント作成日: 2020 年 9 月 4 日 16:12:14 GMT

zabbix_web_service

セクション: メンテナンスコマンド (8)
更新日: 2019-01-29
目次 メインコンテンツに戻る

名前

zabbix_web_service - Zabbix web サービス

概要

zabbix_web_service [**-c** *config-file*]
zabbix_web_service [**-c** *config-file*] **-T**
zabbix_web_service -h
zabbix_web_service -V

説明

zabbix_web_service は、Zabbix コンポーネントに Web サービスを提供するためのアプリケーションです。

オプション

-c, **--config** *config-file*
デフォルトの設定ファイルの代わりに *config-file* で指定されたファイルを使用します。

-T, **--test-config** 設定ファイルを検証して終了します。

-h, **--help**
ヘルプを表示して終了します。

-V, **--version**
バージョン情報を出力して終了します。

ファイル

/usr/local/etc/zabbix_web_service.conf
Zabbix Web サービスの設定ファイルのデフォルト格納先 (コンパイル時に変更されていない場合)。

参照

ドキュメント https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

目次

名前

概要

説明

オプション

ファイル

参照

著者

ドキュメント作成日: 2021 年 6 月 11 日 12:58:30 GMT

著作権に関する通知

Zabbix ドキュメントは、AGPL-3.0 ライセンスの下では配布されません。
Zabbix ドキュメントの利用には、以下の条件が適用されます。

このドキュメントの印刷物は、個人的な利用に限り作成できます。
実際の内容をいかなる形でも変更または編集しない限り、他の形式への変換は許可されます。
このドキュメントは、Zabbix が配布する方法と同様の方法（すなわち、Zabbix の Web サイト上で電子的にダウンロード用として提供する方法）で配布する場合、または USB もしくは同様の媒体で配布する場合を除き、いかなる形式または媒体でも公開または配布してはなりません。ただし、その場合であっても、ドキュメントは同じ媒体上でソフトウェアとともに配布されなければなりません。
印刷物の配布、またはこのドキュメントの全部もしくは一部を他の出版物で使用することを含むその他の利用には、Zabbix の正式な代表者による事前の書面による同意が必要です。
Zabbix は、上記で明示的に許諾された権利を除く、本ドキュメントに関する一切の権利を留保します。